# Problem Identification and Requirements Analysis

## Study Case: Soccer Tournament

|  |  |
| --- | --- |
| Client | ICESI |
| User | Organizers, players, coaches, referees and fans. |
| Functional Requirements | *RF1: Register team*  *RF2: Register player*  *RF3: Register Referee*  *RF4: Preload data* |
| Problem context | *This tournament consists of three phases: a group stage, semifinals, and finals. Teams will be organized into two groups of four, each consisting of 20 players, a coach, and representing a specific country. For each match, it’s essential to manage player availability, assign referees, and maintain comprehensive statistics for players, teams, and referees. We need to develop an application that streamlines this process, standardizing registration and providing an accessible way to view information at any phase. The application should track game scores, goals, assists, cards, and points. At the tournament’s conclusion, it will recognize the champion and present awards, such as the Golden Boot and Fair Play, along with displaying all relevant performance metrics for players, teams, and referees.* |
| Requerimientos no funcionales | * Easy to use. * Fast. * Quick response. |
| Requerimientos de proceso | * The application must be developed using Java. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier and name | *RF1: Register team* | | | |
| Summary | *Teams that want to participate in the tournament will have to complete some questions in order to register correctly. They will be asked about the team’s name, the country where the the team is from, and the name of the coach of the team.* | | | |
| Inputs | Input name | **Type of data** | | **Valid values** |
| nameTeam | String | | *No numbers, just characters.* |
| countryOfTeam | String | | No numbers, just characters. |
| nameCoach | String | | No numbers, just characters. |
| Results or postcondition: | After the user types all the requested data, all of this information will be stored in a class called Team. The constructor for this class already includes attributes such as goals scored, assists, yellow cards, red cards, etc., but all of these values will all start at 0 and, therefore, do not require input upon initialization. | | | |
| Outputs | **Output name** | | **Data type** | **Format** |
| messageConfirmation | | String | *"Team registered successfully."* |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier and name | *RF2: Register player* | | | |
| Summary | *The app has to register all the information of each player. The users would need to type the players name, jersey number, their country of origin, the position where they play on the field, and the team in where they will be playing.* | | | |
| Inputs | **Input name** | **Type of data** | | **Valid Values** |
| namePlayer | String | | *No numbers, just characters.* |
| playerNumber | int | | Only positive numbers from 1 – 100 |
|  | country | String | | *No numbers, just characters.* |
|  | playerPosition | PlayerPosition | | GOALKEEPER, DEFENDER, MIDFIELDER, FORWARD |
|  | nameTeam | String | | The team in which the player will play. |
| Results or postcondition: | The player that was just registered will be related to that team (if that is already registered), with all of the previous information. | | | |
| Outputs | **Output name** | | **Type of data** | **Format** |
| messageConfirmation | | String | *"Player succesfully added to team"* |
| error1 | | String | "Team not found, please check again" |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier and name | *RF3: Register referee* | | | |
| Summary | *The app must be able to register 12 referees for the tournament. Those 12 referees would be separated with 4 central referees and 8 assistant referees. The users must type or enter their name, the referee ID, their country, and the type of referee.* | | | |
| Inputs | **Input name** | **Type of data** | | **Valid Values** |
| nameReferee | String | | *No numbers, just characters.* |
| refereeID | String | | A chain of letters and numbers. |
|  | country | String | | *No numbers, just characters.* |
|  | refType | RefereeType | | *CENTRAL, ASSISTANT* |
| Results or postcondition: | After the user enters all the previous information, similar as what happened with the player, the referee will be created. | | | |
| Outputs | **Output name** | | **Type of data** | **Format** |
| messageConfirmation | | String | *"Referee registered successfully."* |
| error1 | | String | "Cannot register referee.” |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identifier and name | *RF4: Preload information* | | | |
| Summary | *If the user wishes, the app must be able to preload information about teams, players and referees.* | | | |
| Inputs | **Input name** | **Type of data** | | **Valid Values** |
| N/A | N/A | | *N/A* |
| Results or postcondition: | The user will see a message on screen that will confirm if the data was succesfully preloaded. | | | |
| Outputs | **Output name** | | **Type of data** | **Format** |
| messageConfirmation | | String | *"Data preloaded successfully."* |