ICTPRG549 AT2: Gold Coast E-Sports V2



# Scenario

**Gold Coast E-Sports** is a junior e-sports organisation which holds events for juniors throughout the local Gold Coast area. They currently hold about 12 events per year in various locations within the region where young children and teens compete in competitive video games with their teams.

Over the last few months, the organisation’s manager, Mark Brown has noticed it is becoming increasingly difficult to keep track of event results as the organisation keeps growing and having more teams compete each week. Currently the event results are being collected on a series of handwritten and excel documents, but recent events have required Mark and his team to enter repetitive data as some information is used in multiple areas and therefore being recorded in multiple documents.

Mark has approached you to develop a software application to help him keep track of event results for his local region’s e-sports competitions and reduce the need for entering repeated data in multiple locations. In one of his previous roles as part of the local football association he remembers them having a system which used a database and a windows application to enter their results which organised their data and allowed for easy reporting.

Mark has asked if you could look as building something with a similar structure that he can use for his events.

# Requirements

Mark is intending to run the system you develop on either his personal Windows 10 laptop or Windows 10 PC so he can enter data at home or directly at events. Mark desires a GUI based application that connects to an external database.

Therefore, he has asked that the system:

* 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)

The following are legacy examples of how team details were recorded (on paper) for entry into the database.

|  |  |  |  |
| --- | --- | --- | --- |
| **Team Name** | **Primary Contact** | **Contact Phone** | **Contact Email** |
| Coomera Bombers | James Taylor | 0433948765 | [jamestaylor123@coolmail.com](mailto:jamestaylor123@coolmail.com) |
| Nerang Necros | Sophie Jamieson | 0440888222 | [sophie\_jamieson@geemail.com](mailto:sophie_jamieson@geemail.com) |

The following are legacy examples of how event details were recorded (on paper) for entry into the database.

|  |  |  |
| --- | --- | --- |
| **Date** | **Location** | **Event Name** |
| 2022-02-18 | TAFE Coomera | LoL Epic 2022 |
| 2022-01-21 | TAFE Coomera | Dota 2 Battle Royale |

The following are legacy examples of how game details were recorded (on paper) for entry into the database.

|  |  |
| --- | --- |
| **Game Name** | **Type** |
| DOTA 2 | Team or Solo (MOBA) |
| League of Legends | Team (MOBA) |

The following are legacy examples of how competition details were recorded (on paper) for entry into the database.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Event Name** | **Game Name** | **Team 1** | **Team 1 Points** | **Team 2** | **Team 2 Points** |
| LoL Epic 2022 | League of Legends | BioHazards | 0 | Buttercups | 2 |
| Dota 2 Battle Royale | DOTA 2 | Coomera Bombers | 0 | Nerang Necros | 2 |

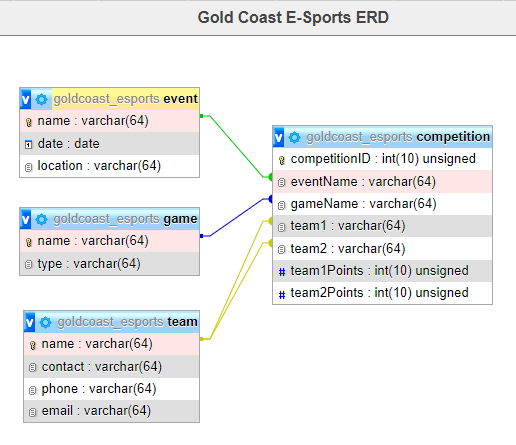
Mark uses a MySQL database named “goldcoast\_esports”.

The MySQL database contains 4 tables (attributes and keys are listed below):

* **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)

**Constraint rules for points**: For any given competition, the team1Points + team2Points must be equal to 2 (team which wins is given 2 points; the team which loses is given 0 points; a draw is 1 point each).

**Gold Coast E-Sports Entity Relationship Diagram** (from MySQL database)



The following are screen shots taken from the MySQL database for each table:

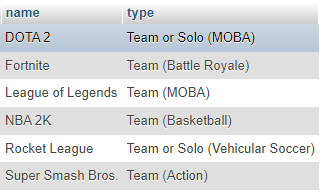
**Team**



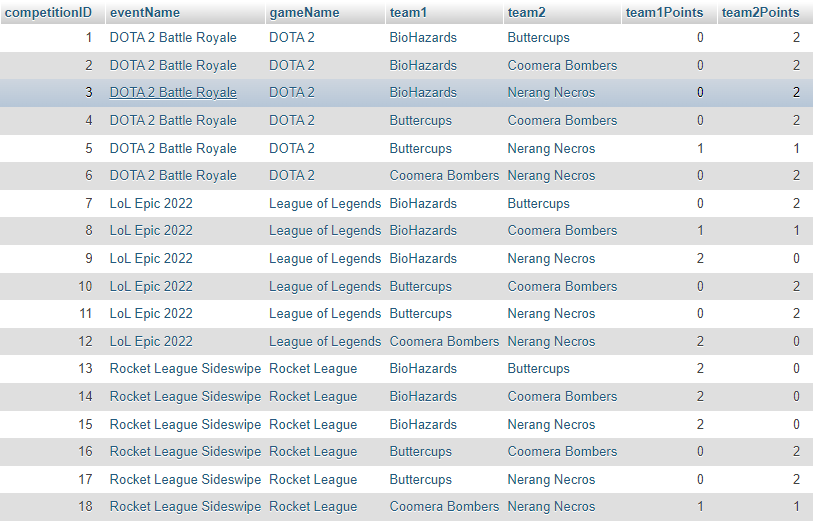
**Event**



**Game**



**Competition**



# GUI Design and Functionalities

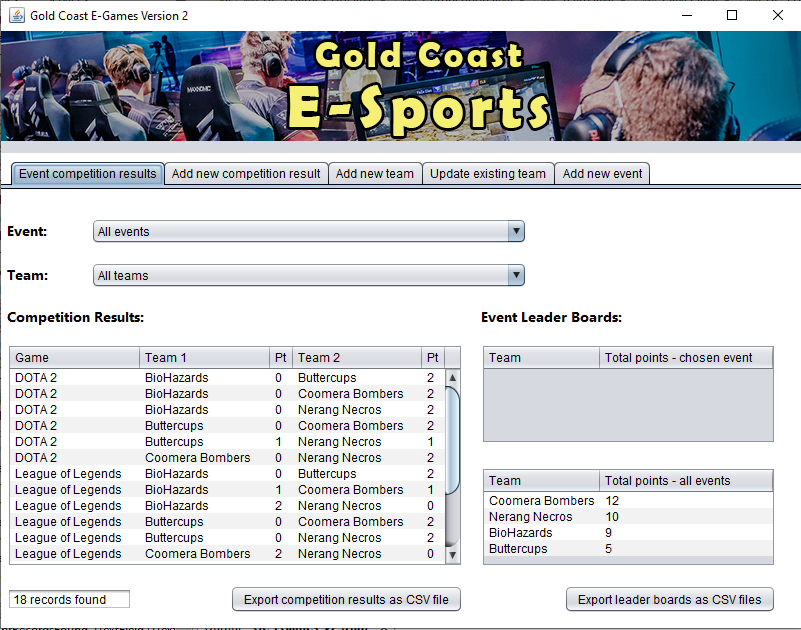
The following describes a suggested GUI design (including functionalities) for the Gold Coast E-Sports V2 prototype application:

When the program launches, it gets all event, team, game, and competition row data from the MySQL database.

The GUI application displays the header jpeg image (provided in the support files), along with five (5) tabbed panels:

* Event competition results
* Add new competition result
* Add new team
* Update existing team
* Add new event

1. The **EVENT COMPETITION RESULTS** tab is selected by default. Refer screen shot example below.

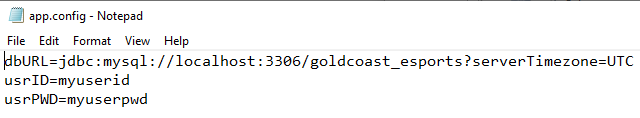


This tab panel contains:

* 4 x JLabels
* 2 x JComboBoxes
* 3 x JTables
* 1 x JTextFields
* 2 x JButtons

To display table data from the database in this tab panel, it is suggested that you develop a class (e.g. **DB\_Read**) which makes a connection to the goldcoast\_esports database and executes read-only SELECT/FROM sql statements that extract and return their resulting table sets back to the instance variables. The DB\_Read constructor method is to read an external file named “app.config” to get the MySQL database URL, the user id and the user’s password in order to access the database.

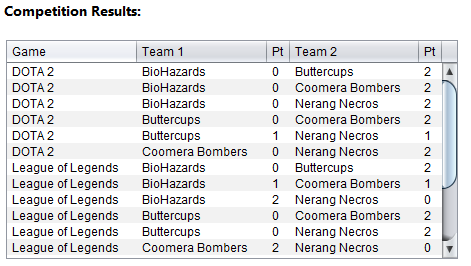
The following is a screen shot of an example external file containing the read-in data.



In a similar manner, you can create a second class (e.g. **DB\_Write**) which makes the database connection (goldcoast\_esports database), and executes write-type sql statements, such as INSERT INTO and UPDATE. This class also needs to read the database url, user id and password from the same external file – “app.config”.

**NOTE: For assessment purposes, you will need to create a JAR file to archive/compress these 2 classes. The JAR file can then be integrated with the GUI project.**

When the program opens, the competition results JTable displays ALL competitions from the competition table with columns: Game, Team 1, Team 1 Points (Pt), Team 2, Team 2 Points (Pt).



A text field below displays the number of records found.



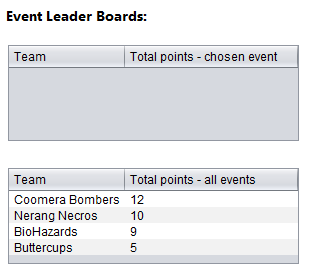
There are two (2) Event Leader Boards displayed as JTables:

The first JTable leader board with the column “Total points – chosen event” shows the teams filtered by a CHOSEN EVENT. When the program launches, the default chosen event in the event combo box is “All events” --- therefore, the leader board with the column “Total points – chosen event” shows nothing.

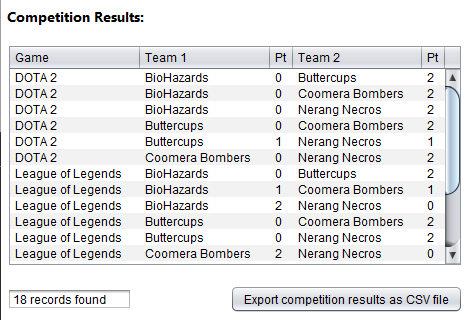
The second JTable leader board with the column “Total points – all events” displays the current total accumulated points for ALL EVENTS (for each team).

To display the leader board for all teams with accumulated points for ALL EVENTS, the event combo box selected item must be “All events”.





The competition results displays all 18 records (being all of the data for all events).



If another choice is made for the event (other than “All events”), then the “All events” leader board shows a blank. The leader board that displays “by chosen event” now displays the total points for each team according to the chosen event made in the event combo box.

Refer example screen shot below.

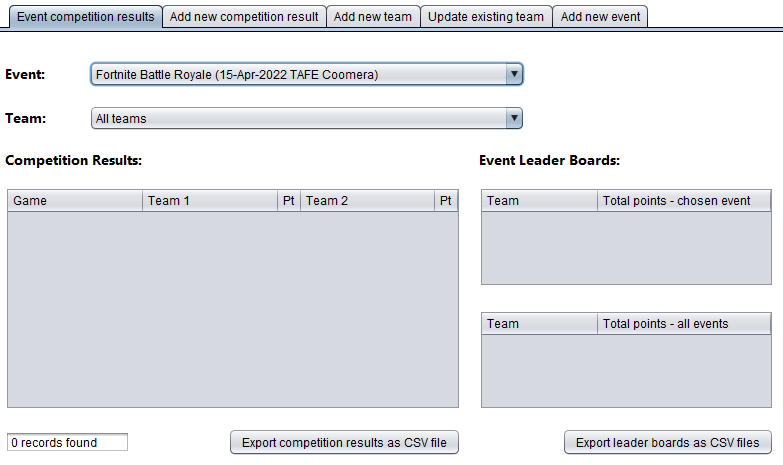


Note: The choice of “All teams” or any other selected team in the team combo box makes no difference to what is displayed in any of the leader boards.



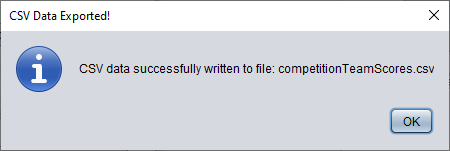
If an event was chosen that did not have any competition result, then all 3 JTable components should be blank.

This example shows the “Fortnite Battle Royale” event chosen. At the time of this writing (27-Mar-2022), this event has no recorded results (15-Apr-2022) and consequently, no competition data has been entered into the database.

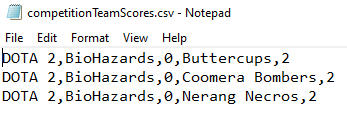


There are two (2) buttons at the bottom of this tab panel – both of which export CSV files of what is currently displayed in the 3 tables.

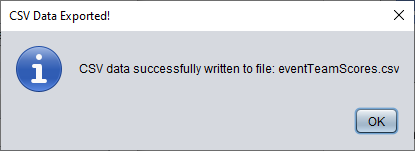
If the EXPORT COMPETITION RESULTS AS CSV FILE button is clicked, and there is row data in the competition results table --- then a pop-up displays that the data was written to the external competitionTeamScores.csv file.

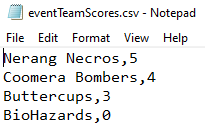


Opening the above example - competitionTeamScores.csv file (Event selected: DOTA 2 Battle Royale and Team selected: BioHazards) in Notepad:

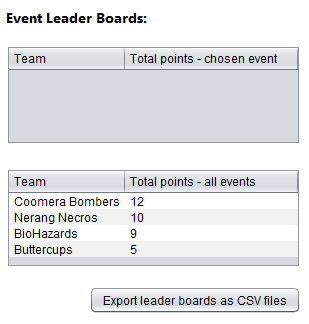


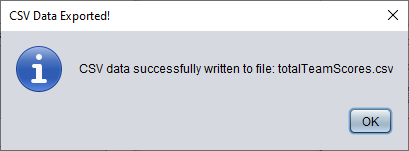
If the EXPORT LEADER BOARDS AS CSV FILE button is clicked, and there is row data in one of the leader board tables – then a pop-up displays that the data was written to the external csv files. The following example is the eventTeamScores.csv file which stores the 4 teams and their scores from the DOTA 2 Battle Royale event (selected from the events combo box).

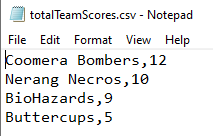




The following example is a save to the totalTeamScores.csv file with all teams and their respective total accumulated point score for all events (events combo box item selected was “All events”).







2. When the **ADD NEW COMPETITION RESULT** tab is selected, the following panel display (with UI controls) should resemble the screen shot example below:

All events stored in the event table are available in the event combo-box.

All games stored in the game table are available in the game combo-box.

All teams stored in the team table are available in the team 1 and team 2 combo-boxes.



**Important note** (when attempting to add a new competition result):

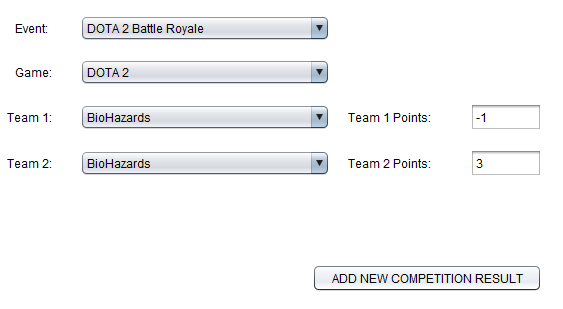
* Team 1 and Team 2 selections cannot be the same team
* Team 1 and Team 2 points can only be numeric values: 0, 1 or 2
* The sum of the two point scores must equal 2.

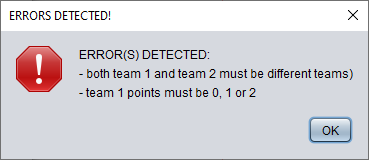
NOTE:

* The same team1/team2 combination playing the same game in the same event should not be allowed, as this represents an existing record in the database (competition table). This is an optional validation for the project.

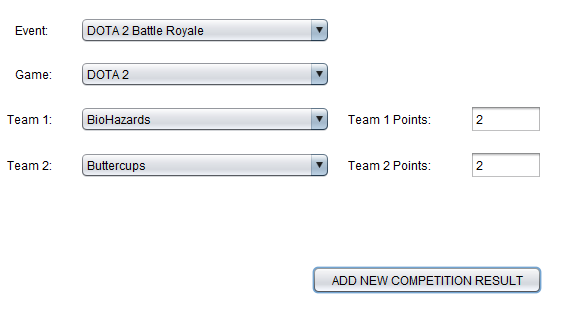
When the ADD NEW COMPETITION RESULT button is clicked, the following are potential error messages from the pop-up windows:

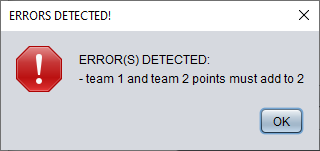
This example shows the two teams with the same name and the points are not 0, 1 or 2 (despite them adding to a total of 2).





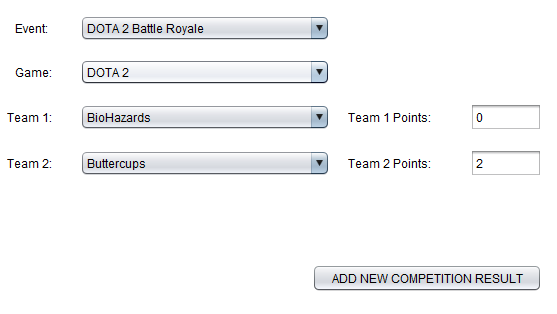
This example shows the two different teams with a total of points being more than 2.

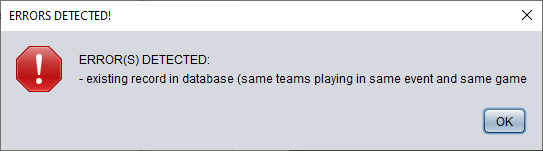




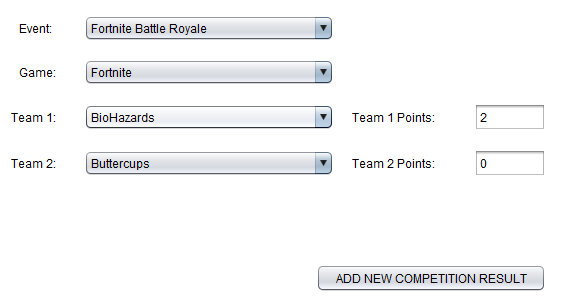
This is an optional validation exercise:

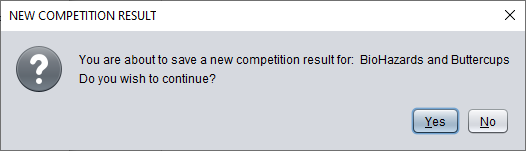
The same team1/team2 combination playing the same game in the same event should not be allowed, as this represents an existing record in the database.





The following screen shot example shows that all selected and input data is valid for the new competition entry. When the ADD NEW COMPETITION RESULT button is clicked, a pop-up displays a prompt message asking the user to click the YES button to proceed or NO to cancel.

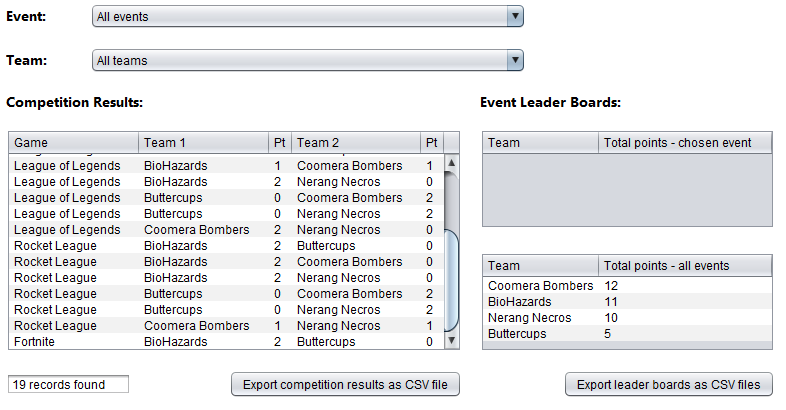




When the NO button is clicked, the action is cancelled. When the YES button is clicked, the program writes the record to the database (competition table).

When a new competition is added to the database (competition table), the application reflects this in the competition results table (EVENT COMPETITION RESULTS tab) and updates the All-Events Leader Board table.

In this example, the BioHazards team gained 2 points for their win over the Buttercups team in the Fortnite game (Fortnite Battle Royale event). The 2 points is added to their previous accumulated total of 9 points (for all events), now making a total of 11 points. This now places the team in second place in the overall leader board for all events.



3. When the **ADD NEW TEAM** tab is selected, the following panel display (with UI controls) should resemble the screen shot example below:

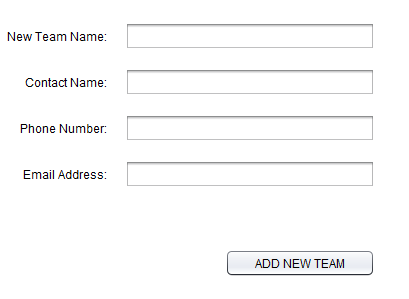


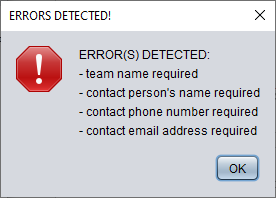
**Important note** (when attempting to add a new team):

* The new team name should NOT ALREADY EXIST in the database (this is a primary key in the team table)

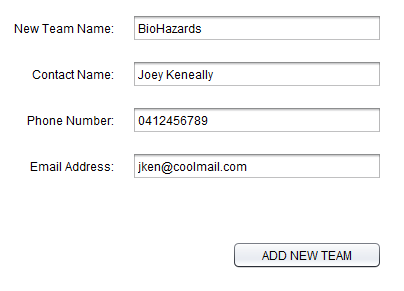
When the ADD NEW TEAM button is clicked, the following are potential error messages from the pop-up windows:

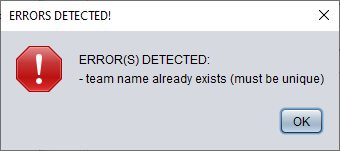
This example shows no input into the fields.



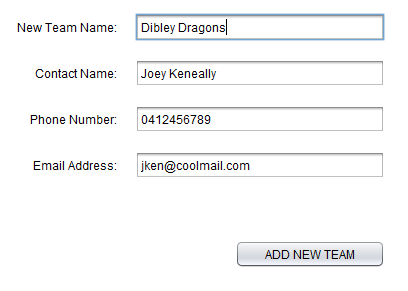


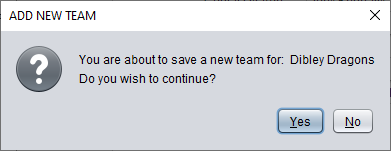
This example shows input into all fields, yet the new team name already exists in the database.





The following screen shot example contains valid data entries for a new team. When the ADD NEW TEAM button is clicked, a pop-up displays a prompt message asking the user to click the YES button to proceed or NO to cancel.

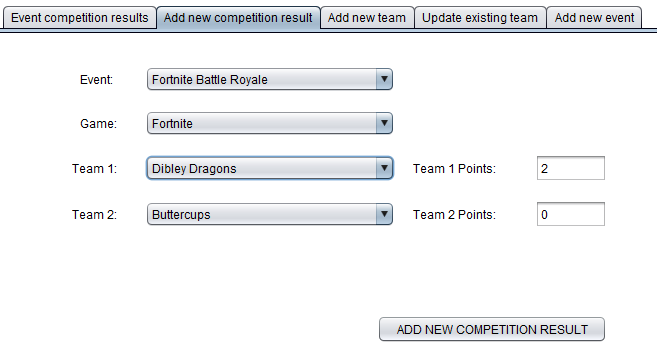




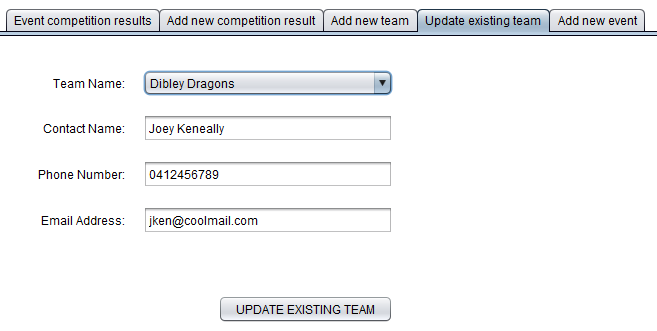
When the NO button is clicked, the action is cancelled. When the YES button is clicked, the program writes the record to the database (team table).

Updates for JComboBoxes to set up in the other tab panels when adding a new team:

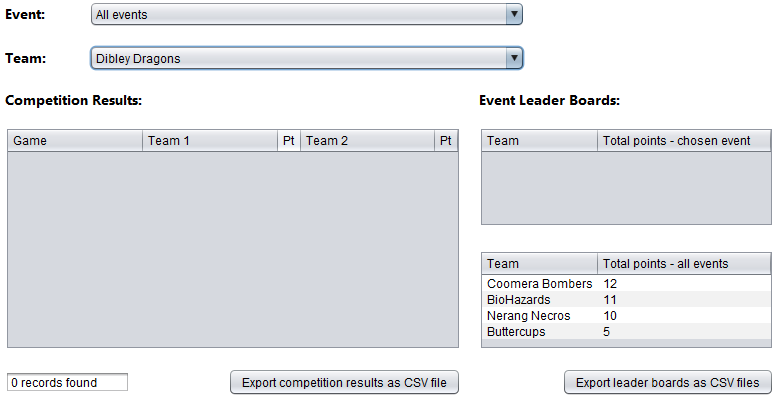
When adding a new team, the two (2) combo-boxes that list the team 1 and team 2 names in the ADD NEW COMPETITION RESULT tab panel must include the new team name.



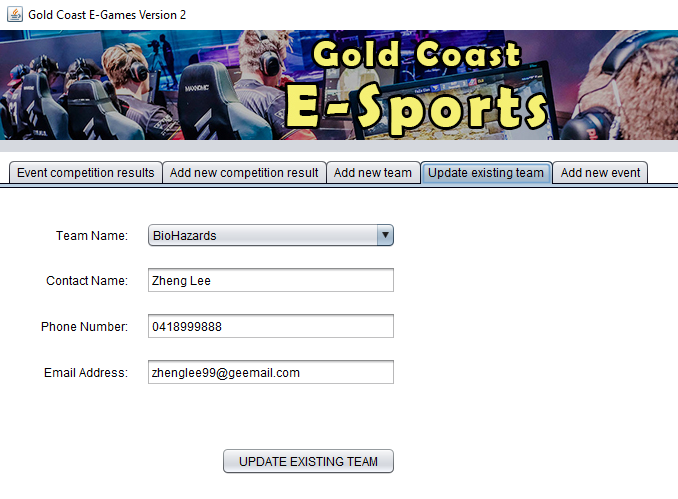
Similarly, the new team name must also be accessible in the combo-box in the UPDATE EXISTING TEAM tab panel.



Finally, the team combo-box in the EVENT COMPETITION RESULTS tab panel must also contain the new team name. In the example below, it is selected and shows no competition results (being a newly added team).



4. When the **UPDATE EXISTING TEAM** tab is selected, the following panel display (with UI controls) should resemble the screen shot example below. The details of the first (alphabetically sorted) team should appear in the combo box and text field controls.

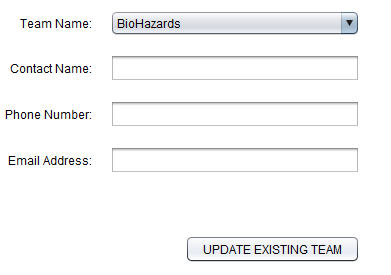


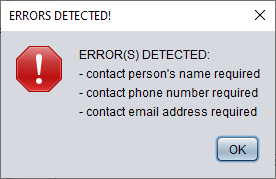
**Important note** (when attempting to update an existing team):

* The program does not allow changes to the team name (this would require a new team to be added)
* Changes can only be made to the contact name, phone number and email address

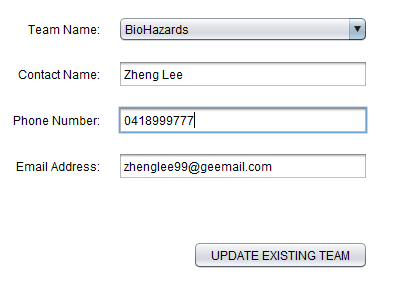
When the UPDATE EXISTING TEAM button is clicked, the following are potential error messages from the pop-up windows:

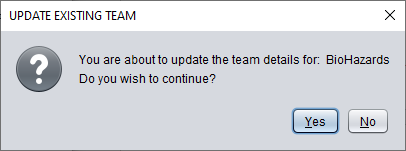
This example shows no input into the three (3) fields.





The following screen shot example contains valid data selection and entries for the update to an existing team. When the UPDATE EXISTING TEAM button is clicked, a pop-up displays a prompt message asking the user to click the YES button to proceed or NO to cancel.

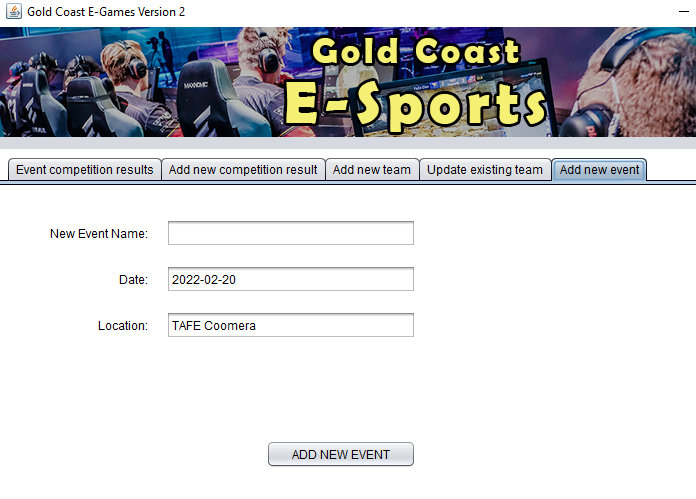




When the NO button is clicked, the action is cancelled. When the YES button is clicked, the program writes the record to the database (team table).

There are no required updates to any control in other tab panels.

5. When the **ADD NEW EVENT** tab is selected, the following panel display (with UI controls) should resemble the screen shot example below. The current date should be displayed in the format required (e.g. “2022-02-20” yyyy-mm-dd). A default location string can be used – for example: “TAFE Coomera”.



**Important note** (when attempting to add a new event):

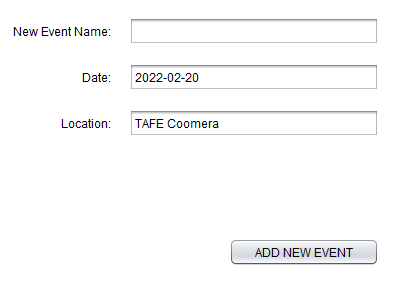
* The program requires a unique event name (and not one that already exists in the database – event table). The event name is the primary key of the event table.
* The program requires the date input to be formatted to: yyyy-mm-dd (10 characters in total)
* The program requires a location

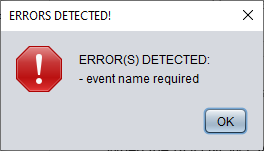
Optional:

* The input date should be validated to ensure that month values are between 1 and 12
* Also, date numeric values are appropriate to the month entered – for example: date values between 1 and 29 are only appropriate for February (02).

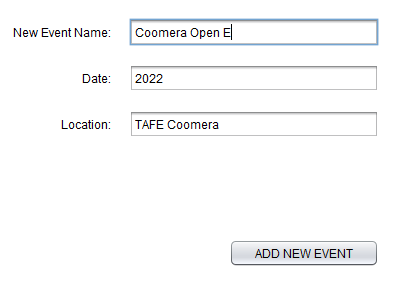
When the ADD NEW EVENT button is clicked, the following are potential error messages from the pop-up windows:

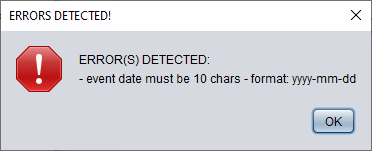
This example shows no input for the new event name field.



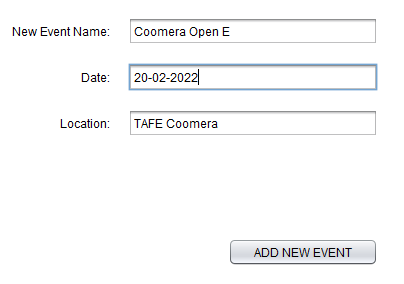


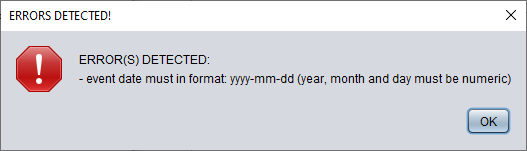
This example shows an input date value less than 10 characters in length and not formatted to: yyyy-mm-dd.



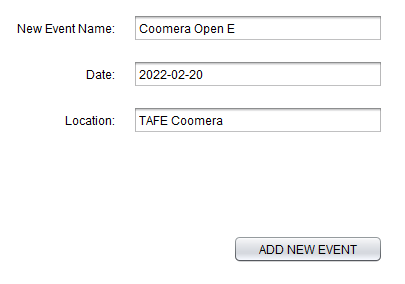


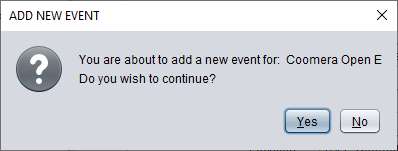
This example shows an improperly formatted data for the new event, despite it being 10 characters in length.





The following screen shot example contains valid input data for the new event record to be added. When the ADD NEW EVENT button is clicked, a pop-up displays a prompt message asking the user to click the YES button to proceed or NO to cancel.

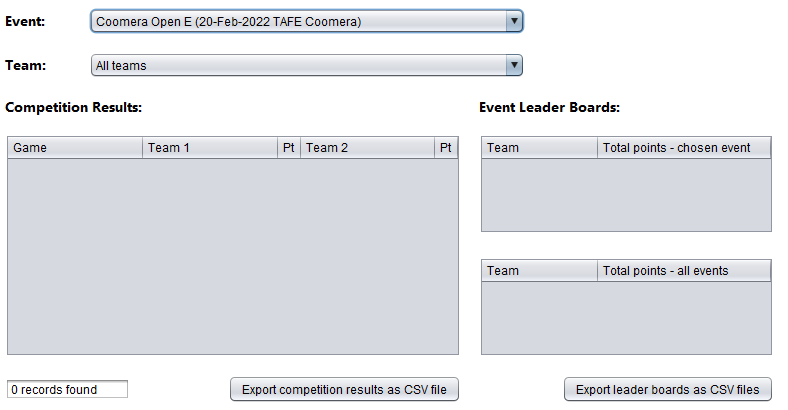




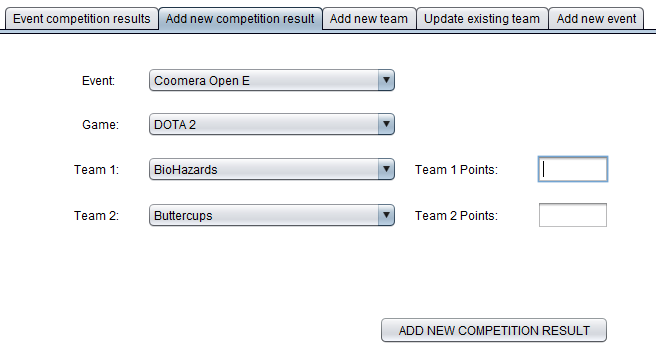
When the NO button is clicked, the action is cancelled. When the YES button is clicked, the program writes the record to the database (team table).

Updates to set up in the other tab panels when adding a new event:

When adding a new event, the event combo-box in the EVENT COMPETITION RESULTS tab panel must include the new event with bracketed date and location – refer screen shot example below).   
Note: In this example, there are no competition results or leader board data displayed for the new event.



The new event should be displayed in the event combo-box of the ADD NEW COMPETITION RESULT tab panel.

  
 **- End of File -**