

This script for managing a sports club. It uses a command-line interface to interact with the user and perform various operations. Here's a breakdown of its functionality:

1-Dynamic Declarations: The script starts by declaring several dynamic predicates (stadium, player, team, occupancy) which will be used to store and manipulate data about stadiums, players, teams, and stadium occupancy.

2-Start and Menu: The start predicate displays a welcome message and calls the menu predicate. The menu predicate displays a list of options for the user to choose from.

3-Reading User Input: The read predicate reads the user's choice and calls the appropriate predicate based on the input.

4-Option 1 - Play a Football Game: If the user chooses option 1, the showstadiums predicate is called. This predicate displays information about available stadiums and allows the user to choose one. The user is then asked to choose their team and the opposing team, and to pick a time for the game. The occupancy predicate is used to check if the chosen time slot is available.

5-Option 2 - Create a Team: If the user chooses option 2, the create\_team predicate is called. This predicate allows the user to create a new team by choosing the size of the team (8 or 10 players) and entering a name for the team. The player\_id8 and player\_id10 predicates are used to add players to the team.

6-Option 3 - Inscription to the Gym: If the user chooses option 3, the inscription\_for\_gym predicate is called. This predicate allows a player to subscribe to the gym by entering their ID and choosing an offer.

7-Data: The script includes some predefined data about stadiums, players, teams, and stadium occupancy.

8-End: The end predicate is used to terminate the program.

This script uses a lot of Prolog's features, such as dynamic predicates, backtracking, and assert/retract operations. It's a good example of how Prolog can be used to create interactive applications. However, it's worth noting that the code could be improved by adding error checking and handling, and by modularizing the code into separate files or modules.