E-Sports Requirements Specifications: Dynamic-Trio

Kabelo Mamadi(10301004)-Thulasizwe Mavuso(29236259)- Renato De Jesus(28113561)

May 23, 2014

1   Project background

The purpose of this system is to create E Sports leagues to facilitate Derivco offices to maintain the leagues that they participate in. This system will be used by Derivco employees in Derivco offices worldwide to be made aware of challenges and record results electronically. The system will replace the current manual system using excel sheets that the employees are currently using. The system will allow for the administrators to setup up a whole league where match dates and opponents are selected automatically by the system. Once the league is set up teams will be entered that will participate in the league. Each team will have a captain that will enter scores and upload proof of scores into the system.

The system must provide and organised storage and presentation of this data. Team captains will also have to confirm each others scores. Team members will be able to log in and view current standing past results, leader boards and upcoming matches. Each team will have a profile that can be viewed by all other teams too. The system will consist of two main functional systems, Namely the database and web interface.

2   Project vision and scope

The system as per client request will be developed in the .Net framework. A MS SQL Server database will be created to conform to the clients current database systems. The database will be interfaced with and through the web client. This will be design in framework 4.5, using asp and visual C-sharp backend. A specific mobile client is not required as there are various platforms being used by Derivco employees. As a result the web client must be mobile browser compatible so that users may be able to access the system through their mobile phones independent of the phones platform. Users must be able to view upcoming matches, view scores and upload a photo of the final result and confirm this result through their mobile browsers.

The web client will work using windows authentication so that users will not have to have a separate login while on their PC. They will automatically be logged in using their windows user account. The users will be able to register on the system with the same details as their windows credentials to log in though their mobile devices.

3   Stakeholders

Client   Douglas Wilson is an employee at Derivco and is responsible for the project

Derivco staff   They will be the ones using the system to create leagues, create teams, join teams, upload and query scores.

4   Architectural requirements

4.1   Access channel requirements

- The system will be accessible by human users through the following channels:
    - From a web browser through a rich web interface. The system must be accessible from any of the widely used web browsers including all recent versions of Mozilla Firefox, Google Chrome, Apple Safari and Microsoft Internet Explorer.
    - The interface will have to support mobile web browsers to cater for the different platforms.

4.2 Quality requirements

- Security

  - All system functionality is only accessible to users who can be authenticated through windows authentication for the PC client. Mobile users will have to be authenticated through username and password.

  - Some services only require role based authorization for the service itself. However, the system must be able to constrain who is allowed to do what with which entity. That it, administrators will be the only ones with access to the audit log.

- Auditability

  - One should be able to query for any entity, any changes made to that entity or any of its components. The information provided must include by whom the change was made, when the change was made, and the new and old value of the field(s) which were changed. The system will provide only services to extract information from the audit log and will not allow the audit log to be modified.

- Testability

  - All services offered by the system must be testable through unit tests which test that the service is provided if all pre-conditions are met (i.e. that no exception is raised except if one of the pre-conditions for the service is not met), and that all post-conditions hold true once the service has been provided.

- Usability

  - 98 percent of users should be able to use the system without prior training.

- Scalability

  - The deployed system must be able to operate effectively under the load of 100 concurrent users.

  - The system must be made to receive any number of games and support photo uploads for each match.

- Performance requirements

  - Web pages should not exceed the size of 150kb.

  - Each page should load within 3 4 seconds.

4.3 Architecture constraints

- The following architecture constraints have been introduced largely for maintainability reasons:

  - The system must be developed using the .Net framework.

  - The unit tests should be developed using the Microsoft unit test module.

  - The system must ultimately be deployed onto an IIS application server.

  - The system must be compatible with mobile browsers.

  - The system must be decoupled from the choice of database. The system will use the MS SQLServer database.

5   Functional requirements

5.1   Required functionality

The web client will provide a login and register page to allow users to register if they are not on the system. If a user has logged in through their windows account they will automatically be logged into the system. If they log in through mobile and have forgotten their password, a reset password page will also be available where the user can request their password be reset by providing their email address and following a rest link. Users who register as administrators will have to be confirmed administrators by one of the current administrators.
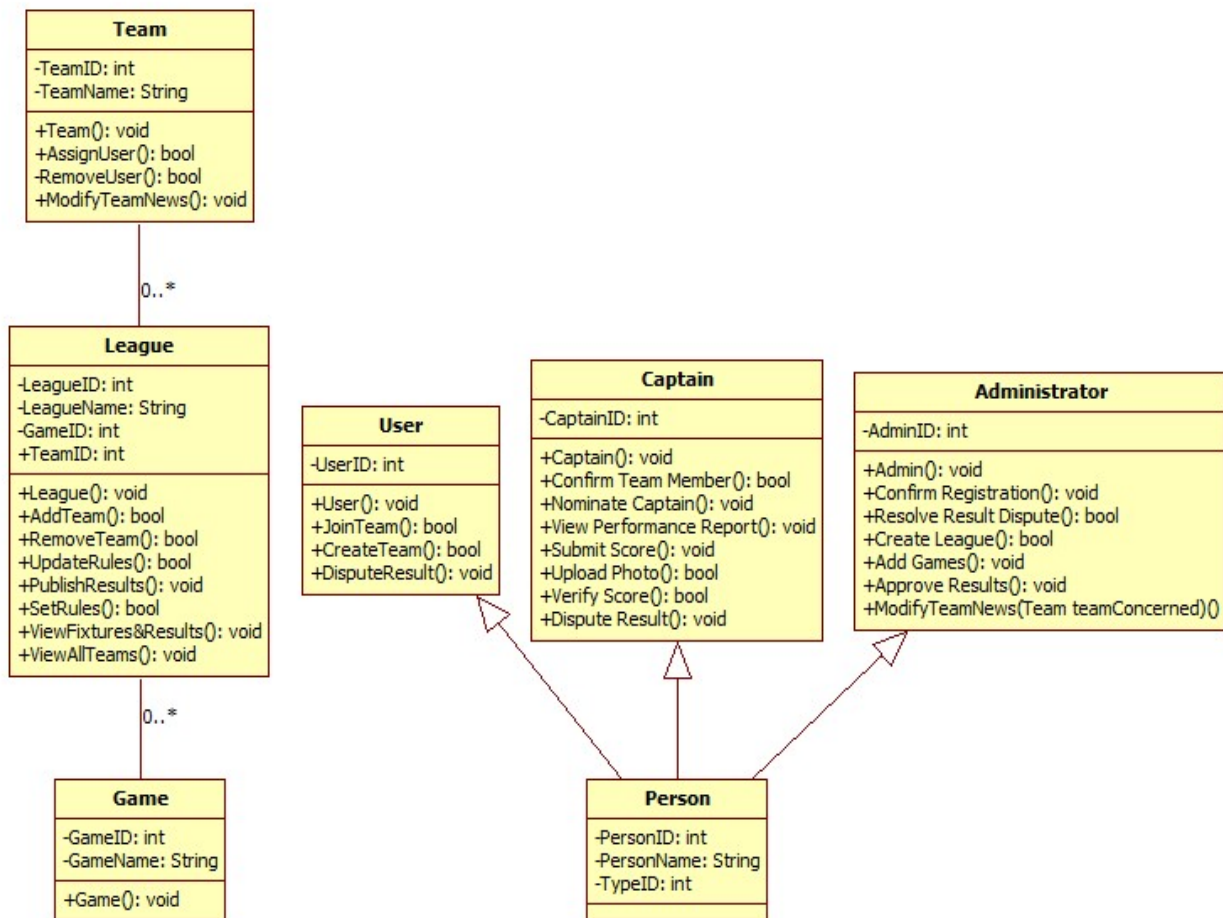
Figure 1: Login
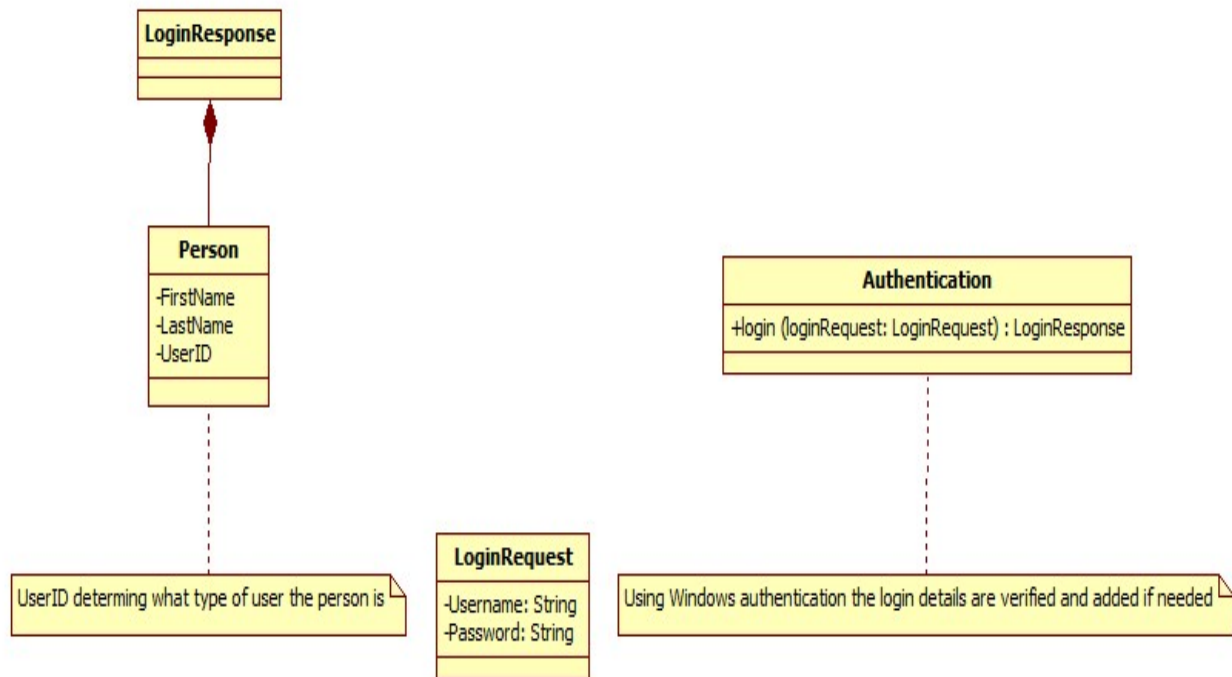
The database will have tables for multiple sections: user information with user roles to determine regular users, administrators and captains. This information will then be used to place users in teams. Tables that will contain the games being played, these will then be made available to create a league for a particular game. Tables will also be created to store the league participants, time and date of matches once generated and results for played matches.

The web user interface will be determined by the role of the user logging in (unconfirmed administrators will have the regular user interface until they have been verified). For a regular user they will be redirected to a home page that will display the latest news about the leagues. They will be able follow links to view their current team, team standings, next fixture and previous results. They will also be able to see who the other team members are and track their progress by following their team profile. A regular user will be able to view all scores and view all teams in their current league in the results and fixtures page. If the user does not belong to a team they may create one or join a particular team. Each team will nominate a team captain and deputy captain. The captains will then be responsible of submitting the scores and verifying them. This can be done via mobile where they will be able to upload a photo as proof of the result, or via the normal web client by uploading a screen shot or photo of the result. Once the scores have been verified by captains and approved by an administrator it will be available for all to see in the fixtures and results page.

3

Figure 2: Main

Team
- -TeamID: int
- -TeamName: String

- +Team(): void
- +AssignUser(): bool
- -RemoveUser(): bool
- +ModifyTeamNews(): void

0..*

League
- -LeagueID: int
- -LeagueName: String
- -GameID: int
- +TeamID: int

- +League(): void
- +AddTeam(): bool
- +RemoveTeam(): bool
- +UpdateRules(): bool
- +PublishResults(): void
- +SetRules(): bool
- +ViewFixtures&Results(): void
- +ViewAllTeams(): void

0..*

Game
- -GameID: int
- -GameName: String

- +Game(): void

User
- -UserID: int

- +User(): void
- +JoinTeam(): bool
- +CreateTeam(): bool
- +DisputeResult(): void

Captain
- -CaptainID: int

- +Captain(): void
- +Confirm Team Member(): bool
- +Nominate Captain(): void
- +View Performance Report(): void
- +Submit Score(): void
- +Upload Photo(): bool
- +Verify Score(): bool
- +Dispute Result(): void

Administrator
- -AdminID: int

- +Admin(): void
- +Confirm Registration(): void
- +Resolve Result Dispute(): bool
- +Create League(): bool
- +Add Games(): void
- +Approve Results(): void
- +ModifyTeamNews(Team teamConcerned)()

Person
- -PersonID: int
- -PersonName: String
- -TypeID: int

The captains will have to confirm that a new player has joint the team in the event that a new user joins an existing team. Users will also be able to request a report on the performance of their own team or opposing teams to view their track record or that of their opponents. Teams will be able to see all upcoming features in the game calendar. Regular users will also have a dispute option to dispute the result sent by the opposing team. These should also be done by the team captains. If a dispute is sent the league administrator will be notified by mail. The mail will have the score screen shots automatically attached. The administrator can then sort the dispute among the teams. Regular users will also have a page where they are able to modify their personal details and customize the look and feel of their interface.

The administrator interface will only be visible in the web client. All users will have the same mobile interface. The administrator interface will also redirect the user to a page with all the latest news. They will have all the functionality described above as per that of a regular user but with additional options and functionality. An administrator will have an option to administrate users, such as granting admin rights, editing user information and adding or removing users. Administrators will be able to add games to the system and create a league for the specific game. Administrators will also be responsible for setting league rules and will be able to update the league rules.

Once an administrator creates a league and adds the teams to the league the fixtures will be automatically generated on the days and times determined by the admin as match days and times. This will also take time zone differences into account. Administrators like regular users may view the league progress. They will also be able to request a team performance report on any of the teams. Once the teams submit the scores, the administrator will have to approve the result. If there is no

dispute, the administrator may just publish the scores. If a dispute has been received the administrator can resolve the dispute then publish the scores. Administrators will also be able to add news to the page that all users will view on login. Team captains will have the same interface as a regular user but they will have added functionality to submit scores, upload photos and verify scores.
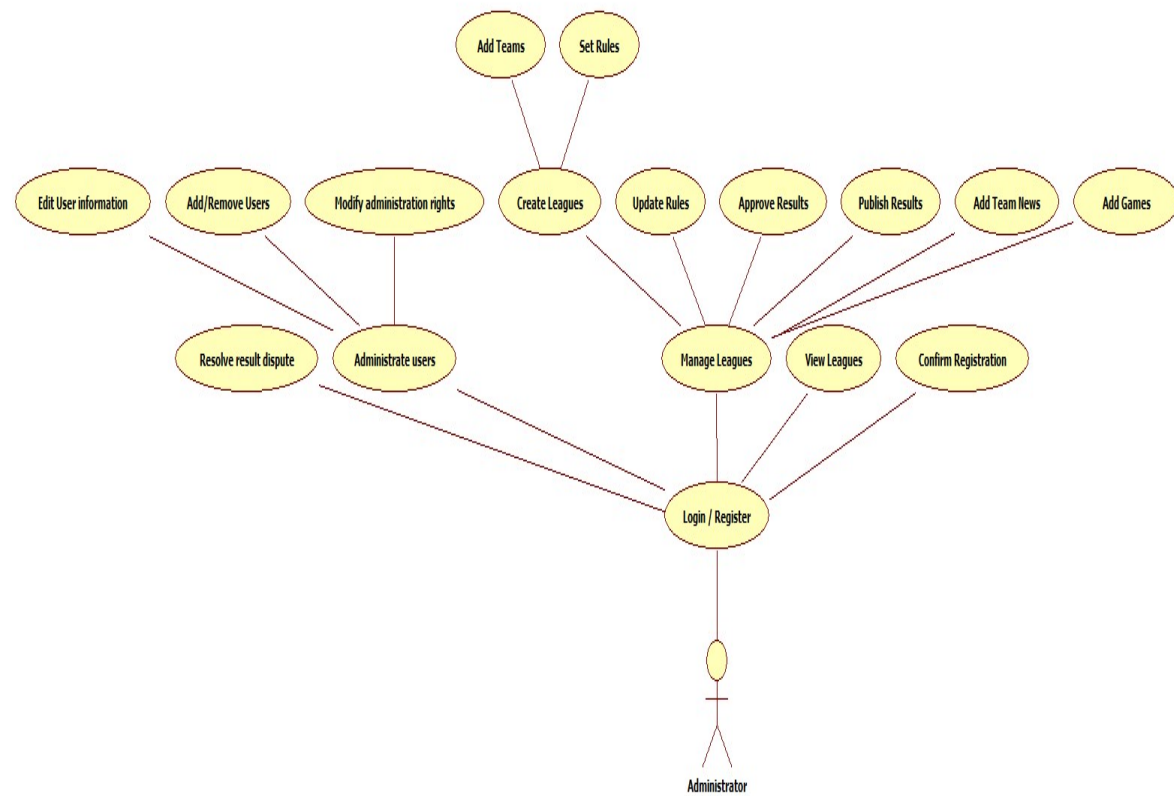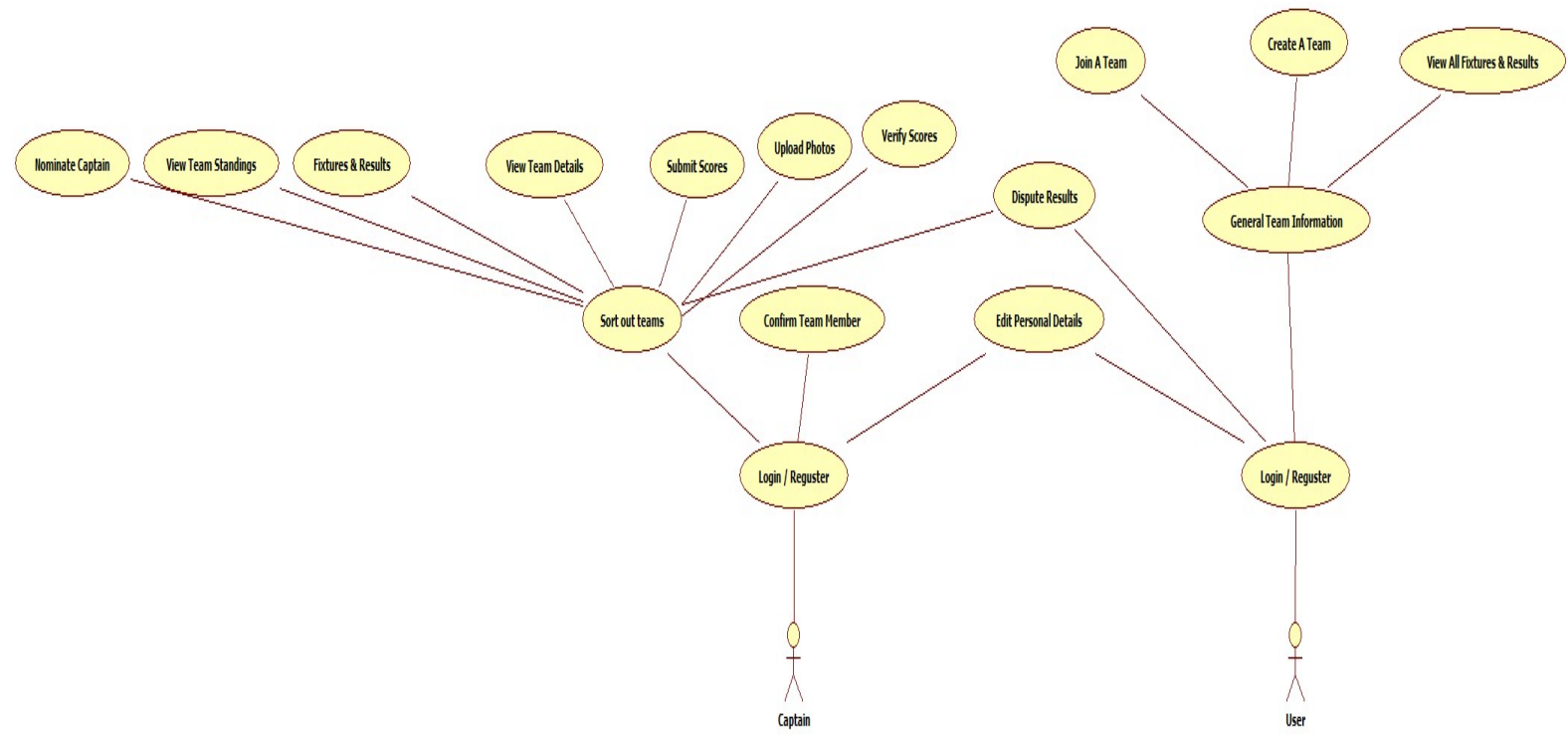
Figure 3: Use Case Diagram

Figure 3: Use Case Diagram

6

Figure 4: Use Case Diagram
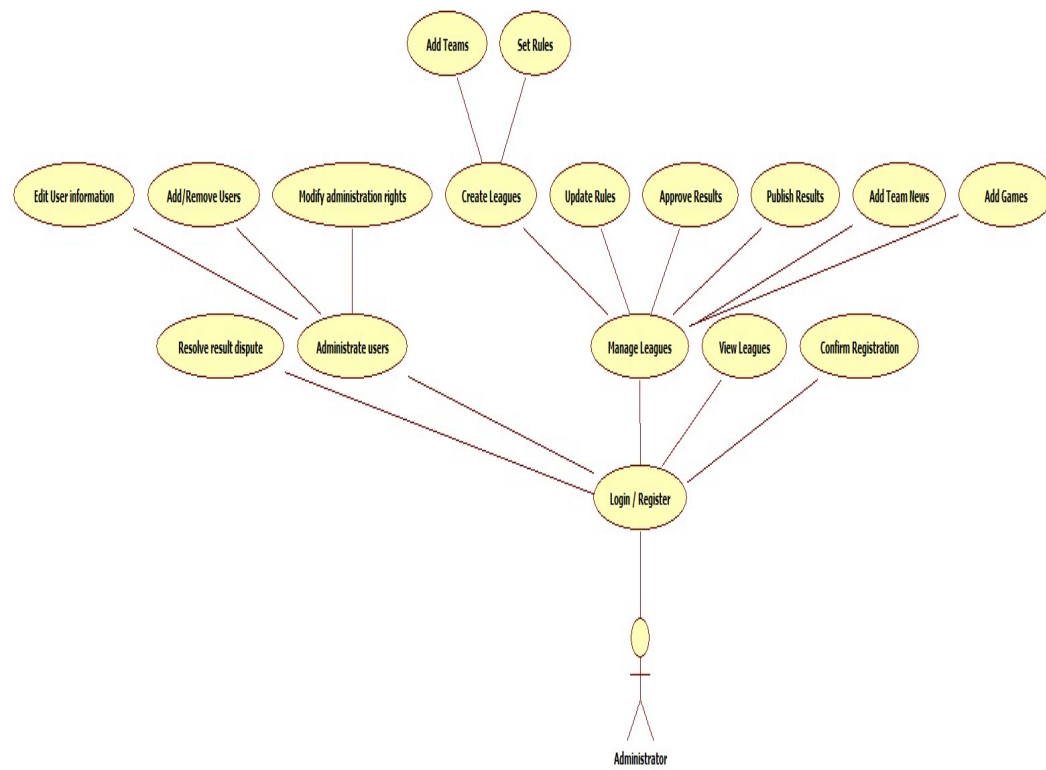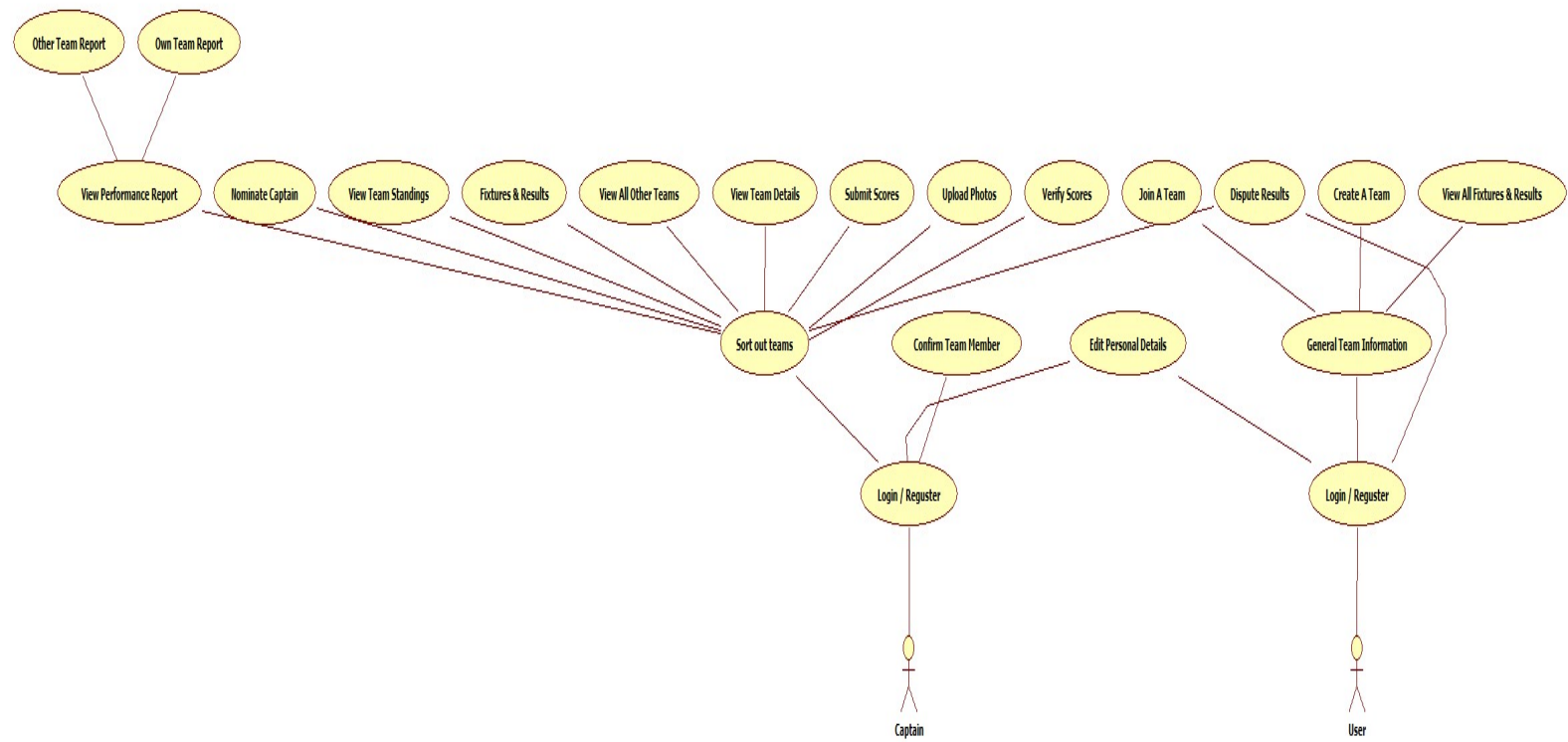
Figure 5: Use Case Diagram

Nominate Captain View Team Standings Fixtures & Results View Team Details Submit Scores Upload Photos Verify Scores

Join A Team Create A Team View All Fixtures & Results

Dispute Results

Sort out teams Confirm Team Member Edit Personal Details General Team Information

Login / Reguster

Login / Reguster

Captain User

Add Teams Set Rules

Edit User information Add/Remove Users Modify administration rights Create Leagues Approve Results Publish Results Add Team News Add Games

Administrate users Manage Leagues

Resolve result dispute Confirm Registration

Login / Register
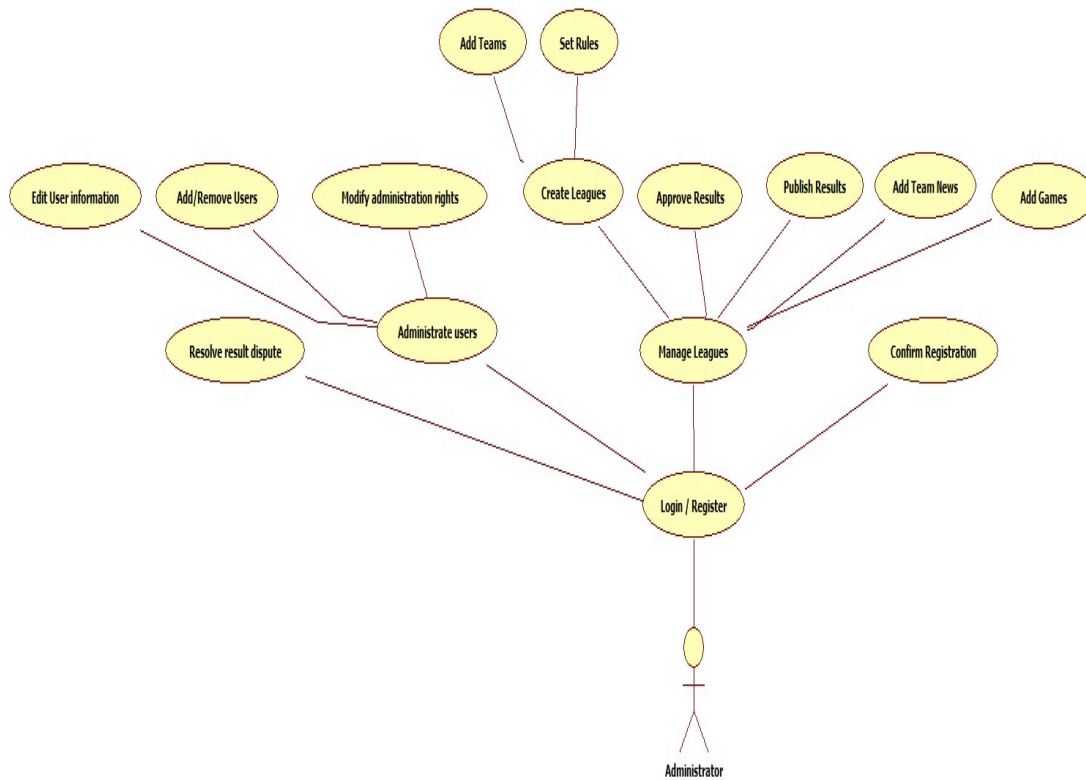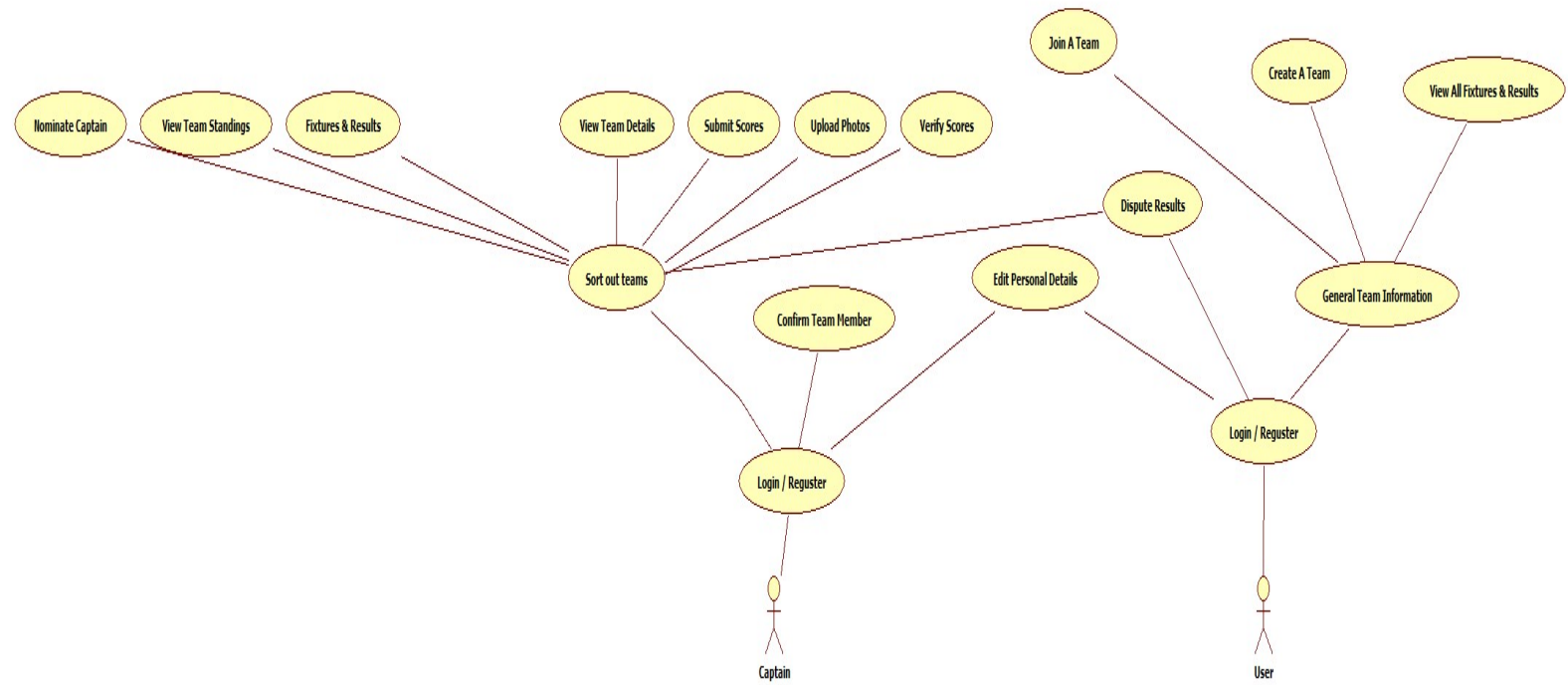
Administrator

Figure 6: Use Case Diagram

Other Team Report
Own Team Report
View Performance Report
Nominate Captain
View Team Standings
Fixtures & Results
View All Other Teams
View Team Details
Submit Scores
Upload Photos
Verify Scores
Dispute Results
Join A Team
Create A Team
View All Fixtures & Results
General Team Information
Sort out teams
Confirm Team Member
Edit Personal Details
<<extend>>
<<extend>>
<<extend>>
<<extend>>
<<include>>
<<include>>
<<include>>
<<include>>
Login / Reguster
Login / Reguster
Captain
User

Add Teams
Set Rules
Publish Results
Add Team News
Approve Results
Edit User information
Add/Remove Users
Modify administration rights
Create Leagues
Update Rules
Add Games
<<include>>
<<include>>
<<include>>
<<include>>
<<extend>>
<<include>>
Resolve result dispute
Administrate users
Manage Leagues
View Leagues
Confirm Registration
Login / Register
Administrator

9