MSSE SOFTWARE, INC.

Software Requirements Specification/ Design Document

GolfScore

Revision 1.1

Neil Bitzengolfer
(with all due respect to Edward Kit)

July 18, 2017

1. Scope

1.1 Objective

This document specifies the functional requirements and high-level design for 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.

This document is intended for the GolfScore development team (software developers and test engineers), Service representatives, and Marketing personnel.

1.2 Identification

GolfScore Release 1.1. The program will display its title and revision number on the screen at execution time.

1.3 System Architecture

GolfScore will be written in either C or C++ and is intended to run on a PC running Windows 2000 or any later version.

2. Functional Requirements

2.1 Program Description

GolfScore is a program used to generate reports of golfers' results for a golf tournament. The input to the program will consist of a file containing two types of records as described in Section 2.4 below. The output from the program will consist of up to 3 reports as described in Section 2.5 below. The program is executed via a command line interface – there is no GUI associated with the application.

The program will be run as a stand-alone executable, and can be run from a command line prompt, from within an IDE (Integrated Development Environment), etc. Input to the program will come from an input record file, and output from the program will go to output record files in a format suitable for printing.

Any errors in the inputs or outputs will be handled as specified in Section 2.6 below.

2.2 Calling GolfScore

Format for calling GolfScore: >**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 [See Section 2.5 below for report descriptions.]
filename		The name of the input file containing the data for the reports. [See Section 2.4 below]. 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.

Examples:	>golf	-h			Display help information
	>golf	-ct	c:\in.txt	golfout	Generate the Course Report and the Tournament Ranking report. The input is in file in.txt in the c:\ root directory, and the output is to be placed in directory c:\golfout.

2.3 Program Functionality

2.3.1 Tournament assumptions

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 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.
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.
Note that the lower a golfer's stroke count (relative to par), the higher his or her score for that hole.

2.3.2 Scoring

The score earned by a golfer for each hole played is as follows:

Stroke count	Score
over par	0
par	1
1 under par	2
2 under par	4
3 or more under par	6

2.4 Data Input

Input to GolfScore will consist of a formatted text file containing the following records in the order given.
Individual records in the file are terminated by the end of a line.
The name of the input file is supplied as a parameter on the program call line. [See Section 2.2.]

2.4.1 Course Records

The purpose of the Course Records is to describe the various courses being used. There will be one Course Record for each golf course used in the tournament. Each record contains the name of the course, an identifier for the Golfer Records, and par for each of its 18 holes. All Course Records come first in the input file.

Column 1	Blank
Columns 2-19	Course name
Column 20	Single-character course identifier
Columns 21-38	Par for holes 1-18, in order, single integer: 3, 4, or 5

2.4.2 Delimiter Record

The end of the Course Records will be specified by a delimiter record with the following format.

Column 1	Non-blank

2.4.3 Golfer Records

There will be one Golfer Record for each golfer for each course played; these records can appear in any order.
Each record contains the name of the golfer, the Course Identifier, and the golfer's stroke count on each of the 18 holes.

Column 1	Blank
Column 2	Course Identifier
Columns 3-9	Ignored
Columns 10-29	Golfer name
Column 30	Ignored
Columns 31-48	Stroke count for each of the 18 holes

2.4.4 Delimiter Record

The end of the Golfer Records, and hence of the input file itself, will also be specified by a delimiter record.

Column 1	Non-blank

2.5 Data Output

GolfScore will generate up to 3 reports, based on input options. The generated reports will be stored as text files in the directory determined from the program call line.

2.5.1 Tournament Ranking Report

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 (1st place, 2nd place, etc.). The list will be in descending order of final score (i.e. best golfer first). In the case of ties, the golfers will be listed alphabetically.

The output filename for this report will be trank.rep

2.5.2 Golfer Report

This report is exactly the same as the Tournament Ranking Report except that the golfers will be listed alphabetically by last name.

The output filename for this report will be golfer.rep

2.5.3 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 for this report will be course.rep

2.6 Error Handling

2.6.1 Input Parameter Errors

For any input parameter errors, including unrecognizable options, the program will stop, and display a message explaining the error.
The first field following *options* will be assumed to be a filename. If the file specified by *filename* does not exist, an input parameter error will be reported.
The first field following *filename* will be assumed to be a directory. If the directory specified by *output-directory* does not exist, an input parameter error will be reported. If *output-directory* is not supplied, the directory that contains *filename* will be used.
Any field beyond the *output-directory* field will be ignored and will not be considered an error.

2.6.2 Input Data Errors

Input Data errors checked for are as follows:

i) Non-numeric data where numeric data is expected: the program will stop with an appropriate error message
ii) Par values that are not 3, 4, or 5: the program will stop with an appropriate error message.
iii) 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.

2.6.3 Errors on Output

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)".

The user will then respond “Y” or “N”. If “Y”, the output file will be overwritten; if “N”, the generated output will be discarded. There will be a separate user prompt for each output report type requested.
If the specified output report does not exist in the path specified, it will be created.
If any other errors occur while attempting to write the output files, program behavior is unspecified.

3. Deliverables

- GolfScore executable file

4. Performance Requirements

Once executed, GolfScore will complete its processing within one minute.