# 

Software Requirements Specification

Version 1.2

December 07, 2016

JOGU

José Esteves

Ricardo Malafaya

Submitted in partial fulfilment

of

# ERMS - Requirements Engineering and Software Modeling, FEUP, 2016

Contents

[ERMS - Requirements Engineering and Software Modeling, FEUP, 2016 1](#_Toc468895113)

[1.0. Introduction 3](#_Toc468895114)

[1.1. Summary 3](#_Toc468895115)

[1.2. Goals 3](#_Toc468895116)

[1.3. Goals And/Or diagram 4](#_Toc468895117)

[1.4. Glossary 4](#_Toc468895118)

[1.5. References 4](#_Toc468895119)

[1.6. Stakeholders 5](#_Toc468895120)

[1.7. Version Control 5](#_Toc468895121)

[2.0. Overall Description 6](#_Toc468895122)

[2.1 System Environment 6](#_Toc468895123)

[2.2 Actors and roles 7](#_Toc468895124)

[2.3 Functional Requirements Specification 8](#_Toc468895125)

[2.3.1 Login use case 8](#_Toc468895126)

[2.3.2 Create a group use case 9](#_Toc468895127)

[2.3.3 Create an event use case 10](#_Toc468895128)

[2.3.4 Consult an event use case 11](#_Toc468895129)

[2.3.5 Accepts invitation use case 12](#_Toc468895130)

[2.3.6 Publish stat use case 13](#_Toc468895131)

[2.3.7 Overwrite stat use case 14](#_Toc468895132)

[2.3.8 Change user permission use case 15](#_Toc468895133)

[2.3.9 Change team use case 16](#_Toc468895134)

[2.3.10 Check stats use case 17](#_Toc468895135)

[2.4 Non-Functional Requirements 18](#_Toc468895136)

[3.0. Appendix 19](#_Toc468895137)

[3.1 Activity Diagram 19](#_Toc468895138)

[3.2 Class Diagram 20](#_Toc468895139)

[3.3 State Diagram 20](#_Toc468895140)

[3.4 Requirements list with their attributes 21](#_Toc468895141)

[3.4.1 Labels Meaning 22](#_Toc468895142)

# 1.0. Introduction

## 1.1. Summary

The software JOGU is a mobile platform where football amateur players can share experiences. The users of this application can create football games, called events, and invite other users to join and create a game. This app permit to share live goals, attendance and results of each event. Each player can also check past games, goals and statistics.

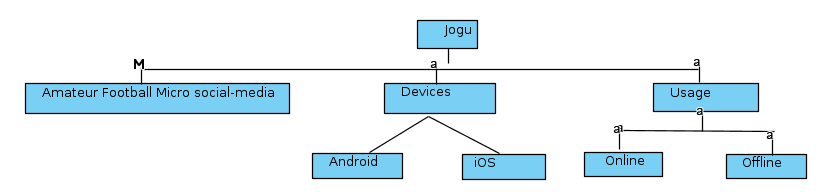
## 1.2. Goals

This software is a platform of share experience of the amateur football players. The user can create events. To each event, other users can be invited and attend. After the users confirm that they will attend the event, the teams are made. All the participants of the event can share stats of the event. The creator of the event is set as an administrator of that event and can overwrite any stat publish by the attendees. The administrator can attribute administrator’s permissions to other users on the event.

Each user can connect to application through login and access their own stats.

This application is targeting mobile devices (Android and IOS). To publish stats the application needs access to internet but also will permit to consult the stats offline.

## 1.3. Goals And/Or diagram



## 1.4. Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| User | Any amateur football player that use the application to create events, attend to events or check stats. |
| Event | Is a meeting created by the users which users can attend with the information about the time and the place where the amateur football game will be taking place. |
| Administrator | In case of incorrect stats, the administrator is the user with the permission to alter the stats publish by the users. |
| Super Administrator | Person of developing team charged to intervene in case of failure. |
| Group | A list of users. A group can be associated to several events. |
| Stats | Published information that can be goals, attendance or final score of the event |
| Team | At event, the participants are divided in to groups. |
| Invitation | An invitation send by the creator of the event. The users can accept this invitation |
| Statistics | Information that the user can access about the events, goals, results and attendance of past events. |

## 1.5. References

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

## 1.6. Stakeholders

The relevant stakeholder of this application are the product owner, the group of developers and the people that will use the app, both those who interact with its features and users as well as the users who are just viewing and exploring the displayed information.

## 1.7. Version Control

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version** | **Revision** | **Name** | **Function** | **Signature** | **Date** |
| 1 | 1 | Iteration 0 | First draft | alpha | 2016-11-10 |
| 1 | 2 | Iteration 1 | PO Modifications | alpha | 2016-11-11 |
| 1 | 3 | Iteration 2 | Classes added | alpha | 2016-11-07 |

# 2.0. Overall Description

## 2.1 System Environment

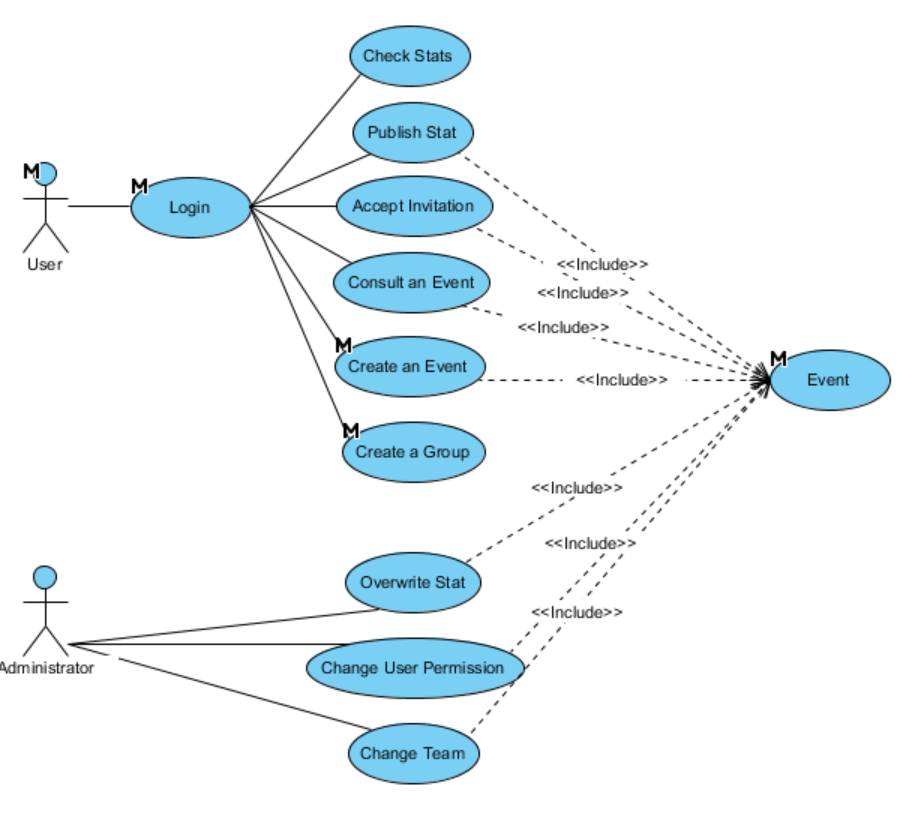


Figure 1 - System Environment

## 2.2 Actors and roles

The JOGU has two active actors.

The User and the Administrator. The Administrator is a user with more permissions. Any user can only access the applications through login. The login can be by email and password or using the social session of Google or Facebook.

The user is expected to create and consult events, check results and publish stats.

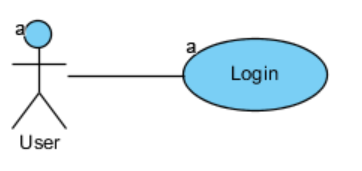
The administrator is expected to change any incorrect stat of an event. By default, user that creates an event is automatically assign as its administrator. The administrator can grant permissions to other users in the event.

## 2.3 Functional Requirements Specification

### 2.3.1 Login use case

#### Use case: The user can login in system to access the events and stats

**Diagram:**



**Brief Description**

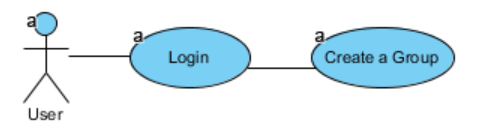
|  |  |
| --- | --- |
| **Use Case Name** | The user can login in system to access the events and stats |
| **XRef** | US 2.3.1 |
| **Trigger** | The user opens application |
| **Precondition** | The user has not yet logon on application |
| **Basic Path** | 1. The user opens the application 2. The system displays login by email, Google or Facebook. 3. The user has to choose the login type and enter the credentials. 4. The system will check if the credentials are correct 5. The main window of the app is shown |
| **Alternative Paths** | In step 2., if user already logon on app go to 5.  In step 4., if is the first login of the user, the system will send a verification email to user.   * 1. System hangs until confirmation of email and proceeds to 5. |
| **Postcondition** | The user credentials are registered in system.  Event registration will only be executed whilst having internet connection or upon its re-connection. |
| **Exception Paths** | In case of no internet, the system message user that cannot proceed without internet |

The user first interaction with the application is the login with the system. This login can be made by Email and Password or Google account or Facebook account. In case the user is not registered, then an email is sent to confirm his identity.

### 2.3.2 Create a group use case

#### Use case: The user creates a group to associate events.

**Diagram:**



**Brief Description**

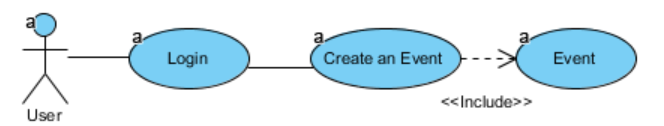
|  |  |
| --- | --- |
| **Use Case Name** | The user creates a group to associate events |
| **XRef** | US 2.3.2 |
| **Trigger** | The user selects the option “Create a Group” |
| **Precondition** | The user has already logon on system |
| **Basic Path** | 1. The user selects the users to add (from past events) or inputs the other users email. 2. The system verifies information and shows the group window |
| **Alternative Paths** |  |
| **Postcondition** | The system opens the group page |
| **Exception Paths** | In case of no internet, the system message user that cannot proceed without internet |

The user creates a group by add users from past events or from inputting an email. Add a group permits to the users to add events related to an event.

### 2.3.3 Create an event use case

#### Use case: The user can create events to invite users

**Diagram:**



**Brief Description**

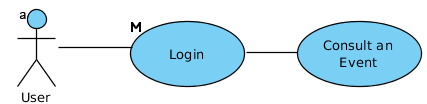
The user creates an event with date and location and invite other users to attend the event.

|  |  |
| --- | --- |
| **Use Case Name** | The user can create events to invite users |
| **XRef** | US 2.3.3 |
| **Trigger** | The user selects the option “Create an event” |
| **Precondition** | The user has already logon on system |
| **Basic Path** | 1. The system ask user to date and confirmation of the event 2. The system verifies information and shows the invite window 3. The user selects the group to invite. 4. The system notifies the invited users |
| **Alternative Paths** | 1. The user can select “Create new Group”   3.1 As soon as the group is created, go to step 4. |
| **Postcondition** | The system opens the event page |
| **Exception Paths** | In case of no internet, the system message user that cannot proceed without internet |

### 2.3.4 Consult an event use case

#### Use case: The user can select and event to consult it

**Diagram:**



**Brief Description**

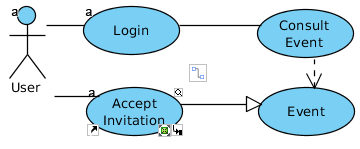
#### The user can consult the events available on the groups he is part of.

|  |  |
| --- | --- |
| **Use Case Name** | The user can consult an event. |
| **XRef** | US 2.3.4 |
| **Trigger** | The user selects the event's division. |
| **Precondition** | The user has already logged on to the system |
| **Basic Path** | 1. The system presents the available group events. 2. The user enters the group dashboard. 3. Selects the event he wishes. |
| **Alternative Paths** | The user will be directed to a specific event page after: Inviting a set of users; publishing stats;  Also, if he has administrative permissions:  Overwriting stats; changing user permissions; change user team. |
| **Postcondition** |  |
| **Exception Paths** |  |

### 2.3.5 Accepts invitation use case

#### Use case: The user can accept an invitation to attend an event.

**Diagram:**



**Brief Description**

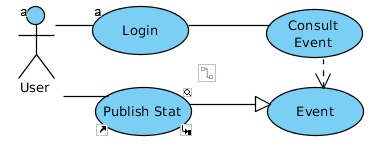
#### The user can receive notifications and accept them so he integrates a new event.

|  |  |
| --- | --- |
| **Use Case Name** | The user can accept invitation. |
| **XRef** | US 2.3.5 |
| **Trigger** | The user confirms the invitation on the notification. |
| **Precondition** | The user has already logged on to the system |
| **Basic Path** | 1. Opens notifications. 2. Confirms the invitation. |
| **Alternative Paths** |  |
| **Postcondition** | User will join the event as a possible team member.  Acceptance registration will only be executed whilst having internet connection or upon its re-connection. |
| **Exception Paths** | 1. User rejects invitation. 2. User ignores notification. |

### 2.3.6 Publish stat use case

#### Use case: The user can publish an event stats.

**Diagram:**



**Brief Description**

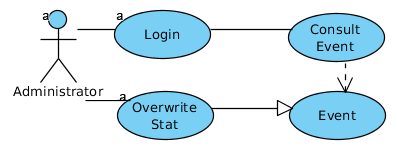
#### The user can input statistical information like goals and presence.

|  |  |
| --- | --- |
| **Use Case Name** | The user can publish event stats like goals and presences. |
| **XRef** | US 2.3.6 |
| **Trigger** |  |
| **Precondition** | The user has already logged on to the system |
| **Basic Path** | 1. Select the event where the statistic is derived from. 2. Input the information.   . |
| **Alternative Paths** | When publishing while having no internet connection, an input can conflict with another user's input, and it's the admin who'll settle the final input. He could decide in one's favor. |
| **Postcondition** | User will join the event as a possible team member.  Stats registration will only be executed whilst having internet connection or upon its re-connection. |
| **Exception Paths** | When publishing while having no internet connection, an input can conflict with another user's input, and it's the admin who'll settle the final input., so he could decide not to include one's input. An admin can also delete stats. |

### 2.3.7 Overwrite stat use case

#### Use case: The administrator can overwrite an event's statistical input.

**Diagram:**



**Brief Description**

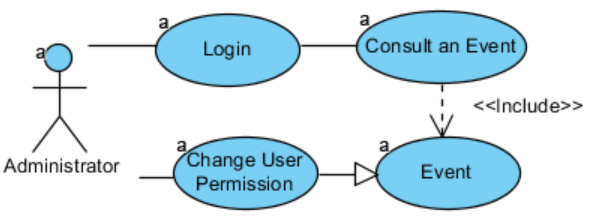
#### The administrator can overwrite a stat which was already in the system.

|  |  |
| --- | --- |
| **Use Case Name** | The user can |
| **XRef** | US 2.3.7 |
| **Trigger** | Event has been materialized and user has recorded metrics to populate the event's stats. |
| **Precondition** | The user has already logged on to the system  Acceptance registration will only be executed whilst having internet connection or upon its re-connection.  Statistical information was uploaded to the system. |
| **Basic Path** | 1. The system shows all stats from a event. 2. The administrator changes the existing value of an entry. |
| **Alternative Paths** |  |
| **Postcondition** | User will join the event as a possible team member.  Stats registration will only be executed whilst having internet connection or upon its re-connection. |
| **Exception Paths** |  |

### 2.3.8 Change user permission use case

#### Use case: The admin can delegate administrative powers to another user.

**Diagram:**



**Brief Description**

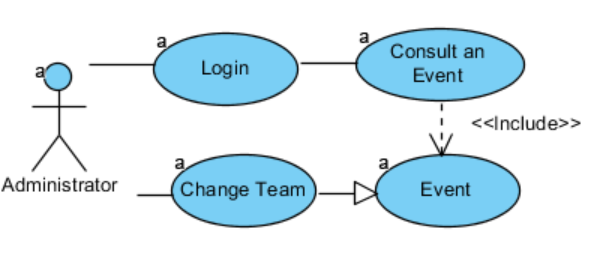
#### The administrator can access an event ad add permissions to other users. These users can now publish stats and change teams.

|  |  |
| --- | --- |
| **Use Case Name** | The admin can delegate administrative powers to another user. |
| **XRef** | US 2.3.8 |
| **Trigger** | The administrator choose option “Change User permission” on event window. |
| **Precondition** | The user is an administrator on event window. |
| **Basic Path** | 1. The system show all participants on event 2. The administrator chooses which users have now administrator rules. 3. The user selects done. 4. The system alters permissions of all selected users. |
| **Alternative Paths** |  |
| **Postcondition** |  |
| **Exception Paths** | Permission grants will only be attributed when having internet connection or upon its re-connection. |

### 2.3.9 Change team use case

#### Use case: The admin can change a user's team side.

**Diagram:**



**Brief Description**

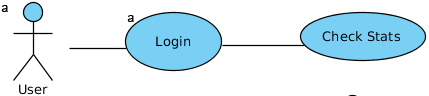
#### By default, in each event, the system will assign each player to the blue or red team. The administrator can move a user from one team to another.

|  |  |
| --- | --- |
| **Use Case Name** | The admin can change a user's team side. |
| **XRef** | US 2.3.9 |
| **Trigger** | On participant window, the administrator selects participant and choose option “Change team” |
| **Precondition** | The user is logged on to the system and is an administrator on participant’s window of the event. The event has participants. |
| **Basic Path** | 1. The administrator chooses user and selects “Change team”. 2. The system will change user team. |
| **Alternative Paths** |  |
| **Postcondition** |  |
| **Exception Paths** | The user will only change team when administrator have internet connection or upon its re-connection. |

### 2.3.10 Check stats use case

#### Use case: The user can check an event's statistical information.

**Diagram:**



**Brief Description**

#### The user can access his own goals and results from past events.

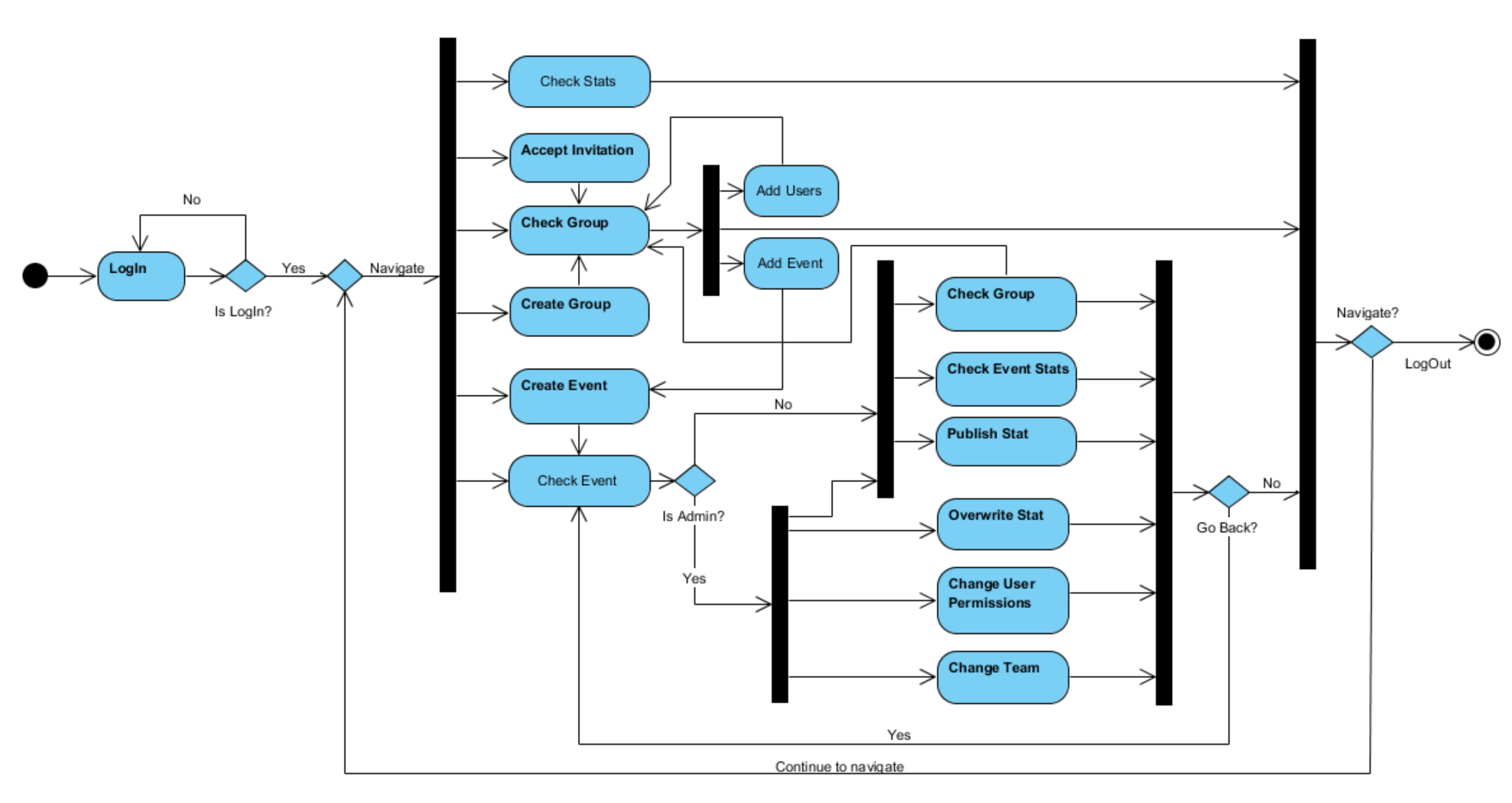
|  |  |
| --- | --- |
| **Use Case Name** | The user can check an event's statistical information. |
| **XRef** | US 2.3.10 |
| **Trigger** | The user selects Check stats from main window |
| **Precondition** | The user has already logged on to the system. |
| **Basic Path** | 1. The system refresh information from internet. 2. The system shows the information of user’s past events. |
| **Alternative Paths** |  |
| **Postcondition** |  |
| **Exception Paths** | In case of no access to internet, the system will only show already downloaded data. |

## 2.4 Non-Functional Requirements

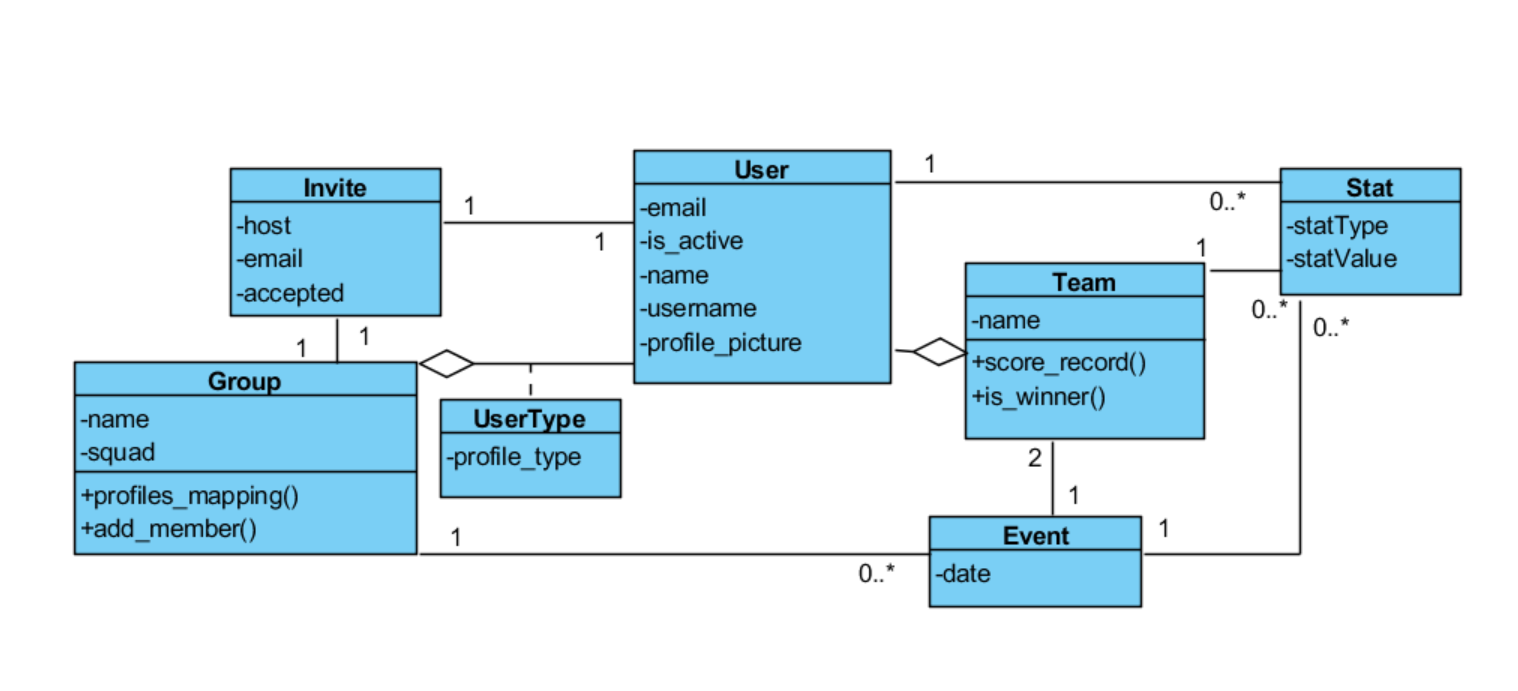
The system presentation has to be appealing to users who identify themselves with football. This so they can link and build up their natural use of the application to their football events routine.

# 3.0. Appendix

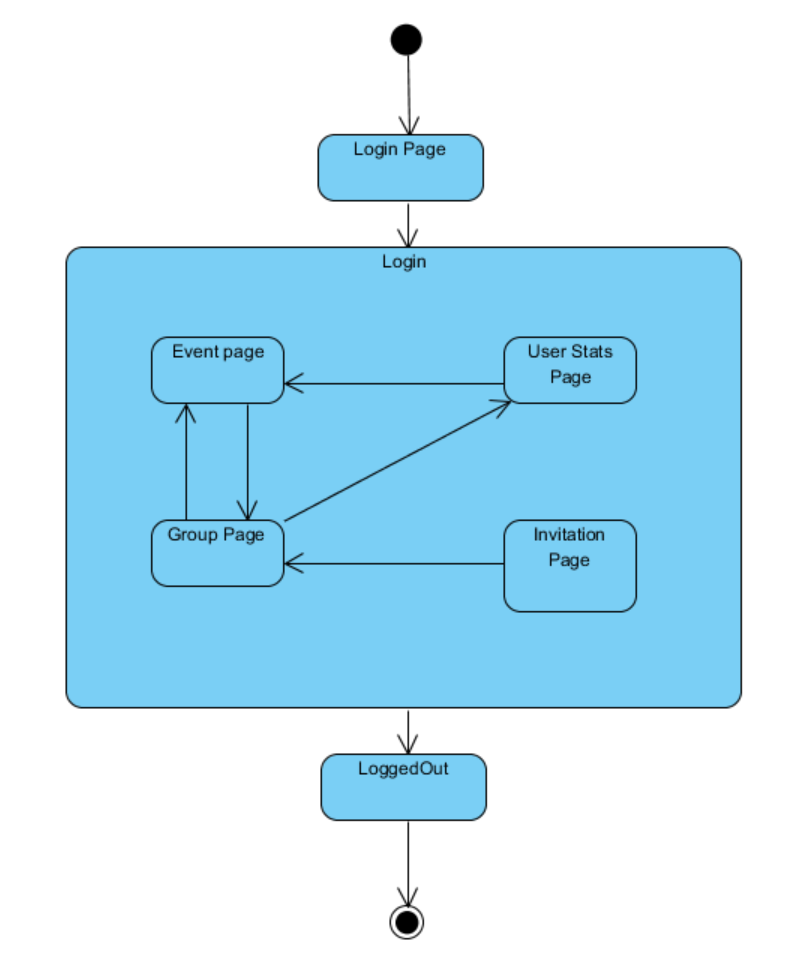
## 3.1 Activity Diagram



## 3.2 Class Diagram



## 3.3 State Diagram



## 3.4 Requirements list with their attributes

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Name of Use Case** | **Identifier** | **Date Created** | **Current Version** | **Author** | **Stability** | **Status** | **Origin/Source** | **Effort** | **Priority** | **Stakeholders** | **Dependcies** |
| Login | US 2.3.1 | 10/11/2016 | 1.1 | José & Ricardo | stable | approved | P.O. | 13 | 1 | P.O./Users/Devs |  |
| Create an Event | US 2.3.2 | 11/11/2016 | 1.2 | José & Ricardo | changing | proposed | P.O. | 21 | 3 | P.O./Users/Devs | US 2.3.1 + US 2.3.3 |
| Create a Group | US 2.3.3 | 11/11/2016 | 1.2 | José & Ricardo | changing | proposed | P.O. | 8 | 2 | P.O./Users/Devs | US 2.3.1 |
| Consult an event | US 2.3.4 | 10/11/2016 | 1.1 | José & Ricardo | stable | approved | P.O. | 8 | 4 | P.O./Users/Devs | US 2.3.2 |
| Accepts invitation | US 2.3.5 | 10/11/2016 | 1.1 | José & Ricardo | stable | approved | P.O. | 13 | 6 | P.O./Users/Devs | US 2.3.2 |
| Publish stat | US 2.3.6 | 10/11/2016 | 1.1 | José & Ricardo | stable | approved | P.O. | 13 | 7 | P.O./Users/Devs | US 2.3.2 |
| Overwrite | US 2.3.7 | 10/11/2016 | 1.1 | José & Ricardo | stable | approved | P.O. | 8 | 9 | P.O./Users/Devs | US 2.3.2 + US 2.3.6 |
| Change user permission | US 2.3.8 | 10/11/2016 | 1.1 | José & Ricardo | stable | approved | P.O. | 8 | 10 | P.O./Users/Devs | US 2.3.2 |
| Change team | US 2.3.9 | 10/11/2016 | 1.1 | José & Ricardo | stable | approved | P.O. | 8 | 5 | P.O./Users/Devs | US 2.3.2 |
| Check stats | US 2.3.10 | 10/11/2016 | 1.1 | José & Ricardo | stable | approved | P.O. | 8 | 8 | P.O./Users/Devs | US 2.3.6 |

### 3.4.1 Labels Meaning

**Identifier** – the id of the specific requirement this must be a unique.

**Date Created** – when the requirement was created.

**Current Version** – Last version that this requirement was modified.

**Author** – the name of the user or class of user that raised the requirement.

**Status** – this should indicate the stage at which the requirement is for example proposed; verified; approved; deleted; implemented; and rejected.

**Priority** – this should indicate the priority in relation to other requirements.

**Effort** – the effort of this requirement in relation to the success of the project.

**Stability** – the likelihood to the requirement changing for instance mature; stable; or still changing.

**Stakeholders** – the stakeholders that should be involved should any change requests be made that impact this requirement