Chapitre 1

**Introduction**

Après avoir donné un aperçu sur le contexte de projet, nous allons consacrer ce

premier chapitre, en premier lieu, à la définition des concepts clés les plus importants pour la bonne compréhension du projet. En second lieu, l’étude de l’existant regroupe les parties qui permettent d’analyser l’existant et nous dégageons, ainsi, les critiques du système actuel afin de développer une application de qualité dans le futur. Et enfin, nous décrivons les objectifs à atteindre et les avantages de l’application.

## Présentation du cadre de projet

### 1.1. Description du projet :

### 1.2. Objectifs du projet :

Dans un terrain où la formation en présentiel est très contraignante (disponibilité, frais de déplacement, etc.), le concept de la formation à distance offre des réponses

complémentaires et une plus grande flexibilité au apprenant qui pourra se former à son rythme en fonction de ses disponibilités.

Notre objectif consiste à créer une plateforme du social Learning qui répond au besoin du professeurs et des apprenants à la fois et de manière à :

* Faciliter l’interaction entre professeurs, étudiants, administration.
* Créer une plateforme pour donner des cours en ligne directement.
* Faciliter l’accès aux ressources et services.
* Permet de programmer des séances ( en ligne ou présentielle ).
* Offrir aux professeurs et apprenants un espace de gestion de homework.
* Faciliter le partage, la production et la collaboration à distance.

## Présentation de l’organisme d’accueil

### 2.1 Présentation générale de CrocoCoder

Crococoder est une startup académique Tunisienne qui offre des formations en nouvelles technologies à toutes personnes intéressées par l’informatique, elle est destinée aux étudiants amoureux du code aux professionnels voulant améliorer leurs compétences et à toutes personnes désirant se reconvertir dans les technologies de l’information, elle est accessible sans condition de diplôme.

### 2.2 Organigramme général de CrocoCoder

1. Abderrahmen Masmoudi : Chief Executive Officer ( Chef de la direction ou directeur général )

2. Yasmine Guirat et Med Amine Oueslati : graphistes

3. Sarra Ben Dhiab : Digital Marketer

4. Chams Hamza : Responsable de la communication

Et les différents professeurs.

2.3. Activités

Elle offre des formations dans l’apprentissage des nouvelles technologies en s'adaptant aux besoins de leurs apprenants.

Crococoder propose des formations en :

1. JavaScript : est un langage de programmation script léger orienté objet qui ajoute de l’interactivité à un site web.

2. Ruby on Rails : également appelé RoR ou Rails c’est un Framework web libre et populaire écrit en Ruby il a conquis beaucoup de professionnels avec sa documentation et ses bibliothèques gratuites.

3. React : c’est une bibliothèque javascript libre qui facilite la création des interfaces utilisateurs interactives, c' est le Framework frontend le plus populaire de nos jours

.

4. Data science : c’est la science des données. Elle permet à une entreprise d'explorer et d'analyser les données brutes pour les transformer en informations précieuses permettant de résoudre les problèmes de l'entreprise.

5. Angular : qui est un Framework open source écrit en javaScript et typeScript qui permet de développer des sites web…

## Etat de l’art

### C'est quoi, une plate-forme d'apprentissage en ligne?

Une plateforme pédagogique (en Anglais Learning Management System) **LMS** est un dispositif technologique et humain qui intègre des outils informatiques à des fins d'enseignement et d'apprentissage. Il a pour finalité l'accès à distance aux contenus pédagogiques, l'individualisation de l'apprentissage et le télétutorat.

Un **LMS** regroupe les outils nécessaires aux trois principaux utilisateurs de la plateforme :

* *enseignant / formateur* : création des parcours pédagogiques, des contenus pédagogiques et suivi de l'activité des apprenants.
* *Étudiant :* consultation en ligne ou téléchargement des contenus pédagogiques et transmission des travaux à corriger.
* *administrateur* : installation et maintenance du système, gestion des accès et création des liens vers d'autres systèmes d'information

### E-learning:

Avec le préfixe « E » pour web, numérique, ou cyber et « learning » pour apprentissage, le e-learning, signifie littéralement « formation sur internet» .

Nouvelle forme d’apprentissage, le e-learning tire son attrait du fait de pouvoir apprendre à son rythme, sur son ordinateur, des contenus pédagogiques sur des sujets variés. Organisée en sessions ou modules, avec tests d’évaluations, la formation peut-être totalement autogérée et suivie via un tableau de bord qui répertorie chacune des avancées du participant.

Pour se former, les plateformes se structurent autour de vidéos, d’animations, de textes, et de tests en tout genre.

Le but c’est d’obtenir pour certaines formations, un certificat d’aptitudes ou de connaissances, mais surtout améliorer sa capital de connaissance dans un domaine précis.

Point positif, les évaluations peuvent être refaites jusqu’à ce que l’exercice soit réussi ou totalement maîtrisé.

### Blended Learning:

Le mode d'apprentissage mixte désigne l'utilisation conjointe du eLearning et du mode classique d'apprentissage appelé souvent "présentiel". En général, l'apprenant va ainsi alterner entre des sessions à distance en ligne et des sessions en face-à-face avec le ou les formateurs. Un modèle souvent utilisé est ainsi d'effectuer une première introduction au sujet avec une ressource à distance, puis une période en face à face avec un enseignant suivra. Une session de débriefing est souvent aussi ajoutée en fin de formation, quelque temps après celle suivie en face-à-face.

### Le social Learning (L’apprentissage social)

Le social learning désigne une méthode d’apprentissage centrée sur les interactions entre les différents membres du groupe. De nombreuses études ont démontré que le social learning est plus efficace que les autres méthodes d’apprentissage.

Dans ce mode d’apprentissage, chacun des membres du groupe est à la fois apprenant et formateur. Les interactions entre les individus permettent d’obtenir plus d’explications et d’éclaircissement. Cette méthode d’apprentissage est transversale car elle peut s’appliquer à tout domaine d’activité. N’importe quel membre du groupe peut apporter sa contribution pour résoudre un problème.

## Etude de l’existant :

### Introduction :

Dans cette partie nous nous intéressons à l’étude de quelques sites similaires à la nôtre. L’analyse de l’existant comprend principalement trois parties qui sont la description de l’existant, la critique de l’existant et la solution proposée.

L’étude de l’existant est une étape préliminaire pour tout projet informatique, Dans cette phase, nous allons faire une analyse de l’existant pour dégager les faiblesses du système et le critique de l’existant.

### Description de l’existant :

Les réseaux sociaux d’apprentissage en ligne est une tendance qui vient d'apparaître dans le milieu académique lors du pandémie. Dans cette section nous allons présenter quelques réseaux sociaux d’apprentissage en ligne:

* DIDACTI : Une plateforme pédagogique qui permet aux enseignants et aux élèves de déposer des contenus sous forme d’activités d’apprentissages et les partager entre eux pour créer une interactivité [3].
* Google Classroom : est une plate-forme d'apprentissage gratuite dédiée aux écoles. Son but est de simplifier la création et la diffusion de cours et d'exercices de façon numérique.
* Zoom : est un logiciel propriétaire de visiophonie développé par Zoom Video Communications.
* L'université virtuelle de Tunis : UVT est une [université virtuelle](https://fr.wikipedia.org/wiki/Formation_%C3%A0_distance) publique [tunisienne](https://fr.wikipedia.org/wiki/Tunisie) fondée par le décret n°112-02 du [28](https://fr.wikipedia.org/wiki/28_janvier) [janvier](https://fr.wikipedia.org/wiki/Janvier_2002) [2002](https://fr.wikipedia.org/wiki/2002).

### Critique de l’existant :

Le tableau ci-dessous récapitule les avantages et les inconvénients de toutes les plateformes traitées.

|  |  |  |
| --- | --- | --- |
| Plateformes | Avantages | Inconvénients |
| DIDÀCTIC | * Répond à la demande, très stable, interface agréable. * facile à utiliser | * Version gratuite réduite * Encore en construction, pas très connue |
| Google Classroom |  | * Manque un chat direct avec le professeur |
| Zoom |  |  |
| université virtuelle de Tunis |  |  |

### Solution proposée:

Après avoir décrit et critiqué l’existant , ainsi l’objectif est de développer un site internet de réseau social pour le cadre universitaire en mettant à leurs dispositions des outils de bases pour échanger des informations , des idées , et de partager des centres d'intérêts , tel ques la planification des séances en lignes à traver live stream , contacter nos profs, nos collègues et l'administration à travers des messages privés , notifié lors d’une mise à jours administrative, publier des posts selon nos besoins , organiser des événements…

1. **Méthodologie du travail**

**5.1 Présentation des méthodologies de développement :**

l’équipe autour du projet était composée de deux développeurs informatique ,deux stagiaires développeurs, informatique encadrée par madame takwa ait hamou.

Dans le cadre de stage et de la gestion du projet , il fallait instaurer une démarche de travail pour le bon suivi des tâches . Les méthodes dites “Agiles” ont permis de faciliter l’organisation de projet et la gestion des différentes tâches.

la méthode utilisée ici pour la gestion du projet est Agile Scrum.Cela permet

une gestion réactive,structurée et interactive des différents projets de l’entreprise .

**1.2 Méthode Scrum qu’est ce que c’est?**

la méthode Scrum définit un cadre de travail permettant la réalisation de projets

complexes.Initialement prévu pour le développement du projet type « Software »

Cette méthode peut être appliquée à tout type de projet, du plus simple au plus innovant, et ce de manière très simple.

Les projets qui suivent la méthode Agile « SCRUM » sont divisés en plusieurs cycles de travail relativement court que l’on appelle « sprints ». Ces derniers permettent aux membres de l’équipe d’évaluer régulièrement les progrès liés au projet et de planifier les prochaines étapes de développement. Mais cela permet surtout de réajuster ou réorienter la direction prise par le projet si besoin est, à partir d’une base de travail déjà achevée et validée (sprint).

Chaque projet utilisant la méthode SCRUM s’organise autour d’une équipe auto-organisée car il n’y a pas de chef d’équipe qui décide des rôles de chacun, ou de la manière dont un problème est résolu, puisque ces problématiques sont traitées par l’équipe dans son ensemble ; et multifonctionnelle car chaque membre de l’équipe est parti prenant dans le développement de chaque fonctionnalité, de l’idée à l’implémentation finale.



La méthodologie SCRUM est composée de quatre phases (on parle aussi de réunion):

**• Planification du Sprint**

**•Revue de Sprint**

**•Rétrospective de Sprint**

**•Mêlée quotidienne**

La planification du sprint correspond au listing des points prioritaires que l'équipe pense pouvoir réaliser au cours d'un sprint.

La revue du sprint a lieu en fin de sprint, l'équipe de développement présente les fonctionnalités terminées au cours du sprint et recueille les retours du représentant des utilisateurs finaux, c'est aussi à ce moment que la mise en place des prochains sprints peut être anticipée.

La Rétrospective de Sprint permet de faire un point sur le sprint en lui-même (productivité, efficacité, qualité…) afin de pouvoir s'améliorer pour les prochains sprints.

Enfin la mêlée quotidienne permet de faire un point sur les avancements de chacun, elle est courte et chacun répond à trois questions principales : Qu'est-ce que j'ai terminé depuis la dernière mêlée ? Qu'est-ce que j'aurai terminé d'ici la prochaine mêlée ? Quels obstacles me retardent ?

Il existe trois rôles principaux à « pourvoir » : le responsable produit, le SCRUM Master, et le membre de l’équipe.

**• Le responsable produit** : Ce dernier est chargé de communiquer la vision globale du produit à l’équipe. Il se doit de représenter le client final, se mettre à sa place et donc de prioriser ses besoins. Il est celui qui a le plus de responsabilités et d’autorité. Le responsable produit est en effet celui qui est en première ligne lorsque quelque chose se passe mal.

**• Le SCRUM Master** : Il ne manage pas l’équipe, mais travaille à éliminer tous les obstacles qui peuvent empêcher l’équipe d’atteindre les objectifs fixés pour chaque sprint de travail. En résumé, ce rôle permet à l’équipe de rester créative et productive, tout en veillant à ce que les réalisations soient visibles pour le responsable produit.

**• Membre de l’équipe** : Dans la méthode SCRUM, l’équipe est responsable de la réalisation opérationnelle des « travaux ». C’est toute l’équipe qui est responsable du résultat final de chaque sprint.La manière dont sont exécutées les tâches est très libre mais cette liberté doit être néanmoins cadrée par l’obligation de répondre aux objectifs du sprint.

**5.2 Choix méthodologique de conception :**

On a utilisé dans ce projet comme langage de modélisation l’UML.

UML ( Unified Modeling Language ou « langage de modélisation unifié ») est un langage de modélisation graphique, il est apparu dans le monde du génie logiciel, dans le cadre de la « conception orientée objet ».

Il représente plusieurs avantages à citer :

⦁ UML est basée sur les mécanismes d’abstraction, de l’hiérarchie et de décomposition.

⦁ UML est un support de communication performant : il permet grâce à sa représentation graphique, d'exprimer visuellement une solution objet, de faciliter la comparaison et l'évolution de solution

⦁ UML est un langage formel et normalisé, il permet d'accéder à des informations précises et de garantir la stabilité.

⦁ UML facilite la compréhension de représentations abstraites complexes.

⦁ UML est basée sur la mise en place de diagrammes représentant les vues statiques et dynamiques dans le développement d’une application informatique.

Pour mieux concevoir notre application, on a besoin d’utiliser diagramme des cas d’utilisation qui permet de spécifier les fonctionnalités du notre système et leurs interactions avec l’utilisateur; aussi on a utilisé le diagramme des classes pour présenter les différentes classes et les interfaces du système ainsi que les relations entre elles et en troisième lieu on a la nécessité d’utiliser le diagramme des séquences et cela pour expliquer en détail la façon dont les opérations sont effectuées selon un ordre chronologique.

**Chapitre 2**

**Introduction :**

L'étape d'analyse et de spécification des besoins joue un rôle important dans les processus suivants: Le développement des systèmes d'information. Cette étape permet de mieux clarifier les besoins pour mieux comprendre la fonctionnalité et la non-fonctionnalité du système Du sujet.

**Présentations des acteurs :**

Un acteur est un personnage, un matériel ou un logiciel qui interagit avec le système dans le but de réaliser une ou plusieurs fonctions concernant les cas d’utilisations.

En ce qui suit, une identification des acteurs principaux qui interagissent directement sur notre application.

En ce qui suit, une identification des acteurs principaux qui interagissent directement sur notre application.

Il existe 3 types d'acteurs avec accès protégé par un login et un mot de passe :

⦁L’administration : C'est un utilisateur puissant, responsable de l'administration de la plateforme.

⦁Le formateur : C'est un usager avec pouvoir restreint, il anime des cours et il fait des publications dans la plateforme.

⦁L’ etudiant: Il se joint à un cour et il consulte les actualités de la plateforme.

**Identifications des fonctionnalités des acteurs :**

**Manquantes**

## 1.3 Spécification et analyse des besoins

### Besoin fonctionnelle :

Les services proposés par notre site se résument en sept actions majeures :

1. **Module d’authentification :**

Ce système permet à l'utilisateur de s’authentifier à travers login et mot de passe reçu par l’admin

1. **Module live screen share :**

Ce système permet au formateur de lancer des séances de cours en ligne et faire l’interaction entre les participants du live(formateur et formés)

1. **Module Live chat :**

Ce système permet de créer des discussions instantané entre

formateur et étudiants.

1. **Module administration :**

Ce système permet aux administrateurs la vérification des étudiants qui viennent de faire leurs inscriptions.

Ce système permet aux administrateurs d'approuver les séances de rattrapage qui sont planifiées par les formateurs.

Le système permet aux administrateurs d'approuver le calendrier des réunions

en ligne organisées par les formateurs.

1. **Module gestion publications :**

Ce système permet aux étudiants et aux professeurs la création , la modification et la suppression des publications.

1. **Module notification :**

Ce système permet aux étudiants et aux professeurs de recevoir les

notifications lorsqu’ils se connectent à leurs comptes.

1. **Module préparation de cours :**

Ce système permet aux professeurs de préparer les

cours et les envoyer aux étudiants.

**Module préparation des Homework :**

Ce système permet aux professeurs de préparer les

Homework et les envoyer aux étudiants.

1. **Module dashboard :**

Ce système permet aux étudiants de consulter leurs coordonnés, de consulter les publications enregistrées par lui et de consulter les séances de cours et de rattrapage à venir.

### Les Besoins non fonctionnels :

Les besoins non fonctionnels décrivent toutes les contraintes auxquelles est soumis le système pour sa réalisation et son bon fonctionnement :

**1. Ergonomie et souplesse**

L'application doit offrir une interface conviviale et ergonomique exploitable par l'utilisateur en envisageant toutes les interactions possibles à l'écran du support tenu.

**2. Rapidité**

L'application doit optimiser les traitements pour avoir un temps de génération de schéma raisonnable.

**3. Efficacité**

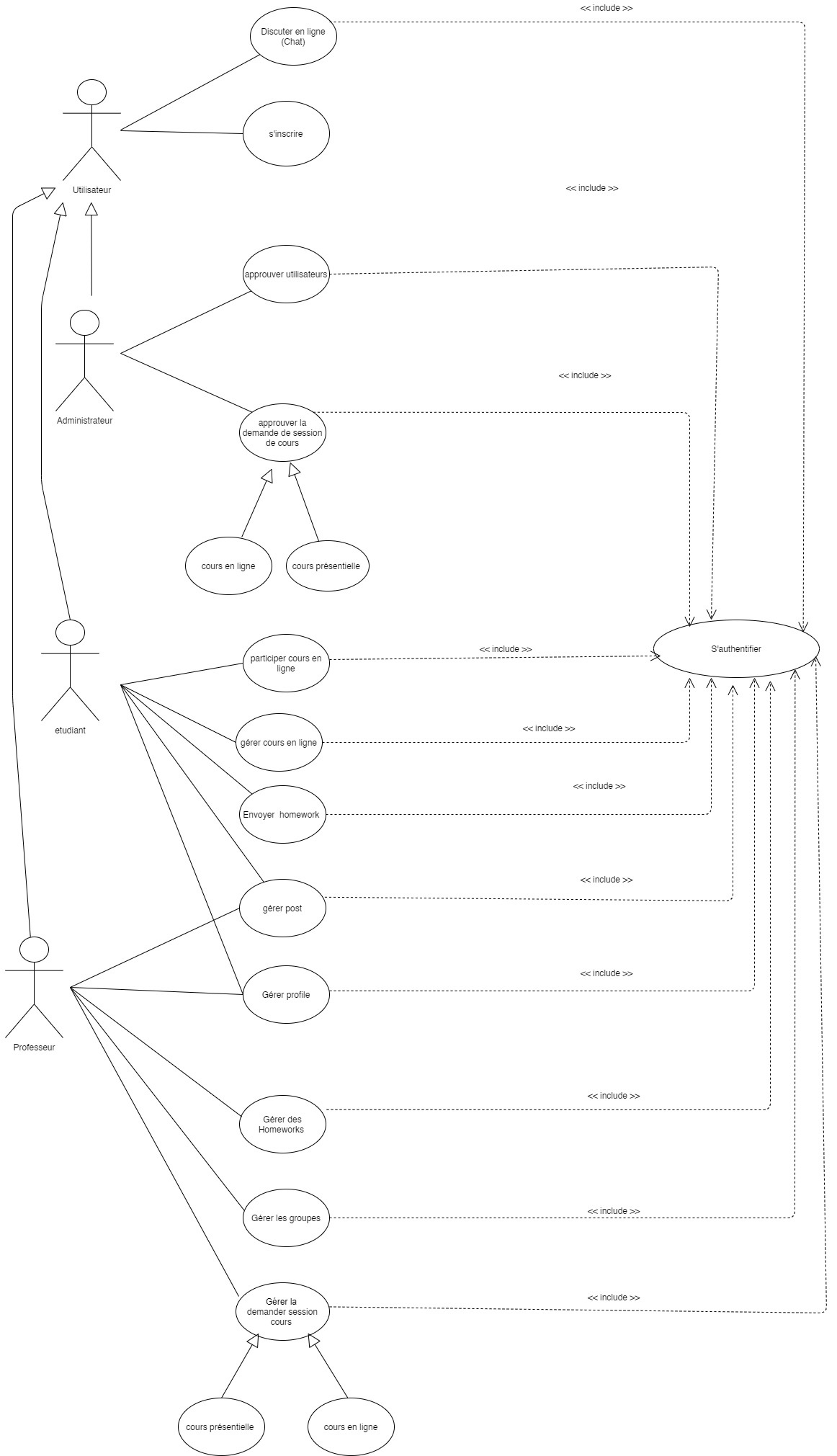
L'application doit être fonctionnelle indépendamment de toutes circonstances pouvant entourer l'utilisateur.

**4. Maintenabilité et scalabilité**

Le code de l'application doit être lisible et compréhensible afin d'assurer son état évolutif et extensible par rapport aux besoins du marché.

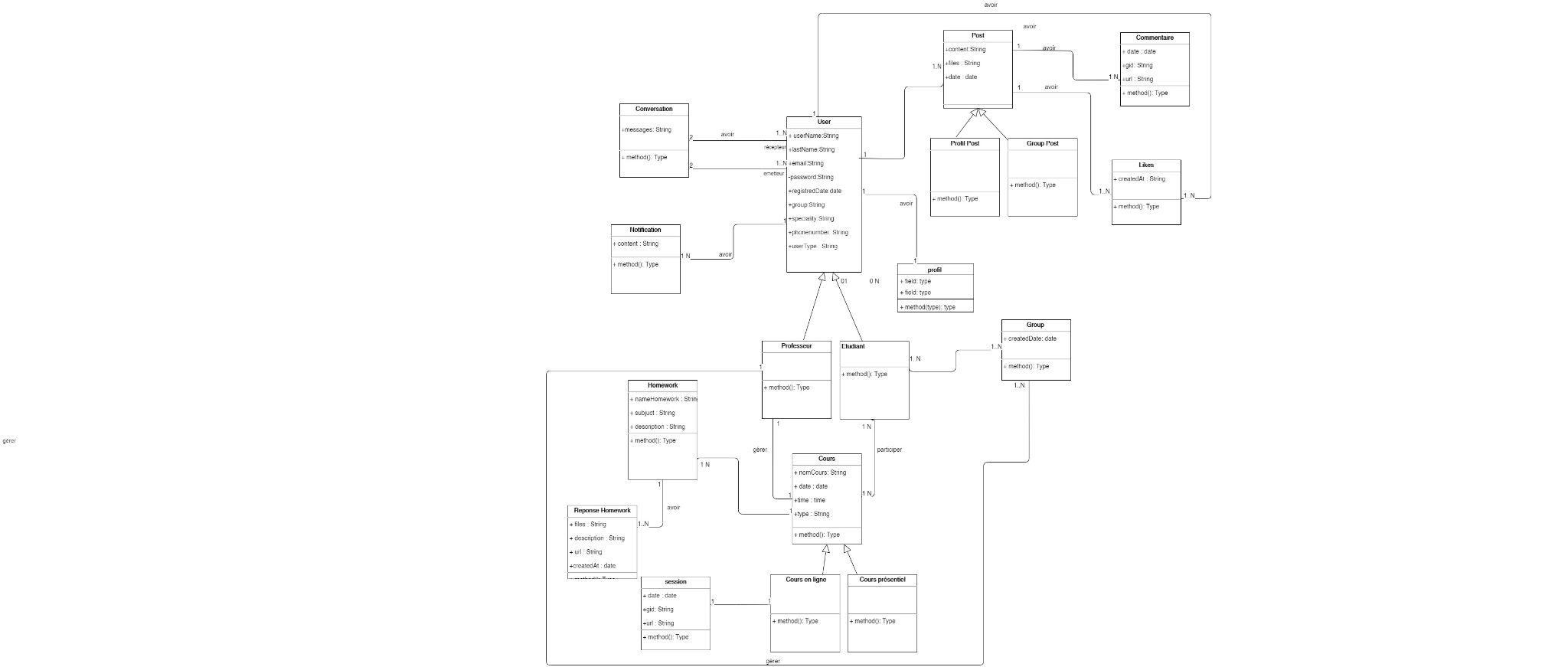
**5.1 Diagramme de cas d’utilisation globale :**

La figure ci-dessous représente les différents cas d’utilisation de notre système :



***figure 1 : Diagramme de cas d’utilisations global***

***\****



***figure 1 : Diagramme de cas d’utilisations global***

3 **Pilotage du projet avec Scrum**

3.1 L’équipe de projet :

**3.2 Backlog du produit :**

En bref, le backlog Scrum est conçu pour collecter tous les besoins des clients auxquels l'équipe projet doit répondre. Il contient donc la liste des fonctions impliquées dans la composition du produit, ainsi que tous les éléments qui nécessitent l'intervention de l'équipe projet. Tous les éléments contenus dans le backlog Scrum sont triés par priorité, indiquant leur ordre de réalisation.

La composition du backlog Scrum commence par la spécification et la définition des objectifs du produit, des utilisateurs cibles et des participants au projet. Ensuite, une liste d'exigences sera créée. Ils peuvent être fonctionnels ou non fonctionnels. Chaque exigence sera analysée par l'équipe de développement, puis le coût de sa mise en œuvre sera évalué au cours du plan Poker Scrum.

Les étapes de l’élaboration du product backlog présentées ici permettent de répondre aux 3 questions suivantes :

* *Quelles seront les fonctionnalités à créer ?*
* *Dans quel ordre devront-elles être livrées ?*
* *À qui sont-elles destinées ?*

**Objectif du backlog :**

Le backlog de produit est la liste des fonctionnalités attendues d’un produit. Plus exactement, au-delà de cet aspect fonctionnel, il contient tous les éléments qui vont nécessiter du travail pour l’équipe. Les éléments y sont classés par priorité ce qui permet de définir l’ordre de réalisation.

### **Erreurs à éviter :**

* ne pas avoir de backlog de produit.
* avoir plusieurs backlogs pour un seul produit (ou d’autres sources parasites, comme un bugtracker).
* ne pas partager le backlog avec toute l’équipe.
* ne pas faire vivre le backlog pendant le projet.
* confondre avec le backlog de sprint en y mettant des tâches.
* avoir plus de 150 éléments à faire dans le backlog.

**3.3 Planification des sprints :**

Après avoir expliqué les principes d’un backlog, nous présentons notre backlog « Comunik Marketing » :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | User Story | Priorité | Complexité | Estimation |
| Authentification | | | | |
| 1 | En tant qu’utilisateur, je veux m’authentifier en tapant une adresse email et un mot de passe pour accéder au tableau de bord. | 1 | Facile | 4 |
| 2 | En tant qu’utilisateur, je veux enregistrer mes données de connexion. | 2 | Facile | 5 |
| Inscription | | | | |
|  | En tant qu' utilisateur, je remplis les champs de formulaire demandés par la plateforme. | 1 | Facile | 5 |
|  | En tant qu' utilisateur je reçois un email de confirmation . | 3 | Facile | 3 |
|  | En tant qu' utilisateur je veux changer mon mot de passe . | 2 | Moyenne | 6 |
| Approuver Utilisateur | | | | |
|  | En tant qu' utilisateur ( étudiant ou professeur), je peux accéder au platforme | 3 | Moyenne | 10 |
|  | En tant qu' utilisateur (Administrateur) je peux vérifier les informations introduite par les utilisateurs qui sont inscrits | 1 | Moyenne | 10 |
|  | En tant qu’ utilisateur (Administrateur) je peux valider ou rejeter les utilisateurs qui sont inscrits | 2 | Facile | 4 |
|  | En tant qu’utilisateur (Administrateur), je veux faire une recherche dans la liste des utilisateurs inscrits. | 1 | Facile | 5 |
|  | En tant qu' utilisateur (Administrateur), j'envoie un email de confirmation. | 2 | Facile | 6 |
| Discuter en ligne( Chat) | | | | |
|  | En tant qu' utilisateur, je fais la recherche de l' utilisateur concerné pour lui envoyer un message. | 1 | Facile | 4 |
|  | En tant qu' utilisateur, je veux avoir un listing des messages précédents dans la conversation. | 1 | Difficile | 20 |
|  | En tant qu' utilisateur, je peux avoir toutes mes conversations. | 3 | Moyenne | 7 |
|  | En tant que utilisateur, je peux accéder à une seul conversation dans une fenêtre en cliquant sur le nom d'utilisateur dans la pop-up de la conversation | 1 | Difficile | 20 |
|  | En tant qu' utilisateur je peux supprimer une conversation. | 2 | Facile | 10 |
|  | En tant qu’utilisateur je reçois les messages d’une façon instantané (real-time). | 1 | Difficile | 30 |
|  | En tant qu'utilisateur je peux avoir un listing des utilisateurs connectés. | 1 | Difficile | 30 |
|  | En tant qu’utilisateur je reçois une notification au cas d’un nouveau message. | 1 | Difficile | 15 |
| Gérer la demande du session de cours | | | | |
|  | En tant qu'utilisateur (professeur), je peux planifier des sessions du cours . | 1 | Facile | 10 |
|  | En tant qu'utilisateur(Professeur), je peux modifier la planification d’une session du cours. | 2 | Facile | 5 |
|  | En tant qu' utilisateur (professeur), je peux supprimer une session du cours. | 3 | Facile | 4 |
|  | En tant qu' Utilisateur (professeur), je peux avoir un listing de tous mes cours planifiés. | 1 | Moyenne | 4 |
|  | En tant qu' utilisateur (professeur), je peux voir si ma demande de la planification des sessions du cours est validée ou rejetée. | 1 | Moyenne | .3 |
| Approuver une session de cours | | | | |
|  | En tant qu' utilisateur (Administrateur), je vérifie l'existence de la date et heure du cours dans la base de données en tapant sur << Vérifier >>. | 1 | Moyenne | 10 |
|  | En tant qu’utilisateur (Administrateur), je veux faire une recherche dans la liste des cours enregistrés. | 1 | Moyenne | 6 |
|  | En tant qu’utilisateur (Administrateur), je veux avoir un listing de tous les cours enregistrés. | 1 | Facile | 4 |
|  | En tant qu’utilisateur (Administrateur), je peux supprimer un ou tous les cours enregistrés. | 2 | Facile | 5 |
|  | En tant qu' utilisateur (Administrateur), j'envoie un email de confirmation. | 3 | Facile | 6 |
| Gérer Posts | | | | |
|  | En tant qu’utilisateur, je peux créer une publication . |  | Facile |  |
|  | En tant qu’utilisateur, je peux modifier une publication . |  | Facile |  |
|  | En tant qu’utilisateur, je peux supprimer une publication. |  | Facile |  |
|  | En tant qu’utilisateur, je peux ajouter ou retirer un like sur une publication . |  | Moyenne |  |
|  | En tant qu’utilisateur, je peux faire un partage à une publication dans mon profil. |  | Moyenne |  |
|  | En tant qu’utilisateur, je peux ajouter ou supprimer un commentaire |  | Moyenne |  |
| Gérer Profil | | | | |
|  | En tant qu’utilisateur, je peux consulter tous les profils |  | Facile |  |
|  | En tant qu’utilisateur, je peux modifier les informations de mon profil |  | Facile |  |
|  | En tant qu'utilisateur, je peux ajouter une photo de profil |  | Moyenne |  |
|  | En tant qu' utilisateur , j’ai le choix de rendre mon profil public ou privé . |  | Moyenne |  |
|  | En tant qu’utilisateur, je peux désactiver mon compte |  | Facile |  |
| Gérer Groupes | | | | |
|  | En tant qu' utilisateur (Professeur), je peux créer un groupe d' étudiants . |  | Facile |  |
|  | En tant qu’ utilisateur (Professeur) je peux ajouter des étudiants dans le groupe. |  | Moyenne |  |
|  | En tant qu’ utilisateur (Professeur) je peux supprimer des étudiants dans le groupe. |  | Facile |  |
|  | En tant qu’utilisateur (Professeur ou Étudiant), je peux consulter le groupe . |  | Moyenne |  |
|  | En tant qu' utilisateur (Professeur), je veux avoir un listing de tous les étudiants dans le groupe |  | Facile |  |
| Gérer des cours en ligne | | | | |
|  | En tant qu’utilisateur(Professeur) ,je peux organiser des séances en ligne. |  | Moyenne |  |
|  | En tant qu’utilisateur(Professeur), je peux supprimer une séance en ligne . |  | Facile |  |
|  | En tant qu'utilisateur(Professeur), je peux modifier une séance en ligne |  | Facile |  |
|  | En tant qu' utilisateur(Professeur), je dois gérer la séance de cours en mode broadcast. |  | Difficile |  |
|  | En tant qu' utilisateur(Professeur), je veux avoir un listing de tous les étudiant pour la séance. |  | Difficile |  |
|  | En tant qu’ utilisateur (Professeur ou Étudiant ) je peux faire un échange de message instantanément au cours de la séance. |  | Difficile |  |
| Gérer Homework | | | | |
|  | En tant qu’utilisateur (Professeur) , je peux créer des homeworks. |  | Moyenne |  |
|  | En tant qu’utilisateur (Professeur) , je peux envoyer des homeworks pour les étudiant concernées |  | Moyenne |  |
|  | En tant qu'utilisateur (Professeur) , je peux modifier un Homework |  | Facile |  |
|  | En tant qu' utilisateur (professeur),je peux supprimer des homeworks . |  | Facile |  |
|  | En tant qu’utilisateur (Professeur ) , je veux avoir un listing de tous mes homeworks. |  | Facile |  |
| Envoyer Homework | | | | |
|  | En tant qu' utilisateur (Étudiant), je peux envoyer le homework demandé . |  | Moyenne |  |
|  | En tant qu’ utilisateur (Étudiant), je peux vérifier si mon homework est envoyé ou non. |  | Moyenne |  |
|  | En tant qu’ utilisateur (Étudiant), je peux modifier mon homework déjà envoyé . |  | Facile |  |
|  | En tant qu’utilisateur ( Étudiant), je veux avoir un listing de tous mes homeworks |  | Facile |  |

**3.3 Planification des sprints :**

Afin d'atteindre notre objectif qui consiste à réaliser un réseau social académique et respect des besoins fonctionnels et non fonctionnels, nous avons décidé de diviser le projet en deux Release et chacun des deux Release contient trois sprints. Ci-dessous le plan Le travail détaillé à faire :

|  |
| --- |
| **Release 1** |

|  |  |  |
| --- | --- | --- |
| **Sprint 1** *2 Semaines* |  |  |

|  |
| --- |
| **Authentification** |

|  |
| --- |
| **s’inscire** |

|  |
| --- |
| **Approuver Utilisateur** |

|  |  |  |
| --- | --- | --- |
| **Sprint 2** *2 Semaines* |  |  |

|  |
| --- |
| **Discuter en ligne (Chat)** |

|  |  |  |
| --- | --- | --- |
| **Sprint 3** *2 Semaines* |  |  |

|  |
| --- |
| **Gérer la demander session cours** |

|  |
| --- |
| **Approuver session de cours** |

|  |
| --- |
| **Release 2** |

|  |  |  |
| --- | --- | --- |
| **Sprint 1** *2 Semaines* |  |  |

|  |
| --- |
| **Gérer profil** |

|  |
| --- |
| **Gérer groupes** |

|  |
| --- |
| **Gérer posts** |

|  |  |  |
| --- | --- | --- |
| **Sprint 2** *2 Semaines* |  |  |

|  |
| --- |
| **Gérer des cours en ligne** |

|  |  |  |
| --- | --- | --- |
| **Sprint 3** *2 Semaines* |  |  |

|  |
| --- |
| **Gérer Homework** |

|  |
| --- |
| **Répondre Homework** |

**4 Environnement de travail**

**4.1 Environnement logiciel**

Les principaux outils de développements utilisés pour l’élaboration de notre application sont présentés comme suit :

**Git**

est un [logiciel de gestion de versions](https://fr.wikipedia.org/wiki/Logiciel_de_gestion_de_versions) [décentralisé](https://fr.wikipedia.org/wiki/Gestion_de_versions#Gestion_de_versions_d.C3.A9centralis.C3.A9e). C'est un [logiciel libre](https://fr.wikipedia.org/wiki/Logiciel_libre) créé par [Linus Torvalds](https://fr.wikipedia.org/wiki/Linus_Torvalds), auteur du [noyau Linux](https://fr.wikipedia.org/wiki/Noyau_Linux), et distribué selon les termes de la [licence publique générale GNU](https://fr.wikipedia.org/wiki/Licence_publique_g%C3%A9n%C3%A9rale_GNU) version 2.

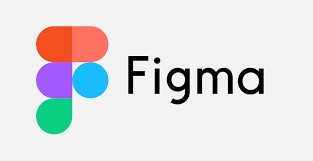
**GitHub** 

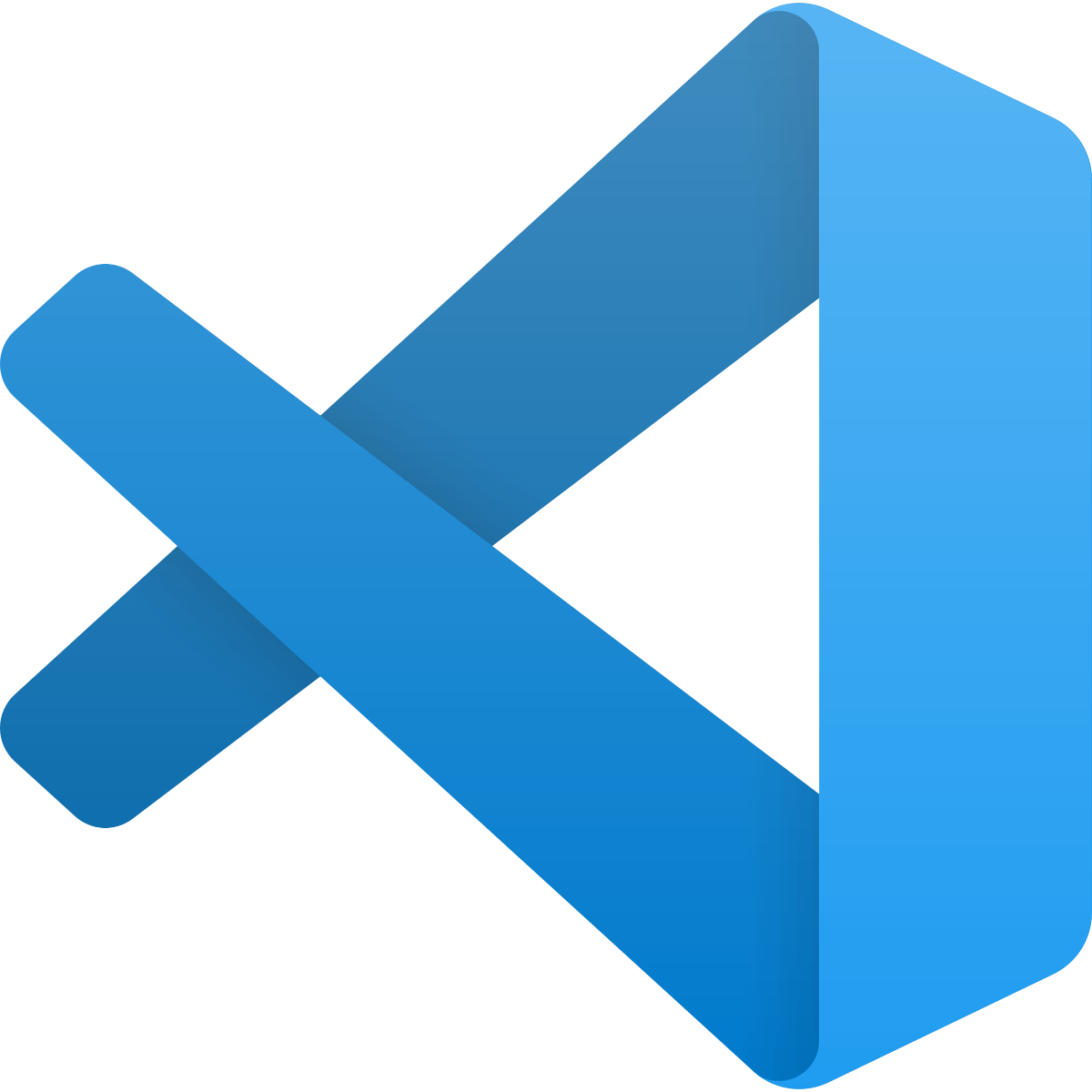
est un service web d'[hébergement](https://fr.wikipedia.org/wiki/H%C3%A9bergeur_web) et de gestion de développement de logiciels, utilisant le [logiciel de gestion de versions](https://fr.wikipedia.org/wiki/Logiciel_de_gestion_de_versions) [Git](https://fr.wikipedia.org/wiki/Git).

**Draw.Io**

Draw.io est une application gratuite en ligne, accessible via son navigateur (protocole https) qui permet de dessiner des diagrammes ou des organigrammes.

**Figma**

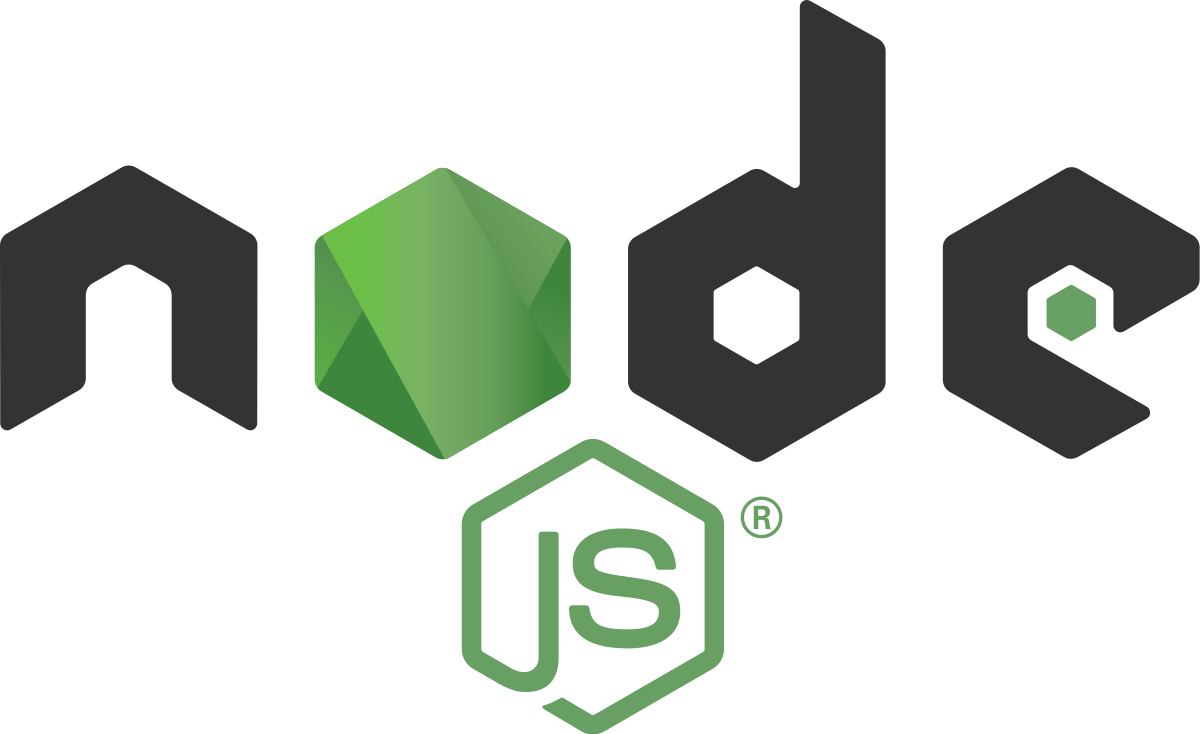
**Figma** est un outil de design collaboratif qui permet de prototyper les interfaces graphiques pour les UX/UI designers. Il aide notamment à concevoir sites web, applications, interfaces utilisateur.

**Visual code Studio**

Visual Studio Code est un [éditeur de code](https://fr.wikipedia.org/wiki/%C3%89diteur_de_texte) extensible développé par [Microsoft](https://fr.wikipedia.org/wiki/Microsoft) pour [Windows](https://fr.wikipedia.org/wiki/Microsoft_Windows), [Linux](https://fr.wikipedia.org/wiki/Linux) et [macOS 2.](https://fr.wikipedia.org/wiki/MacOS)

**4.2 Les langages et les techniques utilisés**

Les principaux langages de développements utilisés pour l’élaboration de notre application sont présentés comme suit :

**Node JS** 

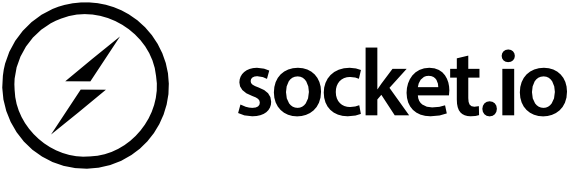
**Node.js** est une plateforme logicielle libre en JavaScript, orientée vers les applications réseau événementielles hautement concurrentes qui doivent pouvoir monter en charge.

**Express.js** est un framework pour construire des applications web basées sur Node.js . C'est de fait le framework standard pour le développement de serveur en Node.js.

**MongoDB** (de l'anglais humongous qui peut être traduit par « énorme ») est un système de gestion de base de données orienté documents, répartissable sur un nombre quelconque d'ordinateurs et ne nécessitant pas de schéma prédéfini des données.

**AngularJS** est un [framework](https://fr.wikipedia.org/wiki/Framework) [JavaScript](https://fr.wikipedia.org/wiki/Framework_JavaScript) [libre](https://fr.wikipedia.org/wiki/Logiciel_libre) et [open source](https://fr.wikipedia.org/wiki/Open_source) développé par Google. Il permet de développer des [pages web](https://fr.wikipedia.org/wiki/Page_web).

**WebRTC** (**Web Real-Time Communication**, littéralement « communication en temps réel pour le [*Web*](https://fr.wikipedia.org/wiki/World_Wide_Web) ») est une [interface de programmation JavaScript](https://fr.wikipedia.org/wiki/Interface_de_programmation) développée au sein du [W3C](https://fr.wikipedia.org/wiki/World_Wide_Web_Consortium) et de l'[IETF](https://fr.wikipedia.org/wiki/Internet_Engineering_Task_Force)..

**Socket.IO** est une bibliothèque [JavaScript](https://en.wikipedia.org/wiki/JavaScript) pour [les applications Web en](https://en.wikipedia.org/wiki/Web_application) temps réel . Elle permet une communication bidirectionnelle en temps réel entre les clients Web et les serveurs. 

**Angular Material** est une bibliothèque de composants d'interface utilisateur pour les développeurs Angular JS.

Le HyperText Markup Language, généralement abrégé HTML ou dans sa dernière version [HTML5](https://fr.wikipedia.org/wiki/HTML5), est le [langage de balisage](https://fr.wikipedia.org/wiki/Langage_de_balisage) conçu pour représenter les [pages web](https://fr.wikipedia.org/wiki/Page_web).

Les feuilles de style en cascade, généralement appelées CSS de l'[anglais](https://fr.wikipedia.org/wiki/Anglais) Cascading Style Sheets, forment un [langage informatique](https://fr.wikipedia.org/wiki/Langage_informatique) qui décrit la présentation des documents [HTML](https://fr.wikipedia.org/wiki/Hypertext_markup_language) et [XML](https://fr.wikipedia.org/wiki/Extensible_markup_language).

**Bootstrap** est un [framework](https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203355-framework/) développé par l'équipe du réseau social Twitter. Proposé en open source (sous licence MIT), ce framework utilisant les langages [HTML](https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203255-html-hypertext-markup-langage-definition-traduction/), CSS et JavaScript fournit aux développeurs des outils pour créer un site facilement.

**4.3 Environnement matériel**

Le développement de l’application est réalisé via deux ordinateurs portables ayant les caractéristiques suivantes :

|  |  |
| --- | --- |
| **Caractéristique** |  |
| **Marque** | HP |
| **Processeur** | Intel ® Core™ i7 CPU 8th Gen |
| **RAM** | 8 GO |
| **Disque dur** | 500 GO |
| **Système d’exploitation** | Microsoft Windows 10 Edition Intégrale 64 bits |

Chapitre 3

**Introduction**

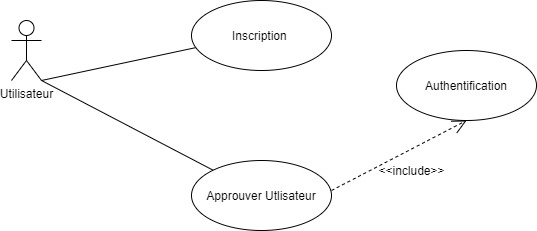
Après avoir morceler le projet dans le chapitre précédent, nous allons commencer par décortiquer les deux sprints du premier release. Pour ce faire, nous présenterons l’analyse fonctionnelle, la conception et implémentations des volets gestion des approuver utilisateurs et gestion des campagnes.

1. **Sprint 1 : Authentification**
2. **Backlog du Sprint 1:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | User Story | Priorité | Estimation |
|  | Authentification |  |  |
| 1 | En tant qu’utilisateur, je veux m’authentifier en tapant une adresse email et un mot de passe pour accéder au tableau de bord. | 1 | 4 |
| 2 | En tant qu’utilisateur, je veux enregistrer mes données de connexion. | 2 | 5 |
|  | Inscription |  |  |
| 4 | En tant qu' utilisateur, je remplis les champs de formulaire demandés par la plateforme. | 1 | 5 |
| 5 | En tant qu' utilisateur je reçois un email de confirmation . | 2 | 6 |
| Approuver Utilisateur | | | |
| 6 | En tant qu' utilisateur ( étudiant ou professeur), je peux accéder au platforme | 3 | 10 |
| 7 | En tant qu' utilisateur (Administrateur) je peux vérifier les informations introduite par les utilisateurs qui sont inscrits | 1 | 10 |
| 8 | En tant qu’ utilisateur (Administrateur) je peux valider ou rejeter les utilisateurs qui sont inscrits | 2 | 4 |
| 9 | En tant qu’utilisateur (Administrateur), je veux faire une recherche dans la liste des utilisateurs inscrits. | 1 | 5 |
| 10 | En tant qu’utilisateur (Administrateur), je veux avoir un listing utilisateurs inscrits. | 1 | 12 |
| 11 | En tant qu' utilisateur (Administrateur), j'envoie un email de confirmation. | 2 | 6 |

1. **Diagramme de cas d’utilisation global**

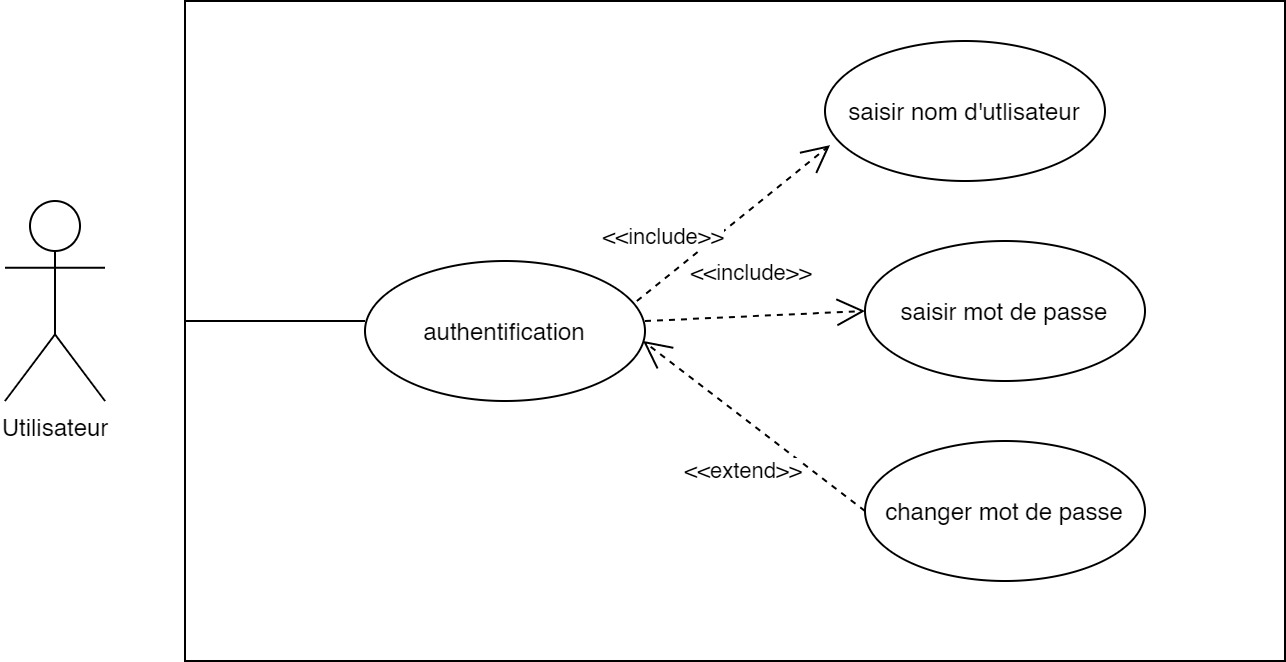
La figure suivante résume les fonctionnalités du premier sprint :

****

***figure 2 : Diagramme de cas d’utilisation global du sprint 1***

**C. Analyse des cas d’utilisation**

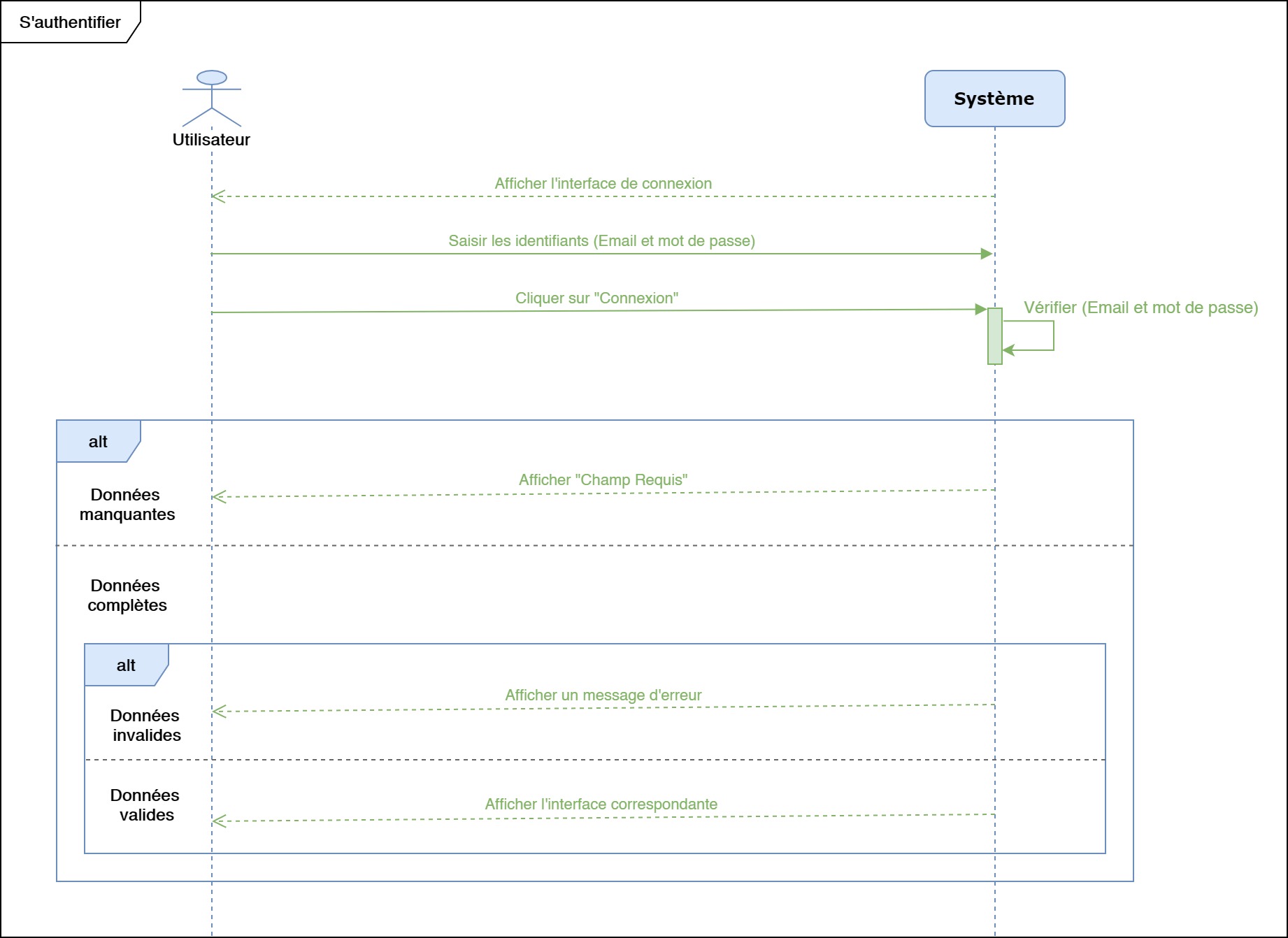
Cette activité consiste à faire ressortir les exigences fonctionnelles du premier sprint, nous avons adopté les diagrammes de cas d’utilisation raffinés, les descriptions textuelles et les diagrammes de séquence système pour cela.

**C.1 Analyse du cas d’utilisation « S’authentifier »**

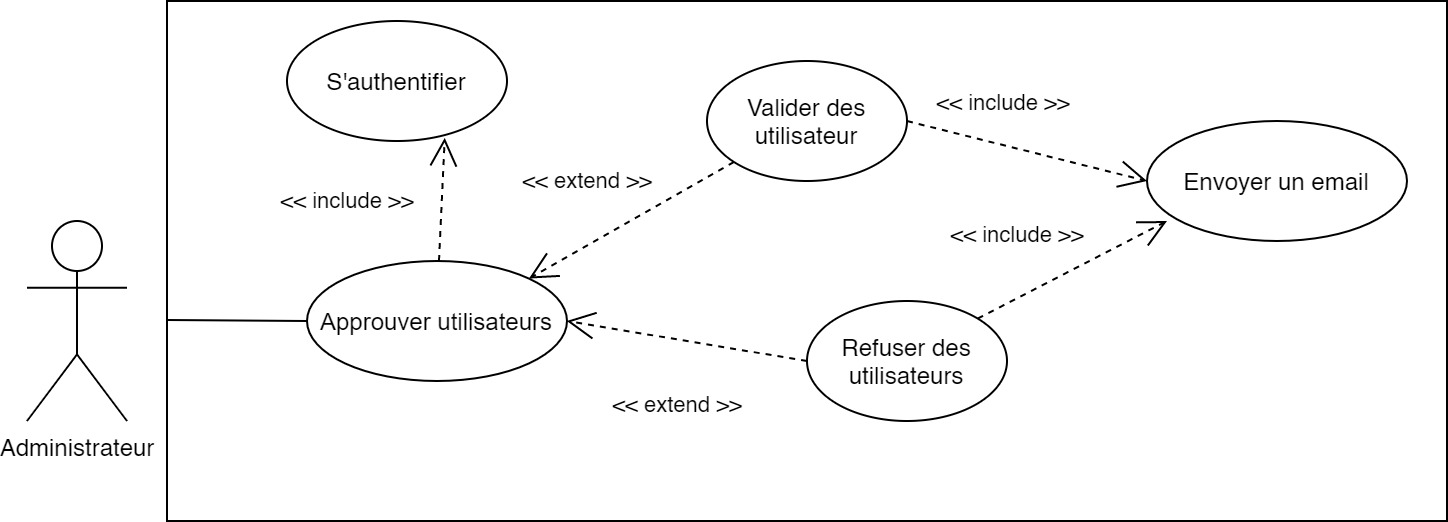
***figure 2 : Cas d'utilisation (authentification)***

-Description textuelle du cas d'utilisation "S’authentifier"

|  |  |
| --- | --- |
| **Cas d’utilisation** | CU1: S’authentifier |
| **description brève** | Ce CU permet à l’acteur d’accéder à la plateforme |
| **Acteur** | Etudiant, administration et professeur |
| **Precondition** | L‘acteur doit être un membre identifié |
| **Enchaînement principal** | 1. Le système affiche le formulaire d’authentification à l'acteur. 2. L'acteur entre son nom et son mot de passe. 3. Le système fait la vérification des information entré par l’utilisateur 4. Le système dans l'accès au plateforme correspondant au profil. |
| **Post-condition** | L’utilisateur est connecté et accéder au tableau de bord. |
| **Enchaînement alternatif** | Le login ou le mot de passe est incorrect.Ce scénario commence au point 03 du l’Enchaînement principal.   1. Le système informe l'utilisateur que les données introduites sont invalides en affichant les messages d’erreur. 2. Retour à l'étape 1 de l’enchaînement principal. |

- Diagramme de séquence système du cas d'utilisation "S'authentifier" 

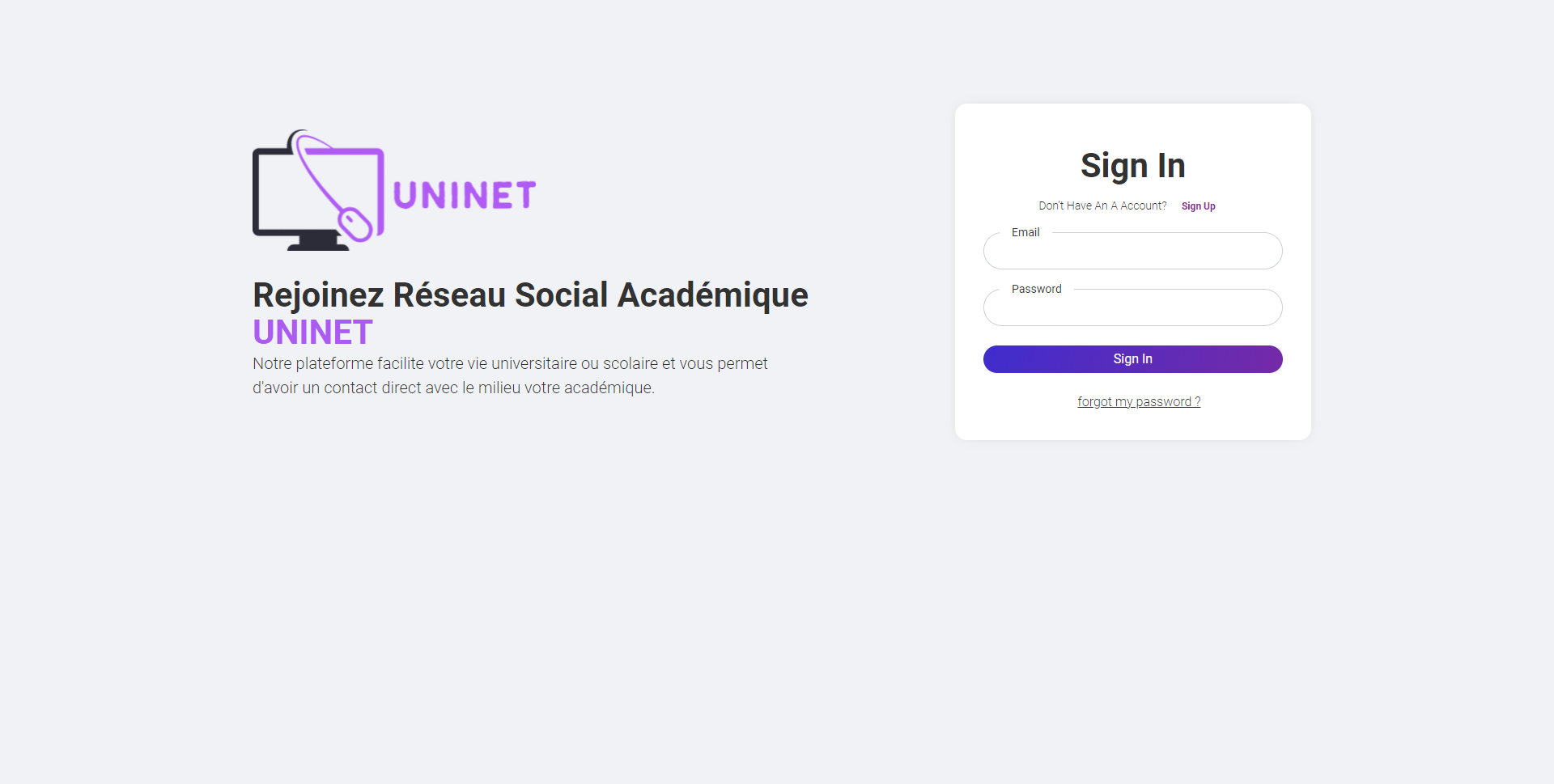
**C.2 Analyse du cas d’utilisation « Approuver Utilisateur »**

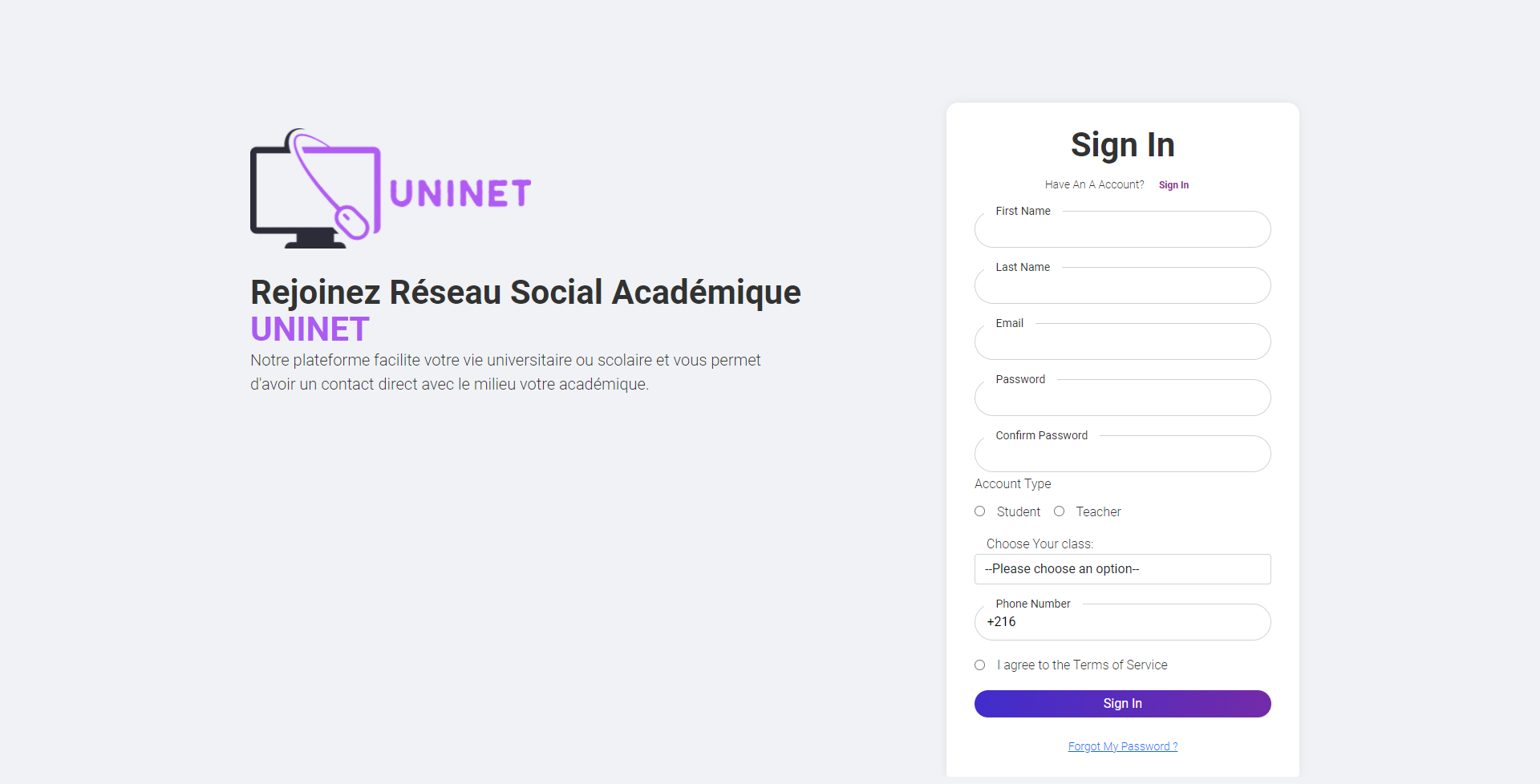


***figure 5 : Cas d’utilisation (Approuver utilisateurs)***

- Description textuelle du cas d'utilisation "Approuver utilisateurs"

|  |  |
| --- | --- |
| **Cas d’utilisation** | Approuver utilisateurs |
| **description brève** | Ce CU2 permet à l'administrateur d'approuver les Utilisateurs inscrit dans le système |
| **Acteur** | administrateur |
| **Precondition** | L‘acteur doit être connecté et un administrateur. |
| **Enchaînement principal** | 1. Le système affiche la liste des étudiants inscrits dans la base de données. 2. l’acteur fait la vérification des informations affichées par le système. 3. l’acteur approuve les informations affichées par le système. 4. le système notifie l’acteur concerné du bon déroulement de la validation du ses informations |
| **Post-condition** |  |
| **Enchaînement alternatif** | Les informations sont déjà enregistrées dans la base. Ce scénario commence au point 02 du enchaînement principal.   1. Le système informe l’acteur que la séance du cours, la date et l’heure du session sont déjà pris par un autre professeur. |



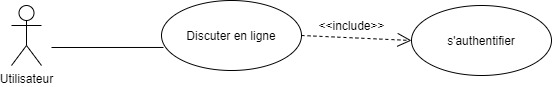


1. **Sprint 2 : Discuter en ligne (Chat)**
2. **Backlog du Sprint :**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | User Story | Priorité | Estimation |
|  | Discuter en ligne( Chat) |  |  |
| 1 | En tant qu' utilisateur, je fais la recherche de l' utilisateur concerné pour lui envoyer un message. | 1 | 4 |
| 2 | En tant qu' utilisateur, je veux avoir un listing des messages précédents dans la conversation. | 1 | 20 |
| 4 | En tant qu' utilisateur, je peux avoir toutes mes conversations. | 3 | 7 |
| 5 | En tant que utilisateur, je peux accéder à une seul conversation dans une fenêtre en cliquant sur le nom d'utilisateur dans la pop-up de la conversation | 1 | 20 |
| 6 | En tant qu' utilisateur je peux supprimer une conversation. | 2 | 10 |
| 7 | En tant qu’utilisateur je reçois les messages d’une façon instantané (real-time). | 1 | 30 |
| 8 | En tant qu'utilisateur je peux avoir un listing des utilisateurs connectés. | 1 | 30 |
| 9 | En tant qu’utilisateur je reçois une notification au cas d’un nouveau message. | 1 | 15 |

1. **Diagramme de cas d’utilisation global**

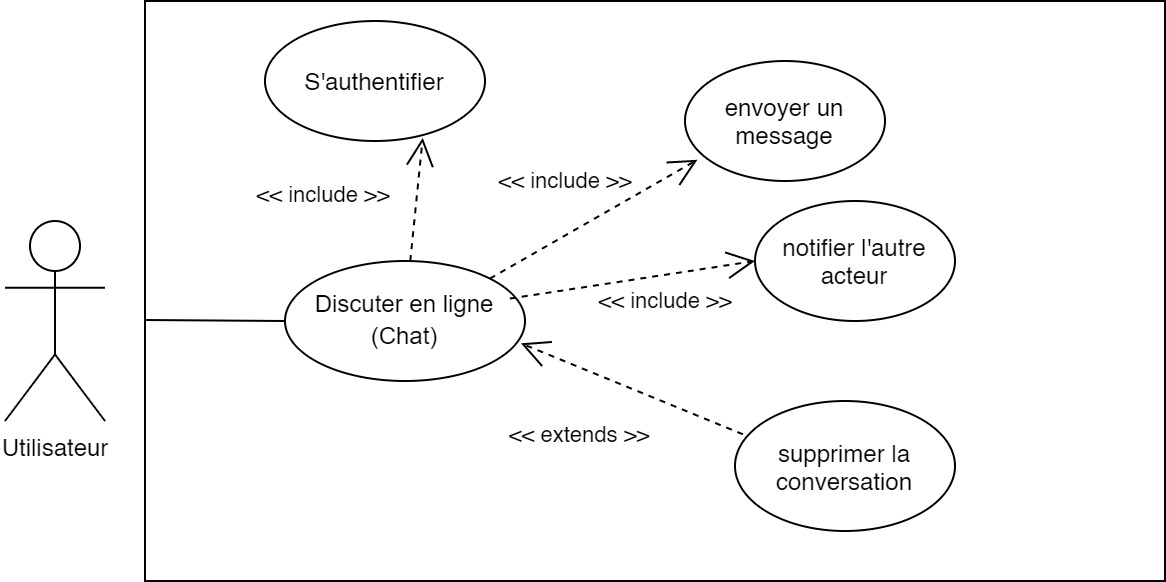
La figure suivante résume les fonctionnalités du deuxième sprint :

****

***figure 3 : Diagramme de cas d’utilisation global du sprint 2***

**c. Analyse des cas d’utilisation**

Cette activité consiste à faire ressortir les exigences fonctionnelles du premier sprint, nous avons adopté les diagrammes de cas d’utilisation raffinés, les descriptions textuelles et les diagrammes de séquence système pour cela.

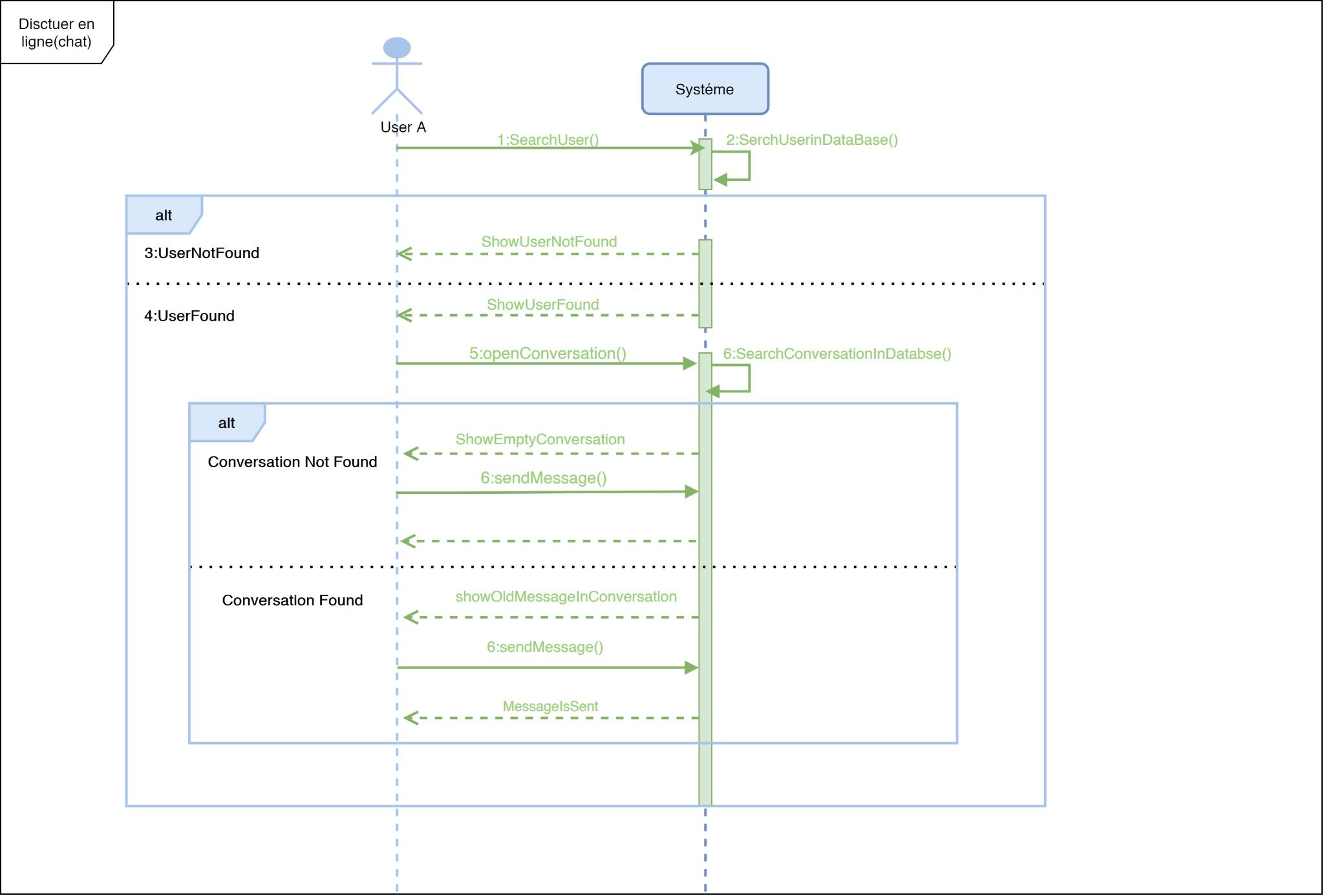


***figure 4 : Cas d'utilisation (Discuter en ligne)***

- Description textuelle du cas d'utilisation "Discuter en ligne (Chat)"

|  |  |
| --- | --- |
| **Cas d’utilisation** | Discuter en ligne (Chat) |
| **description brève** | Ce CU qui permet aux utilisateurs de créer des sessions de discussion instantané. |
| **Acteur** | Etudiant, administration et professeur |
| **Precondition** | L‘acteur doit être connecté. |
| **Enchaînement principal** | 1. Accéder au acteur concernée pour lui adresser un message 2. ecrire le message et l’envoyer 3. Le système vérifie l'existence d’une conversation entre les deux acteurs. 4. le système enregistre les message dans la base de données 5. le système affiche le message pour l’autre acteur instantanément. 6. Le système notifie l’autre acteur de la réception d’un nouveau message. |
| **Post-condition** |  |
| **Enchaînement alternatif** | Les informations sont déjà enregistrées dans la base. Ce scénario commence au point 02 du enchaînement principal.   1. Le système informe l’acteur que la séance du cours, la date et l’heure du session sont déjà pris par un autre professeur. |

- Diagramme de séquence système du cas d'utilisation "Discuter en ligne (Chat)"



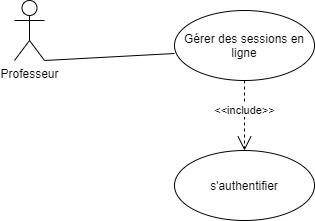
**3. Sprint 3 : Gérer la demande du session de cours et approuver cours**

1. **Backlog du Sprint :**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | User Story | Priorité | Estimation |
|  | Gérer la demande du session de cours |  |  |
| 1 | En tant qu'utilisateur (professeur), je peux planifier des sessions du cours . | 1 | 10 |
| 2 | En tant qu'utilisateur(Professeur), je peux modifier la planification d’une session du cours. | 2 | 5 |
| 3 | En tant qu' utilisateur (professeur), je peux supprimer une session du cours. | 3 | 4 |
| 4 | En tant qu' Utilisateur (professeur), je peux avoir un listing de tous mes cours planifiés. | 1 | 4 |
| 5 | En tant qu' utilisateur (professeur), je peux voir si ma demande de la planification des sessions du cours est validée ou rejetée. | 1 | 3 |
|  | Approuver une session de cours |  |  |
| 6 | En tant qu' utilisateur (Administrateur), je vérifie l'existence de la date et heure du cours dans la base de données en tapant sur << Vérifier >>. | 1 | 10 |
| 7 | En tant qu’utilisateur (Administrateur), je veux faire une recherche dans la liste des cours enregistrés. | 1 | 6 |
| 8 | En tant qu’utilisateur (Administrateur), je veux avoir un listing de tous les cours enregistrés. | 1 | 4 |
| 9 | En tant qu’utilisateur (Administrateur), je peux supprimer un ou tous les cours enregistrés. | 2 | 5 |
| 10 | En tant qu' utilisateur (Administrateur), j'envoie un email de confirmation. | 3 | 6 |

1. **Diagramme de cas d’utilisation global :**

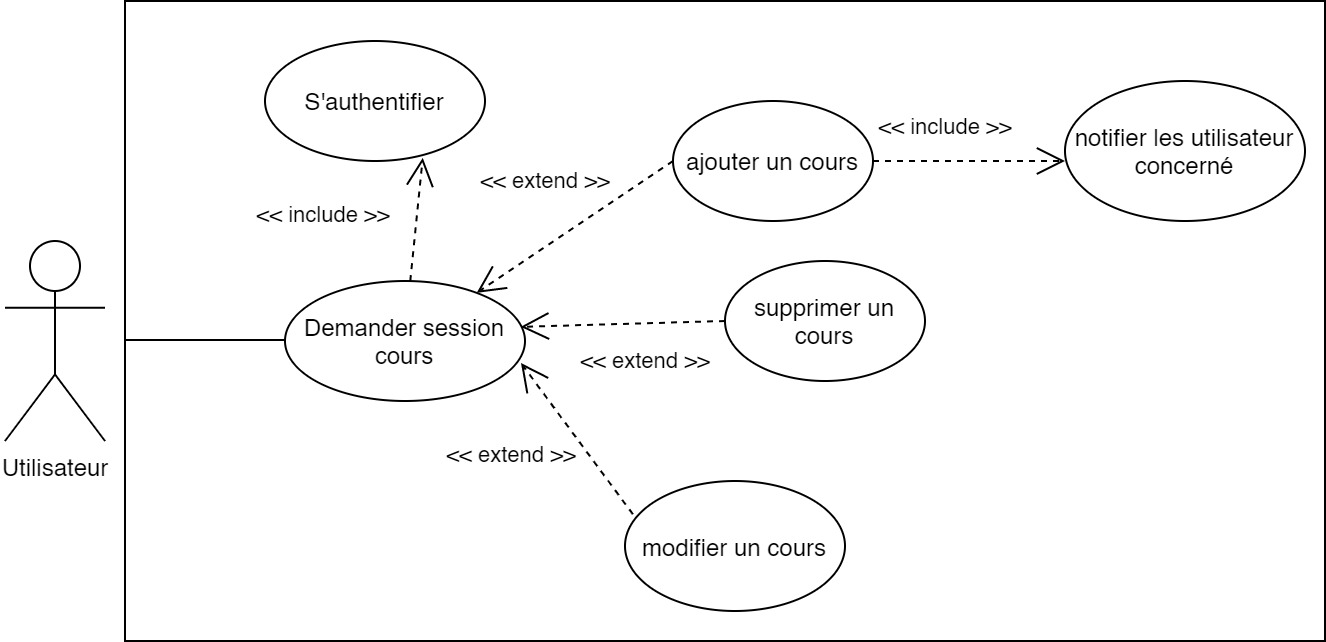
La figure suivante résume les fonctionnalités du troisième sprint :

****

***figure 4 : Diagramme de cas d’utilisation global du sprint 3***

1. **analyse des cas d’utilisation :**

* **Cas d’utilisation « Demander session cours »**

***figure 3 : Cas d'utilisation ( Demander session cours)***

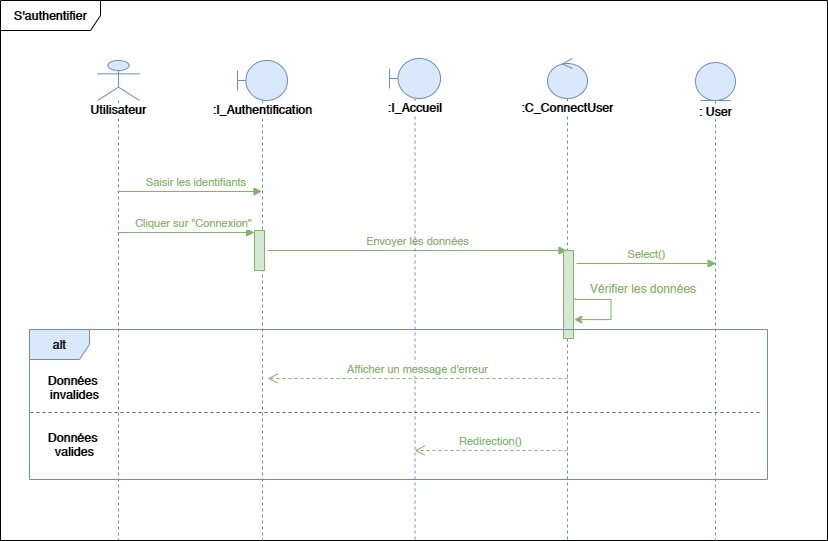
|  |  |
| --- | --- |
| **Cas d’utilisation** | Demander session cours en ligne |
| **description brève** | Ce CU1 permet à l'administrateur de demander des session de cours en ligne ou présentielle |
| **Acteur** | professeur |
| **Precondition** | L‘acteur doit être connecté. |
| **Enchaînement principal** | 1. Le système affiche un formulaire pour l’acteur.. 2. l’acteur saisit les informations relatives à la session de cours qu’il veut la demander. 3. <Méthode 1 : cours en ligne>   l’acteur choisit de donner un cours en ligne.  <Méthode 2 : cours en présentielle>  l’acteur choisit de donner un cours présentiel.   1. Le système enregistre les informations dans la base de données 2. Le système notifie l’acteur concerné du bon déroulement de l’enregistrement du cours. |
| **Post-condition** |  |
| **Enchaînement alternatif** | Les informations sont déjà enregistrées dans la base. Ce scénario commence au point 02 du enchaînement principal.   1. Le système informe l’acteur que la séance du cours, la date et l’heure du session sont déjà pris par un autre professeur. |

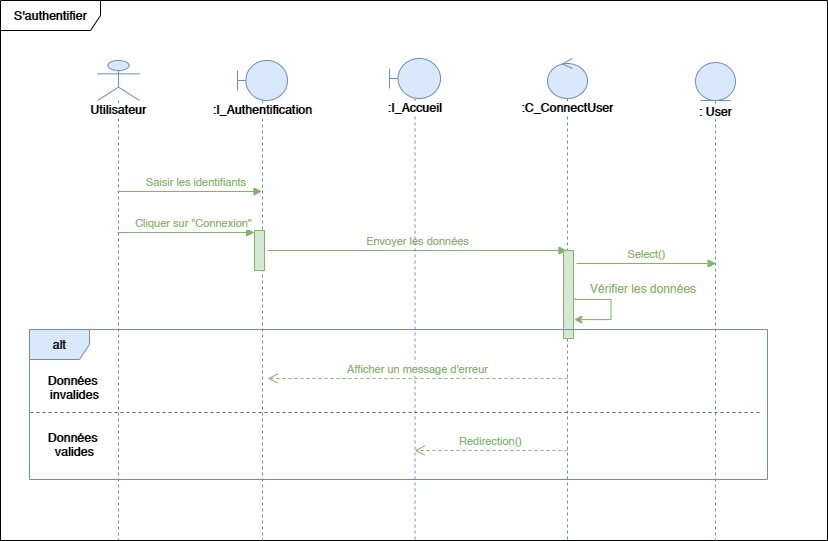
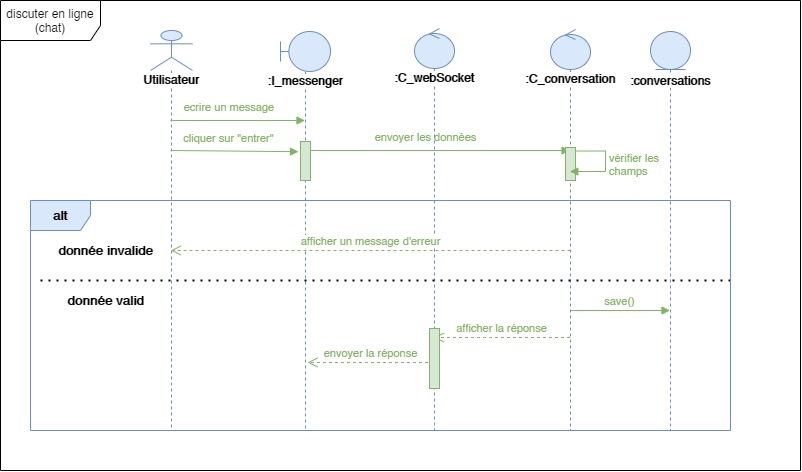
* **Cas d’utilisation « approuver la demande de session de cours »**

**diagramme ne9es**

|  |  |
| --- | --- |
| **Cas d’utilisation** | approuver la demande de session de cours |
| **description brève** | Ce CU1 permet à l'administrateur d'approuver les sessions de cours |
| **Acteur** | administrateur |
| **Precondition** | L‘acteur doit être connecté et un administrateur. |
| **Enchaînement principal** | 1. Le système affiche la liste des cours enregistrés par les professeurs. 2. l’acteur fait la vérification des informations affichées par le système. 3. l’acteur approuve les informations affichées par le système. 4. le système notifie l’acteur concerné du bon déroulement de la validation du cours |
| **Post-condition** |  |
| **Enchaînement alternatif** | Les informations sont déjà enregistrées dans la base. Ce scénario commence au point 02 du enchaînement principal.   1. Le système informe l’acteur que la séance du cours, la date et l’heure du session sont déjà pris par un autre professeur. |

**4. Diagrammes de séquence détaillé :**

***figure 4 :****Diagrammes de séquence détaillé* 

***figure 5 :****Diagrammes de séquence détaillé* 

**Release 2**

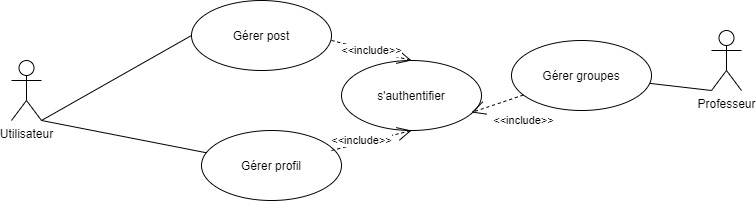
**Introduction**

Après avoir présenté le premier release, nous passons au deuxième qui évoque la partie phare de ce projet. Dans ce deuxième release on va travailler sur trois sprint.

1. **Sprint 1 : Gérer posts, profil et groupe :** 
   1. **Backlog du sprint**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | User Story | Priorité | Estimation |
|  | Gérer Posts |  |  |
| 1 | En tant qu’utilisateur, je peux créer une publication . | 1 | 2 |
| 2 | En tant qu’utilisateur, je peux modifier une publication . | 2 | 4 |
| 3 | En tant qu’utilisateur, je peux supprimer une publication. | 3 | 4 |
| 4 | En tant qu’utilisateur, je peux ajouter ou retirer un like sur une publication . | 2 | 10 |
| 5 | En tant qu’utilisateur, je peux faire un partage à une publication dans mon profil. | 2 | 20 |
| 6 | En tant qu’utilisateur, je peux ajouter ou supprimer un commentaire | 2 | 15 |
|  | Gérer Profil |  |  |
| 7 | En tant qu’utilisateur, je peux consulter tous les profils | 1 | 4 |
| 8 | En tant qu’utilisateur, je peux modifier les informations de mon profil | 2 | 5 |
| 9 | En tant qu'utilisateur, je peux ajouter une photo de profil | 1 | 2 |
| 10 | En tant qu' utilisateur , j’ai le choix de rendre mon profil public ou privé . | 3 | 5 |
| 11 | En tant qu’utilisateur, je peux désactiver mon compte | 3 | 10 |
|  | Gérer Groupes |  |  |
| 12 | En tant qu' utilisateur (Professeur), je peux créer un groupe d' étudiants . | 1 | 10 |
| 13 | En tant qu’ utilisateur (Professeur) je peux ajouter des étudiants dans le groupe. | 2 | 4 |
| 9 | En tant qu’ utilisateur (Professeur) je peux supprimer des étudiants dans le groupe. | 1 | 5 |
| 10 | En tant qu’utilisateur (Professeur ou Étudiant), je peux consulter le groupe . | 1 | 12 |
| 11 | En tant qu' utilisateur (Professeur), je veux avoir un listing de tous les étudiants dans le groupe | 2 | 6 |
|  | En tant qu’utilisateur(Professeur ou Étudiant) , je peux faire des post dans le groupe | 1 | 10 |
|  | En tant qu’utilisateur(Professeur), je peux supprimer le groupe | 3 | 3 |

* 1. **Diagramme de cas d’utilisation global**

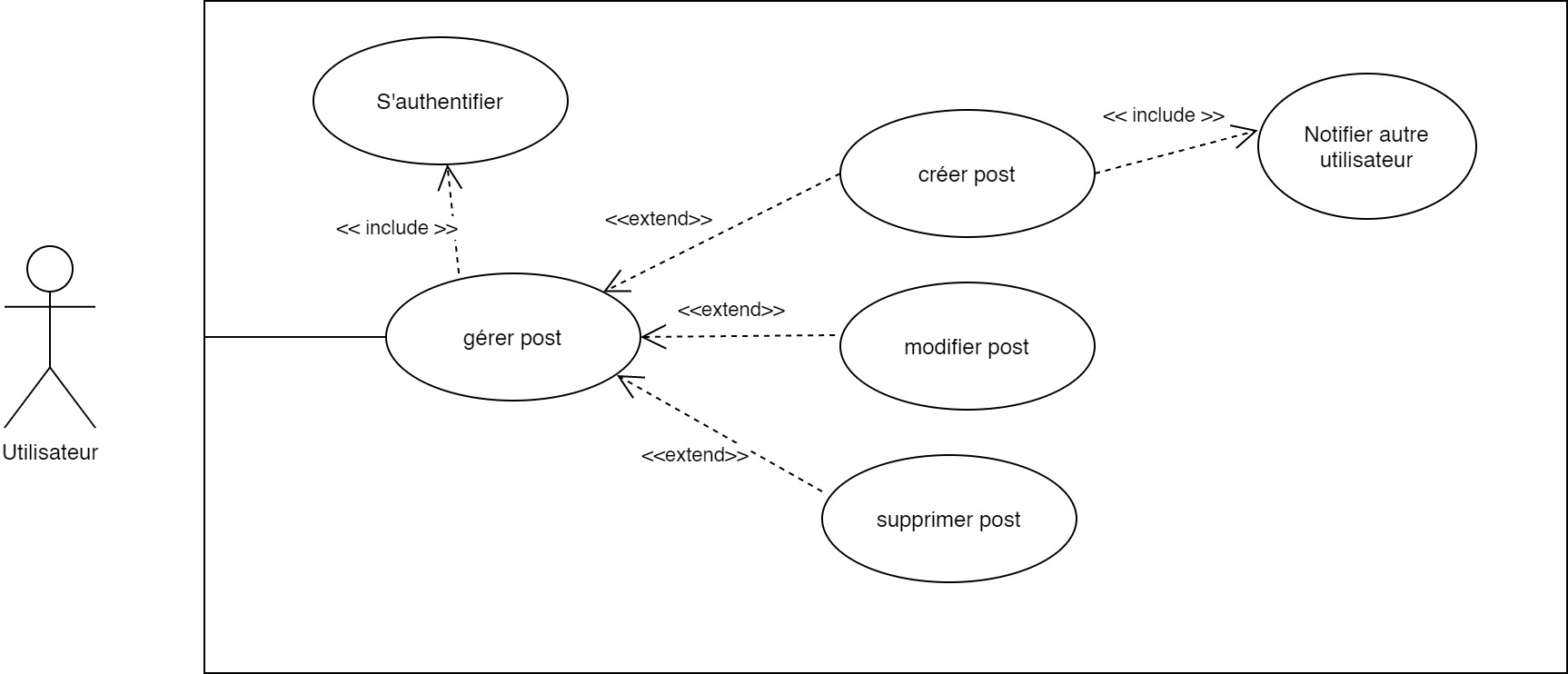
La figure suivante résume les fonctionnalités du premier sprint : 

***figure 1 : Diagramme de cas d’utilisation global du sprint 1 (Release 2)***

* 1. **Analyse fonctionnelle :**

**c.1 Analyse du cas d’utilisation « gérer Posts »**

* **Cas d’utilisation « gérer les posts »**

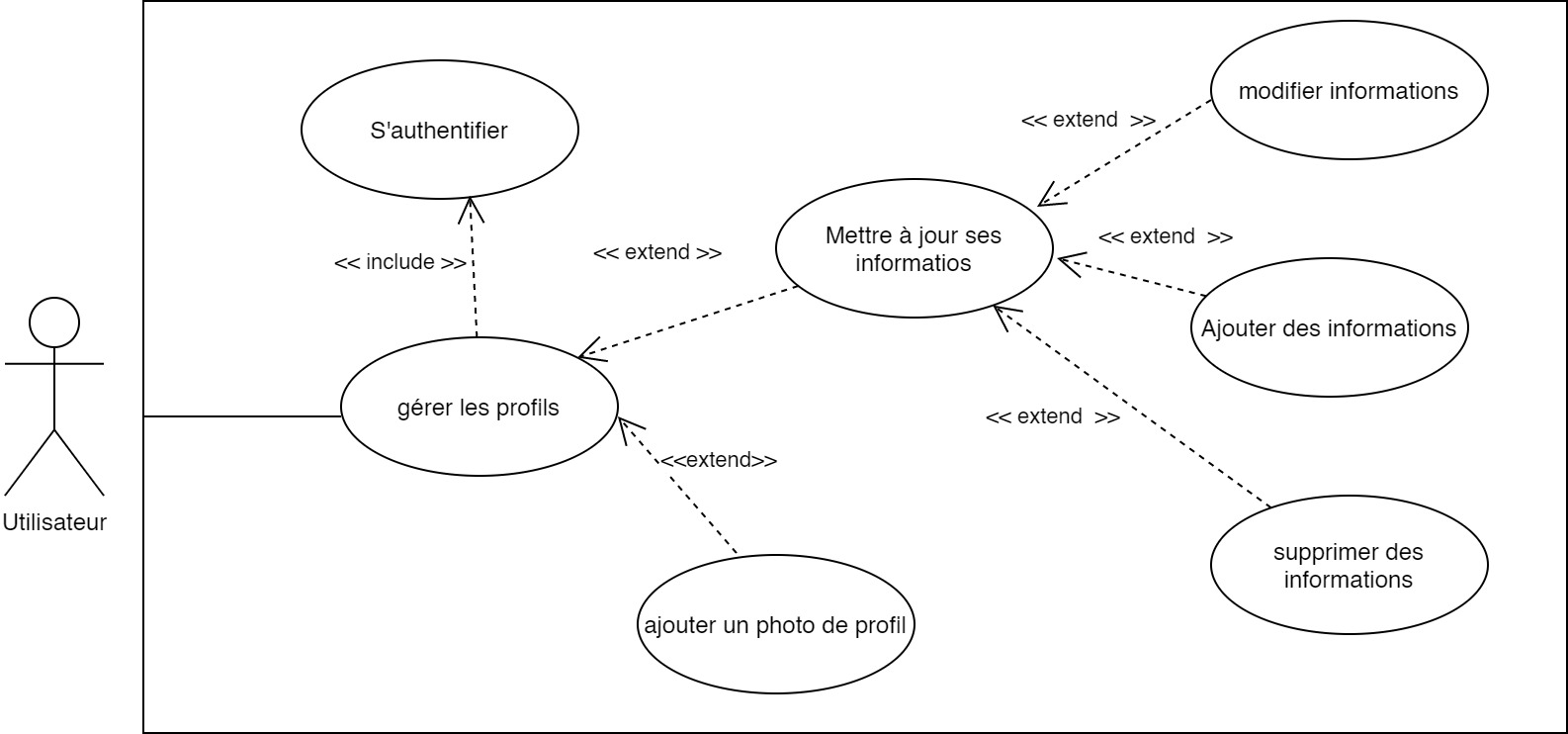


***figure 6 : Cas d’utilisation (Gérer post)***

|  |  |
| --- | --- |
| **Cas d’utilisation** | gérer les posts |
| **description brève** | Ce CU permet à l’acteur de créer, supprimer, modifier ou commenter un article. |
| **Acteur** | Professeur, etudiant, administrateur |
| **Precondition** | L‘acteur doit être connecté. |

**c.2 Analyse du cas d’utilisation « gérer Profil»**

* **Cas d’utilisation « Gérer profil »**

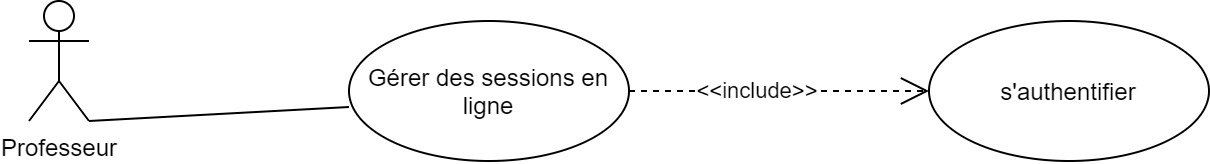


***figure 7 : Cas d’utilisation (Gérer les profil)***

|  |  |
| --- | --- |
| **Cas d’utilisation** | Gérer profile |
| **description brève** | Ce CU qui permet à l’acteur de mettre à jour ses informations. |
| **Acteur** | administrateur, tuteur, apprenant |
| **Precondition** | L‘acteur doit être connecté. |
| **Enchaînement principal** | 1. Le système affiche le profil de l’acteur correspondant. 2. L’acteur met à jour ses informations. 3. Le système fait la vérification des informations entrées par l’utilisateur. 4. Le système notifie l’acteur du bon déroulement de la mise à jour. |
| **Post-condition** |  |
| **Enchaînement alternatif** | Les informations saisies par l'utilisateur sont manquantes ou incorrectes. Ce scénario commence au point 03 du l’enchaînement principal.   1. Le système informe l'utilisateur que les données introduites sont invalides en affichant les messages d’erreur et reprend l’enchaînement du point 2 du l'enchaînement principal. |

1. **Sprint 2 : Gérer des sessions en ligne**
2. **backlog du sprint :**

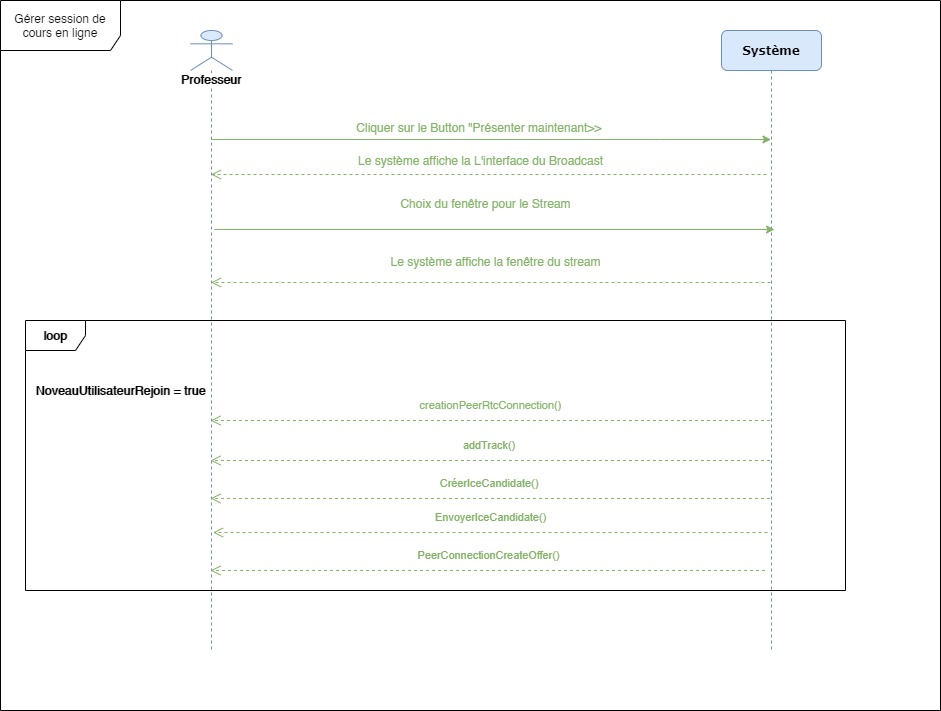
|  |  |  |  |
| --- | --- | --- | --- |
| ID | User Story | Priorité | Estimation |
|  | Gérer des sessions en ligne |  |  |
| 1 | En tant qu’utilisateur(Professeur) ,je peux organiser des séances en ligne. | 1 | 4 |
| 2 | En tant qu’utilisateur(Professeur), je peux supprimer une séance en ligne . | 2 | 5 |
| 3 | En tant qu'utilisateur(Professeur), je peux modifier une séance en ligne |  | 2 |
| 4 | En tant qu' utilisateur(Professeur), je dois gérer la séance de cours en mode broadcast. |  | 40 |
| 5 | En tant qu' utilisateur(Professeur), je veux avoir un listing de tous les étudiant pour la séance. |  | 26 |
| 6 | En tant qu' utilisateur(Étudiant), je peux participer à la séance de cours en ligne en mode viewer. | 1 | 50 |
| 7 | En tant qu’ utilisateur (Professeur ou Étudiant ) je peux faire un échange de message instantanément au cours de la séance. | 2 | 55 |

1. **diagramme de cas d'utilisation << Gérer des sessions en ligne** >>

***figure 2 : Diagramme de cas d’utilisation global du sprint 2 (Release 2)***

1. **Analyse fonctionnelle:**

**c.1 Cas d’utilisation « Gérer des sessions en ligne**  **»**

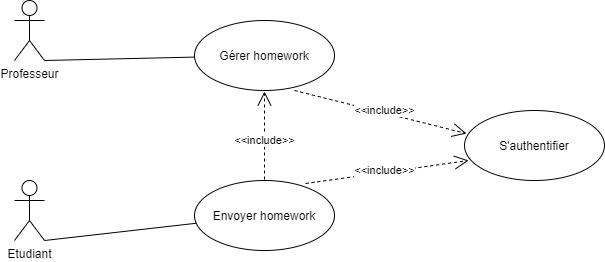
- Diagramme de séquence système du cas d'utilisation "gérer session de cours en ligne" 

**Sprint 3 : Gérer homework et réponse homework :**

1. **backlog du sprint :**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | User Story | Priorité | Estimation |
|  | Gérer Homework |  |  |
| 1 | En tant qu’utilisateur (Professeur) , je peux créer des homeworks. | 1 | 4 |
| 2 | En tant qu’utilisateur (Professeur) , je peux envoyer des homeworks pour les étudiant concernées | 2 | 5 |
| 3 | En tant qu'utilisateur (Professeur) , je peux modifier un Homework |  | 5 |
| 4 | En tant qu' utilisateur (professeur),je peux supprimer des homeworks . |  | 7 |
| 5 | En tant qu’utilisateur (Professeur ) , je veux avoir un listing de tous mes homeworks. |  | 6 |
|  | Envoyer Homework |  |  |
| 7 | En tant qu' utilisateur (Étudiant), je peux envoyer le homework demandé . | 1 | 5 |
| 8 | En tant qu’ utilisateur (Étudiant), je peux vérifier si mon homework est envoyé ou non. | 1 | 8 |
| 9 | En tant qu’ utilisateur (Étudiant), je peux modifier mon homework déjà envoyé . | 2 | 6 |
| 10 | En tant qu’utilisateur ( Étudiant), je veux avoir un listing de tous mes homeworks | 1 | 6 |

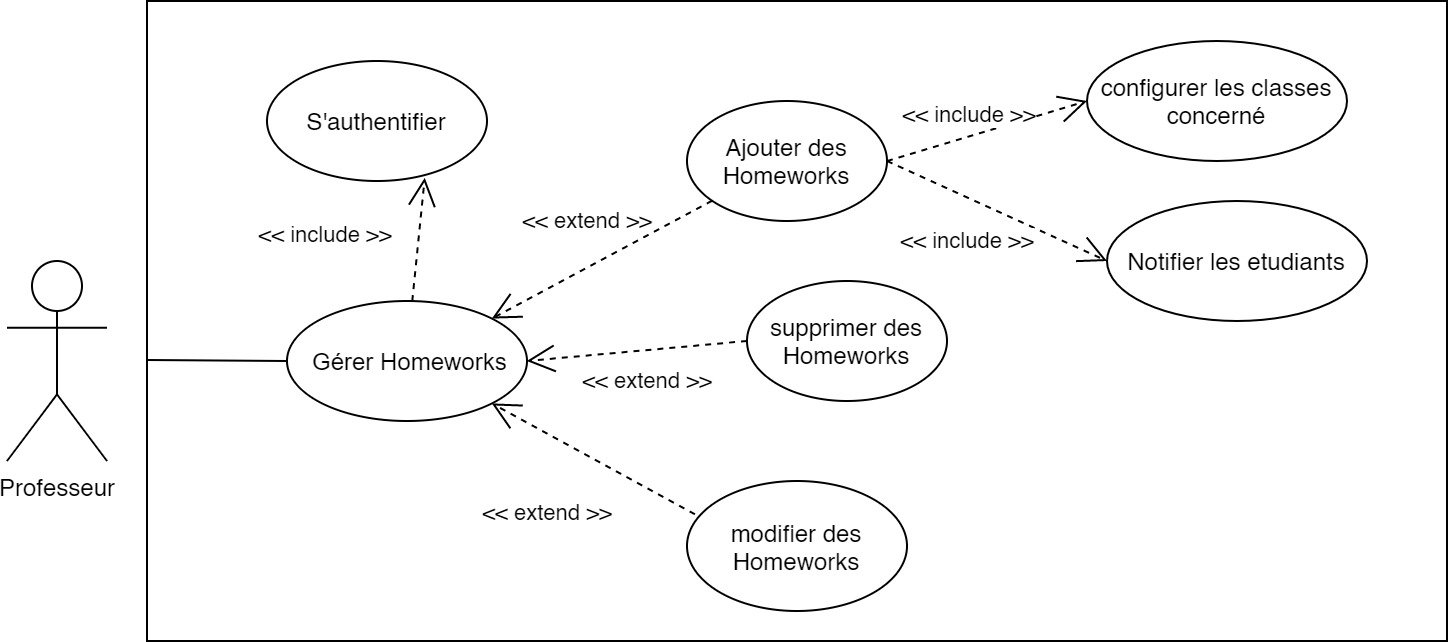
1. **Diagramme de cas d’utilisation global :**

La figure suivante résume les fonctionnalités du troisième sprint : 

***figure 3 : Diagramme de cas d’utilisation global du sprint 3 (Release 2)***

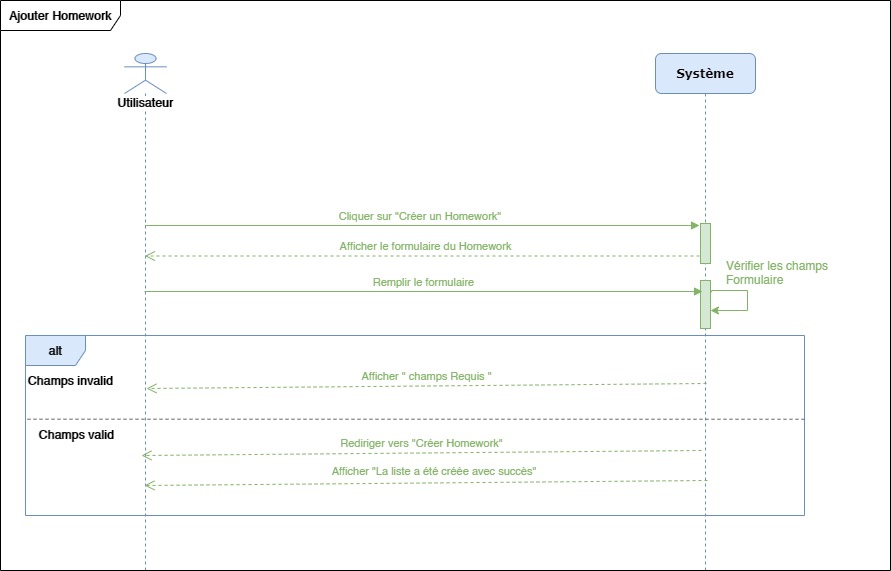
1. **Analyse fonctionnelle:**

**Cas d’utilisation « Gérer Homeworks** **»**

****

***figure 8 : Cas d’utilisation (Gérer Homeworks)***

|  |  |
| --- | --- |
| **Cas d’utilisation** | Gérer Homeworks |
| **description brève** | Ce qui permet à l’acteur d’ajouter, d’annuler et modifier un Homeworks . |
| **Acteur** | Professeur |
| **Precondition** | L‘acteur doit être connecté. |
| **Enchaînement principal** | 1. Le système affiche un formulaire pour . 2. L’acteur remplit les informations concernant le Homework à donner. 3. Le système fait la vérification des informations entrées par l’utilisateur. 4. Le système notifie l’acteur du bon déroulement de l’ajout de homework dans la base de données. |
| **Post-condition** |  |
| **Enchaînement alternatif** | Les informations saisies par l'utilisateur sont manquantes ou incorrectes. Ce scénario commence au point 03 du l’enchaînement principal.   1. Le système informe l'utilisateur que les données introduites sont invalides en affichant les messages d’erreur et reprend l’enchaînement du point 2 du l'enchaînement principal. |

- Diagramme de séquence système du cas d'utilisation "Ajouter Homework" 

**on veut parler du l’architecture de l’app et du model mvvm c’est possible de faire 4eme chapitre qui contient ça**

**1.2 Outils de développement**

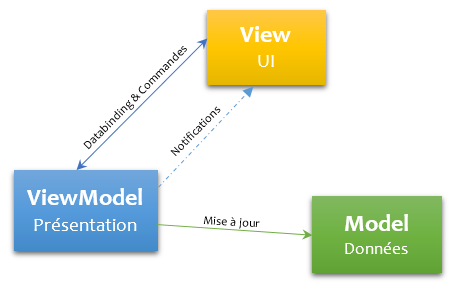
**2. Architecture générale de l'application**

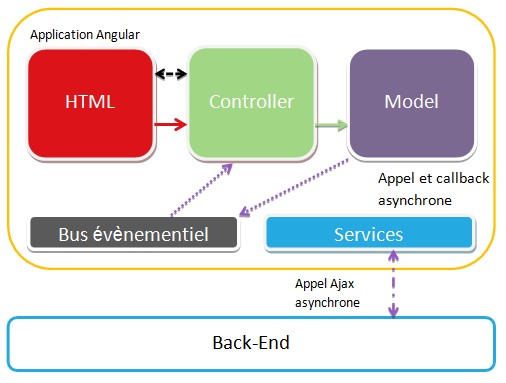
Le système se compose d’un ou plusieurs ordinateurs reliés à l’internet. L’architecture générale de la plateforme permet de donner une vision de haut niveau du système.

L’application réalisée sera fondée sur l’architecture N tiers.

L’architecture N-tier (anglais tier : étage, niveau), ou encore appelée multi-tier, est une architecture client-serveur dans laquelle une application est exécutée par plusieurs composants logiciels distincts.

Exemple d’architecture N-tier :

Dans la réalisation de notre projet, nous avons opté pour une architecture **MVVM** (Model, View, ViewModel) pour assurer la maintenabilité, la modularité de l’application, la rapidité et la facilité du développement.

Le Modèle-Vue-VueModèle (en abrégé MVVM, de l’anglais Model View ViewModel) est une architecture et une méthode de conception utilisée dans le génie logiciel. Cette méthode permet, tel le modèle MVC (Modèle-Vue-Contrôleur), de séparer la vue de la logique et de l’accès aux données en accentuant les principes de binding et d'évènement. 

Pendant plusieurs années, Angular était plus proche de MVC (ou plutôt d'une de ses variantes côté client), mais au fil du temps et grâce à de nombreuses factorisations et améliorations api, il est maintenant plus proche de MVVM…

**Modèle d’architecture combinant Angularjs et MVVM :**

La vue principale, déclare l’application AngularJs. L’application AngularJs contient l’ensemble des modules nécessaires à l’application :

* Les vues
* Les directives / Les filtres
* Les contrôleurs
* Les modèles
* Les services

**LES VUES**

Chaque vue fonctionne de paire avec un contrôleur. La vue consomme les données en data-binding et appelle les méthodes du contrôleur. Elle peut également inclure des directives et faire appel aux filtres déclarés dans l’application.

**LES DIRECTIVES / LES FILTRES**

Les directives sont des add-on au HTML. Une directive est un composant qui peut déclarer ou non son propre scope / contrôleur.

Les filtres sont des méthodes permettant la transformation de la donnée: filtre d’un tableau, formatage d’une date, d’un nombre, … Les filtres peuvent être utilisés directement dans le code HTML ou via le Javascript.

**LES CONTRÔLEURS**

Le contrôleur contient un ou plusieurs model de données. Il contient des méthodes propres à la vue ainsi que des variables d’état ou d’affichage. Il sert également de proxy au model.

**LES MODÈLES**

Les modèles sont dédiés à un type de données. Ils contiennent la donnée et toutes les méthodes de manipulation de cette dernière :

* Appels de services
* Ajout
* Sélection

Les modèles font appel aux services en cas de besoin de communication serveur.

Lors du retour de l’appel de service, le modèle s’appuie sur le bus évènementiel ($rootScope) d’AngularJs pour notifier les changements. Le but de passer par le bus évènementiel est d’éviter toute dépendance du modèle envers les vues qui l’utilisent.

**LES SERVICES**

Les services sont des classes d’appels serveurs