# 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 pass 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 information of the teams, goals and score. The creator of the event is set as an administrator of that event and can overwrite any state, goal, score 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 results. |
| 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. |
| Stats | Published information that can be goals, attendance or final score of the event |
| Goal | Information about a goal in an event with the information about the author of the goal |
| Score/Results | Final score of event between two teams |
| Team | At event, the participants are divided in to groups. |
| Attendance | Each user that receive an invitation can or not attend the event |
| Stats/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.

# 2.0. Overall Description

## 2.1 System Environment

Author

Reader

Editor

HS DB

Online Journal

Article Manager

Web Publishing System

Reviewer

Figure 1 - System Environment

The Web Publishing System has four active actors and one cooperating system.

The Author, Reader, or Reviewer accesses the Online Journal through the Internet. Any Author or Reviewer communication with the system is through email. The Editor accesses the entire system directly. There is a link to the (existing) Historical Society.

<< The division of the Web Publishing System into two component parts, the Online Journal and the Article Manager, is an example of using domain classes to make an explanation clearer. >>

## 2.2 Actors and roles

The Reader is expected to be Internet literate and be able to use a search engine. The main screen of the Online Journal Website will have the search function and a link to “Author/Reviewer Information.”

The Author and Reviewer are expected to be Internet literate and to be able to use email with attachments.

The Editor is expected to be Windows literate and to be able to use button, pull-down menus, and similar tools.

The detailed look of these pages is discussed in section 3.2 below.

## 2.3 Functional Requirements Specification

This section outlines the use cases for each of the active readers separately. The reader, the author and the reviewer have only one use case apiece while the editor is main actor in this system.

### 2.2.1 Reader Use Case

#### Use case: Search Article

**Diagram:**

Reader

Search Article

**Brief Description**

The Reader accesses the Online Journal Website, searches for an article and downloads it to his/her machine.

**Initial Step-By-Step Description**

Before this use case can be initiated, the Reader has already accessed the Online Journal Website.

1. The Reader chooses to search by author name, category, or keyword.
2. The system displays the choices to the Reader.
3. The Reader selects the article desired.
4. The system presents the abstract of the article to the reader.
5. The Reader chooses to download the article.
6. The system provides the requested article.

**Xref:** Section 3.2.1, Search Article

## 2.4 Non-Functional Requirements

The Online Journal will be on a server with high speed Internet capability. The physical machine to be used will be determined by the Historical Society. The software developed here assumes the use of a tool such as Tomcat for connection between the Web pages and the database. The speed of the Reader’s connection will depend on the hardware used rather than characteristics of this system.

The Article Manager will run on the editor’s PC and will contain an Access database. Access is already installed on this computer and is a Windows operating system.

# 3.0. Requirements Specification

## 3.1 External Interface Requirements

The only link to an external system is the link to the Historical Society (HS) Database to verify the membership of a Reviewer. The Editor believes that a society member is much more likely to be an effective reviewer and has imposed a membership requirement for a Reviewer. The HS Database fields of interest to the Web Publishing Systems are member’s name, membership (ID) number, and email address (an optional field for the HS Database).

The *Assign Reviewer* use case sends the Reviewer ID to the HS Database and a Boolean is returned denoting membership status. The *Update Reviewer* use case requests a list of member names, membership numbers and (optional) email addresses when adding a new Reviewer. It returns a Boolean for membership status when updating a Reviewer.

## 3.2 Functional Requirements

The Logical Structure of the Data is contained in Section 3.3.1.

### 3.2.1 Search Article

|  |  |
| --- | --- |
| **Use Case Name** | Search Article |
| **XRef** | Section 2.2.1, Search Article  SDD, Section 7.1 |
| **Trigger** | The Reader assesses the Online Journal Website |
| **Precondition** | The Web is displayed with grids for searching |
| **Basic Path** | 1. The Reader chooses how to search the Web site. The choices are by Author, by Category, and by Keyword. 2. If the search is by Author, the system creates and presents an alphabetical list of all authors in the database. In the case of an article with multiple authors, each is contained in the list. 3. The Reader selects an author. 4. The system creates and presents a list of all articles by that author in the database. 5. The Reader selects an article. 6. The system displays the Abstract for the article. 7. The Reader selects to download the article or to return to the article list or to the previous list. |
| **Alternative Paths** | In step 2, if the Reader selects to search by category, the system creates and presents a list of all categories in the database.   1. The Reader selects a category. 2. The system creates and presents a list of all articles in that category in the database. Return to step 5.   In step 2, if the Reader selects to search by keyword, the system presents a dialog box to enter the keyword or phrase.   1. The Reader enters a keyword or phrase. 2. The system searches the Abstracts for all articles with that keyword or phrase and creates and presents a list of all such articles in the database. Return to step 5. |
| **Postcondition** | The selected article is downloaded to the client machine. |
| **Exception Paths** | The Reader may abandon the search at any time. |
| **Other** | The categories list is generated from the information provided when article are published and not predefined in the Online Journal database. |

## 3.3 Detailed Non-Functional Requirements

### 3.3.1 Logical Structure of the Data

The logical structure of the data to be stored in the internal Article Manager database is given below.

Review

Reviewer

Article

Author

writes

sent to

writes

has

Figure 4 - Logical Structure of the Article Manager Data

The data descriptions of each of these data entities is as follows:

**Author Data Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| Name | Text | Name of principle author |  |
| Email Address | Text | Internet address |  |
| Article | Pointer | Article entity | May be several |

The Logical Structure of the data to be stored in the Online Journal database on the server is as follows:

**Published Article Entity**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Type** | **Description** | **Comment** |
| Name | Text | Name of Article |  |
| Author | Text | Name of one Author | May be several |
| Abstract | Text | Abstract of article | Used for keyword search |
| Content | Text | Body of article |  |
| Category | Text | Area of content | May be several |

### 3.3.2 Security

The server on which the Online Journal resides will have its own security to prevent unauthorized *write*/*delete* access. There is no restriction on *read* access. The use of email by an Author or Reviewer is on the client systems and thus is external to the system.

The PC on which the Article Manager resides will have its own security. Only the Editor will have physical access to the machine and the program on it. There is no special protection built into this system other than to provide the editor with *write* access to the Online Journal to publish an article.