Gestionnaire de club scientifique de l’ÉTS – GCS

**SimplETS**

Document SRS

Version <1.0>

Historique des révisions

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Auteur** |
| 2014-15-07 | 1.0 | Version initiale | Jean Bernier Vibert |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table des matières

1. Introduction 4

1.1 Objectif 4

1.2 Portée 4

1.3 Références 4

1.4 Hypothèses et dépendances 4

2. Survol du modèle des cas d’utilisation 4

2.1 Diagramme des cas d’utilisation 4

2.2 Cas d’utilisation 4

3. Les acteurs 4

4. Les exigences 4

4.1 Les exigences fonctionnelles 4

4.2 Les exigences non fonctionnelles 5

4.2.1 Facilité d’utilisation 5

4.2.2 Fiabilité 5

4.2.3 Performance 6

4.2.4 Facilité d’entretien 6

5. Documentation en direct pour l’utilisateur et exigences du système d’aide 6

6. Contraintes de conception 6

7. Interfaces 6

7.1 Interfaces Utilisateur 7

7.2 Interfaces Matérielles 7

7.3 Interfaces Logicielles 7

7.4 Interfaces de Communications 7

8. Standards applicables 7

8.1 Glossaire 7

8.2 Annexes 7

Document de vision

# Introduction

## Objectif

L’objectif de ce SRS est de définir, détailler et analyser les différentes exigences fonctionnelles et non fonctionnelles du projet de gestion de club étudiant « SimplETS » développé pour l’équipe « Sporacid studio ». Ce document présente les différents cas d’utilisations, les contraintes applicables, les interfaces graphiques ainsi que les différentes modifications apportées suite aux tests avec les utilisateurs.

## Portée

This section provides a brief description of the software application that the SRS applies to, the features or other subsystem grouping, what use-case model(s) it is associated with, and anything else that is affected or influenced by this document.

« Gest-LIFE » est un logiciel actuellement en place, mais ce dernier ne répond pas aux besoins des clubs et il n’est que partiellement utilisé, à cause de diverse frustrations de la part de ces utilisateurs. Excel, un outil très puissant aussi utilisé dans le cadre de la gestion des clubs, souffre de plusieurs lacunes fonctionnelles (impossibilité d’avoir plusieurs simultanément, sécurité difficile, etc.) l’empêchant d’être un outil acceptable dans la situation actuelle.

## Références

Provide a list of project-related references or applicable documents that bear on this project.

## Hypothèses et dépendances

Les hypothèses et dépendances du document de vision sont également applicables dans le SRS. À des fins de simplicité, voici une retranscription.

« Afin de mener à terme l’application Web *GCS* il est important de poser certaines hypothèses. Celles-ci permettront de jauger le risque quant à la réalisation du projet.

* La disponibilité du service VPN de l’école sera accessible de partout.
* L’intégration sur le LDAP de l’école se fera sans problème et l’Active Directory contiendra les données nécessaires pour identifier un étudiant.
* Les utilisateurs auront une connaissance générale de l’informatique et connaîtront en tout temps leurs critères d’authentification
* Les utilisateurs n’abuseront pas des données nominatives des autres utilisateurs de l’application. »

# Survol du modèle des cas d’utilisation

## Diagramme des cas d’utilisation

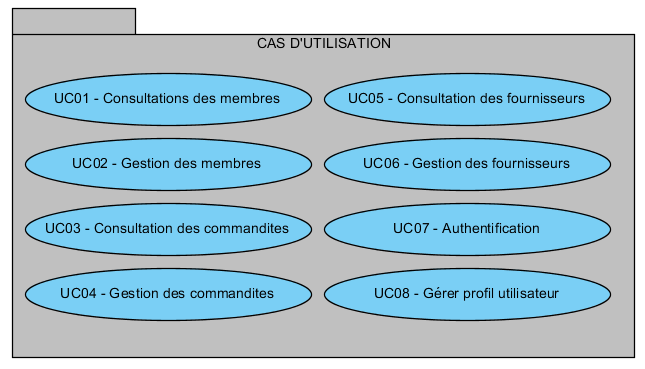


Figure 1 - Diagramme des cas d'utilisation

## Cas d’utilisation

### UC01 – Consultation des membres

Description :   
  
Acteur :

### UC02 – Gestion des membres

Description :   
  
Acteur :

### UC03 – Consultation des commandites

Description :   
  
Acteur :

### UC04 – Gestion des commandites

Description :   
  
Acteur :

### UC05 – Consultation des fournisseurs

Description :   
  
Acteur :

### UC06 – Gestion des fournisseurs

Description :   
  
Acteur :

### UC07 – Authentification

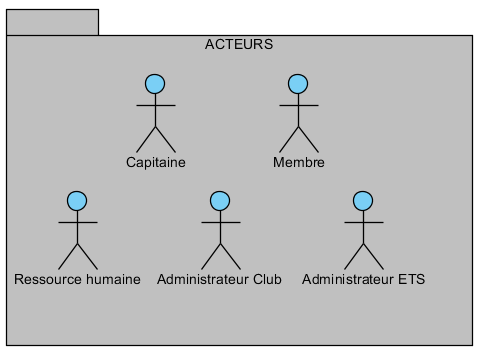
Description :   
  
Acteur :

### UC08 – Gestion du profil utilisateur

Description :   
  
Acteur :

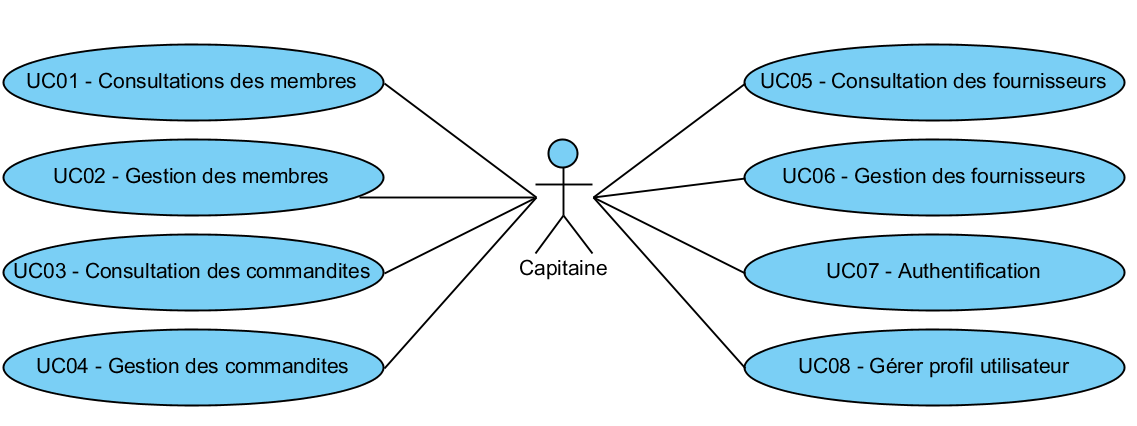
# Les acteurs

### Sommaire



### Capitaine

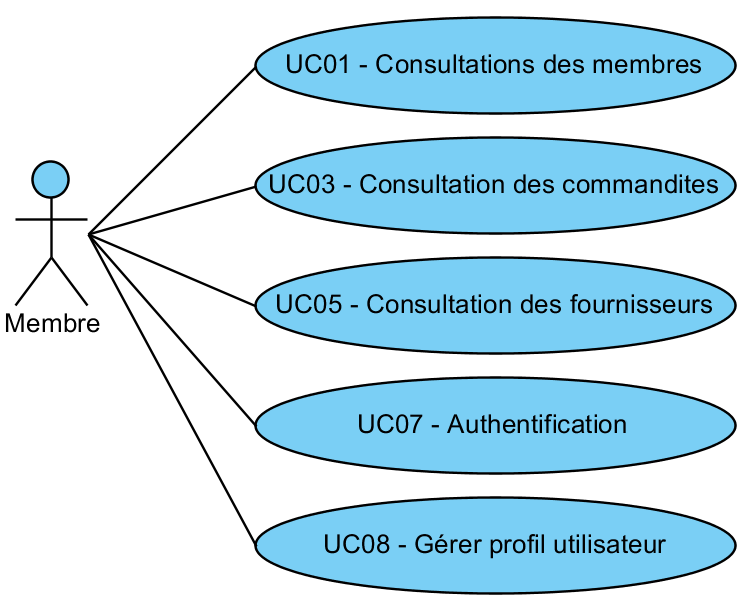
BLABLA



### Membre

Le membre sera l’acteur qui utilisera le système en plus grand nombre. Il représente l’utilisateur lambda du système.

Blabls



### Ressource humaine

### Administrateur Club

### Administrateur ETS

# Les exigences

## Les exigences fonctionnelles

This section describes the functional requirements of the system for those requirements that are expressed in the natural-language style. For many applications, this may constitute the bulk of the package, and thought should be given to the organization of this section. This section is typically organized by feature, but alternative organization methods, by user or by subsystem, may also be appropriate.

Where application development tools (requirements tools, modeling tools, and so on) are used to capture the functionality, this section of the document will refer to the availability of that data and will indicate the location and name of the tool used to capture the data.

## Les exigences non fonctionnelles

Most nonfunctional requirements are typically recorded in natural language in this section for the specification. However, nonfunctional requirements may also be included with a specific use case specification.

### Facilité d’utilisation

This section should include all of those requirements that affect usability. These often include:

* Specify the required training time for normal users and power users to become productive at particular operations.
* Specify measurable task times for typical tasks; alternatively, base usability requirements of the new system on other systems that the users know and like.
* Specify requirements to conform to common usability standards, such as IBM’s CUA standards or the GUI standards published by Microsoft for Windows 98. Refer to the User’s Bill of Rights in Chapter 23 for additional guidelines.

### Fiabilité

Requirements for system reliability should be specified here.

* Availability: Specify percent of time available (xx.xx%), hours of use, maintenance access, degraded-mode operations, and so on.
* Mean time between failures (MTBF): This is usually specified in hours but could also be specified in terms of days, months, or years.
* Mean time to repair (MTTR): How long is the system allowed to be out of operation after it has failed?
* Accuracy: Specify precision (resolution) and accuracy (by some know standard) that is required in the system’s output.
* Maximum bugs or defect rate: Usually expressed in terms of bugs/KLOC (thousands of lines of code) or bugs per function-point.
* Bugs or defect rate: Categorized in terms of minor, significant, and critical bugs. The requirement(s) must define what is meant by a “critical” bug (such as complete loss of data or complete inability to use certain parts of the functionality of the system).

### Performance

The performance characteristics of the system should be outlined in this section. Include specific response times. Where applicable, reference related use cases by name.

* Response time for a transaction (average, maximum).
* Throughput (transactions per second).
* Capacity (the number of customers or transactions the system can accommodate).
* Degradation modes (the acceptable mode of operation when the system has been degraded).
* Resource utilization (memory, disk, communications).

### Facilité d’entretien

This section indicates any requirements that will enhance the supportability or maintainability of the system being built, including coding standards, naming conventions, class libraries, maintenance access, and maintenance utilities.

# Documentation en direct pour l’utilisateur et exigences du système d’aide

Describes the requirements, if any, for online user documentation, help systems, help notices, and so on.

# Contraintes de conception

This section should indicate any design constraints on the system being built. Design constraints represent design decisions that have been mandated and must be adhered to. Examples include software languages, software process requirements, prescribed use of developmental tools, architectural and design constraints, purchased components, and class libraries.

# Interfaces

This section defines the interfaces that must be supported by the application. This section should contain adequate specificity, protocols, ports, and logical addresses, and so on, so that the software can be developed and verified against the interface requirements.

## Interfaces Utilisateur

Describe the user interfaces that are to be implemented by the software.

## Interfaces Matérielles

Define any hardware interfaces that are to be supported by the software, including logical structure, physical addresses, and expected behavior.

## Interfaces Logicielles

Describe software interfaces to other components of the software system. These may be purchased components, components reused from another application, or components being developed for subsystems outside of the scope of this SRS but with which this software application must interact.

## Interfaces de Communications

Describe any communications interfaces to other systems or devices, such as local area networks or remote serial devices.

# Standards applicables

Describe by reference any standards (and the specific sections of any such standards) that apply to the system being described. For example, this could include legal, quality, and regulatory standards, as well as industry standards for usability, interoperability, internationalization, operating system compliance, and so on.

## Glossaire

Describe any terms that are unique to this application context and any definitions, acronyms, abbreviations, or other project or company-specific shorthand that is necessary for an understanding of this document and the application.

## Annexes

You should insert appendixes here as appropriate. Note that the following template appendix is provided specifically to allow you to record use cases. Feel free to insert as many appendixes as you need.