

UML DIAGRAM - Implementations and functioning

Since the first UML diagram Dominic and I, we changed the focus of the project completely because we had one member leaving us and the UML diagram was completely wrong. So here is the explanation of every class that is in the diagram. You can view it accessing the Stage 3 folder and you will find out “UML_Diagram.png”. It is a picture because it was really large and we could not make it as a pdf.

This are main changes that we have implemented:

Main Class: Represents the main entry point of the program. It manages instances of ClientFile, MembershipFile, CompetitionFile, and StaffFile, and contains methods for different menus (clientMenu, membershipMenu, competitionMenu, staffMenu) as well as the main method.

Client Class: Represents a client with attributes like name, email, phone, and membership. It has methods to get and set these attributes.

Membership Class: Represents different membership tiers with attributes like name and description. It also has methods to get and set these attributes.

Competition Class: Represents a competition with attributes like name and description. It has methods to get and set these attributes.

Staff Class: Represents a staff member with attributes like name, ID, and position. It inherits from the Person class and has methods to display staff information.

ClientFile Class: Manages a list of clients and provides methods to add, remove, and retrieve clients. It also has a method to display all clients.

MembershipFile Class: Manages a list of memberships and provides methods to get memberships by name and initialize memberships.

CompetitionFile Class: Manages a list of competitions and provides methods to add, remove, and edit competitions. It also has a method to get all competitions.

StaffFile Class: Manages a list of staff members and provides methods to add, remove, and retrieve staff members.

We also included a couple of relationships:

Main has relationships with ClientFile, MembershipFile, CompetitionFile, and StaffFile, indicating that Main uses instances of these classes.

ClientFile contains instances of Client, shown by the composition relationship.

MembershipFile contains instances of Membership, also shown by the composition relationship.

CompetitionFile contains instances of Competition, shown by the composition relationship.

StaffFile contains instances of Staff, shown by the composition relationship.

Client has a composition relationship with Membership, indicating that a Client object contains a Membership object.

Staff inherits from Person, shown by the inheritance relationship (Staff --> Person).