**Olympic Madness**

It’s 2012, and aside from the world ending this December, it’s the year of the Olympics! This year the Olympics are being held in London, and Australia is set to do quite well in the medal standings.

Television broadcaster Channel 10 would like to show medal standings for all sports in the Olympics. They’d like this available as an App that can be downloaded by their customers, but they don’t really want to pay a production company to make it. As a result, their marketing division have decided to run a community competition to create an Olympics app “to rival the speed of the Australian men’s 4 X 100m swimming relay team” with a $5,000 cash prize for the winner. They did not have anyone on staff tech-savvy enough to create an SRS, so stole one from the boot of Eddie McGuire’s car. They’ve approached a bunch of government schools and asked students to submit entries by 12:45pm, Thursday 23rd August.

Channel 10 have placed a number of requirements on the App beyond how it functions. They want to be able to enter all of the information about which countries have won which medals. They have also requested that the application be easy to read both in sunlight and indoors, so it can be updated anywhere, at any time.

Strathmore Secondary College has decided to submit entries from all students in the Year 12 Software Development class, as cruel and unusual punishment. As a result, you are tasked with completing the App during class time. Your teacher has also said that she will take full credit for your code if you win, and thus claim the prize money for her coffee fund!

**Task Requirements**

**PART 1:**

Read the following SRS. It outlines the analysis of the new system. You have been “asked” to complete the entire software product.

* Write an **algorithm in pseudocode** to represent **reporting statistics (4.1.4).**
* Design a **data dictionary** that shows the core data within the software product.
* Design the **user interface** either on paper or in electronic format.
* Create a complete set of **criteria for evaluating** the efficiency and effectiveness of the solution.
* **Create the software module** using efficient code that appropriately uses data types and structures to create an effective solution to the problem that meets all of the software requirements specifications.
* Include sufficient **internal documentation** so that others will be able to modify the code as part of the whole system development. The documentation should be relevant, non-trivial, clearly expressed and well formatted.
* **Design** a **testing table** to test the program which contains an appropriate range of **test data** to thoroughly test all aspects of the software’s functionality and accuracy
* **Test** the program using test data and fill in the ‘result’ columns in the testing table.

**PART 2:**

* Outline any **legal obligations** of the programmer (you) in a short report.
* Discuss any **security of information** concerns that need to be considered in the final product in a short report.
* Create **user documentation** in the form of a **USER MANUAL**.

**\*Your report on legal obligations and security of information should be placed in a single Word Document named:**

***Part2-Report-YYMMDD.docx***

Software Requirements Specification

for

Olympics Stats Tracker

|  |  |
| --- | --- |
| **VERSION 1.0** | |
| **Author** | **Date** |
| Eddie McGuire | 2/August/2012 |
| Eddie McGuire | 4/August/2012 |
| Eddie McGuire | 7/August/2012 |
| Eddie McGuire | 9/August/2012 |
|  |  |
|  |  |

Preface

|  |
| --- |
| This is the Software Requirements Specification (SRS) for Olympics Stats Tracker (OST), intended to represent a collection of data based on Olympic results for the 2012 London Olympics. In particular, this SRS represents the solution requirements for a new system that will track the medal wins and statistics of any country competing in the Olympics  The intended audience of this SRS is Channel 10 representatives as well as the developer of the new system, a VCE Software Development student.  This document also acts as a contract between the development team and the client as it pertains to the whole system being delivered. |

# Introduction

## Purpose

|  |
| --- |
| The purpose of this document is to provide a detailed description of all the parameters and goals of the software product for the Olympics Stats Tracker App. |

# Scope

## Items within Scope

|  |
| --- |
| The items that are within scope of this project include:  - Algorithm design  - Data dictionary  - Testing table  - Evaluation criteria  - Creation of the software module and internal documentation  - Data validation  - Testing  - User Documentation  - Consideration of legal repercussions and security concerns  The system itself should provide the following:   * Entering event information * Entering and editing event winners * A report of country medal statistics (based on event winners) in sorted order, with best country first and worst country last. |

## Items not in Scope

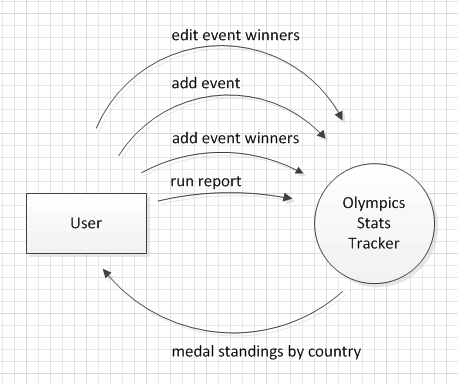
|  |
| --- |
| The items that are not within the scope of this project include:  - Creating a list of the countries competing in the Olympics in 2012.  - Retrieving saved country and event information for editing.  - Access restrictions on who can enter and edit information – this will be added later by Channel 9 programmers. |

## Operating Environment

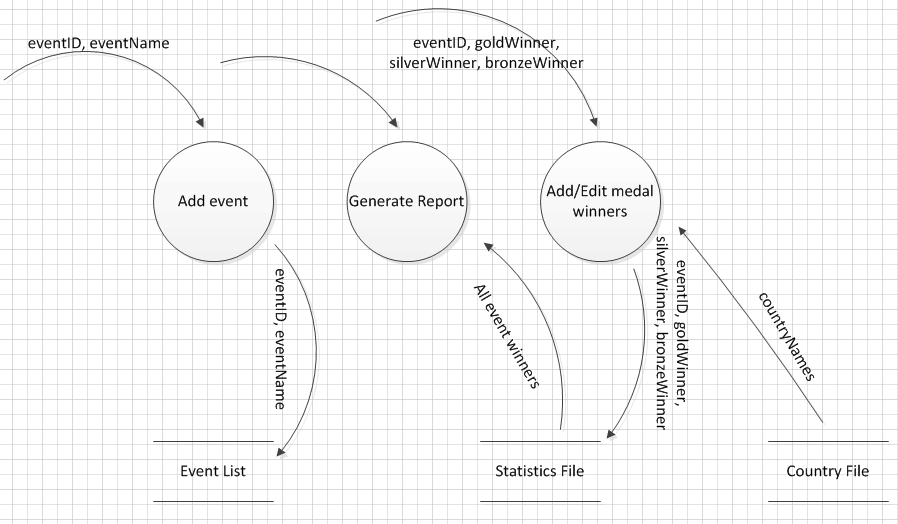
|  |
| --- |
| The software module must be fully functional when running on:  - a Samsung Galaxy S3 in EITHER vertical or horizontal orientation, but not both.  - screen resolution should fit 300 width or 720 length. |

# Documentation

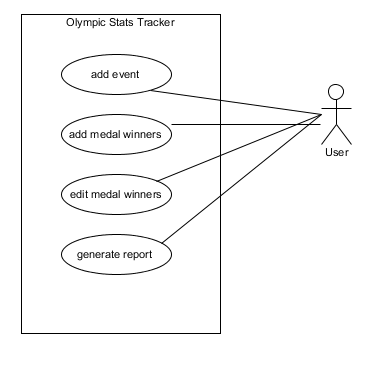
## Context Diagram



## Data Flow Diagram Level 1



## Use Case Diagram



# Requirements

## Functional Requirements

The main program should be run off a “menu” screen, where all functions are available to the user from the one screen.

### ADD EVENT

Explanation:

This screen will allow the entering of event information. Information consists of the event name and a unique event ID (numeric).

Inputs:

Event Name

Event ID (unique)

Sequence of Operation:

Input will come in from the User Interface, all items are required fields. No two events should have the same name, so this should be validated before saving. Once data is entered and validated, it should be saved to an event file.

Functionality is not required for editing or deleting events.

Outputs:

A pop-up box should be displayed showing the success or failure of saving the event information. Once information is saved, the user should be returned to the main menu automatically.

### ADD EVENT WINNERS

Explanation:

Winner information is fairly straightforward. Only the country name is tracked against first, second and third place. A single country may win more than one medal (eg: gold and bronze).

Inputs:

Event ID and Name

GoldWinner

SilverWinner

BronzeWinner

Sequence of Operation:

Event IDs should be selectable from a list of stored events – the ID and name should be displayed in a single field rather than two fields.

**Countries are obtained from the file named allCountries.txt.**

Gold, silver and bronze winners are saved only by the name of the country – the name of the athlete is not stored. Once the information is entered and validated, it is added to a file **except** for the event name – only the ID and the three winning countries should be stored.

Outputs:

If the event already has winners stored in the event statistics file, an error should be displayed in the pop-up and the information should not be saved. If the winners have not already been entered into the statistics file, a pop-up box should be displayed showing success or failure of saving the event information. Successful saves should return the user to the main menu automatically.

### EDIT WINNERS

Explanation:

Given a list of all events that have winners, allow a user to change the winners of any particular event.

Inputs:

Event ID

Sequence of Operation:

The user should be able to select an event from a list of events that have winners saved to the statistics file. From this, they should be taken to a screen that looks like **4.1.2. Add Event Winners**, with the information pre-filled into the screen. They should be able to change the winners of the event, but not the event itself. These changes should be saved to the event statistics file once the save button is pressed.

Outputs:

A pop-up box should be displayed showing success or failure of saving the updated event statistic information. Once information is saved, the user should be returned to the main menu automatically.

### GET COUNTRY STATISTICS

Explanation:

This should be a summary page listing all event statistics, by country. For each country, it should list the number of events for which they have won gold, silver and bronze. It should also list a “total medal tally” of all medals.

Inputs:

None

Sequence of Operation:

For each country that is listed in the country file, the event statistics should be checked to see if any medals have been won. Gold, silver and bronze wins should be counted, as well as the total number of medals won. The report should display all country stats and be **sorted based on the total number of wins** – the country with the most number of wins should be first in the list, with the medal tally breakdown listed next to them.

If a country has not won any medals, they should not be displayed.

Outputs:

An on-screen report listing for each team:

Country Name, Gold Wins, Silver Wins, Bronze Wins, Total Medals

A button should be displayed at the bottom of the report allowing the user to choose to return to the main menu.

## Non-Functional Requirements

|  |  |  |
| --- | --- | --- |
| **No.** | **Requirement** | **Notes** |
| NFR01 | Reliability | The statistics shown must be 100% accurate for any valid inputs. |
| NFR02 | Accessibility | The screen font colour and background colour should have high contrast. |

# Constraints

|  |
| --- |
| Within the Olympic Stats Tracker:   * The interface will be viewed through a Samsung Galaxy S3. * The interface should be high contrast so it may be viewed both indoors and outdoors. |