College Tournament Program Design

# Written By: Benjamin Whybrow

# Software Development Lifecycle

|  |
| --- |
| The software development lifecycle is a process for planning, creating, testing and deploying an information system. It involves six phases: Requirement Analysis, Planning, Software Design, Software Development, Testing, Deployment. |
| Understanding the scope  This is the analysis stage of the SDLC. This involves establishing exactly what the client wants from the project. This will also involve making a list of priorities as a project could be limited on resources. This may also involve cutting out anything that is non-essential, this can be added later on down the line after the initial release of the project. Sometimes it’s also known as a “mission statement” which is written in terms of the needs of the client. |
| Identifying requirements and specification  This is a more detailed examination of the needs and requirements of the project. This includes things like noting inputs, outputs and processes needed within the project. Also known as “translating” business needs into terms relevant to software design. This can also include an investigation into the current infrastructure within the business and how to integrate the new system within it. |
| Designing the system  This will involve a consideration of how the needs of the client can be met using software. This will include detailed specification of user interfaces, program structure, any algorithms needed and how data will be stored and managed. Can also include entity relationship diagrams, data flow diagrams, pseudocode and input/output specifications. |
| Coding the program  This is where the design specification from the previous step is translated into functioning source code. This could include multiple or a singular programmer working on the project. |
| Testing the program  This is the process of checking whether the program is functional and identifying any bugs or problems within the program. All documentation of the bugs within the program are fed back to the implementation process and fixed. |
| Maintaining the system  The initial release of the program is released but bugs may still be within the program that weren’t picked up in the testing stage. Improvements or additions will be added throughout this process and may occur as the program/business grows in size. Adaptation may also include undiscussed factors that didn’t arise during the design process like a change of technology (Windows 7 -> Windows 10) or a natural disaster that would wipe out hard drives and leave irrecoverable data. |

# Tournament Design

## Team Events

The tournament will have the following team events:

|  |  |  |
| --- | --- | --- |
|  | Event Name | How the event will be scored |
| 1 | General Knowledge Quiz | Highest Scoring 3 Teams  1st gets 3 points, 2nd gets 2 points, 3rd gets 1 point, the rest get none (on the system) |
| 2 | Jigsaw Puzzle Race | First 3 teams to complete the puzzle (same as above) |
| 3 | Catchphrase | 3 Highest Scoring Teams (same as above with system scoring) |
| 4 | Scavenger Hunt | First 3 Teams to complete all the given tasks (awarded same as quiz) |
| 5 | Rowing Challenge | First 3 Teams to reach a certain goal (1500m for example) (awarded same as quiz) |

Prizes will be handed out for the top 3 highest scoring teams. The winning team will be determined by the highest number of points based off of how the events are scored.

## Individual Events

The tournament will have the following individual events:

|  |  |  |
| --- | --- | --- |
|  | Event Name | How the event will be scored |
| 1 | Darts | In what order a player got to 0 from 301 (4 points for the 1st player, 3 points for the 2nd, etc) |
| 2 | Board Games | Hierarchy of which a player wins said board game (Highest money in Monopoly, for example) (same as darts) |
| 3 | Puzzle Game (Hostage Situation, etc) | Hierarchy in which a player wins said game (same as darts) |
| 4 | Spelling Bee | Hierarchy in which a player lasts (last player gets 4 points, second to last gets 3, etc) |
| 5 | Chess | Hierarchy in which either player wins (winner gets 4 points, loser gets 1 point) |

Prizes will be handed out for the highest scoring individual. The winner will be determined by the highest number of points based off the scoring system.

# Software Solution Design

## Problem Definition Statement

The requirement for the program, as set out by the client, are a scoring system for an event day of which accounts for team and individual events and must accompany for 4 teams, each with 5 members. It must be able to calculate the overall scores and the overall winning team and individual.

The program will solve the following problems removing the need for pen and paper to write down scores and using a calculator and mental maths to figure out the winner(s)

Constraints which could limit the solution include limited amount of time to construct the program, not being able to use a fluent language of choice and having to learn a new language, will require a laptop/computer to use the program and the COVID-19 pandemic.

The benefits of solving this problem with software are that the data collection and calculation will be faster overall and allows for easy storage of the scores if the events take place over several days.

The impact of not solving this problem with software could be losing the current scores which are noted on paper and just a slower calculation process of the final winner.

The resources which are available to resolve this problem are creating a C# program and using files to store information permanently, allowing for future reference.

## Features of the software

The program will need input data relating to the scores for each event and the names of the teams/individuals.

The program will need to output winning team/individual and overall scores.

In order to output this data, the program will need to calculate the overall scores for each team/individual and the highest score from every team/individual.

The program will need to store team/individual names, scores and overall winners.

**All of the above can be stored in a file which can also be used to reference back to.**

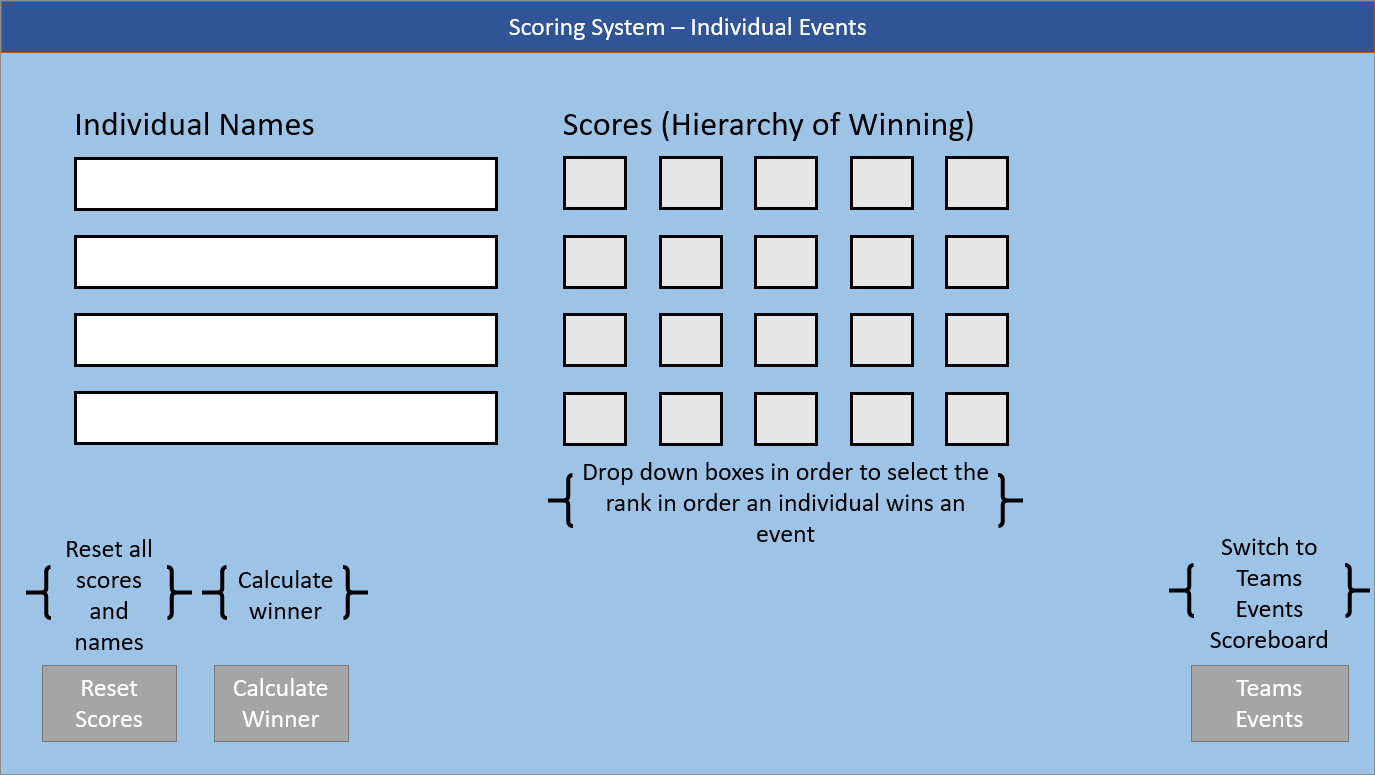
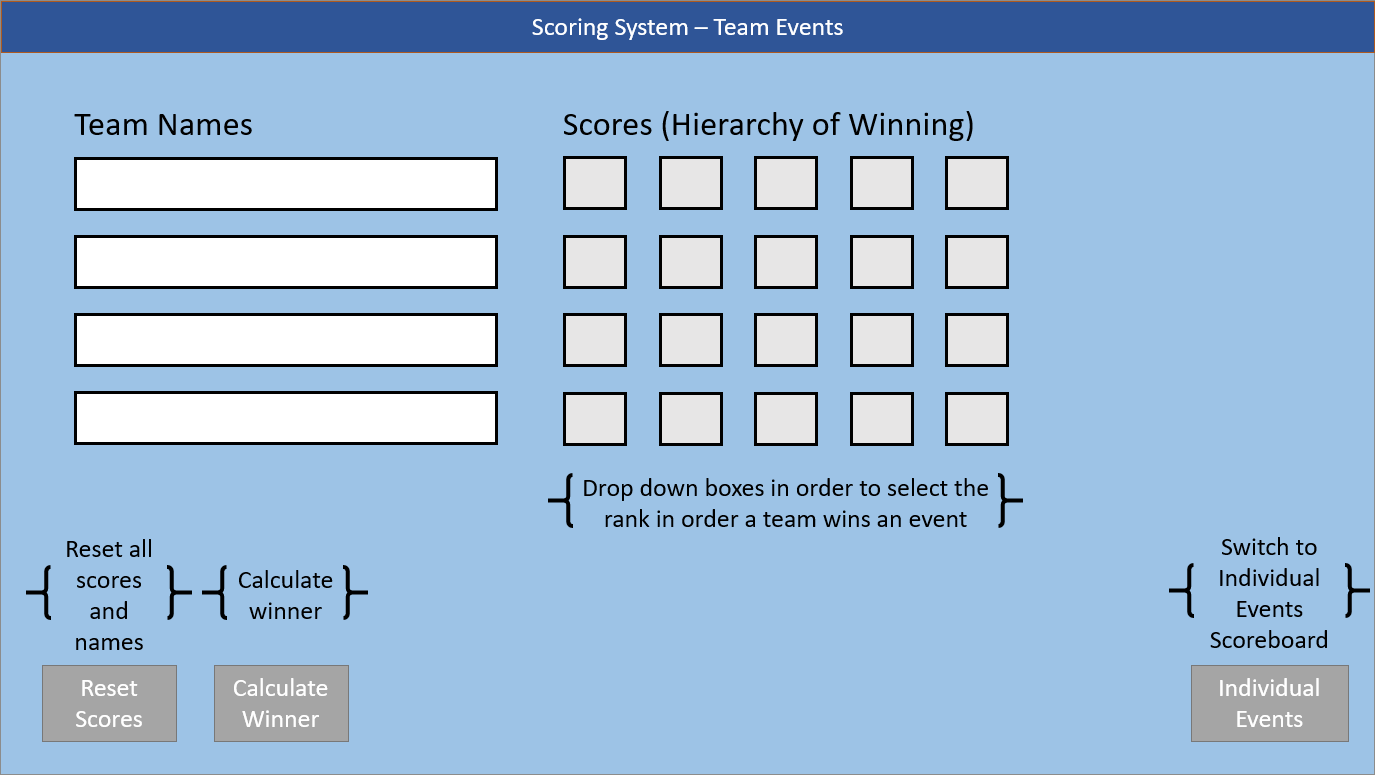
## Input validation

To make sure that any data handled by the program is valid it will perform the following checks on input data:

|  |  |
| --- | --- |
| **Input Data** | **Validation Check** |
| Team Names | String under 20 characters long |
| Individual Names | String under 20 characters long |
| Score | Number, Between 1 and 4 |

## User Interface

Create two designs for what the user interface of your program could look like and place them below:



## Plan for implementation

The following is my plan for completing the work needed for this project:

|  |  |
| --- | --- |
| Week Beginning | What I will accomplish in this week: |
| 23 November 20 | Software Development Life Cycle, Tournament Design, Software Solution Design |
| 30 November 20 | Input Validations, User Interface Designs |
| 7 December 20 | Algorithms and Test Plan |
| 14 December 20 | Begin work on the program |
| 11 January 21 | Complete 90% of the program with only a few minor tweaks and bug testing to be completed. |
| 18 January 21 |  |
| 25 January 21 |  |
| 1 February 21 | Gain feedback from 2 other students and make improvements on said feedback. |
| 8 February 21 | Design, program and evaluation will be complete and ready for submission. |

# Algorithms

## Pseudocode or Flowcharts

The program will use the following algorithms to calculate the required output data:

|  |  |
| --- | --- |
| **Calculate Total** | **Reset Scores** |
| Fetch array of counters For each value of the array  Sum1 = Sum + value  Every 5 counters, create new Sum for new team Sum1, Sum2, Sum3, etc Find highest value  Print highest value | Print message box with yes/no confirmation  If no, close message box and stop ongoing process  If yes, proceed with following code  Fetch array of counters  For each counter in array  Set value to 0  Print message box with validation of reset |

# Testing

## Feedback from others

This program design needs to be shared with two other people who will give their feedback on your plans.

## Feedback 1

**Feedback provided by:** Ivan Tihomirov

**Comments made:** Very good design of the program and it will be very helpful to fill all info from the results. Test Plan seems on point and nothing else to be put in. Calculations are perfect and will do the job. Interesting choice of games for the event.

## Feedback 2

**Feedback provided by:** James Avent

**Comments made:** I like the fact that you have included upgrades to technology (changing OSes) as a constraint as this could be a potential real-life problem. Your design is very pleasing to the eye and has a clear layout out as well. I also really like your table for pseudo code and the mention of entity relationship diagrams in your SDLC section.

Based on the feedback given above, no changes need to be made to the design.

## Test Plan

You will need to carry out tests to ensure that your finished program works as expected. Complete the following test plan to show what tests you will perform.

|  |  |  |  |
| --- | --- | --- | --- |
| Test No. | Test description and purpose | Test data | Expected result |
| 1 | Window view not scalable | Try to scale the window | Window stays the same size |
| 2 | Program starts fine | Opening the .exe file | Program starts up and displays team events form |
| 3 | Empty team name | No value | Error upon clicking any of the calculation buttons |
| 4 | Team name limit | String of 21 characters | Error upon clicking any of the calculation buttons |
| 5 | Empty individual name | No value | Error upon clicking any of the calculation buttons |
| 6 | Individual name limit | String of 21 characters | Error upon clicking any of the calculation buttons |
| 7 | A score not being changed from 0 | No change | Error upon clicking the “Calculate Winner” button |
| 8 | Entering a string into a score | “hi” | Input value is denied and reset back to 0 |
| 9 | Entering a value above 4 into the score | “5” | Input value is denied and reset back to 0 |
| 10 | Calculate Winner button does its job | Click the button | Winner is calculated and displayed in a message box |
| 11 | Individual Events button does its job | Click the button | Message box shows to confirm and if yes, switch to individual events form |
| 12 | Reset Scores button does its job | Click the button | Message box shows to confirm and if yes, all scores are reset back to 0 |
| 13 | Save Data button does its job | Click the button | All data is collected and saved to a .txt file |
| 14 | Load Data button does its job | Click the button | .txt file is read and data is inputted into the form |
| 15 | Load Data button is pressed with no existing file | Click the button | Error displays and process is cancelled |