

**Luke Schaefer LESDMG 18186970**

## **Final Project Description**

### **Contents:**

- Introduction
- Golf Background
- Features
- Dependencies
- Problems Encountered
- Test Results

### **Introduction**

Welcome to my golf management app, where you can track all the golf stats an amateur golfer would ever need. In this WPF application you can track stats such as green in regulation percentage, average putts per hole, percentages of each score per hole, and more. I hope you enjoy my golf tracker!

### **Golf Background**

To understand my application, some basic golf terminology is necessary. The goal of golf is to shoot a small ball into a hole between 170 to 500 yards away. The player who swings a club (the action of taking a swing is usually called a stroke) the least amount of times on each hole on a course wins. A score is the number of strokes on a hole or it can refer to as the combined score of all of the holes on a course. Typically a course has 18 holes, with a total par score of 72 (each course usually has 14 par 4s, 2 par 5s, and 2 par 3s). A full round is considered 18 holes, but usually amateurs track 9 hole stats as well. Some basic terminology is listed below:

- Par - the “expected” score of a hole. This is what golfers measure their score against. Obviously the “expected” is not obtainable by most, and most people shoot worse than par. It is intentionally difficult to meet par.
- Putt - a stroke that is taken on the green with a putter (it is expected to have 2 or less putts per hole, any more is not desirable). These are the strokes that usually put the ball in the hole.
- GIR (green in regulation): This term is used to describe when a player reaches the green (the area around the hole) 2 stroke below par. It is a descriptive stat that a player can use to determine where they need to improve.
- Birdie, Bogey - these two terms are used to describe the score of one stroke better than par (birdie) and one stroke worse than par (bogey). There are other names for strokes 2 worse and 2 better, but we do not use those in this program.
- Handicap - this term refers to the average score of a golfer over/under the “expected” score of 72 on a course. A lot of factors play into the handicap as well, such as course difficulty and number of rounds played. A handicap is calculated as the average of a players top 8 scores out of their past 20 rounds.

## **Features**

The golf management app consists of numerous features.

- Stat calculation:
  - The main functionality of the program is to view stats of the player. The program uses the stored rounds to calculate important stats such as GIR, Average Score, score breakdowns, and more.
  - The best score can only be calculated from rounds of 18. A 9 hole round (half of a round) cannot be used in a best score calculation, but it can be used to calculate stats for the player.
  - The average score can be calculated using a 9 hole score, but that is because an average score calculation usually splits a round of 18 into two 9 hole rounds for calculation.
- Loading/saving rounds:
  - A user can load/save rounds into/out of the app to persist data. The data is stored in a json file so that the user can access them later.
- Add rounds:
  - A user can manually add a round following the usual scorecard convention. A user can select a date, enter a course name, then enter the stats for each hole (par of the hole, player score, and number of putts)
- Deleting rounds:
  - A user can delete a round out of the app that they do not want to store anymore.
- Simulate rounds:
  - A user can simulate a round using their handicap. The program randomly calculates a score and number of putts for each hole based on the user's handicap.

(I implemented all features in my proposal except for the course information feature. I instead implemented the round simulation instead)

Programming Techniques Used:

- Loops
- Methods
- Classes (Golf hole, golf round, golf list and UI classes)
- Inheritance (A golf round list inherited from list and added functionality)
- Strings, Arrays, and Lists
- MVC (the user has view components that interact with the controller (golf list) that interact with the data model (round and hole data))
- Multithreading (timers were used to simulate a round and pause the program from executing but kept updating the UI thread)
- Exception Handling (used to validate user input)

## **Dependencies**

There are no extra dependencies needed in the project other than the use of the windows OS, visual studio, and the WPF framework.

## Problems Encountered

I encountered numerous problems creating this app. I will mention my top 3 most frustrating problems and how I responded to them.

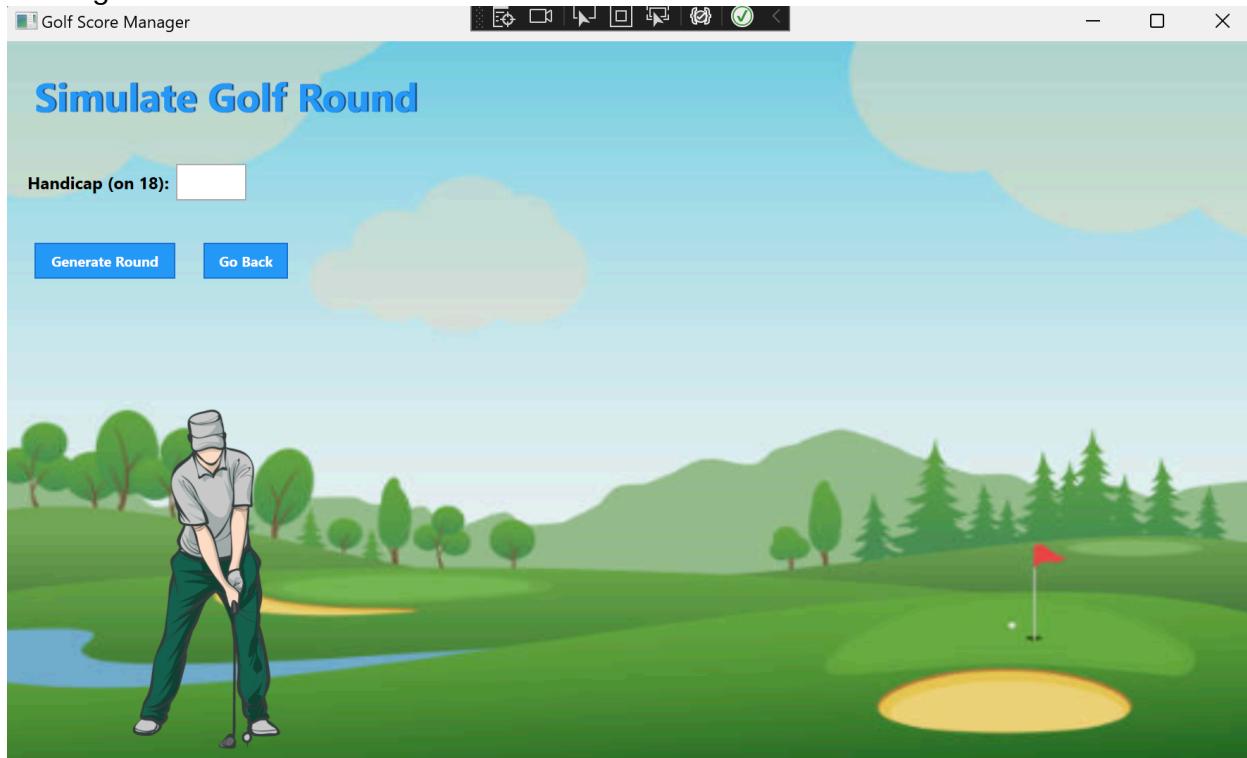
- Simulation Issues
  - I used a Threaded timer to act as if it takes time to “simulate” a round. I came across numerous solutions, each with their own slew of issues. I determined it would be best to freeze the main thread, and use another thread to update the UI then unfreeze when I determined the simulation was complete.
  - How do you properly simulate a round? What kind of math should I use to randomly generate a round that would be accurate to a handicap? Coming up with a solution for this was difficult, but I ended up putting bounds on the high and lows of scores and putts, then randomly generating in those bounds, selecting the more middle ground of the range of scores more often.
- User Interaction
  - Golfers can be really bad sometimes, so determining what kind of input was appropriate was difficult. I ended up using a user check, so after input I would validate with them if that is what they intended.
  - Lots of try/catch statements were necessary to filter out invalid input.
- Serialization
  - JSON serialization can always be tricky sometimes, and I spent a decent amount of time meddling with the innuendos of JSON class serialization, especially with a class list inside of a class.

## Test Results

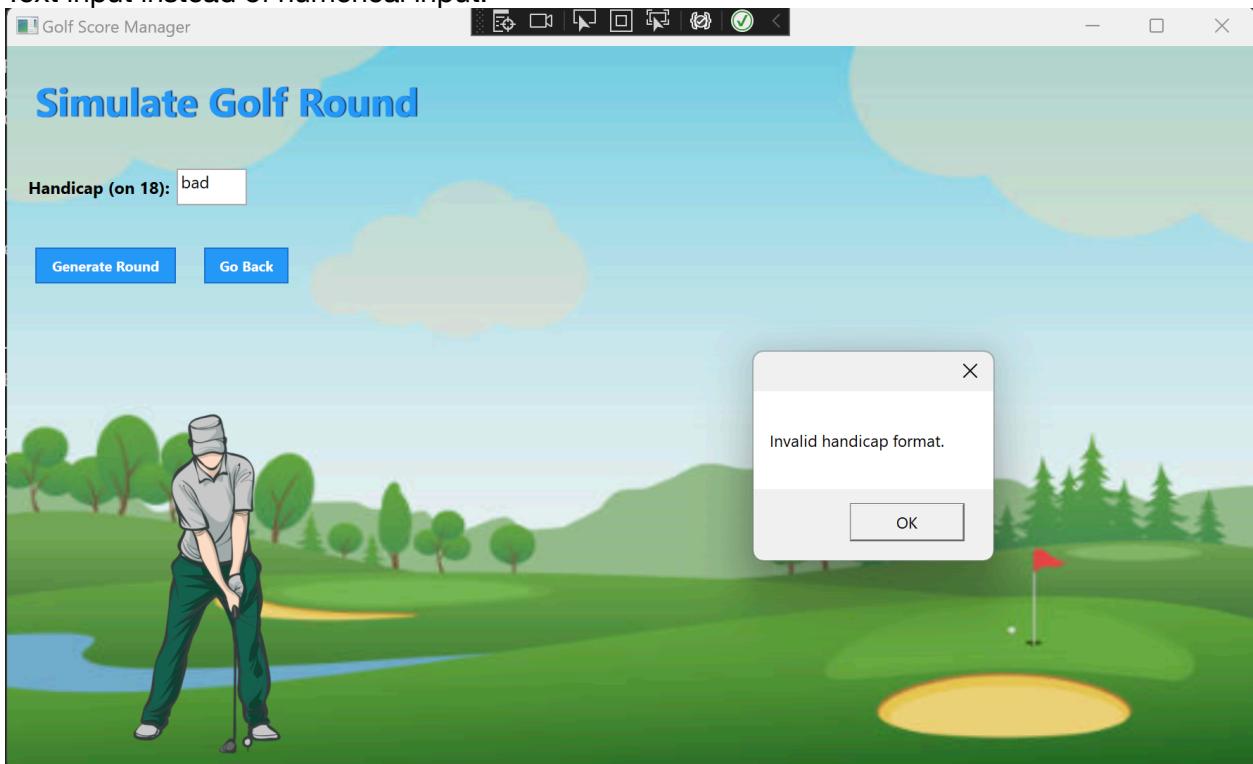
I documented numerous test cases. I have provided screenshots of valid input results and invalid input results below. The program never crashes and runs until the user exits!

### Simulation Testing

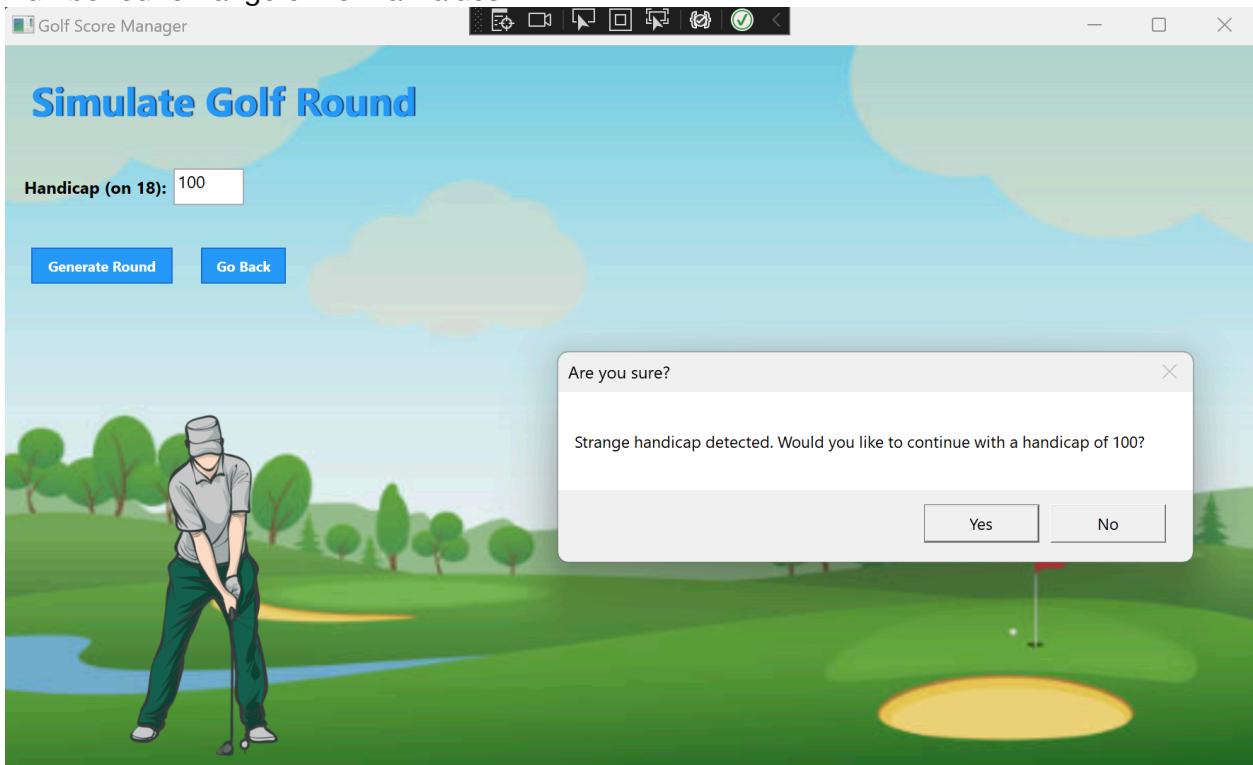
- Starting Screen



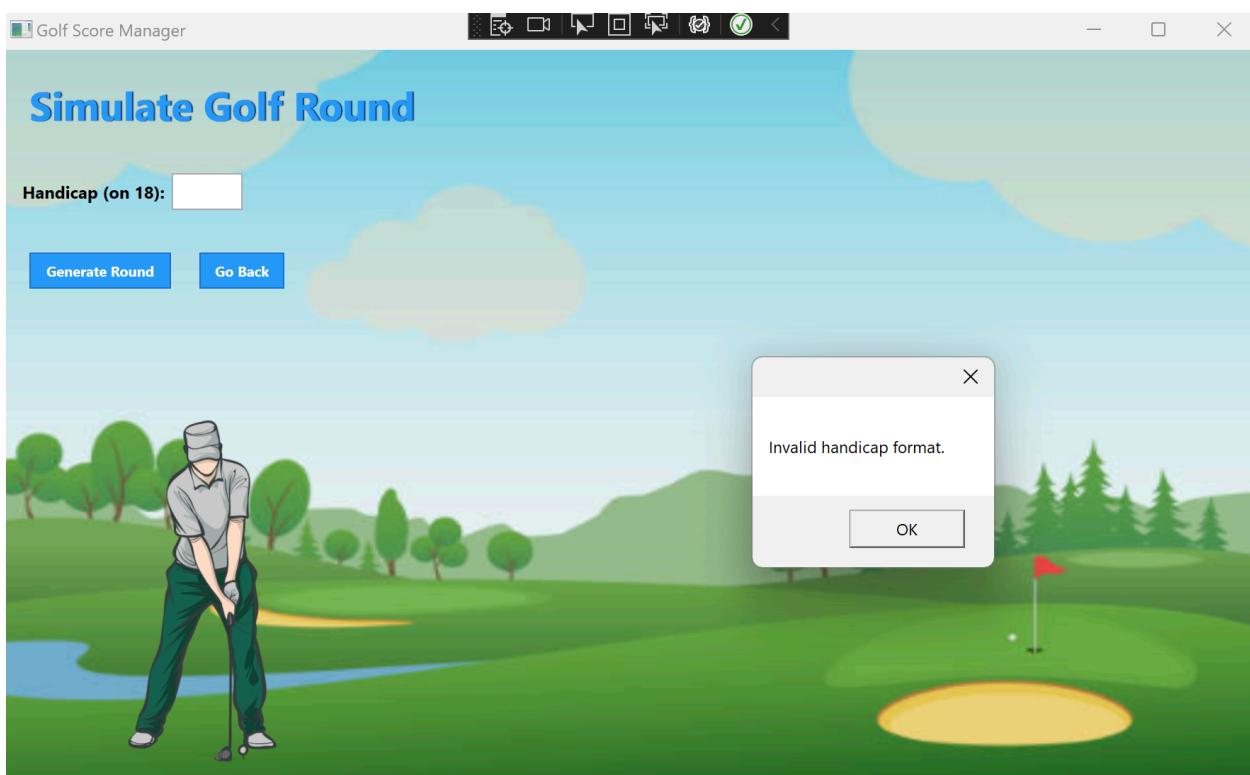
- Text input instead of numerical input:



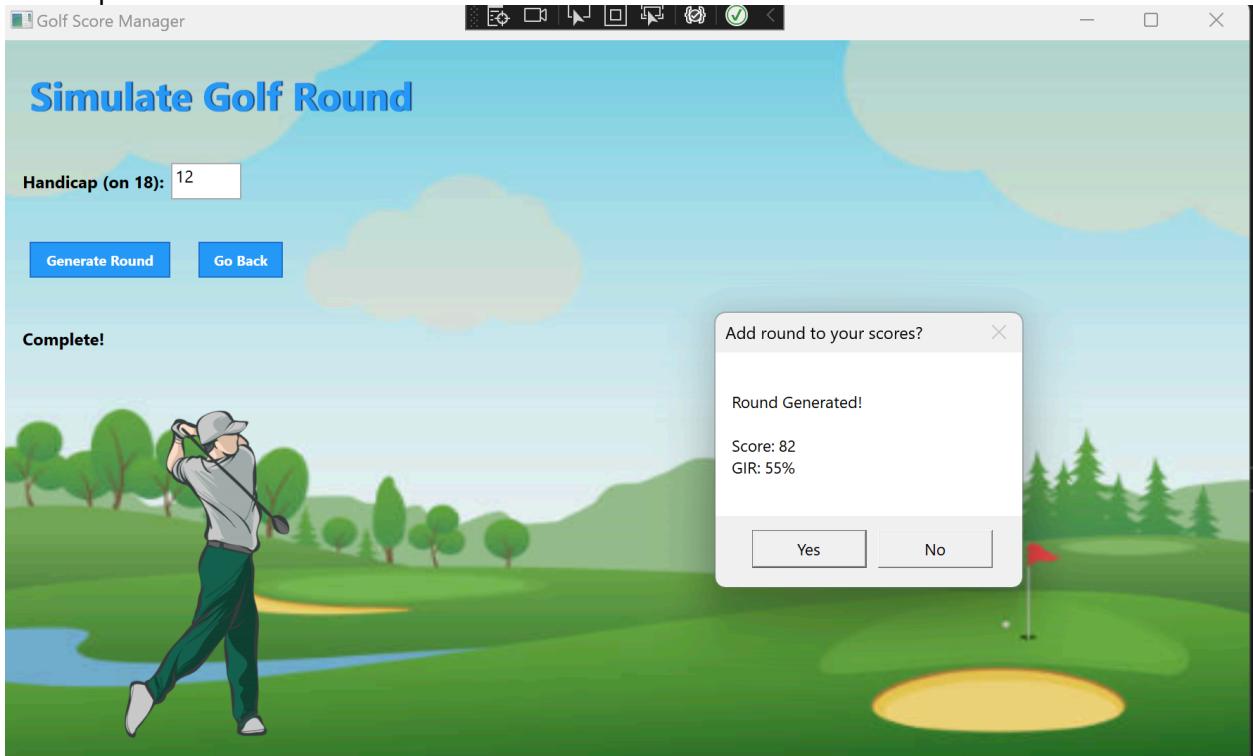
- Number out of range of normal values:



- No input:

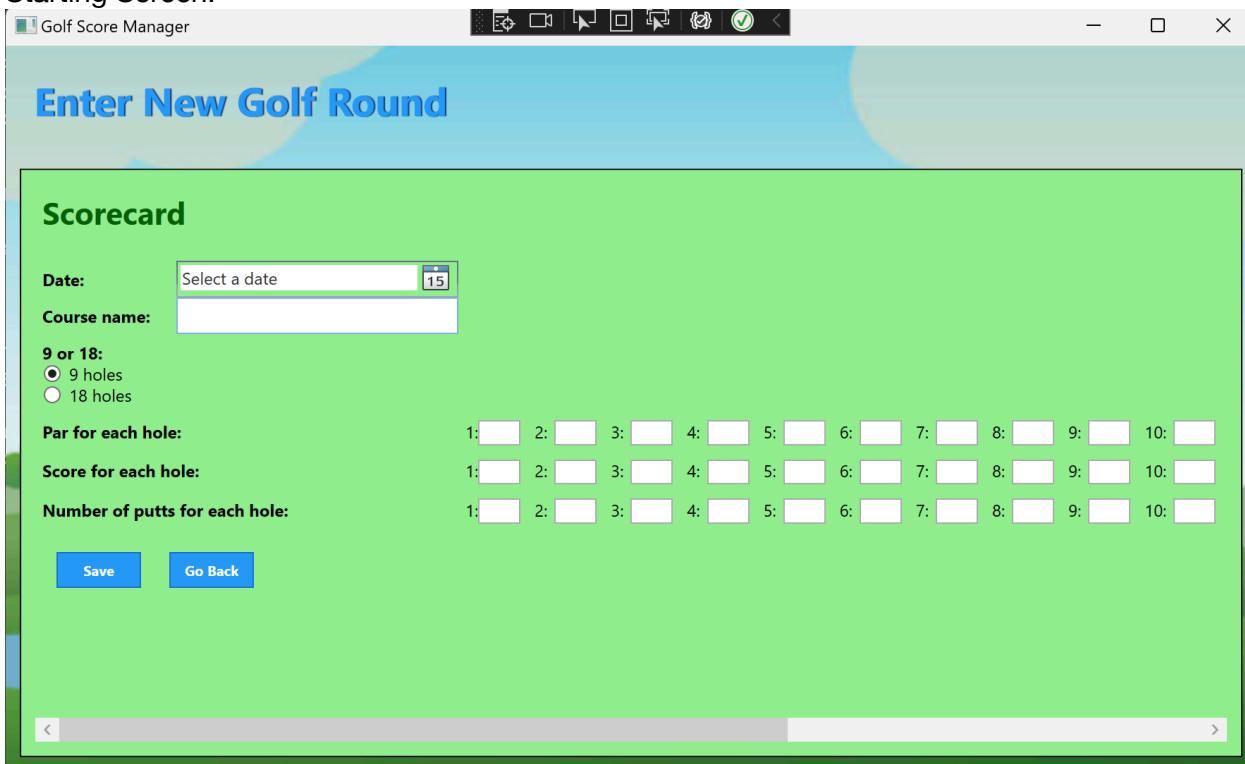


- Valid input:

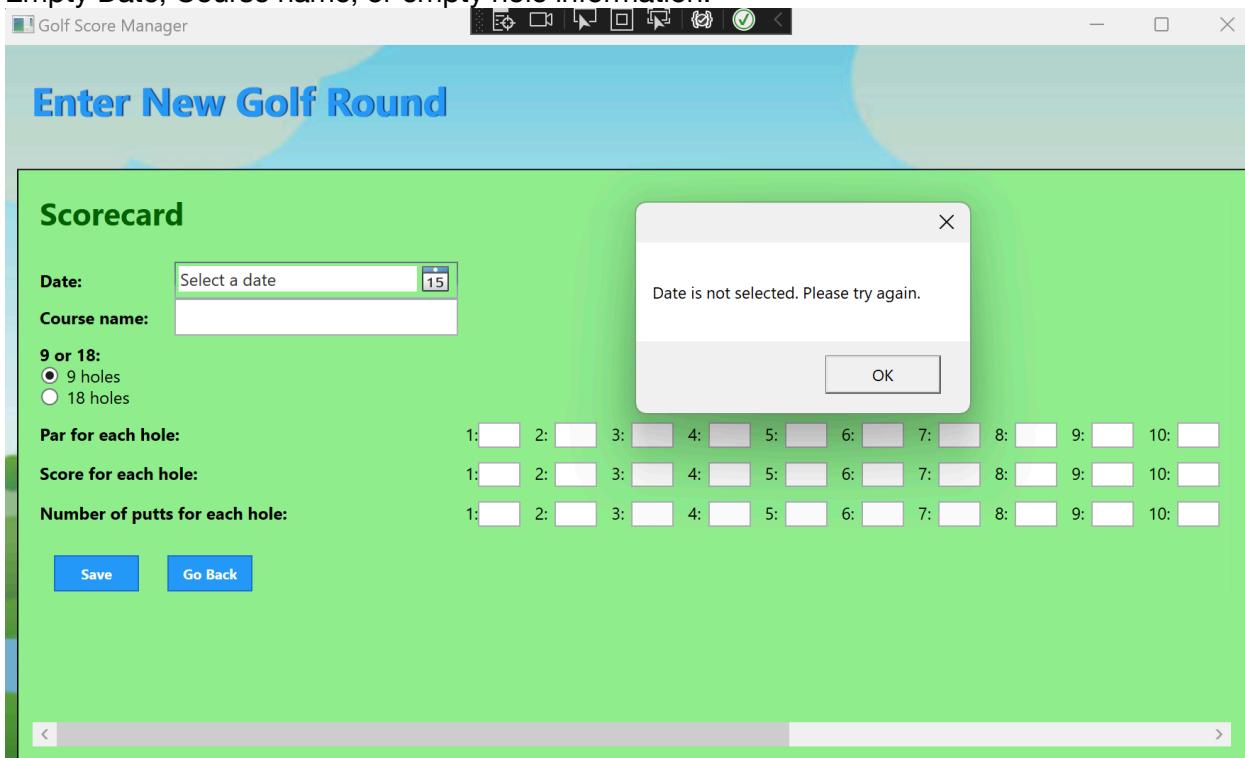


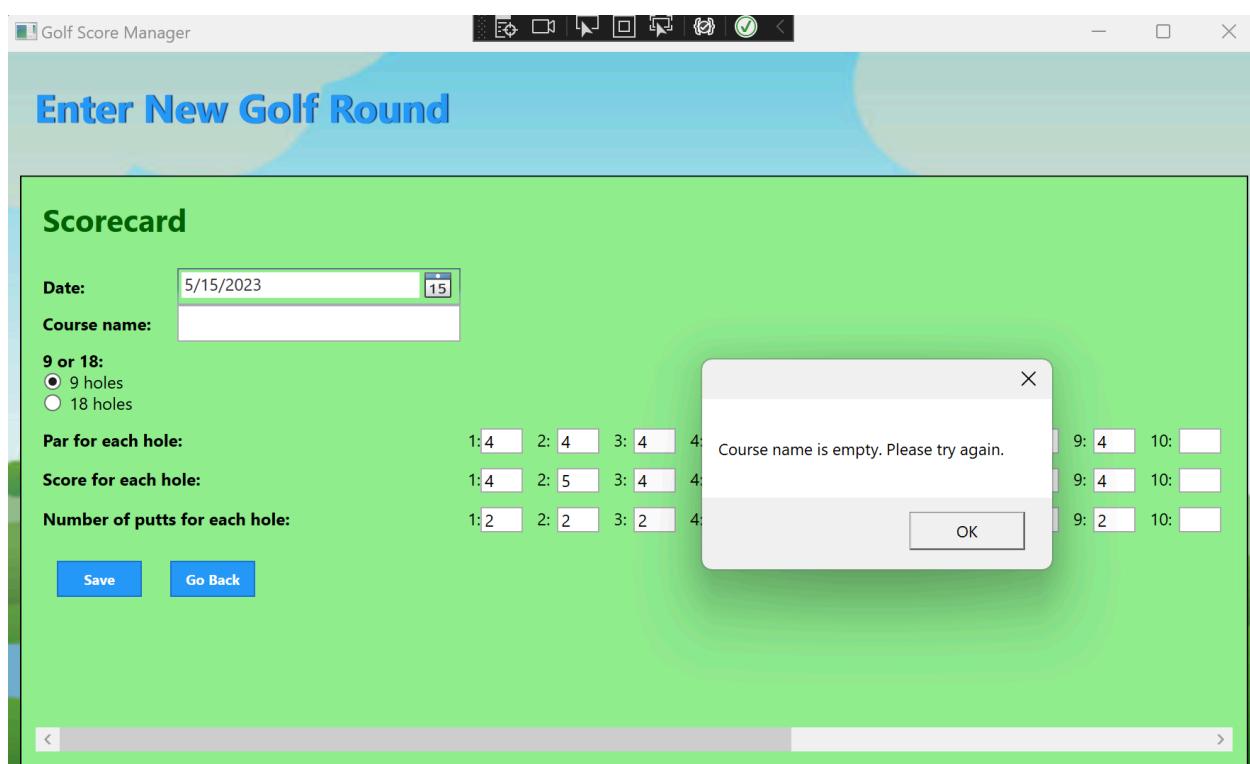
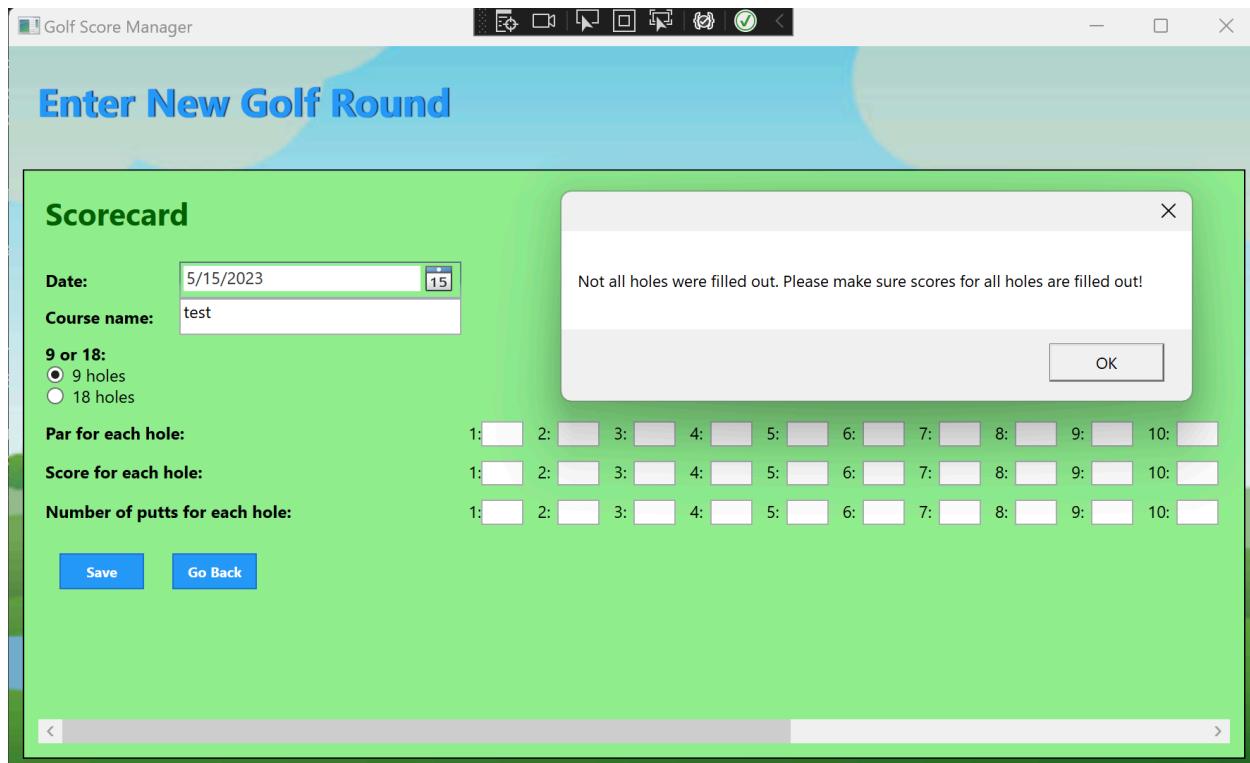
## Add Round Handling

- Starting Screen:

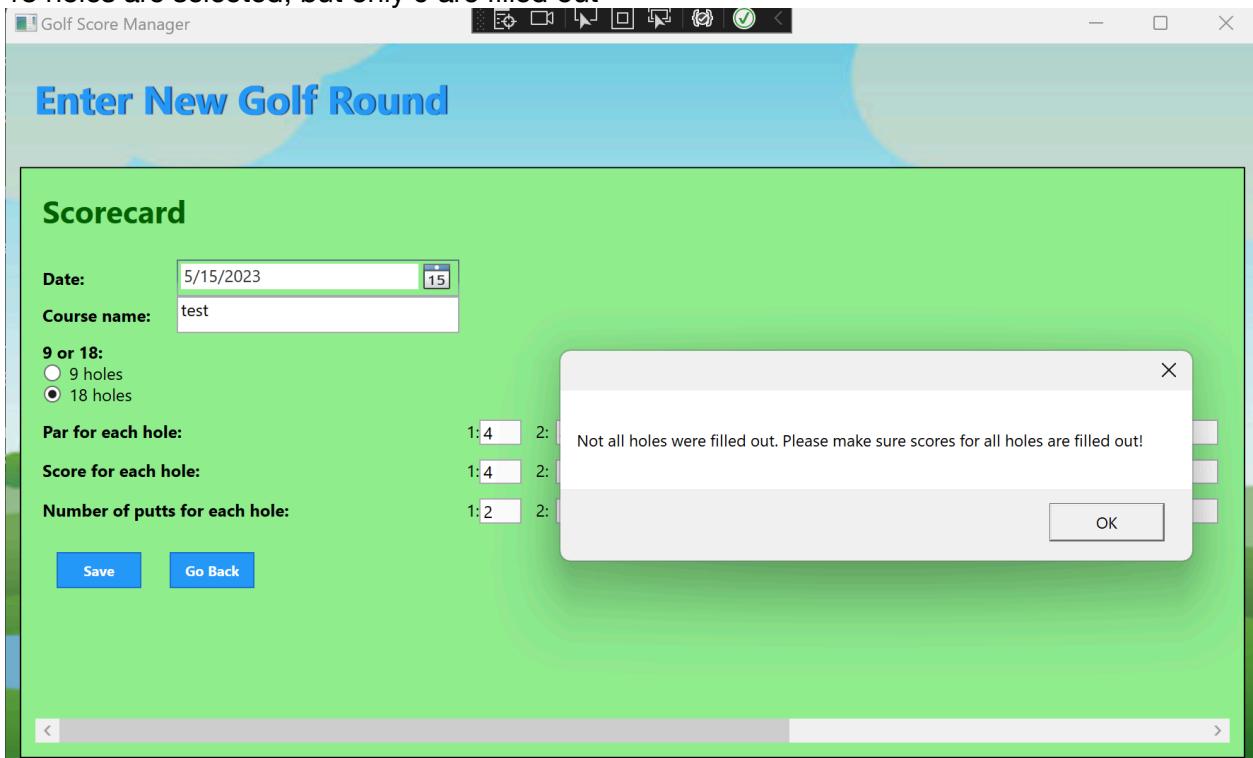


- Empty Date, Course name, or empty hole information:

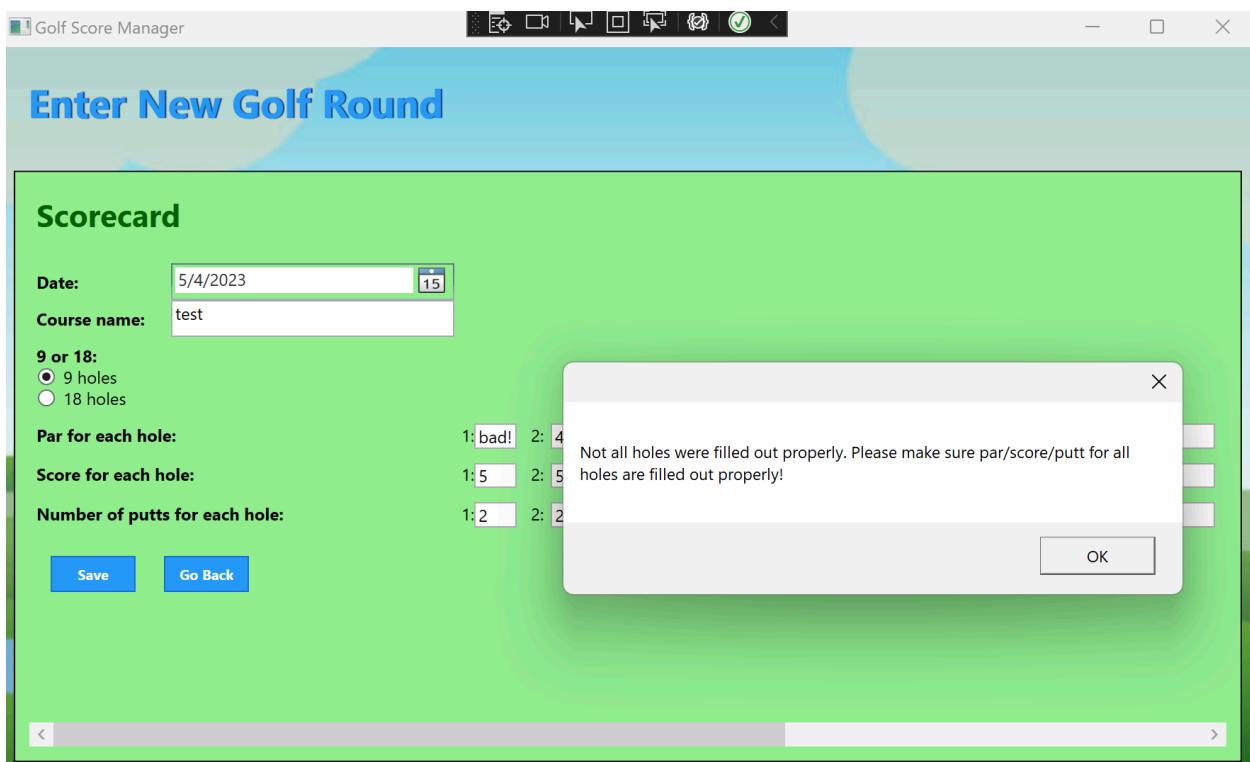


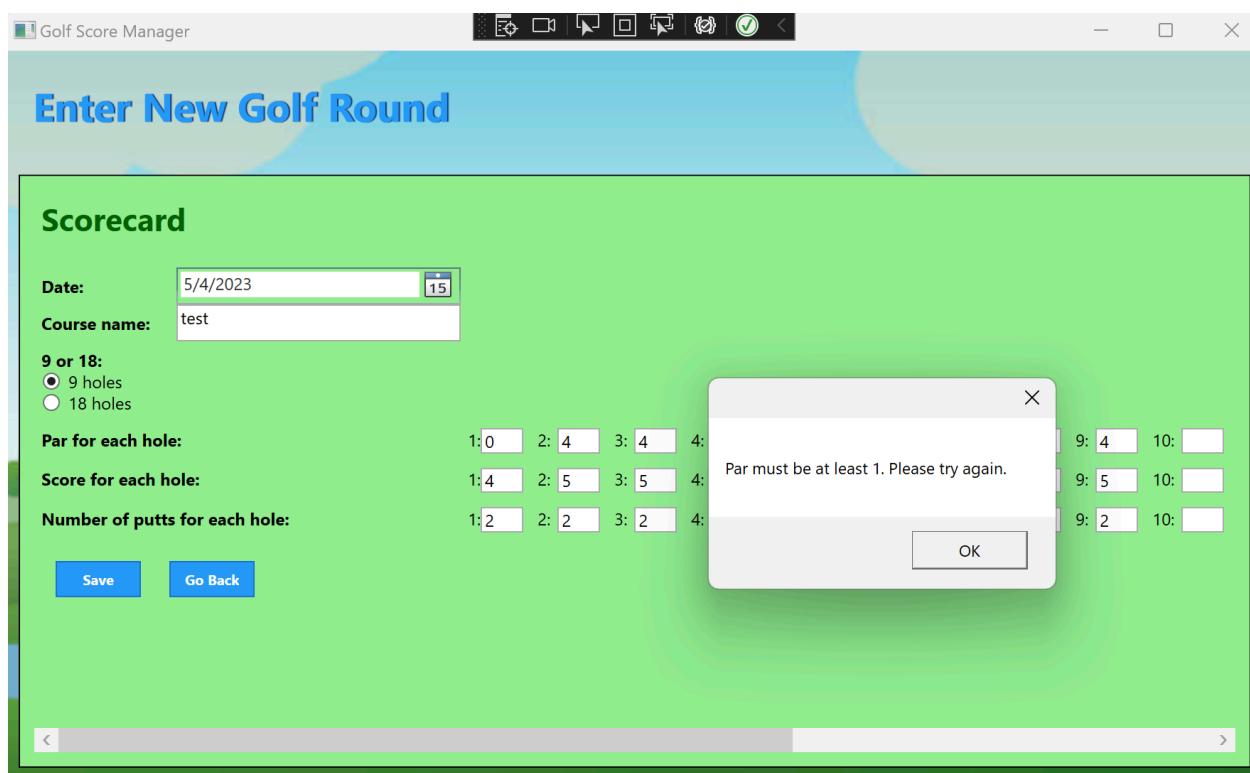
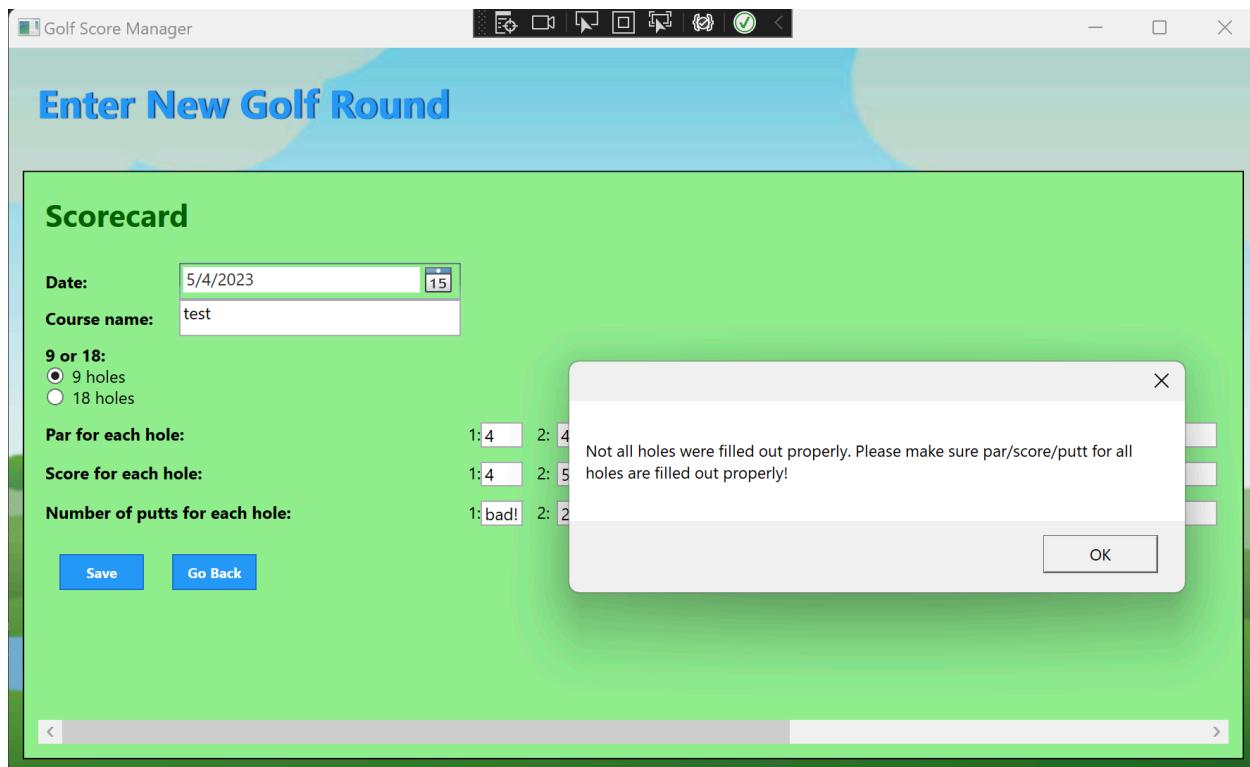


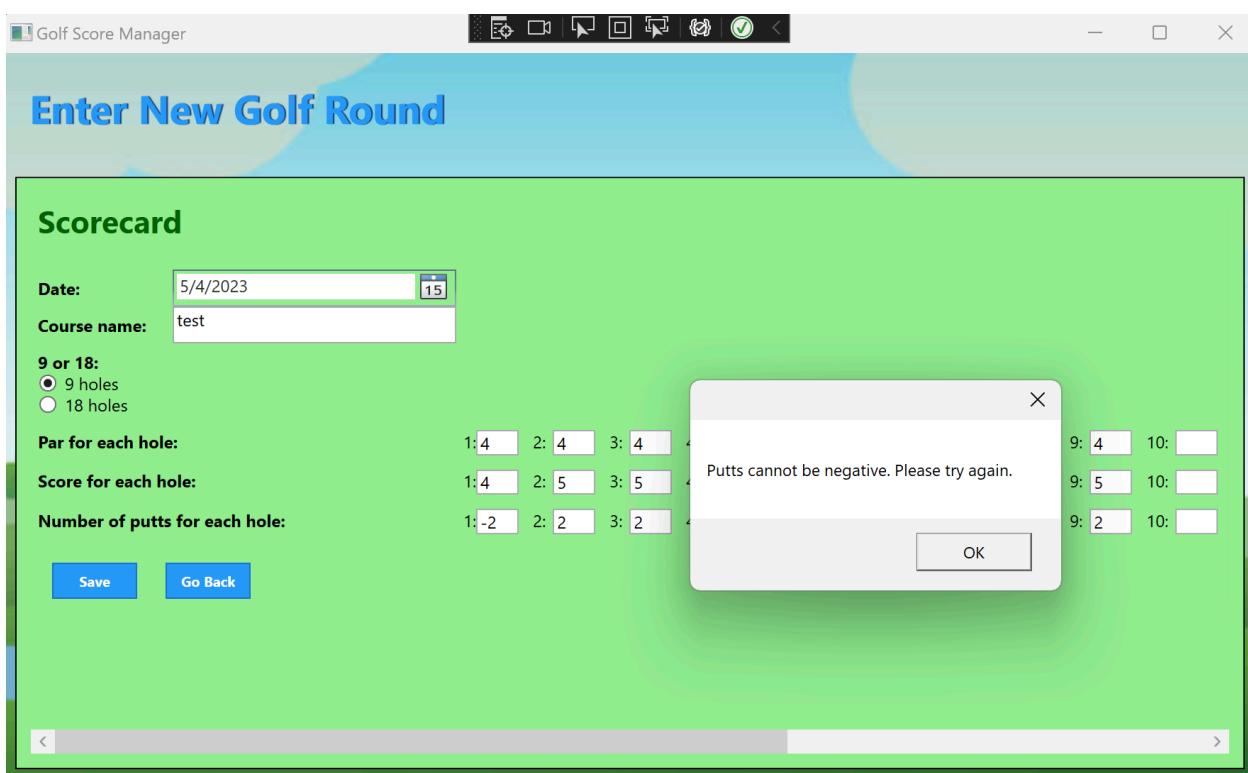
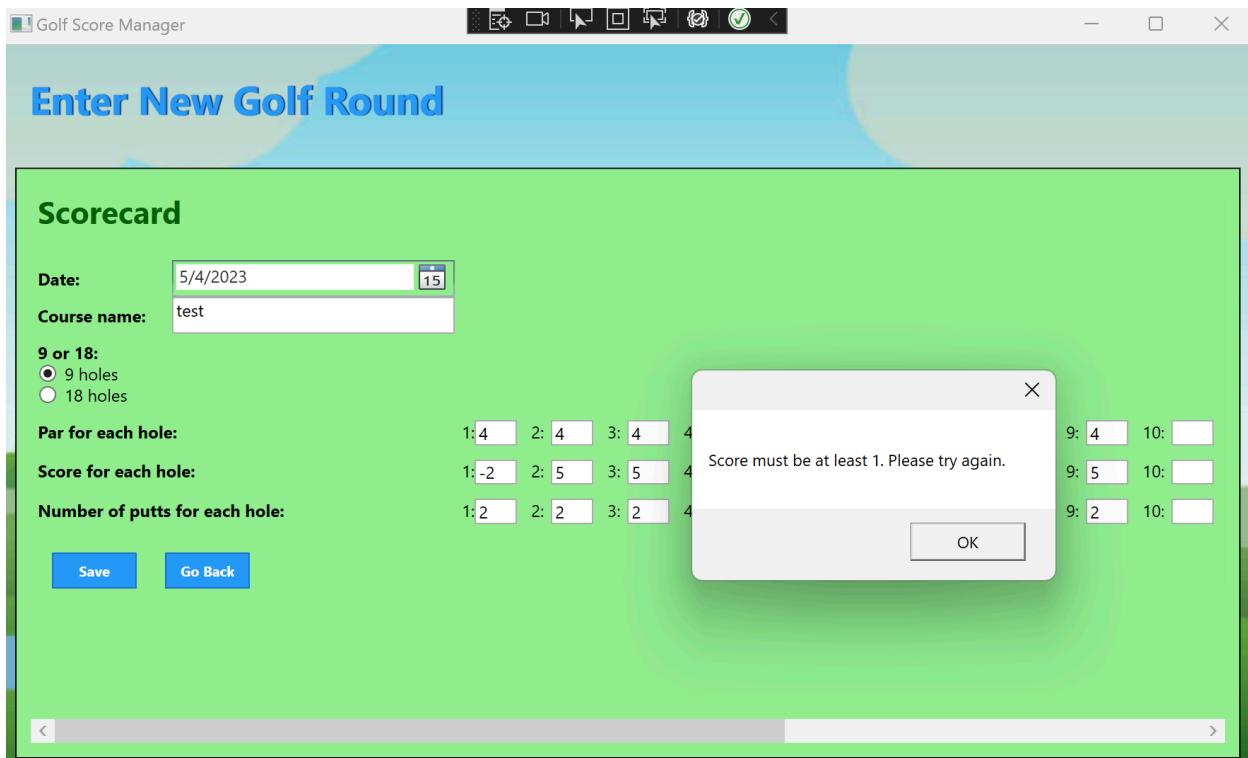
- 18 holes are selected, but only 9 are filled out

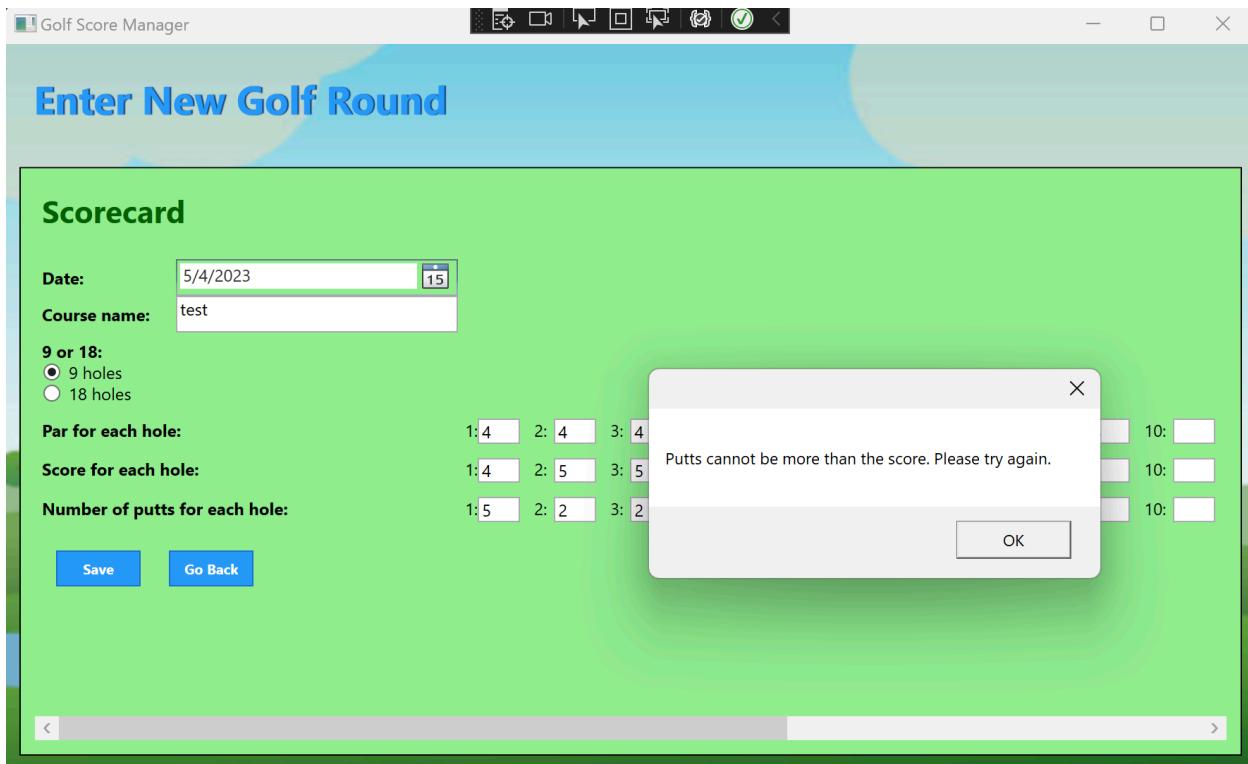


- Invalid score, par, or putt

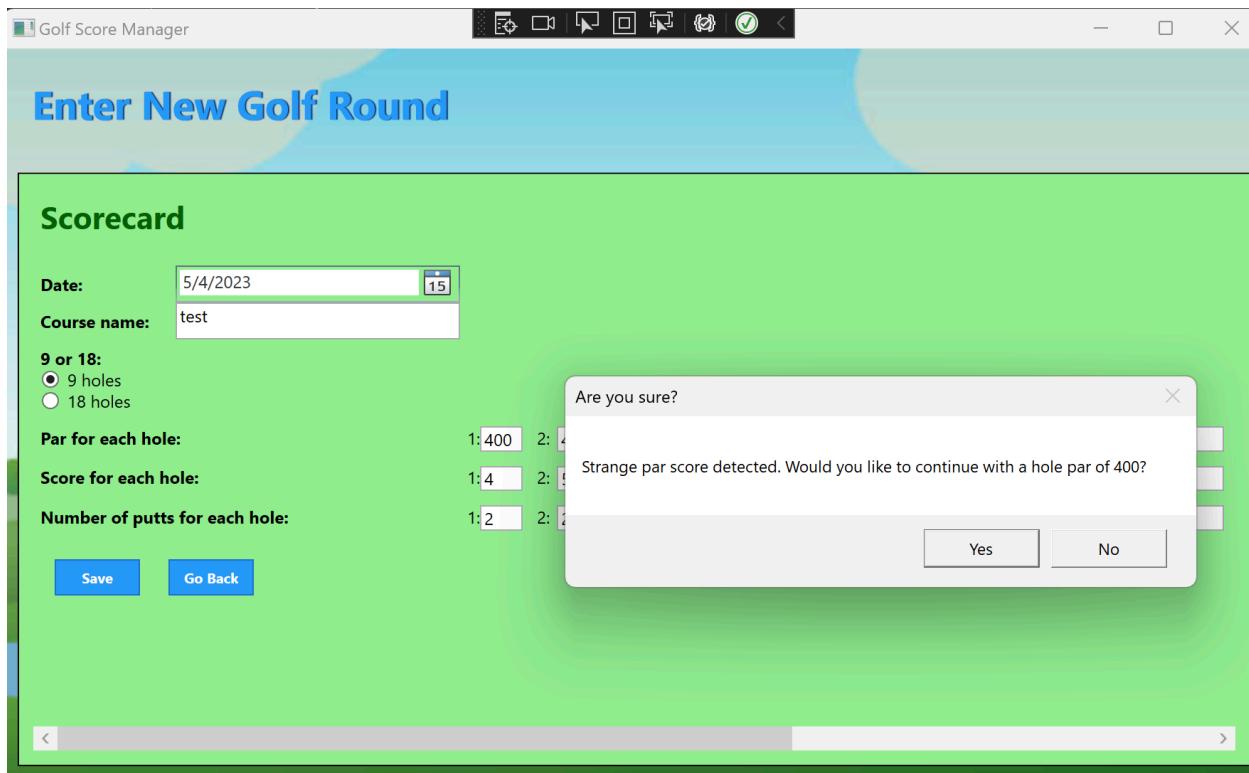


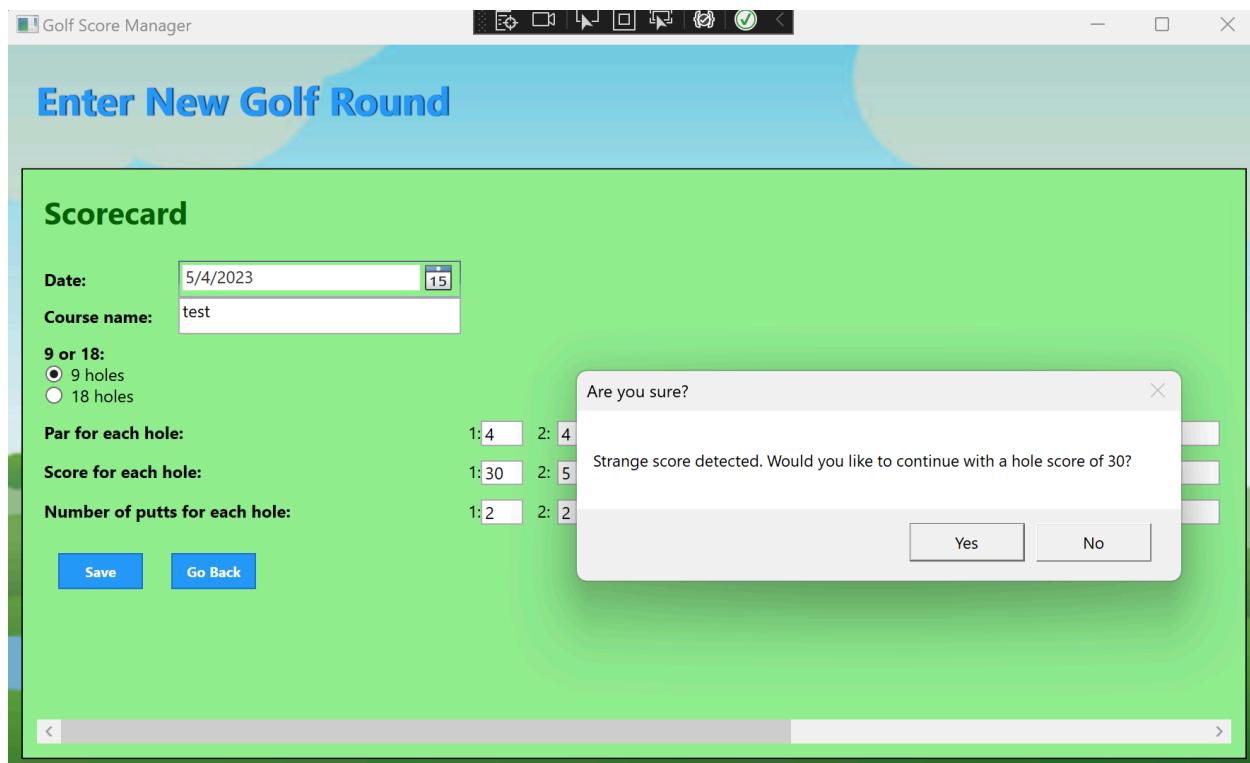




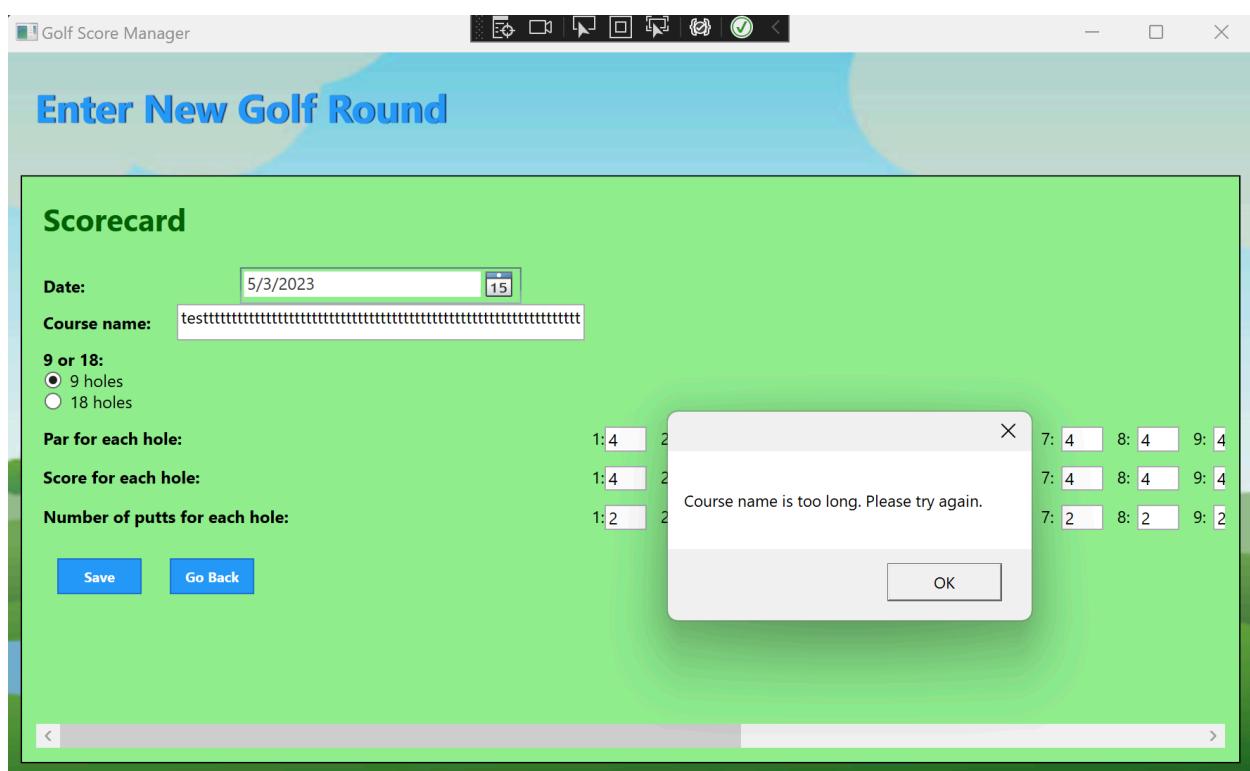


Strange score/par entered:

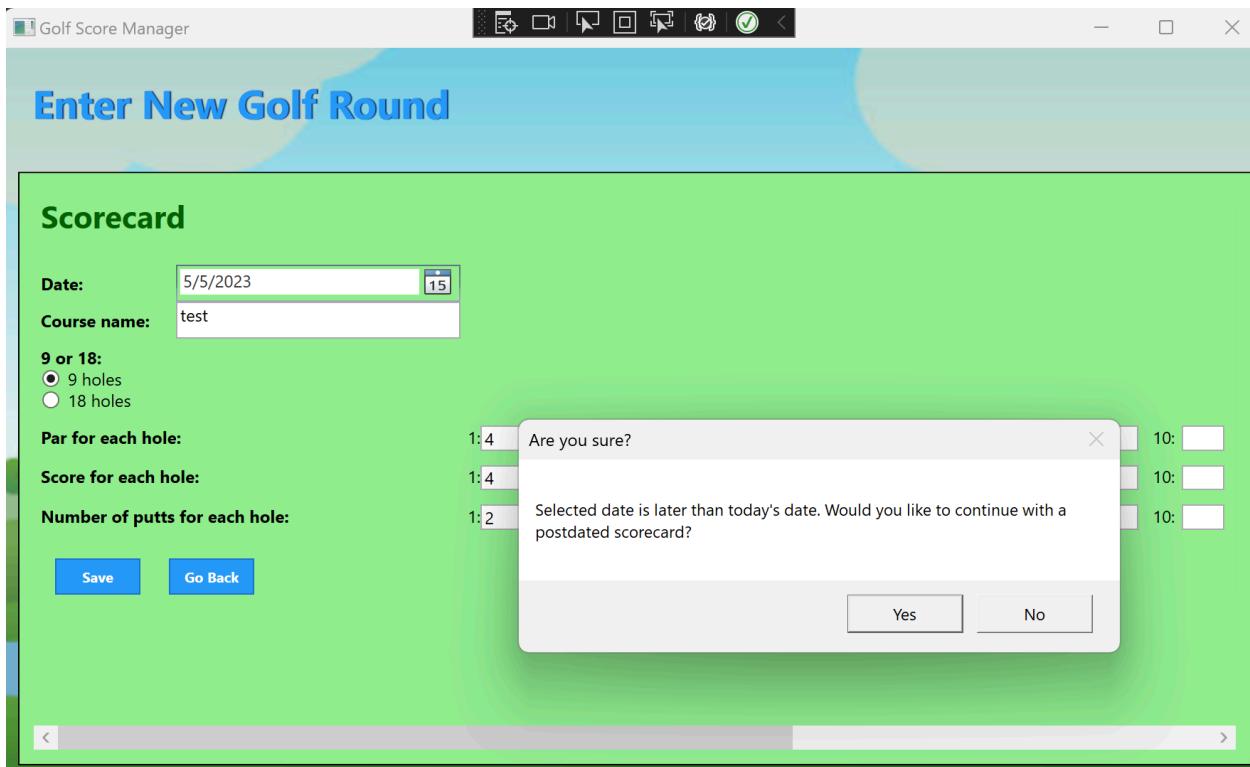




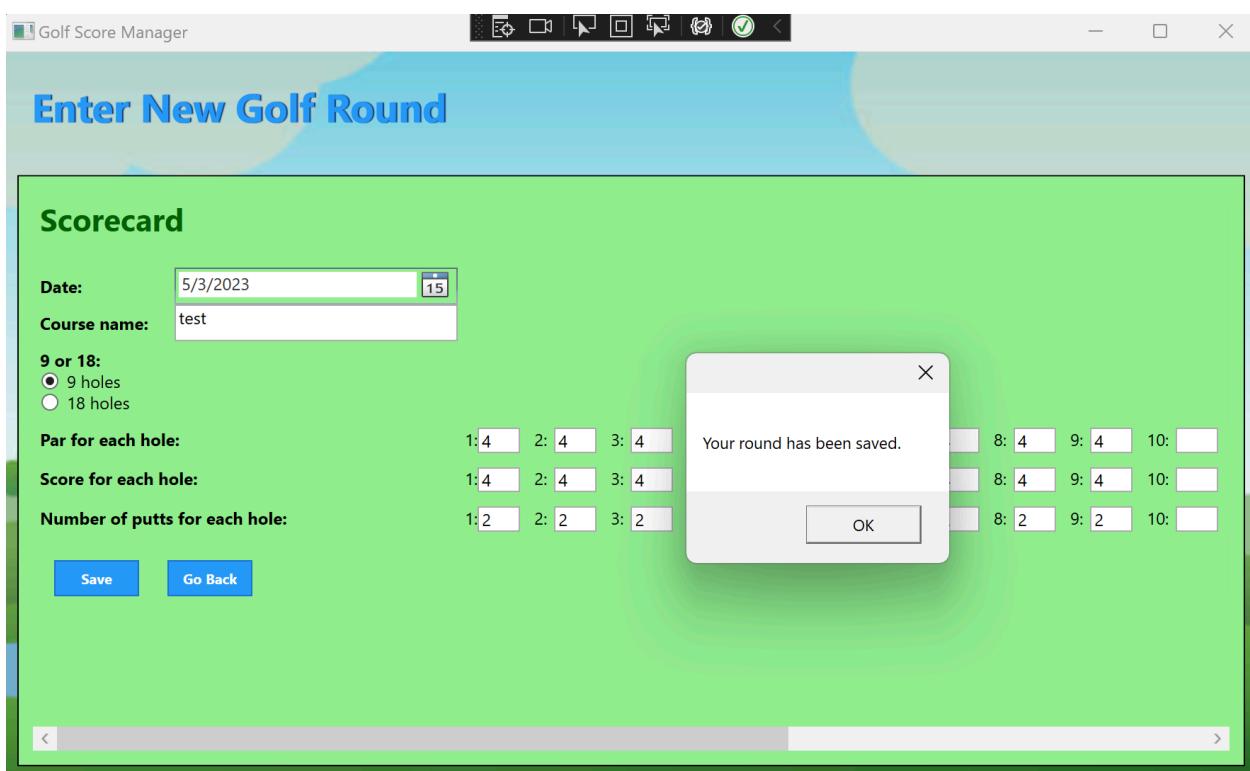
Too long of course name:



Future date:



Valid input:



Home Page Testing  
Page without data:

Golf Score Manager

# Golf Score Manager

Course/Date/Score:  
There are no rounds to view.

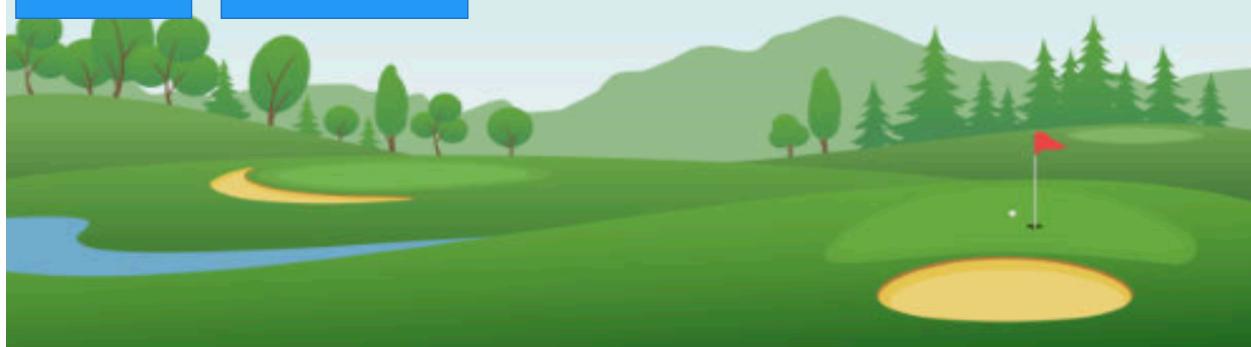
Best Score: n/a   Average Score: n/a

Birdie %: n/a   Par %: n/a   Bogey %: n/a

GIR %: n/a   Putts Per Hole: n/a

Better than Birdie %: n/a   Worse than Bogey %: n/a

Add Round   Simulate Round



Page with data:

Golf Score Manager

# Golf Score Manager

Course/Date/Score:  
SimRound 5/4/2023 82  
SimRound 5/4/2023 85  
SimRound 5/4/2023 89

Best Score: 82   Average Score: 85.3

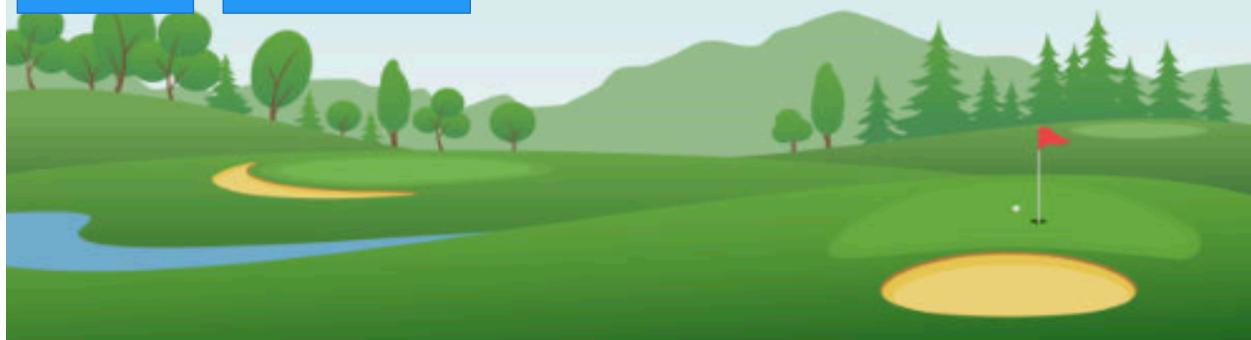
Birdie %: 11%   Par %: 22%   Bogey %: 57%

GIR %: 33%   Putts Per Hole: 2.3

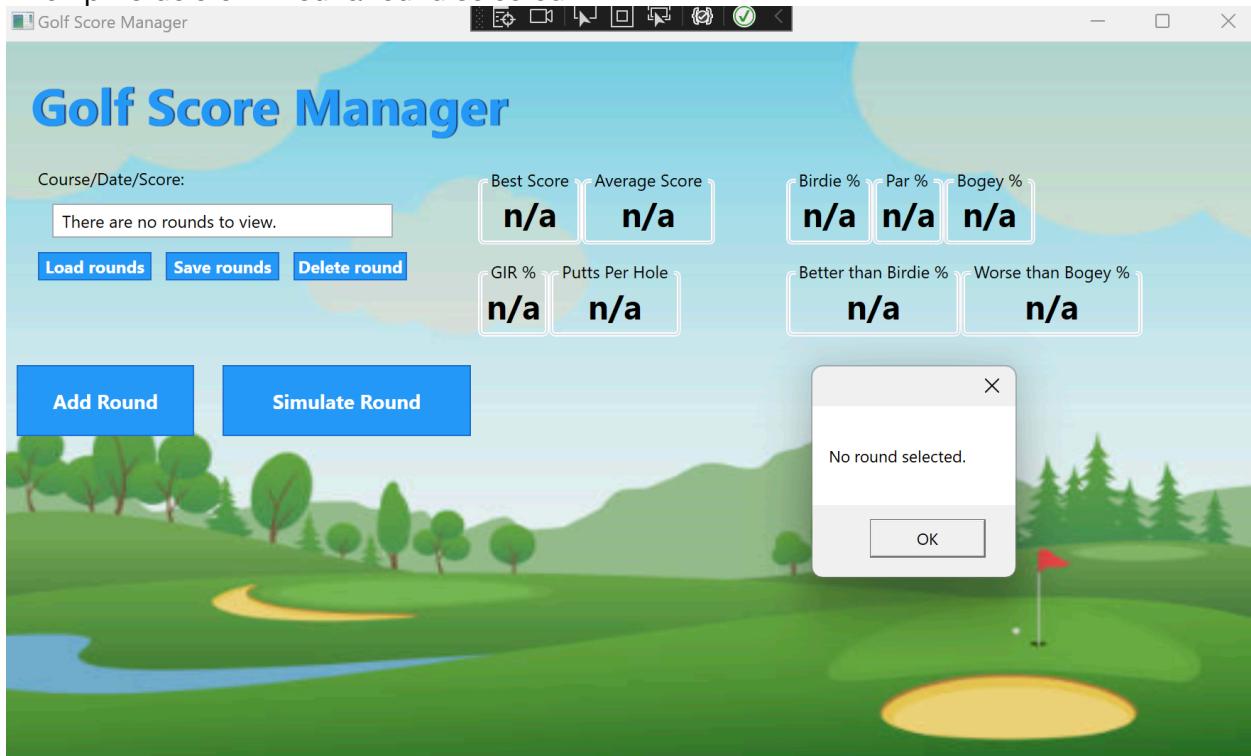
Better than Birdie %: 0%   Worse than Bogey %: 9%

Load rounds   Save rounds   Delete round

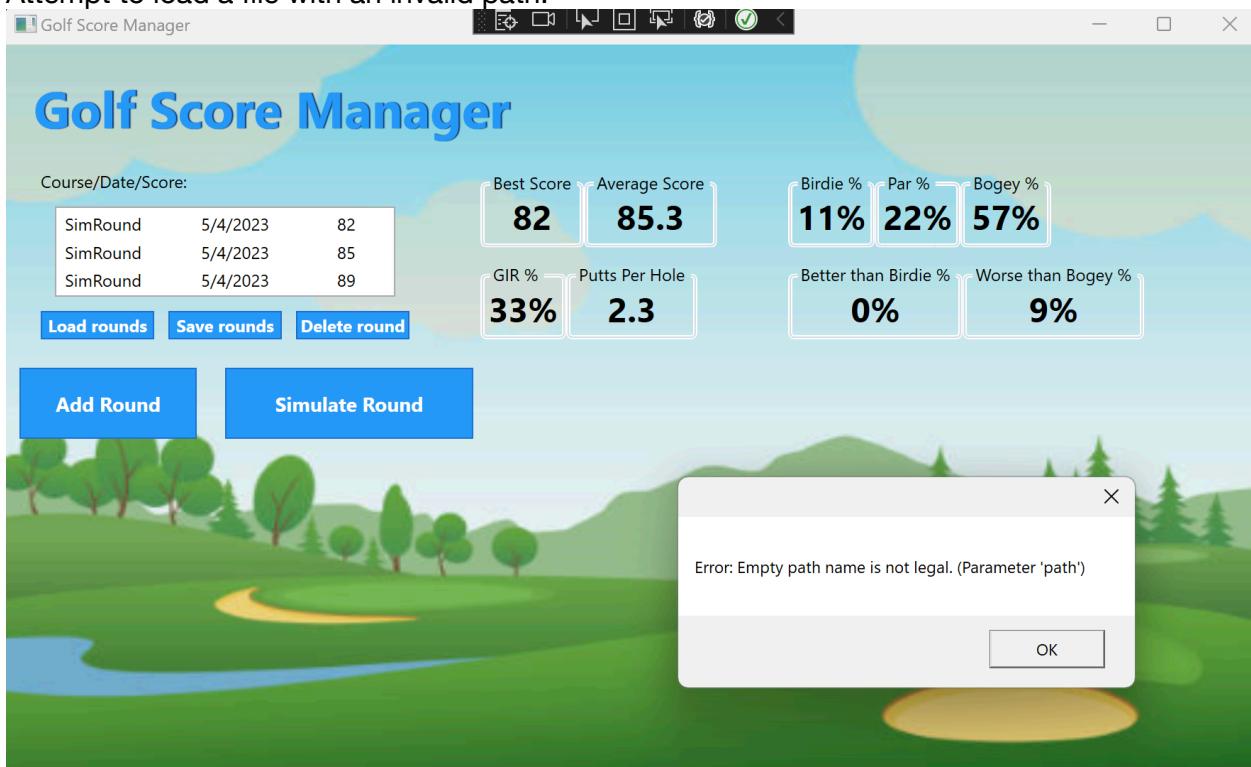
Add Round   Simulate Round



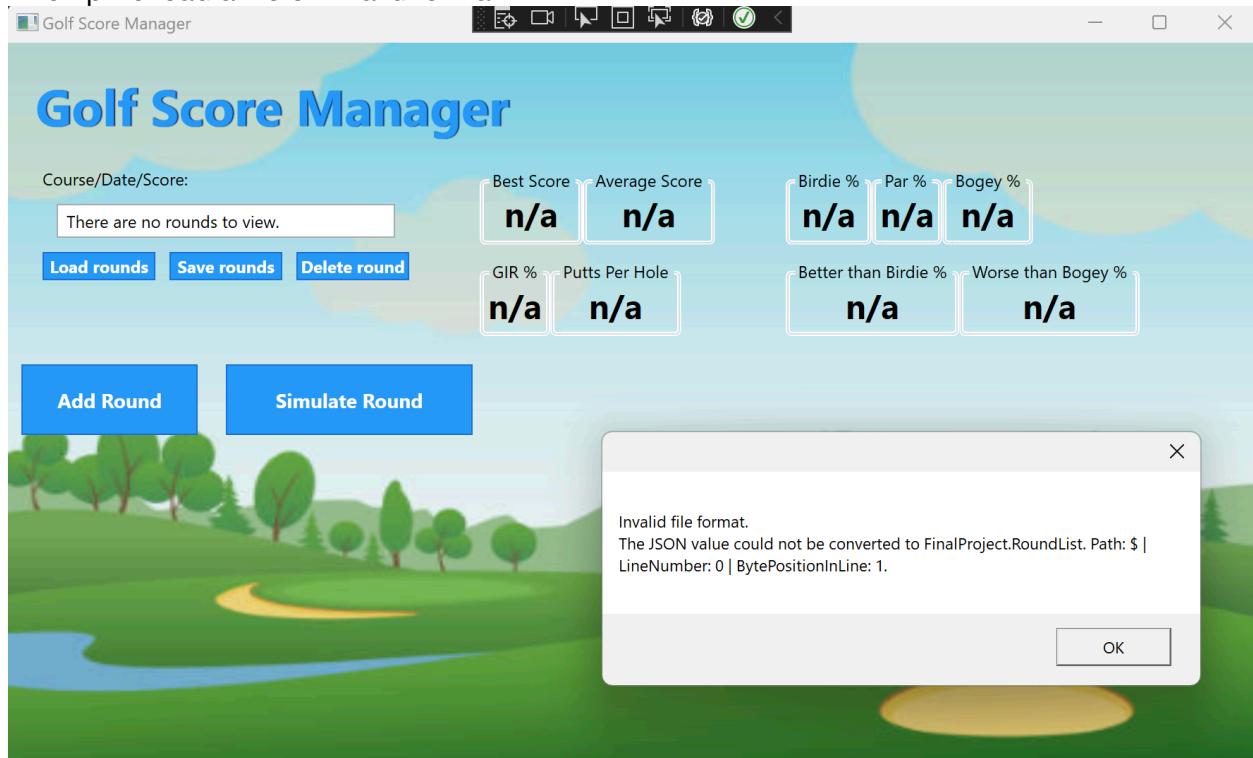
Attempt to delete without a round selected:



Attempt to load a file with an invalid path:



Attempt to load a file of invalid format:





## Enter New Golf Round

### Scorecard

Date: 5/4/2023

Course name: test

9 or 18:

- 9 holes  
 18 holes

Par for each hole:

1: 4    2: 4

Not all holes were filled out properly. Please make sure par/score/putt for all holes are filled out properly!

Score for each hole:

1: bad!    2: 5

Number of putts for each hole:

1: 2    2: 2

< >