Sommaire

[Introduction générale 1](#_Toc99531319)

[Chapitre 1 : Spécifications des besoins 2](#_Toc99531320)

[1-Introduction : 2](#_Toc99531321)

[2-contexte du système : 2](#_Toc99531322)

[4.1- besoins non fonctionnels techniques : 3](#_Toc99531323)

[4.2- besoins non fonctionnels ergonomiques : 4](#_Toc99531324)

[5.1-Identification des cas d'utilisation : 5](#_Toc99531325)

[5.2-Diagramme de cas d’utilisation globale : 6](#_Toc99531326)

# Introduction générale

Digitaliser la vie sociale et entrepreneuriale dans l’université tunisienne, en assurant l’organisation des activités des clubs universitaires à distance, s’inscrit notre projet intitulé «ClubOrganizer».

Pour la modélisation de notre application, nous avons suivi la méthodologie de conception « Scrum ».

Notre rapport est organisé comme suit :

- chapitre 1 intitulé « spécifications de besoins »

- chapitre 2 intitulé «sprint 0 »

- chapitre 3 intitulé «sprint 1»

Nous clôturons notre rapport par une conclusion générale et perspectives.

# Chapitre 1 : Spécifications des besoins

## 1-Introduction :

Dans ce chapitre, on entame et la spécification des besoins. On commence par définir les besoins fonctionnels et non fonctionnels de l’application. On établit ensuite le diagramme des cas d'utilisation. Ce dernier donne une vision globale du comportement fonctionnel de la solution.

## 2-contexte du système :

IL s’agit d’un site web qui gère les activités d’un certain nombre de clubs universitaires, dans le dessein de faciliter la communication au sein d’un même club via le partage des publications et un espace privilégié pour les événements. De plus,

Les visiteurs de notre site après leur inscription qui peuvent être des étudiants ou bien des formateurs, ont également le privilège de partager des publications dans le fil d’actualité, de participer à des évènements, et de rejoindre les clubs. La suppression des adhérents entre autre est tenue par le modérateur de site. Ce dernier gère les demandes de création des clubs aussi.

La demande de créer un club se fait par le président du club, qui gère les publications des membres au sein du club.

3-Identification des besoins fonctionnels :

Un besoin fonctionnel est généralement formulé sous formes d’exigences fonctionnelles, les besoins fonctionnels sont l’expression de ce que le produit ou le service délivré par le projet devrait être ou faire. Les besoins fonctionnels sont spécifiés par les représentants des utilisateurs et des bénéficiaires du produit /service délivré par le projet.

Notre système permet de :

-Consulter le site ;

-Enregistrer la demande d’abonnement ;

-Publier statut ;

-S’authentifier ;

-Gérer les publications ;

-Enregistrer les demandes de création des clubs ;

-traiter les demandes d’abonnement ;

-Consulter les clubs ;

-Traiter les demandes de création du club ;

-Gérer compte ;

4-Identification des besoins non Fonctionnels :

Un besoin non fonctionnel est une restriction ou une contrainte qui pèse sur un service du système, telles que les contraintes liées à l’environnement et à l’implémentation et les exigences en matière de performances, les dépendances du projet, de facilité de maintenance, d’extensibilité et de fiabilité.

Dans le cadre de ce travail, l'application devra répondre à ces besoins :

**La sécurité :** Tous les accès des utilisateurs (admin, visiteur, étudiant membre ou président du club) doivent être protégés par un login et un mot de passe pour parvenir à la sécurité de la plate-forme;

**Efficacité :** L'application doit être fonctionnelle indépendamment de toutes circonstances pouvant entourer l'utilisateur (une solution efficace);

**Validité :**(correction, justesse, conformité), réaliser exactement les tâches définies dans la spécification;

**La performance :** Temps de réponse court.

**La fiabilité :** Les données fournies par l'application doivent êtres fiables et la solution doit rendre des résultats corrects.

Pour notre application, nous devons obéir aux critères ergonomiques de Scapin et Bastien suivantes :

* **Guidage :** L’ensemble des moyens mis en œuvre pour conseiller, orienter, informer et conduire l'utilisateur lors de ses interactions avec l'ordinateur.
* **Feedback immédiat :** Dans tous les cas, l'ordinateur doit répondre à l'utilisateur en fonction des actions et des requêtes de ce dernier.
* **Brièveté :** Limiter le travail de lecture, d'entrée et les étapes par lesquelles doivent passer les usagers.
* **Adaptabilité :** Capacité à réagir selon le contexte et selon les besoins et les préférences des utilisateurs.
* **Prise en compte de l'expérience de l'utilisateur :** Le système doit respecter le niveau d'expérience de l'utilisateur; donc il doit être simple, compréhensible et facile à utiliser.

5-Identification des acteurs :

Un acteur est une personne, un matériel ou un logiciel qui interagit avec le système dans le but de réaliser une ou plusieurs fonctions concernant les cas d’utilisations. Nous décrivons pour chaque acteur les cas d’utilisation. On distingue les cas d'utilisation suivants :

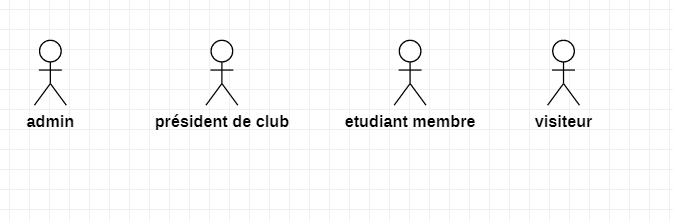


Figure 1: les acteurs

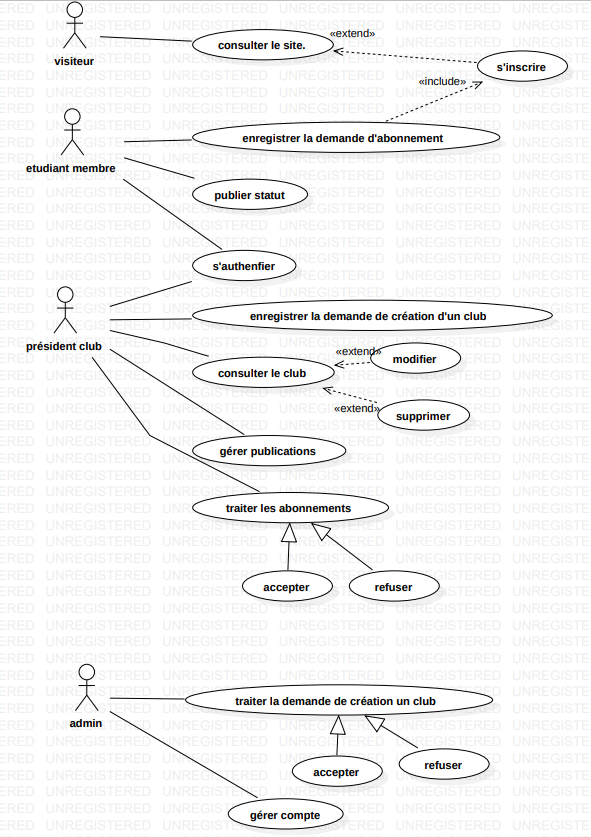
|  |  |
| --- | --- |
| Acteur | Cas d'utilisation |
| visiteur | -consulter le site,  -s’inscrire. |
| Etudiant membre | - consulter le site,  -enregistrer la demande  d’abonnement,  -pulier statut,  -s'authentifier. |
| President du club | -s’authentifier,  -enregistrer la demande de creation un club,  -Gérer publication,  -consulter club,  -traiter les demandes d’abonnement,  -envoyer e-mail. |
| admin | -traiter les demandes de creation d’un club,  -gérer compte,  -recevoir des notifications. |
|  |  |

**Identification des cas d’utilisation**

## 5.-Diagramme de cas d’utilisation globale :

La figure ci-dessous représente les différents

Cas d’utilisation de notre système.



**6-Affectation de priorités : (backlog)**

|  |  |
| --- | --- |
| **Cas d’utilisation** | **Priorité** |
| s’inscrire | 1 |
| S’authentifier | 1 |
| enregistrer la demande d’abonnement au club | 1 |
| enregistrer la demande de création d’un club | 1 |
| Gérer publication | 2 |
| Traiter les abonnements | 1 |
| Traitement des demandes de créations | 1 |
| Consulter le site | 2 |
| Consulter le club | 2 |
| Gérer le compte | 2 |

**7-Environnement du travail:**

**7.1 – Méthodologie de conception(Scrum) :**

Scrum est une [méthodologie agile](https://chef-de-projet.fr/methodologie-agile/) utilisée dans le développement de logiciels basés sur un processus itératif et incrémental. Scrum est un cadre agile adaptable, rapide, flexible et efficace qui est conçu pour apporter de la valeur au client tout au long du développement du projet. L’objectif principal de Scrum est de satisfaire le besoin du client grâce à un environnement de transparence dans la communication, de responsabilité collective et de progrès continu.

Le développement part d’une idée générale de ce qui doit être construit, en élaborant une liste de caractéristiques classées par priorité (backlog du produit) que le propriétaire du produit veut obtenir.

**7.2 – Environnement logiciel :**

Nous avons choisir comme logiciel pour implémenter ce projet :

-Visual studio code

-Angular.

-bootstrap.

-My sql.

Les langages de programmation suivants :

-UML (Unified Modeling Language)  ou « langage de modélisation unifié ») est un langage de modélisation graphique, il est apparu dans le monde du [génie logiciel](http://fr.wikipedia.org/wiki/G%C3%A9nie_logiciel), dans le cadre de la « [conception orientée objet](http://fr.wikipedia.org/wiki/Programmation_orient%C3%A9e_objet) ».

-sql.

-php.

-Html, css.

-Java script.

**8-Conclusion :**

Dans ce chapitre, nous avons spécifié les besoins de notre application en identifiant les acteurs et les fonctionnalités du système. Cette description est traduite à travers des diagrammes de cas d’utilisation. Dans le chapitre suivant nous entamons le sprint n°0.