MSSE
SOFTWARE, INC.

Development Verification Test Plan
for GolfScore

Release 1.1

Lawrence Osagie Aluya

Learner/Software Tester

2/15/2025

Contents

1.0 Introduction

1.1. Objective

This document describes the test plan for GolfScore release 1.1. It includes information on what is to be tested and the test methodology.

Specifically, this document describes the tests to be performed, the testing assumptions/dependencies, testing requirements, test tools, resource requirement, test schedule, risks/mitigation and metrics.

1.2. Project Description

The main purpose of this test is to verify the requirements for GolfScore release 1.1 as set forth in GolfScore SRS/Design Document Revision 1.1.

It is required to verify that the program can process scores from a golf tournament and produce reports showing who won the tournament and how the golfers performed on each course played.

1.3. Process Tailoring

This test will check the **functionality** and **Performance** of the GolfScore release 1.1 program against the requirements as specified in the SRS Revision 1.1.

We will follow through the Agile methodology for testing which will entail defragmenting the workload into sprints.

The software testing lifecycle will also be adhered to while observing the key testing principles throughout the test cycle.

Referenced Document

MSSE SOFTWARE, INC. – Software Requirements Specifications/ Design Document Revision 1.1

2.0 Assumptions/Dependencies

In order to test the GolfScore program, the following pre-conditions need to be met:

1. The program must be written in C or C++
2. It must be running on windows 2000 or any later version
3. The program can only run on a command line interface (CLI), no Graphical User Interface (GUI) associated with the application
4. Inputs to the program are stored in an input record file
5. Outputs from the program should be stored in an output record file

3.0 Test Requirements

This is a Development Verification Test. It is required to test if the GolfScore program can effectively and efficiently generate the reports of golfer's results for a golf tournament by verifying the following:

1. dentification: The program should display its title and release number on the screen at execution time

2. **The program functionality using the tournament assumptions which include**:

   a. The golf courses specified for the tournament can be from 1 to 5
   b. The number of golfer's entered in the tournament can be from 2 to 12
   c. Each golf course has 18 holes, and par for each hole is either 3,4 or 5 strokes
   d. A golfer's score for each hole is determined as shown in Section 2.3.2 and is based on the number of strokes under or over par taken to complete that hole. Thus, score and stroke count are not the same.
   e. A golfer's stroke count for a particular golf course is the sum of the stroke counts for each of the 18 holes.
   f. A golfer's total tournament score is the sum of his or her scores for all courses played.

3. **The GolfScore program can be called from the command line interface using the following format:**

   >golf options filename output-directory
   Options- one hyphen followed by one or more alphabetic characters, with no commas or spaces

   -h Display help information on the screen; filename and output-directory are ignored, if present
   -c, -t, -g Generate the Course Report (-c), Tournament Ranking Report (-t), or the Golfer Report (-g), respectively.

   Options may be combined, e.g. –cg

   Filename- The name of the input file containing the data for the reports. This may be a fully-qualified name; if no path name is supplied, the path that the program was executed from will be used.

   output-directory- The name of the file directory or path where the output files should be placed. If no path name is specified, the path containing filename will be used.

4. **For any input parameter errors, including unrecognizable options, the program will stop and display a message explaining the error**

5. **Input data will be checked for errors relating to numeric data in non-numeric fields and vice versa, par values that are not 3,4 or 5 and any golfer with duplicate records will have only the first record processed. The duplicate records should be ignored**

6. **The output file should be checked if the requested output already exists and there should be a prompt asking the user if the file should be overwritten**.

7. **The performance of the program (speed of execution)**

4.0 Test Tools

A lot of hardware and software test tools are available in the marketplace for this test, but we will use the following tools for this test:

a. Project management and bug tracking tool (Jira)

b. Report management tool (Confluence)

c. Hardware monitor

d. Cucumber

5.0 Resource Requirements

This Development Verification Test will require the following resources:

- A high-speed processor with 2000 or later version of windows OS
- GCC -The GNU compiler collection for and C++
- An open-source Test case and bug management tool (Atlassian Jira)
- An open-source report management tool (Atlassian Confluence)
- Available Development Verification Test Personnel

6.0 Test Schedule

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

Test Sequence	**Start Date**	**# of Days**	**Finish Date**
Test Development	12/14/2024	61	2/13/2025
Module Availability	2/14/2025	0	N/A
Main Test	2/15/2025	45	4/1/2025
Regression Testing	4/3/2025	33	5/6/2025

7.0 Risks/Mitigation

Some of the potential risk in this testing project are as follows:

1. Lack of proper testing tools.
2. Overlooking critical functionalities
3. Failing to consider the external factors that could impact the program performance
4. Schedule delays
5. Poor Communication between teams

All these risks can be mitigated by ensuring the SRS is well read, comprehended and discussed during the sprint planning meeting.

8.0 Metrics

The status and progress of this test will be recorded through the collection of various sets of data.
The following metrics data will be collected. Some will be collected prior to, and some after the application release

Prior to release:

Efforts expended during Development Verification Test (DVT)

Number of defects found during the Test

Test Tracking S-Curve to visualize the progress of the test project

After release:

Number of defects found in the program during the test

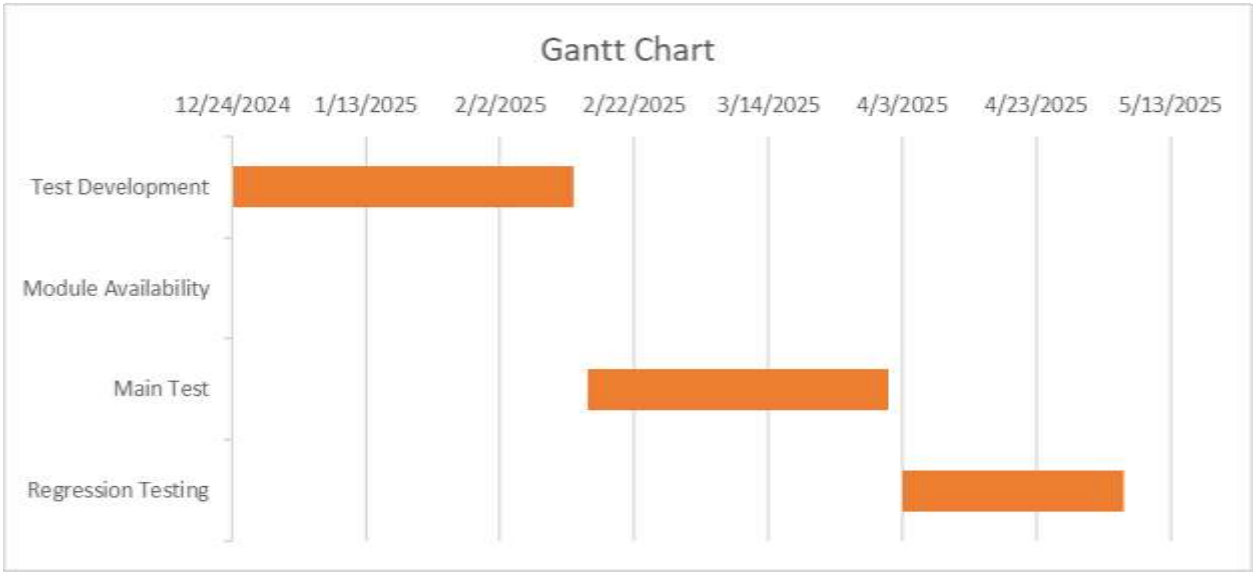
Size of the application or program

Appendix A – Detailed Resource Requirements

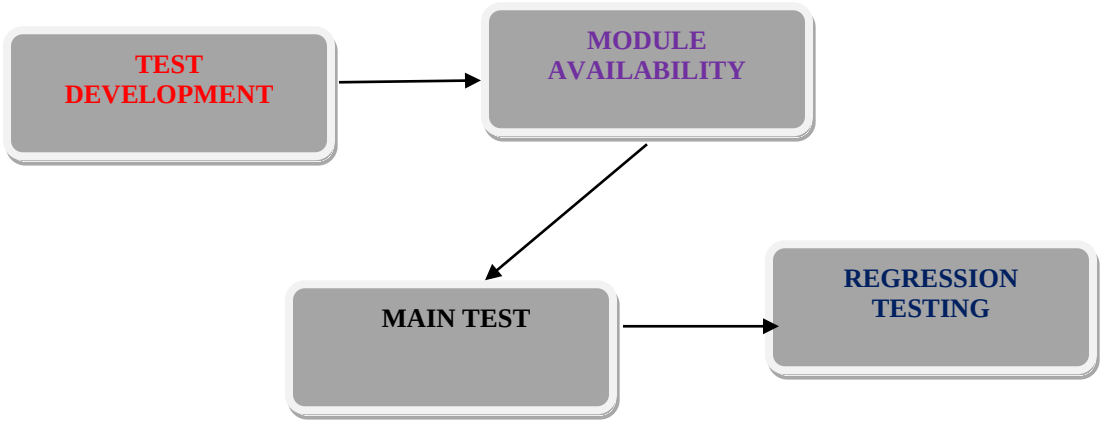
Requirement number	Requirement Name	Resources	Effort estimation(hr)
1 1.3	**Scope** System Architecture	a. Windows OS/later version b. GCC compiler for C and C++	
2 2.2 2.3 2.3.1 2.3.2 2.4 2.4.1 2.4.3	**Functional Requirement** Calling GolfScore **Program Functionality** Tournament assumptions Scoring **Data Input** Course Records Golfer's Record	Atlassian Jira	
2.5 2.5.1 2.5.2 2.5.3	**Data Output** Tournament ranking Reports Golfer's Reports Course Reports	Atlassian Confluence	
2.6 2.6.1 2.6.2 2.6.3	**Error Handling** Input parameter error Input data Errors on Output	Atlassian Jira	

Appendix B – Detailed Test Schedule

Below is the Gantt chart showing the timeframe to perform activities as par the Test schedule in 6.0 above



Below is the PERT diagram showing the dependencies:



Appendix C – Test Case:

S/N	Test Case	Type of Test
1	The program shall display its title and release version at execution time	Functional
2	The program shall be called from the command line	Functional
3	Number of golf course "1" shall be accepted	Functional
4	Number of golf course "2" shall be accepted	Functional
5	Number of golf course "3" shall be accepted	Functional
6	Number of golf course "4" shall be accepted	Functional
7	Number of golf course "5" shall be accepted	Functional
8	Number of golf course "6" shall return an error	Functional
9	Number of golf course "-1" shall return an error	Functional
10	Number of golf course "0" shall return an error	Functional
11	Number of golf course "100" shall return an error	Functional
12	The number of golfer's "1" entered into the tournament shall return an error	Functional
13	The number of golfer's "2" entered into the tournament shall be accepted	Functional
14	The number of golfer's "12" entered into the tournament shall be accepted	Functional
15	The number of golfer's "13" entered into the tournament shall return an error	Functional