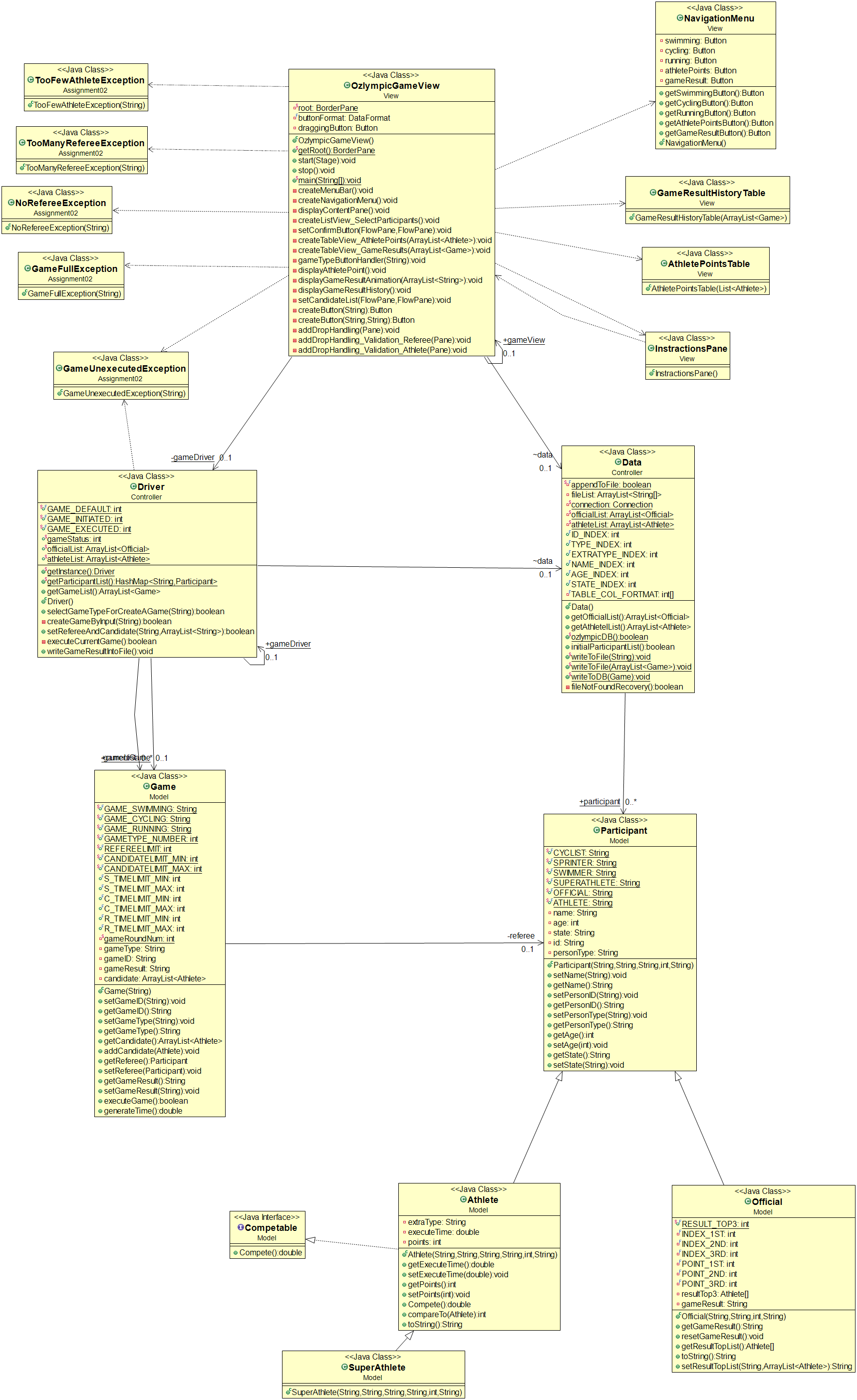
Assignment 2

Class Diagram and Description

By Arion Barzoucas-Evans s3650046

Ya-Huai Lai (Loso) s3579161

# Class Diagram



# Description

Several changes were made to the assignment 1 design in order to accommodate for the new data and user interface requirements. Specifically, the Data class was added containing methods for the following functions: a) to connect to a database and create the tables necessary for the application (participants and results), b) to read a text file containing a list of participants in case of database failure, c) to record result data to both the gameResults.txt file and the results table of the embedded database, and finally d) to provide a backup participant list in case both the database and text fail. Also, multiple customised exception classes were added to prevent incorrect use of the application. These include GameFullException, GameUnexecutedException, and TooManyRefereeException classes. What’s more, classes including OzlympicGameView, NavigationMenu, InstractionsPane, AthletePointsTable, and GameResultHistoryTable were added to create the graphic user interface. Finally, the Driver class size was significantly reduced by creating the Data class, the Swimming, Running, and Cycling classes were combined into the Game class, and the Swimmer, Cyclist, and Sprinter classes were removed from the hierarchy and incorporated in the Athlete class.

As depicted by the class diagram above, the Driver class is still the game controller providing the necessary methods triggered by the GUI events. The Data class is used by the Driver class to initiate the necessary information for the games and by the OzlympicGameView to record result information to the database and file. The customised exception classes interact with the OzlympicGameView and Driver classes when necessary to handle erroneous situations. Finally, the OzlympicGameView class uses the NavigationMenu, InstractionsPane, AthletePointsTable, and GameResultHistoryTable classes to create the corresponding GUI items.

The moment the application is started, the data will be read into arraylists either from the database or the file or the backup system. All user interaction will occur through the GUI entirely with using mouse functions (click, drag and drop). According to the user’s choices, the appropriate data (athletes, officials) is extracted from the corresponding arraylist where the information is stored in the program.

# Team Experience Feedback

***Arion***: Loso was did amazing work for this assignment as well. She’s very helpful and easy to reach making for excellent communication and cooperation. Once again, I have learnt a lot from her experience and coding style as well as her design ideas. This was all around an excellent experience which will undoubtedly come in useful for future studies and work.

***Loso*** : As a team, I believe that we did a good job; even though the features of the Ozlympic application still need to improve to meet the commercial standard, I am satisfied working with Arion in term of the ability of implementation, willing to communicate each other, and the skill of documentation . This is a nice team work experience that I have had.