***REPORT***

***BSc Computer Science***

A diagram of a cube with a white and blue box

Description automatically generated with medium confidence

**Project Title:** Part 1 - Analysis, Architecture Design and Detailed Design

**Author Name:** Dhyan Nilesh Patel

**Student ID:** 21008521

**Module Leader:** Dr John Kanyaru

**Use Case Diagram**

**A diagram of a company

Description automatically generated**

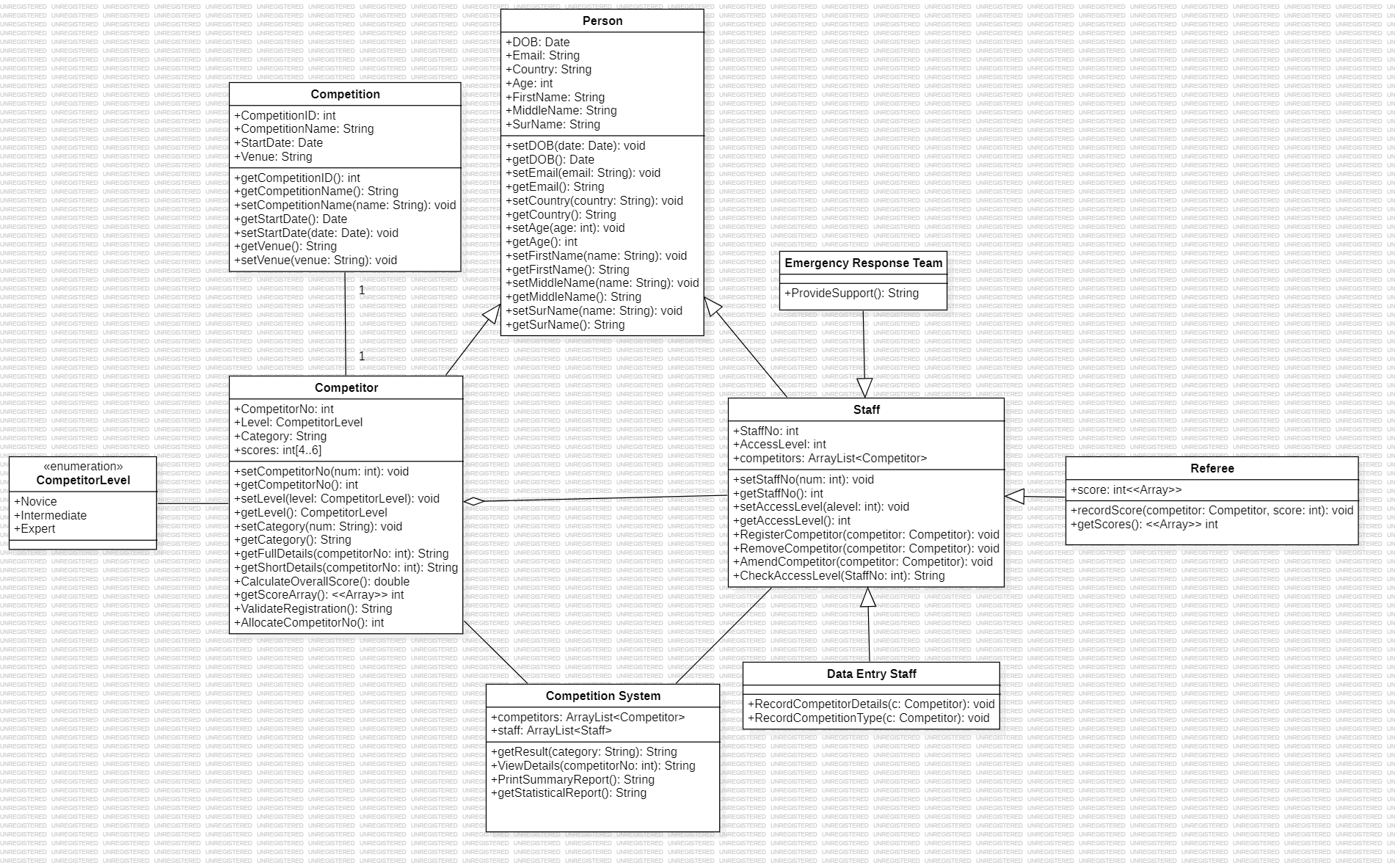
In the design of the Competition Management System, the use case diagram serves as a visual representation of the system's functionality, depicting the various actors, their interactions, and the relationships among different use cases. The primary actors in this system are competitors, staff, data entry staff, referees, fans, and the emergency response team. **Competitors**, as individuals participating in the competition, engage in various use cases such as registration, allocation of competitor numbers, validation of registration, searching competitors using competitor numbers, viewing details including scores and overall scores, printing summary reports, and recording scores, which involves calculating overall scores. **Staff**, the administrative personnel responsible for managing the competition, participate in use cases like registering competitors, retrieving competition results for a specific category, removing competitors from the competition, amending competitor details, and searching for competitor details. **Data Entry Staff**, tasked with entering and managing competition-related data, engage in use cases such as searching for competitor details, recording competition details, and recording specific competitor details. **Referees**, responsible for assigning scores to competitors, participate in the use case of giving out scores based on competitors' performance. **Fans**, who are spectators observing the competition proceedings, engage in use cases such as watching proceedings live and following proceedings on the screen. The **Emergency Response Team** ensures the safety and well-being of the audience, staff, and competitors by providing support in case of emergencies.

The relationships among use cases are denoted by two primary types of associations: <<extends>> and <<include>>. The <<extends>> relationships include scenarios where certain use cases extend others. For example, the registration of competitors extends the validation of registration, and searching for a competitor using a competitor number extends the options to print a summary report or record scores.

On the other hand, <<include>> relationships signify that one use case includes another as an integral part. For instance, validating registration includes allocating competitor numbers, and searching for a competitor using a competitor number includes viewing details.

In conclusion, the use case diagram effectively illustrates the dynamic interactions between various actors and their respective functionalities within the Competition Management System. The relationships, denoted by <<extends>> and <<include>>, provide a comprehensive overview of the system's workflow and user interactions, showcasing the intricate details of the competition management process.

Top of Form

**UML Class Diagram**

The UML class diagram for the Competition Management System provides a comprehensive overview of the system's structure, including classes, attributes, methods, and the relationships between them.

The **Person class** serves as a foundational class with attributes such as FirstName, MiddleName, SurName, DOB, Email, Country, and Age, along with standard get and set methods. The **Competitor class** inherits from the Person class and includes attributes like CompetitorNo, Level, Category, and an array of scores. It features various methods, including get and set methods, functions to retrieve competitor details, calculate overall scores, get score arrays, validate registration, and allocate competitor numbers. The **Competition class** includes attributes such as CompetitionID, CompetitionName, StartDate, and Venue, with standard get and set methods. The **Staff class**, inheriting from the Person class, incorporates attributes like StaffNo, AccessLevel, and an ArrayList of Competitor objects. Its methods include registration, removal, and amendment of competitors, as well as access level verification. The **Data Entry class**, a subclass of Staff, provides methods for recording competitor details and competition types. The **Referee class**, also a subclass of Staff, has attributes for scores and methods to record scores and retrieve them. The **Emergency Response Team class**, inheriting from Staff, offers a method to provide support in emergency situations. The **Competitor Level** Enumeration class defines attributes for Novice, Intermediate, and Expert levels. The **Competition System class** includes attributes for ArrayList of Competitor and Staff objects, along with methods for obtaining results, viewing details, printing summary reports, and generating statistical reports.

In terms of class relationships, Competitor and Staff classes inherit from the Person class, establishing an inheritance relationship. The Staff class exhibits one-to-many association relationship with the Competitor class, indicating that a Staff object may be composed of one or more Competitor. The Competition System class is associated with both Competitor and Staff classes, suggesting a connection between the system and competitors and staff members. The Competitor class has a many-to-many association with the Competition class, allowing a competitor to participate in multiple competitions, and a competition to have multiple competitors. The Competitor Level class is an enumeration associated with the Competitor class, representing the different levels a competitor can belong to. Lastly, the Emergency Response Team, Referee, and Data Entry Staff classes inherit attributes and methods from the Staff class, establishing an inheritance relationship that signifies a hierarchy where specific staff roles inherit general staff attributes and functionalities.

In the Presentation Tier, the Staff GUI subsystem relies on the Competition System class, facilitating staff interactions for tasks such as view details, print summary report, and get result. Display Screens, on the other hand, depend on the Handle Scoring Subsystem to present real-time scores to the audience. The Competitor GUI subsystem is associated with the Competition System class, allowing competitors to interact with the system for registration and participation.

Moving to the Application (Logic) Tier, the Competition Management subsystem is anchored by the Competition class, serving as the core for managing and organizing competitions. It also interacts with the Data Entry Staff class for recording competition details. The Check Access Level subsystem relies on the Staff class to determine the access levels of staff members within the system. Competitor Registration involves the Competition class, handling the registration process for competitors. The Handle Scoring Subsystem comprises the Referee class, responsible for recording and managing scores during competitions. Update Competitor Details involves both the Staff class and Competitor class, facilitating the modification of competitor information. The Remove Competitor subsystem includes the Staff Class and Competitor class, enabling the removal of competitors based on specific criteria. Validate Registration is associated with the Competitor class, ensuring the correctness of competitor registration details.

In the Data Tier, the Competitor DB subsystem primarily involves the Competitor class, serving as a data repository for competitor-related information. Staff DB involves the Staff class, managing data relevant to staff members and their access levels. The Competition DB subsystem is associated with the Competition class, managing data related to competitions and providing support to the Competition Management subsystem.

**3 Tier Architecture Model**

**A screenshot of a computer flowchart

Description automatically generated**

The 3-tier architecture model for the Competition Management System is designed to organize the system into three distinct layers namely Presentation Tier, Application (Logic) Tier, and Data Tier.

In the Presentation Tier, various user interfaces facilitate interaction with the system. The Web Interface and Mobile Interface serve as platforms for staff and other users to access and engage with the competition-related functionalities. The Staff GUI provides a graphical interface for staff members, allowing them to perform tasks such as competitor registration on arrival, updating details, and removing competitors. Display Screens showcase real-time scoring information to the audience, enhancing their experience. The Competitor GUI offers competitors a user-friendly interface for registration and participation.

The Application Tier houses the logic and business processes of the system. The Competition Management subsystem handles the core functionalities related to competitions, ensuring efficient organization and management. The Check Access Level subsystem is responsible for verifying the access levels of staff members, controlling their permissions within the system. Competitor Registration allows competitors to register for competitions, validating their information to ensure accurate and complete registration. The Handle Scoring Subsystem manages the scoring process, a crucial aspect of competition. Update Competitor Details enables officials to modify competitor information, and Remove Competitor allows the removal of competitors who fail to meet rules or do not show up. Validate Registration ensures the correctness of competitor registration details.

The Data Tier involves the storage and management of data through various databases. Competitor DB stores detailed information about competitors, supporting tasks like registration, updating, and removal. Staff DB holds data relevant to staff members, assisting in verifying access levels. Competition DB manages competition-related details, supporting the Competition Management subsystem.

Several dependencies exist between subsystems. For instance, Staff GUI relies on the Web Interface and Mobile Interface to provide a comprehensive user interface, catering to both web and mobile users. Additionally, it depends on the Check Access Level subsystem to verify the access levels of staff members, ensuring secure system interactions. The Competitor Registration subsystem is crucial for staff GUI, enabling the registration of competitors upon arrival. In case competitors fail to show up or violate competition rules, Staff GUI depends on the Remove Competitor subsystem for their exclusion. Furthermore, the system relies on the Update Competitor Details subsystem to modify competitor information when necessary. Competitor GUI, catering to competitor interactions, depends on Competitor Registration for the initial registration process. Additionally, it relies on the Competition Management subsystem to ascertain the type of competition a competitor is participating in. Display Screen, responsible for presenting real-time scores to the audience, is dependent on the Handle Scoring subsystem. This ensures that accurate and timely scoring information is displayed for spectators. Handle Scoring is dependent on Competitor DB to fetch the scores for each competitor, ensuring accurate and real-time scoring information. Check Access Level, essential for verifying staff access levels, relies on Staff DB to obtain the necessary access level information. Competitor Registration depends on Competitor DB to access competitor data, forming a reciprocal relationship. It also relies on Validate Registration to ensure the accuracy and validity of competitor registrations. Competitor DB, serving as a central data repository, is intricately connected with various subsystems. It relies on Update Competitor Details to update the database with any changes made by officials. Similarly, it depends on Remove Competitor to eliminate competitor details when required. This reciprocal dependency ensures that any changes made by officials to remove or modify competitor details are accurately reflected in the database. Lastly, Competition Management is dependent on Competition DB to obtain competition-related details, and, in turn, Competition DB relies on Competition Management for cohesive management of competition-related data. This interdependence ensures that competition details are consistently and accurately managed across the system.