

## **Dédicaces**

*A mes parents, pour vos sacrifices, vos conseils et vos bénédictions.*

*A ma soeur et mon frère, pour le soutien que vous m'avez accordé,*

*A mes collègues, pour m'avoir rendu la période de stage agréable,*

*A toute ma famille,*

*A mes amis et à mes proches,*

*A tous ceux qui ont participé à ma formation,*

*A tous ceux qui m'ont aidé à réaliser ce travail.*

***Toute ma reconnaissance et ma gratitude...***

## REMERCIEMENTS

**J**'aimerais exprimer ma gratitude envers Monsieur Aouadi Bacem, mon encadrant au sein du General Assistance connecté. Son soutien, sa collaboration et son assistance m'ont aidée à mieux profiter de mon stage et ses conseils et orientations me furent d'une inestimable utilité.

**Je** remercie également Monsieur Ben braik Ilyess, un membre de l'équipe, pour son encouragement, ses conseils bien précieux et son encadrement judicieux.

**Mes** remerciements s'adressent également aux membres du jury pour avoir accepté d'évaluer ce travail. J'espère être digne de leur intérêt.

**Finalement** je remercie tous mes enseignants à l'Ecole Nationale Supérieur d'Ingénieurs de Tunis.

## table de matière

<b>INTRODUCTION GENERALE</b>	<b>1</b>
<b>Chapitre 1 : Etat de l'art</b>	<b>3</b>
1.1 Introduction . . . . .	3
1.2 Présentation de l'organisme d'accueil: Général Assistance Connecté . . . . .	3
1.2.1 Présentation du Général Assistance Connecté . . . . .	3
1.2.2 Les missions de Général Assistance Connecté . . . . .	4
1.3 Description du contexte du projet . . . . .	4
1.3.1 Etude de système existant . . . . .	4
1.3.2 Critique de l'existant . . . . .	4
1.3.3 Problématique . . . . .	5
1.3.4 Présentation de la solution proposée . . . . .	5
1.4 Langage et Méthodologie de la conception . . . . .	5
1.4.1 Pilotage du projet avec Scrum . . . . .	6
1.5 Planification et architecture du projet . . . . .	6
1.5.1 Capture des besoins . . . . .	6
1.5.2 Les besoins fonctionnels . . . . .	7
1.5.3 Les besoins non fonctionnels . . . . .	8
1.5.4 Backlog de produit . . . . .	9
1.5.5 Planification de release . . . . .	10
1.5.6 Identification de l'équipe scrum . . . . .	11
1.5.7 Diagramme de cas d'utilisation initiale . . . . .	11
1.5.8 Technologie et outils de développement . . . . .	13
1.5.9 Environnement technique de travail . . . . .	14
1.6 Conclusion . . . . .	16
<b>Chapitre 2 : Release 1 " Développement des modules "</b>	<b>17</b>
2.1 Introduction . . . . .	17
2.2 Sprint 1 : Module d'administration . . . . .	17

2.2.1	Description textuelle du cas d'utilisation "ajouter un utilisateur" . . . . .	18
2.2.2	Description textuelle du cas d'utilisation "chercher un utilisateur" . . . . .	19
2.2.3	Description textuelle du cas d'utilisation "consulter liste des utilisateurs" . . . . .	20
2.2.4	Conception . . . . .	21
2.2.5	Diagramme de séquence "Ajouter un utilisateur" . . . . .	22
2.2.6	Diagramme de séquence "consulter la liste des utilisateurs" . . . . .	23
2.3	Sprint 2 : Module d'assistance . . . . .	23
2.3.1	Description textuelle du cas d'utilisation "ajouter un remorqueur" . . . . .	25
2.3.2	Diagramme de séquence "ajouter un remorqueur" . . . . .	27
2.4	Conclusion . . . . .	28
 <b>Chapitre 3 : Release 2 " Déploiement de l'application et réalisation "</b>		<b>29</b>
3.1	Introduction . . . . .	29
3.2	Sprint 1 : Intégration des modules . . . . .	29

**liste des figures**

1	Figure1 : Logo de la société . . . . .	3
2	Figure 2 : Planification de release . . . . .	11
3	Figure3 : Diagramme de cas d'utilisation . . . . .	12
4	Figure4 : Logo de Spring Boot . . . . .	13
5	Figure5 : Logo de Angular . . . . .	13
6	Figure7 : modèle MVC . . . . .	14
7	Figure9 : Diagramme de classe de module administration . . . . .	17
8	Figure10 : Diagramme de cas d'utilisation " gérer utilisateur " . . . . .	18
9	Figure11 : Diagramme de séquence " gérer utilisateur " . . . . .	22
10	Figure12 : Diagramme de séquence " gérer liste des utilisateurs " . . . . .	23
11	Figure13 : Diagramme de classe de module assistance . . . . .	24
12	Figure14 : Diagramme de cas d'utilisation " gérer remorqueurs " . . . . .	25
13	Figure15 : Diagramme de séquence " gérer remorqueurs " . . . . .	27
14	Figure8 : Le processus Scrum . . . . .	42

**liste des tableaux**

1	tableau1 : les acteurs principaux . . . . .	7
2	tableau2 : Backlog de produit . . . . .	10
3	tableau3 : Identification de l'équipe scrum . . . . .	11
4	tableau4 : Environnement matérielle . . . . .	15
5	tableau 5 : Tableau descriptif du cas d'utilisation "ajouter un utilisateur" . . . . .	19
6	tableau 6 : Tableau descriptif du cas d'utilisation "chercher un utilisateur" . . . . .	20
7	tableau 7 : Tableau descriptif du cas d'utilisation "consulter liste des utilisateurs" . . . . .	21
8	tableau 8 : Tableau descriptif du cas d'utilisation "ajouter un remorqueur" . . . . .	26

## INTRODUCTION GENERALE

De nos jours, avec la mécanisation de tous les secteurs de l'économie et surtout de la modernisation de plus en plus poussée du trafic routier, nous assistons à une augmentation exponentielle du nombre d'accidents de la route.

Selon les statistiques de l'observatoire national de la sécurité routière, le nombre d'accidents en 2015 a dépassé 7225. La plupart des gens trouvent des difficultés à remplir correctement le constat amiable d'accident. Il y a des autres qui ne disposent pas des sommes nécessaires pour payer les réparations de leur véhicule accidenté et même ne savent pas quel réparateur choisir et donc tombent dans un long délai d'attente et une mauvaise qualité de réparation.

c'est pour cela il faut exploiter l'émergence de technologie informatique pour aider la société à gérer ce flux d'une manière informatisée d'où la nécessité d'une application web dynamique et ergonomique.

Dans ce cadre j'ai effectué un stage s'inscrit respectivement dans mon cursus , sur une mission de développement échelonnée sur une période de deux mois entre Juillet et Août 2017.

Le directeur Monsieur Aouadi Bacem et sa société "General Assistance connecté" a d'emblée suscité mon intérêt avec un projet concret lié au développement d'une application web avec une vraie solution innovante et utilisant un nouveau langage, via un Framework simplifiant la création.

Dès cette première prise de contact, j'ai su que l'ambiance de travail allait être encourageante et rigoureuse, en intégrant cette structure dynamique et performante.

Mes attentes par rapport à mon cursus universitaire étaient notamment de travailler en équipe, de m'adapter à différents environnements matériels et logiciels, mais aussi d'identifier les possibilités et les limites des applications de l'informatique, et enfin d'intégrer les connaissances acquises pour la résolution des problèmes rencontrés.

Le projet en soi s'est orienté sur plusieurs axes de travail sur lesquels j'ai dû me focaliser :

- Planification des sprints(modules).
- Développement de module administration.
- développement de module assistance.
- intégration des modules et déploiement de l'application.

Pour ce faire, il a fallu suivre quelques étapes indispensables, préalables au déploiement du projet, en assimilant d'abord les méthodologies de travail, techniques et outils, en s'investissant à participer à de nouvelles fonctionnalités.

Dans le premier chapitre, nous présenterons le cadre de notre projet et il présentera aussi notre sprint 0 ou encore la phase de préparation où nous détaillerons les technologies que nous allons manipuler tout au long de notre projet.

Le deuxième chapitre portera sur les étapes de développement.

Et notre dernier chapitre sera dédié à la partie de l'intégration des module et comparaison entre les choix utilisés. Nous finirons par une conclusion générale ainsi que la proposition de quelques perspectives.



## Chapitre 1 : Etat de l'art

### 1.1 Introduction

Le présent chapitre est consacré en un premier lieu à la présentation de l'entreprise d'accueil dans laquelle nous avons effectué notre projet de fin d'étude.

En deuxième lieu, nous effectuons une étude de l'existant et nous dégageons les défaillances. En troisième lieu, nous présentons notre solution dont le but de remédier à ses limites. En quatrième lieu nous dégageons les besoins fonctionnels et non fonctionnels ainsi que le backlog de produit suivi par la planification des releases et nous terminons par une présentation des technologies et des outils de développement.

### 1.2 Présentation de l'organisme d'accueil: Général Assistance Connecté

Notre stage se déroule au sein de la société Général Assistance Connecté. Dans la section qui suit, nous allons donner une description légère de l'entreprise, ses missions ainsi que ses secteurs d'activités.

#### 1.2.1 Présentation du Général Assistance Connecté

Général Assistance Connecté est une entreprise d'ingénierie et de services informatique tunisienne. Elle s'est installée à Tunis depuis 2015 et elle compte actuellement moins de 20 employés.

Elle est la filiale informatique du groupe Générale Assistance, forte d'un double positionnement d'éditeur de logiciels informatiques, et également de prestataires de service aussi bien dans l'ingénierie logicielle que dans l'administration Système & réseaux.



Figure 1: Logo de la société

### **1.2.2 Les missions de Général Assistance Connecté**

Les missions du Général Assistance Connecté sont :

- Mise à disposition de ressources informatiques compétentes sur les nouvelles technologies,
- Installation, administration et sécurisation du système et réseau informatique,
- Développement web et mobile spécifiques aux besoins métiers du client,
- Développement Back-End, Front-End, Full Stack en utilisant les langages et technologies de pointe.

Le secteur d'activité est :

- Assurance

## **1.3 Description du contexte du projet**

### **1.3.1 Etude de système existant**

L'étude de l'existant est une phase importante pour bien comprendre l'application web et dégager ces fonctionnalités et ces différents points faibles, ce site est consacré pour le groupe Générale Assistance. Il faut signaler que Générale Assistance joue un véritable rôle culturel en participant au développement des services d'assistance aux personnes par la conception de nouvelles prestations répondants aux besoins évolutifs des clients.

Aussi, le groupe exploite les nouvelles technologies d'information et de communication pour améliorer les procédures de gestion des sinistres automobile, habitation et santé en maîtrisant leurs coûts.. Le présent site web assure et simplifie les services précédemment cité d'une manière informatisé.

### **1.3.2 Critique de l'existant**

Vu le grand nombre de demande d'assurance et de collaborateur, le projet doit être un site de qualité (efficace, fiable, compatible, séduisant...) et permet de satisfaire les cibles pour lesquelles le site est conçu, mais dans notre cas, la Template utilisée par le site n'est pas ergonomique et parfois les informations ne sont pas bien lisibles ainsi que le chargement de quelque page prendre un temps cela à cause de l'emploi excessif d'images ou d'animations et l'absence d'un menu fixe qui permette d'atteindre les pages principales du site quel que soit la page chargée.

### 1.3.3 Problématique

Le client de la société veut satisfaire les cibles pour lesquelles le site est conçu et s'adapter aux utilisateurs en améliorant les services offertes mais le problème c'est que le taux de traitement est important et la méthode utilisée est lente et n'est pas performante, d'où le besoin de migrer le site en intégrant un nouveau design et corrigeant les bugs, de développer les modules complémentaires pour améliorer les services.

### 1.3.4 Présentation de la solution proposée

Dans un souci de migrer le site web de notre client vers une nouvelle version plus ergonomique et ne présentant pas de bug, notre solution proposée se résume ainsi dans les fonctionnalités suivantes :

- Migration de site web
- Développement des modules complémentaires

## 1.4 Langage et Méthodologie de la conception

La méthodologie est une démarche organisée rationnellement pour aboutir à un résultat. Parmi les différentes méthodologies existantes, nous pouvons citer le modèle en cascade utilisé souvent dans les projets simples dont les besoins sont clairs et bien définis dès le début, le modèle en y utilisé pour le développement des applications mobiles, ainsi que le processus unifié et les méthodologies agiles (Scrum & extrême programming) caractérisées par leurs souplesses et utilisées dans des grands projets. Pour bien conduire notre projet et nous assurer du bon déroulement des différentes phases, nous avons opté Scrum comme une méthodologie de conception et de développement.

Après le choix de la méthodologie, nous avons besoins, d'un langage de modélisation unifiée pour la modélisation de notre projet. Pour concevoir notre système, nous avons choisi UML<sup>1</sup> comme un langage de modélisation. Notre choix s'est basé sur les points forts de ce langage notamment sa standardisation et les divers diagrammes qu'il propose. Aussi UML présente le meilleur outil pour schématiser des systèmes complexes sous un format graphique et textuel simplifié et normalisé. En effet, UML n'est ni un processus ni une démarche, d'où il fallait choisir une méthodologie de conception et de développement que nous devons l'adopter. Voir Annexe

---

<sup>1</sup>Unified Modeling Language

## **1.4.1 Pilotage du projet avec Scrum**

### **1.4.1.1 Planification d'un projet par Scrum**

Planification du sprint : Elle s'appuie sur la planification de la « release » réalisée en pente. La première réunion du sprint ne se limite pas à planifier, on y trouve les activités suivantes :

1. Valider les «stories» du Backlog pris en compte dans le sprint, concevoir les solutions.
2. Identifier et estimer les tâches.
3. Prise des tâches par chacun des membres de l'équipe ...

Revue du sprint : Elle permet de montrer les résultats du développement effectués au cours du sprint, seule une version opérationnelle est montrée.

Rétrospective : Elle est faite en interne en équipe (avec la présence du Scrum Master), l'objectif est de comprendre ce qui n'a pas bien fonctionné dans le sprint, les erreurs commises et de prendre des décisions pour procéder aux améliorations.

Scrum quotidien : il s'agit d'une réunion de synchronisation de l'équipe de développement qui se fait debout en 15 minutes maximum au cours de laquelle chacun répond principalement à 3 questions :

1. Qu'est-ce que j'ai fait hier ?
2. Qu'est-ce que je ferai aujourd'hui ?
3. Quels obstacles me retardent ?

Voir Annexe

## **1.5 Planification et architecture du projet**

### **1.5.1 Capture des besoins**

Nous procéderons par l'identification des acteurs puis la description des histoires utilisateurs du futur système

#### **1.5.1.1 Identification des acteurs**

Nous avons identifié trois acteurs principaux de notre projet à savoir :

- Le super-administrateur
- L'administrateur( Utilisateur normal, employé dans la société)

Acteur	Définition	Rôle
Le super-administrateur	Il s'agit de la personne qui possède toute les permissions sur le système.	Il est responsable de la gestion interne des profils.
L'utilisateur normal	Il s'agit de la personne qui possède des permissions bien spécifiques	Il peut utiliser toutes les fonctionnalités de l'application sauf l'ajout d'un autre administrateur et la consultation de l'historique de connexion.

Table 1: les acteurs principaux

### 1.5.2 Les besoins fonctionnels

«Les besoins fonctionnels expriment une action que doit effectuer le système en réponse à une demande (sorties qui sont produites pour un ensemble donné d'entrées ». [3]

Dans ce qui suit, nous décrivons les besoins fonctionnels de notre projet appelés dans Scrum :

«User stories ».

- **Le projet permet au super-admin de :**

- \* S'authentifier :

- Se connecter à travers un login et un mot de passe en tant que super-admin.
    - Il y a contrôl de saisie de login et mot de passe, session expirée après une durée bien déterminée autre que blocage de compte après trois tentatives erronées.

- \* Changer ces informations personnelles :

- Changer ses informations personnelles à savoir nom, prénom, mot de passe... etc en cas de besoin.

\* Gestion des rôles :

- Affecter des modifications sur les rôles en ajoutant un nouveau en cas des besoins ou bien en modifiant les profils.

\* Gestion des utilisateurs :

- Ajouter, supprimer ou bien modifier le compte d'un utilisateur en remplissant un formulaire.

\* Gestion des contrats :

- Ajouter, supprimer, bloquer, débloquer ou bien modifier un contrat en remplissant un formulaire.

\* Gestion des assurés :

- Ajouter, supprimer ou bien modifier le profil d'un assuré en remplissant un formulaire.

\* Gestion des marques de voitures :

- Ajouter ou bien modifier les marques des voitures en remplissant un formulaire.

\* Gestion des types de services :

- Ajouter, supprimer ou bien modifier un type de service en remplissant un formulaire.

\* Gestion des remorqueurs :

- Ajouter, supprimer, bloquer, débloquer ou bien modifier un remorqueur en remplissant un formulaire.

\* Gestion des packs :

- Ajouter, supprimer, bloquer, débloquer ou bien modifier un pack en remplissant un formulaire.  
Un pack contient un nombre maximum de kilométrage et de service. Un contrat peut avoir un ou plusieurs packs.

• **Le projet permet à l'utilisateur normal de :**

- \* Faire toutes les fonctionnalités offertes au super-admin sauf la gestion des profils.

### **1.5.3 Les besoins non fonctionnels**

«Il s'agit des besoins qui caractérisent le système. Ce sont des besoins en matière de performance, de type de matériel ou le type de conception. Ces besoins peuvent concerner les contraintes d'implémentation

(langage de programmation, type SGBD.) ».[4]

Les besoins non fonctionnels de notre système, appelés dans Scrum «Technical Stories», se décrivent comme suit :

- **Contraintes ergonomiques**

- La navigation entre les interfaces de notre future site web doit être légère et fluide.
- Les interfaces de notre future application doivent être simples et homogènes.
- L'utilisateur doit être guidé lors de la saisie de certaines informations, afin de respecter les formats des champs de notre base de données.

- **Contraintes techniques**

- Les mots de passe doivent être cryptés au niveau de la base de données afin de garder sécurisé l'accès à l'espace administration.
- Les requêtes doivent être optimisées afin d'assurer un temps de réponse minimal.

#### **1.5.4 Backlog de produit**

Le Backlog produit est utilisé pour planifier la release et aussi à chaque sprint, lors de la réunion de planification du sprint pour décider du sous-ensemble qui sera réalisé. Ses éléments sont classés par ordre de priorité ce qui permet de définir l'ordre de réalisation. Pour chaque user story on identifie le rang, l'estimation, l'importance et la description.

Voir Annexe.

Le tableau suivant décrit le product backlog de notre projet.

<b>Id</b>	<b>Nom</b>	<b>Description</b>	<b>Priorité</b>
1	Développement de module administration	Le super admin peut bénéficier de fonctionnalités d'administration de l'application	5
2	Développement de module d'assistance	Préparer l'environnement de travail pour tous les utilisateurs de l'application	8
3	Intégration des modules d'administration et d'assistances	Fournir une application complète prête à utiliser	13

Table 2: Backlog de produit

### 1.5.5 Planification de release

La réunion de planification des sprints est une étape importante dans SCRUM. Le planning de travail et l'identification du Backlog des sprints sont les buts de cette réunion. Durant ce meeting, la durée de chaque sprint est décidée selon la complexité du contexte et la taille de l'équipe.

La planification de Sprint répond aux questions suivantes :

1. Qu'est-ce qui peut être terminé au cours de ce Sprint ?
2. Comment sera effectué le travail choisi ?

Et donc, une succession de Releases est le produit de cette réunion pour organiser le travail de l'équipe. Un Release est une série de Sprints qui se termine quand les incréments successifs constituent un produit présentant suffisamment de valeurs à ses utilisateurs. Dans notre cas, nous avons décidé de découper notre projet en trois Releases.



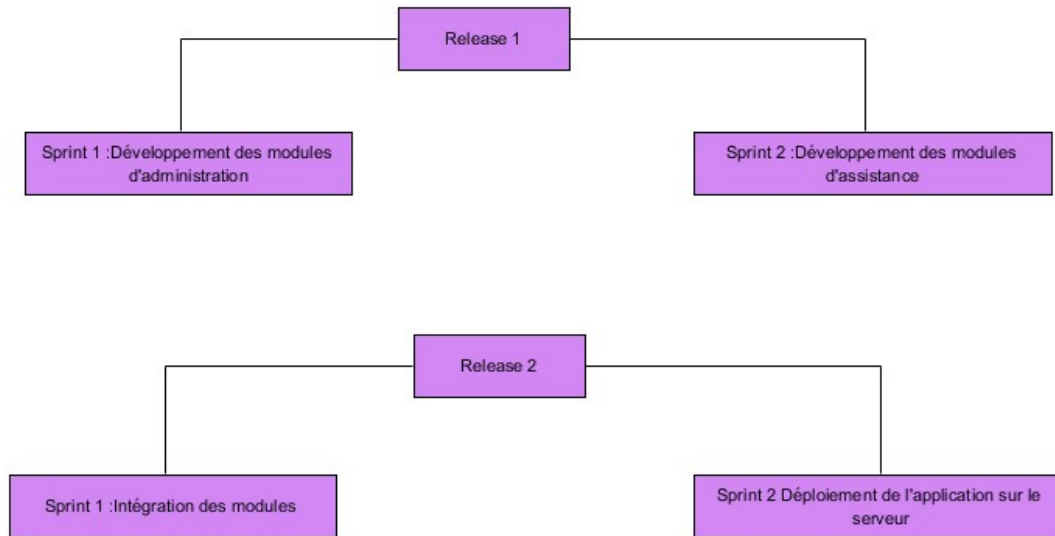


Figure 2: Planification de release

### 1.5.6 Identification de l'équipe scrum

Rôles Scrum	Personnes affectés
Product Owner	Toukebri Mohamed Arbi
Scrum Master	Aouadi Bacem
Team	Choubani amir, Dahmeni Senda, Ardhaoui Aymen, Turki nessrine.

Table 3: Identification de l'équipe scrum

### 1.5.7 Diagramme de cas d'utilisation initiale

Maintenant, nous allons présenter les différents besoins fonctionnels de notre système exprimés précédemment d'une manière informelle en utilisant le diagramme de cas d'utilisation d'UML.

La figure 3 illustre le diagramme de cas d'utilisation global de notre système.

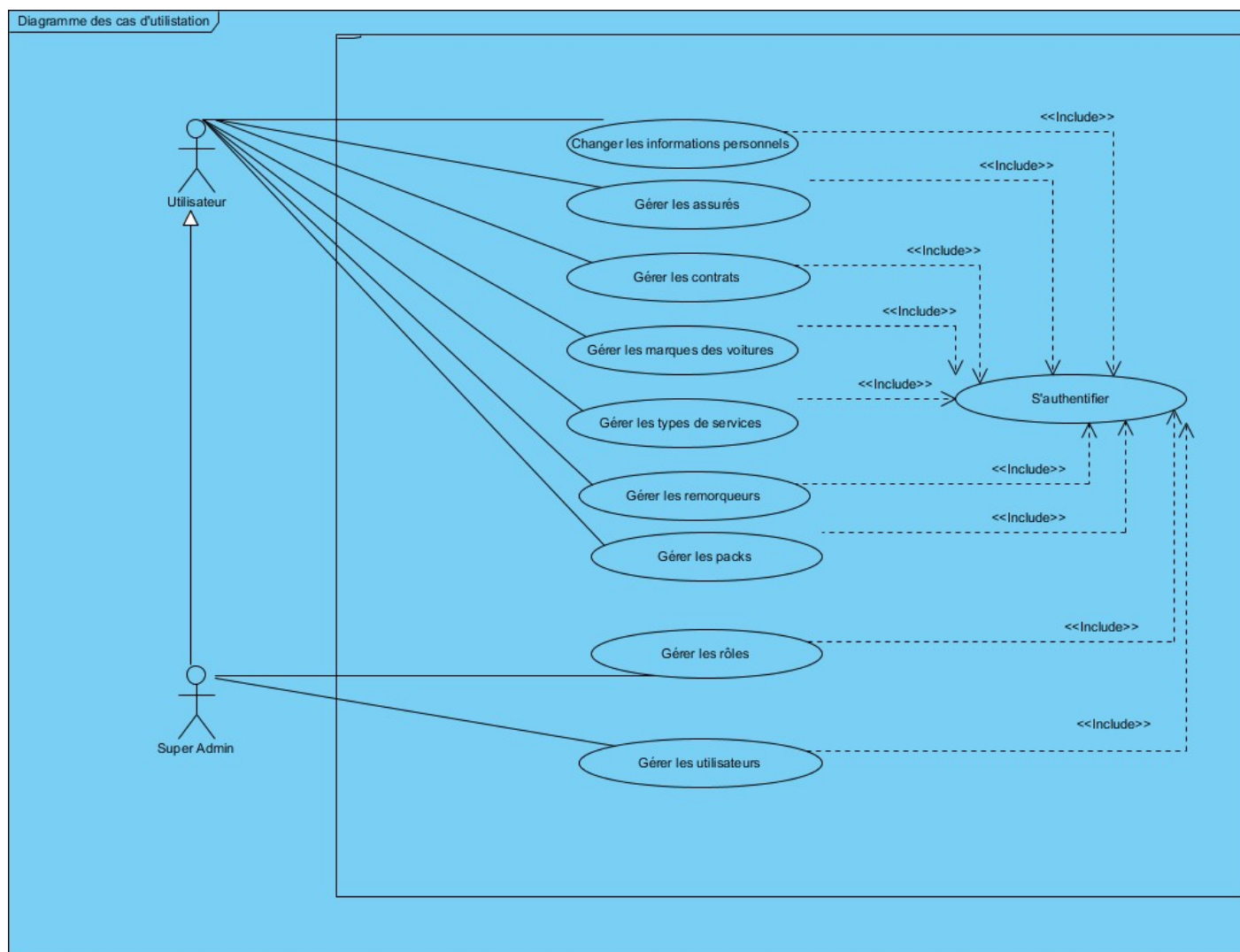


Figure 3: Diagramme de cas d'utilisation

### 1.5.8 Technologie et outils de développement

Dans cette partie, nous décrivons tous les outils matériels et logiciels de développement et de conception, avec lesquels on a réalisé ce projet. Nous allons commencer par une brève description des environnements Spring Boot et Angular, de l'architecture de notre application, et nous allons finir par les outils de développement utilisé pour réaliser ce projet.

#### 1.5.8.1 Présentation des environnements de travail

- **Spring Boot**



Figure 4: Logo de Spring Boot [5]

Spring Boot est un nouveau framework créé par l'équipe de chez Pivotal, conçu pour simplifier le démarrage et le développement de nouvelles applications Spring.

Voir Annexe

- **Angular**



Figure 5: Logo de Angular [6]

Angular est un framework Javascript pour créer des applications monopages, web et mobile.

Il supporte plusieurs langage : ES5,ES6 et typescript.C'est une version qui suit AngularJS et qui a introduit un travail complètement différent en utilisant l'orienté objet (notion de composant).

- **PostgreSQL** Pour la gestion de base de donnée , on a utilisé PostgreSQL qui est similaire à MYSQL.

### 1.5.8.2 Architecture utilisée

L'architecture en couches de notre système se base sur le patron (modèle) de conception MVC (Model View Controller).

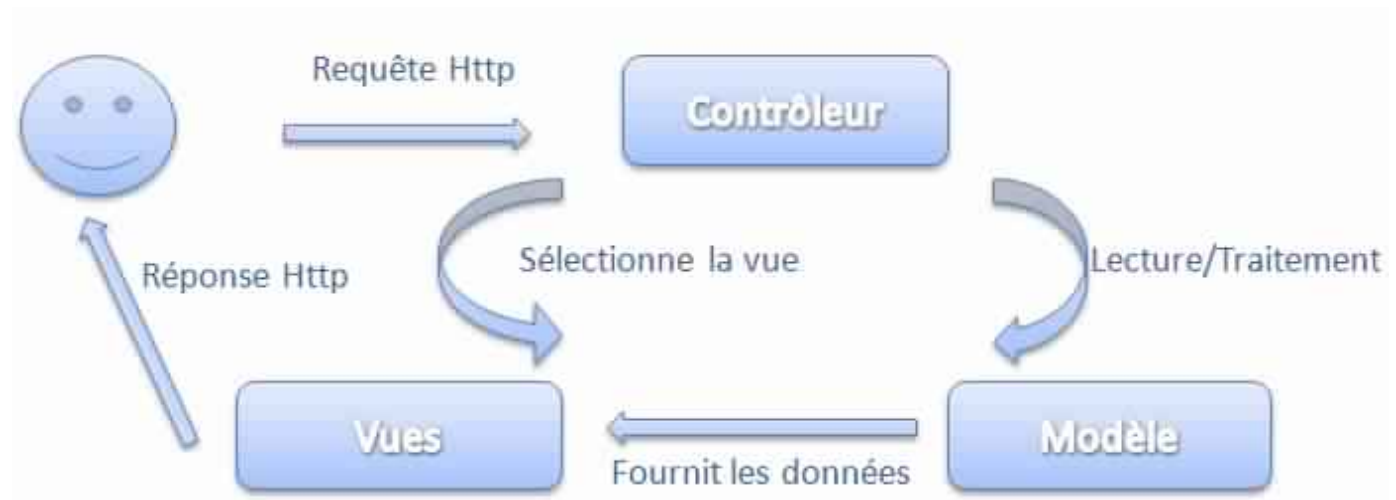


Figure 6: modèle MVC [7]

Modèle-Vue-Contrôleur est un motif d'architecture logicielle destiné pour les applications web. Le motif est composé de trois types de modules ayant trois responsabilités différentes : les modèles, les vues et les contrôleurs.

Voir Annexe

### 1.5.9 Environnement technique de travail

Le tableau détaille la configuration matérielle de machine de développement et de déploiement utilisée ainsi que les différents logiciels qui y sont installés :

Machines	Asus
Propriétaire	Général Assistance Connecté
Processeur	I7
RAM	8 GO
Système d'exploitation	UBUNTO 16.04

Table 4: Environnement matérielle

#### 1.5.9.1 Outil de conception

Visual Paradigm for Unified Modeling Language (VP-UML) est un outil UML. L'outil est conçu pour un large éventail d'utilisateurs, y compris les ingénieurs logiciels, analystes système, analystes et architectes système, ou pour quiconque est intéressé par la construction de manière fiable des systèmes de logiciels à grande échelle en utilisant une approche orientée objet.

#### 1.5.9.2 Outils de développement

Dans cette partie nous allons décrire les outils de développement utilisé pour réaliser ce projet.



### 1.5.9.3 Langages de programmation

- **Java Enterprise Edition**
- **TypeScript**
- **Bootstrap 4**
- **HTML 5**
- **CSS 3**
- **SCSS**

voir annexe.

## 1.6 Conclusion

Dans ce chapitre nous avons présenté le cadre général de notre projet en déterminant la problématique et en proposant la solution envisagée pour faire face à la situation courante. Nous avons dévoilé le langage et la méthodologie de conception qui seront utilisés dans les prochains chapitres de ce rapport. Nous avons présenté le Sprint 0 qui, est en fait une phase de préparation pendant laquelle on a construit notre backlog du produit en se basant sur les besoins fonctionnels et non fonctionnels.

En plus, nous avons découpé notre projet en des releases qui formeront les chapitres suivants. Et finalement, nous avons défini les technologies et les environnements de développement ainsi que l'architecture globale de l'application. Dans le chapitre suivant nous abordons le release 1 dans laquelle nous allons décrire les sprints à réaliser.

## Chapitre 2 : Release 1 " Développement des modules "

### 2.1 Introduction

Après avoir entamé le release 0 de projet , nous pouvons maintenant passer au premier release «Gestion d'administration et d'assistance ». En effet, ce release constitue un objectif principal de notre projet puisqu'il permet de gérer l'accès à l'application et contient les tâches principales .

### 2.2 Sprint 1 : Module d'administration

Au niveau de ce sprint, nous commençons par présenter le diagramme des classes nécessaires à l'implémentation. Ensuite, on présentera la conception avec les diagrammes de cas d'utilisation et de séquences.

- Diagramme des classes

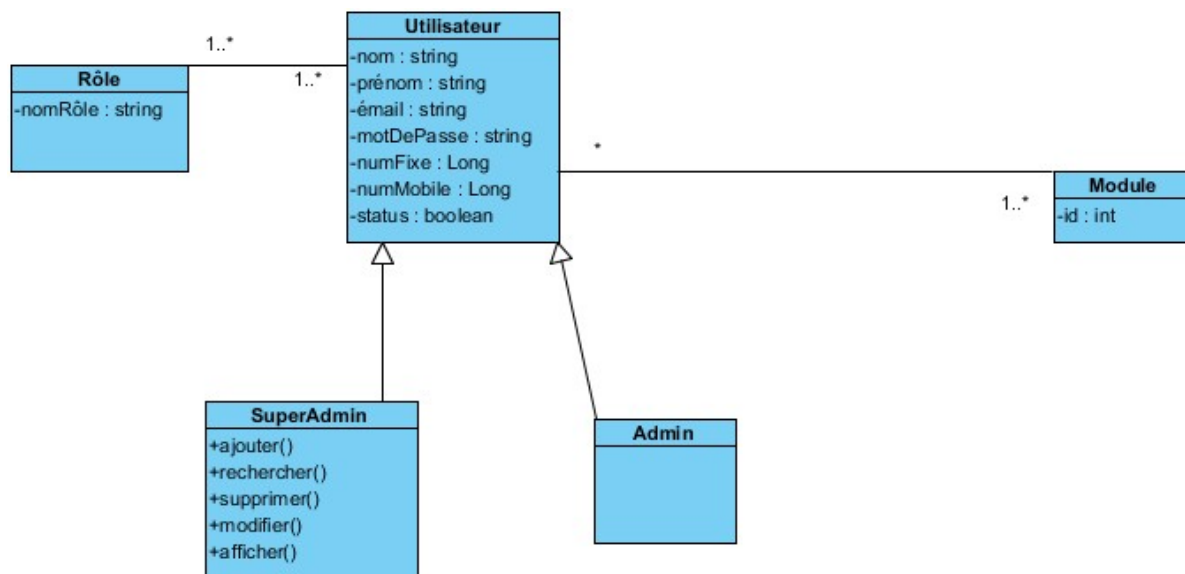


Figure 7: Diagramme de classe de module administration

- Diagramme de cas d'utilisation

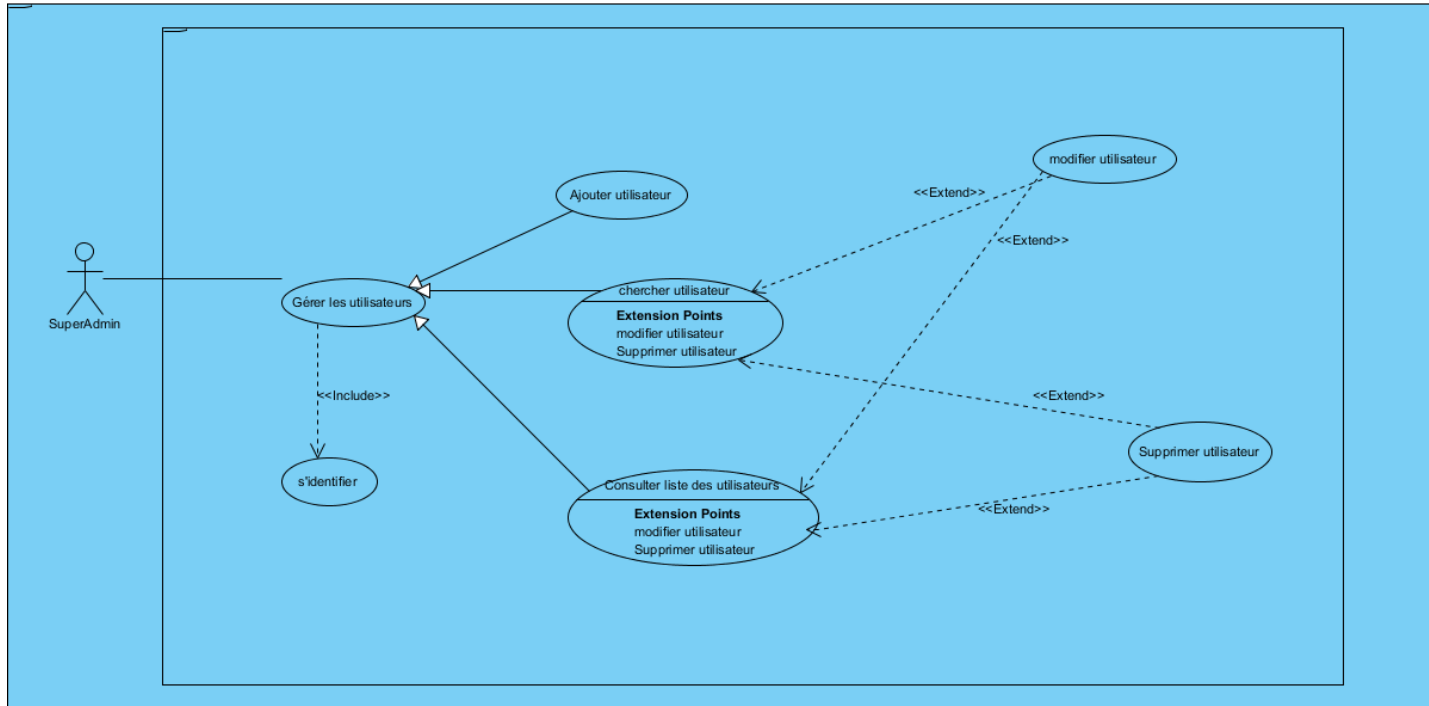


Figure 8: Diagramme de cas d'utilisation " gérer utilisateur "

Le diagramme de cas d'utilisation présenté par la figure 8 décrit la gestion des utilisateurs du système par l'administrateur. Ce diagramme nous montre que l'administrateur ne peut faire sa gestion qu'après identification. Ainsi que la modification ou la suppression d'un utilisateur ne peut être effectuée qu'après consultation des utilisateurs ou recherche de ce dernier .

### 2.2.1 Description textuelle du cas d'utilisation "ajouter un utilisateur"

Nous allons réaliser une description textuelle du scénario de ajout d'un utilisateur pour mieux connaître les étapes à suivre pour que l'administrateur réussit cette ajout.

Cette description est présentée par le tableau 5.



<b>Titre</b>	<b>Ajouter un utilisateur</b>
<b>Résumé</b>	Ajouter un utilisateur de l'application à la base de données
<b>Acteurs</b>	Super Administrateur
<b>Précondition</b>	S'identifier
<b>Postcondition</b>	Un nouvel utilisateur est sauvegardé
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. l'administrateur saisi les données du nouvel utilisateur</li> <li>2. l'administrateur appui sur le bouton sauvegarder pour valider l'ajout</li> <li>3. l'application affiche un message de succès.</li> </ol>
<b>Scénario alternatif</b>	<ol style="list-style-type: none"> <li>1. L'administrateur effectue une erreur de saisi(formulaire non entièrement rempli . . .)</li> <li>2. L'application affiche un message d'erreur</li> <li>3. le scénario nominal reprend au point 1.</li> </ol>

Table 5: Tableau descriptif du cas d'utilisation "ajouter un utilisateur"

### 2.2.2 Description textuelle du cas d'utilisation "chercher un utilisateur"

Nous allons réaliser un tableau descriptif 6 du scénario de la recherche d'un utilisateur pour mieux connaître les étapes à suivre pour que l'administrateur réussit cette tâche.

<b>Titre</b>	<b>Chercher un utilisateur</b>
<b>Résumé</b>	Chercher un utilisateur
<b>Acteurs</b>	Super Administrateur
<b>Précondition</b>	S'identifier
<b>Postcondition</b>	La liste des utilisateurs est affichée.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. l'administrateur saisi le nom et prénom de l'utilisateur</li> <li>2. l'application affiche une liste des utilisateurs qui porte le même nom</li> </ol>
<b>Scénario alternatif</b>	<ol style="list-style-type: none"> <li>1. L'administrateur saisi des données erronées</li> <li>2. le scénario nominal reprend au point 1.</li> </ol>

Table 6: Tableau descriptif du cas d'utilisation "chercher un utilisateur"

### 2.2.3 Description textuelle du cas d'utilisation "consulter liste des utilisateurs"

Nous allons réaliser une description textuelle du scénario de la consultation de la liste des utilisateurs pour mieux connaître les étapes à suivre pour que l'administrateur réussit cette consultation. Cette description est présentée par le tableau 7

<b>Titre</b>	<b>Consulter la liste des utilisateurs</b>
<b>Résumé</b>	Consulter la liste des utilisateurs
<b>Acteurs</b>	Super Administrateur
<b>Précondition</b>	S'identifier
<b>Postcondition</b>	La liste des utilisateurs est affichée.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. l'administrateur appui sur la page de navigation voulu «liste les utilisateurs »</li> <li>2. l'administrateur consulte la liste des utilisateurs.</li> </ol>
<b>Scénario alternatif</b>	<ol style="list-style-type: none"> <li>1. L'administrateur sélectionne la page de navigation non voulu</li> <li>2. le scénario nominal reprend au point 1.</li> </ol>

Table 7: Tableau descriptif du cas d'utilisation "consulter liste des utilisateurs"

#### 2.2.4 Conception

Après la spécification des différentes fonctionnalités du backlog de ce sprint, nous passons à la phase de conception pour mieux expliquer ses fonctionnalités à l'aide des diagrammes de séquence.

### 2.2.5 Diagramme de séquence "Ajouter un utilisateur"

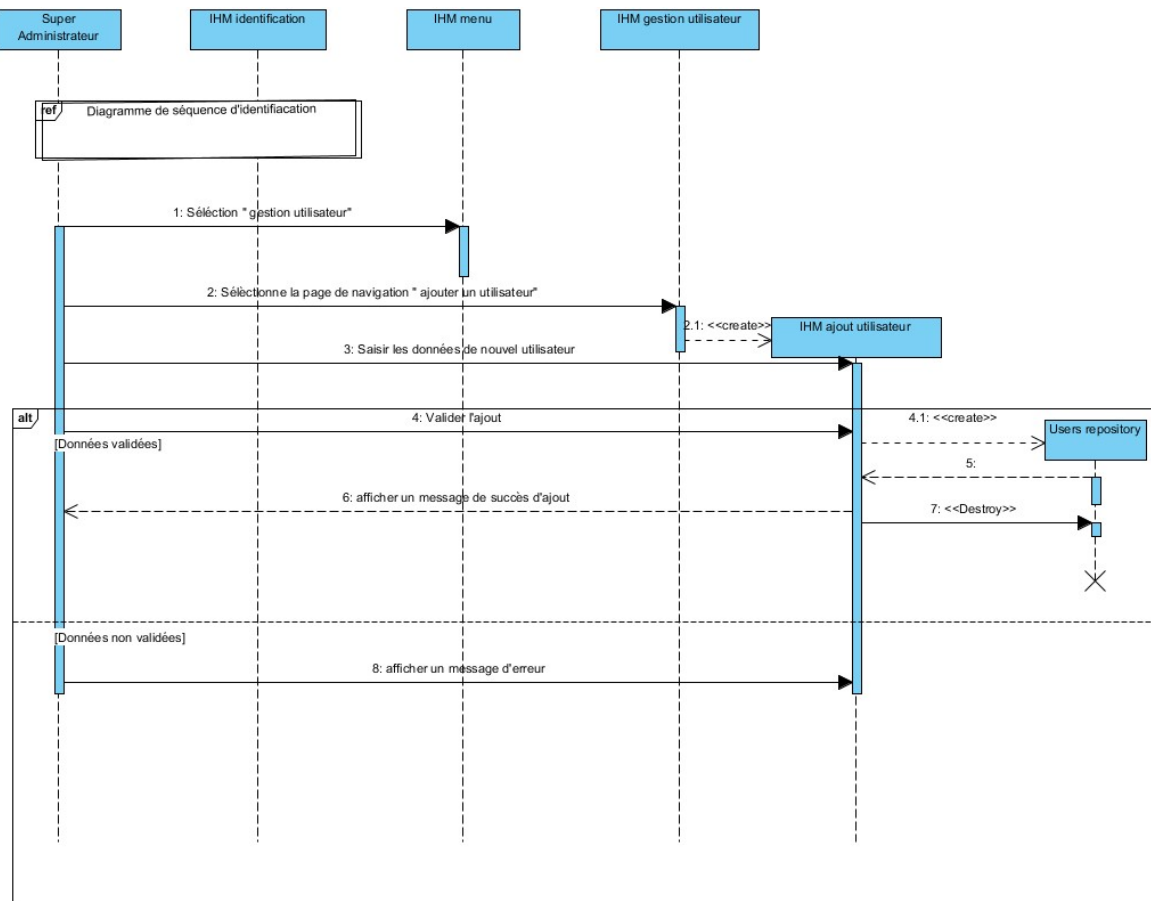


Figure 9: Diagramme de séquence " gérer utilisateur "

## 2.2.6 Diagramme de séquence "consulter la liste des utilisateurs"

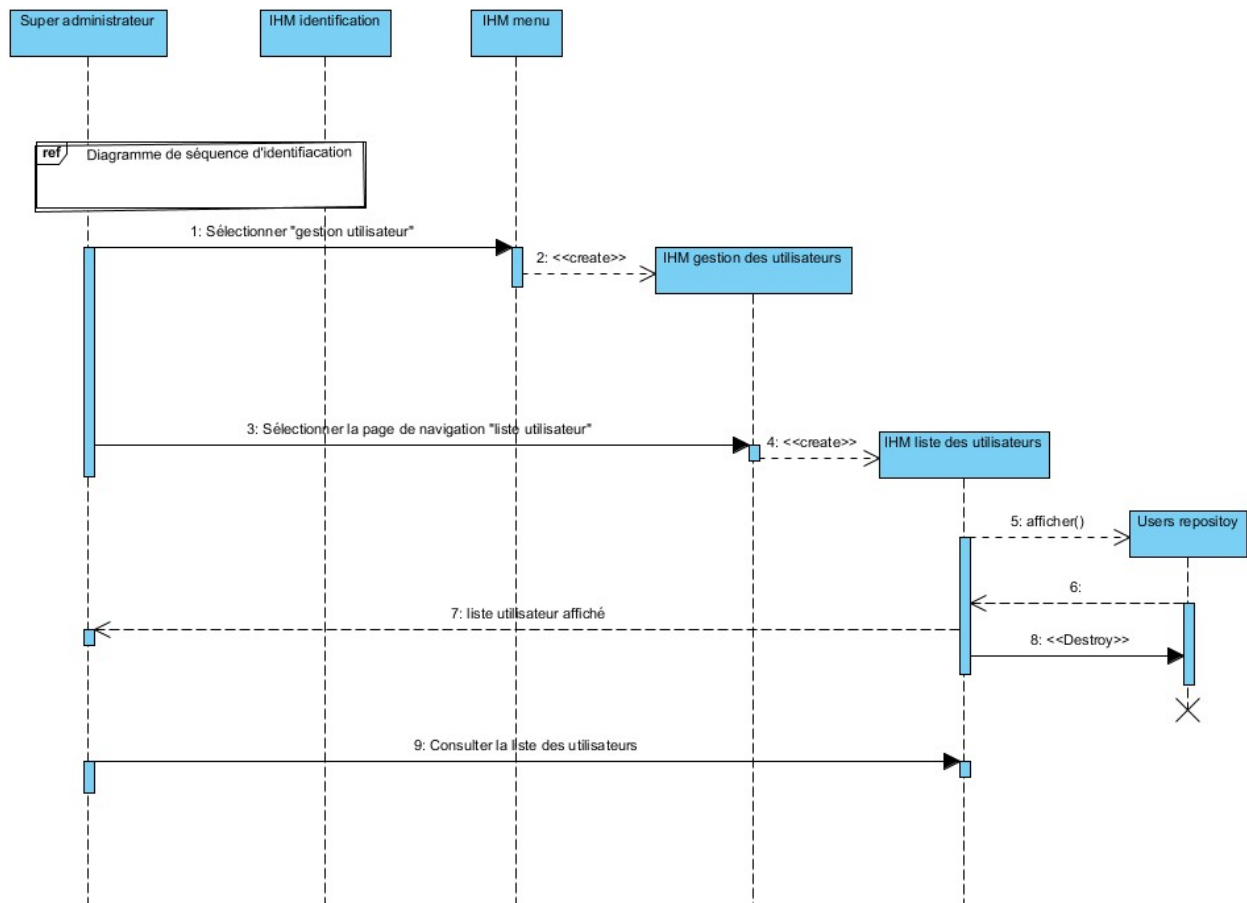


Figure 10: Diagramme de séquence " gérer liste des utilisateurs "

## 2.3 Sprint 2 : Module d'assistance

Au niveau du deuxième sprint de release, nous commençons par présenter le diagramme des classes nécessaires à l'implémentation. Ensuite, on présentera la conception avec les diagrammes de cas d'utilisation et de séquences.

- **Diagramme des classes**

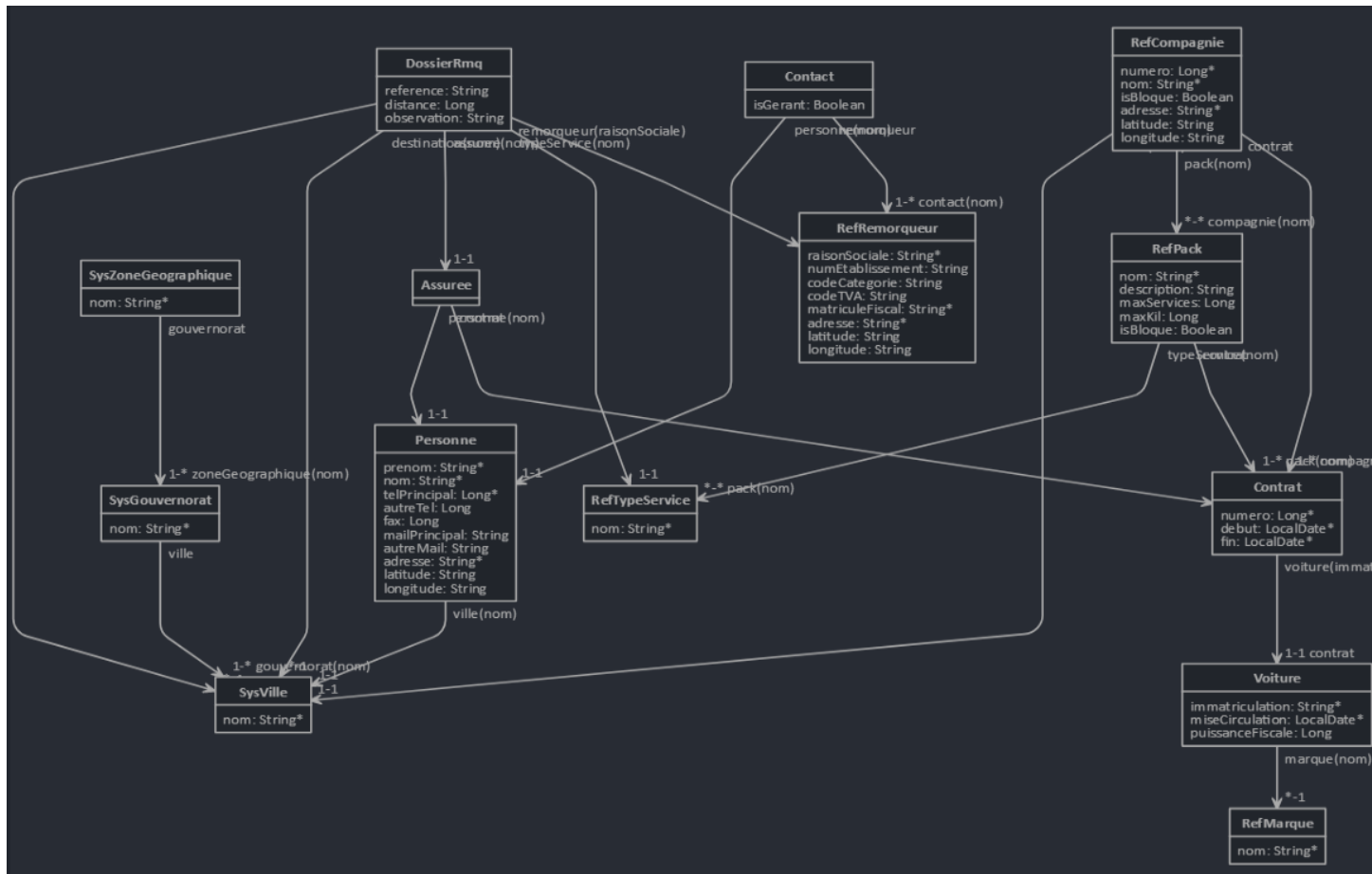


Figure 11: Diagramme de classe de module assistance

On va prendre l'exemple de gestion des remorqueurs.

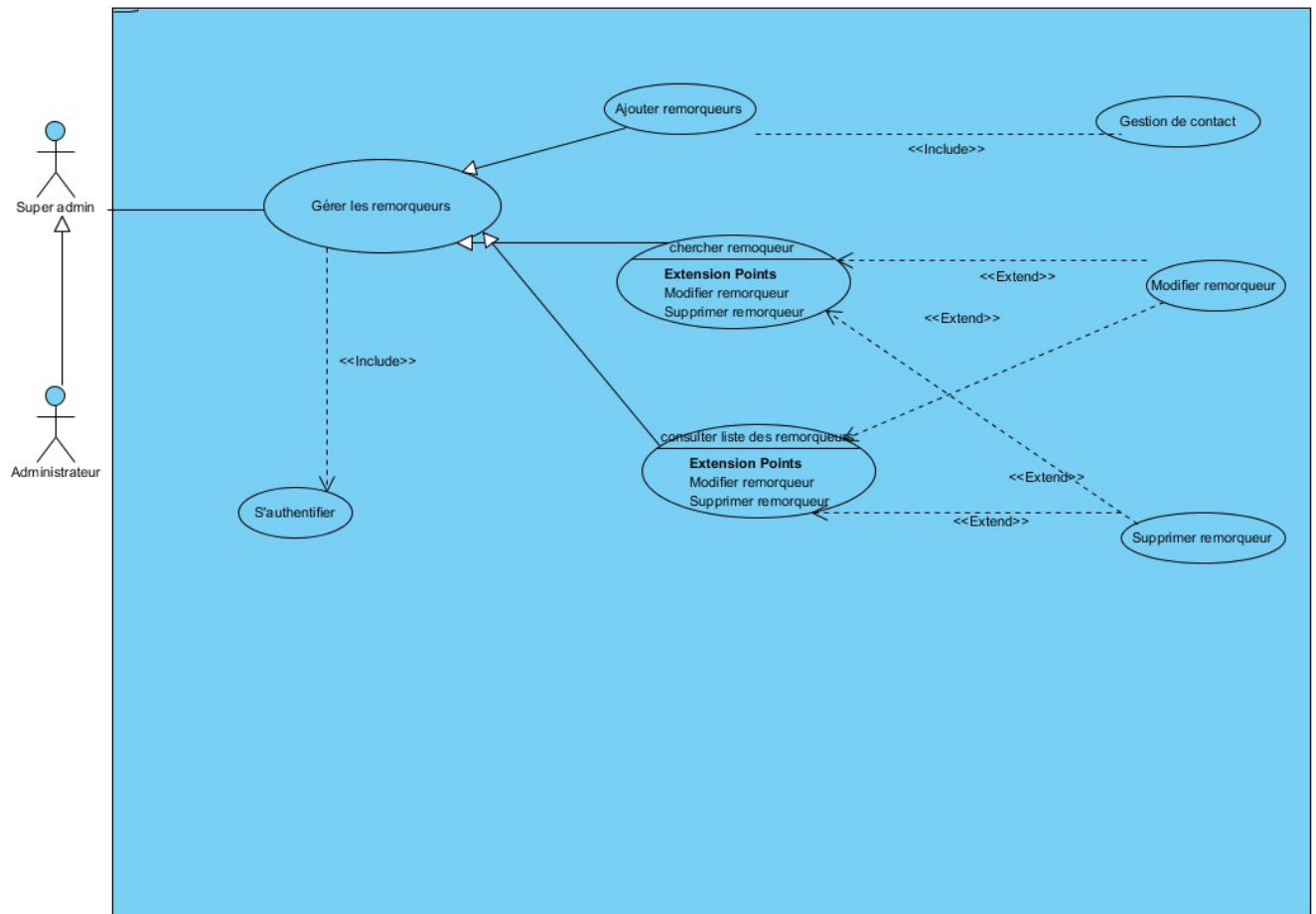


Figure 12: Diagramme de cas d'utilisation " gérer remorqueurs "

Le diagramme de cas d'utilisation présenté par la figure 12 décrit la gestion des remorqueurs par l'administrateur. Ce diagramme nous montre que l'administrateur ne peut faire sa gestion qu'après identification. Ainsi que la modification ou la suppression d'un remorqueur ne peut être effectuée qu'après consultation de la liste des remorqueurs.

### 2.3.1 Description textuelle du cas d'utilisation "ajouter un remorqueur"

Nous allons réaliser une description textuelle du scénario d'ajout d'un remorqueur pour mieux connaître les étapes à suivre pour que l'administrateur réussisse cette ajout.

Cette description est présentée par le tableau 8

<b>Titre</b>	<b>ajouter un nouveau remorqueur</b>
<b>Résumé</b>	ajouter un nouveau remorqueur à la base de donnée
<b>Acteurs</b>	Super Administrateur et administrateur
<b>Précondition</b>	S'identifier
<b>Postcondition</b>	Un nouveau remorqueur est enregistré.
<b>Scénario nominal</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur saisie les données du nouveau remorqueur.</li> <li>2. L'utilisateur appuie sur le bouton «sauvegarder »pour valider l'ajout.</li> <li>3. L'application affiche un message de succès.</li> </ol>
<b>Scénario alternatif</b>	<ol style="list-style-type: none"> <li>1. L'utilisateur effectue une erreur de saisie.</li> <li>2. L'application affiche un message d'erreur.</li> <li>3. Le scénario nominal reprend au point 1.</li> </ol>

Table 8: Tableau descriptif du cas d'utilisation "ajouter un remorqueur"



### 2.3.2 Diagramme de séquence "ajouter un remorqueur"

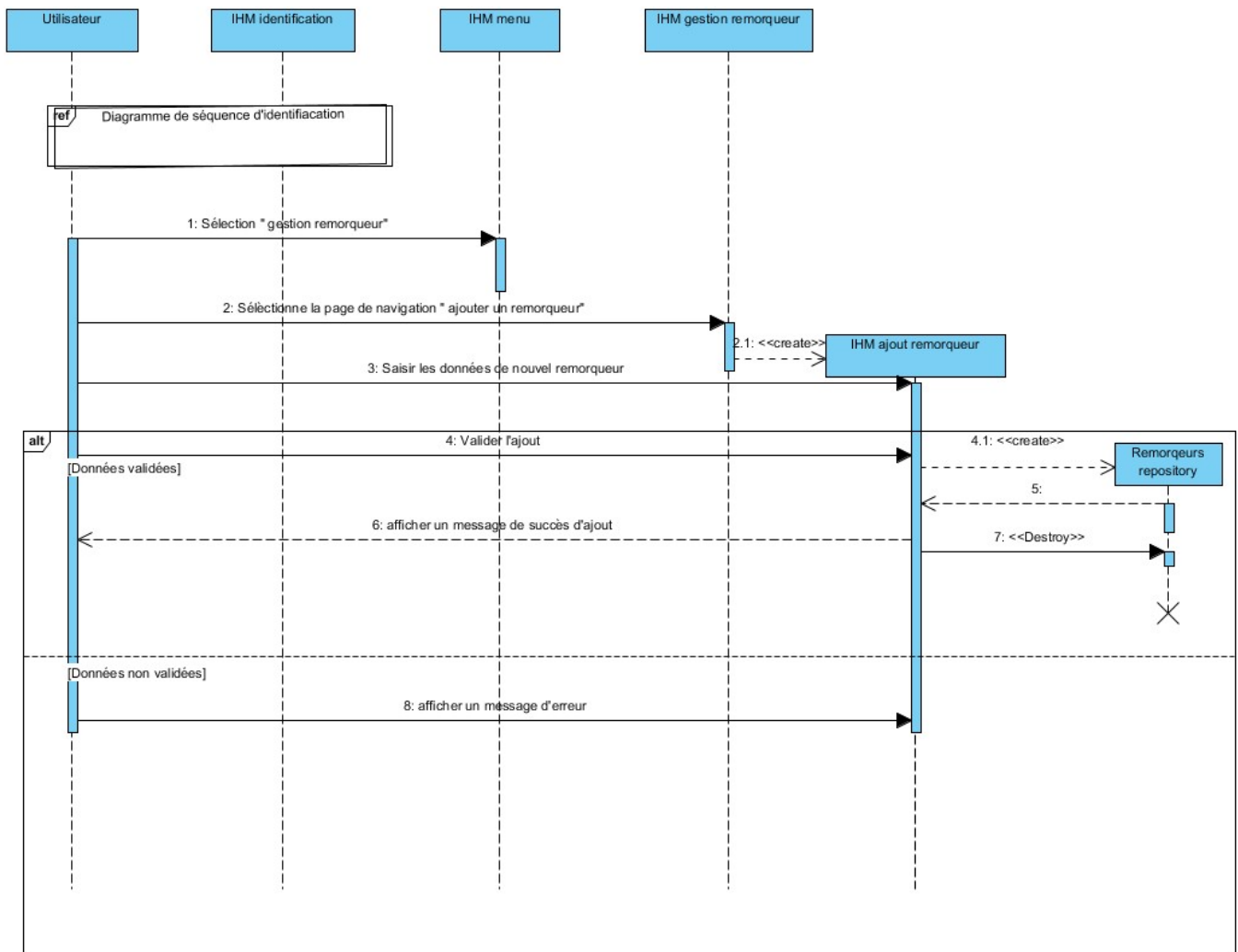


Figure 13: Diagramme de séquence " gérer remorqueurs "

**NB :** De même pour les autres classes.

## **2.4 Conclusion**

A ce stade, nous avons réussi à développer les différentes tâches des modules d'administration et d'assistance. Dans le chapitre qui suit, notre effort sera consacré pour l'intégration des modules dans une seule application, le déploiement sur un serveur physique ainsi que la présentation de quelques écrans pour la démonstration de notre système.

## **Chapitre 3 : Release 2 " Déploiement de l'application et réalisation "**

### **3.1 Introduction**

Après avoir entamé le release de la gestion administration et gestion d'assistance, nous pouvons maintenant passer au deuxième release «Déploiement et réalisation ». En effet, ce release constitue la dernière tâche de notre projet puisqu'elle permet de mettre l'application en mode production et la société peut ainsi profiter de ses fonctionnalités.

### **3.2 Sprint 1 : Intégration des modules**

L'application se constitue par deux modules :

- **Administration**
- **Assistance**

Le premier module consiste à gérer l'accès à l'application et gestion des rôles et d'utilisateur. Il s'agit de deux rôles : Super administrateur et administrateur.

Le deuxième module contient pratiquement les fonctionnalités de l'application : gestion des contact, gestion des packs , gestion de dossier d'assuré et gestion des remorqueurs.

Les deux modules sont développés séparément sur des machines différentes et par des personnes différents. Dans ce sprint, l'objectif était d'assurer une bonne intégration sans conflit pour aboutir à une application fonctionnelle. On a utilisé GIT, qui est un logiciel de gestion de versions.

On a aussi utilisé REDMINE qui est une application web libre de gestion de projets complète en mode web.

### **3.3 Sprint 2 : Réalisation**

## **ANNEXE**

**Pourquoi Scrum :**

«Scrum signifie mêlée au rugby. Scrum utilise les valeurs et l'esprit du rugby et les adapte aux projets de développement. Comme le pack lors d'un ballon porté au rugby, l'équipe chargée du développement travaille de façon collective, soudée vers un objectif précis. Comme un demi de mêlée, le Scrum Master aiguillonne les membres de l'équipe, les repositionne dans la bonne direction et donne le tempo pour assurer la réussite du projet. »[1]

Scrum est issu des travaux de deux des signataires du Manifeste Agile<sup>2</sup>, Ken Schwaber et Jeff Sutherland, au début des années 1990. Il appartient à la famille des méthodologies itératives et incrémentales et repose sur les principes et les valeurs agiles.

Le plus souvent, les experts de Scrum, même ses fondateurs, le décrivent comme un cadre ou un patron de processus orienté gestion de projet et qui peut incorporer différentes méthodes ou pratiques d'ingénierie.

S'il est difficile de définir la nature de Scrum, sa mise en place est beaucoup plus simple et peut être résumée par la Figure 14. Le principe de base de Scrum est le suivant :

- \* Dégager dans un premier lieu le maximum des fonctionnalités à réaliser pour former le backlog du produit,
- \* En second lieu définir les priorités des fonctionnalités et choisir lesquelles seront réalisées dans chaque itération,
- \* Par la suite focaliser l'équipe de façon itérative sur l'ensemble de fonctionnalités à réaliser, dans des itérations appelées Sprints,
- \* Un Sprint aboutit toujours sur la livraison d'un produit partiel fonctionnel appelé incrément.

---

<sup>2</sup>Le manifeste agile est un texte rédigé et signé en 2001 par 17 experts dans le domaine de développement d'applications informatiques.

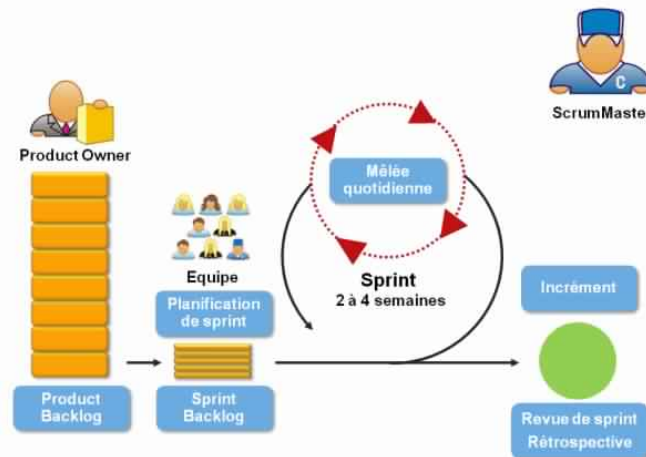


Figure 14: Le processus Scrum[2]

Le choix de Scrum comme une méthodologie de pilotage pour notre projet s'est basé sur les atouts de ce dernier. Il se résume comme suit:

- Plus de souplesse et de réactivité,
- La grande capacité d'adaptation au changement grâce à des itérations courtes,
- La chose la plus importante, c'est que Scrum rassemble les deux cotés théorique et pratique et se rapproche beaucoup de la réalité.

### Equipe et rôles (Scrum) :

«L'équipe a un rôle capital dans Scrum : elle est constituée dans le but d'optimiser la flexibilité et la productivité; pour cela, elle s'organise elle-même et doit avoir toutes les compétences nécessaires au développement du produit. Elle est investie avec le pouvoir et l'autorité pour faire ce qu'elle a à faire »<sup>3</sup>

Bref, Scrum définit trois rôles qui sont :

**Le Product Owner (le propriétaire du produit) :** c'est une personne qui porte la vision du produit à réaliser, généralement c'est un expert dans le domaine.

**Le Scrum Master (le directeur de produit) :** c'est la personne qui doit assurer le bon déroulement des différents sprints du release, et qui doit impérativement maîtriser Scrum.

<sup>3</sup>C. Aubry, SCRUM le guide pratique de la méthode agile la plus populaire, Dunod, 2010.

**Le Scrum Team (l'équipe de Scrum)** : constitué des personnes qui seront chargées d'implémenter les différents besoins du client. Bien évidemment, cette équipe sera constituée des développeurs, des testeurs, etc.

Dans le contexte de notre projet, M. Aouadi Bessem sera le directeur de produit, et M. Toukebri Mohamed Arbi sera le propriétaire et moi Choubani Amir je forme le membre de l'équipe Scrum.

## **Backlog de produit :**

**Rang** : Pour les user stories ayant la même priorité, un rang est assigné pour indiquer l'ordre dans lequel l'équipe doit les implémenter.

**Estimation** : Sert à estimer l'effort nécessaire à une équipe pour implémenter une fonctionnalité. Elle prend en compte l'effort pour le développement.

**Importance** : C'est un nombre qu'attribue le product owner à l'histoire : plus l'importance est grande, plus l'histoire devra être traitée en priorité.

**Priorité** : Le product Owner classe les user story par ordre de priorité dans le Backlog produit en travaillant avec le client pour savoir ce qui est important pour lui.

On utilise la suite de fibonacci pour attribuer les priorités.

**Description** : Elle permet de décrire les user stories. Généralement, pour rendre ce nom explicite, une bonne façon de procéder est d'utiliser le Template suivant : «En tant que X, je veuxY, afin de... ».

## **Spring Boot :**

Le framework propose une approche dogmatique de la configuration, qui permet d'éviter aux développeurs de redéfinir la même configuration à plusieurs endroits du code. Dans ce sens, Boot se veut d'être un acteur majeur dans le secteur croissant du développement d'applications rapide.

Spring Boot facilite la création d'applications et de services de production, avec un minimum d'encombrement absolu. Il prend une vue d'opinion de la plate-forme Spring afin que les utilisateurs nouveaux et existants puissent rapidement accéder aux bits dont ils ont besoin.

**Architecture MVC :**

La partie Modèle d'une architecture MVC encapsule la logique métier ainsi que l'accès aux données. Il peut s'agir d'un ensemble de fonctions ou de classes. Ceci est géré par Spring Boot.

La partie Contrôleur gère la dynamique de l'application. La partie Vue s'occupe des interactions avec l'utilisateur : présentation, saisie et validation des données. Ceci est géré par Angular. Elle fait le lien entre l'utilisateur et le reste de l'application. La demande de l'utilisateur est reçue et interprétée par le Contrôleur. Celui-ci utilise les services du Modèle afin de préparer les données à afficher. Ensuite, le Contrôleur fournit ces données à la Vue, qui les présente à l'utilisateur sous la forme d'une page HTML.



## References

- [1] <https://developers.google.com/places/javascript/?hl=fr> consulté le 12/08/17
- [2] [http://ineumann.developpez.com/tutoriels/alm/agile\\_scrum/](http://ineumann.developpez.com/tutoriels/alm/agile_scrum/) consulté le 12/08/17
- [3] <http://www.redcad.org/members/tarak.chaari/cours/coursUML.pdf> consulté le 13/08/17
- [4] <http://www.redcad.org/members/tarak.chaari/cours/coursUML.pdf> consulté le 13/08/17
- [5] <https://blog.netapsys.fr/wp-content/uploads/2017/03/> consulté le 13/08/17
- [6] <http://blog.oxiane.com/wp-content/uploads/2016/02/> consulté le 13/08/17
- [7] <http://blog.viseo-bt.com/wp-content/uploads/2012/03/> consulté le 13/08/17
- [8] <http://telecharger.cnet.com/Visual-Paradigm-for-UML/> consulté le 13/08/17