Sommaire

[**Dédicaces** 1](#_gjdgxs)

[**Remerciements** 2](#_30j0zll)

[**Introduction Générale** 4](#_1fob9te)

[Chapitre I : 6](#_3znysh7)

[Aperçu général et cadre du projet 6](#_2et92p0)

[**Introduction** 7](#_tyjcwt)

[**I.**](#_3dy6vkm) **Présentation de l’organisme d’accueil :** 7

[**1.**](#_1t3h5sf) **Historique, domaine d’activités :** 7

[**2.**](#_4d34og8) **La vision d’ALTAGEM :** 8

[**3.**](#_2s8eyo1) **Les valeurs d’ALTAGEM :** 8

[**4.**](#_17dp8vu) **Les produits :** 8

[**II.**](#_3rdcrjn) **Etude de l’existant :** 9

[**1.**](#_26in1rg) **Analyse de l’existant :** 9

[**2.**](#_lnxbz9) **Critiques de l’existant :** 9

[**3.**](#_35nkun2) **Solution proposée et objectifs :** 10

[**III.**](#_1ksv4uv) **Langage et méthodologie de conception :** 10

[**1.**](#_44sinio) **Méthodes agiles :** 11

[**2.**](#_2jxsxqh) **SCRUM :** 12

[**IV.**](#_z337ya) **Langage de modélisation UML (Unified Modeling Language) :** 12

[**Conclusion** 13](#_3j2qqm3)

[**Introduction** 15](#_1y810tw)

[**I.**](#_4i7ojhp) **Capture des besoins :** 15

[**1.**](#_2xcytpi) **Identification des acteurs :** 15

[**2.**](#_1ci93xb) **Identification des besoins :** 15

[**II.**](#_3whwml4) **Pilotage du projet avec Scrum :** 17

[**1.**](#_2bn6wsx) **Identification de l’équipe SCRUM :** 17

[**2.**](#_qsh70q) **Le Backlog du produit :** 18

[**3.**](#_3as4poj) **Diagramme des cas d’utilisation global :** 19

[**4.**](#_1pxezwc) **Planification des sprints :** 20

[**5.**](#_49x2ik5) **Prototypage des interfaces** **:** 21

[**III.**](#_2p2csry) **Environnement de travail :** 22

[**1.**](#_147n2zr) **Environnement Matériel :** 22

[**2.**](#_3o7alnk) **Environnement logiciel :** 23

[**3.**](#_23ckvvd) **Technologies** : 24

[**IV.**](#_ihv636) **Architecture générale de l’application :** 27

[**1.**](#_32hioqz) **Choix de l’architecture de l’application :** 27

[**2.**](#_1hmsyys) **Spécification de l’architecture :** 28

[**Conclusion** 29](#_41mghml)

[**Introduction** 31](#_2grqrue)

[**I.**](#_vx1227) **Backlog du sprint :** 31

[**II.**](#_3fwokq0) **Spécification fonctionnelle :** 31

[**1.**](#_1v1yuxt) **Diagramme de cas d’utilisation :** 31

[**2.**](#_4f1mdlm) **Description textuelle des cas d’utilisations :** 32

[**III.**](#_2u6wntf) **La conception :** 35

[**1.**](#_19c6y18) **Diagrammes de séquences détaillés :** 35

[**2.**](#_3tbugp1) **Diagrammes de classes :** 38

[**IV.**](#_28h4qwu) **Réalisation :** 39

[**1.**](#_nmf14n) **Les interfaces d’authentification :** 39

[**2.**](#_37m2jsg) **L’interface « Carte » :** 40

[**V.**](#_1mrcu09) **Test :** 41

[**Conclusion :** 41](#_46r0co2)

[**Introduction :** 43](#_2lwamvv)

[**I.**](#_111kx3o) **Backlog du sprint :** 43

[**II.**](#_3l18frh) **Spécification fonctionnelle :** 43

[**1.**](#_206ipza) **Diagramme de cas d’utilisation :** 43

[**III.**](#_4k668n3) **La conception :** 50

[**1.**](#_2zbgiuw) **Diagrammes de séquences détaillés :** 51

[**2.**](#_1egqt2p) **Diagrammes de classes :** 57

[**IV.**](#_3ygebqi) **Réalisation :** 58

[**1.**](#_2dlolyb) **Interface « Accueil » :** 58

[**2.**](#_sqyw64) **Interface « Consultation des plannings » :** 59

[**3.**](#_3cqmetx) **Interface « Consultation des détails de planning » :** 59

[**V.**](#_1rvwp1q) **Test :** 60

[**Conclusion** : 61](#_4bvk7pj)

[**Introduction :** 63](#_2r0uhxc)

[**I.**](#_1664s55) **Backlog du sprint :** 63

[**II.**](#_3q5sasy) **Spécification fonctionnelle :** 63

[**1.**](#_25b2l0r) **Diagramme de cas d’utilisation :** 63

[**III.**](#_kgcv8k) **La conception :** 67

[**1.**](#_34g0dwd) **Diagrammes de séquences détaillés :** 67

[**2.**](#_1jlao46) **Diagrammes de classes :** 71

[**IV.**](#_43ky6rz) **Réalisation :** 72

[**1.**](#_2iq8gzs) **Interface « Consultation des journées en attente » :** 72

[**2.**](#_xvir7l) **Interface « Consultation des journées pour une date choisit » :** 72

[**3.**](#_3hv69ve) **Interface « Consultation des détails d’une journée » :** 73

[**V.**](#_1x0gk37) **Test :** 73

[**Conclusion :** 74](#_4h042r0)

[**Introduction :** 76](#_2w5ecyt)

[**I.**](#_1baon6m) **Backlog du sprint :** 76

[**II.**](#_3vac5uf) **Spécification fonctionnelle :** 77

[**1.**](#_2afmg28) **Diagramme de cas d’utilisation :** 77

[III. La conception 84](#_pkwqa1)

[**1.**](#_39kk8xu) **Diagrammes de séquences détaillés :** 85

[**2.**](#_1opuj5n) **Diagramme de classes :** 90

[**IV.**](#_48pi1tg) **Réalisation :** 91

[**1.**](#_2nusc19) **Interface « Consultation des clients/fournisseurs » :** 91

[**2.**](#_1302m92) **Interface « Consultation des détails de client/fournisseur » :** 91

[**3.**](#_3mzq4wv) **Interface « Gérer les contacts d’un client/fournisseur » :** 92

[**V.**](#_2250f4o) **Test :** 93

[**Conclusion :** 94](#_haapch)

[**Introduction :** 96](#_319y80a)

[**I.**](#_1gf8i83) **Backlog du sprint :** 96

[**II.**](#_40ew0vw) **Spécification fonctionnelle :** 96

[**1.**](#_2fk6b3p) **Diagramme de cas d’utilisation :** 96

[**III.**](#_upglbi) **La conception :** 101

[**1.**](#_3ep43zb) **Diagrammes de séquences détaillés :** 101

[**2.**](#_1tuee74) **Diagramme de classes**: 103

[**IV.**](#_4du1wux) **Réalisation :** 104

[**1.**](#_2szc72q) **Interface « Consultation des contrats » :** 104

[**2.**](#_184mhaj) **Interface « Consultation des détails du contrat » :** 104

[**V.**](#_3s49zyc) **Test :** 105

[**Conclusion :** 105](#_279ka65)

[**Conclusion générale et Perspectives** 106](#_meukdy)

[**Référence Bibliographies** 107](#_36ei31r)

[**Liste des acronymes** 108](#_1ljsd9k)

[**Les Annexes** 109](#_45jfvxd)

[**Annexe 1 : Dictionnaire des attributs** 109](#_2koq656)

[**Annexe 2 : Identification des classes utilisé dans le sprint 1** 114](#_zu0gcz)

[**Annexe 3 : Identification des classes utilisé dans le sprint 2** 115](#_3jtnz0s)

[**Annexe 4 : Identification des classes utilisé dans le sprint 3** 117](#_1yyy98l)

[**Annexe 5 : Identification des classes utilisé dans le sprint 4** 118](#_4iylrwe)

[**Annexe 6 : Identification des classes utilisé dans le sprint 5** 119](#_2y3w247)

# **Introduction Générale**

Il y a maintenant une demande croissante pour des solutions numériques efficaces et conviviales qui améliorent l'accessibilité et la fonctionnalité des entreprises en ligne. Avec l'évolution rapide de la technologie, les entreprises sont confrontées à la nécessité d'adopter des plateformes en ligne innovantes pour rester compétitives sur le marché mondial. C'est dans ce contexte que notre projet de fin d'études vise à concevoir et développer des sites web et des applications web hautement performants, offrant une expérience utilisateur optimale et contribuant à augmenter le taux de visites et d'engagement des utilisateurs. En combinant une expertise technique approfondie avec une compréhension des besoins spécifiques de nos clients, nous aspirons à créer des solutions sur mesure qui répondent aux défis contemporains du monde numérique.

Consciente des évolutions constantes du marché numérique et des besoins émergents des utilisateurs, **Dev-IT** se positionne stratégiquement vers le développement de solutions innovantes, notamment dans le domaine des applications web. En réponse à la demande croissante de nos clients, notre projet de fin d'études se concentre sur la conception et la réalisation d'une plateforme web révolutionnaire nommée **Jannah**. Cette plateforme offre la possibilité de créer et de rassembler des mini-sites (one page) de manière intuitive et efficace. En permettant aux utilisateurs de créer rapidement des sites web attrayants tout en offrant des fonctionnalités avancées de personnalisation, **Jannah** vise à simplifier le processus de création de sites web et à répondre aux besoins variés des entreprises et des particuliers. Notre engagement est d'apporter une solution complète et flexible pour permettre à nos clients de tirer pleinement parti du potentiel du web dans leur stratégie de communication et de marketing en ligne.

Ce rapport sera organisé en sept chapitres, chacun abordant un aspect spécifique de notre projet. Dans le premier chapitre, nous introduirons le cadre général de notre étude, en présentant l'entreprise hôte, une analyse approfondie de l'état actuel du projet, ainsi que les langages et méthodologies utilisés pour sa conception, notamment le recours à la modélisation UML (Unified Modeling Language). Le deuxième chapitre sera consacré à la phase initiale de notre projet, appelée Sprint 0. Nous y identifierons les parties prenantes et leurs besoins, détaillerons la méthode de pilotage du projet selon les principes SCRUM, et décrirons l'environnement de travail qui a été mis en place, offrant ainsi un aperçu de l'architecture générale de l'application. Le troisième chapitre se concentrera sur le Sprint 1, mettant en lumière l'authentification des utilisateurs et le module de consultation des données spécifiques. Dans le quatrième chapitre, nous explorerons le Sprint 2, décrivant le développement du module de gestion des plannings, ainsi que la consultation de l'avancement et de la présence des équipes de travail. La gestion des activités réalisées au jour le jour sera examinée dans le cinquième chapitre, correspondant au Sprint 3. Le sixième chapitre abordera le Sprint 4, détaillant la mise en place de modules dédiés à la gestion des clients et des fournisseurs. Enfin, le dernier chapitre portera sur le Sprint 5, où nous explorerons la gestion des contrats et la synchronisation des données entre différentes parties de l'application..

À la clôture de ce projet, la conclusion générale mettra en lumière les résultats principaux ainsi que des perspectives prometteuses pour la suite de ce travail.

Top of Form

# Chapitre I :

# Aperçu général et cadre du projet

## **Introduction**

Dans ce chapitre, nous débuterons par une présentation de l'entreprise qui a accueilli notre projet de fin d'études. Ensuite, nous aborderons l'analyse de la situation actuelle ainsi que la solution envisagée, dont les détails seront développés ultérieurement dans le rapport. Enfin, nous clôturerons en exposant le planning des tâches à réaliser..

## **Présentation de l’organisme d’accueil :**

### **Historique, domaine d’activités :**

[](https://www.dev-itgroup.com/Accueil)

« **DevIT** » est une société d’audits informatiques, développement web et mobile qui fait partie du groupe DevITGroup. Notre mission est d'accompagner les entreprises à intégrer les nouvelles technologies de l'information en offrant des solutions et des services qui rendent leurs applications informatiques disponibles, performantes, sécurisées et évolutives.

### **La vision de Dev-IT :**

Nous sommes une équipe des jeunes très motivés et surtout talentueux, les projets qui nous sont confiés, nos clients aussi et force est de constater qu'ils nous le rendent bien ! Voici une petite présentation à partir de laquelle vous allez avoir une petite idée sur notre philosophie de travail.

### **Les valeurs Dev-IT :**

DevIT vous offre des services intégrés de stratégie et de création. notre équipe de graphistes , de stratèges Web et marketing, de développeurs et designers Web et de gestionnaires qualifiés est toujours prête à donner le meilleur sans aucune hésitation. L’esprit ouvert et créatif , nous portons un regard objectif sur votre personnalité.

### **Les produits :**

Dev-IT s'engage dans la conception et la mise en œuvre de solutions techniques sur mesure pour le développement d'applications web et mobiles, ainsi que dans l'adaptation de solutions techniques existantes. Dans ce cadre, elle assume les responsabilités suivantes :

* La qualité
* Penser client
* La passion.
* Développement Web – Mobile
* Création logiciels
* UI/UX Design
* Référencement SEO / SMO

De façon complémentaire, la société participe aussi à :

* la formation des clients lors de la livraison de l'application
* Prend en charge la correction des problèmes signalés par client
* Support technique.

Fort d'une expertise à la fois métier et technique, Dev-IT propose des solutions sur mesure à ses clients, répondant à leurs besoins variés dans divers secteurs d'activité. De plus, elle assure une communication efficace et un service de support de qualité pour toute demande ou réclamation.

## **Etude de l’existant :**

### **Analyse de l’existant :**

Notre groupe opère en Afrique du Nord, en partenariat avec des agences européennes, nous aidons nos clients à tirer parti de l'innovation informatique et à activer et à mettre à niveau leurs organisations en leur fournissant un savoir-faire.

Elle est divisée en deux partie :

* Du côté administrateur, la plateforme de Dev-IT permet une organisation efficace de la répartition de la charge de travail, un suivi en temps réel de la mise en œuvre des équipes, ainsi qu'un traçage précis de la relation client, de stratègies Web et marketing, de développeurs et designers Web et de gestionnaires qualifiés sont toujours prêts à donner le meilleur sans aucune hésitation. L’esprit ouvert et créatif , nous portons un regard objectif sur votre personnalité.
* Du côté des techniciens, Dev-IT propose une version mobile dédiée qui permet de consulter facilement leur planning, de saisir en temps réel les tâches effectuées, de partager des informations clients, ainsi que d'accéder à une base documentaire partagée.

### **Critiques de l’existant :**

Face aux mutations incessantes du marché numérique et à l'émergence de nouveaux besoins chez les utilisateurs, il est devenu impératif pour les entreprises de s'adapter et d'innover dans le domaine des applications web. Dev-IT, consciente de ces évolutions, se positionne stratégiquement pour offrir des solutions novatrices répondant aux attentes croissantes de ses clients. Notre projet de fin d'études, axé sur la conception et la réalisation de la plateforme web Jannah, témoigne de notre engagement à fournir des outils pertinents et efficaces dans un environnement en constante évolution. Jannah offre une approche révolutionnaire en permettant la création facile et rapide de mini-sites (one page), offrant ainsi aux utilisateurs la possibilité de concevoir des sites web attractifs tout en bénéficiant de fonctionnalités avancées de personnalisation. Cette démarche vise à simplifier et optimiser le processus de création de sites web, répondant ainsi aux besoins diversifiés des entreprises et des particuliers. En mettant l'accent sur la flexibilité et la complétude de notre solution, notre objectif est d'aider nos clients à exploiter pleinement le potentiel du web dans leur stratégie de communication et de marketing en ligne, répondant ainsi aux exigences actuelles du marché.

### **Solution proposée et objectifs :**

Suite aux demandes des clients nous détaillerons la plateforme révolutionnaire Jannah que nous avons développée pour répondre aux besoins changeants du marché numérique. En mettant l'accent sur la création rapide et intuitive de mini-sites (one page), Jannah offre une solution complète et flexible pour les entreprises et les particuliers désireux de maximiser leur présence en ligne. Nos objectifs principaux incluent la simplification du processus de création de sites web, l'amélioration de l'expérience utilisateur et la fourniture d'outils avancés de personnalisation. En permettant à nos utilisateurs de tirer pleinement parti du potentiel du web dans leur stratégie de communication et de marketing en ligne, notre objectif est de positionner Jannah comme une référence dans le domaine des plateformes web novatrices..

## **Langage et méthodologie de conception :**

Avant la réalisation d’un projet informatique, il est nécessaire de choisir une méthodologie de travail et un procès de suivi afin d’aboutir à la fin à un logiciel fiable. Cette méthodologie présente un procédé qui a pour objectif de permettre de formaliser les étapes préliminaires du développement d’un système afin de rendre ce développement plus fidèle aux besoins du client. Il existe plusieurs méthodes dans ce qui suit nous nous intéressons aux méthodes agiles vues leurs apports pour des projets ou les besoins évoluent.

### **Méthodes agiles :**

*« Les méthodes agiles sont des méthodologies essentiellement dédiées à la gestion de projets informatiques. Elles reposent sur des cycles de développement itératifs et adaptatifs en fonction des besoins évolutifs du client. Elles permettent notamment d'impliquer l'ensemble des collaborateurs ainsi que le client dans le développement du projet. »* [1]

Cette méthode s’agit de la réitération de cycles courts dans le temps en divisant le projet en de multiples mini-projets et les prioriser selon les besoins.

#### **Les valeurs fondamentales Agiles :**

Les méthodes agiles se reposent sur quatre valeurs fondamentales à savoir :

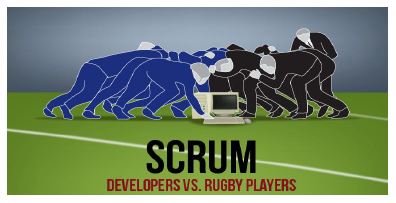
* L’interaction entre acteurs plutôt que les processus et les outils.
* Un produit opérationnel plutôt qu’une documentation pléthorique.
* La collaboration avec le client plutôt que la négociation de contrat.
* La réactivité face au changement plutôt que le suivi d'un plan.

#### **Les 12 principes de méthodes agiles :**

|  |  |  |
| --- | --- | --- |
| **1-** Satisfaire le client en livrant tôt et régulièrement des logiciels utiles qui offrent une véritable valeur ajoutée | **5-**Bâtir le projet autour de personnes motivées en leur fournissant environnement et support et en leur faisant confiance | **9-**Rechercher l’excellence technique et la qualité de la conception |
| **2-**Accepter le changement même tard dans le développement | **6**-Communiquer par des conversations en face à face | **10**-Laisser l’équipe s’auto-organiser |
| **3-**Livrer fréquemment une application qui fonctionne | **7**-Mesurer la progression avec le logiciel qui fonctionne | **11-**Rechercher la simplicité |
| **4-**Collaborer quotidiennement entre clients et développeurs | **8**-Garder un rythme de travail durable | **12**- À intervalles réguliers, réfléchir aux moyens pour devenir plus efficace |

**Tableau 1 : les 12 principes de méthodes agiles.**

### **SCRUM :**

La méthode **Scrum** est une méthode agile, créée en 2002, dont le nom est un terme emprunté au rugby qui signifie « la mêlée ». Elle s'appuie sur le découpage des projets en **itérations** encore nommées « **sprints** ».

SCRUM correspond plus à une démarche de travail qu’à une méthode. Son avantage principal est sa capacité à obtenir un résultat final dans un laps de temps court tout en s’appuyant sur une équipe cohérente. Cette équipe va s’atteler à atteindre un objectif progressif qui évoluera au cours de cycles successifs et itératifs appelés Sprints. La durée d’un sprint varie entre 15 et 30 jours, et à sa fin, l’équipe présentera un produit correspondant aux spécificités énoncées au début du cycle. Parmi les atouts du concept de sprint (court et rapide) est qu’il permet au propriétaire du produit de changer la priorisation des caractéristiques demandées au fur et à mesure de l’avancement du développement.

La fin de chaque sprint est une occasion de voir le produit courant fonctionner, et de prendre la décision de livraison ou du lancement d’un nouveau sprint d’amélioration du produit.

## **Langage de modélisation UML (Unified Modeling Language) :**

Une dizaine d'années après le début de son utilisation dans le cadre de projets de développement orienté objet, UML [2] s'est imposé comme standard. Ce langage est né de la fusion de plusieurs méthodes existant auparavant et est devenu désormais la référence en matière de modélisation objet. La modélisation objet consiste à créer une représentation informatique des éléments du monde réel auxquels on s'intéresse, sans se préoccuper de l'implémentation, ce qui signifie indépendamment d'un langage de programmation. Il s'agit donc de déterminer les objets présents et d'isoler leurs données et les fonctions qui les utilisent.

Donc, après le choix de la méthodologie, on a opté UML comme un langage de modélisation qui est utilisé dans tous les projets logiciels comporte un ensemble des diagrammes, il permet de fournir une représentation informatique d’un ensemble d’objets et de problèmes standards du monde réel.

## **Conclusion**

Ce chapitre introductif nous a permis de mettre notre projet dans son cadre général. En premier lieu, nous avons présenté l’organisme d’accueil. Par la suite nous avons passé à l’analyse détaillée de l’existant puis nous avons proposé notre solution tout en précisant les objectifs visés. Enfin nous avons présenté la méthodologie de conception adoptée pour le développement de notre système.

Le chapitre suivant sera consacré à l’étude des besoins fonctionnels et non fonctionnels, la spécification du Backlog de produit et la préparation du planning de travail.

CHAPITRE II :

SPRINT 0

## **Introduction**

Ce chapitre introduit le sprint zéro, marquant le point de départ de notre projet d'application. Nous débutons en identifiant les différents acteurs impliqués dans notre système. Ensuite, nous répertorions les besoins fonctionnels et non fonctionnels, détaillant chaque aspect selon la méthodologie préalablement sélectionnée. Enfin, nous offrons un aperçu succinct du matériel de base, des technologies, ainsi que des langages de programmation utilisés pour établir l'environnement de développement.

## **Capture des besoins :**

### **Identification des acteurs :**

|  |  |
| --- | --- |
| Acteur | Rôle |
| Dev-IT Administrateur | * Gestion des plannings. * Consultation des positions des techniciens. * Gestion des répertoires. * Gestion des journées réalisées. * Consultation du l’avancement des équipes et la présence des techniciens. * Synchroniser les données. |

**Tableau 2 : Les acteurs de notre système**

### **Identification des besoins :**

Nous nous engageons à répondre aux exigences de l'ensemble de nos utilisateurs. Dans ce qui suit, nous décrivons en détail leurs besoins fonctionnels spécifiques, ainsi que les exigences non fonctionnelles qui sont partagées par tous les acteurs impliqués dans notre application.

#### **Besoins fonctionnels :**

Les besoins fonctionnels doivent refléter les attentes du client. Dans notre situation, les besoins exprimés par l'entreprise ont été pris en compte pour garantir la satisfaction des exigences suivantes :

* **Authentification :**

Permet à l’administrateur de se connecter à travers une clé d’entreprise, son nom et un mot de passe afin de bénéficier des fonctionnalités de l'application.

* **Gestion des plannings** **:**

Permet à l’administrateur de consulter les plannings des équipes dans un intervalle de temps choisi, de supprimer une intervention, de consulter ses détails ainsi de modifier leur date.

* **Consultation des positions des techniciens :**

Permet à l’administrateur de consulter les positions des techniciens ainsi leurs activités courantes sur une carte.

* **Gestion des répertoires :**

Permet à l’administrateur de consulter, chercher et supprimer les commandes, les fiches clients et la liste des fournisseurs ainsi de contacter les clients et les fournisseurs.

* **Gestion des journées réalisées :**

Permet à l’administrateur de consulter, d’approuver ou bien rejeter les journées des techniciens.

* **Consulter l'avancement et la présence des techniciens :**

Permet à l’administrateur de consulter l’avancement des équipes pour le mois courant et la présence des techniciens.

* **Synchroniser les données :**

Permet à l’administrateur de synchroniser les données entre la base de données local et celle du serveur.

#### **Besoins non fonctionnels :**

Les besoins non fonctionnels sont des spécifications qui n’expriment pas une fonction du logiciel mais plutôt des contraintes à respecter. Ceux-ci peuvent être d’ordre technique, organisationnel ou liés à des critères de qualité. Par conséquence, notre futur système doit répondre aux critères suivants :

* **Ergonomie:** les interfaces utilisateur doivent être conviviales et faciles à utiliser.
* **Performance:** l’application mobile doit fournir un temps de réponse réduit. Elle doit être rapide lors de chargement des interfaces et des données pour cela les tests de performance doivent être implémentés et réalisés au long de cycle de développement.
* **La maintenabilité et l’évolutivité:** Notre projet doit être aligné avec les normes architecturales pour une maintenance facile et pour assurer l’évolution du système en cas d’ajout ou de modiﬁcation de besoins.
* **Sécurité:** Notre système devrait être sécurisé. En effet, lors de l’authentiﬁcation des utilisateurs de l’application, il faut prendre en compte les droits d'accès pour protéger les données. Ainsi, il faut s’assurer des données d’authentiﬁcation (notamment les jetons de sécurité pour les sessions des utilisateurs) mais également du cryptage des données au niveau de la base de données (principalement les mots de passe des comptes des utilisateurs).

## **Pilotage du projet avec Scrum :**

### **Identification de l’équipe SCRUM :**

L’équipe a un rôle capital dans Scrum : elle permet d’optimiser la productivité et la flexibilité. Afin de parvenir à ces objectifs, elle doit s’auto organiser et être multi compétente. Elle doit également avoir un champ d’action suffisant pour la soutenir dans la réalisation de son travail.

Dans le contexte de notre projet, on trouve :

|  |  |  |
| --- | --- | --- |
| Rôle | Mission | Acteur |
| Scrum Team | Conception,  Développement,  Tests et Validation. | Akremi Ramy |
| Product Owner | Définition du besoin et des fonctionnalités à développer. | M. Marchand Paul-valère |
| Scrum Master | Approbation du projet. | M.Mansouri Achref |

**Tableau 3 présentation l’équipe SCRUM**

### **Le Backlog du produit :**

Le Backlog du produit est un outil qui sert à collecter les fonctions essentielles et les raffine progressivement. C’est l’ensemble des caractéristiques fonctionnelles « **user story** » ou techniques « **technical story** » qui constituent le produit souhaité.

Un user story s’écrit comme suit :

**En tant que <rôle>**

**Je veux < liste de tâches>**

**Afin de < valeur ajoutée ou résultat>**

Pour chaque user story on identifie le rang, l’estimation, l'importance et la description.

Le tableau suivant décrire le Backlog produit de notre application et évoque les User Stories :

**++ :** Priorité moyenne  **+++ :** Priorité importante.

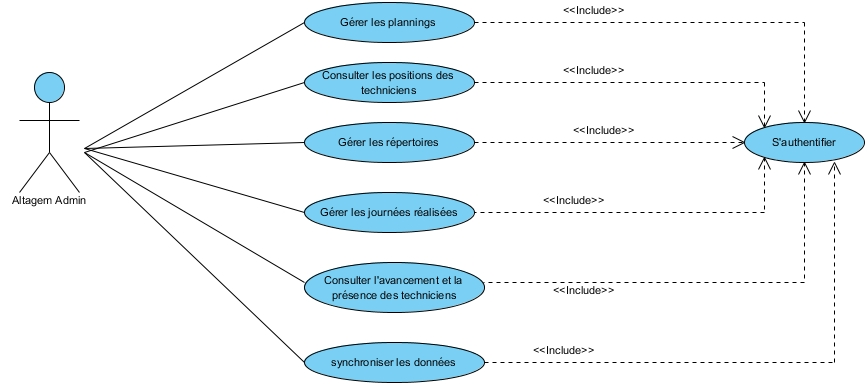
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **En tant que** | **Je veux** | **Pour que** | **Importance** |
| 1 | Altagem Admin | M’authentifier | J’accède à l’application en assurant sa sécurité | + + + |
| 2 | Altagem Admin | Gérer les plannings | -je consulte les plannings des techniciens.  -je consulte les détails des interventions.  -je gère les interventions. | + + + |
| 3 | Altagem Admin | Consulter les positions des techniciens | - je consulte les positions des techniciens sur une carte.  -je cherche un technicien. | + + + |
| 4 | Altagem Admin | Gérer les répertoires | - je consulte les commandes.  - je consulte les détails des commandes.  -je supprime les commandes.  -je cherche une commande.  - je consulte les fiches clients.  - je consulte les détails des clients.  -je supprime les clients.  -je cherche un client.  - je contact un client.  - je gère les contacts du client  - je consulte l'itinéraire vers l'emplacement du client.  - je consulte la liste des fournisseurs.  - je consulte les détails des fournisseurs.  -je supprime les fournisseurs.  -je cherche un fournisseur.  -je contact un fournisseur.  -je gère les contacts du fournisseur  -je consulte l'itinéraire vers l'emplacement du fournisseur. | + + |
| 5 | Altagem Admin | Gérer les journées réalisées | - j’approuve les journées des techniciens.  - je rejette les journées des techniciens. | + + + |
| 6 | Altagem Admin | Consulter l'avancement et la présence des techniciens | -je consulte l’avancement des équipes.  - je consulte la présence des techniciens. | + + + |
| 7 | Altagem Admin | Synchroniser les données | * Je synchronise les données entre la base de donnée local et celle du serveur. | + + + |

**Tableau 4 le Backlog du produit**

### **Diagramme des cas d’utilisation global :**

Dans ce qui suit, nous allons présenter le diagramme de cas d’utilisation global pour notre application.

Le diagramme de cas d’utilisation global est le diagramme général du système qui modélise les fonctionnalités principales du système.



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

### **Planification des sprints :**

Une fois nous avons terminé le backlog du produit, nous avons établi la réunion de planification. Le but de cette réunion est de construire le backlog de sprint en se basant sur le backlog de produit réalisé par le product owner. A la fin, nous avons identifié, les durées prévisionnelles du travail à effectuer durant chaque sprint. Pour notre projet nous avons devisé le travail sur deux releases. Pour le premier contient trois sprints, le premier est caractérisé par une durée d’une semaine et les deux autres sprints caractérisés chacun d’entre eux par une durée de 2 semaines, le deuxième release contient deux sprints aussi caractérisés chacun d’entre eux par une durée de 2 semaines. Dans le tableau suivant, nous présentons la planification des sprints.

|  |  |  |  |
| --- | --- | --- | --- |
| Release | Sprint | Contenu du sprint | Période |
| Release1 | Sprint 1 | -Authentification.  -Consulter les positions des techniciens. | De 15 avril  Au 30 avril |
| Sprint 2 | -Gestion des plannings.  -Consultation l’avancement des équipes et la présence des techniciens. | De 01 mai  Au 15 mai |
| Release2 | Sprint 3 | -Gestion des journées réalisées. | De 16 mai  Au 01 juin |
| Sprint 4 | -Gestion des clients.  -Gestion des fournisseurs. | De 02 juin  Au 16 juin |
| Sprint 5 | -Gestion des contrats.  -Synchronisation des données. | De 17 juin  Au 02 juillet |

**Tableau 5 : Planification des Sprints.**

### **Prototypage des interfaces** **:**

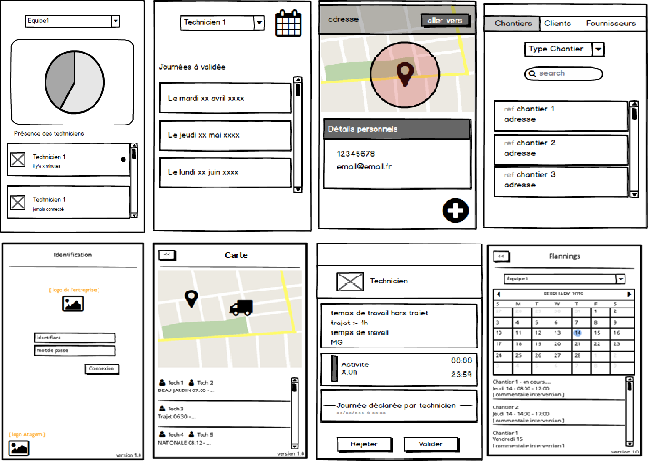
Le **prototypage** connu aussi sous l’appellation de maquettage, c’est le fait de simuler l’interface de l’application attendue d’une manière complète ou partielle afin d’avoir une idée sur l’interaction de l’utilisateur avec la future application.

Il convient de signaler d’emblée, que le but réel de cette étape est de simplifier au maximum l’accès aux informations, de détecter le plus tôt possible les problèmes d’utilisabilité et de vérifier les besoins des utilisateurs avant le codage complet pour gagner en termes de temps et du coût.

Pour réaliser cette étape, on a commencé par choisir la mise en page la plus intelligente et la plus adaptée aux objectifs de l’application attendue. Généralement, il existe de nombreux outils qui permettent de créer des maquettes mais pour ce travail, le choix est porté sur le logiciel **Mockplus**. Finalement, cette phase se termine par la création d’une **arborescence logique**, cohérente, simple et claire de la future application.

Dans cette section, nous allons illustrer quelques prototypes des interfaces de l’application pour mettre en évidence ses fonctionnalités et bien les comprendre.

**Prototypes des interfaces :**



|  |  |
| --- | --- |
| PC-Portable/RAM | Toshiba 16.00Go |
| Disque dur | 256Go SSD |
| Processeur | Intel® Core™ i5-5200U CPU @ 2.40GHz |
| Système d’exploitation | Ubuntu 18.04 |

**Figure 2 : Prototypes des interfaces.**

## **Environnement de travail :**

### **Environnement Matériel :**

Lors de la réalisation de ce projet, nous avons utilisés un ordinateur dont leur configuration sont présentées dans le tableau 6.

**Tableau 6:Caractéristiques du PC.**

En plus des ordinateurs nous avons utilisé un smartphone Samsung Galaxy Xcover 4 pour tester l’application mobile. Ce smartphone a les caractéristiques suivantes :

|  |  |
| --- | --- |
| Dimensions | 146,2 x 73,3 x 9,7 mm |
| Système d’exploitation | Android |
| Version d'OS | Android 7.0 (Nougat) |
| Taille | 5 pouces |
| Résolution | 1920 x 1200 pixels |

**Tableau 7 : Caractéristiques de Galaxy Xcover 4.**

### **Environnement logiciel :**

Dans cette partie, nous spécifions les logiciels utilisés, l’environnement de conception, l’environnement de développement mobile, ainsi que l’environnement de la base de données utilisés lors de la réalisation du notre projet.

* **Balsamiq :**

Balsamiq est l’un des plus populaire des nouveaux outils de prototypage au cours des dernières années. Il se transforme rapidement les maquettes, des images et des croquis dans des prototypes qui ressemblent et se sentent comme le projet achevé. [3]

* **Environnement de conception UML: Visual Paradigm :**

Est un outil de conception UML innovant consacré à rendre la modélisation d’un système facile et rentable en transformant les conditions et les détails d’un système en des solutions graphiques spécifiques et détaillées. Cet outil offre une gamme complète d'outils logiciels de modélisation de tous types de diagrammes, de planification, de modélisation de classe etc. Pour la conception de notre projet, nous avons utilisé la version 10.0.0. [4]

* **Environnement de développement mobile :**

 Basé sur **[IntelliJ IDEA](https://translate.googleusercontent.com/translate_c?depth=1&hl=fr&prev=search&rurl=translate.google.tn&sl=en&u=https://www.jetb)** est l'IDE officiel pour le développement et la création d'applications Android uniques et élevées [5]. Cet environnement de développement permet principalement d'éditer les fichiers [Java](https://fr.wikipedia.org/wiki/Java_(technique)), Kotlin et les fichiers de configuration d'une application Android et fournit les outils les plus rapides pour la création des applications adaptées pour tous les types d’appareil Android.

Pour le développement de notre application Android, nous avons utilisé la dernière version 3.4.

Créé en 2011 est le meilleur émulateur Android sur les ordinateurs [6]. En effet, il est l’environnement parfait pour exécuter et tester un grand nombre d’applications Android. Cet outil permet d’émuler un grand nombre d’appareils Android et il a accès à toutes ces versions. Par conséquent, les tests d’applications sont à la fois simples et efficaces. 

### **Technologies** :

Dans cette partie, nous spécifions les technologies utilisées lors de la réalisation du notre projet.

#### **Kotlin :**



Kotlin [7] est un langage de programmation orienté objet, fonctionnel avec un typage statique. Il tourne sous la machine virtuelle Java (JVM) et peut aussi être compilé sous JavaScript.De la même façon que le langage Java tire son nom de l’île indonésienne Java, le nom Kotlin vient de l’île Kotlin au large de St Pétersbourg en Russie.Annoncé durant la conférence Google I/O 2017 Kotlin est devenu le second langage de programmation supporté sur Android après évidemment Java.

#### **SQLite :**



SQLite [8] est un système de gestion de base de données relationnelle. Ce moteur de base de données est connu pour implémenter une grande partie du standard SQL-92 et des propriétés ACID.

Cette bibliothèque écrite en C est directement intégrée au programme. Ce système et son code source sont entièrement dans le domaine public ce qui permet à tout à chacun d’utiliser et de participer à l’évolution de ce projet.

#### **XML :**



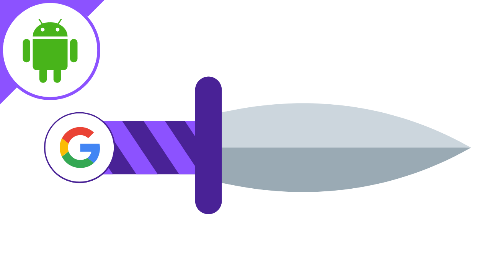
XML [9] (eXtend Markup Language) est un langage de description des documents qui utilise des balises, permet leur utilisation personnalisés et l’échange de donnés. Ce langage est utilisable lors de la création des interfaces graphiques (layout) sous Android.

#### **JSON :**



JSON [10] est un format de données textuelles, générique, dérivé de la notation des objets du langage ECMAScript. Il permet de représenter de l’information structurée. Créé par Douglas Crockford, il est décrit par la RFC 4627 de l’IETF.

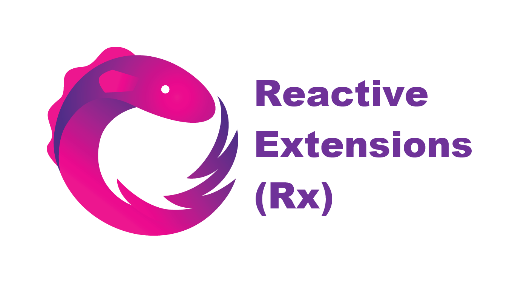
#### **Dagger :**



Dagger [11] est un framework d'injection de dépendance entièrement statique et à la compilation pour Android. Il s'agit d'une adaptation d'une version antérieure créée par Square et maintenant maintenue par Google.

Dagger vise à résoudre bon nombre des problèmes de développement et de performances qui ont entaché les solutions basées sur la réflexion.

#### **ReactiveX :**



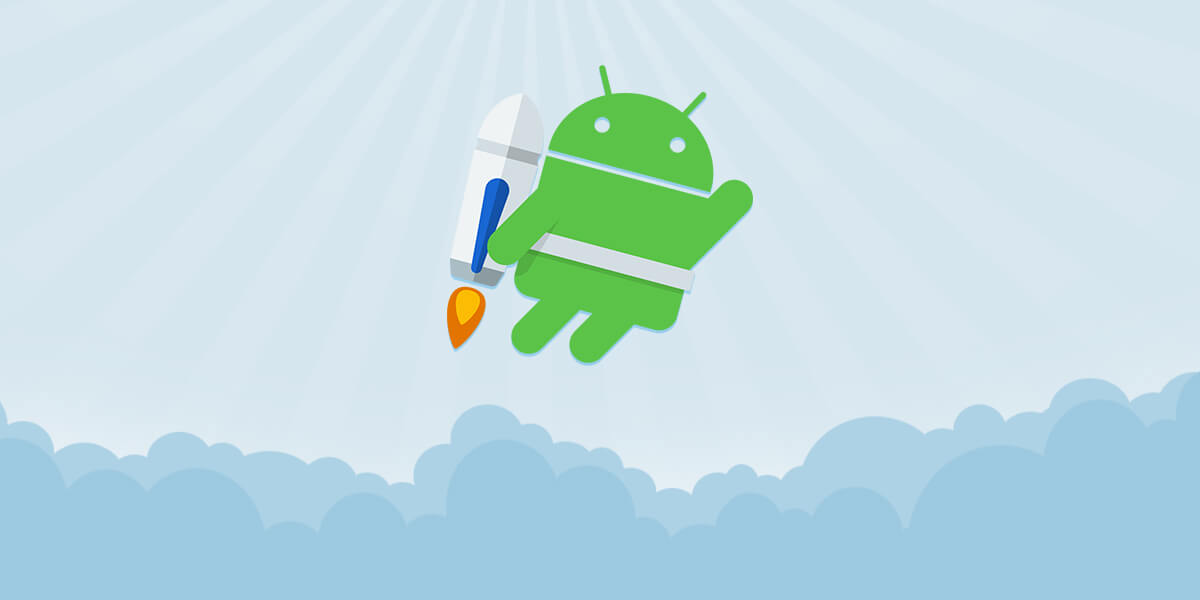
En programmation logicielle, Reactive Extensions (également appelé ReactiveX) [12] est un ensemble d’outils permettant aux langages de programmation impératifs d’opérer sur des séquences de données, que ces données soient synchrones ou asynchrones. Il fournit un ensemble d'opérateurs de séquence qui agissent sur chaque élément de la séquence. C'est une implémentation de la programmation réactive et fournit un schéma directeur pour les outils à implémenter dans plusieurs langages de programmation.

#### **Retrofit :**



Retrofit [13] est un client REST pour Android par Square. Il est relativement facile de récupérer et de télécharger du JSON (ou d’autres données structurées) via un service Web basé sur REST.

#### **Les composants d'architecture Android :**



Les composants d'architecture Android [14] sont un ensemble de bibliothèques qui vous aident à concevoir des applications robustes, testables et maintenables. Commencez avec des classes pour gérer le cycle de vie de vos composants d'interface utilisateur et gérer la persistance des données.

* **Gérez facilement le cycle de vie de votre application :** Les nouveaux composants sensibles au cycle de vie vous aident à gérer votre activité et à fragmenter les cycles de vie. Survivez aux changements de configuration, évitez les fuites de mémoire et chargez facilement des données dans votre interface utilisateur.
* Utilisez **LiveData** pour créer des objets de données qui avertissent les vues lorsque la base de données sous-jacente est modifiée.
* **ViewModel** Stocke les données liées à l'interface utilisateur qui ne sont pas détruites lors des rotations d'application.
* **Room** est une bibliothèque de mappage d'objets SQLite. Utilisez-le pour éviter le code général et convertir facilement les données de la table SQLite en objets Java. Room fournit des contrôles de compilation des instructions SQLite et peut renvoyer des observables RxJava, Flowable et LiveData.

#### **Android :**

Android [15] est un système d'exploitation open source pour les Smartphone, PDA « Personal Digital Assistant » et certaines stations mobiles. Les différentes étapes de la préparation de l’environnement de développement sous Android.

## **Architecture générale de l’application :**

### **Choix de l’architecture de l’application :**

L’architecture de notre application mobile est une architecture 3-tiers partagée entre:

* **Une base de données** comporte une importante capacité de stockage, elle doit être disponible afin qu’on puisse y accéder à tout moment, et doit avoir une puissante capacité de traitement dans le cas où plusieurs clients y accèdent en même temps.
* **Un client mobile** est une personne qui a un smartphone ayant l’application Android.
* **Un serveur web** qui répond aux requêtes des clients mobiles.

### **Spécification de l’architecture :**

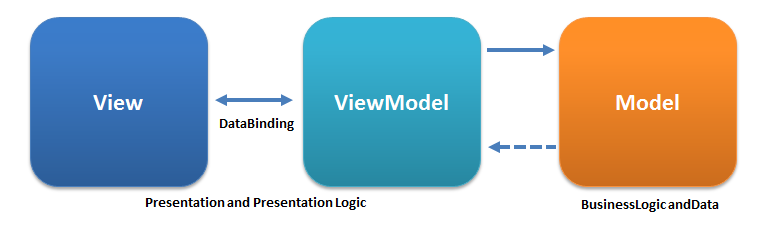
Il est indispensable à la conception de tout système informatique de choisir le modèle d'architecture qui lui sera adéquat pour assurer un bon fonctionnement et des meilleures performances.

C'est pour cette raison que nous optons pour le modèle **MVVM** qui sera également très pratique pour gérer l'interaction entre les différents composants de notre application Android. Nous décrivons cette architecture dans la section suivante.

* **Architecture** **MVVM :**

Le modèle-vue-vue modèle (en abrégé MVVM, de l'anglais Model View ViewModel) [16] est une architecture et une méthode de conception utilisée dans le génie logiciel.

Apparu en 2004, MVVM est originaire de Microsoft et adapté pour le développement des applications basées sur les technologies Windows Presentation Foundation et Silverlight via l'outil MVVM Light par exemple. 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’événement.



**Figure 3 : présentation du MVVM.**

* **Model** (Modèle en français) :

Contient les données. Généralement, ces données proviennent d’une base de données ou d’un service externe comme un API.

* **View** (Vue en français) :

La vue correspond à ce qui est affiché (Les activités et fragments dans notre cas). La vue contient les différents composants graphiques (boutons, liens, listes) ainsi que le texte.

* **ViewModel** (Vue-Modèle en français) :

Ce composant fait le lien entre le modèle et la vue. Il s’occupe de gérer les liaisons de données et les éventuelles conversions. C’est ici qu’intervient le binding.

## **Conclusion**

Dans ce chapitre nous avons identifié les besoins fonctionnels et non fonctionnels de notre système ainsi que les acteurs. Ensuite, nous avons détaillé la première étape de la méthodologie que nous avons choisie à savoir l’identification de l’équipe de travail et la réalisation de backlog du produit et des sprints. Puis nous avons présenté l’environnement matériel et logiciel que nous utiliserons pour développer notre plateforme. Dans le chapitre suivant nous entamons le développement du premier release.

CHAPITRE III :

SPRINT 1

## **Introduction**

Dans ce chapitre, nous présentons la réalisation du premier sprint, en organisant le travail sur trois phases principales qui sont l’analyse, la conception, et la réalisation.

## **Backlog du sprint :**

Le sprint est le cœur de Scrum. Il s’agit d’un bloc de temps durant lequel un incrément du produit sera réalisé. Tous les sprints d’une release ont une durée constante et ne se chevauchent jamais, c’est-à-dire qu’un sprint ne peut pas démarrer tant que le précédent n’est pas encore terminé. Avant de se lancer dans un sprint, l’équipe Scrum doit obligatoirement définir le but de ce dernier qui doit être un tableau descriptif qui précise la charge du travail pour chaque tâche en nombre de jours.

Le tableau si dessous décrit les histoires de notre backlog du sprint :

|  |  |  |
| --- | --- | --- |
| ID | Histoires | Estimation |
| 1 | S’authentifier. | 9 |
| 2 | Consulter les positions des techniciens. | 5 |

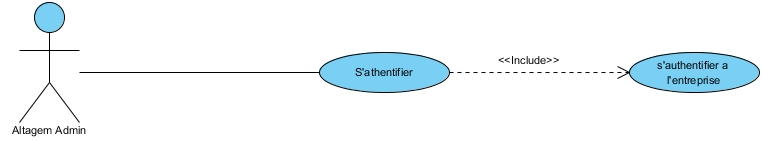
**Tableau 7 : Backlog du sprint 1.**

## **Spécification fonctionnelle :**

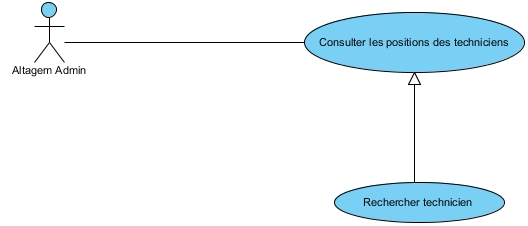
Dans cette partie nous présentons la phase d’analyse qui répond à la question « que fait le système ».

### **Diagramme de cas d’utilisation :**

Les figures ci-dessous décrit les diagrammes de cas d’utilisation du premier sprint.



**Figure 4 : Diagramme de cas d'utilisation « S'authentifier ».**



**Figure 5 : Diagramme de cas d'utilisation « Consulter les positions des techniciens ».**

### **Description textuelle des cas d’utilisations :**

* **Description textuelle du cas d’utilisation « S’authentifier » :**

Le tableau ci-dessous décrit la description textuelle du cas d’utilisation « S’authentifier »

|  |  |
| --- | --- |
| SOMMAIRE D’IDENTIFICATION | |
| CU1 | S’authentifier |
| Acteur | Altagem Admin |
| Résumé | L’admin introduit le code d’entreprise, sélection son propre nom et saisie son mot de passe pour accéder au système. |
| DESCRIPTION DES ENCHAINEMENTS SOMMAIRE | |
| Précondition | Le code d’entreprise doit être appartenu à une entreprise déjà créer et son compte doit être associé à cette entreprise. |
| Post-condition | Accès à son espace privé. |
| Scénario principal | 1. L’administrateur demande l’accès au système, 2. Le système affiche le formulaire de saisie de code d’entreprise, 3. L’administrateur introduit le code de l’entreprise, 4. Le système vérifie le champ, 5. Le système vérifie la présence du connexion internet, 6. Le système vérifie l’existence de l’entreprise, 7. Si l’entreprise est validée, le système affiche le formulaire d’authentification, 8. L’administrateur sélection son nom et introduit son mot de passe, 9. Le système vérifie le champ de mot de passe,   **10.**Si l’administrateur est identifié, le système commence le téléchargement de la base de données et vérifie le schéma,  **11.**Si la base de données est validée, le système affiche l’interface « Accueil ». |
| Scénario d'exception | -Un message d’erreur s’affichera, si la connexion internet n’est pas valable.  -Si un champ ou la liste des administrateurs est vide, un message d’erreur s’affichera.  -Si le code d’entreprise ou mot de passe non valide, le système affiche un message d’erreur.  -Si une erreur s’est produit lors du téléchargement de la base de données ou lors de la vérification de schéma, un message d’erreur s’affichera. |

**Tableau 8 : Tableau descriptif du diagramme de cas d'utilisation « S’authentifier ».**

* **Description textuelle du cas d’utilisation « Consulter les positions des techniciens » :**

Le tableau si dessous décrit la description textuelle du cas d’utilisation « Consulter les positions des techniciens »

|  |  |
| --- | --- |
| SOMMAIRE D’IDENTIFICATION | |
| CU2 | Consulter les positions des techniciens |
| Acteur | Altagem Admin |
| Résumé | L’admin consulte les différentes positions des techniciens, sélection un marqueur pour consulter la dernière activité travailler par le technicien et le temps de la dernière synchronisation. |
| DESCRIPTION DES ENCHAINEMENTS SOMMAIRE | |
| Précondition | Authentification et accès autorisé. |
| Post-condition | Consulter les différentes positions des techniciens et ses détails. |
| Scénario principal | 1. Après l’authentification, le système affiche l’interface « Accueil », 2. L’administrateur clique sur la rubrique « Carte », 3. Le système affiche les marqueurs des techniciens sur la carte, 4. L’administrateur clique sur un marqueur, 5. Le système affiche les détails de la dernière activité et le temps de la dernière synchronisation du technicien, |
| Scénario d'exception | -Un message d’erreur s’affichera, si la connexion internet n’est pas valable.  -Un message d’erreur s’affichera, si une erreur s’est produit lors de la récupération des positions. |

**Tableau 9 : Tableau descriptif du cas d'utilisation « Consulter les positions des techniciens ».**

* **Description textuelle du cas d’utilisation « Rechercher un technicien » :**

Le tableau si dessous décrit la description textuelle du cas d’utilisation « Rechercher technicien »:

|  |  |
| --- | --- |
| SOMMAIRE D’IDENTIFICATION | |
| CU3 | Rechercher un technicien |
| Acteur | Altagem Admin |
| Résumé | L’administrateur recherche un technicien en tapant son nom, le système affiche le technicien qui correspond à sa requête. |
| DESCRIPTION DES ENCHAINEMENTS SOMMAIRE | |
| Précondition | Authentification et accès autorisé. |
| Post-condition | Rechercher une équipe. |
| Scénario principal | 1. L’administrateur tape sa requête dans le formulaire de recherche, 2. Le système cherche le technicien qui correspond à sa requête, 3. Le système affiche le résultat. |
| Scénario d'exception | Si le technicien est introuvable. |

**Tableau 10 : Tableau descriptif du cas d'utilisation « Rechercher un technicien».**

## **La conception :**

Dans cette section, nous présentons les différents diagrammes de séquence détaillés pour ce sprint.

### **Diagrammes de séquences détaillés :**

Un diagramme de séquence détaillée permet une représentation détaillée des interactions entre les objets métiers de notre système selon un ordre chronologique. Nous présentons dans ce qui suit les diagrammes des séquences détaillés des histoires du premier sprint.

* **Diagramme de séquence de l’opération « S’authentifier » :**

*Figure 6 Diagramme de séquence détaillé du cas d’utilisation « S’authentifier »*

**Description :**

L’administrateur commence par la saisie du code entreprise, après la validation un contrôle se fait pour vérifier la validité de champ saisi tel que modélisé par les flèches 1,2 et 3. Si le code saisi est invalide le système affiche un message d’erreur tel que modélisé par le flèche 26. D’où l’utilisation de l’étiquette « loop ».

Si le code saisi est valide, le système vérifie l’existence de l’entreprise en appelant la fonction « getEntreprise » du ViewModel « LoginViewModel » tel que modélisé par les flèches 4,5,6 et 7.

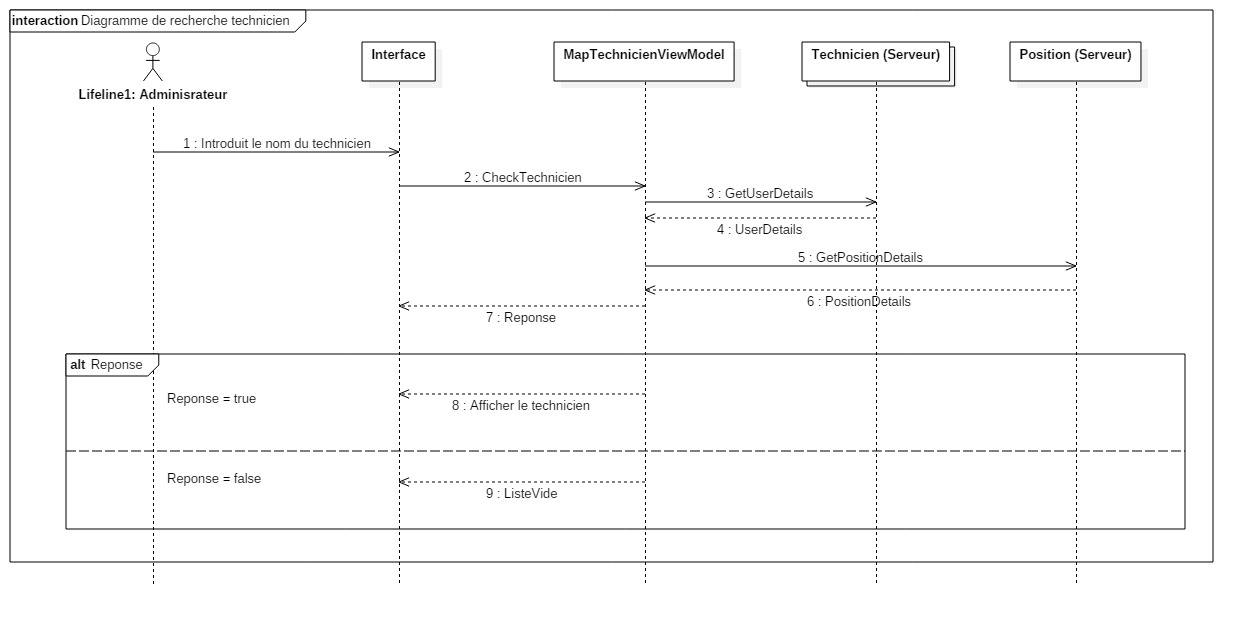
S’il s’agit d’un « succès » le système affiche l’interface d’authentification pour l’administrateur et le ViewModel appel la fonction « getAdmins » pour récupérer la liste des administrateurs tel que modélisé par les flèche 8,9,10 et 11 sinon il retourne un message d’erreur tel que modélisé par le flèche 25. Cette notion est modélisée par une étiquette « loop »

S’il s’agit d’un « succès » l’administrateur sélection son nom et introduit son mot de passe, après la validation un contrôle se fait pour vérifier la validité de champ saisi tel que modélisé par les flèches 12,13 et 14. Si le code saisi est invalide le système affiche un message d’erreur tel que modélisé par le flèche 24. D’où l’utilisation de l’étiquette « alt ».

Si le mot de passe saisi est valide, la fonction « getDatabase » du ViewModel est appelée pour commencer le téléchargement de la base de données tel que modélisé par les flèches 15 jusqu’au 20, si un problème s’est produit lors du téléchargement de la base de données le système affiche un message d’erreur tel que modélisé par le flèche 23. D’où l’utilisation de l’étiquette « alt ».

Si le téléchargement est terminé avec succès le système affiche l’interface d’accueil tel que modélisé par le flèche 21. Sinon le système affiche un message d’erreur tel que modélisé par le flèche 22. D’où l’utilisation de l’étiquette « alt ».

* **Diagramme de séquence de l’opération « Recherche un technicien » :**

**Figure 7 :Diagramme de séquence détaillé du cas d’utilisation « rechercher technicien »**.

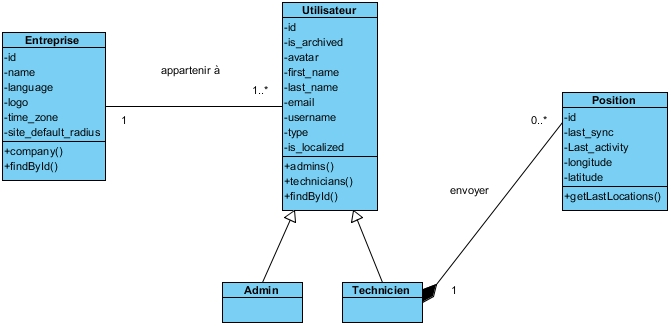
**Description :**

L’administrateur introduit le nom du technicien alors la fonction « checkTechnicien » du ViewModel « MapTechnicienViewModel » est appelé tel que modélisé par les flèches 1 et 2.

La fonction « checkTechnicien » appelle les méthodes « getUserDetails » et getPositionDetails » pour récupérer les données du technicien tel que modélisé par le flèche 3, 4et 5.

Le ViewModel récupère la réponse, s’il s’agit d’un « succès » il retourne les données au « map\_technician\_activity » qui permet d’afficher les données tel que modélisé par les flèches 6,7 et 8 sinon il retourne une liste vide tel que modélisé par le flèche 9. Cette notion est modélisée par une étiquette « alt »

### **Diagrammes de classes :**



**Figure 8 :Diagramme de classes du sprint 1.**

## **Réalisation :**

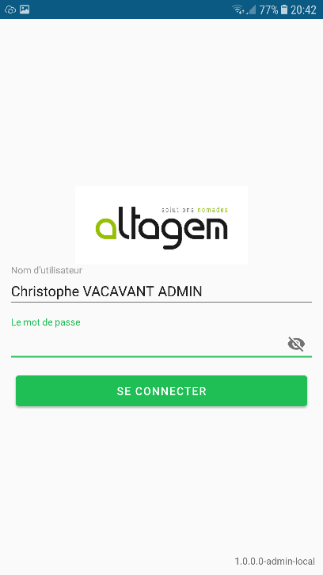
Cette partie est consacrée à l’exposition du travail achevé à travers des captures d’écrans de différentes interfaces développées pendant ce sprint.

### **Les interfaces d’authentification :**

Au démarrage de notre application, une interface « d’authentification par code entreprise » s’affiche dans laquelle l’utilisateur commence le processus d’authentification.



**Figure 9 : interface « authentification par code entreprise».**

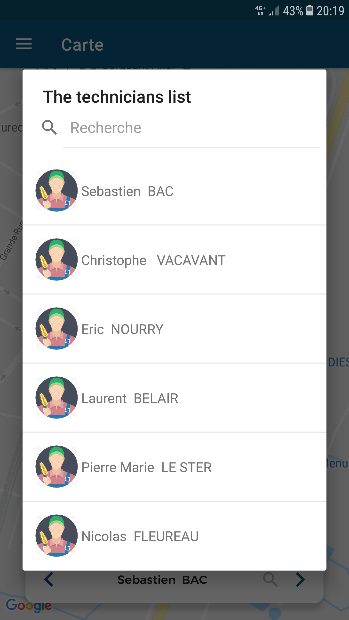


Ensuite l’interface « d’authentification administrateur » s’affiche poursuivre le processus d’authentification afin accéder à la page d’accueil.

**Figure 10 : interface « authentification administrateur ».**

### **L’interface « Carte » :**

Lors de la consultation des positions des techniciens, l’utilisateur choisit un marquer pour afficher les détails de la position du dernier.

**Figure 11 : interface « Carte ».**

Lorsque l’utilisateur clique sur le nom du technicien, une liste s’affiche pour permettre au administrateur de rechercher un technicien.

**Figure 12 interface « Recherche technicien »**

## **Test :**

Le test d’un produit logiciel est un processus consistant qui vise à garantir le bon fonctionnement du système à travers une comparaison des comportements attendu et des résultats obtenus. Avant la fin de chaque Sprint nous avons testé les fonctionnalités du module. Ensuite nous avons validé toutes les fonctionnalités avec le « Product Owner ». Pour ce fait, nous avons élaboré dans le tableau ci-dessous un ensemble de cas de scénario de tests relatif au sprint 1.

|  |  |  |  |
| --- | --- | --- | --- |
| Cas de test | Démarche | Comportement attendu | Résultats |
| Test d’authentification | Introduire un clé d’entreprise, sélection d’utilisateur et introduire un mot de passe pour accéder au système. | Accès à un espace privé. | Conforme |
| Test de la consultation des détails de position d’un technicien | Sélection un marqueur depuis la carte | Consulter les détails du technicien | Conforme |
| Test de recherche d’un technicien | Rechercher un technicien en tapant son nom | Affichage du technicien qui correspond à la requête. | Conforme |

**Tableau 11 : Tests du sprint 1.**

## **Conclusion :**

Au cours de ce chapitre, nous avons présenté le premier sprint. Pour ce faire, nous avons passé par l’analyse, la conception et la réalisation. Dans le chapitre suivant nous entamons le deuxième sprint.

CHAPITRE IV :

SPRINT 2

## **Introduction :**

Après avoir présenté notre premier sprint et atteindre ses objectifs, nous orienterons vers la deuxième.

Nous commençons par présenter le **Backlog** du sprint, ensuite nous passons à l’analyse et la conception. Nous clôturons par la partie la réalisation.

## **Backlog du sprint :**

Le tableau ci-dessous décrit les histoires de notre backlog du sprint.

|  |  |  |
| --- | --- | --- |
| **ID** | **Histoires** | **Estimation** |
| 3.1 | Consulter les plannings. | 4 |
| 3.2 | Supprimer une intervention. | 2 |
| 3.3 | Consulter les détails d’une intervention. | 3 |
| 3.4 | Modifier les dates d’une intervention. | 2 |
| 4.1 | Consulter l’avancement des équipes. | 2 |
| 4.2 | Consulter la présence des techniciens | 2 |

**Tableau 12 : Backlog du sprint 2.**

## **Spécification fonctionnelle :**

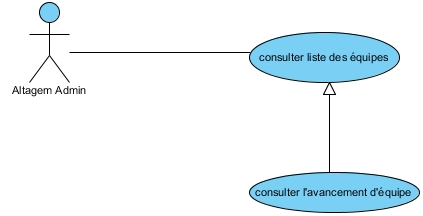
Dans cette partie, nous présentons les diagrammes de cas d’utilisation et la description textuelle de chacun d’entre eux.

### **Diagramme de cas d’utilisation :**

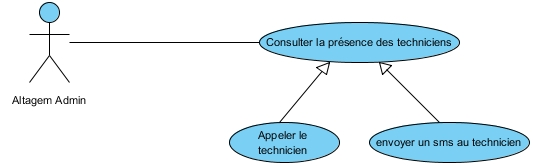
Les figures ci-dessous représentent les diagrammes de cas d’utilisation du sprint 2.



**Figure 13 : Diagramme de cas d'utilisation  « Gérer les plannings ».**



**Figure 14 : Diagramme des cas d'utilisations « Consultation l’avancement des équipes ».**



**Figure 15 : Diagramme des cas d'utilisations «  Consultation la présence des techniciens ».**

**Description textuelle des cas d’utilisations :**

* **Description textuelle du cas d’utilisation « Consulter les plannings des interventions » :**

Le tableau ci-dessous décrit la description textuelle du cas d’utilisation « Consulter les plannings des interventions ».

|  |  |
| --- | --- |
| SOMMAIRE D’IDENTIFICATION | |
| CU4 | Consulter les plannings des interventions |
| Acteur | Altagem Admin |
| Résumé | L’admin sélection une équipe puis choisit un intervalle de temps afin de consulter les plannings de l’équipe pour cette intervalle. |
| DESCRIPTION DES ENCHAINEMENTS SOMMAIRE | |
| Précondition | L’équipe doit avoir un planning pour l’intervalle de temps choisit.  Authentification et accès autorisé. |
| Post-condition | L’administrateur consulte le planning de l’équipe. |
| Scénario principal | 1. L’administrateur clique sur la rubrique « Planning », 2. Le système affiche la liste déroulante contenant les équipes et le calendrier, 3. L’administrateur choisit une équipe parmi la liste des équipes, 4. L’administrateur choisit un intervalle de temps depuis le calendrier, 5. Le système vérifie l’existence du planning, 6. Si l’équipe avait un planning, le système les affichent dans une liste. |
| Scénario d'exception | Si l’équipe n’as pas de planning. |

**Tableau 13 : Tableau descriptif du cas d'utilisation « consulter les plannings des interventions ».**

* **Description textuelle du cas d’utilisation « Supprimer une intervention » :**

Le tableau ci-dessous décrit la description textuelle du cas d’utilisation « Supprimer une intervention ».

|  |  |
| --- | --- |
| SOMMAIRE D’IDENTIFICATION | |
| CU5 | Supprimer une intervention |
| Acteur | Altagem Admin |
| Résumé | Permet à l’administrateur de supprimer des interventions. Le système effectue une vérification puis une mise à jour de la base de données. |
| DESCRIPTION DES ENCHAINEMENTS SOMMAIRE | |
| Précondition | Authentification et accès autorisé.  L’intervention cible existe. |
| Post-condition | L’intervention est supprimée de la base. |
| Scénario principal | 1. L’administrateur glisse la case de l’intervention souhaité pour la supprimer, 2. Le système affiche un message de confirmation, 3. L’administrateur confirme la suppression, 4. La nouvelle liste des interventions s’affiche. |
| Scénario d'exception | Si l’administrateur annule la suppression |

**Tableau 14 : Tableau descriptif du cas d'utilisation « supprimer une intervention ».**

* **Description textuelle du cas d’utilisation « Consulter les détails de l’intervention » :**

Le tableau ci-dessous décrit la description textuelle du cas d’utilisation « Consulter les détails de l’intervention ».

|  |  |
| --- | --- |
| SOMMAIRE D’IDENTIFICATION | |
| CU6 | Consulter les détails de l’intervention |
| Acteur | Altagem Admin |
| Résumé | Permet à l’administrateur de consulter les détails d’une intervention sélectionné. |
| DESCRIPTION DES ENCHAINEMENTS SOMMAIRE | |
| Précondition | Authentification et accès autorisé.  L’intervention cible existe. |
| Post-condition | Les détails de l’intervention sont consultés par l’administrateur. |
| Scénario principal | 1. L’administrateur appui sur une intervention, 2. Le système affiche une interface contenant les détails de l’intervention tels que les prestations, les spécifications, les membres d’équipe qui travaille sur cette intervention ainsi la date de début et de fin de l’intervention. |
| Scénario d'exception | Si l’intervention ne contient pas des détails. |

**Tableau 15 : Tableau descriptif du cas d'utilisation « consulter les détails de l'intervention ».**

* **Description textuelle du cas d’utilisation « Modifier la date d’une intervention » :**

Le tableau 19 décrit la description textuelle du cas d’utilisation « Modifier la date d’une intervention ».

|  |  |
| --- | --- |
| **SOMMAIRE D’IDENTIFICATION** | |
| CU7 | Modifier la date d’une intervention |
| Acteur | Altagem Admin |
| Résumé | Permet à l’administrateur de modifier la date de début ou de fin d’une intervention.  Le système effectue une vérification puis une mise à jour de la base de données. |
| **DESCRIPTION DES ENCHAINEMENTS SOMMAIRE** | |
| Précondition | Authentification et accès autorisé.  L’intervention cible existe.  L’état de l’intervention est prévue. |
| Post-condition | Les modifications abouties à l’intervention sont enregistrées dans la base. |
| Scénario principal | 1. L’administrateur clique sur le bouton « Modifier ». 2. Le système affiche le formulaire de modification des données. 3. L’administrateur modifie le formulaire selon les nouvelles données. 4. L’administrateur valide les modifications en cliquant sur le bouton « valider ». 5. Le système vérifie les dates choisit. 6. Le système affiche la nouvelle date. |
| Scénario d'exception | Si la date modifier est mise avant la date courante. |

**Tableau 16 : Tableau descriptif du cas d'utilisation « modifier la date d'une intervention ».**

* **Description textuelle du cas d’utilisation « Consulter l’avancement d’une équipe » :**

Le tableau 20 décrit la description textuelle du cas d’utilisation « Consulter l’avancement d’une équipe ».

|  |  |
| --- | --- |
| **SOMMAIRE D’IDENTIFICATION** | |
| CU8 | Consulter l’avancement d’une équipe |
| Acteur | Altagem Admin |
| Résumé | Permet à l’administrateur de consulter l’avancement d’une équipe dans une représentation graphique. |
| **DESCRIPTION DES ENCHAINEMENTS SOMMAIRE** | |
| Précondition | Authentification et accès autorisé.  L’équipe cible existe. |
| Post-condition | L’avancement de l’équipe est présenté dans une charte graphique. |
| Scénario principal | 1. Après l’authentification, le système affiche l’interface « Accueil », 2. L’administrateur choisit une équipe partir de la liste déroulante, 3. Suite à la sélection d’une équipe le système génère une charte qui représente le taux d’état des interventions pour le mois courant. |
| Scénario d'exception | Si l’équipe n’as pas de planning pour le mois courant. |

**Tableau 17 : Tableau descriptif du diagramme de cas d'utilisation « Consulter l’avancement d’une équipe ».**

* **Description textuelle du cas d’utilisation « Consulter la présence des techniciens » :**

Le tableau 21 décrit la description textuelle du cas d’utilisation « Consulter la présence des techniciens ».

|  |  |
| --- | --- |
| **SOMMAIRE D’IDENTIFICATION** | |
| CU9 | Consulter la présence des techniciens |
| Acteur | Altagem Admin |
| Résumé | Permet à l’administrateur de consulter la présence des techniciens en ayant la possibilité d’appeler, envoyer un sms ou bien un email au technicien. |
| **DESCRIPTION DES ENCHAINEMENTS SOMMAIRE** | |
| Précondition | Authentification et accès autorisé.  La liste des techniciens existe. |
| Post-condition | La présence des techniciens est consultée par l’administrateur. |
| Scénario principal | 1. Après l’authentification, le système affiche l’interface « Accueil », 2. Le système affiche la liste des techniciens, 3. L’administrateur consulte la présence, 4. L’administrateur choisit un technicien, 5. L’administrateur choisit le type souhaité tels que passer un appel ou bien envoyer un sms. |
| Scénario d'exception | Si la liste des techniciens est vide. |

**Tableau 18 : Tableau descriptif du diagramme de cas d'utilisation « Consulter la présence des techniciens».**

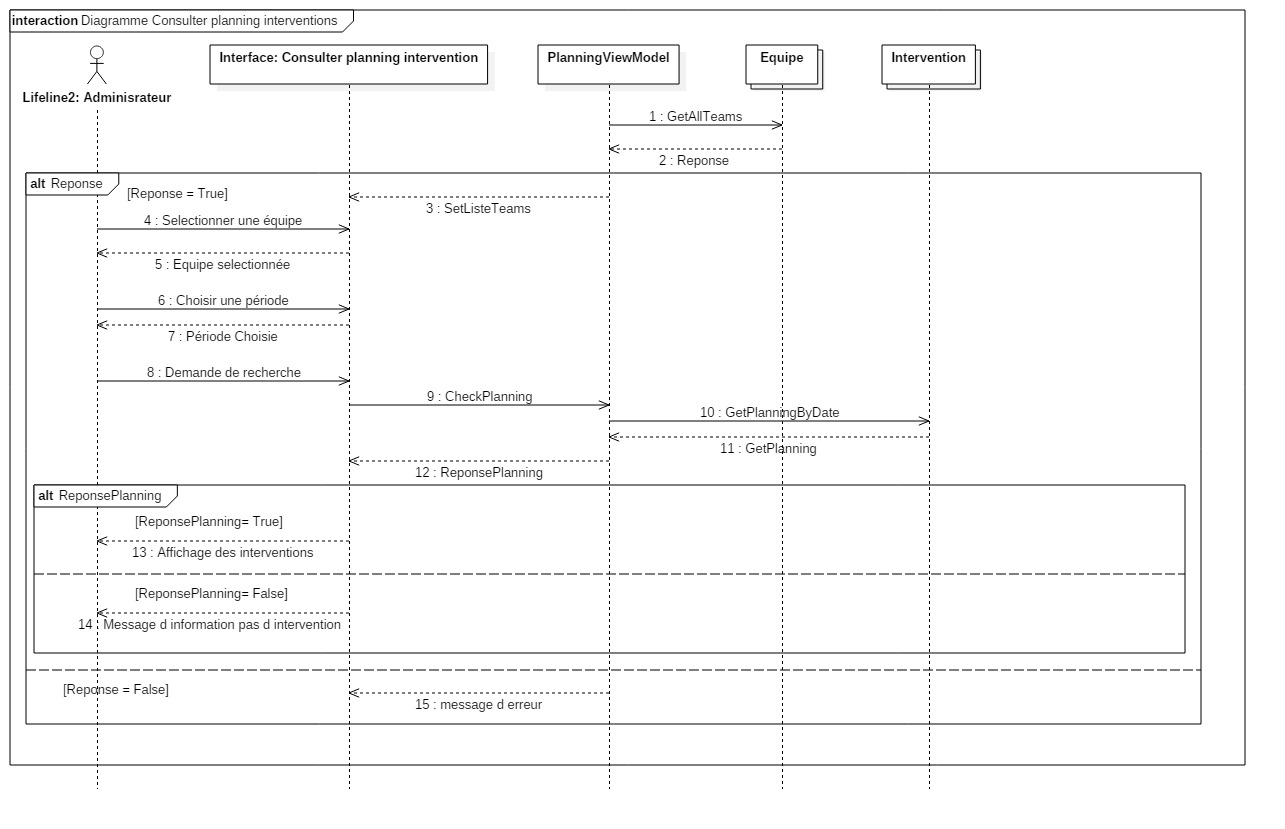
## **La conception :**

Dans cette section, nous présentons les différents diagrammes de séquence détaillés ainsi que le diagramme de classe pour ce sprint.

### **Diagrammes de séquences détaillés :**

Après avoir terminé les descriptions textuelles du diagramme de cas d’utilisation du deuxième sprint. Nous passons à présenter les diagrammes de séquence de cette partie.

* **Diagramme de séquence de l’opération « Consulter les plannings des interventions » :**



**Figure 16 : Diagramme de séquence détaillé du cas d’utilisation « consulter les plannings des interventions ».**

**Description :**

L’administrateur demande la consultation de la liste des plannings, alors la fonction « getAllTeams » du ViewModel « PlanningViewModel » est appelé tel que modélisé par les flèches 1 et 2.

Le ViewModel récupère la réponse s’il s’agit d’un « succès » il retourne les données au « Consulter planning intervention » qui permet d’afficher les données tel que modélisé par le flèche 3 sinon il retourne un message d’erreur tel que modélisé par le flèche 15. Cette notion est modélisée par une étiquette « alt »

S’il s’agit d’un « succès » pour demander les intervention, l’administrateur sélection une équipe puis il choisit une période après la fonction « checkPlanning » qui fait appel au méthode « getPlanningByDate » pour récupérer les données à partir de la table « intervention » tel que modélisé par les flèches 4 jusqu’au12.

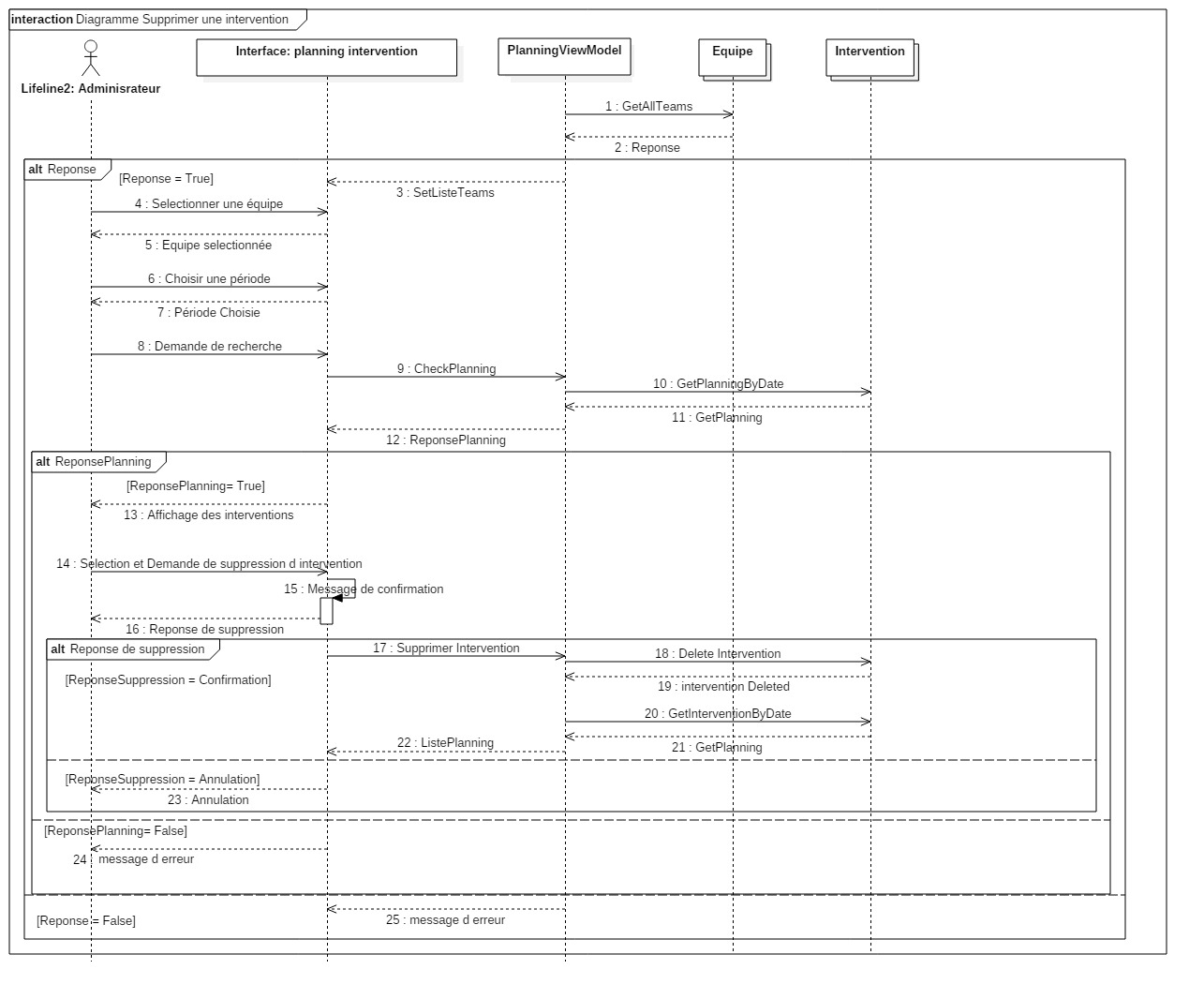
Le ViewModel récupère la réponse, s’il s’agit d’un « succès » il retourne les données au « Consulter planning intervention » qui permet d’afficher les interventions tel que modélisé par le flèche 13 sinon il retourne un message d’erreur tel que modélisé par le flèche 14. Cette notion est modélisée par une étiquette « alt »

* **Diagramme de séquence de l’opération « Supprimer une intervention » :**

**Description**

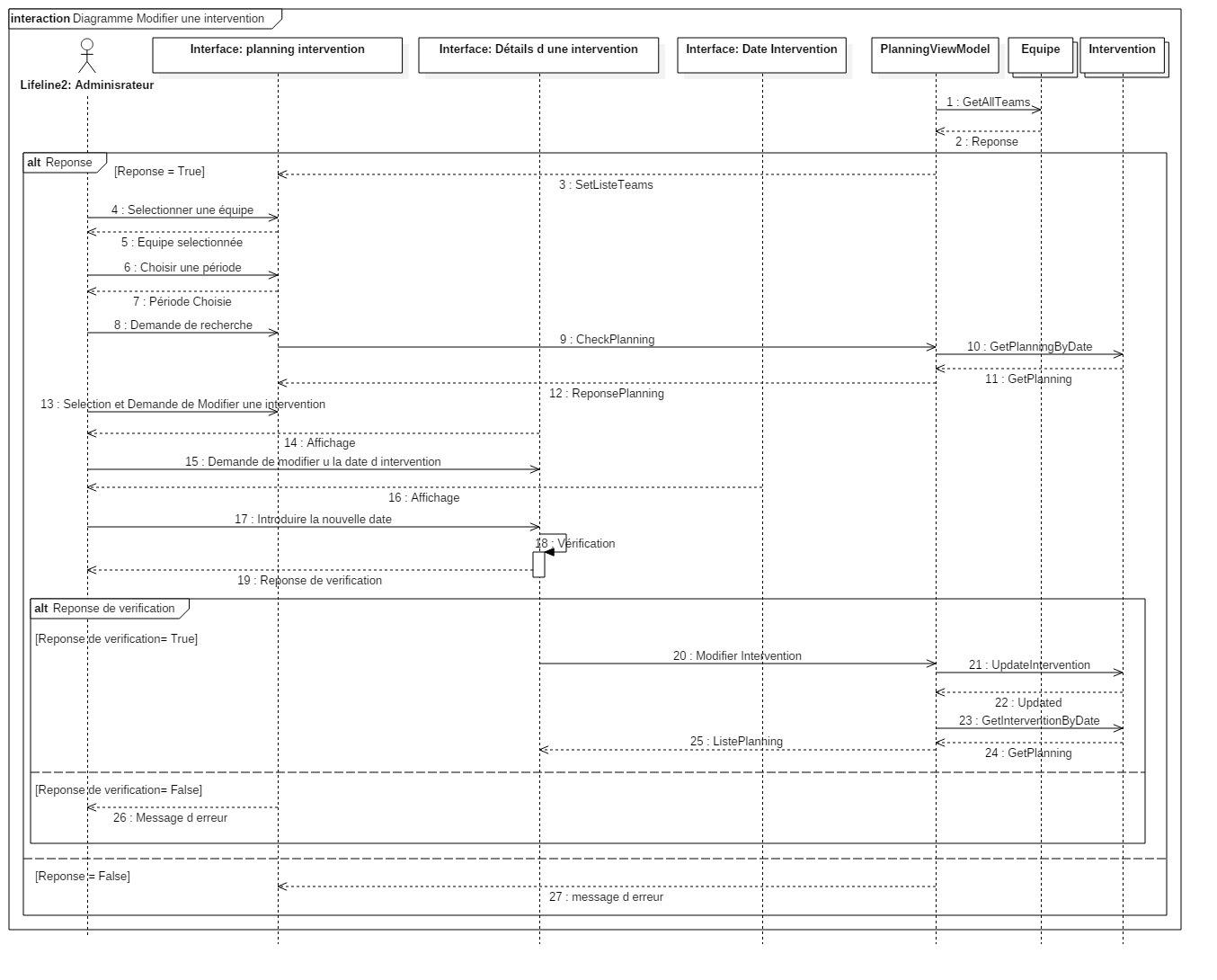
Après avoir consulter les plannings comme indiquer lors de la description ci-dessus l’administrateur glisse la case de l’intervention souhaité pour la supprimer, le système demande la confirmation de la suppression tel que modélisé par les flèches 14 et 15 sinon il l’annule tel que modélisé par le flèche 23. Cette notion est modélisée par une étiquette « alt ».

Le ViewModel récupère la réponse, s’il s’agit d’une « confirmation » la fonction « supprimer intervention » est appelé qui fait appel au méthode « removeIntervention » pour supprimer l’intervention de la table « intervention » et retourne la nouvelle liste des plannings au « Consulter planning intervention » qui permet d’afficher les interventions tel que modélisé par les flèches 16 jusqu’au 22 sinon il retourne un message d’erreur tel que modélisé par le flèche 24. Cette notion est modélisée par une étiquette « alt ».



**Figure 17 : Diagramme de séquence détaillé du cas d’utilisation « supprimer une intervention ».**

* **Diagramme de séquence de l’opération « Modifier la date d’une intervention » :**

****

**Figure 18 : Diagramme de séquence détaillé du cas d’utilisation « modifier date d'intervention ».**

**Description :**

Après avoir consulter les plannings, L’administrateur sélection une intervention pour consulter leur détail et demande de modifier la date d’intervention. Le système affiche une interface de modification après l’administrateur introduit la nouvelle date et le système vérifie la modification tel que modélisé par les flèches 13 jusqu’au 19 sinon il affiche un message d’erreur tel que modélisé par le flèche 27. Cette notion est modélisée par une étiquette « alt ».

Le ViewModel récupère la réponse, s’il s’agit d’une « validation » la fonction « modifier intervention » est appelé qui fait appel au méthode « updateIntervention » pour modifier l’intervention de la table « intervention » et retourne la nouvelle liste des plannings au « Consulter planning intervention » qui permet d’afficher les interventions tel que modélisé par les flèches 20 jusqu’au 25 sinon il retourne un message d’erreur tel que modélisé par le flèche 26. Cette notion est modélisée par une étiquette « alt ».

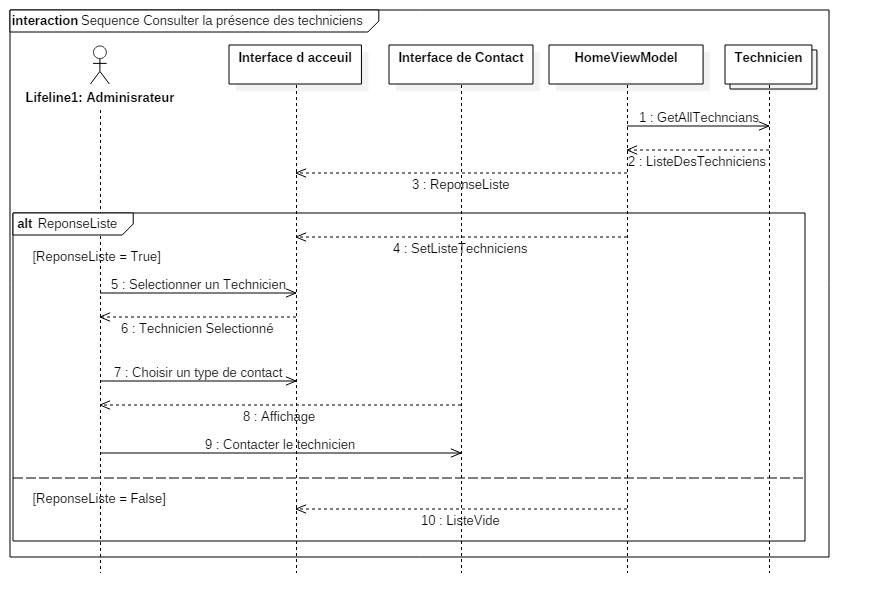
* **Diagramme de séquence de l’opération « Consulter la présence des techniciens » :**

**Description :**

L’administrateur demande la consultation de la présence des techniciens, alors la fonction « getAllTechnicians » du ViewModel « HomeViewModel » est appelé tel que modélisé par le flèche 1 et 2.

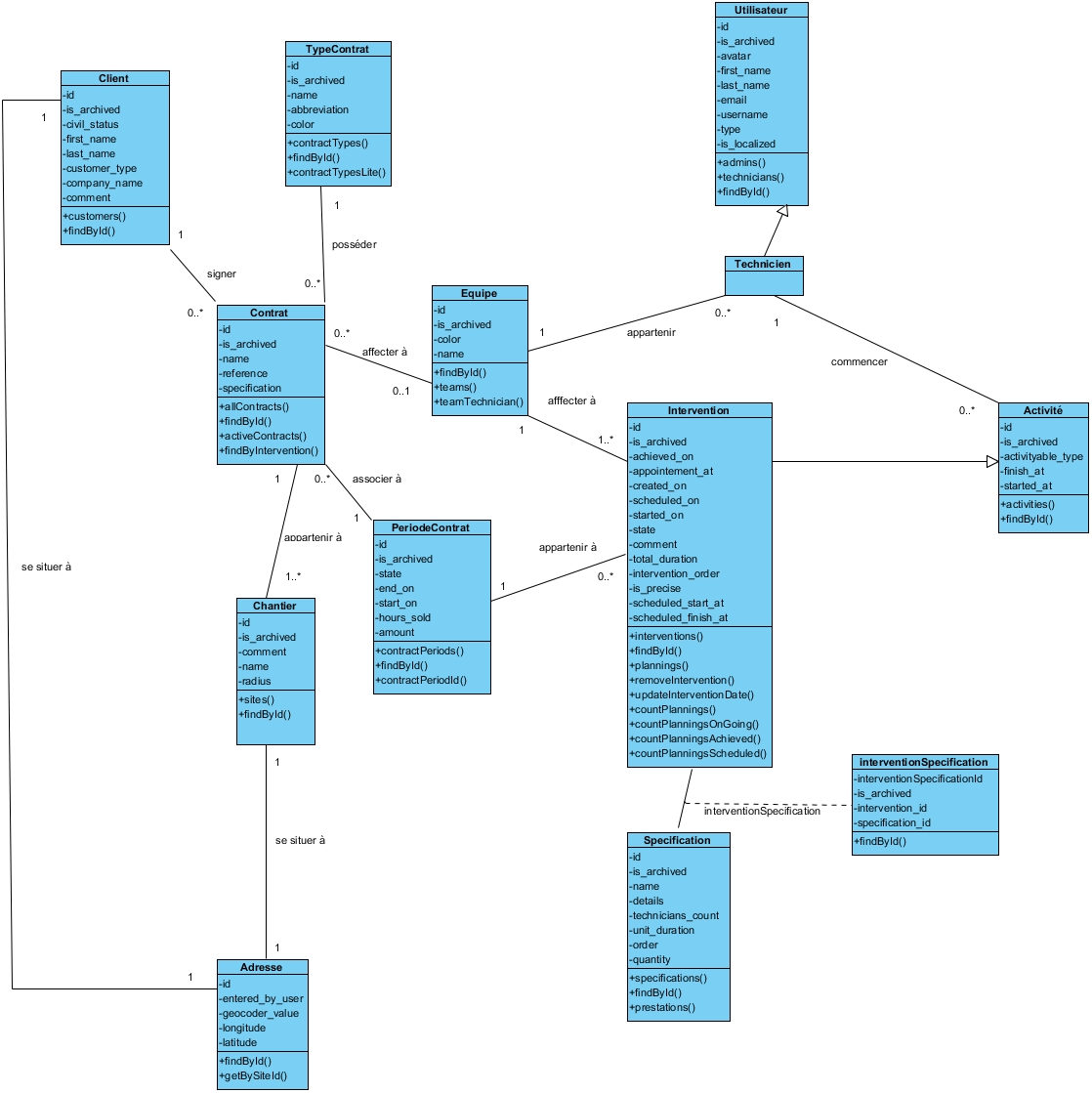
Le ViewModel récupère la réponse s’il s’agit d’un « succès » il retourne les données à l’interface « d’accueil » qui permet d’afficher les données tel que modélisé par les flèches 3 et 4 sinon il retourne une liste vide tel que modélisé par le flèche 10. Cette notion est modélisée par une étiquette « alt ».

S’il s’agit d’un « succès », pour contacter un technicien, l’administrateur sélection un technicien puis choisit un type de contact tels que passer un appel ou bien envoyer un sms comme modélisé par les flèches 5 jusqu’au 9.



**Figure 19 : Diagramme de séquence détaillé du cas d’utilisation « Consulter la présence des techniciens ».**

### **Diagrammes de classes :**



**Figure 20 : Diagramme de classes du sprint 2.**

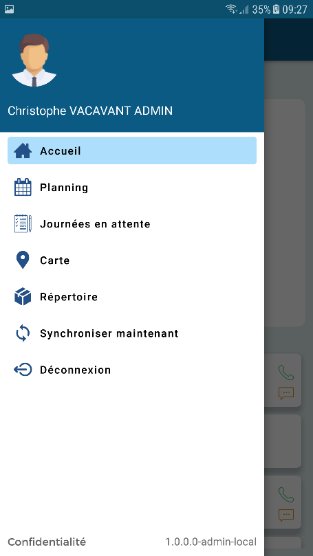
## **Réalisation :**

Cette partie est consacrée à l’exposition du travail achevé à travers des captures d’écrans de différentes interfaces développées pendant ce sprint.

### **Interface « Accueil » :**

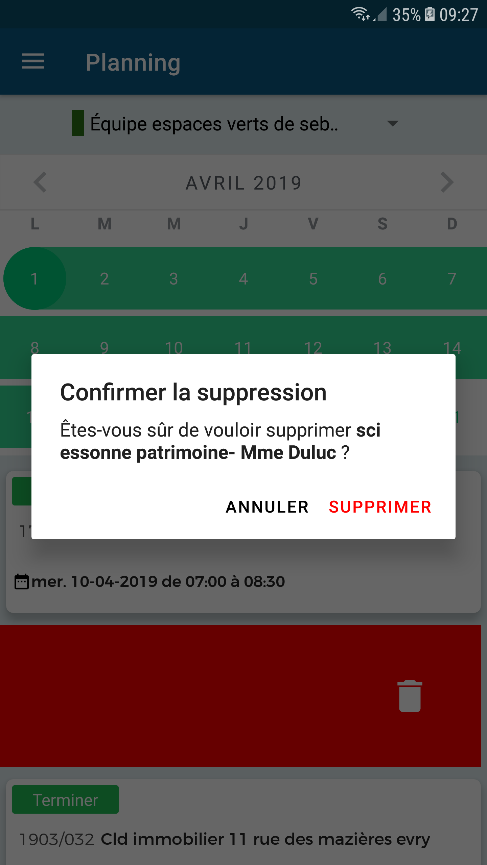
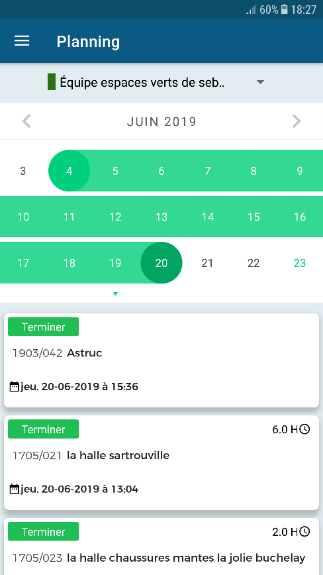
Dans cette interface d’accueil l’administrateur peut consulter l’avancement et la présence des techniciens, Ainsi le menu latéral pour naviguer vers les différents pages de l’application.

**Figure 21 : interface « Accueil ».**



**Figure 22 : interface « Menu latéral ».**

### **Interface « Consultation des plannings » :**



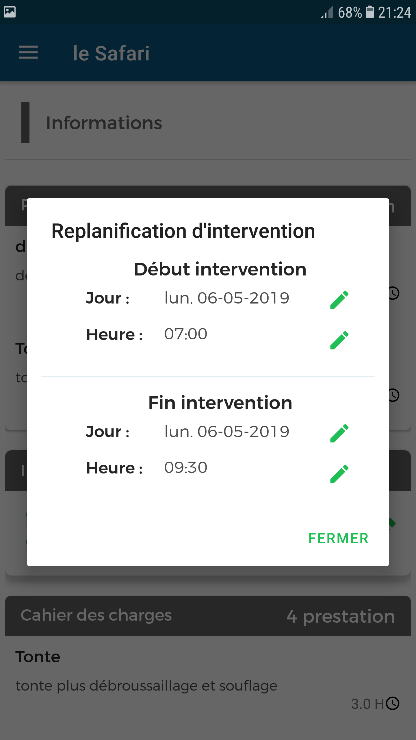
**Figure 23 : interface « Consultation des plannings » Figure 24 :interface « Supprimer intervention »**

Quand l’administrateur glisse la case de l’intervention souhaité pour la supprimer, un message de confirmation s’affiche comme indique la figure ci-dessous.

### **Interface « Consultation des détails de planning » :**



**Figure 25 : interface « Détail planning ».**



Lors du consultation d’une intervention prévue l’administrateur a la possibilité de change les dates de l’intervention.

**Figure 26 : interface « Replanification d’intervention ».**

## **Test :**

Le test d’un produit logiciel est un processus consistant qui vise à garantir le bon fonctionnement du système à travers une comparaison des comportements attendu et des résultats obtenus. Avant la fin de chaque Sprint nous avons testé les fonctionnalités du module. Ensuite nous avons validé toutes les fonctionnalités avec le Product Owner. Pour ce fait, nous avons élaboré dans le tableau ci-dessous un ensemble de cas de scénario de tests relatif au sprint 2.

|  |  |  |  |
| --- | --- | --- | --- |
| Cas de test | Démarche | Comportement attendu | Résultats |
| Test de consultation de l’avancement d’une équipe. | Choisir une équipe à partir de la liste déroulante. | Une charte contenant les détails d’avancement des techniciens pour le mois courant est généré. | Conforme |
| Test de passage d’un appel téléphonique à un technicien à partir de la liste des présence. | Passer un appel téléphonique pour un technicien choisit | le composeur téléphonique s’ouvre en affichant le numéro de téléphone du technicien choisit | Conforme |
| Test d’envoi d’un sms à un technicien à partir de la liste des présence. | Envoyer un sms à un technicien choisit | la messagerie téléphonique s’ouvre en ajoutant le numéro de téléphone du technicien choisit comme destinataire. | Conforme |
| Test de consultation des plannings | Choisir une équipe à partir de la liste déroulante.  Choisir un intervalle de temps depuis le calendrier | La liste des plannings s’affiche | Conforme |
| Test de suppression d’une intervention | Supprimer une intervention | L’intervention est supprimé de la base | Conforme |
| Test de consultation des détails d’une intervention | Choisir une intervention | Les détails de l’intervention sont affichées. | Conforme |
| Test de modification de date d’une intervention prévue. | Modifier les données existantes | Les modifications abouties au intervention sont enregistrées dans la base | Conforme |

**Tableau 19 : Tests du sprint 2.**

Après avoir vérifié le bon fonctionnement du code et des fonctionnalités, le premier livrable a été

Testé par le Product Owner. Ainsi, il nous a transmis la liste des améliorations, des fonctionnalités à ajouter et des bugs à signaler.

## **Conclusion** :

Au cours de ce chapitre, nous avons présenté le premier sprint. Pour ce faire, nous avons passé par l’analyse, la conception et la réalisation. Dans le chapitre suivant nous entamons le troisième sprint.

CHAPITRE V :

SPRINT 3

## **Introduction :**

Dans le chapitre précédent, nous avons présentait notre deuxième sprint. A l’issue de ce dernier nous avons donc obtenu une deuxième version de notre application.

Dans ce chapitre nous étudions en premier temps la spécification fonctionnelle. En second temps la conception détaillée et enfin la réalisation.

## **Backlog du sprint :**

Le tableau ci-dessous décrit les histoires de notre backlog du sprint.

|  |  |  |
| --- | --- | --- |
| **ID** | **Histoires** | **Estimation** |
| 5.1 | Consulter les journées en attente. | 3 |
| 5.2 | Consulter les journées d’une date choisit. | 3 |
| 5.3 | Consulter les détails d’une journée. | 4 |
| 5.4 | Évaluer une journée d’un technicien. | 5 |

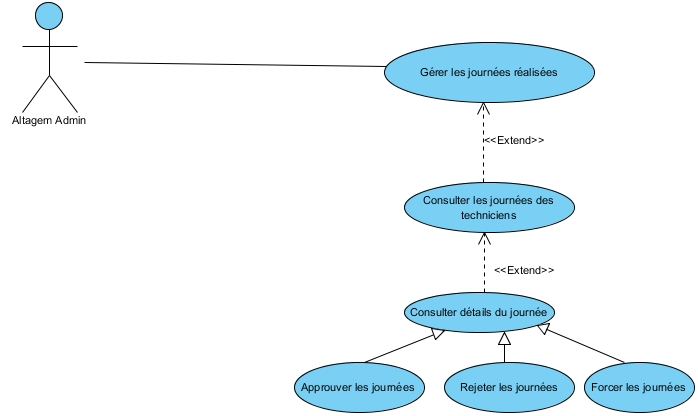
**Tableau 20 : Backlog du sprint 3.**

## **Spécification fonctionnelle :**

Dans cette partie, nous présentons les diagrammes de cas d’utilisation et la description textuelle de chacun d’entre eux.

### **Diagramme de cas d’utilisation :**

La figure 27 représentent le diagramme de cas d’utilisation global du sprint 3.



**Figure 27 : Diagramme des cas d'utilisations« Gérer les journées réalisées».**

**Description textuelle des cas d’utilisations :**

* **Description textuelle du cas d’utilisation « Consulter les journées en attente » :**

Le tableau ci-dessous décrit la description textuelle du cas d’utilisation « Consulter les journées en attente ».

|  |  |
| --- | --- |
| SOMMAIRE D’IDENTIFICATION | |
| CU10 | Consulter les journées en attente |
| Acteur | Altagem Admin |
| Résumé | L’admin sélection un technicien à partir du la liste déroulante des techniciens pour consulter les journées en attente du dernier. |
| DESCRIPTION DES ENCHAINEMENTS SOMMAIRE | |
| Précondition | Authentification et accès autorisé.  Au moins un technicien doit avoir une journée en attente. |
| Post-condition | L’administrateur consulte les journées en attente du technicien. |
| Scénario principal | 1. L’administrateur clique sur la rubrique « Journées en attente », 2. Le système affiche la liste déroulante contenant la liste des techniciens, 3. L’administrateur choisit un technicien, 4. Le système affiche la liste des journées en attente du dernier, |
| Scénario d'exception | Si les techniciens n’ont pas des journées en attente. |

**Tableau 21 : Tableau descriptif du cas d'utilisation « consulter les journées en attente».**

* **Description textuelle du cas d’utilisation « Consulter les journées d’une date**

**Choisit » :**

Le tableau ci-dessous décrit la description textuelle du cas d’utilisation « Consulter les journées d’une date choisit ».

|  |  |
| --- | --- |
| SOMMAIRE D’IDENTIFICATION | |
| CU11 | Consulter les journées d’une date choisit |
| Acteur | Altagem Admin |
| Résumé | L’admin sélection une date à partir du calendrier afin de consulter les journées de la date choisit. |
| DESCRIPTION DES ENCHAINEMENTS SOMMAIRE | |
| Précondition | Authentification et accès autorisé. |
| Post-condition | L’administrateur consulte les journées d’une date choisit. |
| Scénario principal | 1. L’administrateur clique sur le calendrier, 2. L’administrateur choisit une date 3. Le système affiche la liste des journées pour la date choisit. |
| Scénario d'exception | S’il n’y a pas des journées pour la date choisit. |

**Tableau 22 : Tableau descriptif du cas d'utilisation « consulter les journées d'une date choisit ».**

* **Description textuelle du cas d’utilisation « Consulter les détails d’une journée réalisée » :**

Le tableau ci-dessous décrit la description textuelle du cas d’utilisation « Consulter les détails d’une journée réalisée ».

|  |  |
| --- | --- |
| **SOMMAIRE D’IDENTIFICATION** | |
| CU12 | Consulter les détails d’une journée réalisée |
| Acteur | Altagem Admin |
| Résumé | L’admin sélection une journée à partir de la liste des journées afin de consulter leurs détails. |
| **DESCRIPTION DES ENCHAINEMENTS SOMMAIRE** | |
| Précondition | Authentification et accès autorisé. |
| Post-condition | L’administrateur consulte les détails d’une journée choisit. |
| Scénario principal | 1. L’administrateur choisit une journée, 2. Le système affiche une interface contenant les détails du journée tels que la listes des activités réalisées, l’historique du journées et des rubrique sur le temps du travail. |
| Scénario d'exception | Si la liste des journées réalisées est vide. |

**Tableau 23 : Tableau descriptif du cas d'utilisation « consulter les détails d'une journée réalisée ».**

* **Description textuelle du cas d’utilisation « Évaluer une journée d’un technicien » :**

Le tableau ci-dessous décrit la description textuelle du cas d’utilisation « Évaluer une journée d’un technicien ».

|  |  |
| --- | --- |
| **SOMMAIRE D’IDENTIFICATION** | |
| CU13 | Évaluer une journée d’un technicien |
| Acteur | Altagem Admin |
| Résumé | Après la consultation de détails d’une journée l’admin l’évalue en choisissant entre valider, rejeter ou bien forcer (valider sans attendre la signature du technicien) la journée du technicien. |
| **DESCRIPTION DES ENCHAINEMENTS SOMMAIRE** | |
| Précondition | Authentification et accès autorisé.  Présence d’une connexion internet. |
| Post-condition | L’administrateur évalue la journée d’un technicien . |
| Scénario principal | 1. Après avoir consulter les détails de la journée choisit l’administrateur choisit le bouton convenable pour son évaluation de la journée, 2. Le système affiche un dialogue du saisie d’un commentaire, 3. L’administrateur introduit son commentaire sur la journée, 4. L’administrateur confirme son commentaire. |
| Scénario d'exception | Lors de l’absence d’une connexion internet. |

**Tableau 24 : Tableau descriptif du cas d'utilisation « Évaluer une journée d’un technicien».**

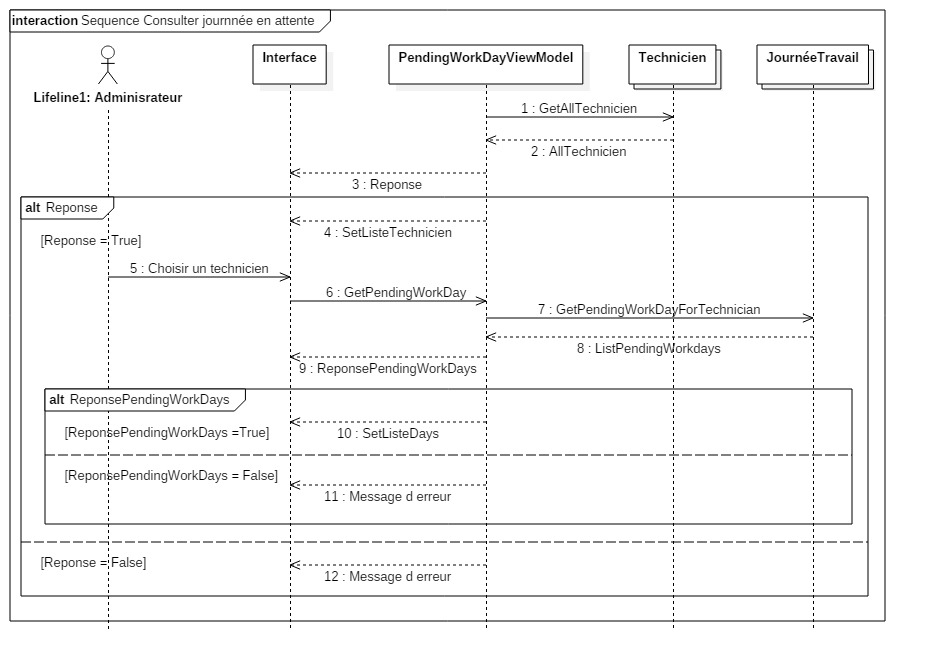
## **La conception :**

Dans cette section, nous présentons les différents diagrammes de séquence détaillés ainsi que le diagramme de classe pour ce sprint.

### **Diagrammes de séquences détaillés :**

Après avoir terminé les descriptions textuelles du diagramme de cas d’utilisation du troisième sprint. Nous passons à présenter les diagrammes de séquence de cette partie.

* **Diagramme de séquence de l’opération « Consulter les journées en attente » :**



**Figure 28 :Diagramme de séquence détaillé du cas d’utilisation « Consulter les journées en attente».**

**Description :**

L’administrateur demande la consultation des journées en attente, alors la fonction « getAllTechnicians » du ViewModel « PendingWorkDayViewModel » est appelé tel que modélisé par les flèches 1 et 2.

Le ViewModel récupère la réponse, s’il s’agit d’un « succès » il retourne les données au « Consulter pending work day » qui permet d’afficher les données tel que modélisé par les flèches 3 et 4 sinon il retourne un message d’erreur tel que modélisé par le flèche 12. Cette notion est modélisée par une étiquette « alt »

S’il s’agit d’un « succès », l’administrateur sélection un technicien après la fonction « getPendingWorkDay » qui fait appel au méthode « getPendingWorkDayForTechnician » pour récupérer les données à partir du table « JournéeTravail » tel que modélisé par les flèches 4 jusqu’au 9.

Le ViewModel récupère la réponse s’il s’agit d’un « succès » il retourne les données au « Consulter pending work day » qui permet d’afficher les journées en attente tel que modélisé par le flèche 10 sinon il retourne un message d’erreur tel que modélisé par le flèche 11. Cette notion est modélisée par une étiquette « alt ».

* **Diagramme de séquence de l’opération « Consulter les journées d’une date choisit »**



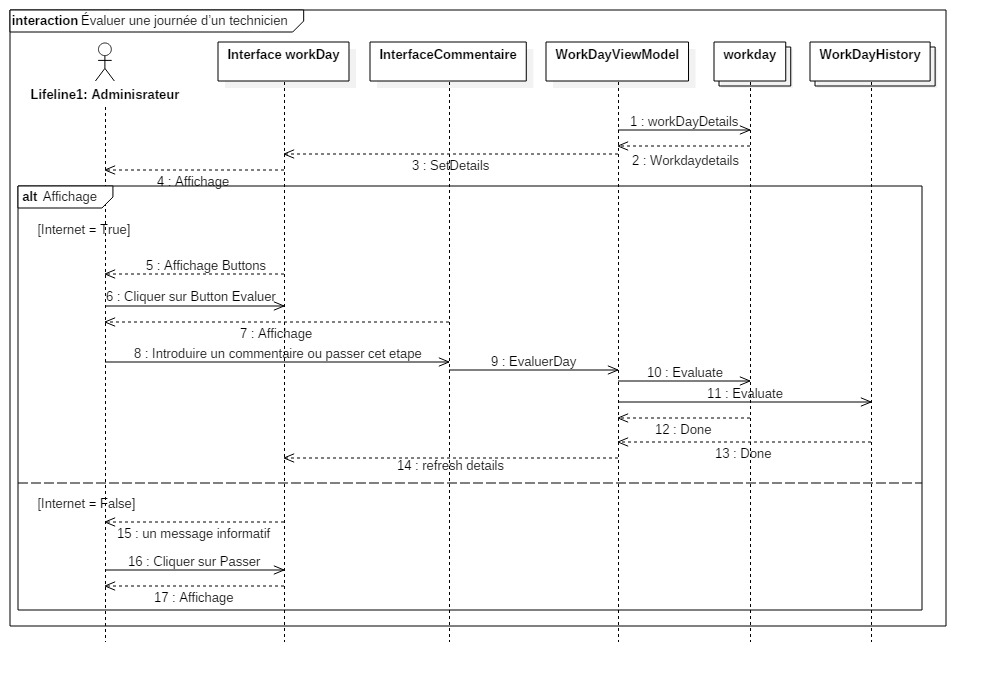
**Figure 29 : Diagramme de séquence détaillé du cas d’utilisation « Consulter les journées d’une date choisit ».**

**Description :**

L’administrateur commence par cliquer sur l’icône du calendrier afin de choisir une date, alors la fonction « getWorkDays » du ViewModel « PendingWorkDayViewModel » qui fait appel au méthode « getWorkDaysByDate » pour récupérer les données tel que modélisé par les flèches 1 jusqu’au 6.

Le ViewModel récupère la réponse s’il s’agit d’un « succès » il retourne les données au « Consulter pending work day » qui permet d’afficher les journées tel que modélisé par les flèches 7 et 8 sinon il retourne un message d’erreur tel que modélisé par le flèche 9. Cette notion est modélisée par une étiquette « alt »

* **Diagramme de séquence de l’opération « Évaluer une journée d’un technicien » :**



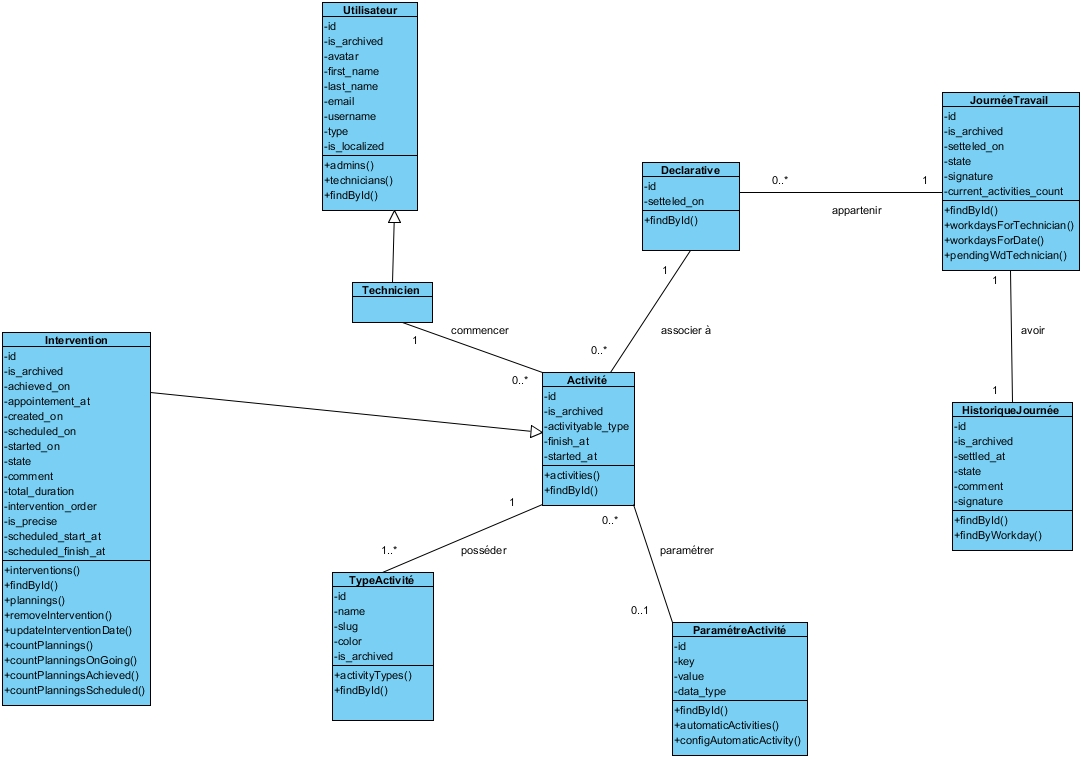
**Figure 30 : Diagramme de séquence détaillé du cas d’utilisation « Évaluer une journée d’un technicien ».**

**Description :**

Lors de la consultation du détails d’une journée, la fonction « workDaysDetails » du ViewModel « WorkDayViewModel » qui fait appel au méthode « workDays » pour récupérer les données tel que modélisé par les flèches 1 jusqu’au 4.

Lors de l’affiche s’il y a une connexion internet, le système affiche les boutons d’évaluations et l’administrateur peut choisir une bouton d’évaluation puis le système affiche l’interface de saisie d’un commentaire, alors la fonction « evaluateDate » du ViewModel « WorkDayViewModel » qui fait appel au méthode « evaluate » pour changer le statut du journée afin de rafraîchir les nouveaux détails tel que modélisé par les flèches 5 jusqu’au 14, sinon un message informatif sera affiché avec une bouton passer qui permet de retourner vers la liste des journées tels que modélisé par les flèches 15,16 et 17, Cette notion est modélisée par une étiquette « alt ».

### **Diagrammes de classes :**



**Figure 31 : Diagrammes de classes du sprint 3.**

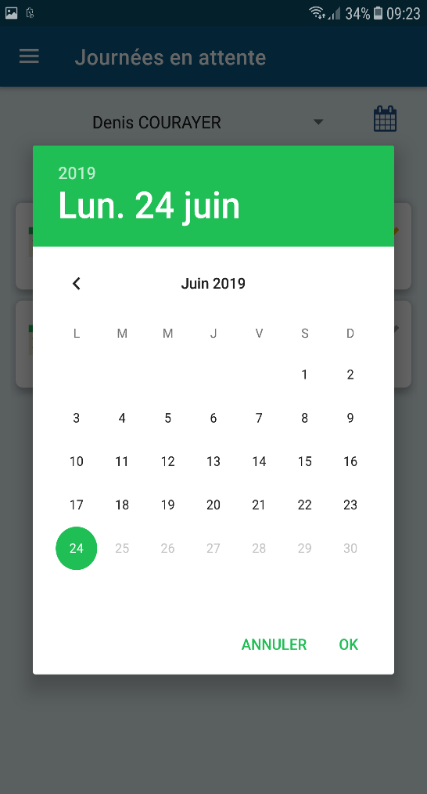
## **Réalisation :**

### **Interface « Consultation des journées en attente » :**

L’administrateur choisit un technicien à partir de la liste déroulante afin de consulter les journées en attente du dernier.

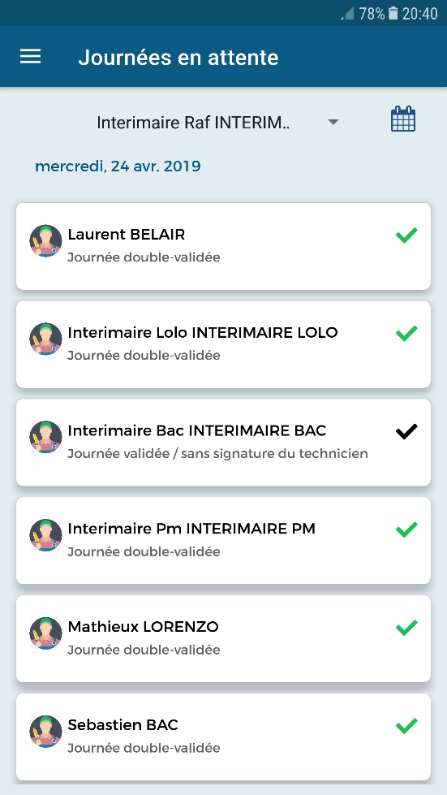
**Figure 32 :interface « Consultation des journées en attente ».**

### **Interface « Consultation des journées pour une date choisit » :**



Pour consulter les journées des techniciens pour une date choisit, il suffit de choisir une date à partir du calendrier.

**Figure 33 : interface « choix d'une date ».**



**Figure 34 : interface « Consultation des journées pour une date choisit ».**

### **Interface « Consultation des détails d’une journée » :**

Après avoir choisir une journée, l’administrateur consulte leur détails en ayant la possibilité de l’évaluer.

**Figure 35 : interface « Consultation des détails d’une journée».**

## **Test :**

Nous avons élaboré dans le tableau 28 un ensemble de cas de scénario de tests relatif au sprint 3.

|  |  |  |  |
| --- | --- | --- | --- |
| Cas de test | Démarche | Comportement attendu | Résultats |
| Test de consultation des journées en attente | Choisir un technicien à partir de la liste déroulante. | La liste des journée en attente pour le technicien choisit s’affiche | Conforme |
| Test de consultation des journées pour une date choisit | Choisir une date à partir du calendrier | La liste des journée en attente pour la date choisit s’affiche | Conforme |
| Test de consultation des détails d’une journée | Choisir une journée à partir de la liste des journées. | Les détails de journée choisit s’affiche | Conforme |
| Test de validation d’une journée | Choisir une journée en attente,  Valider la journée,  Ajouter un commentaire | La journée est validée | Conforme |
| Test de rejet d’une journée | Choisir une journée en attente,  Rejeter la journée,  Ajouter un commentaire | La journée est rejetée | Conforme |
| Test de force d’une journée | Choisir une journée en attente,  Forcer la journée,  Ne pas ajouter un commentaire | La journée est forcée | Conforme |

**Tableau 25 : Tests du sprint 3.**

## **Conclusion :**

Au cours de ce chapitre, nous avons présenté le premier sprint. Pour ce faire, nous avons passé par l’analyse, la conception et la réalisation. Dans le chapitre suivant nous entamons le quatrième sprint.

CHAPITRE VI :

SPRINT 4

## **Introduction :**

Dans le chapitre précédent, nous avons présentait notre troisième sprint.

Dans ce chapitre nous étudions en premier temps la spécification fonctionnelle. En second temps la conception détaillée et enfin la réalisation du quatrième sprint.

## **Backlog du sprint :**

Le tableau ci-dessous décrit les histoires de notre backlog du sprint.

|  |  |  |
| --- | --- | --- |
| ID | Histoires | Estimation |
| 6.1 | Consulter les fiches clients. | 1 |
| 6.2 | Consulter les détails des clients. | 1 |
| 6.3 | Supprimer les clients. | 1 |
| 6.4 | Rechercher un client. | 1 |
| 6.5 | contacter un client | 1 |
| 6.6 | Gérer les contacts d’un client | 1 |
| 6.7 | Consulter l'itinéraire vers l'emplacement d’un client | 1 |
| 7.1 | consulter la liste des fournisseurs. | 1 |
| 7.2 | consulter les détails des fournisseurs | 1 |
| 7.3 | supprimer les fournisseurs | 1 |
| 7.4 | Rechercher un fournisseur | 1 |
| 7.5 | Contacter un fournisseur | 1 |
| 7.6 | Gérer les contacts d’un fournisseur | 1 |
| 7.7 | Consulter l'itinéraire vers l'emplacement d’un fournisseur. | 1 |

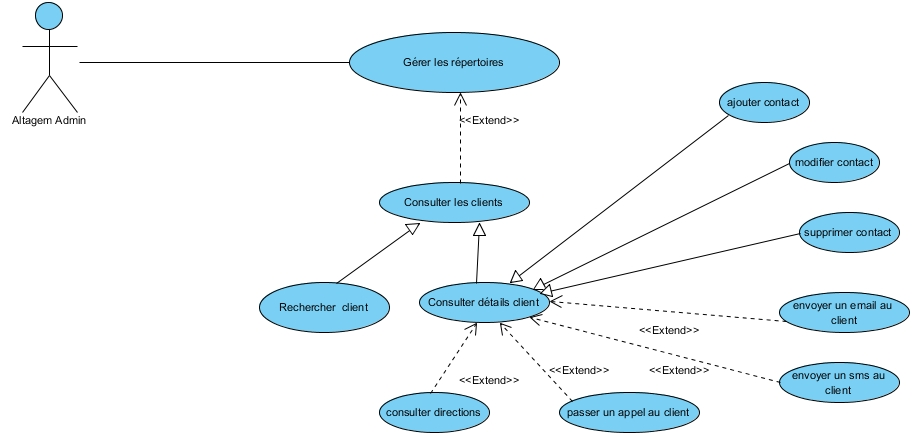
**Tableau 26 : Backlog du sprint 4.**

## **Spécification fonctionnelle :**

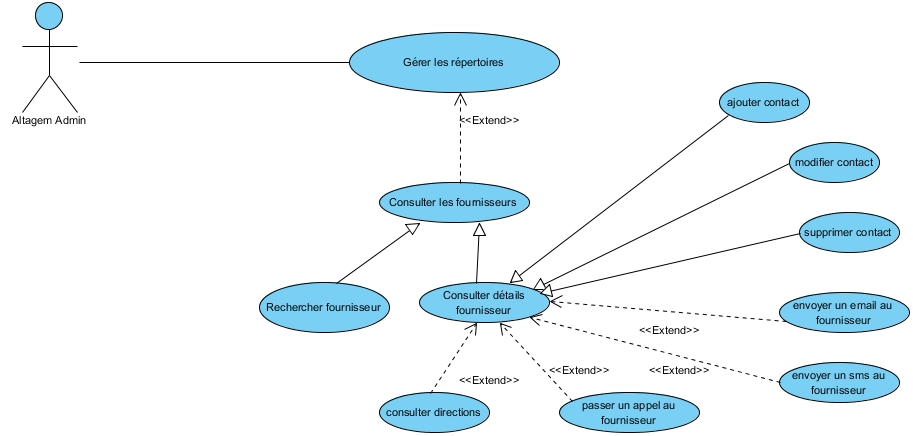
Dans cette partie, nous présentons les diagrammes de cas d’utilisation et la description textuelle de chacun d’entre eux.

### **Diagramme de cas d’utilisation :**

La figure ci-dessous représentent le diagramme de cas d’utilisation global du sprint 4.



**Figure 36 : Diagramme de cas d'utilisation « Gérer les clients».**



**Figure 37 : Diagramme de cas d'utilisation « Gérer les fournisseurs».**

**Description textuelle des cas d’utilisations :**

* **Description textuelle du cas d’utilisation « Consulter les fiches clients/fournisseurs »**

Le tableau ci-dessous décrit la description textuelle du cas d’utilisation « Consulter les fiches clients/fournisseurs ».

|  |  |
| --- | --- |
| SOMMAIRE D’IDENTIFICATION | |
| CU14 | Consulter les fiches clients/fournisseurs |
| Acteur | Altagem Admin |
| Résumé | L’admin sélection la section clients/fournisseurs pour consulter leurs fiches. |
| DESCRIPTION DES ENCHAINEMENTS SOMMAIRE | |
| Précondition | Authentification et accès autorisé. |
| Post-condition | L’administrateur consulte les fiches clients/fournisseurs. |
| Scénario principal | 1. L’administrateur clique sur la rubrique « Répertoires », 2. L’administrateur clique sur la section clients/fournisseurs, 3. Le système affiche la liste clients/fournisseurs |
| Scénario d'exception | Si La liste des clients/fournisseurs est vide. |

**Tableau 27 :Tableau descriptif du cas d'utilisation « Consulter les fiches clients/fournisseurs ».**

* **Description textuelle du cas d’utilisation « Consulter les détails du client/fournisseur »**

Le tableau ci-dessous décrit la description textuelle du cas d’utilisation « Consulter les détails du client/fournisseur ».

|  |  |
| --- | --- |
| SOMMAIRE D’IDENTIFICATION | |
| CU15 | Consulter les détails du client/fournisseur |
| Acteur | Altagem Admin |
| Résumé | L’admin sélection un client/fournisseur afin de consulter ses détails. |
| DESCRIPTION DES ENCHAINEMENTS SOMMAIRE | |
| Précondition | Authentification et accès autorisé. |
| Post-condition | L’administrateur consulte les détails d’un client/fournisseur. |
| Scénario principal | 1. L’administrateur appui sur un client/fournisseur, 2. Le système affiche une interface contenant les détails du dernier tels que sa position sur une carte, et ses contacts. |
| Scénario d'exception | Si La liste des clients/fournisseurs est vide. |

**Tableau 28 : Tableau descriptif du cas d'utilisation « Consulter les détails du client/fournisseur ».**

* **Description textuelle du cas d’utilisation « Supprimer un client/fournisseur » :**

Le tableau 32 décrit la description textuelle du cas d’utilisation « Supprimer un client/fournisseur ».

|  |  |
| --- | --- |
| **SOMMAIRE D’IDENTIFICATION** | |
| CU16 | Supprimer un client/fournisseur |
| Acteur | Altagem Admin |
| Résumé | Permet à l’administrateur de supprimer des clients/fournisseurs. Le système effectue une vérification puis une mise à jour de la base de données. |
| **DESCRIPTION DES ENCHAINEMENTS SOMMAIRE** | |
| Précondition | Authentification et accès autorisé.  Le client/fournisseur cible existe. |
| Post-condition | Le client/fournisseur est supprimé de la base. |
| Scénario principal | 1. L’administrateur glisse la case de Le client/fournisseur souhaité pour la supprimer, 2. Le système affiche un message de confirmation, 3. L’administrateur confirme la suppression. 4. La nouvelle liste des clients/fournisseurs s’affiche. |
| Scénario d'exception | Si l’administrateur annule la suppression |

**Tableau 29 : Tableau descriptif du cas d'utilisation « Supprimer un client/fournisseur».**

* **Description textuelle du cas d’utilisation « Rechercher un client/fournisseur » :**

Le tableau ci-dessous décrit la description textuelle du cas d’utilisation « Rechercher un client/fournisseur ».

|  |  |
| --- | --- |
| **SOMMAIRE D’IDENTIFICATION** | |
| CU17 | Rechercher un client/fournisseur |
| Acteur | Altagem Admin |
| Résumé | L’administrateur recherche un client/fournisseur en tapant son nom, le système affiche le client/fournisseur qui correspond à sa requête. |
| **DESCRIPTION DES ENCHAINEMENTS SOMMAIRE** | |
| Précondition | Authentification et accès autorisé. |
| Post-condition | Rechercher un client/fournisseur |
| Scénario principal | 1. L’administrateur tape sa requête dans le formulaire de recherche, 2. Le système cherche le client/fournisseur qui correspond à sa requête, 3. Le système affiche le résultat. |
| Scénario d'exception | Si le client/fournisseur est introuvable. |

**Tableau 30 : Tableau descriptif du cas d'utilisation « Rechercher un client/fournisseur».**

* **Description textuelle du cas d’utilisation « Contacter un client/fournisseur » :**

Le tableau ci-dessous décrit la description textuelle du cas d’utilisation « Contacter un client/fournisseur ».

|  |  |
| --- | --- |
| **SOMMAIRE D’IDENTIFICATION** | |
| CU18 | Contacter un client/fournisseur |
| Acteur | Altagem Admin |
| Résumé | L’administrateur choisit le type de contact souhaité tels que passer un appel, envoyer un sms ou bien un email. |
| **DESCRIPTION DES ENCHAINEMENTS SOMMAIRE** | |
| Précondition | Authentification et accès autorisé. |
| Post-condition | L’administrateur contacte un client/fournisseur |
| Scénario principal | Le système affiche la liste des contacts,  L’administrateur choisit un contact,  L’administrateur choisit le type souhaité tels que passer un appel, envoyer un sms ou bien un email. |
| Scénario d'exception | Si la liste des contacts est vide. |

**Tableau 31 : Tableau descriptif du cas d'utilisation « Contacter un client/fournisseur».**

* **Description textuelle du cas d’utilisation « Ajouter un contact » :**

Le tableau ci-dessous décrit la description textuelle du cas d’utilisation « Ajouter un contact ».

|  |  |
| --- | --- |
| **SOMMAIRE D’IDENTIFICATION** | |
| CU19 | Ajouter un contact |
| Acteur | Altagem Admin |
| Résumé | Permet à l’administrateur d’ajouter un nouveau contact,  Le système effectue une vérification puis une mise à jour de la base de données. |
| **DESCRIPTION DES ENCHAINEMENTS SOMMAIRE** | |
| Précondition | Authentification et accès autorisé.  Le contact cible n’est pas existant. |
| Post-condition | Le nouveau contact est bien ajoutée dans la base. |
| Scénario principal | 1. L’administrateur clique sur le bouton « ajouter ». 2. Le système affiche le formulaire d’ajout du contact. 3. L’administrateur remplit le formulaire avec les nouvelles données. 4. L’administrateur clique sur le bouton « valider ». 5. Le système vérifie le champ saisi. 6. Le système affiche la nouvelle liste des contacts. |
| Scénario d'exception | Si le champ est vide, un message d’erreur s’affichera.  Si le contact existe déjà dans la base, le système affiche un message d’erreur « contact déjà existe ». |

**Tableau 32 : Tableau descriptif du cas d'utilisation « Ajouter un contact».**

* **Description textuelle du cas d’utilisation « Modifier un contact » :**

Le tableau 36 décrit la description textuelle du cas d’utilisation « Modifier un contact ».

|  |  |
| --- | --- |
| **SOMMAIRE D’IDENTIFICATION** | |
| CU20 | Modifier un contact |
| Acteur | Altagem Admin |
| Résumé | Permet à l’administrateur de modifier un contact.  Le système effectue une vérification puis une mise à jour de la base de données. |
| **DESCRIPTION DES ENCHAINEMENTS SOMMAIRE** | |
| Précondition | Authentification et accès autorisé.  Le contact existe. |
| Post-condition | Les modifications abouties au contact sont enregistrées dans la base. |
| Scénario principal | L’administrateur clique sur le bouton « Modifier ».  Le système affiche le formulaire de modification des données.  L’administrateur modifie le formulaire selon les nouvelles données.  L’administrateur valide les modifications en cliquant sur le bouton « valider ».  Le système vérifie le champ saisi  Le système affiche la nouvelle date. |
| Scénario d'exception | Si le champ est vide, un message d’erreur s’affichera.  Si le contact existe déjà dans la base, le système affiche un message d’erreur « contact déjà existe ».  Si les données saisies ne sont pas valide un message d’erreur s’affichera. |

**Tableau 33 : Tableau descriptif du cas d'utilisation « Modifier un contact».**

* **Description textuelle du cas d’utilisation « Consulter l'itinéraire vers l'emplacement d’un client/fournisseur » :**

Le tableau ci-dessous décrit la description textuelle du cas d’utilisation « Consulter l'itinéraire vers l'emplacement d’un client/fournisseur ».

|  |  |
| --- | --- |
| **SOMMAIRE D’IDENTIFICATION** | |
| CU21 | Consulter l'itinéraire vers l'emplacement d’un client/fournisseur |
| Acteur | Altagem Admin |
| Résumé | Permet à l’administrateur Consulter l'itinéraire vers l'emplacement d’un client/fournisseur,  Le système affiche les directions vers l’emplacement souhaité sur Google Map. |
| **DESCRIPTION DES ENCHAINEMENTS SOMMAIRE** | |
| Précondition | Authentification et accès autorisé.  La position existe. |
| Post-condition | L’administrateur consulte l'itinéraire vers l'emplacement d’un client/fournisseur. |
| Scénario principal | L’administrateur clique sur le bouton « Naviguer vers ».  Le système ouvre Google Map et affiche l'itinéraire vers l'emplacement d’un client/fournisseur. |
| Scénario d'exception | Si la position n’est pas valide. |

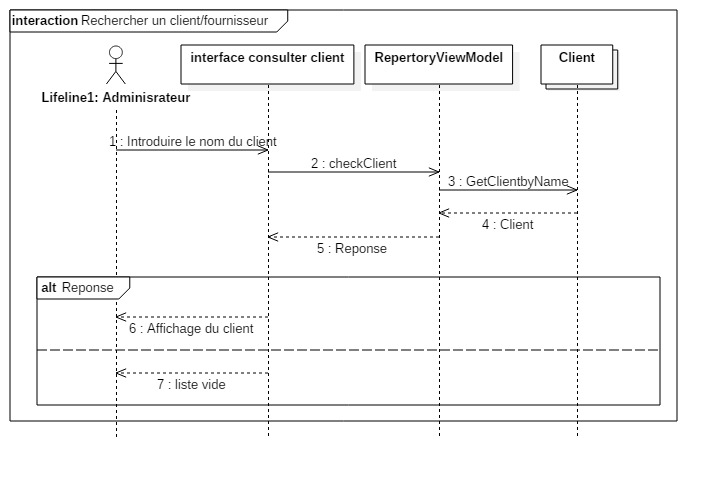
**Tableau 34 :Tableau descriptif du cas d'utilisation « Consulter l'itinéraire vers l'emplacement d’un client/fournisseur».**

## **La conception :**

Dans cette section, nous présentons les différents diagrammes de séquence détaillés ainsi que le diagramme de classe pour ce sprint.

### **Diagrammes de séquences détaillés :**

* **Diagramme de séquence de l’opération « Rechercher un client/fournisseur » :**



**Figure 38 : Diagramme de séquence détaillé du cas d’utilisation «rechercher un client».**

**Description :**

L’administrateur introduit le nom du client alors la fonction « checkClient » du ViewModel « repertoryViewModel » est appelé tel que modélisé par les flèches 1 et 2.

La fonction « checkClient » appelle les méthode « getClientByName » pour récupérer les données du technicien tel que modélisé par le flèche 3, 4et 5.

Le ViewModel récupère la réponse, s’il s’agit d’un « succès » il retourne les données au « client interface » qui permet d’afficher les données tel que modélisé par le flèche 6 sinon il retourne une liste vide tel que modélisé par le flèche 7. Cette notion est modélisée par une étiquette « alt ».

* **Diagramme de séquence de l’opération « Ajouter un contact » :**



**Figure 39 : Diagramme de séquence détaillé du cas d’utilisation «ajouter un contact».**

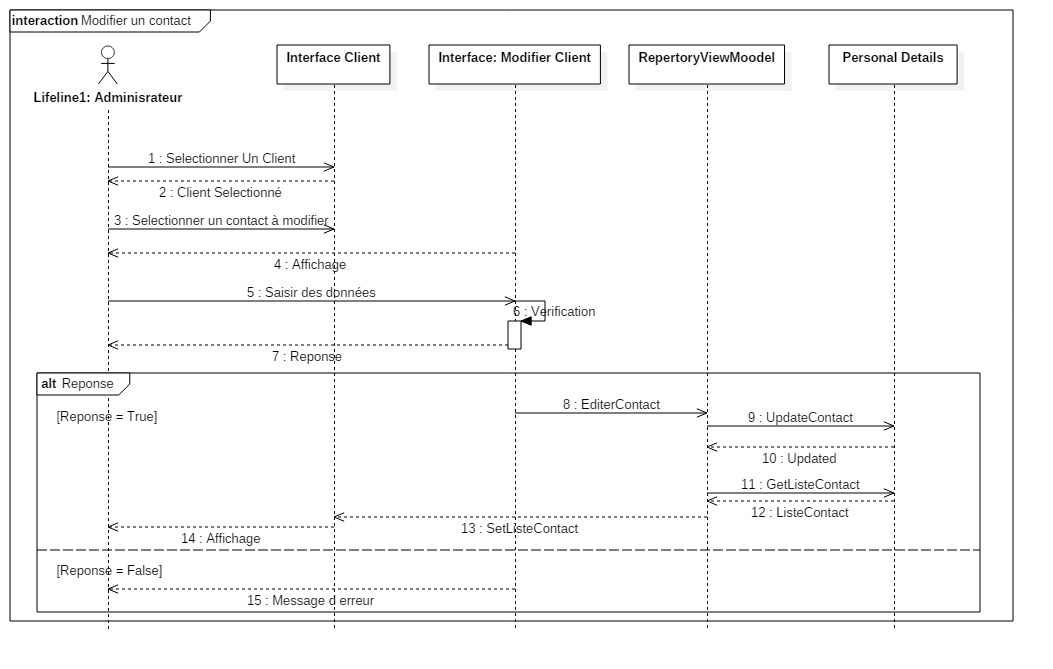
**Description :**

Lors de l’ajout d’un nouveau contact, l’administrateur commence par saisir les données puis le system vérifie la validité du champ tels que modélisé par les flèches 1 et 2.

Si les données sont valides, le viewModel « RepertoryViewModel » fait appel à la fonction « checkContact » pour vérifier l’existence du contact dans la table personalDetails tels que modélisé par les flèches 3 jusqu’au 7 sinon un message d’erreur sera affiché tels que modélisé par le flèche 16. Cette notion est modélisée par une étiquette « alt ».

Si le contact n’est pas existant le viewModel fait appel à la méthode « addContact » pour ajouter le nouveau contact afin de récupérer la nouvelle liste de contact tels que modélisé par les flèches 8 jusqu’au 14 sinon un message informatif sera affiché tels que modélisé par le flèche 15. Cette notion est modélisée par une étiquette « alt ».

* **Diagramme de séquence de l’opération « Modifier un contact » :**



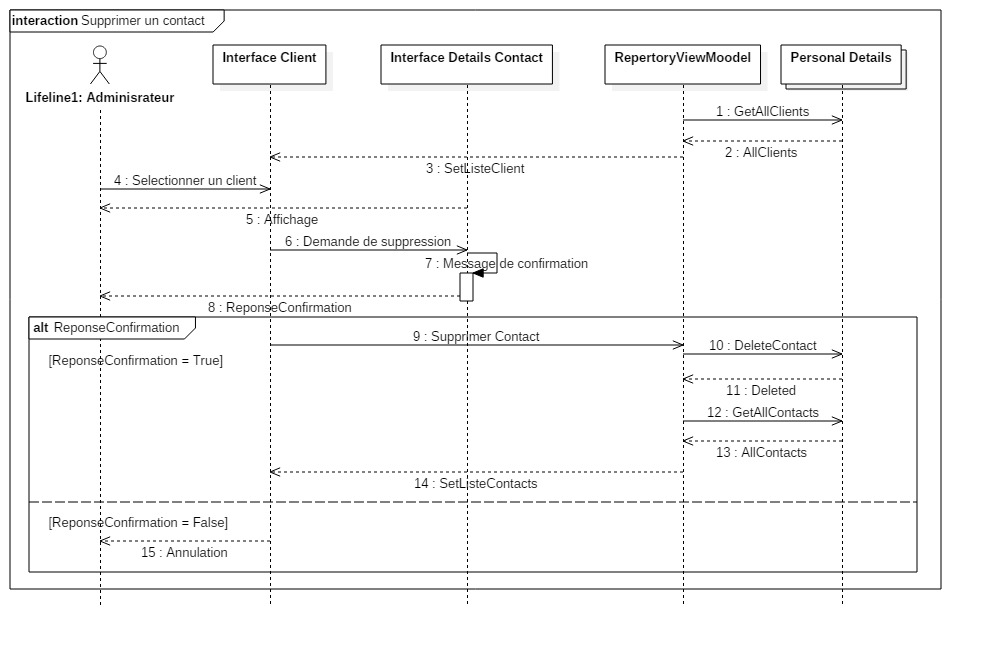
**Figure 40 : Diagramme de séquence détaillé du cas d’utilisation «modifier un contact».**

**Description :**

Lors de la modification d’un contact, l’administrateur commence par sélectionner un client puis il clique sur le bouton « modifier », le système affiche l’interface de modification et l’administrateur commence par saisir les données puis le system vérifie la validité du champ tels que modélisé par les flèches 1 jusqu’au 7.

Si les données sont valides, le viewModel « RepertoryViewModel » fait appel à la fonction « updateContact » pour modifier le contact afin de récupérer la nouvelle liste de contact tels que modélisé par les flèches 8 jusqu’au 14 sinon un message d’erreur sera affiché tels que modélisé par le flèche 15. Cette notion est modélisée par une étiquette « alt ».

* **Diagramme de séquence de l’opération « Supprimer un contact » :**



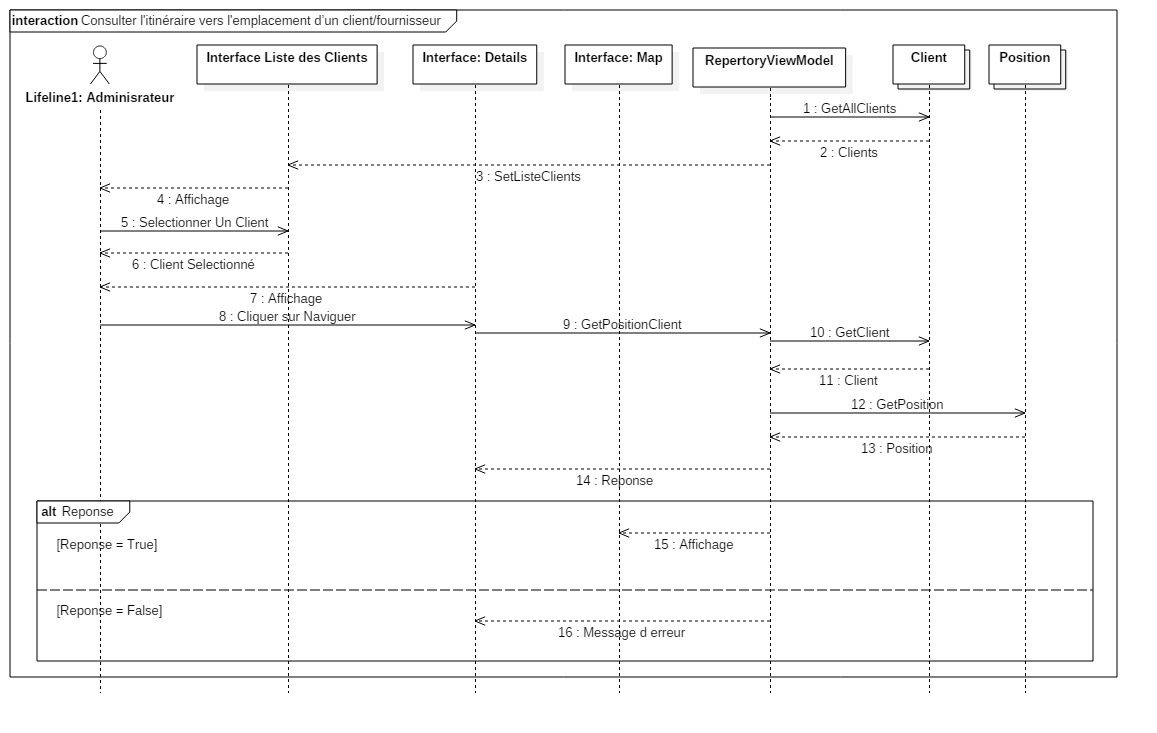
**Figure 41 : Diagramme de séquence détaillé du cas d’utilisation «supprimer un contact».**

**Description :**

Après avoir consulter les contacts et lors de la suppression d’un contact, l’administrateur glisse la case de l’intervention souhaité pour la supprimer, le système demande la confirmation de la suppression tel que modélisé par les flèches 1 jusqu’au 8.

Le ViewModel récupère la réponse, s’il s’agit d’une « confirmation » la fonction « supprimer contact » est appelé qui fait appel au méthode « deleteContact » pour supprimer le contact de la table « détails Personnel » et retourne la nouvelle liste des contacts au « interface détails client » qui permet d’afficher la nouvelle liste des contacts tel que modélisé par les flèches 9 jusqu’au 14 sinon il l’annule tel que modélisé par le flèche 15. Cette notion est modélisée par une étiquette « alt ».

* **Diagramme de séquence de l’opération « Consulter l'itinéraire vers l'emplacement d’un client/fournisseur » :**



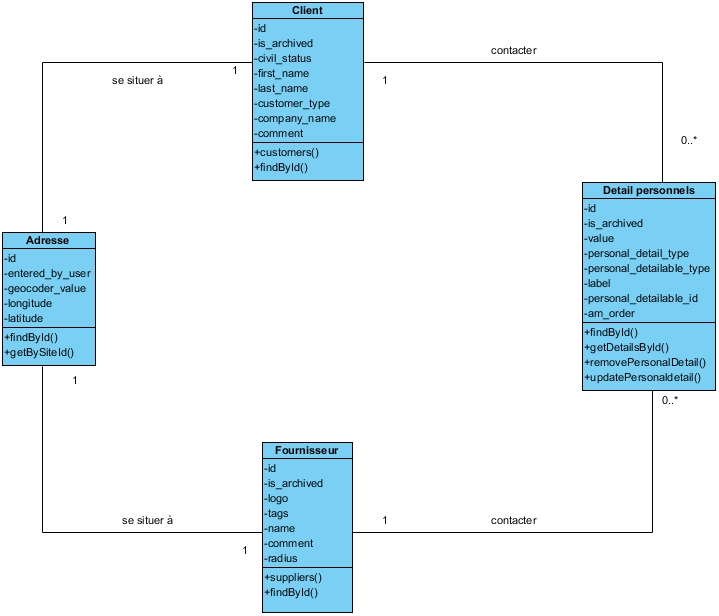
***Figure 42 : Diagramme de séquence détaillé du cas d’utilisation «*Consulter l'itinéraire vers l'emplacement d’un client*».***

**Description :**

Pour Consulter l'itinéraire vers l'emplacement d’un client et après avoir consulter ses détails, l’administrateur clique sur le bouton « naviguez vers » pour cela le viewModel « RepertoryViewModel » fait appel à la fonction « getPositionClient » pour récupérer les coordonnées du client tels que modélisé par les flèches 1 jusqu’au 14.

S’il s’agit d’un « succès », le système affiche l’itinéraire vers l’emplacement du client sur Google Map tels que modélisé par le flèche 15, sinon un message d’erreur sera affiché tels que modélisé par le flèche 16. Cette notion est modélisée par une étiquette « alt ».

### **Diagramme de classes :**



**Figure 43 :Diagramme de classes du sprint 4.**

## **Réalisation :**

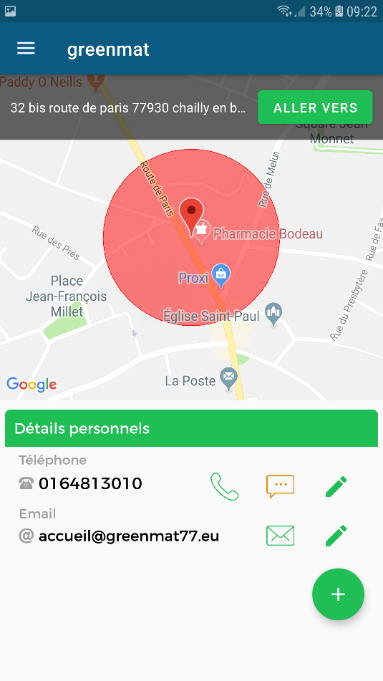
Cette partie est consacrée à l’exposition du travail achevé à travers des captures d’écrans de différentes interfaces développées pendant ce sprint.

### **Interface « Consultation des clients/fournisseurs » :**

Après avoir choisi la rubrique « Répertoire », l’administrateur choisit la section clients ou fournisseurs pour consulter la liste des clients ou bien des fournisseurs.

**Figure 44 : Interface « Consultation des fournisseurs ».**

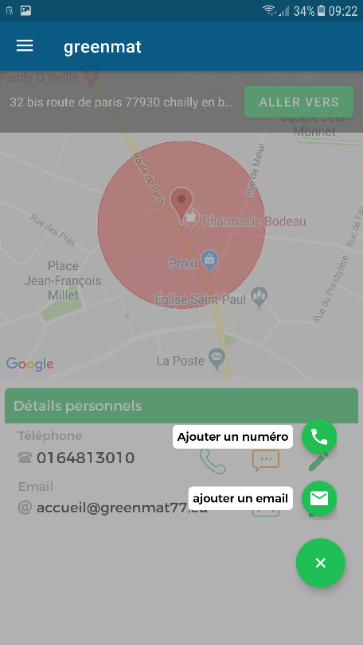
### **Interface « Consultation des détails de client/fournisseur » :**



Après avoir choisi un client/fournisseur, une interface contenant leurs détails s’affiche pour les consulter.

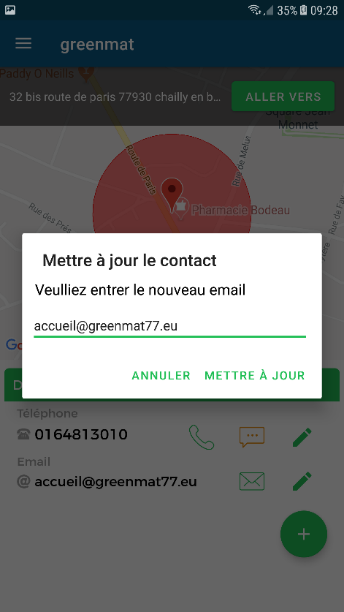
**Figure 45 : Interface « Consultation des détails de client/fournisseur ».**

### **Interface « Gérer les contacts d’un client/fournisseur » :**



**Figure 46 :Interface « Ajouter du contact ».**

Lors de la clique sur le bouton flottant, l’administrateur a la possibilité de choisir entre ajouter un nouveau numéro du téléphone ou bien un email.



**Figure 47 : Interface « Modifier un contact ».**

Lors de la clique sur le bouton de modification une interface s’affiche pour permettre à l’administrateur de modifier le contact choisi.



Quand l’administrateur glisse la case du contact souhaité pour la supprimer, un message de confirmation s’affiche comme indique la figure ci-dessus.

**Figure 48 Interface « Supprimer un contact ».**

## **Test :**

Nous avons élaboré dans le tableau ci-dessous un ensemble de cas de scénario de tests relatif au sprint 4.

|  |  |  |  |
| --- | --- | --- | --- |
| Cas de test | Démarche | Comportement attendu | Résultats |
| Test de consultation du détails d’un client | Choisir un client à partir de la liste des clients. | Les détails du client choisit sont consultées. | Conforme |
| Test de recherche d’un client. | Rechercher un client en tapant son nom | Affichage du client qui correspond à la requête. | Conforme |
| Test de suppression d’un contact | Supprimer un contact. | Le contact est bien supprimé dans la base. | Conforme |
| Test d’ajout d’un nouveau numéro de téléphone pour un client | Créer un nouveau numéro de téléphone | Le nouveau numéro est bien ajouté dans la base. | Conforme |
| Test de modification d’un email pour un fournisseur | Modifier un email pour un fournisseur | La modification aboutie à l’email est enregistrée dans la base | Conforme |
| Test de consultation de l'itinéraire vers l'emplacement d’un fournisseur | Consulter l'itinéraire vers l'emplacement d’un fournisseur | l'itinéraire vers l'emplacement du fournisseur est consulter sur Google Map | Conforme |

**Tableau 35 : Tests du sprint 4.**

## **Conclusion :**

Au cours de ce chapitre, nous avons présenté le premier sprint. Pour ce faire, nous avons passé par l’analyse, la conception et la réalisation. Dans le chapitre suivant nous entamons le dernier sprint.

CHAPITRE VII :

SPRINT 5

## **Introduction :**

Nous allons entamer dans ce chapitre le dernier Sprint. Pour cela, Nous commençons par

Présenter le Backlog du sprint, ensuite nous passons à l’analyse et la conception et nous clôturons par la partie la réalisation et test.

## **Backlog du sprint :**

Le tableau ci-dessous décrit les histoires de notre backlog du sprint.

|  |  |  |
| --- | --- | --- |
| **ID** | **Histoires** | **Estimation** |
| 8.1 | Consulter la liste des contrats. | 1 |
| 8.2 | Consulter les détails du contrat. | 1 |
| 8.3 | Rechercher contrat | 1 |
| 8.4 | Consulter l'itinéraire vers l'emplacement d’un chantier. | 1 |
| 9 | Synchroniser les données. | 10 |

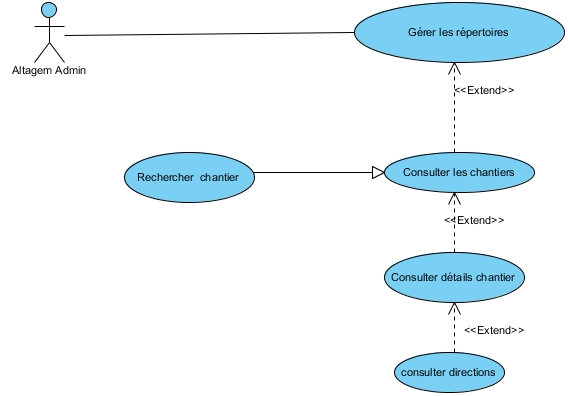
**Tableau 36 : Backlog du sprint 5.**

## **Spécification fonctionnelle :**

Dans cette partie, nous présentons les diagrammes de cas d’utilisation et la description textuelle de chacun d’entre eux.

### **Diagramme de cas d’utilisation :**

La figure 49 représentent le diagramme de cas d’utilisation global du sprint 5.



**Figure 49 : Diagramme de cas d'utilisation « Gérer les contrats».**

**Description textuelle des cas d’utilisations :**

* **Description textuelle du cas d’utilisation « Consulter les contrats » :**

Le tableau ci-dessous décrit la description textuelle du cas d’utilisation « Consulter les contrats »

|  |  |
| --- | --- |
| SOMMAIRE D’IDENTIFICATION | |
| CU22 | Consulter les contrats |
| Acteur | Altagem Admin |
| Résumé | L’admin sélection la section contrats pour consulter les différents contrats selon leur type. |
| DESCRIPTION DES ENCHAINEMENTS SOMMAIRE | |
| Précondition | Authentification et accès autorisé. |
| Post-condition | L’administrateur consulte les contrats. |
| Scénario principal | 1. L’administrateur clique sur la rubrique « Répertoires », 2. L’administrateur clique sur la section contrat, 3. Le système affiche la liste contrat. |
| Scénario d'exception | Si La liste des contrats est vide. |

**Tableau 37 : Tableau descriptif du cas d'utilisation « Consulter les contrats ».**

* **Description textuelle du cas d’utilisation « Consulter les détails du contrat » :**

Le tableau ci-dessous décrit la description textuelle du cas d’utilisation « Consulter les détails du contrat ».

|  |  |
| --- | --- |
| SOMMAIRE D’IDENTIFICATION | |
| CU23 | Consulter les détails du contrat |
| Acteur | Altagem Admin |
| Résumé | L’admin sélection un contrat afin de consulter ses détails. |
| DESCRIPTION DES ENCHAINEMENTS SOMMAIRE | |
| Précondition | Authentification et accès autorisé. |
| Post-condition | L’administrateur consulte les détails d’un contrat. |
| Scénario principal | 1. L’administrateur appui sur un contrat, 2. Le système affiche une interface contenant les détails du dernier tels que sa position sur une carte, et son cahier des charges. |
| Scénario d'exception | Si La liste des contrats est vide. |

**Tableau 38 : Tableau descriptif du cas d'utilisation « Consulter les détails du contrat ».**

* **Description textuelle du cas d’utilisation « Rechercher un contrat » :**

Le tableau ci-dessous décrit la description textuelle du cas d’utilisation « Rechercher un contrat »

|  |  |
| --- | --- |
| **SOMMAIRE D’IDENTIFICATION** | |
| CU24 | Rechercher un contrat |
| Acteur | Altagem Admin |
| Résumé | L’administrateur recherche un contrat en tapant son nom, le système affiche le contrat qui correspond à sa requête. |
| **DESCRIPTION DES ENCHAINEMENTS SOMMAIRE** | |
| Précondition | Authentification et accès autorisé. |
| Post-condition | Rechercher un contrat |
| Scénario principal | 1. L’administrateur tape sa requête dans le formulaire de recherche, 2. Le système cherche le contrat qui correspond à sa requête, 3. Le système affiche le résultat. |
| Scénario d'exception | Si le contrat est introuvable. |

**Tableau 39 : Tableau descriptif du cas d'utilisation « Rechercher un contrat ».**

* **Description textuelle du cas d’utilisation « Consulter l'itinéraire vers l'emplacement d’un chantier » :**

Le tableau ci-dessous décrit la description textuelle du cas d’utilisation « Consulter l'itinéraire vers l'emplacement d’un chantier ».

|  |  |
| --- | --- |
| **SOMMAIRE D’IDENTIFICATION** | |
| CU25 | Consulter l'itinéraire vers l'emplacement d’un chantier. |
| Acteur | Altagem Admin |
| Résumé | Permet à l’administrateur Consulter l'itinéraire vers l'emplacement d’un chantier,  Le système affiche les directions vers l’emplacement souhaité sur Google Map. |
| **DESCRIPTION DES ENCHAINEMENTS SOMMAIRE** | |
| Précondition | Authentification et accès autorisé.  La position existe. |
| Post-condition | L’administrateur consulte l'itinéraire vers l'emplacement d’un chantier. |
| Scénario principal | 1. L’administrateur clique sur le bouton « Naviguer vers ». 2. Le système ouvre Google Map et affiche l'itinéraire vers l'emplacement d’un chantier. |
| Scénario d'exception | Si la position n’est pas valide. |

**Tableau 40 : Tableau descriptif du cas d'utilisation « Consulter l'itinéraire vers l'emplacement d’un chantier ».**

* **Description textuelle du cas d’utilisation « Synchroniser les données » :**

Le tableau ci-dessous décrit la description textuelle du cas d’utilisation « Synchroniser les données ».

|  |  |
| --- | --- |
| **SOMMAIRE D’IDENTIFICATION** | |
| CU26 | Synchroniser les données |
| Acteur | Altagem Admin |
| Résumé | Permet à l’administrateur de Synchroniser les données,  Le système mettre à jour les données sauvegarder en local avec celle du serveur. |
| **DESCRIPTION DES ENCHAINEMENTS SOMMAIRE** | |
| Précondition | Authentification et accès autorisé.  Présence d’une connexion internet. |
| Post-condition | Les données sont à jour. |
| Scénario principal | 1. L’administrateur clique sur la rubrique « Synchroniser maintenant ». 2. Le système demande les données du serveur, 3. Le système récupère les données et les sauvegarde dans la base de données local, 4. Le système envoie les données modifier en local vers le serveur. |
| Scénario d'exception | Si la connexion internet n’est pas disponible. |

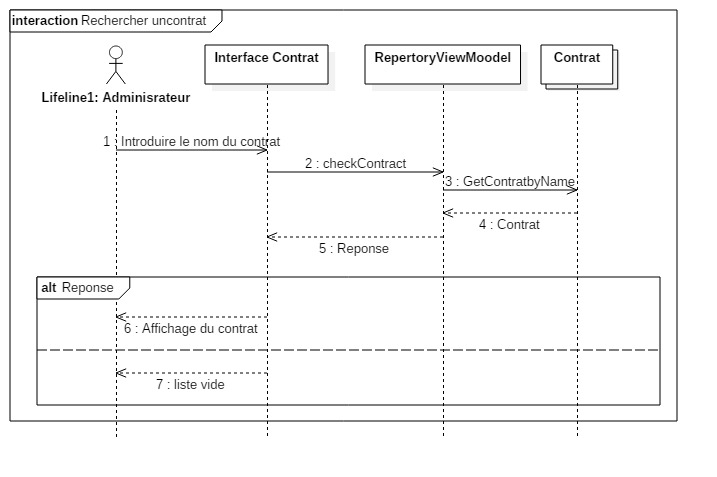
**Tableau 41 : Tableau descriptif du cas d'utilisation «Synchroniser les données ».**

## **La conception :**

Dans cette section, nous présentons les différents diagrammes de séquence détaillés ainsi que le diagramme de classe pour ce sprint.

### **Diagrammes de séquences détaillés :**

* **Diagramme de séquence de l’opération « Rechercher un contrat » :**



**Figure 50 : Diagramme de séquence détaillé du cas d’utilisation «rechercher un contrat».**

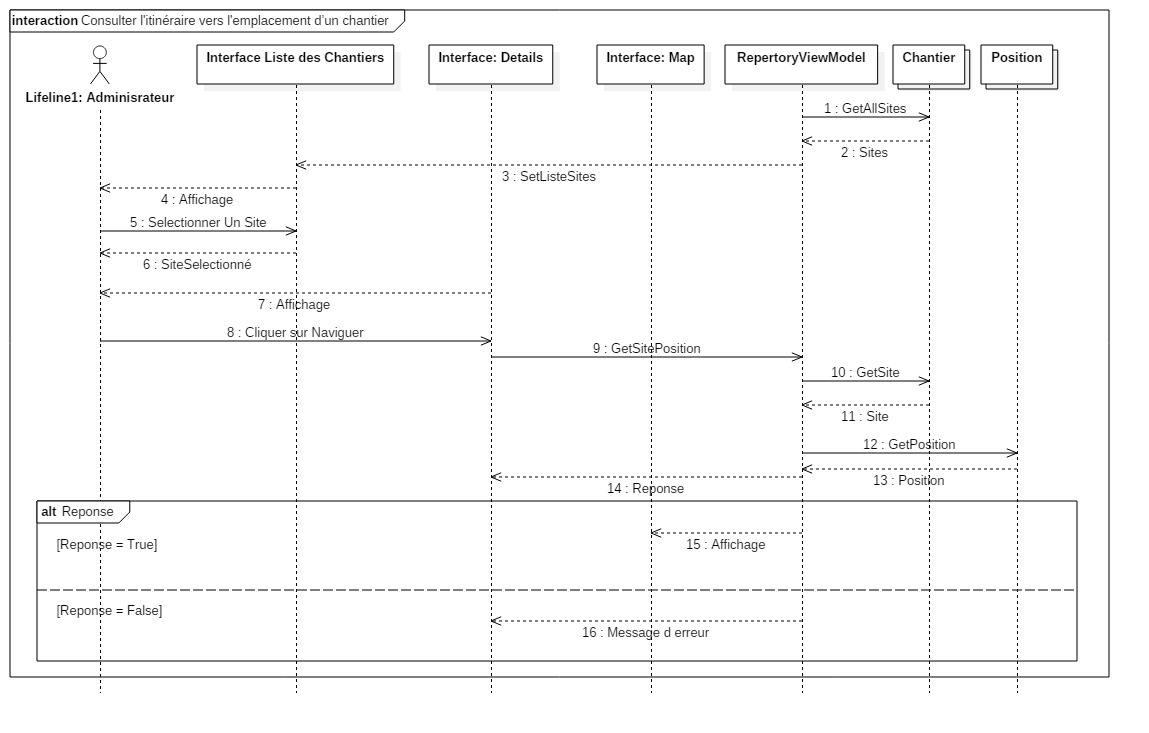
**Description :**

L’administrateur introduit le nom du contrat alors la fonction « checkContract » du ViewModel « repertoryViewModel » est appelé tel que modélisé par les flèches 1 et 2.

La fonction « checkContract » appelle les méthode « getContractByName » pour récupérer les données du technicien tel que modélisé par le flèche 3, 4et 5.

Le ViewModel récupère la réponse, s’il s’agit d’un « succès » il retourne les données au « contrat interface » qui permet d’afficher les données tel que modélisé par le flèche 6 sinon il retourne une liste vide tel que modélisé par le flèche 7. Cette notion est modélisée par une étiquette « alt ».

* **Diagramme de séquence de l’opération « Consulter l'itinéraire vers l'emplacement d’un chantier » :**



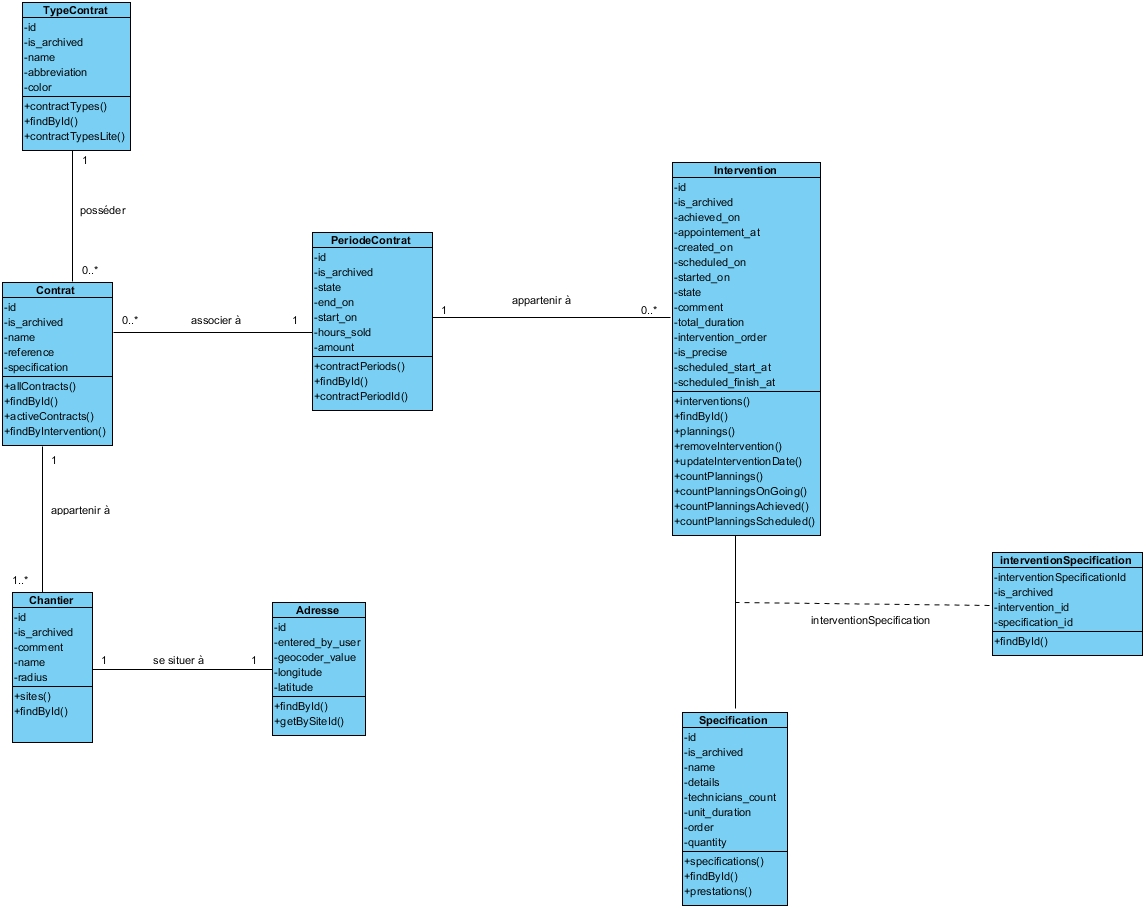
**Figure 51 : Diagramme de séquence détaillé du cas d’utilisation «Consulter l'itinéraire vers l'emplacement d’un chantier».**

**Description :**

Pour consulter l'itinéraire vers l'emplacement d’un chantier et après avoir consulter ses détails, l’administrateur clique sur le bouton « naviguez vers » pour cela le viewModel « RepertoryViewModel » fait appel à la fonction « getSitePosition » pour récupérer les coordonnées du chantier tels que modélisé par les flèches 1 jusqu’au 14.

S’il s’agit d’un « succès », le système affiche l’itinéraire vers l’emplacement du chantier sur Google Map tels que modélisé par le flèche 15, sinon un message d’erreur sera affiché tels que modélisé par le flèche 16. Cette notion est modélisée par une étiquette « alt ».

### **Diagramme de classes**:

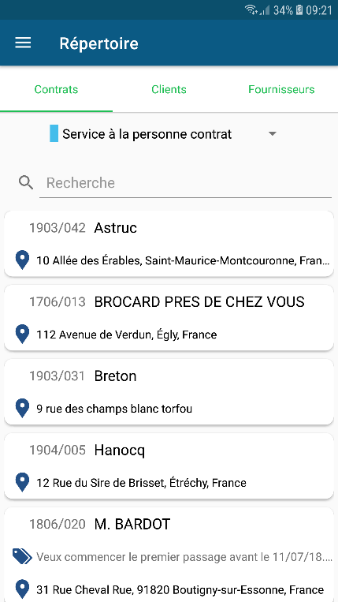


**Figure 52 : Diagramme de classes du sprint 5.**

## **Réalisation :**

Cette partie est consacrée à l’exposition du travail achevé à travers des captures d’écrans de différentes interfaces développées pendant ce sprint.

### **Interface « Consultation des contrats » :**



Après avoir choisi la rubrique « Répertoire », l’administrateur choisit la section contrats pour consulter la liste des contrats.

**Figure 53 : Interface « Consultation des contrats »**

### **Interface « Consultation des détails du contrat » :**

Après avoir choisi un contrat, une interface contenant leurs détails s’affiche pour les consulter.

**Figure 54 : Interface « Consultation des détails du contrat »**

## **Test :**

Nous avons élaboré dans le tableau ci-dessous un ensemble de cas de scénario de tests relatif au sprint 5.

|  |  |  |  |
| --- | --- | --- | --- |
| Cas de test | Démarche | Comportement attendu | Résultats |
| Test de consultation des contrats | Choisir un type de contrat à partir de la liste déroulante | Les contrats appartenant au type choisit s’affiche | Conforme |
| Test de consultation du détails d’un contrat | Choisir un contrat à partir de la liste des contrats. | Les détails du contrat choisit sont consultées. | Conforme |
| Test de recherche d’un contrat. | Rechercher un contrat en tapant son nom | Affichage du contrat qui correspond à la requête. | Conforme |
| Test de consultation de l'itinéraire vers l'emplacement d’un chantier | Consulter l'itinéraire vers l'emplacement d’un chantier | l'itinéraire vers l'emplacement du chantier est consulter sur Google Map | Conforme |

**Tableau 42 : Tests sprint 5.**

## **Conclusion :**

A ce stade, nous avons réussi donc à développer le dernier release de notre application pour arriver à un produit complet et fonctionnel. Notre application est donc prête à être exploiter en offriront aux administrateurs la possibilité de gérer convenablement toutes les fonctionnalités de l’application.

# **Conclusion générale et Perspectives**

En guise de conclusion, Cette période de stage au sein de la société « **Com&Dev** » a été plus qu'enrichissantes pour nous. En tant que futures informaticiennes, il nous paraissait indispensable d'effectuer un stage nous permettant de découvrir de nouvelles technologies, de nouvelles méthodes de travail et un nouvel environnement de développement.

Dans ce cadre, le présent projet de fin d'études avait pour but la conception et la réalisation d’une application mobile pour la gestion et le contrôle des équipes terrains appelée « **Altagem Admin** », afin de faciliter le travail des administrateurs du solution **ALTAGEM**.

En effet, notre formation s’est consolidée au cours de ce stage par un véritable travail de réflexion, de recherche, de synthèse, de conception et de programmation à l’aide d’un cas réel. En fait, Pour accomplir ce travail, nous avons adopté la méthode agile « SCRUM » pour la gestion de projet, UML est désormais notre langage de modélisation. L'application mobile est développée sous Android.

Tout au long de l’élaboration du projet, nous avons rencontré plusieurs difficultés tant au niveau conceptuel qu’au niveau de la réalisation. Tout de même, nous avons réussi à les surmonté pour présenter finalement une application opérationnelle.

En définitive, durant la période de ce projet de fin d’études, l’aspect humain était toujours présent. En effet, le dialogue, les conseils des encadrants et le professionnalisme du groupe **Com&Dev** nous ont aidées et encouragées à réaliser et réussir ce projet.

Finalement, notre travail ne s'arrête pas à ce niveau. En effet, plusieurs perspectives s'offrent à ce projet.

Parmi les fonctionnalités que nous pouvons envisager pour « Altagem Admin » :

* Consulter les rapports des activités,
* La gestion électronique des documents **GED**.
* Paramétré l’application pour être adapté aux différents domaines d’activités.

# **Référence Bibliographies**

[2] : *Roques, Pascal. Les Cahiers du Programmeur UML2 Modéliser une application Web. 4éme édition. Paris : ÉDITIONS EYROLLES, 2008, 264 p.*

[1] : <http://www.cloudsmart.lu/en_US/page/methode-scrum> (consulté le 20/04/2019)

[3] :<https://blog.zenika.com/2011/05/10/edition-de-maquettes-ihm-avec-balsamiq-mockups/> (consulté le 22/04/2019)

[4] :<https://online.visual-paradigm.com/fr/solutions/free-online-diagram-software/> (consulté le 25/04/2019)

[5] : <https://www.lemagit.fr/definition/Android> (consulté le 02/05/2019)

[6] :<https://docops.ca.com/devtest-solutions/8-0-1/fr/installation-et-migration/installation/configuration-de-l-environnement-de-test-d-application-mobile/utilisation-de-genymotion> (consulté le 05/05/2019)

[7] : <https://www.lemagit.fr/definition/Kotlin> (consulté le 08/05/2019)

[8] :<https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203607-sqlite-definition/> (consulté le 12/05/2019)

[9] : <http://glossaire.infowebmaster.fr/xml/> (consulté le 16/05/2019)

[10] :<https://www.lemagit.fr/definition/JSON-JavaScript-Object-Notation> (consulté le 20/05/2019)

[11] :<https://dzone.com/articles/an-introduction-to-dagger-2-android-di-part-1-3> (consulté le 22/05/2019)

[12] :<https://blog.mindorks.com/rxjava-anatomy-what-is-rxjava-how-rxjava-is-designed-and-how-rxjava-works-d357b3aca586> (consulté le 04/06/2019)

[13] : <http://tutos-android-france.com/retrofit-webservices-rest/> (consulté le 06/06/2019)

[14] :<https://blog.ippon.fr/2018/05/31/introduction-aux-android-architecture-components/> (consulté le 08/06/2019)

[15] : <https://www.futura-sciences.com/tech/definitions/smartphone-android-15017/> (consulté le 12/06/2019)

[16] : <https://www.dotnetdojo.com/mvvm/> (consulté le 16/06/2019)

# **Liste des acronymes**

**OS** : Operating system

**SE** : système d’exploitation

**IDE** : integrated development environment

**SDK** : Software Développement Kit

**XML** : Extensible Markup Language (langage de balisage extensible)

**JSON** : JavaScript Object Notation

**UML** : Unified Modeling Language / Langage de modélisation unifié

**APK** : Android Package Kit

# **Les Annexes**

## **Annexe 1 : Dictionnaire des attributs**

|  |  |  |
| --- | --- | --- |
| N° | Code | Définition |
| 1 | Id Company | Identifiant de l’entreprise |
| 2 | Company Name | Nom de l’entreprise |
| 3 | Language | Langage de l’application pour l’entreprise |
| 4 | Logo | Logo de l’entreprise |
| 5 | Time zone | Le Fuseau horaire de l’entreprise |
| 6 | Site\_default\_radius | Le rayon de zone du dépôt indiquée sur la carte |
| 7 | Id User | Identifiant de l’utilisateur |
| 8 | Is\_archived | Définir si le champ est archivé |
| 9 | Avatar | L’image de l’utilisateur |
| 10 | First\_name | Prénom de l’utilisateur |
| 11 | Last\_name | Nom de l’utilisateur |
| 12 | Email | L’email de l’utilisateur |
| 13 | Username | Pseudo de l’utilisateur |
| 14 | Type | Type de l’utilisateur |
| 15 | Is\_localized | Définir si l’utilisateur sera localisé |
| 16 | IdSite | Identifiant du dépôt |
| 17 | Is\_archived | Définir si le champ est archivé |
| 18 | Name | Nom du dépôt |
| 19 | Is\_default | Définir si c’est le dépôt par défaut |
| 20 | radius | Le rayon de zone du dépôt |
| 21 | IdPosition | Identifiant de la position |
| 22 | lastSync | Date de la dernière synchronisation |
| 23 | lastActivity | La dernière activité |
| 24 | Longitude | La longitude de la position |
| 25 | Latitude | La latitude de la position |
| 26 | IdTeam | Identifiant de l’équipe |
| 27 | Is\_archived | Définir si le champ est archivé |
| 28 | Color | Le couleur de l’équipe |
| 29 | Name | Le nom de l’équipe |
| 30 | ContractId | Identifiant du contrat |
| 31 | Is\_archived | Définir si le champ est archivé |
| 32 | Name | Le nom du contrat |
| 33 | Reference | Le référence du contrat |
| 34 | Specification | La spécification du contrat |
| 35 | contractTypeId | Identifiant du type de contrat |
| 36 | Is\_archived | Définir si le champ est archivé |
| 37 | Name | Nom du type de contrat |
| 38 | Abbreviation | L’abréviation du type de contrat |
| 39 | Color | Le couleur du type de contrat |
| 40 | contractPeriodId | Identifiant du période de contrat |
| 41 | Is\_archived | Définir si le champ est archivé |
| 42 | State | Statut du contrat |
| 43 | End\_on | Date de fin du contrat |
| 44 | Start\_on | Date de début du contrat |
| 45 | Hours\_sold | Nombre des heures vendues |
| 46 | Amount | Le prix du contrat |
| 47 | SiteId | Identifiant du chantier |
| 48 | Is\_archived | Définir si le champ est archivé |
| 49 | Comment | Commentaire à propos du chantier |
| 50 | Name | Le nome du chantier |
| 51 | Radius | Le rayon de la zone du chantier |
| 52 | IdAddress | Identifiant de l’adresse |
| 53 | Entered\_by\_user | Adresse entrer par l’utilisateur |
| 54 | Geocoder\_value | La valeur de géo codeur |
| 55 | Longitude | La longitude de l’adresse |
| 56 | Latitude | La latitude de l’adresse |
| 57 | CustomerId | Identifiant du client |
| 58 | Is\_archived | Définir si le champ est archivé |
| 59 | Civil\_status | L’état civil du client |
| 60 | First\_name | Le prénom du client |
| 61 | Last\_name | Le nom du client |
| 62 | Customer type | Le type du client |
| 63 | Company\_name | Le nom du l’entreprise |
| 64 | Comment | Commentaire à propos du client |
| 65 | IdSupplier | Identifiant du fournisseur |
| 66 | Is\_archived | Définir si le champ est archivé |
| 67 | Logo | Le logo du fournisseur |
| 68 | Tags | Les mots clés du fournisseur |
| 69 | Name | Le nom du fournisseur |
| 70 | Comment | Commentaire sur le fournisseur |
| 71 | Radius | Le rayon de la zone de l’emplacement du fournisseur |
| 72 | PersonalDetailsId | Identifiant des détails personnels |
| 73 | Is\_archived | Définir si le champ est archivé |
| 74 | Value | La valeur du détail personnel |
| 75 | Personal\_detail\_type | Type de détail personnel |
| 76 | Personal\_detailable\_type | Type de propriétaire de détail personnel |
| 77 | Personal\_detailable\_id | Identifiant de propriétaire de détail personnel |
| 78 | Label | Libellé de détail personnel |
| 79 | Order | Ordre d’affichage de détail personnel |
| 80 | InterventionId | Identifiant de l’intervention |
| 81 | Is\_archived | Définir si le champ est archivé |
| 82 | Achieved\_on | Date de réalisation |
| 83 | Appointement\_at | Date de rendez-vous |
| 84 | Created\_on | Date de création |
| 85 | Scheduled\_on | Date prévu de l’intervention |
| 86 | Started\_on | Date début de l’intervention |
| 87 | State | Statut de l’intervention |
| 88 | Comment | Commentaire sur l’intervention |
| 89 | Total\_duration | Durée total de l’intervention |
| 90 | Intervention\_order | Ordre de l’intervention |
| 91 | Is\_précise | Définir si la date de l’intervention est précisée |
| 92 | Scheduled\_start\_at | L’heure de début prévu pour l’intervention |
| 93 | Scheduled\_finish\_at | L’heure de fin prévu pour l’intervention |
| 94 | specificationId | Identifiant de spécification |
| 95 | Is\_archived | Définir si le champ est archivé |
| 96 | Name | Nom de la spécification de l’intervention |
| 97 | Details | Détails de la spécification de l’intervention |
| 98 | techniciansCount | Le nombre des techniciens pour l’intervention |
| 99 | Unit\_duration | La durée de l’intervention |
| 100 | Order | L’ordre de la spécification |
| 101 | Quantité | Le nombre des spécifications des pour l’intervention |
| 102 | ActivityId | Identifiant de l’activité |
| 103 | Is\_archived | Définir si le champ est archivé |
| 104 | activiyableType | Type d’activité |
| 105 | Finish\_at | Date de fin de l’activité |
| 106 | Start\_at | Date de début d’activité |
| 107 | activityTypeId | Identifiant de type d’activité |
| 108 | Name | Le nom de type d’activité |
| 109 | Slug | Le slogan de type d’activité |
| 110 | Color | Le couleur de type d’activité |
| 111 | Is\_archived | Définir si le champ est archivé |
| 112 | declarativeId | Identifiant du déclarative |
| 113 | Setteled\_on | Date de démarrage du déclarative |
| 114 | workDayId | Identifiant de la journée de travail courante |
| 115 | Setteled\_on | Date de démarrage de la journée de travail courante |
| 116 | State | Statut de la journée de travail courante |
| 117 | Signature | Signature du technicien lors de la clôture de la journée de travail courante |
| 118 | Current\_activities\_count | Nombre des activités faites par le technicien |
| 119 | Is\_archived | Définir si le champ est archivé |
| 120 | workDayHitoryId | Identifiant de l’historique de la journée de travail |
| 121 | Settled\_on | Date de démarrage de la journée de travail |
| 122 | State | Statut de la journée de travail |
| 123 | Comment | Commentaire sur la journée de travail |
| 124 | Signature | Signature du technicien lors de la clôture de la journée de travail |
| 125 | Is\_archived | Définir si le champ est archivé |
| 126 | settingId | Identifiant du paramètre d’activité |
| 127 | Key | Le nom du paramètre |
| 128 | Value | La valeur du paramètre |
| 129 | dataType | Le type de donnée du paramètre |
| 130 | InterventionSpecificationId | Identifiant du table InterventionSpecification |
| 131 | Is\_archived | Définir si le champ est archivé |
| 132 | intrventionId | Identifiant de l’intervention |
| 133 | specificationId | Identifiant de la spécification |

## **Annexe 2 : Identification des classes utilisé dans le sprint 1**

* Une **entreprise** est caractérisée par :

|  |  |  |
| --- | --- | --- |
| **N°** | **Les attributs** | **Les Fonctions** |
| **1** | 1 : Id Company  2 : CompanyName  3 : Language  4 : TimeZone  5 : SiteDefaultRadius | company()  findById() |

* Un **utilisateur** est caractérisé par :

|  |  |  |
| --- | --- | --- |
| **N°** | **Les attributs** | **Les Fonctions** |
| **2** | 6 : Id User  7 : IsArchived  8 : Avatar  9 : FirstName  10 : LastName  11: Email  12: UserName  13: Type  14: IsLocalized | admins()  technicians()  findById() |

* Une **position** est caractérisée par :

|  |  |  |
| --- | --- | --- |
| **N°** | **Les attributs** | **Les Fonctions** |
| **3** | 21 : Id Position  22 : LastSync  23 : LastActivity  24 : Longitude  25 : Latitude | getTechniciansPositions() |

## **Annexe 3 : Identification des classes utilisé dans le sprint 2**

* Une **Activité** est caractérisée par :

|  |  |  |
| --- | --- | --- |
| **N°** | **Les attributs** | **Les Fonctions** |
| **4** | 102 : ActivityId  103: Is\_archived  104 : ActivityableType  105 : finishAt  106 startedAt | findById()  activities() |

* Une **Adresse** est caractérisée par :

|  |  |  |
| --- | --- | --- |
| **N°** | **Les attributs** | **Les Fonctions** |
| **5** | 52 : IdAddress  53: EnteredByUser  54 : geocoderValue  55: longitude  56: latitude | findById()  getBySiteId() |

* Un **Contrat** est caractérisé par :

|  |  |  |
| --- | --- | --- |
| **N°** | **Les attributs** | **Les Fonctions** |
| **6** | 30 : contractId  31: IsArchived  32 : name  33: reference  34: specification | findById()  allContracts()  activeContracts()  findByIntervention() |

* Une **Période Contrat** est caractérisée par :

|  |  |  |
| --- | --- | --- |
| **N°** | **Les attributs** | **Les Fonctions** |
| **7** | 40 : contractId  41: IsArchived  42 : state  43: endOn  44: startOn  45: hoursSold  46: amount | findById()  contractPeriods()  contractPeriodId() |

* Une **équipe** est caractérisée par :

|  |  |  |
| --- | --- | --- |
| **N°** | **Les attributs** | **Les Fonctions** |
| **8** | 26 : Id Team  27: Is\_archived  28 : Color  29 : Name | findById()  Teams ()  teamTechnicians() |

* Une **intervention** est caractérisée par :

|  |  |  |
| --- | --- | --- |
| **N°** | **Les attributs** | **Les Fonctions** |
| **9** | 80 : interventionId  81: is\_archived  82 : achievedOn  83 : appointementAt  84: createdOn  85: scheduledOn  86: startedOn  87: state  88: comment  89: totalDuration  90: interventionOrder  91: isPrecise  92: scheduledStartAt  93: cheduledFinishAt | findById()  interventions()  countPlannings()  plannings()  removeIntervention()  updateIntervention()  countPlanningsOnGoing()  countPlanningsAchieved()  countPlanningsScheduled() |
|  |  |  |
|  |  |  |

* Une **InterventionSpecification** est caractérisée par :

|  |  |  |
| --- | --- | --- |
| **N°** | **Les attributs** | **Les Fonctions** |
| **10** | 130 : InterventionSpecification Id  131: Is\_archived  132 : InterventionId  133 : SpecificationId | findById() |

* Une **spécification** est caractérisée par :

|  |  |  |
| --- | --- | --- |
| **N°** | **Les attributs** | **Les Fonctions** |
| **11** | 94 : SpecificationId  95: Is\_archived  96 : Name  97 : Details  98: techniciansCount  99: unitDuration  100: order  101: quantity | getSpecifications()  findById()  getPrestations() |

## **Annexe 4 : Identification des classes utilisé dans le sprint 3**

* Une **TypeActivité** est caractérisée par :

|  |  |  |
| --- | --- | --- |
| **N°** | **Les attributs** | **Les Fonctions** |
| **12** | 107 : activityTypeId  108: name  109 : slug  110: color  111 : archived | findById()  activityTypes() |

* Un **Déclarative** est caractérisé par :

|  |  |  |
| --- | --- | --- |
| **N°** | **Les attributs** | **Les Fonctions** |
| **13** | 112 : declarativeId  113: settledOn | findById() |

* Une **JournéeTravail** est caractérisée par :

|  |  |  |  |
| --- | --- | --- | --- |
| **N°** | **Les attributs** | **Les Fonctions** | |
| **14** | 114 : workDayId  115: settledOn  116 : state  117:signature  118 : currentActivitiesCount  119: Is\_archived | | findById()  workdaysForTechnician()  workdaysForDate()  pendingWdTechnicians() |

* Un **HistoriqueJournée** est caractérisé par :

|  |  |  |
| --- | --- | --- |
| **N°** | **Les attributs** | **Les Fonctions** |
| **15** | 120 : workDayHistoryId  121: settledOn  122 : state  123: comment  124 : signature  125: Is\_archived | findById()  findByWorkDay() |

* Un **Paramètre** est caractérisé par :

|  |  |  |
| --- | --- | --- |
| **N°** | **Les attributs** | **Les Fonctions** |
| **16** | 126 : settingId  127: key  128 : value  129: dataType | findById()  automaticActivities()  configAutomaticActivity() |

## **Annexe 5 : Identification des classes utilisé dans le sprint 4**

* Un **Client** est caractérisé par :

|  |  |  |
| --- | --- | --- |
| **N°** | **Les attributs** | **Les Fonctions** |
| **17** | 57 : idCustomer  58: IsArchived  59 : CivilStaus  60: FirstName  61: LastName  62: CustomerType  63: CompanyName  64: comment | findById()  customers() |

* Un **Fournisseur** est caractérisé par :

|  |  |  |
| --- | --- | --- |
| **N°** | **Les attributs** | **Les Fonctions** |
| **18** | 65 : IdSupplier  66: IsArchived  67 : Logo  68: Tags  69: Name  70: Comment  71:Radius | findById()  suppliers() |

* Un **DetailPersonnels** est caractérisé par :

|  |  |  |
| --- | --- | --- |
| **N°** | **Les attributs** | **Les Fonctions** |
| **19** | 72 : personalDetailsId  73: IsArchived  74 : Value  75: PersonalDetailType  76: PersonalDetailableType  77: PersonalDetailableId  78:Label  79: Order | findById()  getDetailsById()  removePersonalDetail()  updatePersonalDetail() |

## **Annexe 6 : Identification des classes utilisé dans le sprint 5**

* Un **Chantier** est caractérisé par :

|  |  |  |
| --- | --- | --- |
| **N°** | **Les attributs** | **Les Fonctions** |
| **20** | 47 : SiteId  48: Is\_archived  49: Comment  50 : Name  51 : Radius | findById()  sites() |

* Un **TypeContrat** est caractérisé par :

|  |  |  |
| --- | --- | --- |
| **N°** | **Les attributs** | **Les Fonctions** |
| **21** | 35 : ContractTypeId  36: Is\_archived  37 : Name  38: Abbreviation  39 : Radius | findById()  contractTypes()  contractTypesLite() |