|  |
| --- |
|  |

|  |
| --- |
| **Technical Report Template** |
| ICTPRG549 – Apply Intermediate Object-Oriented Language Skills AT2 |
|  |
|  |
|  |
| **Luke Wait** |
| **10/10/2022** |
|  |

Table of Contents

[Part 1 – Planning Documentation 3](#_Toc99275556)

[Task 1&2 – Email 3](#_Toc99275557)

[Task 3 – Report 4](#_Toc99275558)

[Proposed data structures 4](#_Toc99275559)

[Object Oriented Language details 4](#_Toc99275560)

[Concepts of object oriented language 4](#_Toc99275561)

[Internal and External Documentation plan 4](#_Toc99275562)

[Conceptual model of data requirements for database project 4](#_Toc99275563)

[Task 4 – Informal Meeting 5](#_Toc99275564)

[Part 2 – Application Development 5](#_Toc99275565)

[Task 1– NetBeans project in Java 5](#_Toc99275566)

[NetBeans project database classes and connection 5](#_Toc99275567)

[App.Config file 5](#_Toc99275568)

[Task 2 – Class Library 5](#_Toc99275569)

[Connection String and Settings 5](#_Toc99275570)

[Try…Catch 5](#_Toc99275571)

[Method Overloading 6](#_Toc99275572)

[Manipulate Database Schema 6](#_Toc99275573)

[Programming Guidelines 6](#_Toc99275574)

[Task 3 – Graphical User Interface 7](#_Toc99275575)

[Frame Design 7](#_Toc99275576)

[Control Events (Event Handling) 7](#_Toc99275577)

[Sorting and Searching 7](#_Toc99275578)

[Try…Catch (Exception Handling) 7](#_Toc99275579)

[Programming Guidelines 7](#_Toc99275580)

[Part 3 – Documentation 8](#_Toc99275581)

[Task 1 – Errors and Debugging 8](#_Toc99275582)

[Defect Logs 8](#_Toc99275583)

[Debugging Tools in Action 8](#_Toc99275584)

[Task 2 – Stability and Optimisation 9](#_Toc99275585)

[Profiling tools 9](#_Toc99275586)

[Profiling Summary 9](#_Toc99275587)

[Task 3 – Testing of functionalities 10](#_Toc99275588)

[Testing Summary 11](#_Toc99275589)

# Part 1 – Planning Documentation

## Task 1&2 – Email

**EMAIL TO PROJECT MANAGER**

|  |  |
| --- | --- |
| **To:** | Mark Brown |
| **From:** | Luke Wait |
| **Subject:** | Re: Gold Coast E-Sports Desktop Application V2 (with MySQL database) |
| **Message:** | Greetings Mark,  Following our discussion about the development of a desktop application which connects to a MySQL database to manage and record competition information for e-sports teams, I have drawn up a list of required features:   * Must be able to **run on Windows 10** * Must **connect to a MySQL database** (ERD diagram and SQL file that creates tables and initialises row data for each table provided). * Must provide functionality to create, view, update and delete data to the database tables: event, competition, game and team * Must provide the ability to **add new entries** to existing data – this includes: - **Add a new event** (event name, event date in the format “yyyy-mm-dd”, and location) - **Add a new competition result** for two (2) competing teams (event, game, team 1, team 2, team 1 points earned, and team 2 points earned) - **Add a new team** (team name, contact person, phone, email) * Must be able to **update existing team** data (e.g. change contact person name, phone, email) * Must be able to **display competition results** in a table by selecting an event (or defaulted to “All events”). The display is to be further filtered by selecting a team. * Must be able to **display a leader board of teams and accumulated points for all events** (in order of team with the highest tally of points) * Must be able to **display a second leader board of teams and accumulated points for a selected (specific) event** (in order of team with the highest tally of points for the chosen event)   Please advise if I’ve understood the requirements in full. When you are comfortable that all considerations have been addressed, I’ll proceed with development.  Regards,  Luke Wait |

## Task 3 – Report

### Proposed data structures

Classes:

* GC\_Egames\_V2\_GUI
* DB\_Read (to be converted to .jar file)
* DB\_Write (to be converted to .jar file)

Java dynamic data structures:

GC\_EGames\_V2.1 doesn’t use any java dynamic structures, instead it accesses the database for all data storage and retrieval!

GC\_EGames\_V2.2 utilizes ArrayLists to sort data sets from the event leader board array:

* ArrayList<Object> uniqueTeamNamesList
* ArrayList<Integer> totalPointsList

### Object Oriented Language details

* Java SE (Standard Edition)

### Concepts of object oriented language

* Abstraction (use of classes and objects)
* Encapsulation (public methods accessing private data)
* Inheritance (main class extends javax.swing.JFrame)
* Polymorphism (override methods - toString())

### Internal and External Documentation plan

Internal:

* Header comments at the beginning of each source code file with date and version
* Comments at the beginning of each method
* Comments to explain complex algorithms or logic
* Comments to declare noteworthy variables and objects

External:

* Javadocs (class documentation)
* GitHub (version control)

### Conceptual model of data requirements for database project

team:

* name (string) PRIMARY KEY
* contact (string)
* phone (string)
* email (string)

event:

* name (string) PRIMARY KEY
* date (date)
* location (string)

game:

* name (string) PRIMARY KEY
* type (string)

competition:

* competitionID (integer) PRIMARY KEY
* eventName (string) FOREIGN KEY (references event.name)
* gameName (string) FOREIGN KEY (references game.name)
* team1 (string) FOREIGN KEY (references team.name)
* team2 (string) FOREIGN KEY (references team.name)
* team1Points (integer)
* team2Points (integer)

## Task 4 – Informal Meeting

* Record the date and time of the meeting and any notes that were taken during the meeting below

|  |  |
| --- | --- |
| Date/Time: | Thurs 15.09.22 10:00am |
| Notes: | Meeting with Hans.  Discussed all functionalities for project with demonstrations made on mock-app.  Functionalities: - refer to Task 1&2 – Email (also see header comments in GUI class).  Hans also raised questions around security to be implemented at a latter date. Storing login credentials in a notepad file isn’t advised! | |

# Part 2 – Application Development

## Task 1– NetBeans project in Java

The following is a checklist for the project requirements submitted with the full NetBeans project

### NetBeans project database classes and connection

* Project in NetBeans shows the separate database connection classes (for read and write) and the GUI class.
* The database connection classes are compiled to a separate .jar file and this .jar file is utilised by the GUI class.

### App.Config file

* Include the app.config file content showing the database connection settings

## Task 2 – Class Library

The following is a checklist for the source code requirements submitted with the full NetBeans project:

### Connection String and Settings

* Java file i/o code that retrieves the connection string values from external app.config file.

### Try…Catch

* Suitable try…catch (exception handling) for the file i/o and database connection operations

### Method Overloading

* Java code showing Method Overloading (note: this can be a simple override toString() method for a class)

### Manipulate Database Schema

* Enter the Java code (here) that could be used to create, modify or drop database tables or other schema components:

### Programming Guidelines

* Ensure that suitable comment blocks, methods and variable declarations with proper naming conventions and indentations are utilised in your project

## Task 3 – Graphical User Interface

The following is a checklist for the GUI requirements submitted with the full NetBeans project:

### Frame Design

* Frame design view showing buttons, labels, textboxes etc.

### Control Events (Event Handling)

* Java code that responds to user events such as Button click event (event handler methods)

### Sorting and Searching

* Java code used to sort and search in your application

### Try…Catch (Exception Handling)

* Relevant Java code with try…catch from the GUI project

### Programming Guidelines

* GUI code from any class that shows suitable comment blocks, methods and variable declarations with proper naming conventions and indentations

# 

# Part 3 – Documentation

## Task 1 – Errors and Debugging

### Defect Logs

* Include all syntax and logic errors that you have come across during the development of your application and what steps were taken to solve the identified issues

|  |  |
| --- | --- |
| **Syntax / Logic Errors** | **Solutions Applied** |
| **Syntax**  Spelling errors | Fixed spelling |
| **Syntax**  Missing brackets/squiggles | Added brackets/squiggles |
| **Syntax**  Missing semi-colon | Added semi-colon |
| **Logic**  Comparison operator missing second character (eg ==) | Added second character |
| **Logic**  Overwriting csv despite no data in jTables | Moved jTable row check to before opening the buffered reader |
| **Logic**  Csv data not lining up with database columns/rows | Changed variables in for loop within for loop to track rows of jTable properly |
| **Logic**  Index out of bounds | Adjusted for loop to accurately represent the bounds of loop |

### Debugging Tools in Action

* Screenshot of the code with breakpoints and watches

Example: set a breakpoint at a for loop and set a variable (teamsStrArray[i]) to watch for changes to check that data was being added as expected, with correct formatting.



* Explain the steps on how to set and use breakpoints and watches

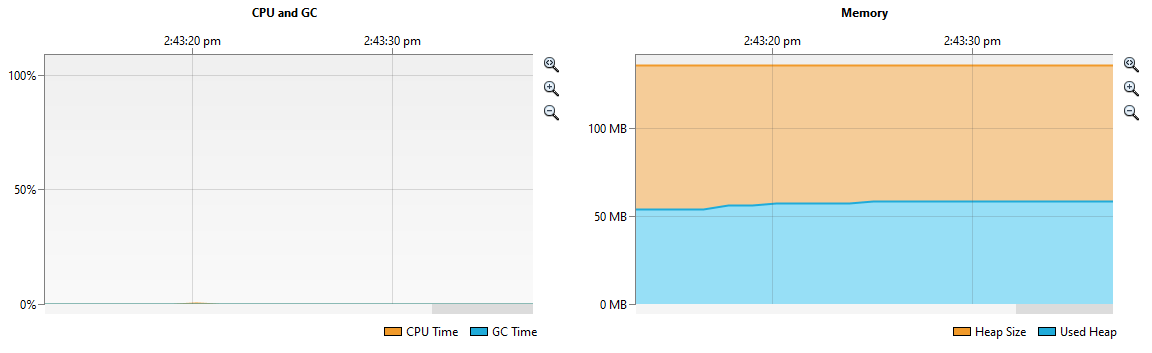
1. Check the system messages for clues as to where to problem is occurring
2. Set a breakpoint at the beginning of the block of code suspected of causing the problem by clicking on the line number
3. Set a watch by highlighting and dragging variables to be tracked into the lower window under the Watches tab
4. Step through the code with F8 and monitor changes to the watched variables for unexpected changes to pinpoint the problematic line of code

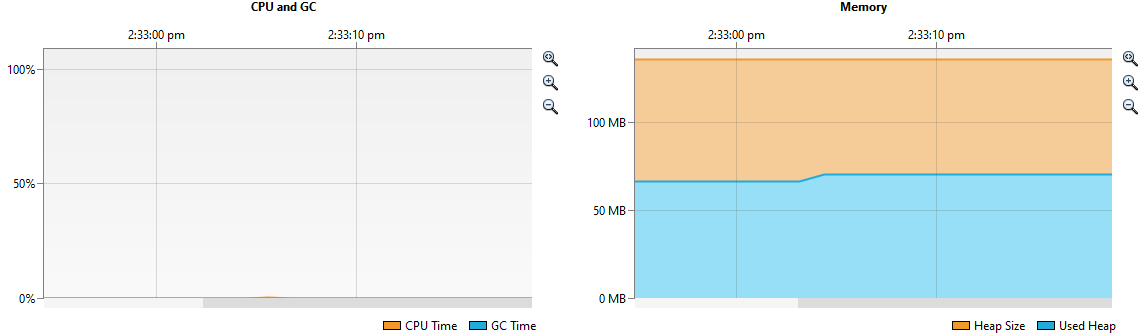
## Task 2 – Stability and Optimisation

### Profiling tools

* Screenshot of the results of running a Memory profiling tool and CPU profiling tool against your developed application

GC\_EGames\_V2.1 (uses database for storage):



GC\_EGames\_V2.2 (uses arrays to store data retrieved from database):  


### Profiling Summary

* Outline any issues identified, and any possible optimisations that have been identified as part of this process

No issues were identified during the profiling process. CPU usage is negligible on both versions. Memory usage is slightly higher on V2.2 which is to be expected as it stores retrieved data in arrays/arraylists. The payoff is the number of SQL queries and database connections is greatly reduced.

* If an optimisation or correction is required, provide an explanation of how you may solve this issue

No correction required. Have already developed 2 versions to optimise application efficiency.

## Task 3 – Testing of functionalities

Prepare tests that will be used to confirm application stability and functionality against at least **5** requirements of the Database side of the project and at least **5** requirements of the GUI side of the project. Ensure that you set up specific test steps with specific input data – refer two examples provided below.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Function to test** | **Test Steps** | **Expected Result** | **Actual Result** | **Fix Required** | **Status**  **Pass/Fail** |
| 1. GUI: Add a new event (test with all validated inputs) | 1.1 Launch app  1.2 Click ADD NEW EVENT tab  1.3 Enter "Coomera Open E-G" as the new event name  1.4 Enter "2022-02-26" as the date  1.5 Enter "TAFE Coomera" as the location  1.4 Click ADD NEW EVENT button  1.5 When pop-up appears, click the YES button to accept | Under EVENT COMPETITION RESULTS tab panel, the Event combo-box contains  "Coomera Open E-G (26-Feb-2022 TAFE Coomera)"  Under ADD NEW COMPETITION RESULT tab panel, the Event combo-box contains  "Coomera Open E-G" | Under EVENT COMPETITION RESULTS tab panel, the Event combo-box contains  "Coomera Open E-G"  Under ADD NEW COMPETITION RESULT tab panel, the Event combo-box contains  "Coomera Open E-G" | Under EVENT COMPETITION RESULTS tab panel, the Event combo-box (new event string value) is missing the bracketed data and location details e.g.  "Coomera Open E-G (26-Feb-2022 TAFE Coomera)" | Fix made to string values of Event combo-box and retested  PASS |
| 2. GUI: Add new competition results (test with all validated inputs) | 2.1 Launch app  2.2 Click ADD NEW COMP RESULTS tab  2.3 Enter “Fornite Battle Royale” as the event  2.4 Enter “Fortnite” as the game  2.5 Enter “BioHazards” and “Buttercups” as team 1 and 2  2.6 Enter “2” and “0” as team 1 and 2 points  2.7 Click ADD NEW COMP RESULT button  2.8 When pop-up appears, click the YES button to accept | Under EVENT COMP RESULTS tab panel, the competition results table contains the saved details  Under EVENT COMP RESULTS tab panel, the event leader boards updated to show the new accumulated points totals for teams involved | Under EVEN COMP RESULTS tab panel, the competition results table contains the saved details  Under EVENT COMP RESULTS tab panel, the event leader boards updated to show the new accumulated points totals for teams involved | No fix required as actual results match expected results | PASS |
| 3. GUI: Add new team (test with all validated inputs) | 3.1 Launch app  3.2 Click ADD NEW TEAM tab  3.3 Enter “Teamy McTeam” as new team name  3.4 Enter “Max Power” as contact name  3.5 Enter “0400000000” as phone number  3.6 Enter “maxpower@  hotmale.com” as the email address  3.7 Click ADD NEW TEAM button  3.8 When pop-up appears, click the YES button to accept | Under EVENT COMP RESULTS tab panel, the Team combo-box contains “Teamy McTeam”  Under NEW COMP RESULTS tab panel, the Team combo-boxes contain “Teamy McTeam”  Under UPDATE EXISTING TEAM tab panel, the Team combo-box contains “Teamy McTeam” and when selected the text fields are populated with the team data | Under EVENT COMP RESULTS tab panel, the Team combo-box contains “Teamy McTeam”  Under NEW COMP RESULTS tab panel, the Team combo-boxes contain “Teamy McTeam”  Under UPDATE EXISTING TEAM tab panel, the Team combo-box contains ‘Teamy McTeam” however when selected the text fields are not populated correctly | Under UPDATE EXISTING TEAM tab panel, the text fields display information for incorrect team or no data at all | Fix made to method which populates text fields. Used a gate to prevent combo box events from occurring during update  PASS |
| 4. GUI: Update team (test with missing inputs) | 4.1 Launch app  4.2 Click UPDATE EXISTING TEAM tab  4.3 Choose “Teamy McTeam” from Team combo box  4.4 Remove “Max Power” from the contact name  4.5 Click UPDATE EXISTING TEAM button | When clicking UPDATE EXISTING TEAM an error pop-up indicating “contact person’s name required” displays | When clicking UPDATE EXISTING TEAM a pop-up asking to continue displays | Validation for empty fields needed as contact name, phone number and email address is a requirement of every team | Added empty field checks to the update existing team validation method  PASS |
| 5. GUI: Write to external leader board csv | 5.1 Launch app  5.2 Choose “Fornite Battle Royale…” from event combo box  5.3 Choose “All teams” from team combo box  5.4 Click “Export leader boards as CSV files” button | When clicking “Export leader boards as CSV files” button a pop-up indicated that data was successfully written to “eventTeam  Scores.csv” displays (single event) | When clicking “Export leader boards as CSV files” button a pop-up indicated that data was successfully written to “eventTeam  Scores.csv” displays (single event) | No fix required as actual results match expected results | PASS |
| 6. Database: Display all competition results | 6.1 Launch app  6.2 Select the EVENT COMPETITION RESULTS tab  6.3 Select ALL EVENTS in event combo box  6.4 Select ALL TEAMS in the team combo box | 18 records displayed – Columns: Game, Team 1, Pt (team 1), Team 2, Pt (team 2) | 18 records displayed – Columns: Game, Team 1, Pt (team 1), Team 2, Pt (team 2) | No fix required as actual results match expected results | PASS |
| 7. Database: Display in bottom leader board table when ‘All events’ is chosen. Total points in descending order. | 7.1 Launch app  7.2 Select the EVENT COMP RESULTS tab  7.3 Select ALL EVENTS in event combo box | 4 records displayed in bottom table – Columns: Team, Total points – all events  Records are displayed in descending order of total points  0 records displayed in top table | 6 records displayed in bottom table – Columns: Team, Total points – all events  Records are displayed in descending order of total points | There are two teams that are displayed twice with different values. If they are merged they will display the right number of teams with the correct total points | Adjusted SQL statement. Used a union to merge the columns and then selected from the merged table and grouped by team  PASS |
| 8. Database: Display in top leader board table when a particular event is chosen. Total points in descending order. | 8.1 Launch app  8.2 Select the EVENT COMP RESULTS tab  8.3 Select “LoL Epic 2022…” from the event combo box | 0 records displayed in bottom table  4 records displayed in top table – Columns: Team, Total points – chosen event  Records are displayed in descending order of total points | 0 records displayed in bottom table  4 records displayed in top table – Columns: Team, Total points – chosen event  Records are displayed in descending order of total points | No fix required as actual results match expected results | PASS |
| 9. Database: Display competition results by team | 9.1 Launch app  9.2 Select the EVENT COMPETITION RESULTS tab  9.3 Select ALL EVENTS in event combo box  9.4 Select “Nerang Necros” in the team combo box | 9 records displayed – Columns: Game, Team 1, Pt (team 1), Team 2, Pt (team 2) | 0 records displayed – Columns: Game, Team 1, Pt (team 1), Team 2, Pt (team 2) | Data not displaying in competition results table, should be returning 9 records | Fixed SQL statement syntax error in displayComp  Results() method  PASS |
| 10. Database: Show team data when selected from combo box | 10.1 Launch app  10.2 Select UPATE EXISTING TEAM tab  10.3 Choose “Buttercups” from combo box  10.4 Choose “Biohazards” from combo box | Contact name, phone number and email address are populated with data relevant to BioHazards  Data in text fields changes to Buttercups team data when selected  Data changes back to BioHazards when selected | Contact name, phone number and email address are populated with data relevant to BioHazards  Data in text fields doesn’t change when new team is selected | Data to update with team selection in combo box | Fixed SQL statement syntax in displayTeam  Data() method  PASS |

### Testing Summary

* Analyse the results and provide a test summary about the errors that occurred during testing and the possible solutions or fixes.

Testing functionality was performed as the app was developed (unit testing). I worked systematically through the functional requirements and addressed any syntax and method logic problems as they arose.

The integration and system testing lead to more thought about data flow solutions. I came into some problems when trying to update the combo boxes with newly saved data as it would trigger the combo box events. Using the debugger to follow the program as it was being used, I was able to overcome the issue by using the comboboxstatus variable to gate the update methods. This also meant the updated combo boxes would reflect the database in terms of order (alphabetic for teams, date for events).

The other challenge I found was trying to problem solve any SQL/database queries. I found utilising system messages to inform the user about all SQL/database query results to be helpful. These show whether an error has occurred, as well as displaying the returned results whenever the program interacts with the database. Sometimes the SQL query found records but returned null values so having this constant feedback was useful in finding solutions. It also provided some insight as to how many times the program made a connection to the database, making it easier to track when combo box events where occurring.

In summary: debugger, watches and system messages are useful! I hoped to get more out of the profiling tool, unfortunately it didn’t want to co-operate on the SQL side of things.