Youth Hockey Tournament Management System

Ashok Enukonda

Date Finished: 21-03-2025

Group Number: 19

Table of Contents

1. Introduction

The objective of this project is to establish a management system for a youth hockey tournament. The system allows for the creation, administration, and storage of information regarding the youth hockey teams that are participating in the tournament. Users have the ability to join new teams, view their details, update existing teams, delete teams, and manage their participation status, including payment and cancellation. The program facilitates the storage and retrieval of team data from files.

The primary objectives of the system include:

- Managing essential team information, including the team's name, category, coach, age group, and squad size.
- Calculating and tracking the status of fee payments.
- Storing and retrieving team information from files.
- Providing an intuitive user interface for interacting with the system.

2. Description of the Solution

The solution consists of two main classes: **Team** and **User Interface.**

2.1 Team Class

The Team class represents a youth hockey team participating in the tournament. Each team has several attributes, such as:

- **ID**: A unique identifier for each team, automatically generated upon team creation.
- **Name**: The team’s name, validated to ensure it’s not empty and is formatted correctly.
- **Type**: The team’s type (either 'boys' or 'girls'), is validated to ensure only valid types are used.
- **Fee Paid**: A boolean value indicating whether the team has paid the participation fee.
- **Fee**: The fixed participation fee amount, is set to 500.
- **Coach**: The name of the coach, validated for correctness.
- **Age Group**: The age group of the players (between 5 and 18 years), validated to ensure the age falls within this range.

- **Team Size**: The size of the team (between 5 and 20 players), validated to ensure the size is reasonable.
- **Cancellation Date**: The date when the team cancels its participation in the tournament, if applicable.

The class provides methods for:

- **Team creation**: Instantiating a team with all the required attributes.
- **Validating data**: Ensuring the correctness of data inputs, including name, type, age group, and team size.
- **Cancelling participation**: Handling cases when a team decides to cancel its participation in the tournament.
- **Getting details**: A method to retrieve a dictionary of the team's details.
- **Converting to/from dictionaries**: Methods for converting a team to a dictionary and creating a team from a dictionary.

2.1.1 Class Structure

- **Attributes:**
  - **_id:** A unique identifier for each team.
  - **_date:** The date the team was created.
  - **name:** The team's name.
  - **type:** The type of team, either "boys" or "girls."
  - **fee_paid:** A boolean indicating if the participation fee is paid.
  - **fee:** The fixed participation fee (500).
  - **coach:** The coach's name.
  - **age_group:** The team's age group (between 5 and 18 years).
  - **team_size:** The number of players in the team (between 5 and 20).
  - **cancellation_date:** The date when the team cancels its participation (if applicable).
- **Methods:**
  - **init():** Initializes a new team with specified attributes.

- **_validate_name():** Validates and formats the team name.
- **_validate_type():** Ensures the team type is either "boys" or "girls."
- **_validate_age_group():** Ensures the age group is between 5 and 18 years.
- **_validate_team_size():** Ensures the team size is between 5 and 20 players.
- **cancel_participation():** Cancels the team's participation and stores the cancellation date.
- **get_details():** Returns a dictionary of all the team details.
- **from_dict():** Creates a Team instance from a dictionary.
- **to_dict():** Converts the Team instance to a dictionary.

2.2 UserInterface Class

The User Interface class controls how the user interacts with the program. It provides several choices using a text-based menu, therefore allowing the user to:

- Create new teams.
- View team details by ID.
- List all teams or filter by team type (boys/girls).
- Update existing team information.
- Delete teams.
- Save teams to a file.
- Retrieve teams from a file.
- Cancel a team’s participation.
- Show the total number of teams.
- Show the percentage of teams that have paid their fee.

2.2.1 Class Structure

- **Attributes:**
  - **teams:** A list that stores instances of the Team class.
- **Methods:**
  - **display_menu():** Displays the main menu options to the user.

- **create_team():** Prompts the user to create a new team.
- **view_team():** Allows the user to view a team's details by its ID.
- **list_teams():** Lists all registered teams.
- **list_boys_teams():** Lists all boys' teams.
- **list_girls_teams():** Lists all girls' teams.
- **update_team():** Allows the user to update an existing team's details.
- **delete_team():** Prompts the user to delete a team.
- **save_to_file():** Saves the list of teams to a file.
- **load_from_file():** Loads teams from a file into the list.
- **cancel_team_participation():** Allows the user to cancel a team's participation.
- **show_total_teams():** Displays the total number of teams.
- **show_fee_paid_percentage():** Shows the percentage of teams that have paid their fee.
- **_find_team_by_id():** Searches for a team by its ID.
- **run():** Runs the program, displaying the menu and handling user input.

3. User Guide

To use the system, follow the steps outlined below:

3.1 Running the Program

- **Start the program:** To run the program, execute the **main.py** file. This file will initiate the program and display the main menu with multiple options for interaction.

3.2 Main Menu options

- The menu includes the following choices:
    - **Create a new team:** Add a new team to the system by entering the team’s details (name, type, fee status, coach, age group, team size).
    - **View team details by ID:** View the details of a team by entering the team’s unique ID.
    - **List all teams:** Displays all teams in the system.
    - **List all boy’s teams:** Displays all teams categorized as 'boys'.

- **List all girl’s teams:** Displays all teams categorized as 'girls'.
- **Update an existing team:** Modify the details of an existing team by selecting the team ID.
- **Delete a team:** Remove a team from the system by confirming its deletion.
- **Save teams to file:** Save all the team details to a file for future retrieval.
- **Retrieve the teams from the file:** Load previously saved teams from a file.
- **Cancel team participation:** Mark a team as cancelled by entering the cancellation date.
- **Show the total number of teams:** Display the total number of teams in the system.
- **Show the percentage of teams that have paid their fee:** Show the percentage of teams that have completed the payment.

3.3 Files in the System

- **team.py:**
  - This file defines the Team class, which is central to the system. The Team class handles all the core functionalities of managing teams, such as creating a new team, validating team data, updating, deleting, marking cancelled teams, and calculating the percentage of teams that have paid their fees.
  - The Team class ensures that all team-related data is correctly processed, stored, and validated.
- **user_interface.py:**
  - This file contains the UserInterface class, which is responsible for managing the interaction between the user and the program. The UserInterface class handles the display of the main menu and prompts the user for input. It also communicates with the Team class to perform actions such as adding, deleting, updating, and listing teams.
  - The UserInterface class acts as the intermediary between the user and the core functionality of the system.
- **main.py:**
  - The program's entry point is main.py. It imports the necessary classes from **user_interface.py** and **team.py** then starts the program running the User

Interface class. The **main.py** file is responsible for initiating the program flow and displaying the menu for user interaction.

3.4 Running the System

- **Execute main.py:** To start the system, run the main.py file. This will invoke the UserInterface class and display the main menu.
- **Choose an option:** Enter the number corresponding to the desired action.
- **Follow prompts:** Based on the action chosen, the program will prompt for additional information (e.g., team name, coach, etc.).
- **Exit:** Choose the option to exit the program when finished.

Link to the recorded Video: https://youtu.be/WGTw_F_RGdo