

## Algorithms and programming

Requirements analysis and specification By: Miguel Perez Ojeda

### Problem identification and requirements analysis Case Study: soccer tournament

<b>Customer</b>	<i>University professor</i>
<b>User</b>	<i>Soccer teams which can participate in the tournament, tournament referees, sports fans and the community.</i>
<b>Requerimientos funcionales</b>	<p>RF1: Register a team. RF2: Register a player. RF3: Register a referee. RF4: Preload the necessary information to start a tournament: teams, players, and referees. RF5: Calculate and display the draw (matches) for the group stage and the respective dates of this stage. RF6: Assign the referee team to a match according to their availability (nationality). RF7: Record the score of a match. RF8: Register player who scores goal and player who assists. RF9: Register cards for the players. RF10: Consult the group stage information at any time (position table of all groups). RF11: Calculate and display the top scorer of the tournament. RF12: Calculate and show the team with Fair Play. RF13: Calculate and show the efficiency of a team (number of games won compared to games played). RF14: Calculate and show a player's efficiency (number of goals compared to games played).</p>

	RF15: Calculate and display the card index of a central referee (number of cards with respect to refereed matches).
<b>Problem Context</b>	<i>The system to be developed will manage an international football tournament with 8 teams, inspired by the Copa América. It must allow the registration of teams and players, generate the tournament fixture, and record scores, goals, cards, and referee assignments. Teams must be registered in the platform beforehand. Fans will have access to statistics, team and player lists, and prize details. The system must be web-based, accessible from various devices, and ensure high availability and quick response times.</i>
<b>Non-Functional Requirements</b>	<ul style="list-style-type: none"> <li>• Response Time: The system must respond to user requests in less than 2 seconds.</li> <li>• Scalability: It must be able to handle an increase in the number of users and data without significantly affecting performance.</li> <li>• Platform Compatibility: The system must be compatible with different operating systems (Windows, Linux, macOS) and modern web browsers, though certain browsers may be recommended to users for optimal performance.</li> <li>• Code Readability: The code must be readable and easy to understand for any developer.</li> <li>• Extensibility: The system must be easy to extend to include new functionalities in the future.</li> <li>• Availability: The system must have 99.9% availability, ensuring it is operational and accessible almost all the time.</li> </ul>

	<ul style="list-style-type: none"> <li>• <b>Backup and Recovery:</b> It must have mechanisms for data backup and recovery to protect information in case of system failures or data loss.</li> <li>• <b>Security:</b> The system must ensure the protection of team, player, referee, and match result data against unauthorized access and information loss.</li> <li>• <b>Availability and Maintenance:</b> This requirement establishes the need to guarantee high availability and system maintenance.</li> </ul>
<p style="text-align: center;"><b>Process Requirements</b></p>	<ul style="list-style-type: none"> <li>• The program must be developed in Java.</li> <li>• The program must be designed under the object-oriented programming (OOP) paradigm.</li> <li>• The most critical process requirements for this project focus on ensuring quality, collaboration, and flexibility. A development process that includes constant testing is essential to ensure the system works correctly, along with rigorous change management. Effective communication among all stakeholders and the adoption of agile methodologies are key to ensuring that the system adapts to the project's changing needs and meets the highest quality standards. Additionally, the user interface must be very user-friendly.</li> </ul>

Identifier and name	<i>RF1: Register a Team</i>		
Summary	<i>The program will carry out the registration of each team, where the user will be asked for information such as: team name, country of origin, list of 20 players, head coach's name, and assistant coach's name..</i>		
Inputs	<b>Input Name</b>	<b>Data Type</b>	<b>Condition / Valid Values</b>
	Team Name	String	This field cannot be empty.
	Country of Origin	String	Must be an existing and officially recognized country.
	List of 20 Players	String	Exactly 20 players must be entered.
	Head Coach's Name	String	The first letter of the name must be uppercase and follow the structure: first name, second name (optional), last name, and second last name.
Postcondition	The information of the team and its players has been successfully saved in the system.		

Outputs	Output Name	Data Type	Format
	Team Registration Confirmation	String	“Player [Player_1] has been successfully registered.”
	Error Message	String	“Invalid information, please verify that it meets the conditions.”

Identifier and name	<i>RF2: Register a Player</i>		
Summary	The program will carry out the registration of each team, where the user will be asked for information such as: team name, country of origin, list of 20 players, head coach's name, and assistant coach's name. Additionally, this function will include the player registration, which includes their number, name, country of origin, and position (goalkeeper, defender, midfielder, or forward). This function will primarily be used by the team coach.		
Inputs	Input Name	Data Type	Condition / Valid Values
	Name	String	This field cannot be empty.
	Country of Origin	String	Must be an existing and officially recognized country.

	Id	String	A number from 0 to 99
	Position	String	It can only be a goalkeeper, forward or midfielder.
Postcondition	The information of the team and its players has been successfully saved in the system.		
Outputs	<b>Output Name</b>	<b>Data Type</b>	<b>Format</b>
	Player Registration Confirmation	String	“Player [Player_Name] has been successfully registered.”

Identifier and name	<i>RF3: Register a Referee</i>		
Summary	The system will allow the storage of information about referees, requiring input of an ID, referee name, country of origin, and referee type (Central or Assistant). A total of 4 central referees and 8 assistant referees are required. This function will primarily be used by the tournament administrator. It is important to ensure that the referee does not belong to the same country as one of the teams they will officiate.		
Inputs	<b>Input Name</b>	<b>Data Type</b>	<b>Condition / Valid Values</b>
	Referee Name	String	The name must follow the structure: first name, second name (optional),

			last name, and second last name (optional). No numbers are allowed in this field.
	Referee ID	String	Must contain only numbers, with a minimum of 5 digits and a maximum of 15. Cannot contain periods.
	Country of Origin	String	Must be an existing and officially recognized country. No invented names or unrecognized countries are allowed.
	Authentication of Country of Origin	String	The referee must provide a document (visa or passport) verifying their country of origin to ensure the information is accurate.
	Referee Type	String	The value entered must be either “Central” or “Assistant.” The maximum number of each type cannot be exceeded (8 central referees and 16 assistant referees). If an attempt is made to register more, a warning message will be generated.
Postcondition	The referee information has been successfully saved in the system, and the total number of referees meets the established requirements.		
Outputs	Output Name	Data Type	Format

	Referee Registration Confirmation	String	“Referee [Referee_Name] has been successfully registered.”
	Error Message	String	“Invalid information, please verify that it meets the conditions.”

Identifier and name	<i>RF4: Preload Necessary Information to Start a Tournament: Teams, Players, and Referees</i>		
Summary	The system will preload essential information needed to start a tournament, including data on teams, players, and referees. This function will streamline the setup process, ensuring that all necessary participants are registered before the tournament begins.		
Postcondition	All necessary information for teams, players, and referees has been preloaded successfully, enabling a smooth start to the tournament.		
Outputs	<b>Output Name</b>	<b>Data Type</b>	<b>Format</b>
	Preloading Confirmation	String	“All necessary information has been preloaded successfully.”
	Error Message	String	“Invalid information, please verify that it meets the conditions.”



Identifier and name	RF5: Calculate and display the draw (matches) for the group stage and the respective dates of this stage.		
Summary	The system will automatically calculate and display the schedule for the group stage matches, assigning dates and alternating between groups.		
Postcondition	The match schedule for the group stage is generated, ensuring intergroup alternation and that the final matchday for each group is simultaneous.		
Outputs	<b>Output Name</b>	<b>Data Type</b>	<b>Format</b>
	Fixture (matches)	String	(Match Date, Team B, Team A)
	Match Dates	String	DD-MM-YYYY
	Group A and B Matches	String	PJ, G, E, P, GF, GC, DG, Pts
	Error Message	String	“Invalid information, please verify that it meets the conditions.”

Identifier and name	RF6: Assign the referee team to a match according to their availability (nationality).
Summary	The system will assign the referee team (1 central referee and 2 assistant referees) to each match, ensuring that none of the

	referees have the same nationality as the teams playing the match and that they are available for the scheduled date.		
Postcondition	The referee team (1 central, 2 assistants) is assigned to the match. No referee shares nationality with any team involved, and they are all available for the match.		
Outputs	<b>Output Name</b>	<b>Data Type</b>	<b>Format</b>
	Error Message	String	“Invalid information, please verify that it meets the conditions.”

Identifier and name	RF7: Record the score of a match.		
Summary	The system will allow the user to record the final score of a match, including the goals scored by each team and the players who scored. The system will update the match statistics, such as goals, assists, and points for the teams.		
Inputs	<b>Input Name</b>	<b>Data Type</b>	<b>Condition / Valid Values</b>
	matchIndex	String	The index or ID of the match to be updated.
	homeScore	String	Number of goals scored by the home team.
	awayScore	String	Number of goals scored by the away team.

Postcondition	The score and match statistics are successfully recorded. The points and goal statistics of the teams are updated accordingly.		
Outputs	<b>Output Name</b>	<b>Data Type</b>	<b>Format</b>
	Success Message	String	"The match result has been successfully recorded."
	Error Message	String	"Invalid information, please verify that it meets the conditions."

Identifier and name	RF8: Register player who scores goal and player who assists.		
Summary	The system will allow users to record individual player contributions to a match by registering the players who score goals and the players who assist them. This information will update the players' personal statistics and be reflected in the match summary.		
Inputs	<b>Input Name</b>	<b>Data Type</b>	<b>Condition / Valid Values</b>

	matchId	String	The unique identifier for the match in which the goal was scored.
	scorerId	String	The unique identifier of the player who scored the goal.
	assisterId	String	The unique identifier of the player who assisted the goal. May be null if there was no assist.
	goalMinute	Double	The minute in which the goal was scored.
Postcondition	<p>The scoring player's total goals in the tournament are updated.</p> <p>The assisting player's total assists in the tournament are updated (if applicable).</p>		
Outputs	<b>Output Name</b>	<b>Data Type</b>	<b>Format</b>
	SuccessMessage	String	Confirmation message: "The goal and assist have been successfully recorded."
	Error Message	String	"Invalid information, please verify that it meets the conditions."

Identifier and name	RF9: Register cards for the players.		
Summary	The system will allow users to record disciplinary actions taken during a match by registering the cards (yellow or red) issued to players. This information updates the players' statistics, the team's Fair Play standing, and the match summary.		
Inputs	<b>Input Name</b>	<b>Inputs</b>	<b>Condition / Valid Values</b>
	matchId	String	The unique identifier for the match where the card was issued.
	playerId	String	The unique identifier of the player who received the card.
	cardType	String	The type of card issued: "Yellow" or "Red".
	cardMinute	double	The minute of the match in which the card was issued. Must be within the match duration.
Postcondition	<p>The card is successfully recorded in the match summary.</p> <p>The player's statistics are updated to reflect the issued card.</p> <p>The team's Fair Play standing is updated accordingly.</p>		

Outputs	Output Name	Data Type	Format
	SuccessMessage	String	Confirmation message: "The card has been successfully recorded for the player."
	Error Message	String	"Invalid information, please verify that it meets the conditions."

Identifier and name	RF10: Consult the group stage information at any time (position table of all groups).		
Summary	The system will allow users to view the current standings of all groups during the group stage of the tournament. The position table will display detailed statistics for each team, including matches played, matches won, drawn, and lost, goals scored, goals conceded, goal difference, and points.		
Inputs	Input Name	Inputs	Condition / Valid Values
	groupId	String	The identifier for a specific group to filter the table. If omitted, all groups are shown.
Postcondition	The system displays the requested group standings with the latest data.		

	No data is modified during this operation.		
Outputs	<b>Output Name</b>	<b>Data Type</b>	<b>Format</b>
	GroupTable	String	<p>A tabular format with the following columns:</p> <p><b>TeamName:</b> String  <b>PJ:</b> Int (Matches Played)  <b>G:</b> Int (Matches Won)  <b>E:</b> Int (Matches Drawn)  <b>P:</b> Int (Matches Lost)  <b>GF:</b> Int (Goals Scored)  <b>GC:</b> Int (Goals Conceded)  <b>DG:</b> Int (Goal Difference = GF - GC)  <b>Pts:</b> Int (Points)  </p>
	SuccessMessage	String	"Group stage standings successfully retrieved."
	Error Message	String	"Invalid information, please verify that it meets the conditions."

Identifier and name	RF11: Calculate and display the top scorer of the tournament.
---------------------	---

Summary	The system will calculate and display the player with the highest number of goals scored during the tournament. If there is a tie in the number of goals, additional criteria such as assists will be used as tiebreakers.		
Postcondition	<p>The top scorer's information is successfully retrieved and displayed.</p> <p>No data is modified during this operation.</p>		
Outputs	<b>Output Name</b>	<b>Data Type</b>	<b>Format</b>
	TopScorerInfo	String	<p>Full name of the top scorer, e.g., "Player_1".</p> <p>Total number of goals scored by the player, e.g., "10".</p> <p>Total number of assists by the player, e.g., "5". (Used as a tiebreaker if needed).</p>
	SuccessMessage	String	"The top scorer has been successfully retrieved."
	Error Message	String	"Invalid information, please verify that it meets the conditions."



Identifier and name	RF12: Calculate and show the team with Fair Play.		
Summary	The system will calculate and display the team with the best Fair Play score, based on the number of yellow and red cards received throughout the tournament. Teams with fewer cards will rank higher. If there is a tie in the number of cards, alphabetical order of team names will be used as a tiebreaker.		
Postcondition	The team with the best Fair Play score is successfully identified and displayed.		
Outputs	<b>Output Name</b>	<b>Data Type</b>	<b>Format</b>
	FairPlayTeam Name	String	Name of the team with the best Fair Play score, e.g., "Team B". Total number of yellow cards received by the team, e.g., "3". Total number of red cards received by the team, e.g., "1".
	SuccessMessage	String	"The team with Fair Play has been successfully retrieved."
	Error Message	String	"Invalid information, please verify that it meets the conditions."

Identifier and name	RF13: Calculate and show the efficiency of a team (number of games won compared to games played).		
Summary	<p>The system will calculate and display the efficiency of a team in the tournament based on the number of games won compared to the number of games played. Efficiency is calculated as the ratio of games won to total games played.</p> <p>The result will be presented as a percentage.</p>		
Postcondition	The efficiency of the team is calculated and displayed as a percentage.		
Outputs	<b>Output Name</b>	<b>Data Type</b>	<b>Format</b>
	Efficiency	String	Name of the team, e.g., "Team A". Total games played, e.g., "5". Total games won, e.g., "3". Efficiency percentage, e.g., "60%".
	SuccessMessage	String	"The team efficiency has been successfully retrieved."
	Error Message	String	"Invalid information, please verify that it meets the conditions."

Identifier and name	RF14: Calculate and show a player's efficiency (number of goals compared to games played).
---------------------	--

Summary	The system will calculate and display a player's efficiency in the tournament, based on the number of goals scored compared to the number of games played. Efficiency is calculated as the ratio of goals scored to total games played. The result will be presented as a percentage.		
Postcondition	The efficiency of the player is calculated and displayed as a percentage.		
Outputs	<b>Output Name</b>	<b>Data Type</b>	<b>Format</b>
	PlayerEfficiency	String	Name of the player, e.g., " Player_1". Total games played, e.g., "5". Total goals scored, e.g., "4". Efficiency percentage, e.g., "80%".
	SuccessMessage	String	"The player's efficiency has been successfully retrieved."
	Error Message	String	"Invalid information, please verify that it meets the conditions."

Identifier and name	RF15: Calculate and display the card index of a central referee (number of cards with respect to refereed matches).		
Summary	The system will calculate and display the card index of a central referee, based on the number of yellow and red cards given during the matches they have refereed. The card index is calculated as the total number of cards divided by the total number of matches officiated. The result will be presented as a ratio or a number (e.g., "2 cards per-match").		
Postcondition	The card index of the central referee is successfully calculated and displayed.		
Outputs	<b>Output Name</b>	<b>Data Type</b>	<b>Format</b>
	CardIndex	String	Name of the referee, e.g., "Player_1". Total matches refereed, e.g., "10". Total cards given (yellow + red), e.g., "15". Card index, e.g., "1.5 cards per match".
	SuccessMessage	String	"The referee's card index has been successfully retrieved."
	Error Message	String	"Invalid information, please verify that it meets the conditions."