

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
Universite de Jendouba



Faculté des Sciences, Juridiques, Économiques et de Gestion de Jendouba
Département d'Informatique

MÉMOIRE DE MASTERE

MASTERE PROFESSIONEL EN COMMERCE ELECTRONIQUE

Conception et développement d'une application
web et mobile pour le diagnostic des maladies des
plantes des grandes cultures

Réalisé au sein de l'Institut Nationale des Grandes Cultures (INGC) :



Présenté par :
OuertaniAsma

Encadrant académique :
Mme Jouini Mouna (FSJEGJ)

Encadrant professionnel :
Mr. Bousselmi Anis (INGC)

Année universitaire : 2021/2022

Remerciements

C'est avec un grand plaisir que je réserve cette page en signe de gratitude et profonde reconnaissance à tous ceux qui m'aident de près ou de loin pour mener à la réalisation de ce travail.

Au terme de ce projet, je tenais à remercier dans un premier temps toute l'équipe pédagogique de la faculté des Sciences Juridiques, Économiques et de Gestion de Jendouba.

Je tiens à exprimer tout mon respect, à mon encadrante Madame Jouini Mouna pour sa disponibilité, ses conseils et son temps précieux dont il nous a fait part.

Mes profonds remerciements vont aux membres de jury qui ont accepté d'évaluer ce travail.

Enfin, je remercie ma famille et mes amis qui m'ont toujours soutenue dans mon cursus universitaire que dans mes choix personnels.

Table des matières

Remerciements	i
Table des matières	iv
Liste des figures	vii
Liste des tables	ix
Introduction générale	1
1 Cadre général du projet	3
1.1 Introduction	3
1.2 présentation de l'organisme d'accueil :	3
1.2.1 Présentation :	3
1.2.2 Organigramme :	4
1.3 Présentation du projet :	5
1.3.1 Contexte :	5
1.3.2 Objectifs du projet :	5
1.4 Étude de l'existant :	5
1.4.1 Étude des applications similaires :	6
1.4.2 Fonctionnement actuel dans L'INGC :	13
1.4.3 Critique de l'existant :	13
1.4.4 Solution proposée :	13
1.5 Méthodologie et formalisme adoptés :	14
1.5.1 Formalisme de modélisation :	14
1.5.2 Méthodologie agile	14
1.5.3 La méthode du SCRUM	15
1.6 Conclusion	17
2 Sprint 0	18
2.1 Introduction	18
2.2 Spécification des besoins	18
2.2.1 Identification des acteurs	18
2.3 Identification des besoins	19
2.3.1 Les besoins fonctionnels	19
2.3.2 besoins non fonctionnels	20
2.4 Planning du traitement des cas d'utilisation	20
2.4.1 Pilotage du projet avec Scrum	20

TABLE DES MATIÈRES

2.4.2	Les fonctionnalités du backlog :	21
2.5	Diagramme de cas d'utilisation global	22
2.6	Planification des sprints :	24
2.7	Environnement de travail :	24
2.7.1	Environnement matériel	24
2.7.2	Environnement logiciel :	25
2.8	Architecture générale de l'application	29
2.9	Architecture logicielle	30
2.10	Diagramme de déploiement :	31
2.11	Diagramme de Package :	32
2.12	Conclusion :	32
3	Mise en œuvre du sprint 1	33
3.1	Introduction	33
3.2	Planification de sprint 1	33
3.3	Backlog de sprint 1	33
3.4	Tableau des tâches	34
3.5	Phase d'analyse	34
3.5.1	Diagramme de cas d'utilisation global de sprint 1	34
3.6	Phase de conception	41
3.6.1	Conception statique de sprint	41
3.7	Conception dynamique du premier sprint	43
3.7.1	Les diagramme de séquence	43
3.8	Diagrammes d'activités :	49
3.9	Réalisation et Tests	52
3.9.1	Les Interfaces :	52
3.9.2	Les tests :	56
3.10	Conclusion	58
4	Mise en œuvre du sprint 2	59
4.1	Introduction	59
4.2	Planification de sprint2	59
4.2.1	Backlog de sprint2	59
4.2.2	Tableau des tâches	60
4.2.3	Phase d'analyse	60
4.3	Phase de conception	66
4.3.1	Conception statique de sprint	67
4.4	Conception dynamique du deuxième sprint	68
4.4.1	Les diagrammes de séquence	68
4.5	Diagrammes d'activités :	72
4.6	Réalisation et tests	75
4.6.1	Les Interfaces :	75
4.6.2	Les tests :	77
4.7	Conclusion	79
5	Mise en œuvre du sprint 3	80
5.1	Introduction	80
5.2	Planification de sprint3	80
5.2.1	Backlog de sprint	80
5.3	Tableau des tâches	81
5.4	Phase d'analyse	81
5.5	Phase de conception	85

TABLE DES MATIÈRES

5.5.1	Conception statique de sprint	85
5.6	Conception dynamique du troisième sprint	88
5.6.1	Les diagrammes de séquence	88
5.7	Diagrammes d'activités :	91
5.8	Réalisation et tests :	94
5.8.1	Les Interfaces :	94
5.8.2	Les tests :	95
5.9	Conclusion :	96
6	Mise en œuvre du sprint 4	97
6.1	Introduction	97
6.1.1	Backlog de sprint	97
6.2	Tableau des tâches	98
6.3	Phase d'analyse	98
6.4	Phase de conception	102
6.4.1	Conception statique de sprint	102
6.5	Conception dynamique du quatrième sprint	104
6.5.1	Les diagrammes de séquence	104
6.6	Diagramme de classe global	108
6.7	Réalisation et tests :	109
6.7.1	Les Interfaces :	109
6.7.2	Les Interfaces mobiles	109
6.7.3	Les Interfaces web	110
6.7.4	Les tests :	112
6.8	Conclusion :	112
Bibliographie		114

Table des figures

1.1	Logo de L'INGC	3
1.2	Localisation de l'organisme d'accueil	4
1.3	Organigramme de la société	4
1.4	Capture d'écran de la page d'accueil de l'applicationn	6
1.5	Capture d'écran des quelques pages de l'application	7
1.6	Capture d'écran de quelque page de l'application	8
1.7	Capture d'écran de la page d'accueil de l'application	9
1.8	Capture d'écran de la page accueil de l'application	10
1.9	Capture d'écran de la page DisCover de l'application	11
1.10	Capture d'écran de quelque page de l'application	12
1.11	Cycle de vie de la méthode Scrum[12]	16
2.1	Diagramme de Cas d'utilisation global	23
2.2	Planification des sprints	24
2.3	Logo de mongoDB Atlas	25
2.4	Logo node js	26
2.5	Logo React.js	26
2.6	Logo React Native	27
2.7	Logo visual code	27
2.8	Logo Overleaf	28
2.9	Logo bootstrap	28
2.10	Logo Draw.io	29
2.11	Architecture REST	30
2.12	Architecture logicielle	30
2.13	Diagramme de déploiement	31
2.14	Diagramme de Package	32
3.1	Planification de sprints1	33
3.2	Tableau de tâche de premier sprint	34
3.3	Diagramme de cas d'utilisation de sprint1	35
3.4	Raffinement du cas d'utilisation « S'inscrire »	36
3.5	Raffinement de cas d'utilisation « S'authentifier»	37
3.6	Raffinement de cas d'utilisation « Gérer profil»	38
3.7	Raffinement de cas d'utilisation « Gérer agriculteur»	39
3.8	Raffinement de cas d'utilisation « Gérer Ingénieur»	40
3.9	Diagramme de classe de sprint1»	42
3.10	Diagramme de séquence détaillées du cas « s'inscrire »	45

TABLE DES FIGURES

3.11	Diagramme de séquence détaillé du cas « s'authentifier »	46
3.12	Diagramme de séquence détaillé du cas « Ajouter Ingénieur»	47
3.13	Diagramme de séquence détaillé du cas « Modifier Ingénieur»	48
3.14	Diagramme de séquence détaillé du cas « Supprimer Ingénieur»	49
3.15	Diagramme d'activité du cas d'utilisation « S'inscrire»	50
3.16	Diagramme d'activité du cas d'utilisation « S'authentifier »	51
3.17	Diagramme d'activité du cas d'utilisation « Modifier Ingénieur»	52
3.18	Interface Inscription	53
3.19	Interface Modifier profil	54
3.20	Interface authetification	55
3.21	Interface Liste-Ingénieur	55
3.22	Interface Liste-Agriculteur	56
4.1	Planification de sprint 2	59
4.2	planification de sprint 2	60
4.3	Diagramme de cas d'utilisation de sprint 2	61
4.4	Raffinement de cas d'utilisation « Gérer plante»	61
4.5	Raffinement de cas d'utilisation « Gérer maladie»	63
4.6	Raffinement de cas d'utilisation « Gérer traitement»	65
4.7	Diagramme de classe de sprint 2	67
4.8	Diagramme de séquence de cas d'utilisation « Ajouter Plante »	69
4.9	Diagramme de séquence de cas d'utilisation « Chercher Plante »	70
4.10	Diagramme de séquence de cas d'utilisation « Modifier Plante »	71
4.11	Diagramme de séquence de cas d'utilisation « Supprimer Plante »	72
4.12	Diagramme d'activité du cas d'utilisation "Ajouter Plante "	73
4.13	Diagramme d'activité du cas d'utilisation "Chercher Plante "	74
4.14	Diagramme d'activité du cas d'utilisation "Modifier Plante "	75
4.15	Interface Ajout plante	76
4.16	Interface Liste-plante	76
4.17	Interface Liste-Traitement	77
5.1	Planification de sprints3	80
5.2	Planification de sprint 3	81
5.3	Diagramme de cas d'utilisation de sprint 3	82
5.4	Raffinement de cas d'utilisation « Participer au diagnostic»	82
5.5	Raffinement de cas d'utilisation « Créer rapport »	83
5.6	Raffinement de cas d'utilisation « Envoyer réclamation »	84
5.7	Diagramme de classe de sprint 3	86
5.8	Diagramme de séquence de cas d'utilisation « Participer au diagnostic»	89
5.9	Diagramme de séquence de cas d'utilisation « Créer rapport »	90
5.10	Diagramme de séquence de cas d'utilisation « Envoyer réclamation»	91
5.11	Diagramme d'activité de cas d'utilisation « Participer au diagnostic»	92
5.12	Diagramme d'activité de cas d'utilisation « Envoyer réclamation »	93
5.13	Interface Demande aux diagnostic	94
5.14	Interface envoyer réclamation	95
6.1	Planification du sprints4	97
6.2	planification de sprint 4	98
6.3	Diagramme de cas d'utilisation de sprint 4	99
6.4	Raffinement de cas d'utilisation « Répondre aux réclamations»	99
6.5	Raffinement de cas d'utilisation « Gérer réclamation»	100
6.6	Raffinement de cas d'utilisation « Visualiser statistique»	101

6.7	Raffinement de cas d'utilisation « Contacter»	102
6.8	Digramme de classe de sprint4	103
6.9	Diagramme de séquence de cas d'utilisation «Répondre aux réclamations»	105
6.10	Diagramme de séquence de cas d'utilisation «Visualiser statistique»	106
6.11	Diagramme de séquence de cas d'utilisation « Contacter»	107
6.12	Diagramme de classe global	108
6.13	Interface consulter réclamation	109
6.14	Interface consulter réclamation	110
6.15	Interface consulter réclamation	111
6.16	Interface supprimer réclamation	111

Liste des tableaux

1.1	comparaison entre les méthodes classiques et les méthodes agiles	15
2.1	Description des acteurs et leurs rôles.	19
2.2	Pilotage du projet avec Scrum.	21
2.3	Les fonctionnalités du backlog	22
3.1	Backlog de sprint1	34
3.2	Description textuelle de cas d'utilisation «s'inscrire pour l'acteur Agriculteur ».	36
3.3	Description textuelle de cas d'utilisation " S'authentifier "	37
3.4	Description textuelle de cas d'utilisation « Modifier profil ».	38
3.5	Description textuelle de cas d'utilisation " Gérer les agriculteurs "	39
3.6	Description textuelle de cas d'utilisation " Gérer les ingénieurs "	41
3.7	Dictionnaire de données de la table Utilisateur	43
3.8	Dictionnaire de données de la table Agriculteur	43
3.9	Dictionnaire de données de la table Ingénieur	43
4.1	Backlog de sprint1	60
4.2	Description textuelle de cas d'utilisation " Gérer plante "	62
4.3	Description textuelle de cas d'utilisation " Gérer maladie"	64
4.4	Description textuelle de cas d'utilisation " Gérer traitement"	66
4.5	dictionnaire de données de la table Plante	68
4.6	dictionnaire de données de la table Traitement	68
4.7	dictionnaire de données de la table Traitement	68
5.1	Backlog de sprint3	81
5.2	Description textuelle de cas d'utilisation "participer au diagnostic"	83
5.3	Description textuelle de cas d'utilisation "Créer rapport"	84
5.4	Description textuelle de cas d'utilisation "Envoyer réclamation "	85
5.5	Dictionnaire de données de la table Utilisateur	87
5.6	dictionnaire de données de la table Agriculteur	87
5.7	dictionnaire de données de la table Ingénieur	87
5.8	dictionnaire de données de la table Diagnostic	88
5.9	dictionnaire de données de la table Réponse	88
5.10	dictionnaire de données de la table Réclamation	88
6.1	Backlog de sprint4	98
6.2	Description textuelle de cas d'utilisation "Répondre aux réclamations "	100
6.3	Description textuelle de cas d'utilisation "Gérer réclamations "	101

6.4	Description textuelle de cas d'utilisation "Visualiser statistique "	101
6.5	Description textuelle de cas d'utilisation "Contacter "	102
6.6	dictionnaire de données de la table Ingénieur	103
6.7	dictionnaire de données de la table Réclamation	104
6.8	dictionnaire de données de la table contact	104

Introduction générale

Le secteur agricole occupe une place importante dans l'économie Tunisienne en contribuant à assurer la sécurité alimentaire du pays, la génération de revenus, les opportunités d'emploi, l'équilibre régional et la gestion des ressources naturelles.

De nos jours, les ravageurs des cultures peuvent avoir divers effets sur la récolte : réduire les rendements, réduire le stockage, affecter l'apparence et réduire la qualité saine des produits. Ces effets sont plus importants, la récolte peut être complètement détruite.

En effet, les plantes sont menacées par des maladies causées par des microorganismes : virus, bactéries ou encore champignons. Ces maladies causent d'importantes pertes de rendement, sur les cultures. Donc il faut trouver une solution Pour réduire la propagation des maladies agricoles.

La diffusion de l'information liée à l'agriculture aux agriculteurs dans les communautés les plus pauvres est plus facile avec l'aide de l'informatique, les systèmes informatiques intégrés, l'éducation en ligne et la prolifération des téléphones mobiles. Ce qui a rendu l'adoption de ces technologies plus favorables dans les pays en développement est l'augmentation de l'utilisation des téléphones mobiles et des ordinateurs portables. Ce qui représente une augmentation considérable par rapport, Les avantages primordiaux des TIC dans l'agriculture sont qu'ils réduisent les déchets de transport, transactionnels et de corruption. Cela peut entraîner la traçabilité des produits, le suivi des ravageurs et le stockage. Des études ont montré que les données de traçabilité permettent non seulement de gérer les risques alimentaires, mais augmentent également la performance globale de l'entreprise et réduire l'impact négatif de micro-environnemental de l'agriculture

Dans ce cadre, nous proposons dans ce présent mémoire une application web et mobile dont l'objectif est la plupart des dommages de micro environnement des agriculteurs essentiellement les maladies des plantes. Pour atteindre et réaliser notre objectif, nous avons choisi de décomposer ce rapport en six chapitres comme suit :

- Le premier chapitre intitulé « cadre du projet » dans lequel, nous présentons le contexte général de notre projet suivi d'une étude et critique de l'existant ainsi qu'une présentation de la solution proposée et de la méthodologie du travail adoptée.
- Le deuxième chapitre, intitulé « Sprint 0 », nous nous intéressons à définir le Backlog produit, et à présenter une vue architecturale et conceptuelle globale de notre application. C'est également à ce niveau que nous présenterons les outils et technologies utilisés pour le développement.

LISTE DES TABLEAUX

- Les quatre derniers chapitres, se concentreront sur l'étude et la réalisation des sprints de notre projet. Dans chaque sprint, nous commencerons, par le « Backlog du Sprint » qui décrit les tâches à faire et ensuite nous présenterons le diagramme de classe, les diagrammes de séquence et les diagrammes d'activité. Enfin, nous illustrerons l'exécution de notre application par des captures écrans. Nous finirons par une conclusion générale et les perspectives éventuelles de l'application développée.

Chapitre 1

Cadre général du projet

1.1 Introduction

Dans ce chapitre introductif, nous allons définir le cadre du projet en présentant tout d'abord la société de l'institut nationale des grandes cultures INGC dans laquelle nous avons fait notre stage. Ensuite, une évaluation de la situation actuelle et mettre en place les problèmes auxquels nous avons été confrontés et une étude de certaines applications similaires existantes, cette étude est suivie par une critique par apport à notre objectif. Enfin, nous présentons la méthodologie de conception choisie pour l'étape de la conception.

1.2 présentation de l'organisme d'accueil :

Dans cette section nous présentons l'organisme d'accueil ainsi que son organigramme

1.2.1 Présentation :

L'INGC[1] a été créé en 2009 en tant qu'institution publique à caractère non administratif. Il jouit d'une identité morale et de l'indépendance financière et il est sous la supervision du Ministère de l'Agriculture et des Ressources hydrauliques et de la pêche. L'INGC est l'un des principaux organismes publics participant à la concrétisation de la stratégie nationale visant à atteindre la sécurité alimentaire afin d'améliorer le développement de technologies agricoles durables et leur transfert et diffusion afin d'accroître le rendement des grandes cultures en termes de production et de qualité. [http ://ingc.com.tn/index.php/ar/](http://ingc.com.tn/index.php/ar/)



FIGURE 1.1 – Logo de L'INGC

- Forme juridique :**

Etablissement public à caractère non administratif

- Création :** 16 Mars(2009).

- Localisation :**

A130 Km de Tunis, délégation de bousalem, Gouvernorat de Jendouba.



FIGURE 1.2 – Localisation de l'organisme d'accueil

1.2.2 Organigramme :

Afin d'assurer une bonne prestation de ses services, L'institut National des Grandes Cultures INGC est basée sur l'interaction de plusieurs directions et départements complémentaires comme la montre la figure ci-dessous :

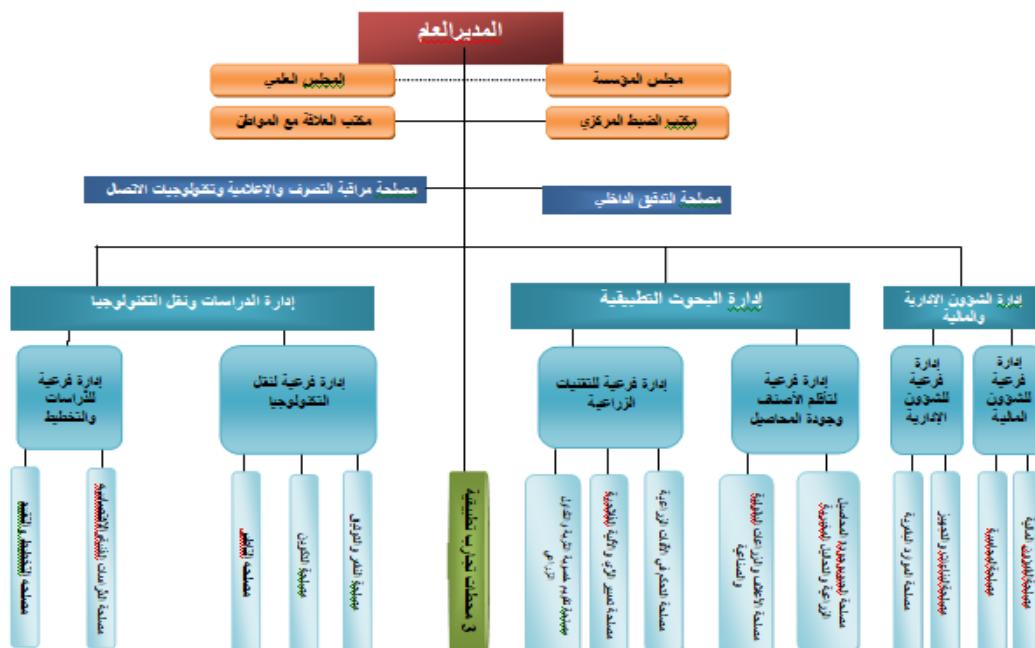


FIGURE 1.3 – Organigramme de la société

1.3 Présentation du projet :

1.3.1 Contexte :

Dans tout pays du monde et même dans toute la région, y compris la région de Jendouba, l'agriculture représente un secteur très important, car l'agriculture est l'un des secteurs qui soutiennent grandement l'économie tunisienne. Jendouba est considérée comme l'une des plus grandes zones agricoles de Tunisie, malgré l'importance de ce secteur, la ville présente encore de nombreuses lacunes et problèmes. Comme une forte augmentation de la proportion de maladies qui menacent les cultures.

Dans ce contexte, il est important pour les entreprises du secteur agricole de bien gérer leurs investissements, de formuler des stratégies pour maintenir leur croissance à long terme afin d'attirer plus de clients, d'accroître la fidélité des clients et l'efficacité opérationnelle. Il est important d'avoir un site web ou une application mobile qui fait répondre efficacement et rapidement des besoins des utilisateurs.

Dans ce cadre L'institut national des grandes cultures INGC est chargé de promouvoir et de développer le secteur agricole informatiser ses différents services. En effet, l'étude des symptômes des maladies des plantes jouent un rôle crucial pour le développement agricole. Pour cela l'institut national de grandes cultures INGC vise une application web et mobile qui permet aux agriculteurs de connaître les maladies des plantes en question et c'est le rôle de l'application de déduire et afficher la maladie et les traitements adéquats.

Il s'agit d'une application simple, facile et efficace qui assure une bonne supervision des travaux pour atteindre la traçabilité pour assurer l'évaluation et l'amélioration dans ce domaine.

1.3.2 Objectifs du projet :

Notre travail consiste à l'élaboration d'une nouvelle application web et mobile pour confirmer le statut sanitaire des plantes et d'identifier les agents pathogènes. Il permet aux agriculteurs la détection précoce, la gestion et le contrôle des maladies des plantes.

En plus des objectifs classiques que n'importe qu'elle entreprise cherche à réaliser à savoir :

- Minimiser la saisie des données et les risques d'erreurs
- Gain du temps

Nous avons voulu également atteindre les objectifs suivants :

- Déetecter les maladies des plantes
- Promouvoir l'amélioration de conditions alimentaires saines,
- Partager et échanger des informations sur les traitements.
- Participer à un diagnostic avec des symptômes plus répandu dans notre environnement

1.4 Étude de l'existant :

Le but de cette section est d'étudier et de découvrir les lacunes du système existant et de proposer des solutions appropriées. En effet, avant de commencer à concevoir notre application, nous avons analysé certaines applications similaires qui existent dans la littérature. Je cite principalement : Hortiquid [2], Di@gnoplant[3], Clinique des plantes[4], Étiquettes de jardin[5], Nature ID[6].

1.4.1 Étude des applications similaires :

1.4.1.1 Présentation de l'application Hortiquid[2]

Hortiquid, ce sont un site internet et une application qui permettent aux particuliers de poser leurs questions sur le jardinage à des experts de la SNHF. Ce service regroupe 150 experts bénévoles dans différents domaines : nomenclature, économie, génétique, santé des plantes, techniques horticoles. Plus, de 550 questions-réponses y sont recensées et accessibles.

Comment ça marche ?

Dans un premier temps, nous vous proposons de découvrir les réponses déjà publiées par les experts, soit en consultant directement la rubrique Réponses des experts, soit en utilisant le moteur de recherche. Si vous ne trouvez pas de réponse adéquate, nous vous invitons à poser votre question aux experts. Envoyez votre question détaillée et des photographies de l'objet de vos interrogations.



FIGURE 1.4 – Capture d'écran de la page d'accueil de l'applicationn



FIGURE 1.5 – Capture d’écran des quelques pages de l’application

1.4.1.2 Présentation de l’application Di@gnoplant[3] :

Di@gnoplant, ces outils d’aide au diagnostic sont non seulement disponibles sur ordinateur grâce au site internet e-phytia. Les applications sont simples d’emploi et permettent, en seulement quelques clics, d’identifier à l’aide de photos, les maladies et ravageurs des cultures. Une fois le diagnostic terminé, vous pouvez accéder aux fiches descriptives contenant de nombreuses photos, une description des symptômes, des éléments de biologie, les principaux facteurs de risques ainsi que les moyens de lutte. Vous pouvez également accéder directement à ces fiches grâce à l’index. Une application est développée pour chacune des cultures : pomme de terre, cultures tropicales, pomme, prune, tomate, salade, melon, courgette.

Pour cela, deux modules complémentaires sont proposés :

- Diagnostic : est un outil d’aide à l’identification des maladies par l’image qui permet de sérier progressivement la cause du problème rencontré, qu’il soit parasitaire ou non. L’utilisateur précise dans un premier, grâce à des séries d’images successives, la localisation, puis la nature des symptômes observés sur les plantes malades. Par la suite, il choisit parmi les séries d’images suivantes celle qui présente le ou les symptômes les plus ressemblants à ceux observés sur les plantes malades. Chaque image proposée est accompagnée d’une légende décrivant les symptômes observables et précisant leur cause. A terme, un diagnostic est proposé. Afin de le confirmer, il est possible de consulter d’autres symptômes de la maladie en cause dans une fiche. Dans cette dernière, des informations portant sur la biologie du bio agresseur en cause et sur les méthodes de protection permettant de le contrôler sont aussi proposées. De plus, il

est aussi possible de reprendre le diagnostic en cours et de tester les hypothèses associées à d'autres images aux symptômes comparables.

- Index : liste l'ensemble des maladies et des bioagresseurs de chaque culture ; ainsi, il est possible aux initiés d'accéder directement au nom d'une maladie, au nom latin d'un bioagresseur, et rend possible un accès direct aux fiches synthétisant les connaissances disponibles sur le sujet grâce à un filtre de recherche.

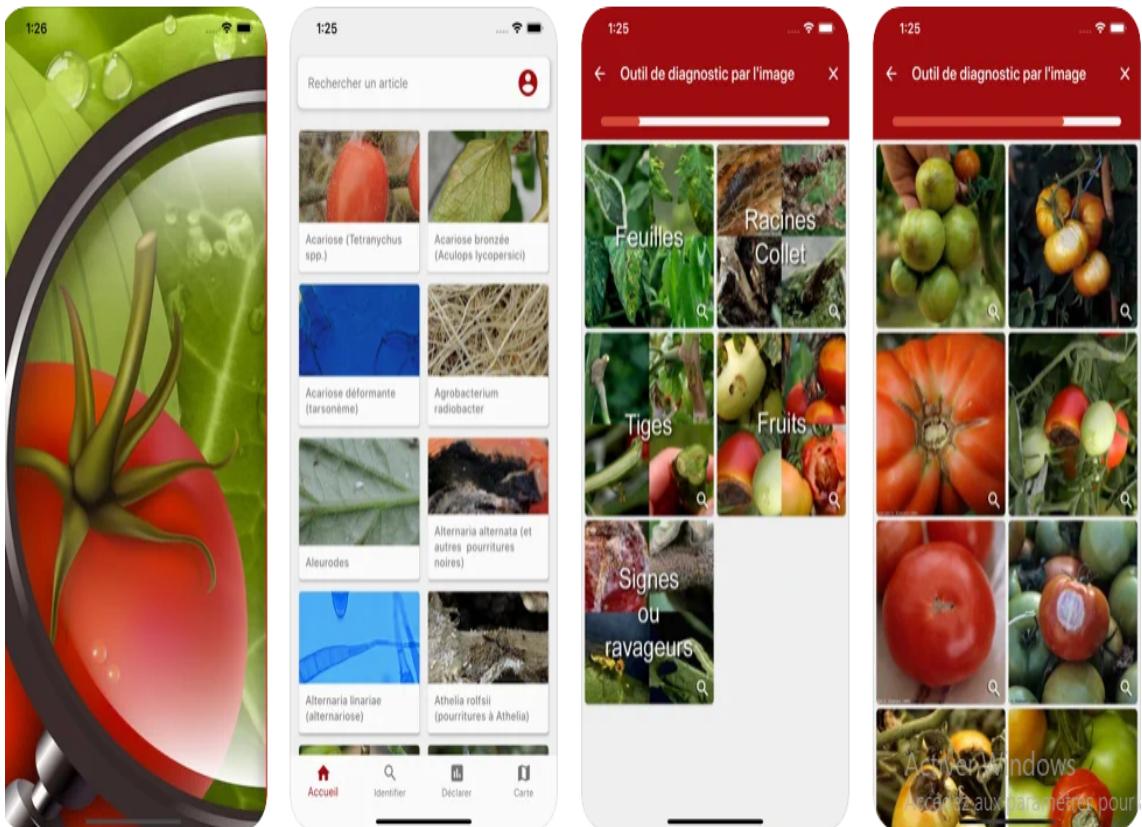


FIGURE 1.6 – Capture d'écran de quelque page de l'application

1.4.1.3 Présentation de l'application Clinique des plantes[4] :

Lancée en mars 2017, l'application "La Clinique des Plantes" se télécharge gratuitement sur IOS et Android, sur Apple Store et Play Store. Son utilisation est d'une simplicité enfantine. Le principe est d'établir le diagnostic du problème rencontré par le jardinier. Il suffit de choisir parmi un choix de plantes proposées automatiquement .On indique ensuite la partie de la plante où se situe le symptôme (tige, tronc, écorce, feuille, fleurs, fruits, racine). Puis à partir de vos observations, on détermine le type de l'infection. Le résultat tombe immédiatement avec plusieurs options. Si un utilisateur a un doute sur le type de plante, la maladie ou le parasite ? Un formulaire de contact direct avec les "Docteurs plantes" est disponible. Il faut impérativement envoyer une photo qui est bien plus efficace que toutes les explications du monde.



FIGURE 1.7 – Capture d'écran de la page d'accueil de l'application

1.4.1.4 Présentation de l'application Étiquettes de jardin[5]

Garden Tags permet de rejoindre toute une communauté de jardiniers. Il permet de publier des photos sur des plantes et des jardins tout en étant inspiré par des milliers de compagnons amoureux des plantes.

Cette application permet de prendre des photos et puis les postez sur les étiquettes de jardin. Les jardiniers experts qui utilisent l'application peuvent identifier la plante pour les utilisateurs et peuvent même les informer de toute maladie qu'elle pourrait avoir.

Elle permet également garder une trace des plantes des jardins des utilisateurs en les prenant en photo et en les cataloguant. Puis, recherchez une plante dans l'encyclopédie et par la suite de trouver de nombreuses informations, notamment sa résistance au gel, les besoins en lumière solaire, le niveau de pH, le temps de plantation idéal, etc.

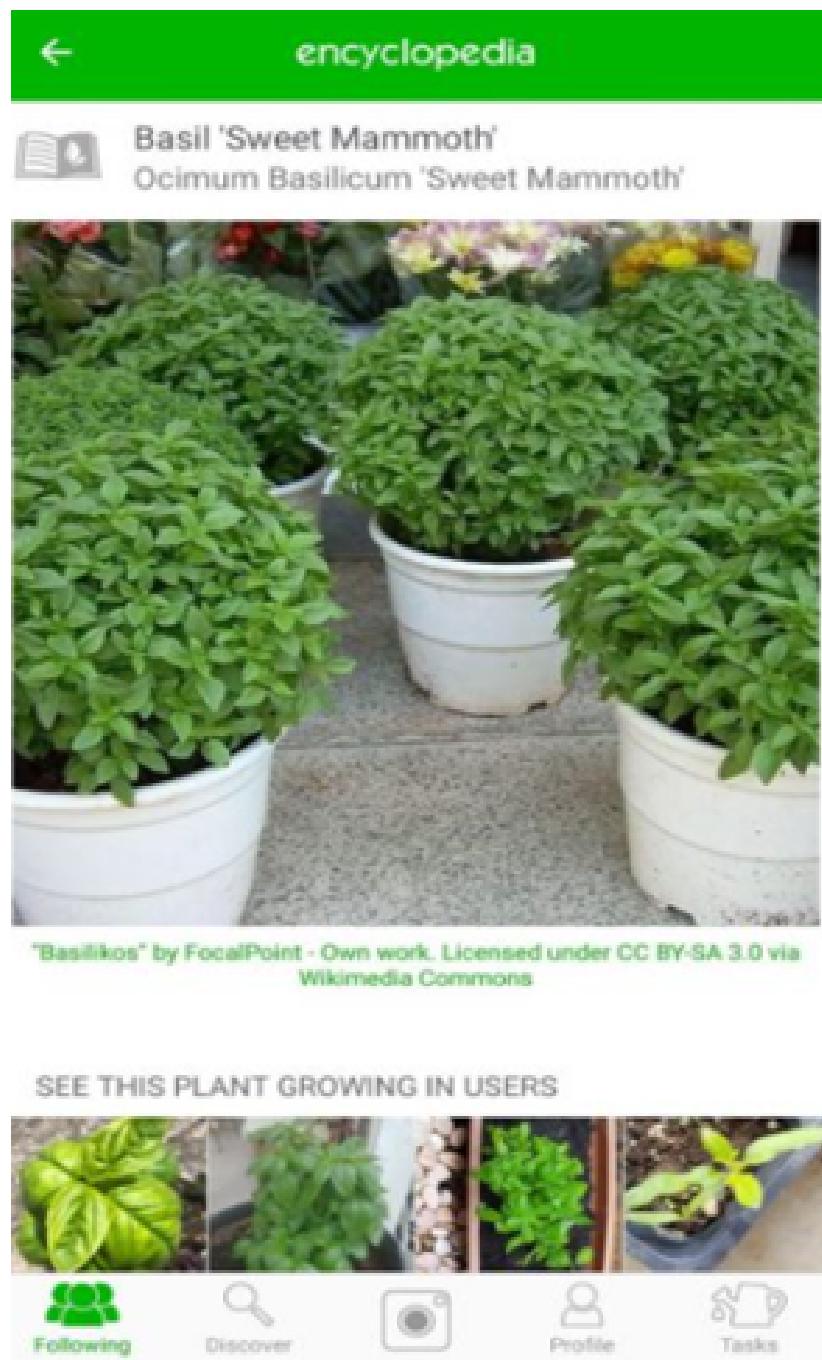


FIGURE 1.8 – Capture d'écran de la page accueil de l'application



FIGURE 1.9 – Capture d'écran de la page DisCover de l'application

1.4.1.5 Présentation de l'application Nature ID[6]

Cette application identifie les plantes d'une simple pression de doigt et donne plus de connaissance sur la nature qui entoure un utilisateur.

Vous est-il déjà arrivé de voir une plante et de vous demander ce qu'elle est ? Aimeriez-vous avoir un expert personnel en botanique à votre disposition en cas de besoin ? Voici votre identificateur de plante !

COMMENT UTILISER

- Pour identifier les plantes via cette application, il suffit de diriger l'appareil photo sur le sujet

qui intéresse l'utilisateur et de prendre une photo.

- Obtenez une description de chaque plante.
- joutez des plantes aux jardins.
- Définissez des rappels de soins.
- Diagnostiquez une plante grâce à notre identificateur de maladie végétale.
- Utilisez les photos sur les différents téléphones mobiles.

CARACTERISTIQUES AVANCEES

- Identifiez les plantes - il y en a plus de 10 000 dans leur base de données !
- Précision de 95 pourcent photographiez une feuille, une fleur ou un arbre et voyez par les utilisateurs
- L'algorithme de reconnaissance a été considérablement amélioré pour vous permettre d'obtenir l'identification des plantes la plus précise qui soit !
- Reconnaissez les arbres et reconnaître les fleurs tout au long de la journée !
- Recherche par nom - trouvez facilement une espèce de plante en entrant son nom
- Prenez des photos ou téléchargez des photos depuis votre appareil.
- Profitez d'une interface claire et agréable de cette application de reconnaissance des plantes efficaces et ergonomiques !

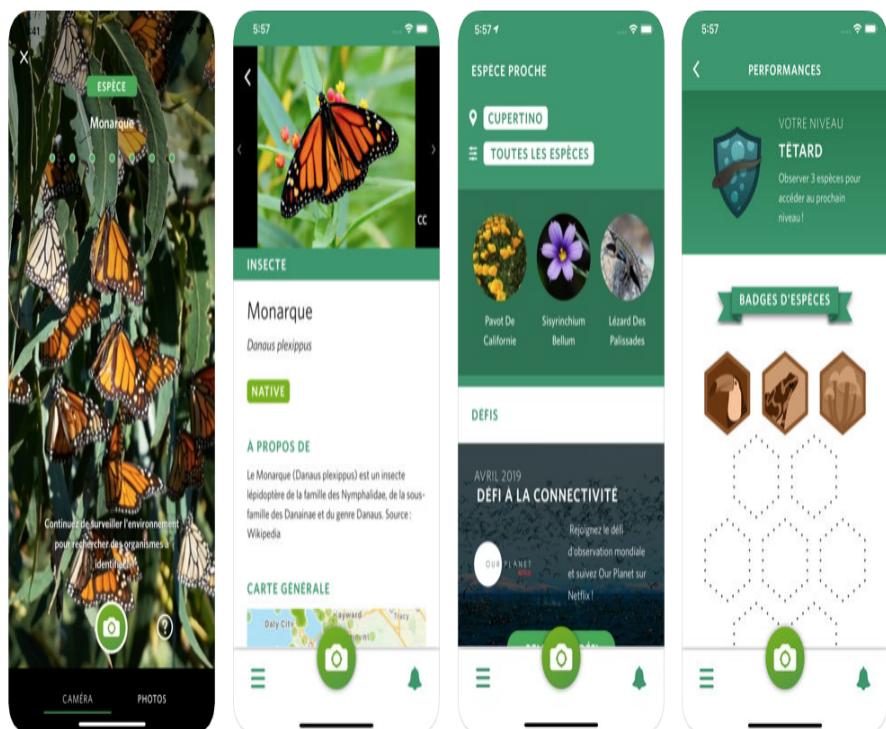


FIGURE 1.10 – Capture d'écran de quelque page de l'application

1.4.2 Fonctionnement actuel dans L'INGC :

La méthode adoptée pour effectuer le diagnostic des maladies des plantes actuellement est basée sur des fiches élaborées manuellement, qui sont éparpillés partout dans l'institut. toutes les tâches sont alors manuelles et il n'y a plus de système informatique qui gère les maladies des plantes.

1.4.3 Critique de l'existant :

L'étude de l'existant des applications web de diagnostic des maladies des plantes, nous a permis de dégager un certain nombre de lacunes :

- Difficulté d'utilisation de ces applications pour les utilisateurs et manque d'efficacité.
- Problèmes d'ergonomie et interfaces non visibles.
- Difficulté de traitement des maladies.
- Absence des sites tunisiens spécifiés aux agricultures et plantes.
- Beaucoup de temps perdu lors de la recherche d'informations sur les plantes et les maladies associées.
- Manque de campagnes de sensibilisation.
- Manque des cabinets d'ingénieurs. Toutes ces faiblesses ont eu un impact négatif sur l'agriculture et ont par la suite réduit la production agricole.

D'autre part, et comme déjà signalé dans la section précédente, la méthode actuelle adoptée pour faire la gestion des maladies des plantes au sein de l'institut nationale des grandes cultures INGC est manuelle, le stockage des dossiers des agriculteurs est effectué sur des supports papier. Ce mécanisme engendre également quelques problèmes :

- Perte de temps : par le remplissage manuel des données.
- Données éparpillées : en cas d'un incident, volée ou mal classée à l'archive.
- incohérences des données.

1.4.4 Solution proposée :

Afin de combler les lacunes du logiciel de gestion des maladies agricoles existant, nous fournissons une solution pour assurer le bon fonctionnement des différentes tâches de gestion des maladies agricoles et des traitements en considérant l'expérience utilisateur en termes de vitesse et d'efficacité. Cette solution a été proposée pour les raisons suivantes :

- Participer au diagnostic plus courant dans notre environnement.
- Faire des photos de maladies des plantes, ainsi qu'un bref questionnaire pour les demandes des agriculteurs.
- Créer rapport (résultat du diagnostic(réponse)).
- Afficher les causes et la prévention de la maladie.
- Afficher les traitements pour chaque maladie.
- Ajouter un espace de communication entre l'agriculteur et l'administration .
- Consulter les traitements (informations, mode utilisation)
- Ajouter des images pour les traitements et son prix.
- Visualiser les statistiques.
- Réduire la perte de temps et Accélérer le traitement des maladies.

Ses dernières fonctionnalités seront bénéfiques et bénéfiques pour résoudre les problèmes mentionnés ci-dessus.

1.5 Méthodologie et formalisme adoptés :

Avant de mettre en œuvre un projet informatique, il est nécessaire de choisir une méthode de travail et un processus de suivi afin d'obtenir enfin un logiciel fiable. Dans cette section, nous présentons le formalisme utilisé dans la conception des systèmes tout au long du stage et les méthodes utilisées pour gérer et suivre nos activités.

UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue [7].

1.5.1 Formalisme de modélisation :

Pour la spécification et la conception de notre projet, nous avons choisi la notation UML2, qui est un langage de modélisation utilisé pour décrire les exigences, les systèmes de documents et l'architecture logicielle d'esquisse. Nous fournissons quelques diagrammes que nous pensons utiles et suffisants pour comprendre le projet :

- Des diagrammes de cas d'utilisation décrivent le comportement fonctionnel de notre application,
- Diagramme de séquence pour décrire comment les éléments du système s'affectent les uns les autres et l'interaction avec les participants,
- Un diagramme de classes pour montrer la structure interne du système.
- Des diagrammes d'activités pour représenter le flux de travail à partir d'un point de départ au point d'arrivée.

1.5.2 Méthodologie agile

La méthode Agile se base sur un cycle de développement qui porte le client au centre. Le client est impliqué dans la réalisation dès le début à la fin du projet. L'implication du client dans le processus permet à l'équipe d'obtenir un feedback régulier afin d'appliquer directement les changements nécessaires. Cette méthode vise à accélérer le développement d'un projet. De plus, elle assure la réalisation d'une application fonctionnelle tout au long de la durée de sa création. Les méthodes agiles se basent sur 4 valeurs fondamentales [8] :

- Les individus et interactions plutôt que processus et outils.
- Développement logiciel plutôt que documentation exhaustive.
- Collaboration avec le client plutôt que négociation contractuelle.
- Ouverture au changement plutôt que suivi d'un plan rigide.

Afin de fixer une méthode particulière, nous avons effectué une étude comparative entre les méthodes classiques et la méthode agile. Nous présentons dans le tableau comparatif entre l'approche classique et l'approche agile.

Tableau 1 : comparaison entre les méthodes classiques et les méthodes agiles[9]

	Approche classique	Approche Agile
Cycle de vie	Phases séquentielles	Itératif et incrémentale
Gestion des risques	Processus strict et rigoureux de gestion des risques.	Satisfaction du client par la livraison de valeur souhaitée.
Mesure des succès	Respect des engagements initiaux en termes de couts, de budget et de niveau de qualité	Satisfaction du client par la livraison de valeur souhaitée.
Equipe	Equipe avec ressources spécialisés dirigées par un chef de projet	Equipe responsabilisée, soutenue par le chef de projet
Qualité	Contrôle qualité à la fin de cycle de développement.	Contrôle qualité permanent au niveau du produit et du processus.
Planification	Prédiktive	Adaptative.
Changement Résistance au changement.	Processus lourds de gestion des changements acceptés.	Accueil favorable au changement intégré dans le processus

TABLE 1.1 – comparaison entre les méthodes classiques et les méthodes agiles.

Après l'étude comparative présentée dans le tableau ci-dessus, nous avons décidé d'utiliser Scrum comme méthode de conception pour la gestion du processus du développement.

1.5.3 La méthode du SCRUM

La méthodologie Scrum est une méthodologie 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 ». Un sprint peut avoir une durée qui varie généralement entre deux semaines et un mois, et à la fin de chaque sprint, l'équipe présente un produit partiel qui, peut être considéré comme livrable[10]. Avant chaque sprint, les tâches sont estimées en temps et en complexité à l'aide de certaines pratiques comme le « planning poker ». Ces estimations permettent à la fois de planifier les livraisons, mais aussi d'estimer le coût de ces tâches auprès du client. La figure suivante présente le processus de déroulement de la méthode SCRUM [11].

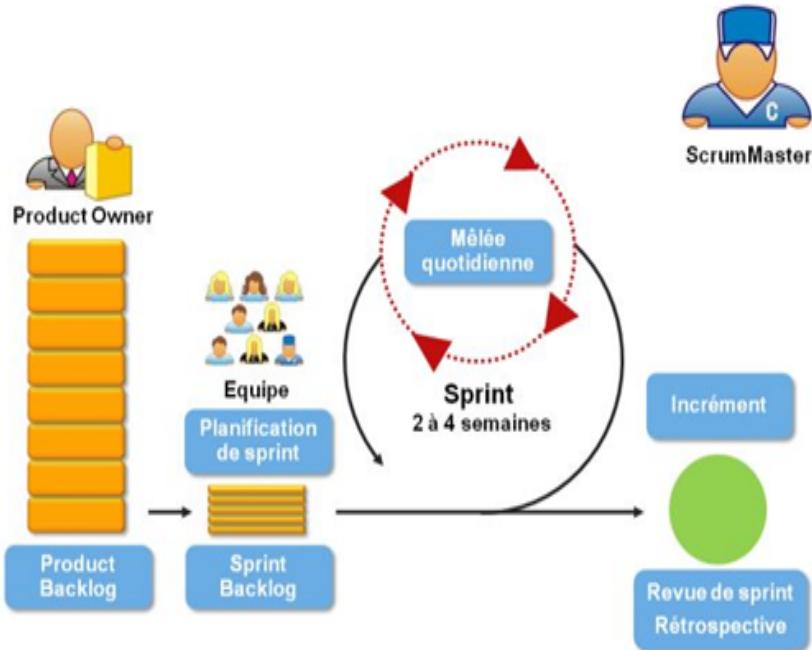


FIGURE 1.11 – Cycle de vie de la méthode Scrum[12]

Comme indiqué dans la figure cidessus, pour mettre en place la méthode SCRUM, il faut dégager dans un premier lieu le maximum de fonctionnalités à réaliser pour former le Backlog du produit. En second lieu, définir les priorités des fonctionnalités et choisir lesquelles seront réalisées dans chaque itération. Par la suite, focaliser l'équipe de façon itérative sur l'ensemble des fonctionnalités à réaliser dans des itérations appelées Sprints. Un sprint abouti toujours à la livraison d'un produit partiel fonctionnel appelé incrément. Ainsi, vers la fin de chaque Sprint, une réunion aura lieu pour effectuer la revue de l'itération. L'objectif de cette réunion consiste à valider l'incrément qui a été produit pendant l'itération.

1.5.3.1 Les rôles SCRUM :

SCRUM définit trois rôles principaux [13] :

1. Product Owner (Directeur du produit) :

Il s'agit du représentant officiel du client au sein d'un projet Scrum. Il est l'interlocuteur principal du Scrum Master et des membres de l'équipe. Il définit les besoins du produit et rédige les spécifications. Il peut se faire aider de responsables fonctionnels pour la rédaction des spécifications. Il est également chargé de définir et prioriser les user stories pour chaque sprint.

2. Scrum Master :

Il s'agit d'une personne chargée de veiller à la mise en application de la méthode et au respect de ses objectifs. Il ne s'agit pas d'un chef de projet, mais d'une personne chargée de lever les obstacles éventuels qui empêcheront l'avancement de l'équipe et du projet pendant les différents sprints.

3. Développement Team (L'équipe) :

Il s'agit des personnes chargées de la réalisation du sprint et d'un produit utilisable en fin de sprint. Il peut s'agir de développeurs, architectes, personnes chargées de faire des tests fonctionnels.

1.5.3.2 Les artefacts de Scrum[14]

Les artefacts de Scrum sont spécialement conçus pour assurer une meilleure compréhension et la transparence d'informations essentielles. Les artefacts de Scrum sont :

1. Backlog du produit :

Le backlog est la liste des fonctionnalités successives à accomplir pour réaliser le projet. Le propriétaire du produit est responsable de prioriser les fonctionnalités. Ces derniers servent à articuler et à finaliser ce que le client désire obtenir. Des fonctionnalités peuvent être ajoutées ou retirées à tout moment au backlog de produit. Cette flexibilité offerte au propriétaire du produit est un grand avantage de Scrum. Dans la méthodologie Scrum, le projet est décomposé en fonctionnalités appelées User Stories.

2. Backlog du sprint :

A chaque sprint, on choisit une partie des fonctionnalités du backlog de produit pour les réaliser. La réalisation de ces fonctionnalités doit permettre d'atteindre un objectif de sprint défini au début de celui-ci. Chaque fonctionnalité est alors découpée en tâches qui pourront être réalisées par différents membres de l'équipe.

1.6 Conclusion

Dans ce chapitre, nous avons présenté un aperçu général sur notre projet. Ensuite nous avons passés à l'étude de l'existant et préciser les critiques ainsi que les solutions à atteindre. Puis nous avons présentés la méthodologie et le formalisme adopté. Maintenant nous sommes en mesure d'entamer notre deuxième chapitre qui contiendra une planification du projet et une identification des fonctionnalités de notre application.

Chapitre 2

Sprint 0

2.1 Introduction

Dans ce chapitre, nous présentons la première phase du projet avec la méthodologie Scrum, à savoir la phase de planification ou « Sprint Zéro ». Dans cette phase, nous identifions tout d'abord les acteurs de notre projet, les exigences fonctionnelles et non fonctionnelles. Ensuite, nous allons présenter les besoins de notre système à travers le diagramme de cas d'utilisation global, finissons par produire le backlog initial ainsi qu'une première planification des sprints.

2.2 Spécification des besoins

La Spécification des besoins constitue la phase de départ de toute application à développer. Au niveau de cette phase nous allons identifier les besoins de notre application. Nous distinguons des besoins fonctionnels qui présentent les fonctionnalités attendues de l'application et les besoins non fonctionnels pour éviter le développement d'une application non satisfaisante.

2.2.1 Identification des acteurs

Un acteur représente une entité externe qui interagit avec le système. En réponse à l'action d'un acteur, le système fournit un service qui répond à ses besoins [15]. C'est une abstraction d'un ensemble des rôles joués par des entités externes (utilisateur, dispositif matériel ou autre système) et qui interagissent directement avec le système étudié. Notre application contient trois principaux acteurs :

- Administrateur,
- Agriculteur,
- Ingénieur.

Elle regroupe trois parties :

- Une partie "back office" accessible que par l'administrateur.
- Une deuxième partie "front office" c'est la partie Androïde de l'application faite pour l'agriculteur.
- Une troisième partie "front office" aussi est une autre partie Androïde de l'application faite pour l'ingénieur.

Le tableau ci dessous présente la liste d'acteurs et leurs rôles :

Acteurs	Description
Administrateur	L'administrateur joue un rôle primordial et fondamental, c'est la seule personne qui dispose du droit d'exécuter les différentes tâches offertes par notre application. En effet, il peut saisir des informations nécessaires, il peut aussi soit les mettre à jour soit les supprimer. C'est une personne qui se connecte à notre application pour gérer les services suivants : gérer les agriculteurs, gérer les ingénieurs.
Agriculteur	Acteur principal, sa fonction est de participer au diagnostic sur l'application. De plus il peut contacter l'ingénieur.
Ingénieur	Acteur principal, sa fonction est de contacter un agriculteur .De plus il est capable de créer un rapport pour le diagnostic de maladie.

TABLE 2.1 – Description des acteurs et leurs rôles.

2.3 Identification des besoins

Nous exposons dans ce qui suit leurs besoins fonctionnels ainsi que les besoins non fonctionnels communs à tous les acteurs.

2.3.1 Les besoins fonctionnels

Après une étude de contexte de système, cette partie est réservée à la description des exigences fonctionnelles des différents acteurs de l'application. Ces besoins se regroupent dans le diagramme de cas d'utilisation. Suite aux différentes réunions avec les responsables du projet et l'équipe, nous avons fait le choix d'incorporer à la future application les fonctionnalités suivantes :

L'administrateur doit bénéficier des fonctionnalités suivantes :

- S'authentifier : l'administrateur doit saisir son adresse et son mot de passe pour pouvoir connecter à son espace personnelle.
- Gérer les utilisateurs (agriculteur, ingénieur) : l'administrateur a l'accès de Ajouter, supprimer et consulter un compte d'ingénieur et d'agriculteur.
- Gérer maladies des plantes : L'administrateur peut modifier, ajouter ou supprimer une maladie associée à une plante donnée.
- Gérer les plantes : L'administrateur peut modifier, ajouter, consulter, chercher ou supprimer une plante.
- Gérer les traitements : L'administrateur peut modifier, ajouter, consulter ou supprimer un traitement.
- Gérer les réclamations : l'administrateur a l'accès de consulter la liste des réclamations et les supprimer.

- Visualiser statistique : l'administrateur peut visualiser statistique pour les maladies les plus popularisées.

L'Agriculteur (Client) sera capable de :

- S'inscrire : l'agriculteur remplit le formulaire d'inscription en créant une adresse et un mot de passe.
- S'authentifier : l'agriculteur doit écrire l'adresse et mot de passe pour connecter à son espace personnel.
- Gérer profil : l'agriculteur peut consulter ou modifier son profil.
- Participer au diagnostic : l'agriculteur doit choisir une plante, puis prendre une photo sur son plante, après il répond aux questions et finalement il reçoit le résultat de diagnostic dans un rapport.
- Passer réclamation : on cas de problème l'utilisateur peut faire une réclamation.
- Contacter : l'utilisateur peut contacter l'administration .

L'ingénieur sera capable de :

- S'authentifier : l'ingénieur doit écrire l'adresse et mot de passe pour connecter à son espace personnel.
- Effectué diagnostic : l'ingénieur faire un diagnostic puis suggérer les traitements associés à une maladie donnée.
- Créer les rapports : L'ingénieur est capable de créer des rapports pour le diagnostic de maladie. puis il est envoyé à l'agriculteur.
- Répondre aux réclamations : L'ingénieur doit recevoir et faire une résoudre aux problèmes de ces agriculteurs.

2.3.2 besoins non fonctionnels

Les besoins non fonctionnels décrivent toutes les contraintes techniques, ergonomiques et esthétiques auxquelles est soumis le système pour sa réalisation et pour son bon fonctionnement. Et ce qui concerne notre application, nous avons dégagé les besoins suivants :

- **Convivialité de l'interface graphique** : l'application doit fournir une interface conviviale et simple pour tout type d'utilisateur, car elle présente le premier contact de l'utilisateur avec l'application et par le biais de celle-ci on découvrira ses fonctionnalités..
- **Sécurité** :L'accès à l'application doit être sécurisé, les informations de chaque utilisateur doivent donc être protégées.
- **La fiabilité** : l'application doit être fiable. Elle ne doit pas contenir des bugs et ne doit pas affecter la performance du système.
- **Haute disponibilité** : L'application doit être opérationnelle tous les jours 7j/7 24h/24.
- **Haute disponibilité** : La modularité de code : Avoir un code simple facile à maintenir et à comprendre en cas de besoin.

2.4 Planning du traitement des cas d'utilisation

2.4.1 Pilotage du projet avec Scrum

Nous présentons dans cette partie l'ensemble des acteurs participants au déroulement des différentes phases du projet. Dans le tableau ci dessous, nous présentons les rôles, les acteurs associés et leurs missions.[15]

Détails de l'équipe	
Nom Projet	Application web,mobile pour le diagnostic des maladie des plantes des grandes cultures.
Propriétaire (Product Owner)	Mme MounaJouini
Directeur de produit (Scrum Master)	Mr Anis bousselmi
Membre de l'équipe (Team members)	Asma ouertani

TABLE 2.2 – Pilotage du projet avec Scrum.

2.4.2 Les fonctionnalités du backlog :

Le Backlog de produit correspond à une liste priorisée des besoins et des exigences du client. Les éléments du Backlog de produit, appelé aussi les histoires utilisateurs, sont formulés en une ou deux phrases décrivant de manière claire et précises la fonctionnalité désirée par le client, généralement, écrit sous la forme suivante « En tant que X, je veux Y, afin de Z. Le Backlog de produit présenté dans le tableau 1 comprend les champs suivants :

- **ID** : C'est un nombre unique et auto-incrémenté pour chaque histoire utilisateur.
- **User story** : C'est une phrase décrivant la fonctionnalité désirée par le client.
- **Priorité** : C'est la valeur métier qui dirige la priorisation du développement des histoires utilisateurs suivant les attentes et les besoins du client, allant de 0 à 100.

ID	Feature	User Story	Pri- orité	Comp- lexité
1	Gérer Agriculteur	En tant qu'administrateur, je peux contrôler mes agriculteurs pour rendre notre application représentative et conforme.	1	M
2	Gérer Ingénieur	En tant qu'administrateur, je peux contrôler mes ingénieurs pour rendre notre application représentative et conforme.	1	M
3	S'inscrire	En tant que Agriculteur, je peux m'inscrire pour accéder aux fonctionnalités offertes par l'application.	1	M
4	S'authentifier	En tant qu'administrateur ou agriculteur ou ingénieur, je dois m'authentifier pour assurer une connexion sécurisée.	1	M
5	Gérer Profil	En tant qu'agriculteur, je dois modifier mon profil.	1	M
6	Gérer plante	En tant que administrateur, je peux gérer les plantes.	2	S
7	Gérer maladie	En tant qu'administrateur, je peux gérer les maladies associées à une plante	2	S
8	Gérer traitement	En tant qu'administrateur, je peux gérer le traitement associé à une maladie	2	S
9	Participer au diagnostic	En tant qu'agriculteur, je veux prendre une photo qui apparaîtra sur ma plante et répondre au questionnaire à ce sujet, afin que le système élégant puisse me guider pour connaître le nom de la maladie.	3	S
10	Gérer réclama-tions	En tant qu'administrateur, je peux consulter ou supprimer les réclamations, Et En tant qu'ingénieur je peux résoudre les problèmes de mes agriculteurs.	3	S
11	Créer rapport	En tant que ingénieur, je dois créer des rapports sur le diagnostic des maladies	3	C
12	visualiser statis-tiques	En tant qu'administrateur je peux visualiser les statistiques sur les maladies et les traitements.	4	C
13	Contacter	En tant qu'agriculteur, je peux contacter l'administration.	4	C
14	Envoyer réclama-tions	En tant qu'agriculteur, je peux faire une réclamation en cas d'un problème afin que j'obtienne la réponse	4	C

TABLE 2.3 – Les fonctionnalités du backlog

2.5 Diagramme de cas d'utilisation global

Le diagramme de cas d'utilisation a pour but de donner une vision globale sur les fonctionnalités les plus importants de futur système. Il permet de représenter l'ensemble d'actions réalisées par le

système en réponse à une action d'un acteur.

Le diagramme de cas d'utilisation global de notre système est représenté par la figure ci dessous. Ce diagramme nous donne une vue globale sur notre système ainsi que les principaux besoins que notre application doit satisfaire. Il permet aussi de schématiser les interactions entre les différents acteurs de notre système.

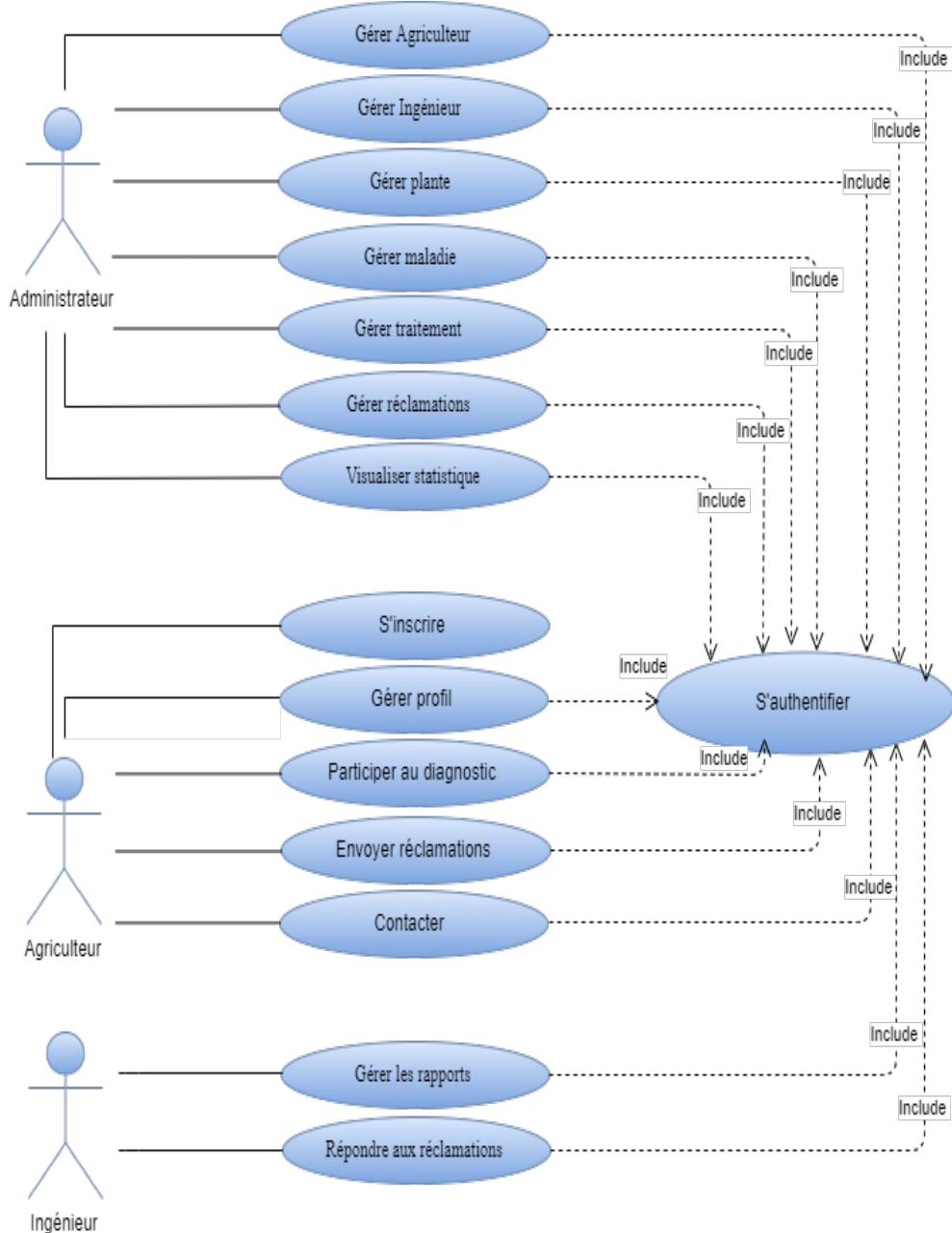


FIGURE 2.1 – Diagramme de Cas d'utilisation global

2.6 Planification des sprints :

Afin de planifier l'ordre de travail et le découpage des sprints, une réunion de planification avec l'équipe de travail a eu lieu après l'élaboration du Product Backlog. À chaque sprint, nous avons choisi une partie des fonctionnalités du backlog de produit pour les réaliser. La réalisation de ces fonctionnalités doit permettre d'atteindre un objectif de sprint défini au début de celui-ci. Chaque fonctionnalité est alors découpée en tâches qui pourront être réalisées.

Dans ce découpage il faut tenir compte de l'importance des cas d'utilisation, et de leurs dépendances.

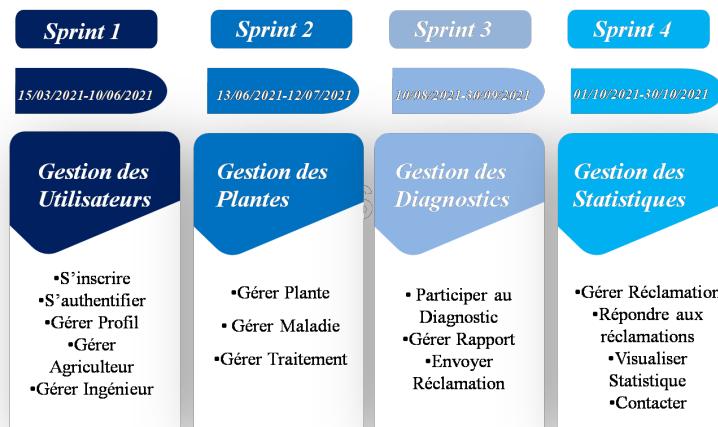


FIGURE 2.2 – Planification des sprints

2.7 Environnement de travail :

Cette partie couvre les prés requis matériels et logiciels qui sont utilisés pour la réalisation de notre application.

2.7.1 Environnement matériel

Pour la réalisation de notre projet, nous avons utilisé un ordinateur caractérisé par :

- Marque :HP
- Processeur : Intel(R) Core(TM) 2 Duo CPU T6570 @ 2.10 GHz 2.10 GHZ
- RAM : 4 Go — Disque dur : 500 Go — Système d'exploitation :
- Windows 10

Pour les différentes étapes de test, d'installation et du déploiement de l'application nous avons eu besoin d'un terminal mobile supportant le système d'exploitation Android 9.0 dont les caractéristiques sont les suivantes

- Nom : Galaxy A10 s.
- Processeur : HelioP22,2.0 GHz
- Système d'exploitation (OS) : Android 9.0.
- Mémoire interne : 16 Go / 2 Go RAM.

2.7.2 Environnement logiciel :

Afin de mieux comprendre le fonctionnement de l'application web, on a utilisé les outils logiciels suivants :

★ Atlas MongoDB :

MongoDB est le meilleur moyen de créer des applications basées sur les données. C'est une base de données NoSQL orientée document. Elle est utilisée pour le stockage de volumes massifs de données. Contrairement à une base de données relationnelle SQL traditionnelle.

MongoDB ne repose pas sur des tableaux et des colonnes. Les données sont stockées sous forme de collections et de documents. Chaque base de données MongoDB contient des collections, contenant elles-même des documents.

Les documents sont des paires de valeurs / clés servant d'unité de données de base.

Les collections quant à elles contiennent des ensembles de documents et de fonctions. Elles sont l'équivalent des tableaux dans les bases de données relationnelles classiques.

Dans mongoDB les données sont stocké sous format JSON, ce qui rend la base de données très flexible et évolutive.[16]



FIGURE 2.3 – Logo de mongoDB Atlas

★ Node.js :

Node.js est une plateforme logicielle libre en JavaScript, orientée vers les applications réseau événementielles qui doivent pouvoir monter en charge. Parmi les modules natifs de Node.js, on retrouve http qui permet le développement de serveur HTTP. Il est donc possible de se passer de serveurs web tels que Nginx ou Apache lors du déploiement de sites et d'applications web développés avec Node.js. Concrètement, Node.js est un environnement bas niveau permettant l'exécution de JavaScript côté serveur.[17]



FIGURE 2.4 – Logo node js

★ React.js :

React (aussi appelé React.js ou ReactJS) est une bibliothèque JavaScript développée par Facebook . est de faciliter la création d'application web monopage, via la création de composants dépendant d'un état et générant une page à chaque changement d'état.[18]



FIGURE 2.5 – Logo React.js

★ React Native :

React Native est un framework d'applications mobiles open source créé par Facebook, utilisé pour développer des applications pour Android , iOS et UWP en permettant aux développeurs d'utiliser React avec les fonctionnalités native de ces plateformes. Une application native est une application développée spécifiquement pour un type de système d'exploitation de mobile ou tablette. Développée d'une certaine manière pour IOS et d'une autre pour Android. Elle utilise des langages de développement différent (Objective-C/Swift pour une application sur iOS, Java pour Android.) Ces applications souvent plus rapides, plus performantes et proposent des interfaces et expériences utilisateur optimisées. Mais leur défaut est le temps de développement très long qu'elles nécessitent.[19]

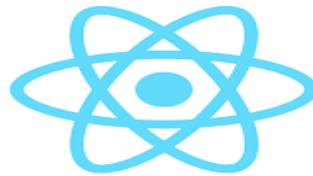


FIGURE 2.6 – Logo React Native

★ Visual code :

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS supportant un très grand nombre de langages grâce à des extensions. Il supporte l'autocomplétion, la coloration syntaxique, le débogage, et les commandes git. Le code source de Visual Studio Code provient du projet logiciel open source VSCode de Microsoft publié sous la licence MIT permissive, mais les binaires compilés sont des logiciels gratuits pour toute utilisation.[20]

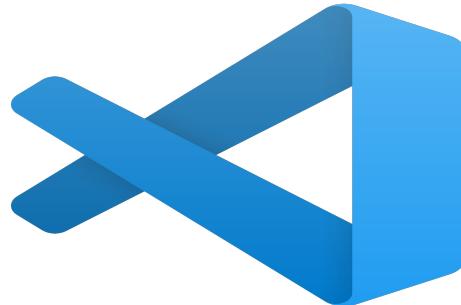


FIGURE 2.7 – Logo visual code

★ OverLeaf :

OverLeaf est un éditeur LaTeX en ligne, collaboratif en temps réel. LaTeX est un langage de mise en page permettant de produire des documents d'une grande qualité typographique et rigoureusement homogènes dans leur présentation. Ce langage est à comparer au (X)HTML, en ceci qu'il permet de baliser un texte, alors découpé proprement en sections, sous-sections, paragraphes, titres ou notes de bas de page. Le texte accompagné de ses balises est ensuite compilé, pour être mis en forme selon les spécifications de maquette choisies par l'utilisateur. TeX et LateX constituent des alternatives aux logiciels de traitement de texte wysiwyg tels que Microsoft Word...

Des alternatives essentiellement utilisées par la communauté scientifique, Tex permettant entre autres la mise en page d'équations et formules mathématiques.[21]



FIGURE 2.8 – Logo Overleaf

★ **Bootstrap4 :**

c'est une collection d'outils utiles à la création du design (graphisme, animation et interactions avec la page dans le navigateur, etc.) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. Permettant aux projets web développés de s'adapter dynamiquement au format des supports sur lesquels ils sont consultés (PC, tablette, smartphone)[22].



FIGURE 2.9 – Logo bootstrap

★ **Draw.io :**

Draw.io est une application gratuite en ligne, accessible via son navigateur (protocole https) qui permet de dessiner des diagrammes ou des organigrammes. Cet outil vous propose de concevoir toutes sortes de diagrammes, de dessins vectoriels, de les enregistrer au format XML puis de les exporter. Draw.io est un véritable couteau suisse de la frise chronologique, de la carte mentale et des diagrammes de tout genre.[23]



FIGURE 2.10 – Logo Draw.io

2.7.2.1 Protocol et format des données

★ Protocol utilisé

Dans cette application le Protocole http est utilisé comme un support pour communiquer les données entre le client Android et le serveur web. Dans ce Protocole on'a utilisé la méthode POST pour envoyer des données à partir de client Android vers le serveur et GET pour la récupération des données.

★ Format des données utilisé

Après avoir définir le Protocol utilisé, un format de données doit être présenté pour le transférées :

JSON (JavaScript Object Notation) : JSON (JavaScript Object Notation) est un format d'échange de données en texte lisible. Il est utilisé pour représenter des structures de données et des objets simples dans un code qui repose sur un navigateur Web.

JSON se compose de paires « name : object » et d'une ponctuation sous forme de parenthèses, crochets, deux-points et points-virgules. Chaque objet est défini au moyen d'un opérateur tel que « texte : » ou « image : », puis associé à une valeur de cet opérateur.

2.8 Architecture générale de l'application

Dans un monde où les architectures micro services envahissent nos systèmes d'information, surtout avec l'évolution du Cloud, on entend souvent parler des APIs REST ou RESTful. Tellement apprécié, le terme s'est transformé en symbole de maturité dans les architectures des SI.

REST est un ensemble de principes qui définissent comment les standards du Web, tels que HTTP et les URIs, sont supposés être utilisés (ce qui diffère assez de ce que beaucoup de gens font réellement). La promesse est que si vous adhérez aux principes REST en concevant votre application, vous obtiendrez un système capable de tirer parti de l'architecture du Web. En résumé, les cinq principes clés sont :

- Donner un ID à toute "élément"
- Lier les éléments entre eux
- Utiliser les méthodes standard
- Avoir des représentations multiples pour les ressources
- Communiquer sans état

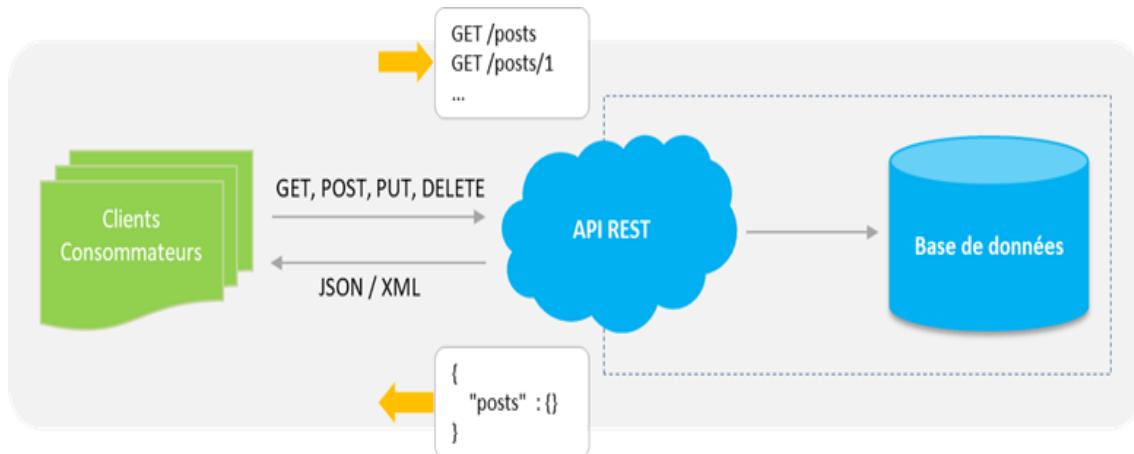


FIGURE 2.11 – Architecture REST

2.9 Architecture logicielle

Aujourd’hui nous parlons d’une séparation des applications en différentes couches, et nous parlons alors d’applications multi niveaux. Nous avons choisi l’architecture MVC pour notre projet, cette architecture peut être décrite par la figure ci-dessous.

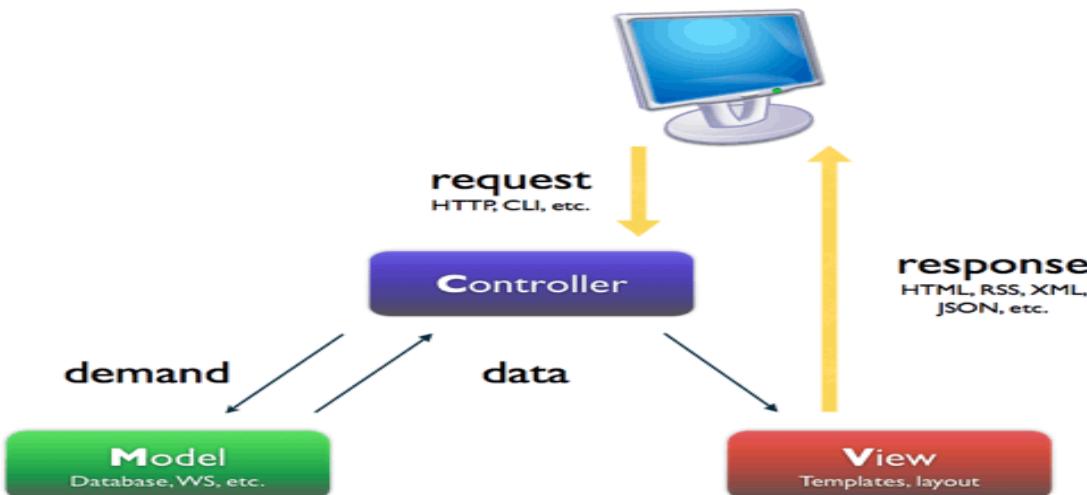


FIGURE 2.12 – Architecture logicielle

Le MVC est un motif de conception (design pattern) qui propose une solution générale au problème de la structuration d’une application.

Le MVC définit des règles qui déterminent dans quelle couche de l’architecture, et dans quelle classe (orientée-objet) de cette couche, doit être intégrée une fonctionnalité spécifique. Une application conforme à ces règles est plus facile à comprendre, à gérer et à modifier. Ces règles sont issues d’un processus d’expérimentation et de mise au point de bonnes pratiques qui a aboutit à

une architecture standard.

- Un modèle contient les données de l'application et la logique métier. Il définit aussi l'interaction avec la base de données et le traitement de ces données.
- Une vue contient tout ce qui est visible à l'écran et qui propose une interaction avec l'utilisateur : Cela correspond aux fichiers layout en XML
- Un contrôleur contient la logique concernant les actions effectuées par l'utilisateur : il s'agit de l'ensemble des activités.

2.10 Diagramme de déploiement :

Avant de se lancer dans la conception et le développement de tout système informatisé, il est important de préparer l'architecture de ce dernier. Pour cela nous modélisons notre architecture par un diagramme de déploiement. Ce diagramme présente la vue de déploiement d'une architecture logicielle. En effet il permet de décrire la disposition physique des ressources matérielles qui composent notre système et montre la répartition des composants sur ces matériels. Le diagramme de déploiement modélise les ressources matérielles par des nœuds, et précise les connexions entre eux .

La figure ci-dessous présente le diagramme de déploiement de nos applications selon l'architecture physique adoptée.

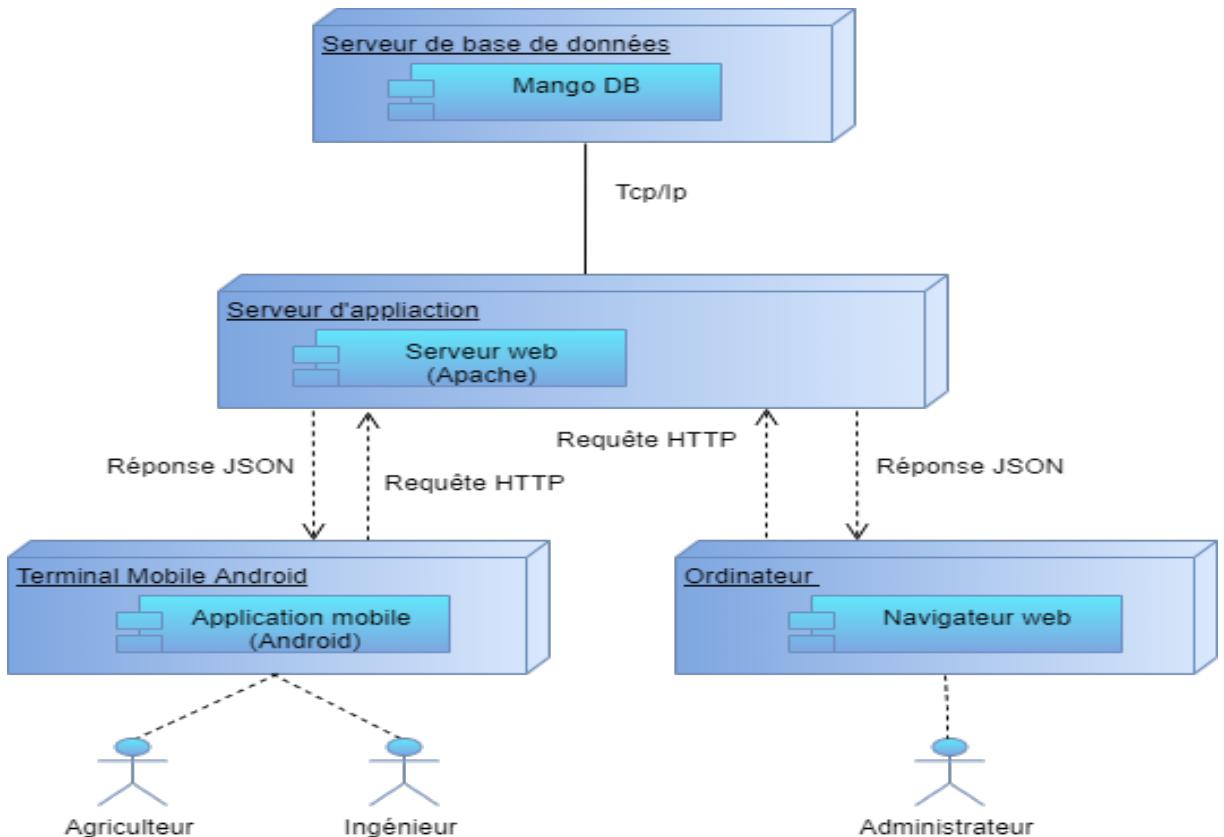


FIGURE 2.13 – Diagramme de déploiement

2.11 Diagramme de Package :

Un diagramme de packages est un diagramme fait partie des diagrammes UML qui fournit une représentation graphique de haut niveau de l'organisation d'une application, il permet d'identifier les liens de généralisation et de dépendance entre les packages. En effet, cette figure présente les différents paquetages qui composent notre système et les relations qui lient ces différents paquetages.

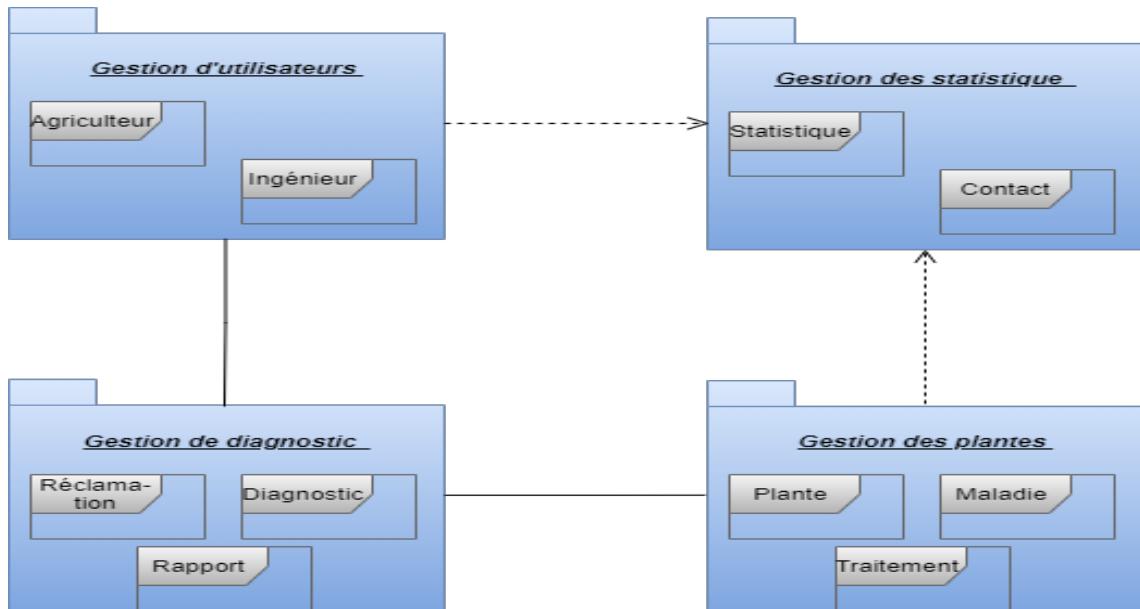


FIGURE 2.14 – Diagramme de Package

2.12 Conclusion :

Durant ce chapitre, nous avons établi le planning de notre travail. Nous avons présenté le Backlog du produit ainsi que l'architecture de la solution à réaliser et la planification des sprints. En partant du backlog du produit défini, nous précisons la liste des tâches à exécuter dans chaque sprint, en passant par la spécification fonctionnelle, l'analyse et la réalisation.

Nous allons enchaîner maintenant avec notre premier sprint, dans le chapitre qui suit.

Chapitre 3

Mise en œuvre du sprint 1

3.1 Introduction

Ce chapitre présente les détails du premier Sprint. Nous commençons par la planification du Sprint en précisant les cas d'utilisations associés. Puis, Nous passons par une phase d'analyse, ensuite nous présenterons quelques interfaces réalisées au cours de ce sprint. Nous détaillons chaque cas en présentant les diagrammes d'analyse, séquences, activités et la réalisation.

3.2 Planification de sprint 1

Cette étape est réalisée pour présenter le sprint 1 ainsi les cas d'utilisations qu'elle compose. Ce schéma montre la planification de sprint 1 :



FIGURE 3.1 – Planification de sprints1

3.3 Backlog de sprint 1

Tous les sprints d'une release ont une durée constante et ne se chevauchent. Le sprint est le cœur de Scrum. Il s'agit d'un bloc de temps durant lequel un incrément du produit sera réalisé 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[24.] Le tableau 1 décrit les histoires de notre backlog du sprint1.

ID	Histoire	Date début	Date fin	Responsable
1	S'inscrire	15/03/2021	20/03/2021	Ouertani Asma
2	S'authentifier	22/03/2021	28/03/2021	
3	Gérer profil	01/10/2021	10/04/2021	
4	Gérer Agriculteur	15/04/2021	25/05/2021	
4	Gérer Ingénieur	27/04/2021	10/06/2021	

TABLE 3.1 – Backlog de sprint1

3.4 Tableau des tâches

Le tableau de répartition des tâches permet de visualiser les attributions de chaque poste au sein d'un groupe, pour une période donnée (une semaine, par exemple). Il fait généralement apparaître le temps attribué à chaque poste pour chaque activité. Le tableau de répartition des tâches permet à chacun de connaître son rôle dans le groupe et, si nécessaire, d'analyser une situation pour essayer de l'améliorer.

Notre tableau de tâche de sprint 1 est réalisé à T= 15/03/2021.

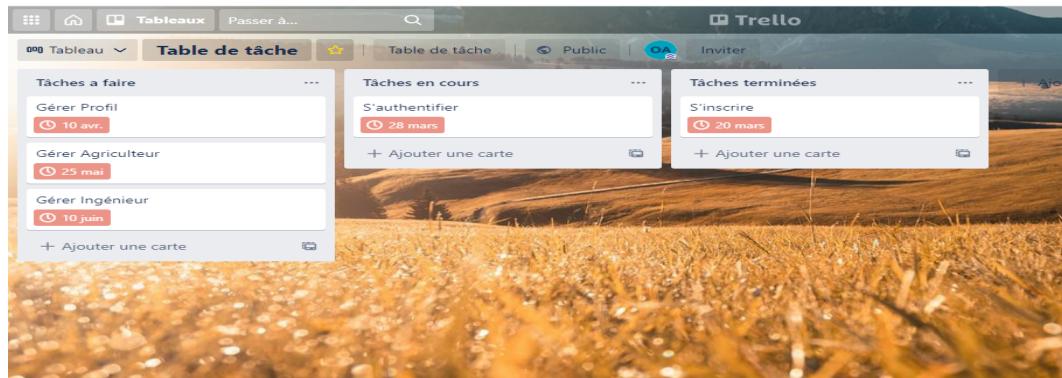


FIGURE 3.2 – Tableau de tâche de premier sprint

3.5 Phase d'analyse

Dans cette partie nous présentons la phase d'analyse qui répond à la question « que fait le système ». La réponse de cette question se traduit par la présentation du diagramme des cas d'utilisation puis la description textuelle de chacun d'entre eux.

3.5.1 Diagramme de cas d'utilisation global de sprint 1

Les acteurs qui interagissent dans le sprint 1 sont :

Agriculteur : c'est acteur principale de notre application, en premier lieu peut faire une inscription pour être capable de gérer son profile après l'authentification

Administrateur : c'est l'acteur du back office qui capable de contrôler et configurer l'accès des utilisateurs de l'application :

- Gérer agriculteurs : consulter ou supprimer.
- Gérer Ingénieur : ajouter ,consulter ,modifier et supprimer.

Ingénieur : c'est acteur du front office ,en premier lieu doit être authentifié pour être capable de créer les rapports après l'authentification.

Cette figure ci-dessous présente le diagramme de cas d'utilisation global de sprint1 :

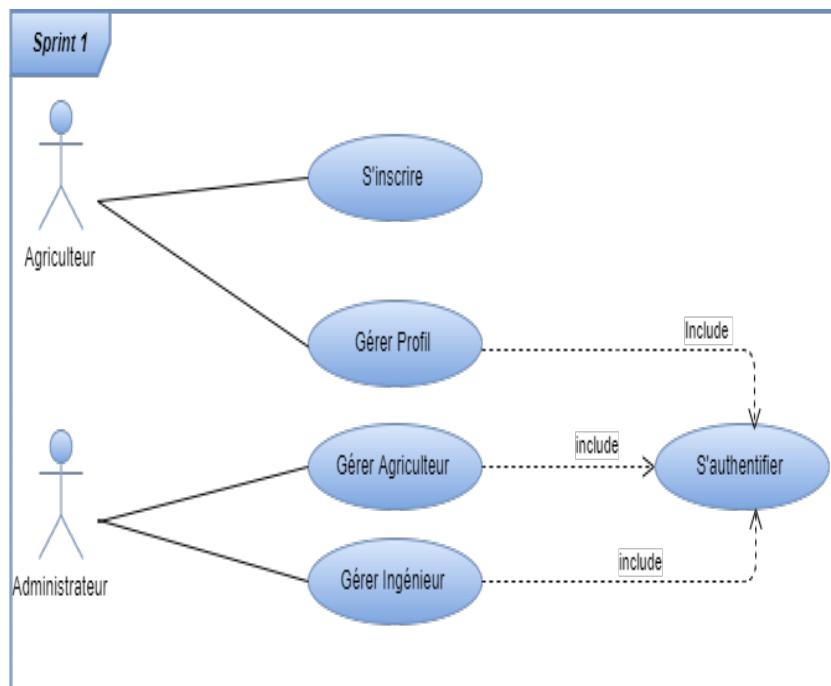


FIGURE 3.3 – Diagramme de cas d'utilisation de sprint1

3.5.1.1 Raffinement et description des cas d'utilisations

Dans cette partie on va décrire les scénarios de chaque cas d'utilisation de se sprint

★ Raffinement de cas d'utilisation « S'inscrire»

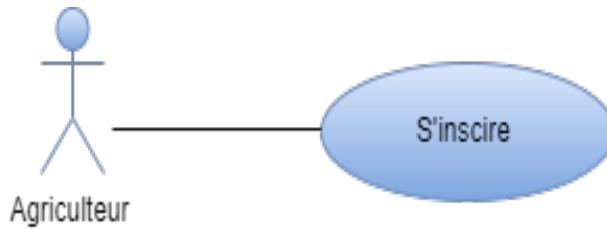


FIGURE 3.4 – Raffinement du cas d'utilisation « S'inscrire »

★ Description textuelle de cas d'utilisation " s'inscrire pour l'acteur Agriculteur"

Titre	S'inscrire
Acteur	Agriculteur
Objectif	L'agriculteur remplit le formulaire d'inscription pour écrire un compte effectuée son inscription.
Pré-condition	L'agriculteur doit accéder à la page d'accueil.
Post-condition	L'agriculteur doit accéder à son espace personne.
Scénario Principal	<p>1- L'agriculteur demande la page d'inscription, 2- Le système lui affiche la page demandée,</p> <p>3- L'agriculteur remplit le formulaire d'inscription,</p> <p>4- Le système vérifie les données saisies,</p> <p>5- Le système ajoute l'agriculteur dans la base de données et affiche le message suivant : « Vous êtes inscrit avec succès ».</p> <p>6- Puis l'administrateur revoie un mot passe sur l'email de l'agriculteur,</p> <p>7- L'agriculteur prend sa mot passe et peut se connecter à l'application .</p>
Scénario alternatif et d'exception :	<p>1- L'agriculteur quitte la page d'inscription :</p> <ul style="list-style-type: none"> • Le cas d'utilisation se termine avec échec. <p>2- Les données saisies sont incorrectes ou manquantes :</p> <ul style="list-style-type: none"> • Le système affiche un message d'erreur et signale au agriculteur de recommencer la saisie. <p>3- Identifiants déjà existant dans la base de données :</p> <ul style="list-style-type: none"> • Le système affiche un message d'erreur suivant : « Vous devez saisir un Identifiants valides » et signale à l'agriculteur de recommencer la saisie

TABLE 3.2 – Description textuelle de cas d'utilisation «s'inscrire pour l'acteur Agriculteur ».

★ Raffinement de cas d'utilisation « S'authentifier»

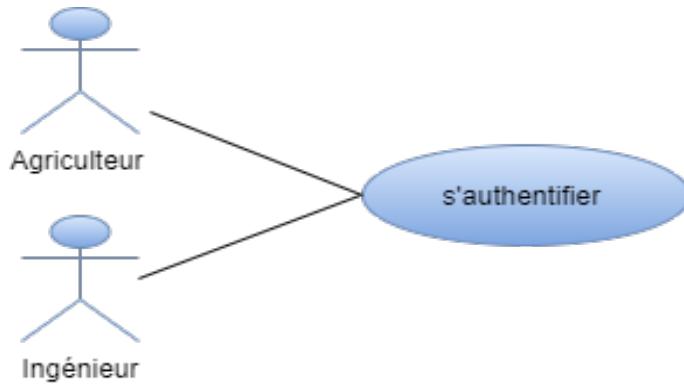


FIGURE 3.5 – Raffinement de cas d'utilisation « S'authentifier »

★ Description textuelle de cas d'utilisation " S'authentifier "

Titre	S'authentifier
Acteur	Administrateur, Agriculteur, Ingénieur
Objectif	Lors de l'accès à l'application, chaque acteur doit s'authentifier pour accéder aux différentes fonctionnalités de l'application.
Pré-condition	<ul style="list-style-type: none"> • L'administrateur doit être inscrit. • Identifiant correct • Mot de passe correcte.
Post-condition	Acteur authentifié.
Scénario Principal	1- L'administrateur saisit son login et son mot de passe, 2- L'administrateur clique sur le bouton « se connecter », 3- Le système vérifie la combinaison login et mot de passe, 4- Le système affiche l'interface propre à l'administrateur.
Scénario alternatif et d'exception :	1- L'administrateur quitte (annule) la page d'authentification : <ul style="list-style-type: none"> • Le cas d'utilisation se termine avec échec. 2- Le système affiche un message d'erreur informant l'administrateur que son login ou mot de passe sont incorrects.

TABLE 3.3 – Description textuelle de cas d'utilisation " S'authentifier ".

★ Raffinement de cas d'utilisation « Gérer profil »

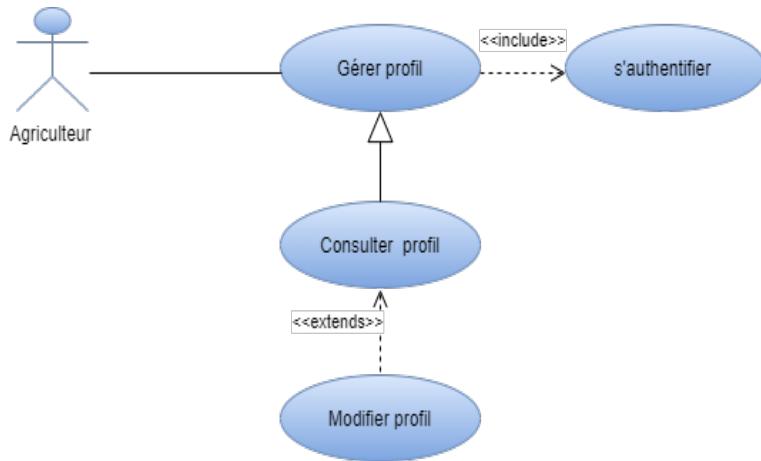


FIGURE 3.6 – Raffinement de cas d'utilisation « Gérer profil»

★ Description textuelle de cas d'utilisation " Gérer profil"

Titre	Gérer profil
Acteur	Agriculteur
Objectif	Lors de l'accès au l'application, l'agriculteur doit s'authentifier pour accéder à la page de modifier profil.
Pré-condition	<ul style="list-style-type: none"> • L'agriculteur doit être authentifié. • L'agriculteur doit accéder à l'interface gérer profil.
Post-condition	L'agriculteur mis à jour son profil (Consulter, modifier) avec succès .
Scénario Principal	<p>Ce cas d'utilisation commence lorsque l'agriculteur demande au système de faire la modification de son profil :</p> <ul style="list-style-type: none"> ■ Enchainement(a) :Consulter Profil 1- L'agriculteur demande la consultation de profil, ■ Enchainement(a.1) : Modifier Profil 1- L'agriculteur clique sur le bouton « modifier profil » 2- Le système affiche l'interface demandée avec les informations de profil, 3- L'agriculteur saisit les nouvelles données, 4- Le système vérifie les données saisies, 5- Le système met à jour les informations du profil dans la base de données et affiche le message suivant : « profil modifié avec succès ».
Scénario alternatif et d'exception :	<p>1- L'agriculteur annule la mise à jour de son profil :</p> <ul style="list-style-type: none"> • Le cas d'utilisation se termine avec échec. <p>2- Les données saisies sont incorrectes :</p> <ul style="list-style-type: none"> • Le système affiche un message d'erreur « vérifier vos données» • Le cas d'utilisation se termine avec échec.

TABLE 3.4 – Description textuelle de cas d'utilisation « Modifier profil ».

★ Raffinement de cas d'utilisation « Gérer Agriculteur »

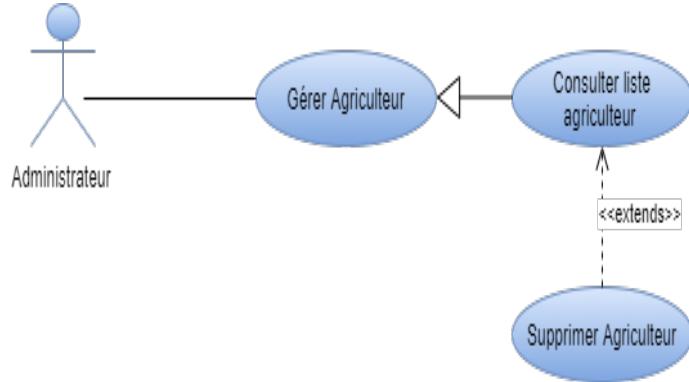


FIGURE 3.7 – Raffinement de cas d'utilisation « Gérer agriculteur»

★ Description textuelle de cas d'utilisation " Gérer les agriculteurs "

Titre	Gérer Agriculteur
Acteur	Administrateur
Objectif	de l'accès à l'application, l'administrateur doit consulter la liste des agriculteurs, et supprimer un Agriculteur.
Pré-condition	<ul style="list-style-type: none"> L'administrateur doit être authentifié. L'administrateur doit accéder à l'interface gérer agriculteur.
Post-condition	Mise à jour (consulter, supprimer) des agriculteurs avec succès
Scénario Principal	<p>Ce cas d'utilisation commence lorsque l'administrateur demande au système de faire la gestion de ses agriculteurs :</p> <ul style="list-style-type: none"> ■ Enchainement(a) : consulter la liste des agriculteurs <p>1- L'administrateur demande la consultation des agriculteurs 2- Le système affiche la liste des agriculteurs sélectionné de la base de données</p> <ul style="list-style-type: none"> ■ Enchainement(a.1) : supprimer agriculteur <p>1-L'administrateur choisit un agriculteur de la liste affichée (liste des agriculteurs déjà inscrit dans l'application) et clique sur le bouton « supprimer », 2- Le système affiche un message pour confirmer la suppression, 3- L'administrateur choisit confirmer, 4- Le système supprime l'agriculteur de la base de données.</p>
Scénario alternatif et d'exception :	<p>1- L'administrateur annule la gestion des agriculteurs :</p> <ul style="list-style-type: none"> • Le cas d'utilisation se termine avec échec. <p>2. mauvaise connexion avec la base de données :</p> <ul style="list-style-type: none"> • Le cas d'utilisation se termine avec échec..

TABLE 3.5 – Description textuelle de cas d'utilisation " Gérer les agriculteurs "

★ Raffinement de cas d'utilisation « Gérer Ingénieur »

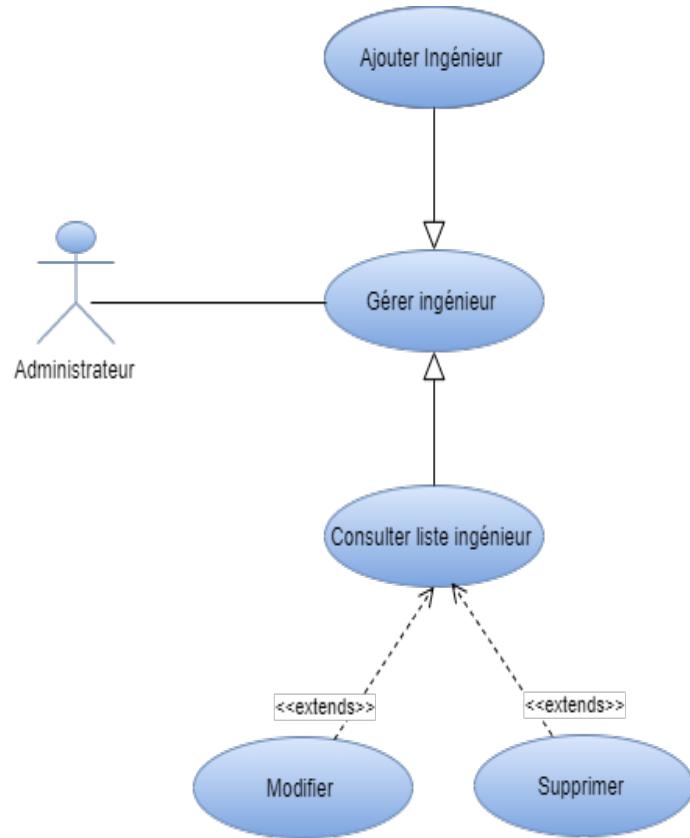


FIGURE 3.8 – Raffinement de cas d'utilisation « Gérer Ingénieur»

★ Description textuelle de cas d'utilisation " Gérer les ingénieurs "

Titre	Gérer Ingénieur
Acteur	Administrateur
Objectif	Lors de l'accès au l'application, l'administrateur doit modifier, consulter, ajouter et supprimer un ingénieur.
Pré-condition	<ul style="list-style-type: none"> • L'administrateur doit être authentifié. • L'administrateur doit accéder à l'interface gérer Ingénieur.
Post-condition	Mise à jour (ajouter, consulter, modifier, supprimer) des ingénieurs avec succès.
Scénario Principal	<p>Se cas d'utilisation commence lorsque l'administrateur demande au système de faire la gestion de ses ingénieurs :</p> <ul style="list-style-type: none"> ■ Enchainement(a) :ajouter ingénieur <ol style="list-style-type: none"> 1- L'administrateur demande la page d'ajouter ingénieur, 2- Le système affiche le formulaire à remplir 3- L'administrateur remplit le formulaire et clique sur « ajouter » 4- Puis L'administrateur revoie un mot passe sur l'email de l'ingénieur 5- Le système vérifie les données et affiche un message (" ingénieur ajouté avec succès "), 5- L'administrateur revoie un mot passe sur l'email de l'ingénieur Pour connecté sur son application. ■ Enchainement(b) : consulter ingénieur <ol style="list-style-type: none"> 1- L'administrateur demande la consultation des ingénieurs, 2- Le système affiche la liste des ingénieurs à sélectionner de la base de données. ■ Enchainement(b.1) modifier ingénieur <ol style="list-style-type: none"> 1- L'administrateur choisie un ingénieur, 2- L'administrateur demande la page « modifier ingénieur » , 3- Le système affiche un formulaire pour modifier les données de l'ingénieur, 4- L'administrateur remplit le formulaire et clique sur modifier, 5- Le système vérifier les données et affiche un message (" ingénieur modifier avec succès "). ■ Enchainement(b.2) : supprimer ingénieur <ol style="list-style-type: none"> 1- L'administrateur choisit un ingénieur de la liste afin de l'afficher (liste des ingénieurs déjà inscrit dans l'application) et clique sur le bouton « supprimer ingénieur », 2- Le système affiche un message pour confirmer la suppression, 3- L'administrateur choisi le bouton « confirmer », 4- Le système supprime l'ingénieur de la base de données.
Scénario alternatif et d'exception :	<ol style="list-style-type: none"> 1- L'administrateur annule le contrôle : <ul style="list-style-type: none"> • Le cas d'utilisation se termine avec échec. 2- Les données saisies sont incorrectes : <ul style="list-style-type: none"> • Le système affiche un message d'erreur «vérifier vos données» • Le cas d'utilisation se termine avec échec.

TABLE 3.6 – Description textuelle de cas d'utilisation " Gérer les ingénieurs "

3.6 Phase de conception

3.6.1 Conception statique de sprint

a) Diagramme de classe de sprint 1

Pour ce faire, nous allons présenter la structure statique de l'application via un diagramme de classe. Pour clarifier le diagramme de classe et le rendre plus lisible, nous avons procédé à

schématiser les interfaces, les classes métiers et les entités d'une manière simplifiée en prenant quelques exemples.

Le diagramme de la figure 3.9 représente un modèle de classes que nous avons conçu lors du premier Sprint :

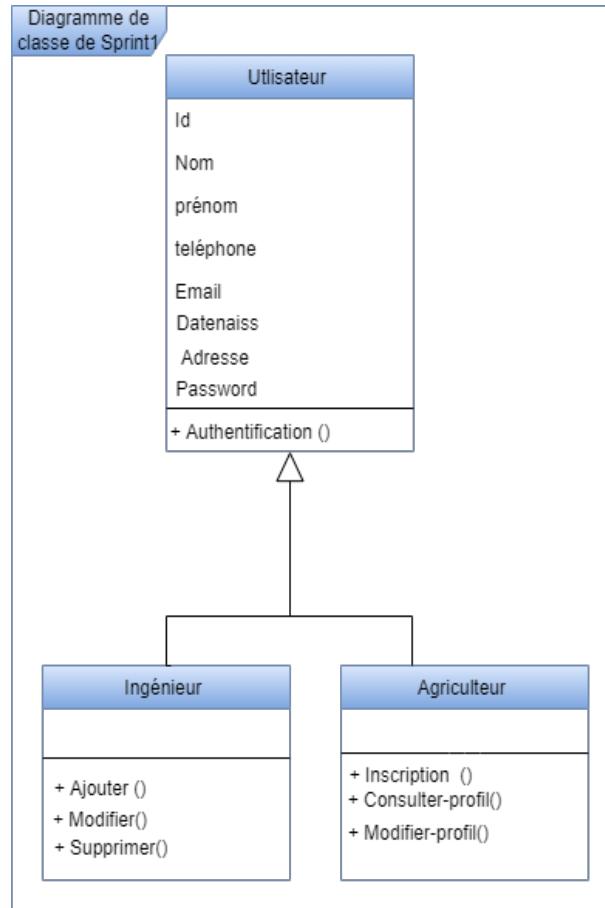


FIGURE 3.9 – Diagramme de classe de sprint1»

b) **Dictionnaire de données :**

Le dictionnaire de base de données est une définition de la caractéristique des attributs de chaque classe de sprint.

Dictionnaire de données de la table Utilisateur

Utilisateur				
Attribut	Type	Taille	Obligatoire	Contrainte
Id	Int	50	Oui	Clé primaire
Nom	Varchar	20	Oui	
Prénom	Varchar	20	Oui	
Téléphone	Int	8	Oui	
Email	Varchar	30	Oui	

TABLE 3.7 – Dictionnaire de données de la table Utilisateur

Dictionnaire de données de la table Agriculteur

Agriculteur				
Attribut	Type	Taille	Obligatoire	Contrainte
Id	Int	50	Oui	Clé primaire
Nom	Varchar	20	Oui	
Prénom	Varchar	20	Oui	
Téléphone	Int	8	Oui	
Email	Varchar	30	Oui	
Password	varchar	50	Oui	
Datenaiss	Varchar	50	Oui	
Adresse	Varchar	50	Oui	

TABLE 3.8 – Dictionnaire de données de la table Agriculteur

Ingénieur				
Attribut	Type	Taille	Obligatoire	Contrainte
Id	Int	50	Oui	Clé primaire
Nom	Varchar	20	Oui	
Prénom	Varchar	20	Oui	
Téléphone	Int	8	Oui	
Email	Varchar	30	Oui	
Password	varchar	50	Oui	
Datenaiss	Varchar	50	Oui	
Adresse	Varchar	50	Oui	

TABLE 3.9 – Dictionnaire de données de la table Ingénieur

Dictionnaire de données de la table Ingénieur

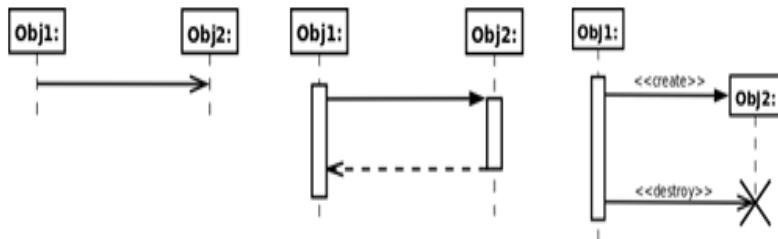
3.7 Conception dynamique du premier sprint

3.7.1 Les diagramme de séquence

Un diagramme de séquence est un diagramme d’interaction dont le but est de décrire comment les objets collaborent au cours du temps et quelles responsabilités ils assument. Il décrit un scénario d’un cas d’utilisation avec les intéressants ajouts au diagramme de séquences apportés par UML 2, en particulier les cadres d’interactions (avec les opérateurs Loop, opt et Alt par exemple), ainsi que la possibilité de référencer une interaction décrite par ailleurs, le diagramme de séquence système

nous semble constituer une excellente solution[[25]].

Les différents messages représentés par un diagramme de séquence sont :



Message asynchrone :

Il n'attend pas de réponse et ne bloque pas l'émetteur qui ne sait pas si le message arrivera à destination.

Message synchrone :

L'émetteur reste alors bloqué le temps que dure l'invocation de l'opération.

Messages de création et destruction d'instance :

La création d'un objet est matérialisée par une flèche qui pointe sur le sommet d'une ligne de vie et la destruction d'un objet est matérialisée par une croix qui marque la fin de la ligne de vie de l'objet

Nous présentons dans ce qui suit des diagrammes de séquence détaillés des histoires du premier sprint.

★ Diagramme de séquence de cas d'utilisation « s'inscrire » pour l'acteur Agriculteur

Chaque Agriculteur doit s'inscrire, où il doit remplir le formulaire d'inscription pour accéder à l'application. Si les données saisies sont invalides, le système affiche un message d'erreur. Sinon, il se connecte et affiche la page d'accueil correspondante. La Figure ci-dessous présente le diagramme de séquence détaillé du cas d'utilisation «S'inscrire »

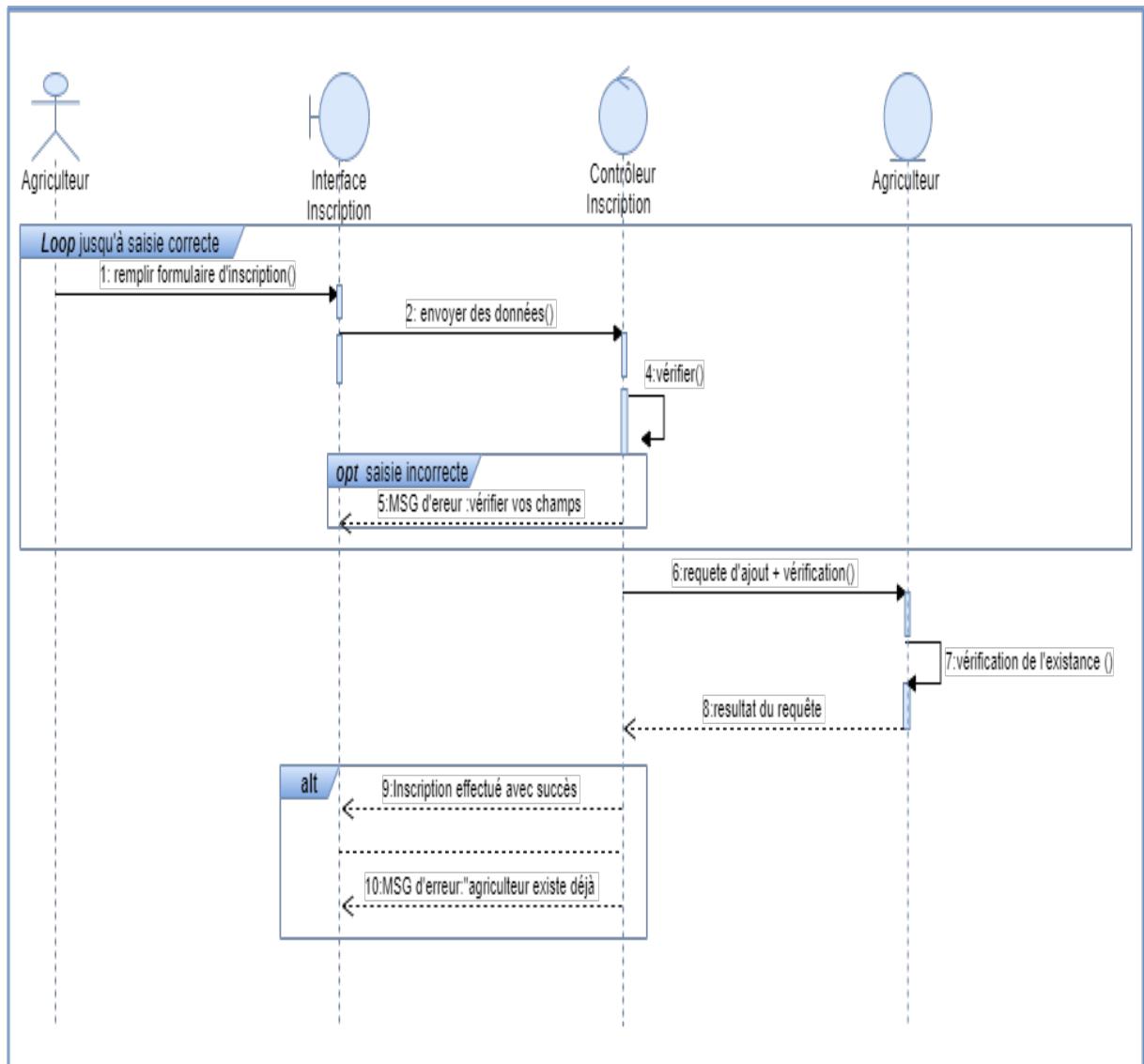


FIGURE 3.10 – Diagramme de séquence détaillées du cas « s'inscrire »

★ Diagramme de séquence de cas d'utilisation « s'authentifier »

Chaque utilisateur doit s'authentifier (Agriculteur, Ingénieur) où il doit saisir son email et son mot de passe pour accéder à l'application. Si les données saisies sont invalides, le système affiche un message d'erreur. Sinon, il se connecte et affiche la page d'accueil correspondante. La Figure ci-dessous présente le diagramme de séquence détaillé du cas d'utilisation "S'authentifier".

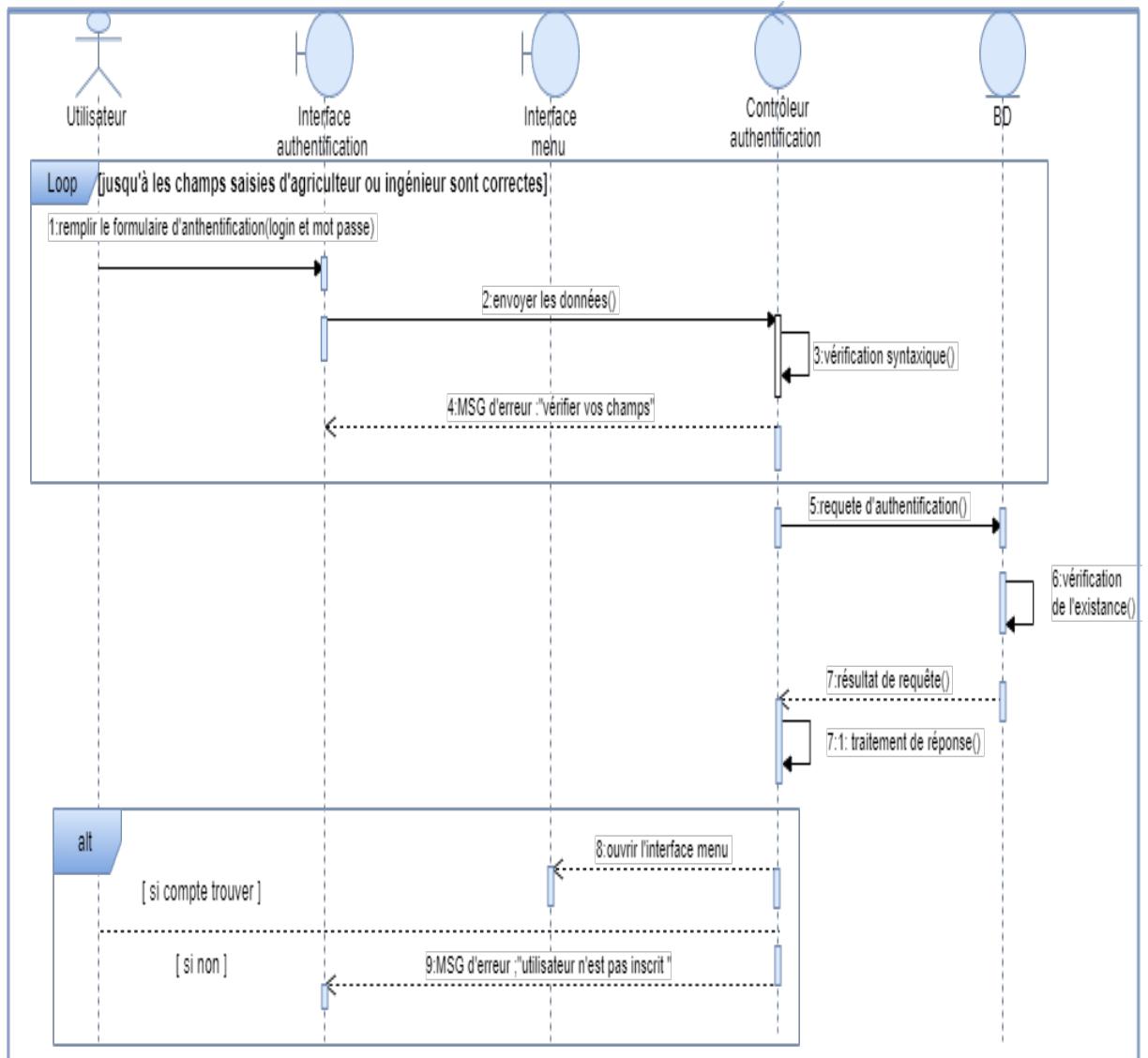


FIGURE 3.11 – Diagramme de séquence détaillé du cas « s'authentifier »

★ Diagramme de séquence de cas d'utilisation « Ajouter Ingénieur »

La figure 3.12 illustre le diagramme de séquence du cas « Ajouter Ingénieur ». L'administrateur, qui doit être d'abord authentifié, remplit le formulaire d'ajout. Lorsque les données saisies sont valides, le système procède à l'enregistrement, puis fait une redirection vers la page où est affichée la liste des ingénieurs.

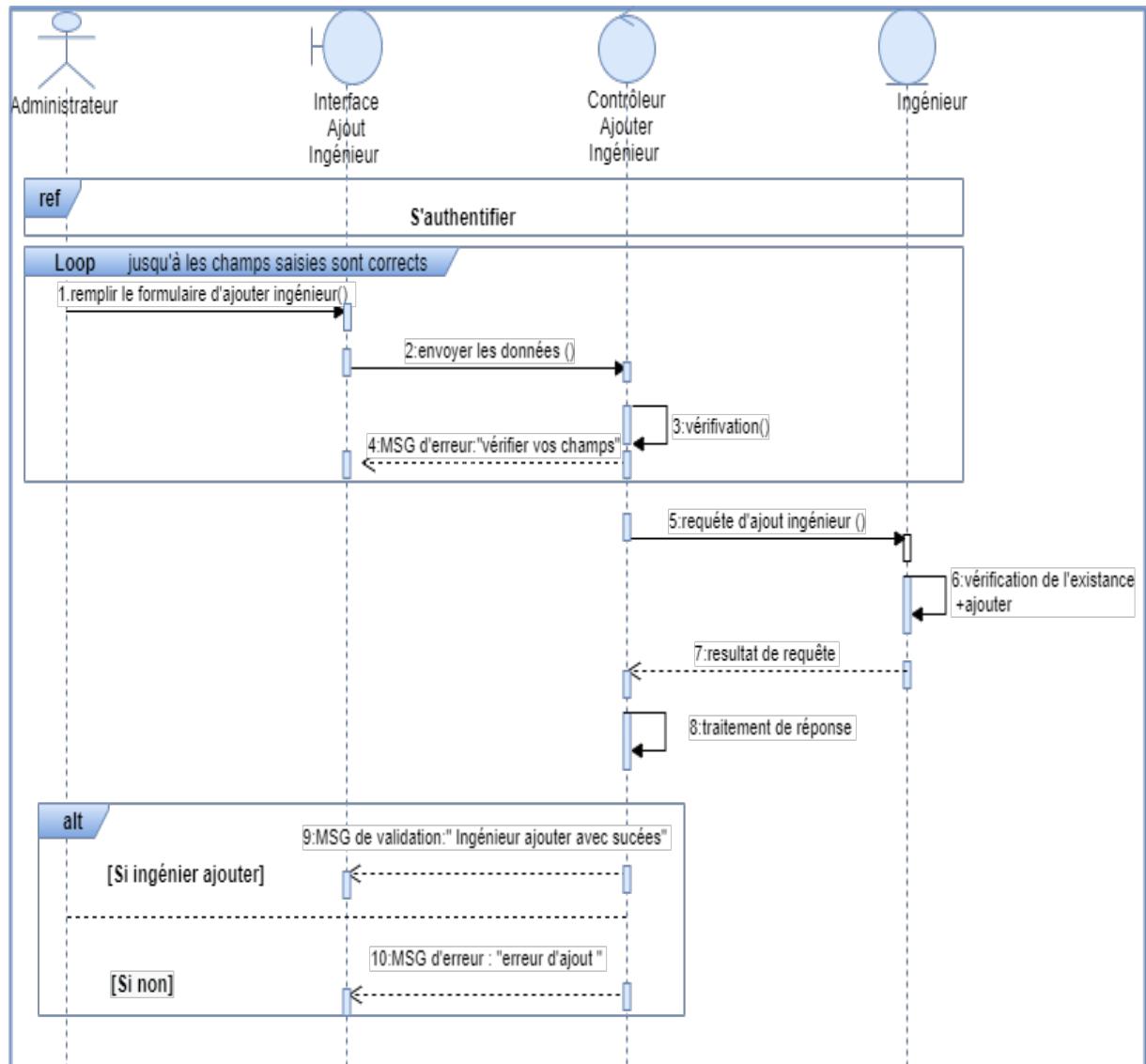


FIGURE 3.12 – Diagramme de séquence détaillé du cas « Ajouter Ingénieur »

★ Diagramme de séquence de cas d'utilisation « Modifier ingénieur »

La figure 3.13 illustre le diagramme de séquence du cas « Modifier Ingénieur ». Après authentication, l'administrateur demande la liste des Ingénieur. Une fois la liste est affichée, il peut modifier un ingénieur par un simple clic sur l'icône « modifier ». La suppression génère une pop-up de confirmation.

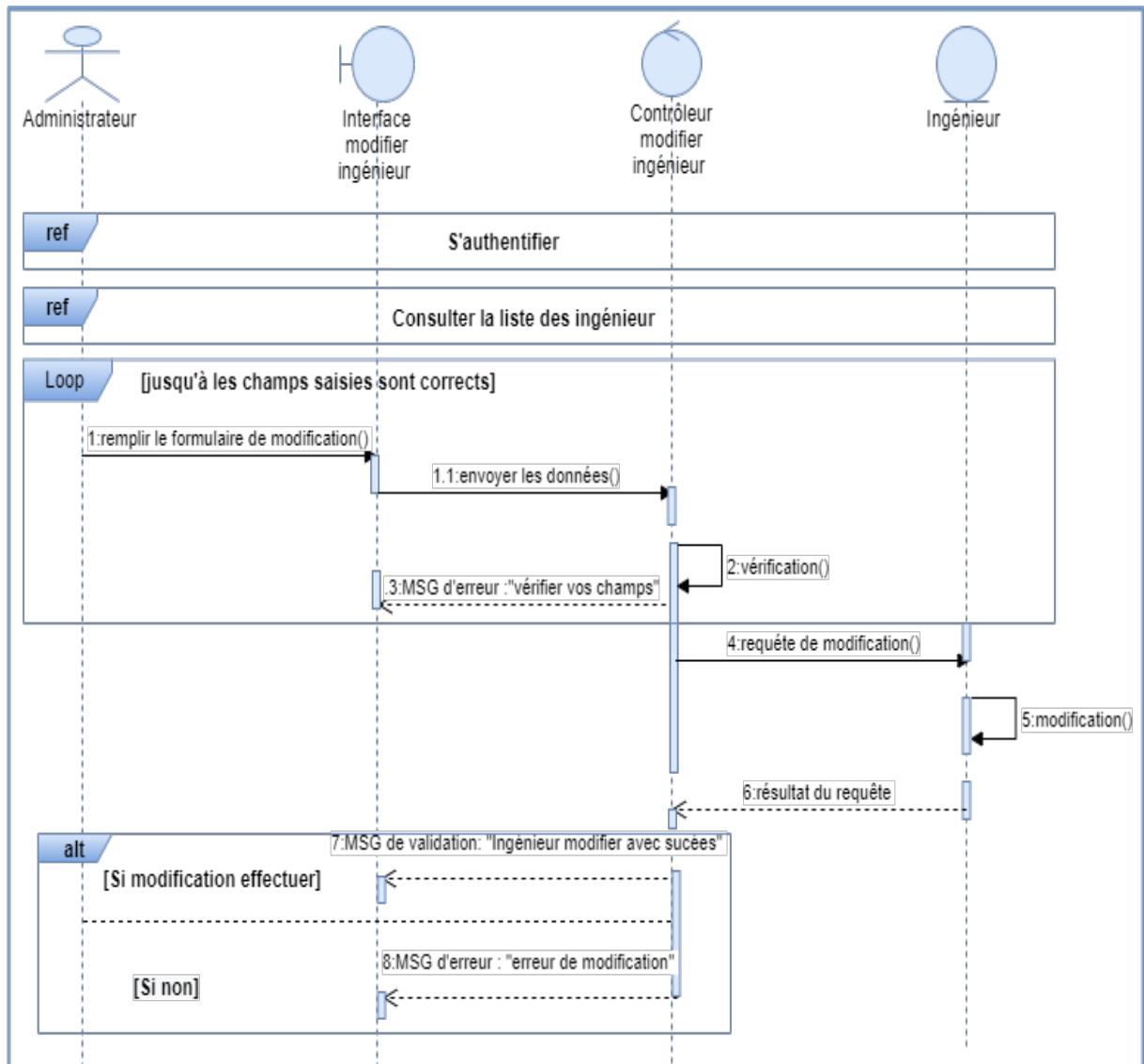


FIGURE 3.13 – Diagramme de séquence détaillé du cas « Modifier Ingénieur »

★ Diagramme de séquence de cas d'utilisation « Supprimer Ingénieur »

La figure 3.14 illustre le diagramme de séquence du sous cas « Supprimer Ingénieur ». Après authentification, l'administrateur demande la liste des Ingénieur. Une fois la liste est affichée, il peut supprimer un ingénieur par un simple clic sur l'icône « Supprimer ». La suppression génère une pop-up de confirmation et pour continuer l'administrateur doit cliquer sur bouton « Supprimer ».

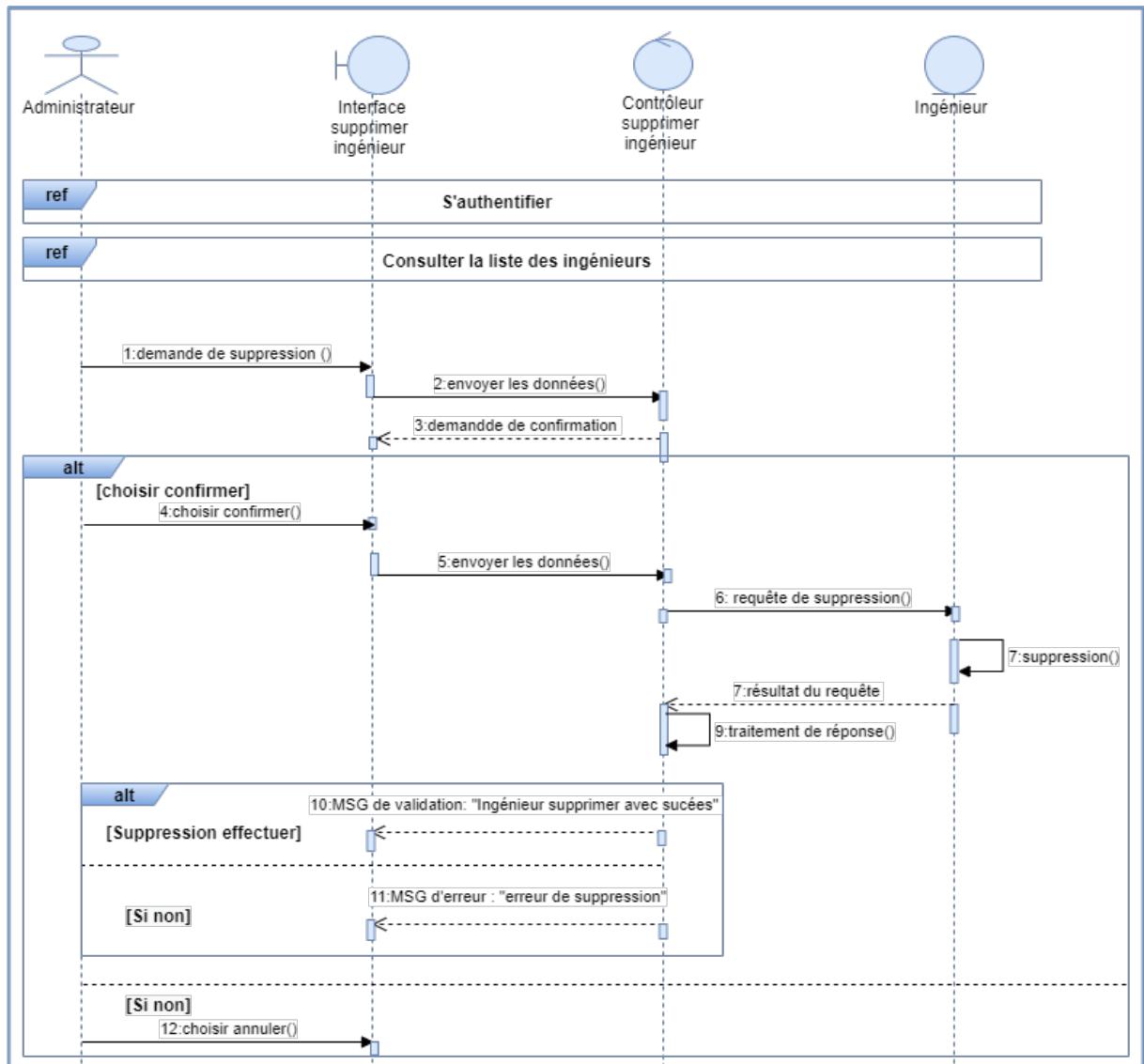


FIGURE 3.14 – Diagramme de séquence détaillé du cas « Supprimer Ingénieur »

3.8 Diagrammes d’activités :

Diagrammes d’activités Les diagrammes d’activités permettent de déterminer des traitements à priori séquentiels. Ils offrent un pouvoir d’expression très proche des langages de programmation objet tels que la spécification des actions de base (déclaration de variables, affectation etc.), les structures de contrôle (conditionnelles, boucles), ainsi que les instructions particulières à la programmation orientée objet (appels d’opérations, exceptions etc.). Ils sont donc bien adaptés à la spécification détaillée des traitements en phase de réalisation. On peut également utiliser de façon plus informelle pour décrire des enchaînements d’actions de haut niveau[26].

★ Diagramme d’activité du cas d’utilisation "S’inscrire "

La figure 3.15 illustre le diagramme d’activité du sous cas d’utilisation « S’inscrire ».

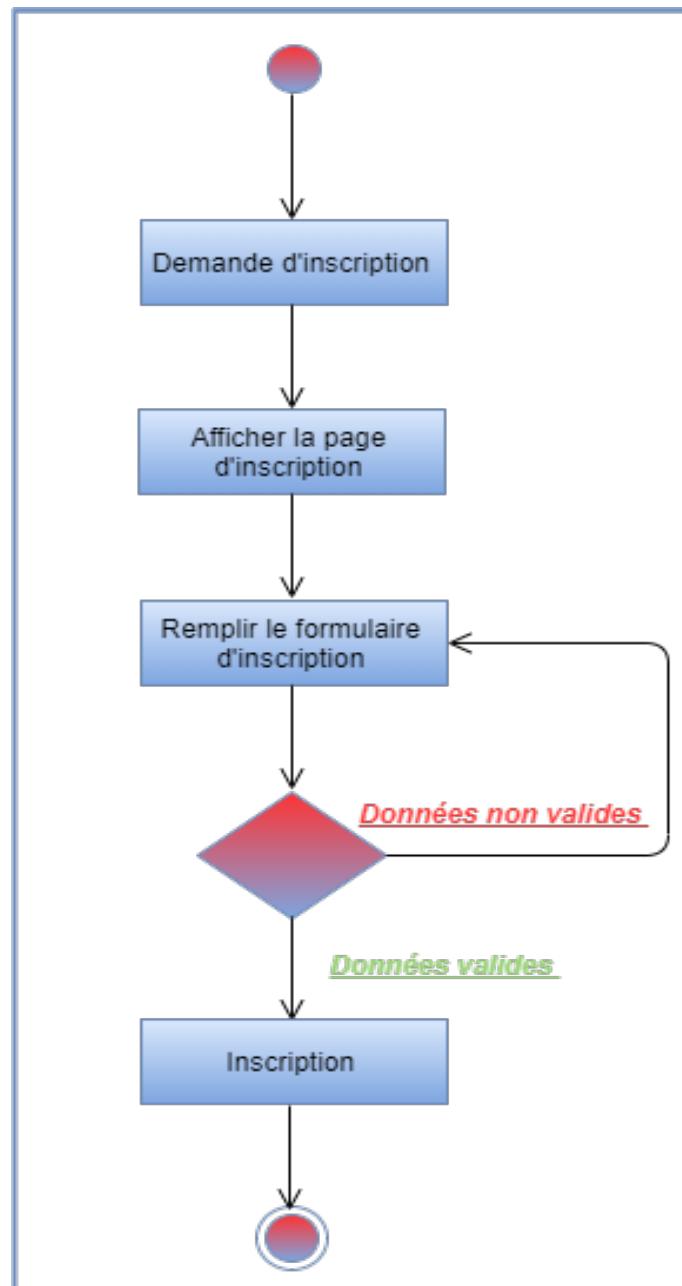


FIGURE 3.15 – Diagramme d'activité du cas d'utilisation « S'inscrire »

★ Diagramme d'activité du cas d'utilisation "S'authentifier"

La figure 3.16 illustre le diagramme d'activité du sous cas d'utilisation « S'authentifier ».

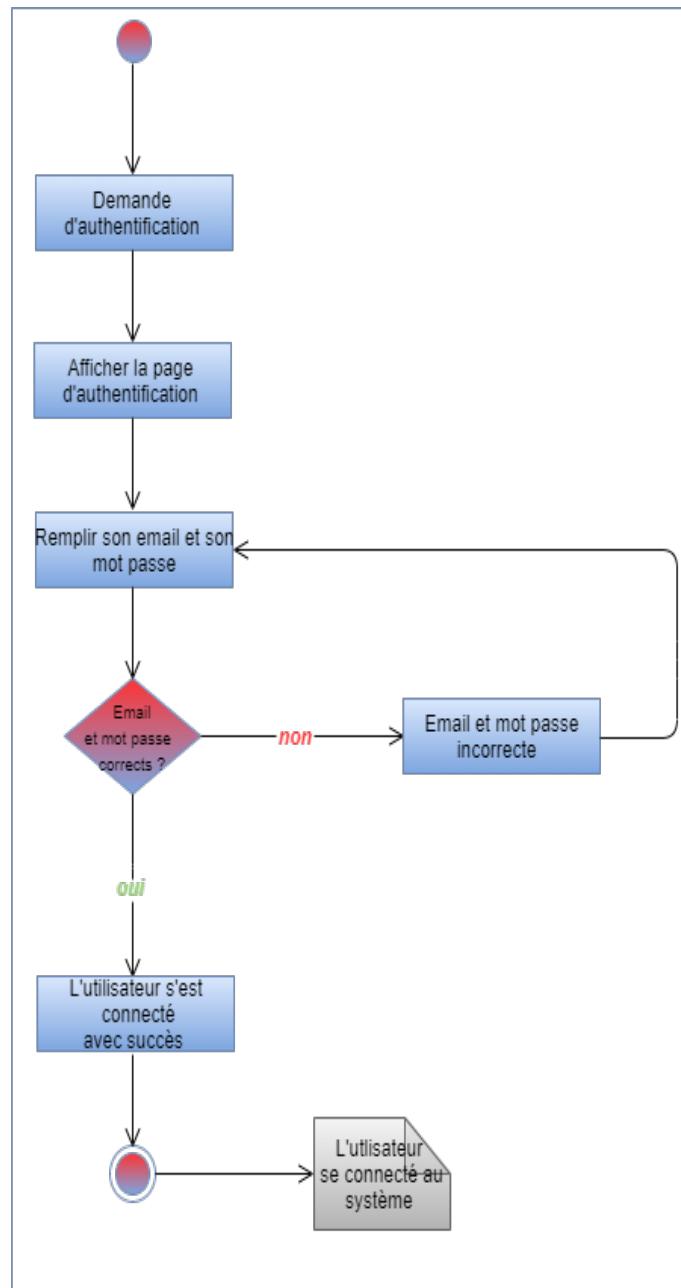


FIGURE 3.16 – Diagramme d'activité du cas d'utilisation « S'authentifier »

★ Diagramme d'activité du cas d'utilisation "Modifier Ingénieur "

La figure 3.17 illustre le diagramme d'activité du sous cas d'utilisation « Modifier Ingénieur ».

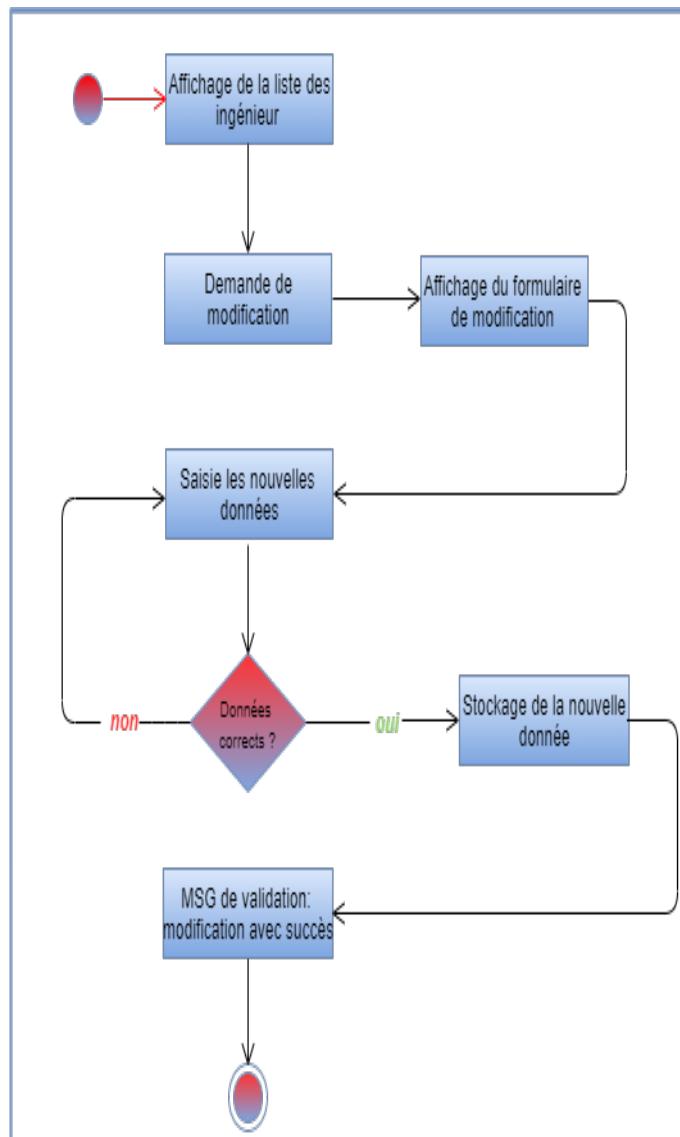


FIGURE 3.17 – Diagramme d'activité du cas d'utilisation « Modifier Ingénieur»

3.9 Réalisation et Tests

Cette partie est consacrée à l'exposition du travail achevé à travers des captures d'écrans de différentes interfaces développées et les différents tests réalisés pendant ce sprint.

3.9.1 Les Interfaces :

3.9.1.1 Les intefaces mobiles :

La figure ci-dessous présente l'interface Inscription



FIGURE 3.18 – Interface Inscription

La figure ci-dessous présente l'interface Modifier profil



FIGURE 3.19 – Interface Modifier profil

3.9.1.2 Les interfaces web

La figure ci-dessous présente l'interface d'authentication

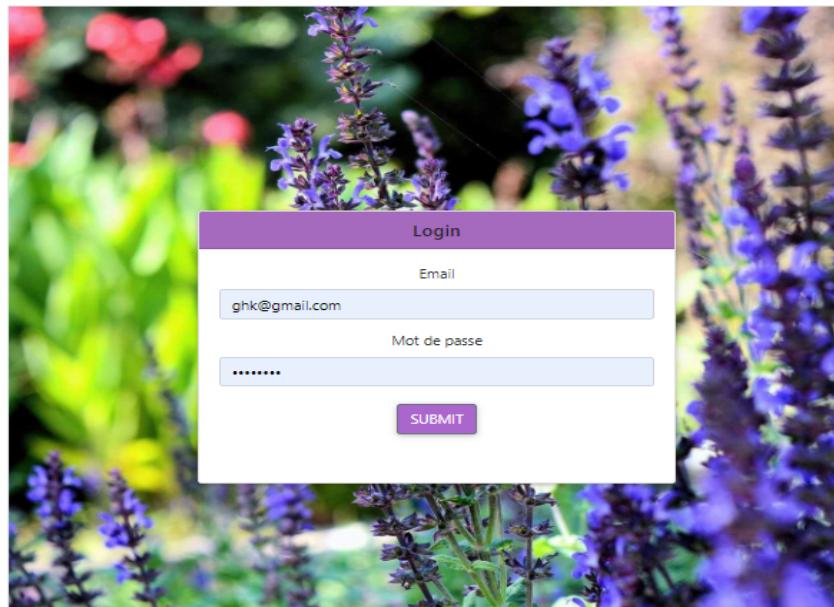


FIGURE 3.20 – Interface authetification

La figure ci-dessous présente la liste des ingénieurs.

Nom	Prenom	Email	Téléphone	Action
asma	ouertani	asmaouertani1995@gmail.com	28365744	
hana	habbachi	habbachi@gmail.com	28365744	

FIGURE 3.21 – Interface Liste-Ingénieur

La figure ci-dessous présente la liste des agriculteurs.

Nom	Prénom	Date de naissance	E-mail	Téléphone	Adresse	action
asma	ouertani	04-11-1195	asmaouertani123456@gmail.com	28365744	bousalem	 
wertenli	asma	Ven 11 juillet 1997 20:00:14 GMT+0000 (GMT)	asma@gmail.com	22333666	bousalem	 
balti	ahmed	Ven 27 août 2021 09:38:47 GMT+0000 (GMT)	ahmed@gmail.com	22333366	bousalem	 
asma	jamaï	Mer 13 oct. 2021 19:26:55 GMT+0100 (CET)	asmajamai1994@gmail.com	21396523	ghardimaou	 

FIGURE 3.22 – Interface Liste-Agriculteur

3.9.2 Les tests :

Le test d'un produit logiciel est un processus consistant qui visent à garantir le bon fonctionnement du système à travers une comparaison de comportement attendu et de résultat obtenu nous avons élaboré dans ce tableau un ensemble de cas de scénario de tests fonctionnels relatifs à notre premier sprint.

En effet on a deux types d'interaction homme machine (IHM), à chacune l'action de l'utilisateur se différencié d'un cas à une autre a un autre :

- i. La première IHM, l'utilisateur doit remplir un formulaire pour qu'il communique avec son SGBD :

- Inscription
- Authentification
- Ajout
- Modification

Le tableau ci-dessous présente la fiche test fonctionnel de l'interface inscription :

ID Inscription
Type Fonctionnel
Testeur : Asma Ouertani
<p>Object attendu : l'utilisateur doit insère un Email et un mot de passe pour créer un compte</p> <p>Description : Après le développement de l'interface d'inscription on va introduire les informations nécessaires dans le formulaire et on va faire des tests. (Test sur la syntaxe de chaque champ, tests sur l'existence des clés)</p> <p>Règles métier</p> <p>Règle 1 : l'utilisateur accède à l'interface inscription</p> <p>Règle 2 : l'utilisateur doit remplir le formulaire d'inscription</p> <p>Règle 3 : l'utilisateur doit respecter les contrôles de saisie</p>
Tests d'acceptation sur l'entité : Agriculteur
<p>Nom de scénario 1 :</p> <p>contrôle syntaxique des champs lorsque l'utilisateur remplit le formulaire il doit respecter les champs suivants :</p> <ul style="list-style-type: none"> ✓ Nom : l'agriculteur doit écrire les caractères composer par des lettre [a..z] ou [A..Z] ✓ Prénom : l'agriculteur doit écrire les caractères composer par des lettre [a..z] ou [A..Z] ✓ Datenaiss : l'agriculteur doit écrire les dates composer par [jj/mm/aaaa] ✓ Email : l'agriculteur écrit les caractères composer par des lettre [a..z] ou [A..Z] et : ou [0..9], ce champs doivent contenir aussi le « @ » ✓ Password : l'agriculteur écrit les caractères composer par des lettre [a..z] ou [A..Z] et :ou [0..9], ✓ Téléphone : l'agriculteur doit écrire huit caractères composer par des numéro [0..9] ✓ Adresse : l'agriculteur écrit les caractères composer par des lettre [a..z] ou [A..Z] et : ou [0..9], <p>Nom de scénario :</p> <p>la non redondance des clés dans la table de le SGBD.</p> <ul style="list-style-type: none"> ✓ Chaque agriculteur est défini par son « Email » et son « Password » ✓ Lorsque l'agriculteur remplit le formulaire, il doit saisie des nouveau identifiants (Email, Password) non enregistré dans l'entité agriculteur <p>Nom de scénario :</p> <p>L'utilisateur doit compléter le processus d'inscription</p> <ul style="list-style-type: none"> ✓ L'utilisateur doit remplir tous les champs du formulaire ✓ L'utilisateur doit cliquer sur le bouton « s'inscrire »

ii. La deuxième IHM : l'utilisateur doit cliquer sur un bouton pour qu'il communique avec le SGBD :

- Supprimer

Le tableau ci-dessous présente le fiche test fonctionnel de l'interface Suppression :

ID Suppression
Type : Fonctionnel
Testeur : Asma Ouertani
Object attendu : l'administrateur doit Supprimer l'agriculteur désiré
Description : Après le développement de l'interface de suppression on va vérifier toutes les interactions de système après la consultation de la liste des agriculteurs, choisir un agriculteur le système demande la confirmation de suppression L'administrateur valider la suppression.
Règles métier Règle 1 : l'administrateur accède à l'interface supprimer agriculteur Règle 2 : l'administrateur consulte la liste des agriculteurs Règle 3 : l'administrateur choisie un agriculteur Règle 4 : l'administrateur confirme la suppression
Tests d'acceptation sur l'entité : Agriculteur
Nom de scénario 1 : l'administrateur confirme la suppression <ul style="list-style-type: none">✓ L'administrateur consulte la liste des agriculteurs✓ L'administrateur choisir un agriculteur✓ L'administrateur demande la suppression✓ L'administrateur confirmer la suppression✓ Dans la table agriculteur : l'état de l'agriculteur doit être modifié :

3.10 Conclusion

Au cours de ce chapitre, nous avons présenté le premier sprint. Pour ce faire, nous avons passé par la présentation du backlog product, la spécification des besoins, la conception et la réalisation. Dans le chapitre suivant nous entamons le deuxième sprint.

Chapitre 4

Mise en œuvre du sprint 2

4.1 Introduction

Ce chapitre présente les détails du deuxième Sprint. Nous commençons par la planification du Sprint en précisant les cas d'utilisations associés. Puis, Nous passons par une phase d'analyse, ensuite nous présenterons quelques interfaces réalisées au cours de ce sprint. Nous détaillons chaque cas en présentant les diagrammes d'analyse séquences, activité et la réalisation.

4.2 Planification de sprint2

Cette étape est réalisée pour présenter le sprint2 ainsi les cas d'utilisations qu'elle compose.

Ce schéma montre la planification de sprint 2 :



FIGURE 4.1 – Planification de sprint 2

4.2.1 Backlog de sprint2

En partant du même principe que le sprint précédent, nous commençons par définir le but de notre deuxième sprint. Le tableau4.1 décrit le Backlog du sprint2.

ID	Histoire	Date début	Date fin	Responsable
1	Gérer plante	13/06/2021	20/06/2021	Ouertani Asma
2	Gérer maladie	21/06/2021	30/06/2021	
3	Gérer traitement	01/07/2021	12/07/2021	

TABLE 4.1 – Backlog de sprint1

4.2.2 Tableau des tâches

Les tâches relatives au deuxième sprint (A T= 13/06/2021) sont illustrées dans le tableau des tâches suivant.

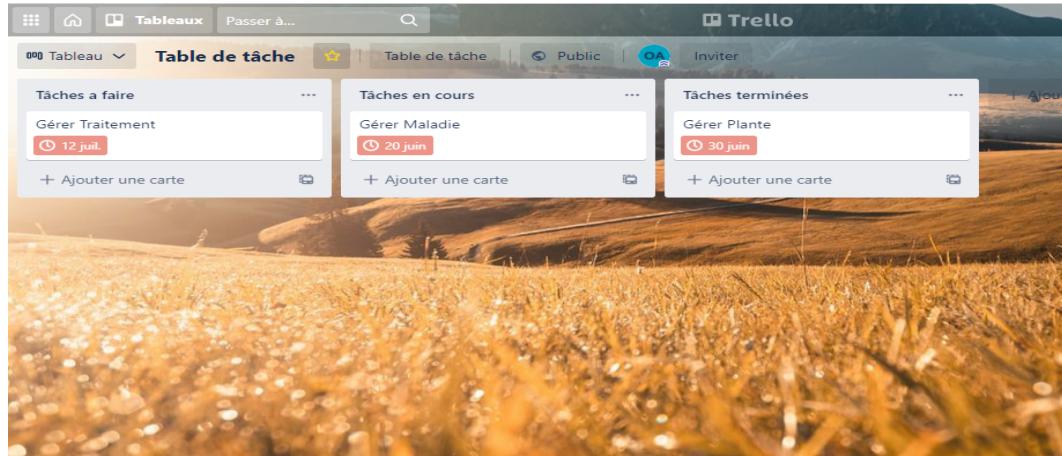


FIGURE 4.2 – planification de sprint 2

4.2.3 Phase d'analyse

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

- a) **Définition des acteurs et des fonctionnalités de sprint 2**
 L'acteur principal de ce sprint est :

Administrateur c'est l'acteur a pour rôle de contrôler et configurer l'insertion des plantes, des maladies et des traitements dans la base de données.

Le diagramme de cas d'utilisation global de sprint 2 est montré dans la figure ci-dessous.

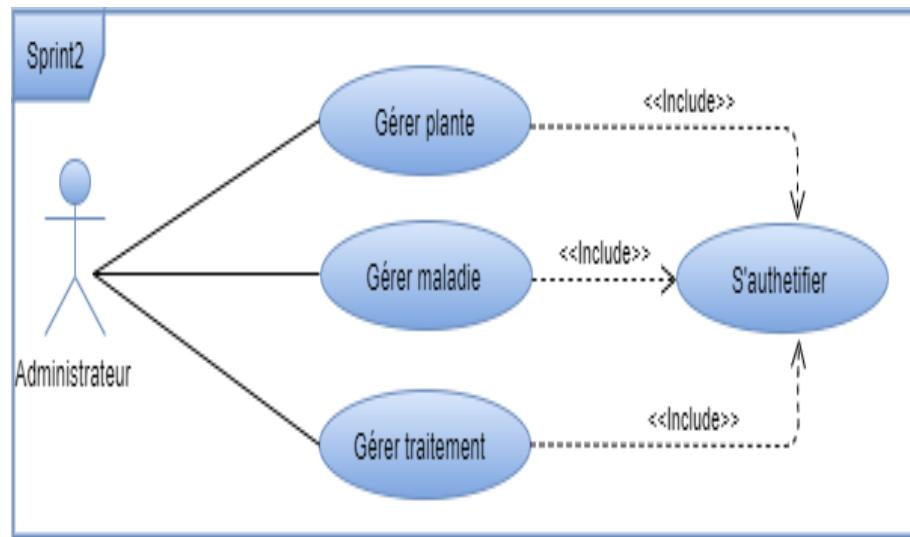


FIGURE 4.3 – Diagramme de cas d'utilisation de sprint 2

b) Raffinement des cas d'utilisation de sprint 2

★ Raffinement de cas d'utilisation « Gérer plante»

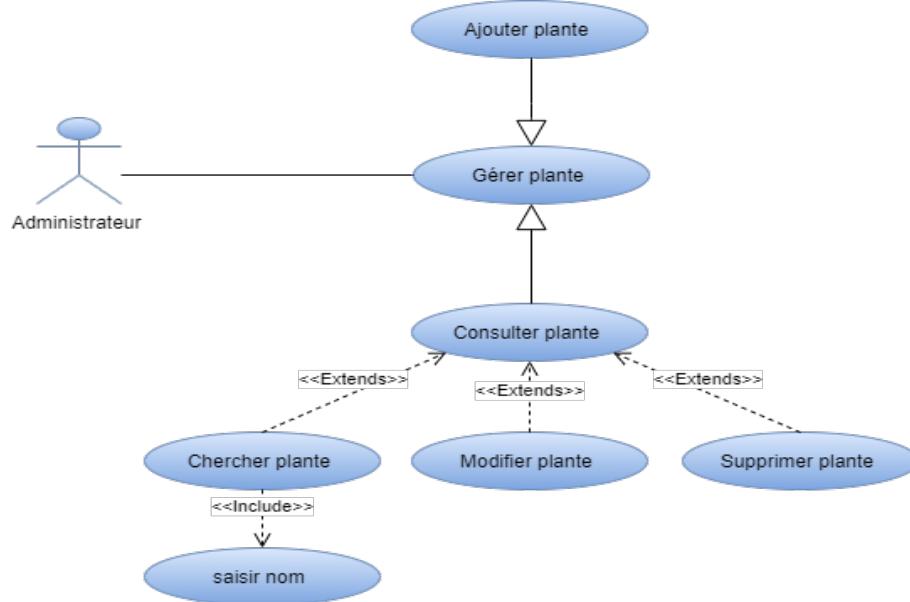


FIGURE 4.4 – Raffinement de cas d'utilisation « Gérer plante»

★ Description textuelle de cas d'utilisation " Gérer plante "

Titre	Gérer Plante
Acteur	Administrateur
Objectif	Lors de l'accès au l'application, l'administrateur doit ajouter , consulter,chercher,modifier,et supprimer une plante.
Pré-condition	<ul style="list-style-type: none"> • L'administrateur doit être authentifié. • L'administrateur doit accéder à l'interface gérer plante.
Post-condition	(ajouter, consulter, modifier, supprimer) des plantes avec succès.
Scénario Principal	<p>Se cas d'utilisation commence lorsque l'administrateur demande au système de faire la gestion de ses plantes :</p> <ul style="list-style-type: none"> ■ Enchainement(a) : Ajouter plante <ol style="list-style-type: none"> 1- L'administrateur demande la page d'ajouter plante, 2- Le système affiche un formulaire pour remplir, 3- L'administrateur remplir le formulaire et clique sur enregistrer , 4- Le système vérifier les données et affiche un message (" plante ajouter avec succès "). ■ Enchainement(b) : Consulter plante <ol style="list-style-type: none"> 1- l'administrateur clique sur le bouton « consulter », 2- Le Système affiche la liste des plantes de la base des données, 3- Le Système donne trois scenarios à l'administrateur « chercher » « modifier », ou « supprimer ». ■ Enchainement(b.1) : chercher plante <ol style="list-style-type: none"> 1- L'administrateur accédé à la zone de recherche, 2- L'administrateur saisit le nom de la plante, 3-Le système affiche la liste de résultat de recherche 4- Le système vérifie les données saisies, ■ Enchainement(b.2) :Modifier plante <ol style="list-style-type: none"> 1- L'administrateur choisit une plante, 2- L'administrateur demande la page « modifier plante » 3- Le système affiche un formulaire pour modifier les données de la plante, 4- L'administrateur remplit le formulaire et clique sur enregistrer , 5- Le système vérifier les données et affiche un message (" plante modifier avec succès "). ■ Enchainement(b.3) :Supprimer plante <ol style="list-style-type: none"> 1- L'administrateur choisit une plante de la liste affichée (liste des plantes déjà inscrite dans l'application) et clique sur le bouton « supprimer plante», 2- Le système supprime la plante de la base de données.
Scénario alternatif et d'exception :	<p>1-L'administrateur annule le contrôle :</p> <ul style="list-style-type: none"> • Le cas d'utilisation se termine avec échec. 2- Les données saisies sont incorrectes : <ul style="list-style-type: none"> • Le système affiche un message d'erreur « vérifier vos données » • Le cas d'utilisation se termine avec échec.

TABLE 4.2 – Description textuelle de cas d'utilisation " Gérer plante "

★ Raffinement de cas d'utilisation « Gérer maladie»

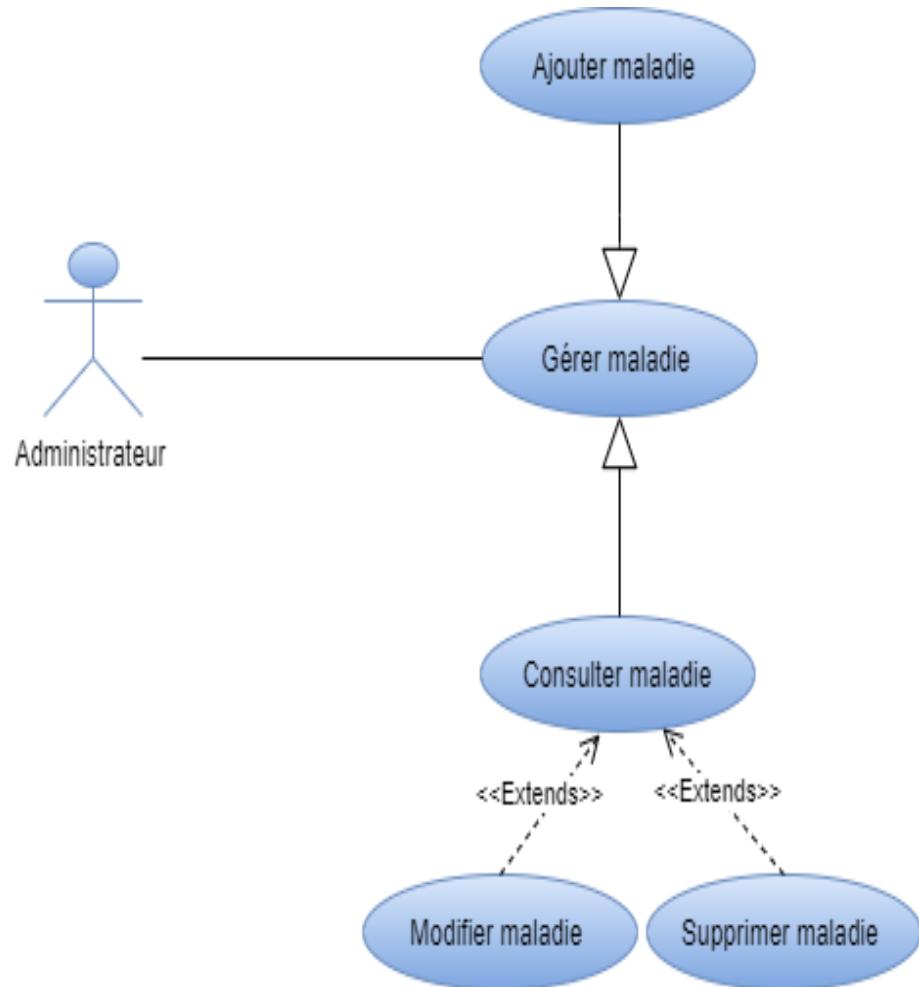


FIGURE 4.5 – Raffinement de cas d'utilisation « Gérer maladie »

★ Description textuelle de cas d'utilisation " Gérer maladie "

Titre	Gérer Maladie
Acteur	Administrateur
Objectif	Lors de l'accès au l'application, l'administrateur doit ajouter, consulter, modifier, et supprimer une maladie.
Pré-condition	<ul style="list-style-type: none"> • L'administrateur doit être authentifié. • L'administrateur doit accéder à l'interface gérer maladie.
Post-condition	(ajouter, consulter, modifier, supprimer) des maladies avec succès.
Scénario Principal	<p>Ce cas d'utilisation commence lorsque l'administrateur demande au système de faire la gestion de ses maladies :</p> <ul style="list-style-type: none"> ■ Enchainement(a) : Ajouter maladie <ol style="list-style-type: none"> 1- L'administrateur demande la page d'ajout des maladies , 2- Le système affiche un formulaire à remplir, 3- L'administrateur remplit le formulaire et clique sur enregistrer , 4- Le système vérifie les données saisies vérifier les données et affiche un message (" malaide ajoutée avec succès "). ■ Enchainement(b) : Consulter maladie <ol style="list-style-type: none"> 1- L'administrateur clique sur le bouton « consulter », 2- Le système affiche la liste des maladies de la base des données, 3- Le système donne deux scénarios à l'administrateur « modifier »ou « supprimer ». ■ Enchainement(b.1) :Modifier maladie <ol style="list-style-type: none"> 1- L'administrateur choisit une maladie, 2- L'administrateur demande la page « modifier maladie » 3- Le système affiche un formulaire pour modifier les données de la maladie, 4- L'administrateur remplir le formulaire et clique sur enregistrer , 5- Le système vérifie les données et affiche un message (" maladie modifiée avec succès "). ■ Enchainement(b.2) :Supprimer maladie <ol style="list-style-type: none"> 1- L'administrateur choisit une maladie de la liste affichée (liste des maladies déjà inscrite dans l'application) et clique sur le bouton « supprimer maladie», 2- Le système supprime la maladie de la base de données.
Scénario alternatif et d'exception :	<p>1-L'administrateur annule le contrôle :</p> <ul style="list-style-type: none"> • Le cas d'utilisation se termine avec échec. 2- Les données saisies sont incorrectes : <ul style="list-style-type: none"> • Le système affiche un message d'erreur « vérifier vos données » • Le cas d'utilisation se termine avec échec.

TABLE 4.3 – Description textuelle de cas d'utilisation " Gérer maladie"

★ Raffinement de cas d'utilisation « Gérer traitement»

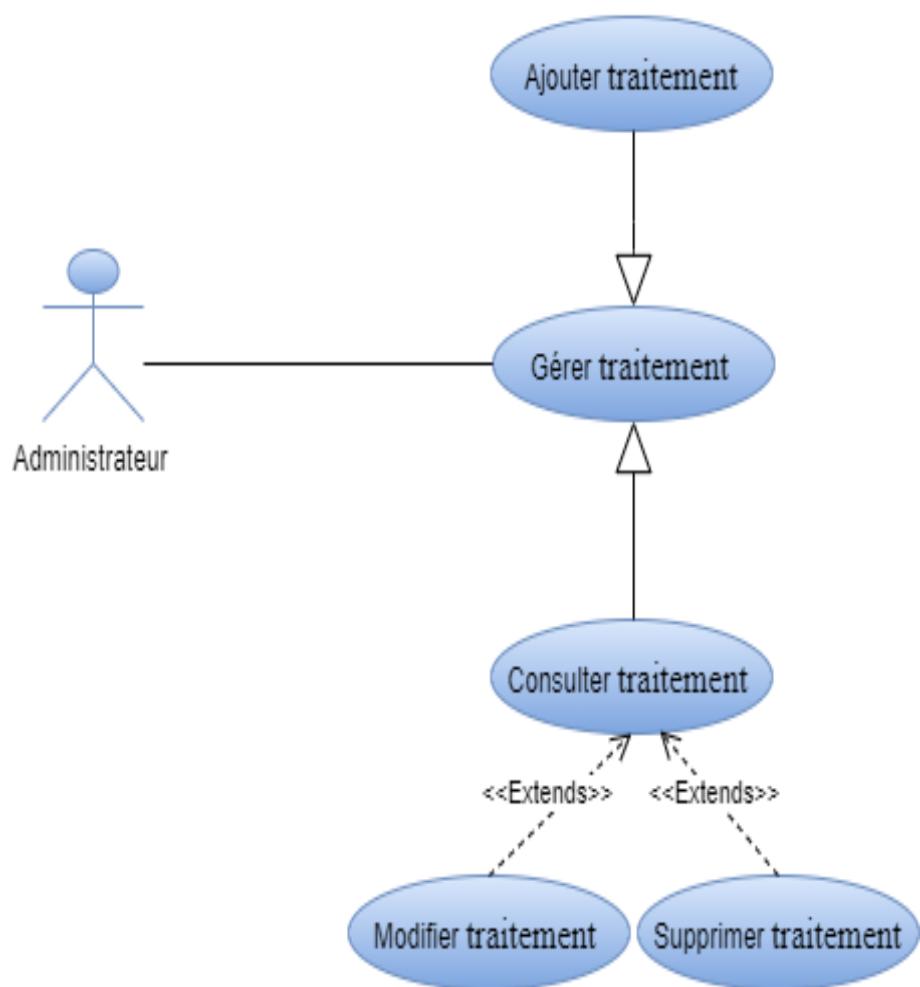


FIGURE 4.6 – Raffinement de cas d'utilisation « Gérer traitement »

★ Description textuelle de cas d'utilisation " Gérer traitement "

Titre	Gérer Traitement
Acteur	Administrateur
Objectif	Lors de l'accès au l'application, l'administrateur doit ajouter, consulter, modifier, et supprimer un traitement.
Pré-condition	<ul style="list-style-type: none"> • L'administrateur doit être authentifié. • L'administrateur doit accéder à l'interface gérer traitement.
Post-condition	(ajouter, consulter, modifier, supprimer) des traitements avec succès.
Scénario Principal	<p>Ce cas d'utilisation commence lorsque l'administrateur demande au système de faire la gestion de ses traitements :</p> <ul style="list-style-type: none"> ■ Enchainement(a) : Ajouter traitement <ol style="list-style-type: none"> 1- L'administrateur demande la page d'ajouter traitement, 2- Le système affiche le formulaire à pour remplir, 3- L'administrateur remplit le formulaire et clique sur enregistrer , 4- Le système vérifie les données et affiche un message (" traitement ajouter avec succès "). ■ Enchainement(b) : Consulter traitement <ol style="list-style-type: none"> 1- l'administrateur clique sur le bouton « consulter », 2- Le système affiche la liste des traitements de la base des données, 3- Le système donne deux scenarios à l'administrateur « modifier »ou « supprimer ». ■ Enchainement(b.1) :Modifier traitement <ol style="list-style-type: none"> 1- L'administrateur choisit un traitement, 2- L'administrateur demande la page « modifier traitement » 3- Le système affiche un formulaire pour modifier les données de ce traitement , 4- L'administrateur remplir le formulaire et clique sur enregistrer , 5- Le système vérifie les données et affiche un message ("traitement modifier avec succès "). ■ Enchainement(b.2) :Supprimer traitement <ol style="list-style-type: none"> 1- L'administrateur choisit un traitement de la liste affichée (liste des traitements déjà inscrit dans l'application) et clique sur le bouton « supprimer traitement», 2- Le système supprime la traitement de la base de données.
Scénario alternatif et d'exception :	<p>1-L'administrateur annule le contrôle :</p> <ul style="list-style-type: none"> • Le cas d'utilisation se termine avec échec. 2- Les données saisies sont incorrectes : <ul style="list-style-type: none"> • Le système affiche un message d'erreur « vérifier vos données » • Le cas d'utilisation se termine avec échec.

TABLE 4.4 – Description textuelle de cas d'utilisation " Gérer traitement"

4.3 Phase de conception

Au niveau de cette section, nous présentons le diagramme de classe, les diagrammes de séquences ainsi que les diagrammes d'activités de ce sprint.

4.3.1 Conception statique de sprint

a) Diagramme de classe de sprint 2

La figure ci-dessous représente le diagramme de classes utilisé pour le développement du sprint 2.

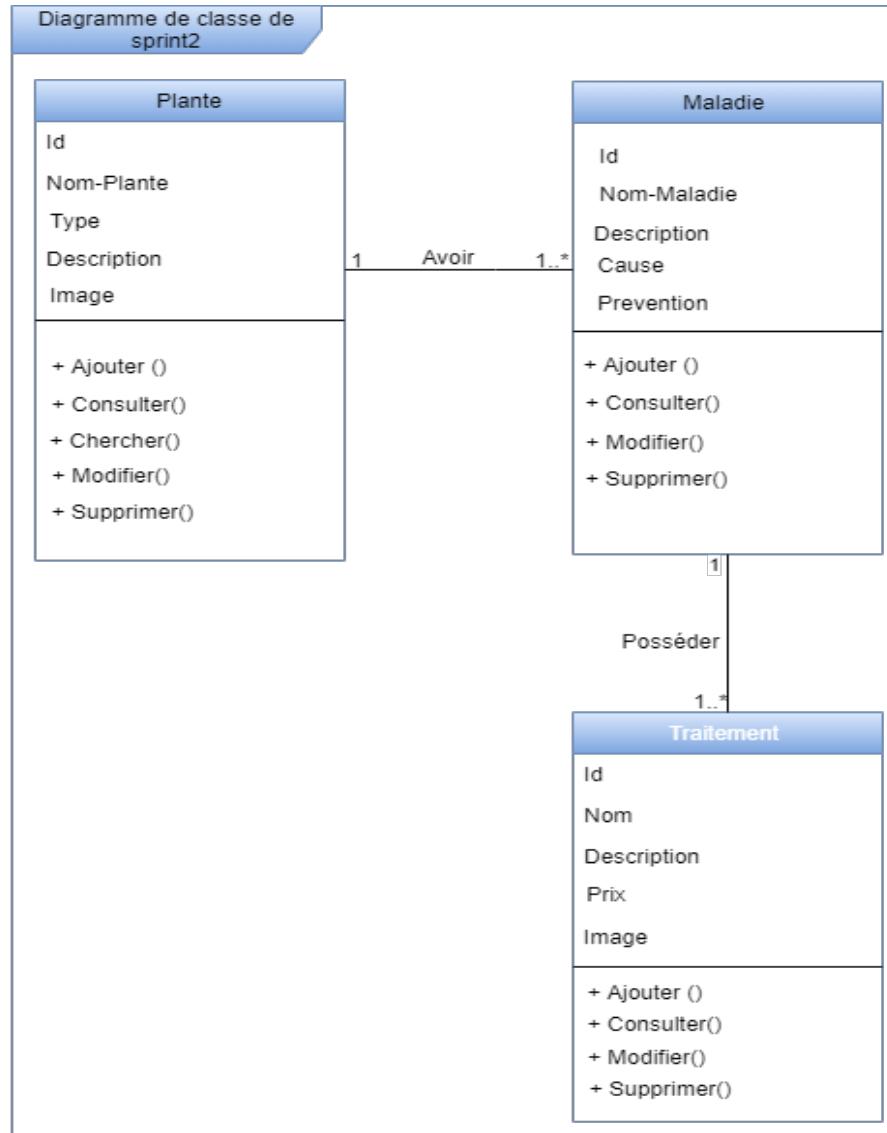


FIGURE 4.7 – Diagramme de classe de sprint 2

b) Dictionnaire de données

Le dictionnaire de base de données est une définition des attributs de chaque classe de sprint deux.

Dictionnaire de données de la table Plante

Plante				
Attribut	Type	Taille	Obligatoire	Contrainte
Id	Int	50	Oui	Clé primaire
Nom-Plante	Varchar	20	oui	
Type	Varchar	20	Oui	
Description	Varchar		Oui	
Image	varchar		Oui	

TABLE 4.5 – dictionnaire de données de la table Plante

Dictionnaire de données de la table Maladie

Maladie				
Attribut	Type	Taille	Obligatoire	Contrainte
Id	Int	10	Oui	Clé primaire
Nom-Maladie	Varchar	20	Oui	
Description	Varchar		Oui	
Cause	Varchar		Oui	
Prevention	Varchar		Oui	

TABLE 4.6 – dictionnaire de données de la table Traitement

Dictionnaire de données de la table Traitement

Traitement				
Attribut	Type	Taille	Obligatoire	Contrainte
Id	Int	10	Oui	Clé primaire
Nom	Varchar	20	Oui	
Description	Varchar		Oui	
Image	Varchar			
Prix	Double		Oui	

TABLE 4.7 – dictionnaire de données de la table Traitement

4.4 Conception dynamique du deuxième sprint

4.4.1 Les diagrammes de séquence

★ Diagramme de séquence de cas d'utilisation « Ajouter Plante »

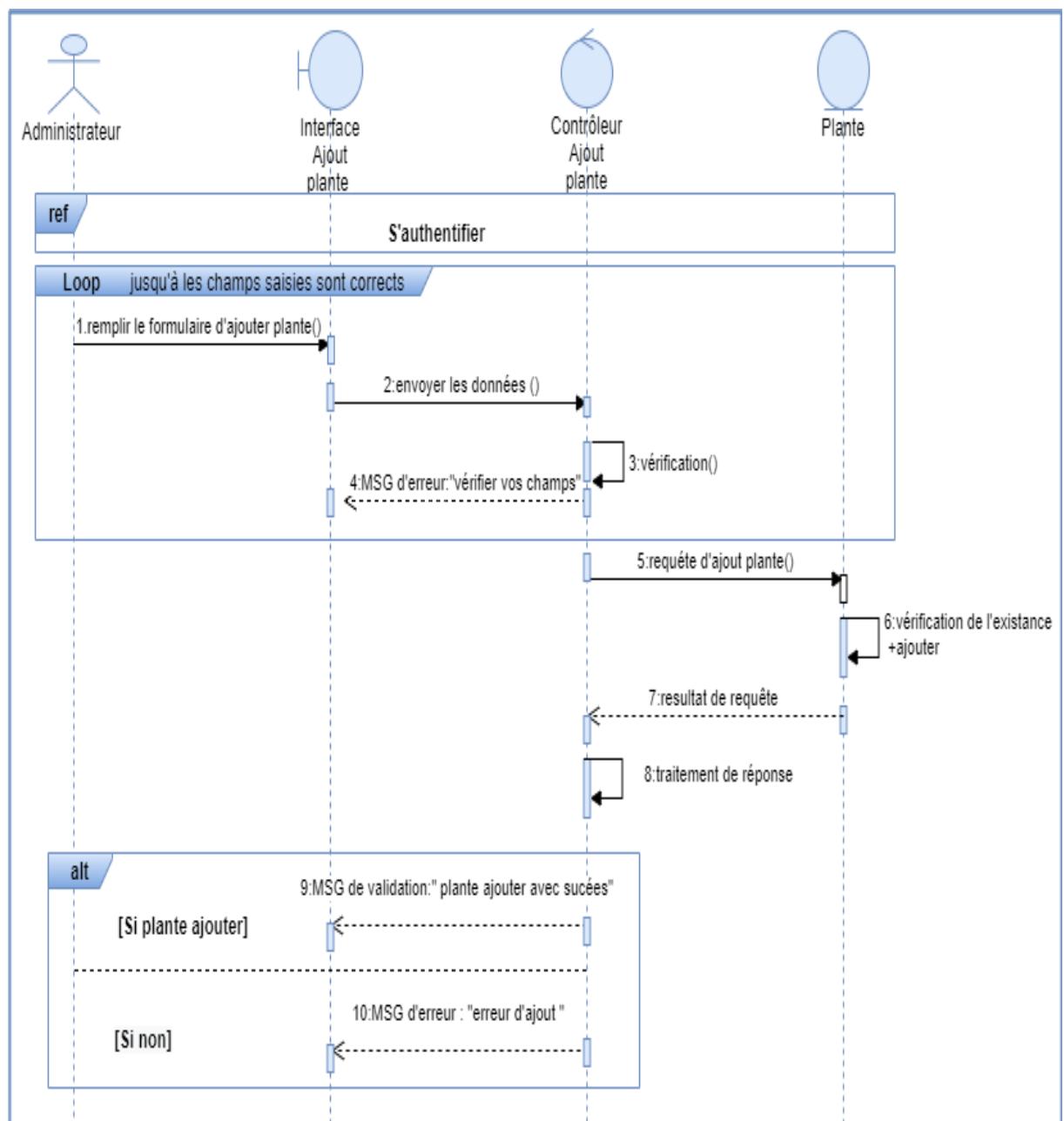


FIGURE 4.8 – Diagramme de séquence de cas d'utilisation « Ajouter Plante »

★ Diagramme de séquence de cas d'utilisation « Chercher Plante »

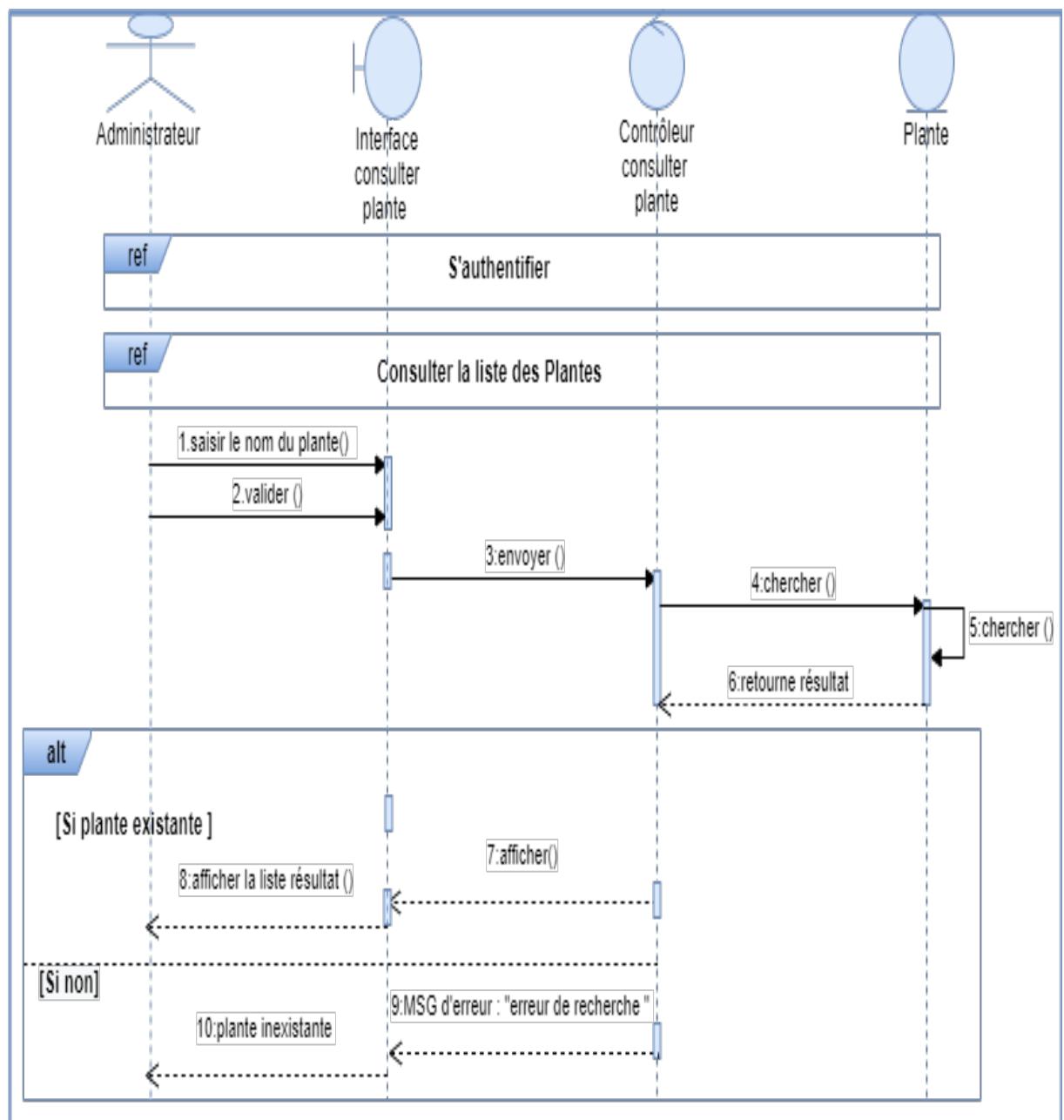


FIGURE 4.9 – Diagramme de séquence de cas d'utilisation « Chercher Plante »

★ Diagramme de séquence de cas d'utilisation « Modifier Plante »

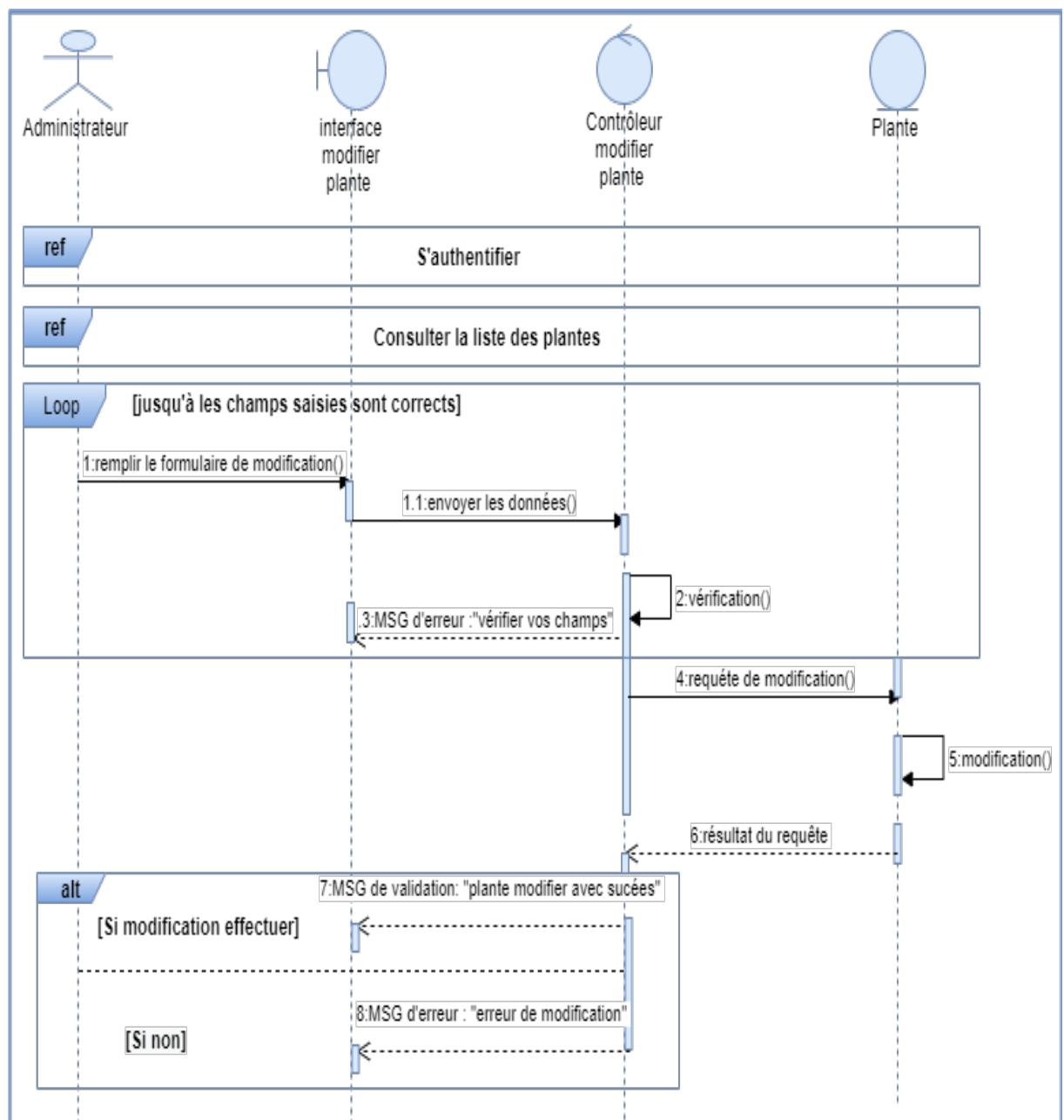


FIGURE 4.10 – Diagramme de séquence de cas d'utilisation « Modifier Plante »

★ Diagramme de séquence de cas d'utilisation « Supprimer Plante »

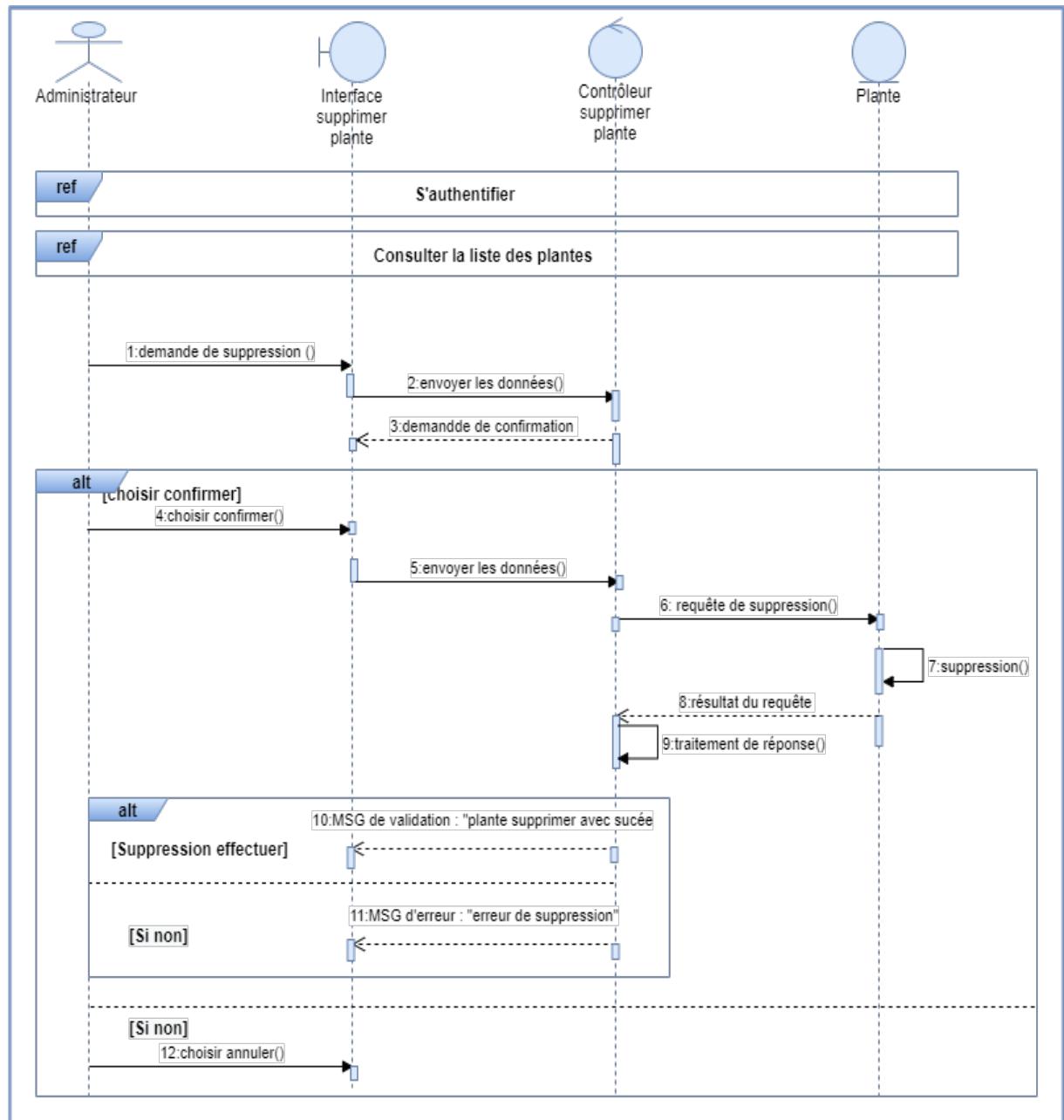


FIGURE 4.11 – Diagramme de séquence de cas d'utilisation « Supprimer Plante »

4.5 Diagrammes d'activités :

★ Diagramme d'activité du cas d'utilisation "Ajouter Plante "

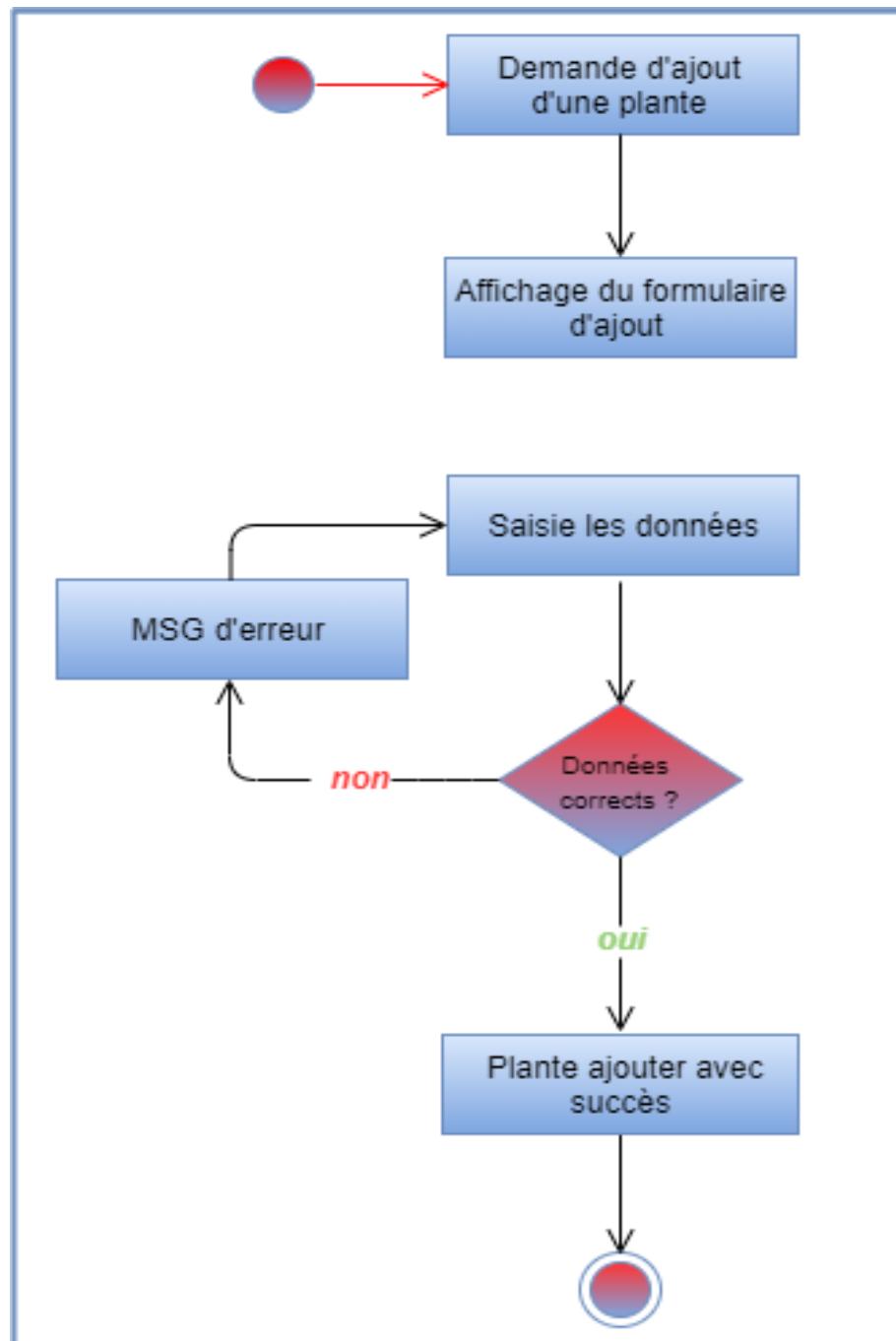


FIGURE 4.12 – Diagramme d'activité du cas d'utilisation "Ajouter Plante "

★ Diagramme d'activité du cas d'utilisation "Chercher Plante "

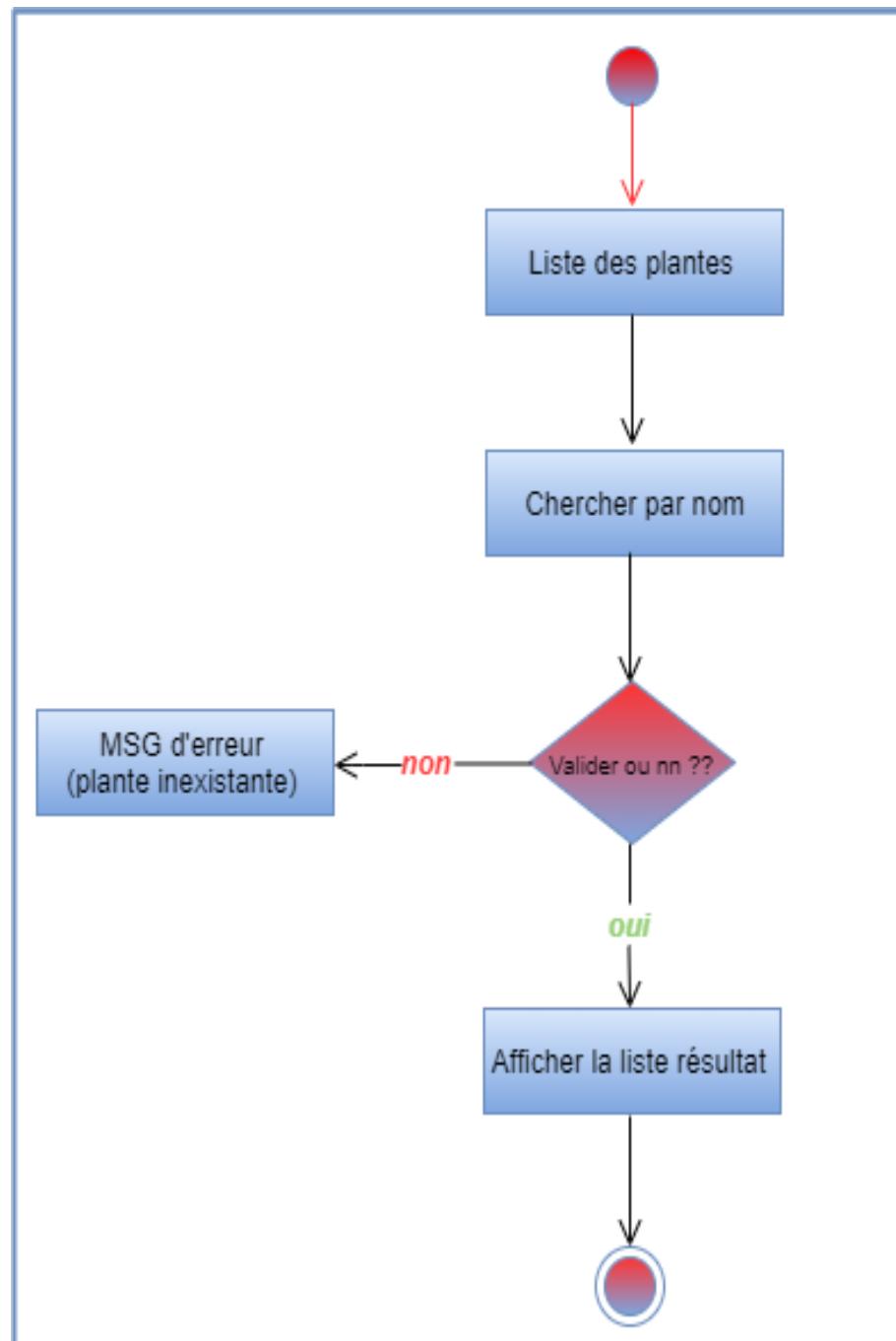


FIGURE 4.13 – Diagramme d’activité du cas d’utilisation “Chercher Plante”

★ Diagramme d’activité du cas d’utilisation “Modifier Plante”

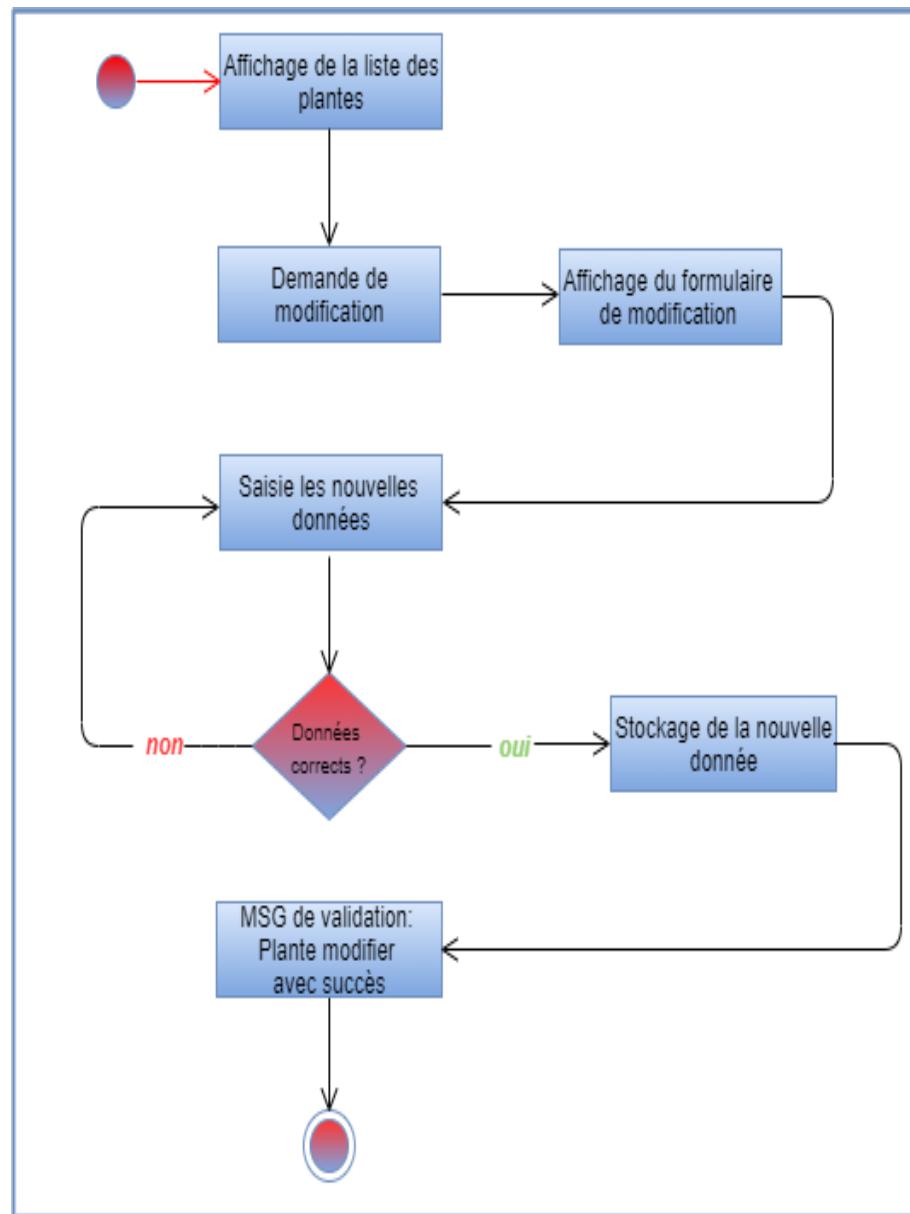


FIGURE 4.14 – Diagramme d’activité du cas d’utilisation "Modifier Plante"

4.6 Réalisation et tests

Cette partie est consacrée à l’exposition du travail achevé à travers des captures d’écrans de différentes interfaces développées pendant ce sprint. On a validé ces interfaces par des plusieurs types de tests.

4.6.1 Les Interfaces :

La figure ci-dessous présente l’interface Ajout plante.

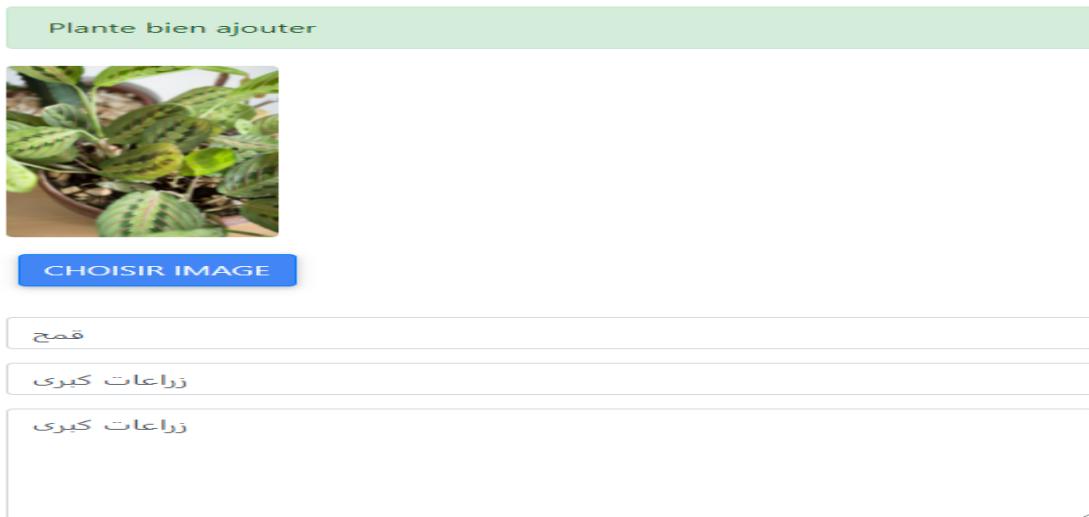


FIGURE 4.15 – Interface Ajout plante

La figure ci-dessous présente l'interface Liste-plante.

Image	Nom	Type	Description	Action
	قمح	زراعات كبيرة	لتغذية الحنطة حشائش حولي من الفصيلة التجلبية، وتزرع منه أنواع كثيرة منها الشعير الأحمر أو البليت وهي شبه القمح، وبخته الشعير الأسود، وكذا يدخل الشعير ضمن الصناعة الغذائية، وهي غالباً في المرتبة الرابعة بعد القمح والأرز والذرة الصفراء، وذلك من حيث المساحة والإنتاج.	
	الشعير	زراعات كبيرة	الشعير نبات حشبي حولي من الفصيلة التجلبية، وتزرع منه أنواع كثيرة منها الشعير الأحمر أو البليت وهي شبه القمح، وبخته الشعير الأسود، وكذا يدخل الشعير ضمن الصناعة الغذائية، وهي غالباً في المرتبة الرابعة بعد القمح والأرز والذرة الصفراء، وذلك من حيث المساحة والإنتاج.	
	الفول	زراعات كبيرة	جذور نبات الفول وهي يتمدّق في التربة إلى مسافات قد تصل إلى 80 سم، يتغذى من الأعلى إلى جذوره تتدلى بشكل أفقى إلى مسافة تصل 50 سم تقريباً، هذا التدرج ساعد النبات على امتصاص غذائه من التربة كما يساعد على الزيادة في تكون الحقد الكثيرة المتشعبة لازروت الجوزي في إطار المحاصيل.	
	القصبة	زراعات كبيرة	الشوفان هو نبات حشبي حولي شبه الحنطة والشعير في الشكل وهو ينتمي عادةً إلى عائلة القéréales ويزوره متزورةً من حب الحنطة والشعير ويعرف عادةً بالقرآن والعلاء عادةً تقول الروان والزيوان، لم يربط اسم الشوفان في المعاجم العربية المقيدة ولا في المفردات وقد عرف في الماضي باسماء مختلفة مثل كريطمأن	

FIGURE 4.16 – Interface Liste-plante

La figure ci-dessous présente l'interface Liste-traitement.

	Image	Prix	Description	Action
220			<p>يحتوى الكلورحيدروكسيد النحاس فى صورة مسحوق قابل للذوبان هيدروكسيد النحاس على 383.7 جرام من مادة هيدروكسيد النحاس تعدل 25% نحاس معدنى يتميز هذا المركب بدلا جم حبيباته لذا فهو يعطى تغطية وانتشار الفضل على سطح النباتات المعاملة مما يساعد في الوقاية من الإصابة بالأمراض النقرية لفترة طويلة دون أن يحدث أي على النباتات المعاملة به يتميز المركب بعد إحداث تكاثر أو تجمادات من المسحوق في خزان الله (No Phytotoxicity) اختبار الرش ولا يتسبب في إحداث إنسداد للشرايين يتميز هذا المركب بفاعليته العالية في الوقاية ضد العديد من الأمراض النقرية والتكتونية التي تصيب العديد من محاصيل الحقل ويساهم الفاكهة كما أنه يستخدم بكفاءة عالية في القضاء على أنواع الأشتلات والموت العكسي في المروج وأشجار الفاكهة أكدت التجارب أن استخدام هذا المركب لمكافحة الأشتلات على اتجاه الوضى لا يضر بالثمار إذا تم الرش أثناء وجود الثمار على الأشجار - هذا المركب مصرح باستخدامه في الزراعات المضبوطة حيث أنه مسجل بالمعهد المركزي للزراعة الأمراض التي يعالجها فيه زورو 2007 (Zero PHI) هو صفر برو (Zero PHI) الفضوري فهو مما يتميز هذا المركب أن قدرة ما في المصد طبقاً لتوصيات وزارة الزراعة 25% WP</p>	  
80			<p>هذا المركب قابل لذابتاج بالماء بنسبة 100% مما يزيد من كفاءته في مكافحة سبب الأمراض النقرية والتكتونية بسبب تركيبته المميزة فإنه يساعد على التوزيع المتجانس لأيونات النحاس على سطح النباتات المعاملة مما يساعد في القضاء على جراثيم النظريات في أسرع وقت ممكن لابسبي تأكيل أو إنسداد شرايين أورت الرش كما أنه لا يترك عوالق داخل خزانات الرشاشات وكلها مشكلات تواجه مستخدمي مركبات النحاس التقليدية بفضل هذا المركب بسرعة امتصاصه داخل أنسجة النبات المعاملة به وليس له أثر باق على سطح النباتات المعاملة والتي قد تسبب مشكلة لبعض المزارعين خاصة منتجي وبصري زهور ونباتات الزينة لا يحتاج إلى إضافة مواد لإصبع أو نشرة وذلك نظراً لأن تركيبة المركب تحتوي على مواد ترفع كفاءة انتشار، والتصادف بالنباتات بالإضافة أن النحاس يدخل على مواد رائحة طبيعية تساعد على دخوله وانتشاره داخل أنسجة النباتات المعاملة به بسبب سرعة امتصاصه داخل أنسجة النباتات المعاملة (دخل ساحة واحدة بعد الرش) تقل احتمالات فقدة من على سطح النباتات المعاملة بفضل الأمطار أو الندى أو ماء الرش وبالتالي ليس هناك حاجة لإعاذه الرش إذا سقطت الإمطار بعد رشه بساعة</p>	  

FIGURE 4.17 – Interface Liste-Traitement

4.6.2 Les tests :

- La première IHM, l'administrateur doit remplir un formulaire pour qu'il communique avec son SGBD :

- Ajout
- Consulte
- Recherche
- Modification

Le tableau ci-dessous présente la fiche test fonctionnel de l'interface inscription :

ID Ajout
Type : Fonctionnel
Testeur : Asma Ouertani
Object attendu : l'administrateur doit remplir un formulaire pour ajouter une plante
Description : Après le développement de l'interface d'ajout on va introduire les informations nécessaires dans le formulaire et on va faire des tests. (Test sur la syntaxe de chaque champ, tests sur l'existence des clés)
Règles métier Règle 1 : l'administrateur accède à l'interface d'ajout Règle 2 : l'administrateur doit remplir le formulaire d'ajout Règle 3 : l'administrateur doit respecter les contrôles de saisies
Tests d'acceptation sur l'entité : Plante
Nom de scénario 1 : contrôle syntaxique des champs lorsque l'administrateur remplit le formulaire il doit respecter les champs suivants : <ul style="list-style-type: none"> ✓ Nom : l'administrateur doit écrire les caractères composer par des lettre [...] ✓ Type : l'administrateur doit écrire les caractères composer par des lettre [..] ✓ Description : l'administrateur doit écrire les caractères composer par des lettre [..] ✓ Image : l'administrateur doit mettre en photo sur un serveur en ligne et saisir son URL Nom de scénario : l'administrateur doit compléter le processus d'ajout . <ul style="list-style-type: none"> ✓ L'administrateur doit remplir tous les champs du formulaire ✓ L'administrateur doit cliquer sur le bouton « Ajouter »

ii. La deuxième IHM : L'administrateur doit cliquer sur un bouton pour qu'il communique avec le SGBD :

- Suppression

Le tableau ci-dessous présente la fiche test fonctionnel de l'interface Supprimer :

ID Suppression
Type : Fonctionnel
Testeur : Asma Ouertani
Object attendu : l'administrateur doit supprimer la plante désiré
Description : Après le développement de l'interface de suppression on va vérifier toutes les interactions de système après la consultation de la liste des plantes, choisir une plante, le système demande la confirmation de suppression
Règles métier Règle 1 : l'administrateur consulte la liste des plantes Règle 3 : l'administrateur choisie une plante Règle 4 : l'administrateur confirme la suppression
Tests d'acceptation sur l'entité : Plante
Nom de scénario 1 : l'administrateur confirme la suppression <ul style="list-style-type: none">✓ L'administrateur consulte la liste des Plantes✓ L'administrateur choisir une Plante✓ L'administrateur demande la suppression✓ L'administrateur confirmer la suppression✓ Dans la table Plante : l'état de Plante doit être modifié :

4.7 Conclusion

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

Chapitre 5

Mise en œuvre du sprint 3

5.1 Introduction

Ce chapitre présente les détails du troisième Sprint. Nous commençons par la planification du Sprint en précisant les cas d'utilisations associés. Puis, Nous passons par une phase d'analyse, ensuite nous présenterons quelques interfaces réalisées au cours de ce sprint. Nous détaillons chaque cas en présentant les diagrammes d'analyse séquences, activité et la réalisation.

5.2 Planification de sprint3

Cette étape est réalisée pour présenter le sprint 3 ainsi les cas d'utilisations qu'elle compose.

Ce schéma montre la planification de sprint 3 :

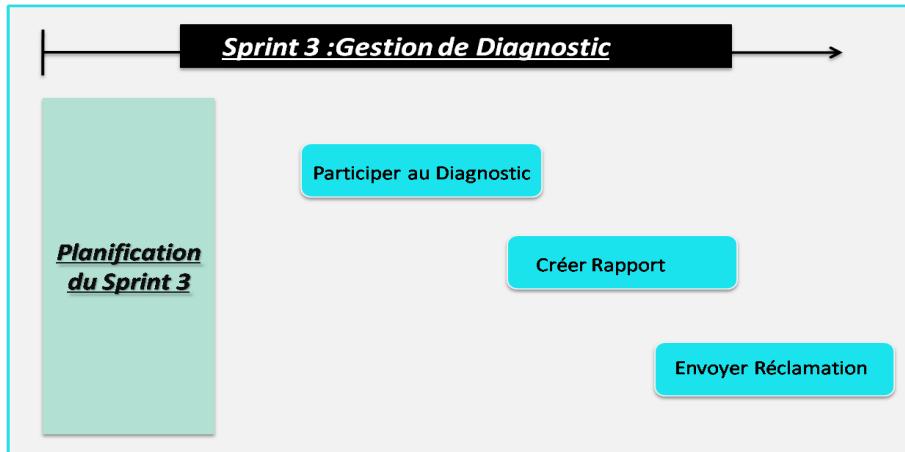


FIGURE 5.1 – Planification de sprints3

5.2.1 Backlog de sprint

En partant du même principe que le sprint précédent, nous commençons par définir le but de notre troisième sprint. Le tableau1.5 décrit le Backlog du sprint.

ID	Histoire	Date début	Date fin	Responsable
1	Participer au Diagnostic	10/08/2021	30/08/2021	Ouertani Asma
2	Créer rapport	01/09/2020	11/09/2021	
3	Envoyer réclamation	14/09/2021	30/09/2021	

TABLE 5.1 – Backlog de sprint3

5.3 Tableau des tâches

Les tâches relatives au troisième sprint (A T= 10/08/2021) sont illustrées dans le tableau des tâches suivant.

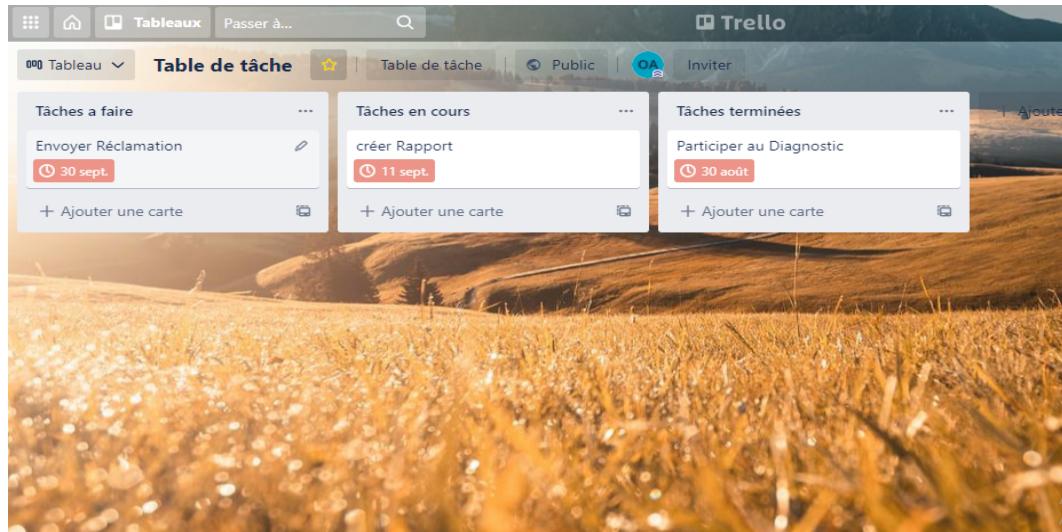


FIGURE 5.2 – Planification de sprint 3

5.4 Phase d'analyse

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

a) Définition des acteurs et des fonctionnalités de sprint 3

Le principal acteur de ce sprint est :

- Agriculteur : c'est l'acteur qui a pour rôle de participer au diagnostic et envoyer réclamation
- Ingénieur : c'est l'acteur qui a pour rôle de créer un rapport (réponse au diagnostic) pour l'agriculteur.

Le diagramme de cas d'utilisation global de sprint 3 est montré dans la figure ci-dessous.

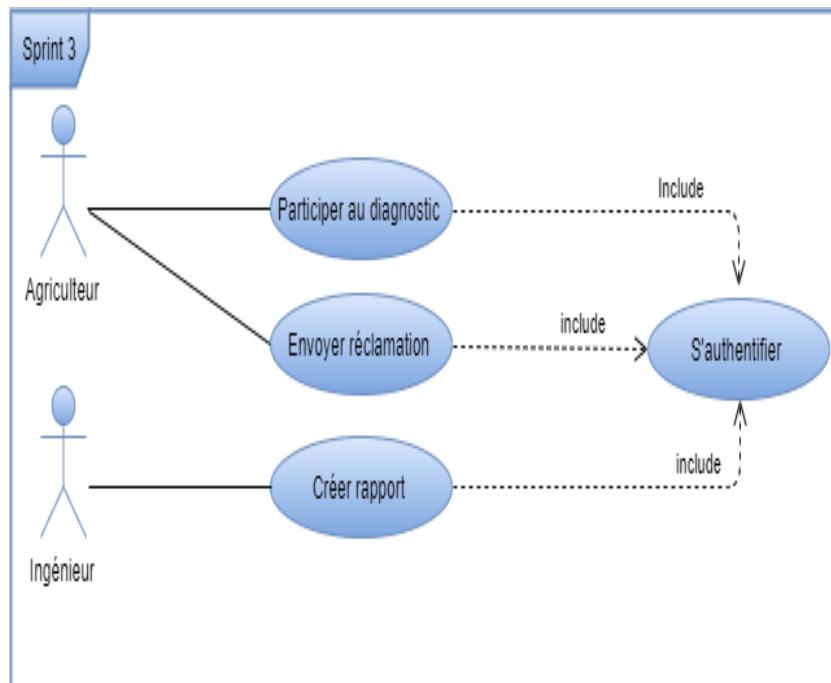


FIGURE 5.3 – Diagramme de cas d'utilisation de sprint 3

b) Raffinement de cas d'utilisation « Participer au diagnostic »

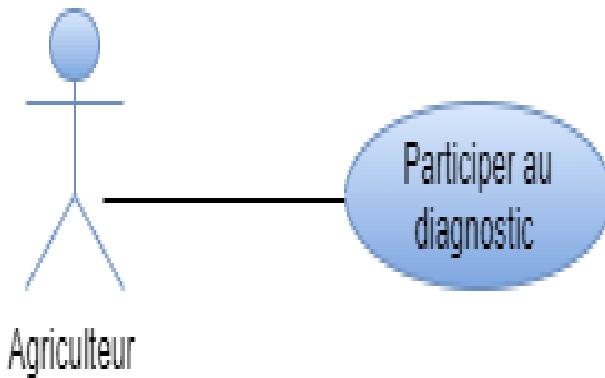


FIGURE 5.4 – Raffinement de cas d'utilisation « Participer au diagnostic»

★ Description textuelle de cas d'utilisation "participer au diagnostic"

Titre	Participer au diagnostic
Acteur	Agriculteur
Objectif	L'agriculteur prendre une photo et faire un diagnostic pour connaitre la maladie la plus proche selon les questions proposer. .
Pré-condition	<ul style="list-style-type: none"> • L'agriculteur doit être authentifié. • L'agriculteur doit accéder à la page « diagnostic »
Post-condition	L'agriculteur reçoit un rapport de la maladie selon ses réponses.
Scénario Principal	<ol style="list-style-type: none"> 1- L'agriculteur demande la page de diagnostic 2- Le système lui affiche la page demandée 3- L'agriculteur remplit le formulaire demandé : Sélectionne une plante de la liste des plantes, prendre une photo sur de la plante puis répondre aux questions posées et puis on clique sur le bouton « suivant» 4- Le système envoie une requête pour les données et les réponses introduites par l'agriculteur 5- Le système affiche un rapport qui contient le résultat du diagnostic introduit par l'ingénieur
Scénario alternatif et d'exception :	<ol style="list-style-type: none"> 1- L'agriculteur quitte la page de participer au diagnostic : <ul style="list-style-type: none"> • Le cas d'utilisation se termine avec échec. 2- Les données saisies sont incorrectes ou manquantes : <ul style="list-style-type: none"> • Manque de répond aux questions 3- Mauvaise connexion avec la base de données : • Se termine avec échec

TABLE 5.2 – Description textuelle de cas d'utilisation "participer au diagnostic"

★ Raffinement de cas d'utilisation «Créer rapport »

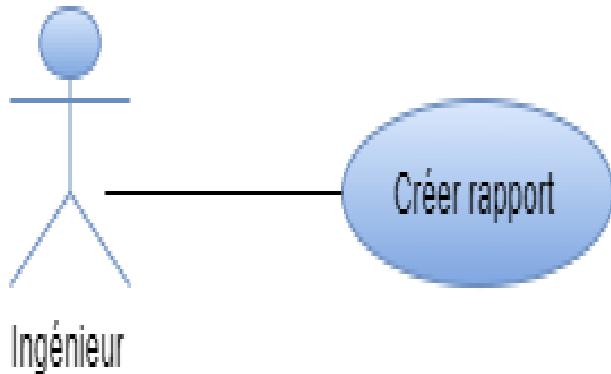


FIGURE 5.5 – Raffinement de cas d'utilisation « Créer rapport »

★ Description textuelle de cas d'utilisation "Créer rapport "

Titre	Créer rapport
Acteur	Ingénieur
Objectif	L'ingénieur doit créer un rapport du diagnostic pour l'agriculteur connaitre la maladie
Pré-condition	L'ingénieur doit être authentifié.
Post-condition	L'ingénieur renvoi un rapport de la maladie à l'agriculteur
Scénario Principal	1- L'ingénieur demande la page de création du rapport 2- Le système lui affiche la page demandée 3- L'ingénieur remplit le formulaire du rapport (nom de maladie, cause, et le traitement qui possible à utiliser) selon la requête (des données et des réponses introduites par l'agriculteur), puis cliquer sur le bouton « envoyé ». 4- Le système renvoie le rapport vers l'agriculteur.
Scénario alternatif et d'exception :	1- Les données saisies par l'agriculteur sont incorrectes ou manquantes : • Manque de connaitre la maladie 2- Mauvaise connexion avec la base de données : • Se termine avec échec

TABLE 5.3 – Description textuelle de cas d'utilisation "Créer rapport"

★ Raffinement de cas d'utilisation « Envoyer réclamation »

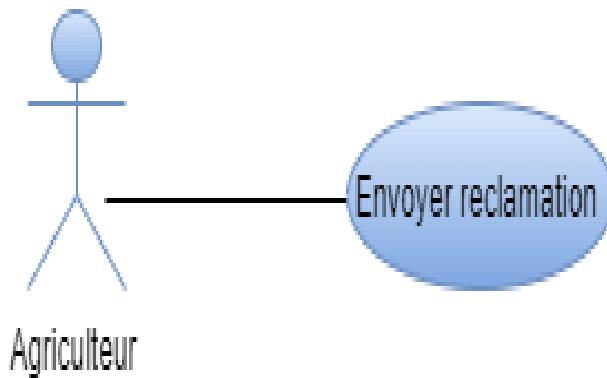


FIGURE 5.6 – Raffinement de cas d'utilisation « Envoyer réclamation »

★ Description textuelle de cas d'utilisation " Envoyer réclamation "

Titre	Envoyer réclamation
Acteur	Agriculteur
Objectif	L'agriculteur a l'accès d'envoyer une réclamation lorsqu'on a un problème.
Pré-condition	L'agriculteur doit être authentifié.
Post-condition	L'agriculteur envoi une réclamation
Scénario Principal	1- L'agriculteur demande la page d'envoyer réclamation 2- Le système affiche la page demandée 3- L'agriculteur remplit les champs et clique sur envoyer 4- Le système enregistre les données dans la base et envoi un message de succès
Scénario alternatif et d'exception :	1- L'agriculteur quitte la page de réclamation : • Le cas d'utilisation se termine avec échec. 2- Mauvaise connexion avec la base de données : • Se termine avec échec

TABLE 5.4 – Description textuelle de cas d'utilisation "Envoyer réclamation "

5.5 Phase de conception

Au niveau de cette section, nous présentons le diagramme de classe, les diagrammes de séquences ainsi que les diagrammes d'activités de ce sprint.

5.5.1 Conception statique de sprint

a) Diagramme de classe de sprint 3

La figure ci-dessous représente le diagramme de classes utilisé pour le développement du sprint 3

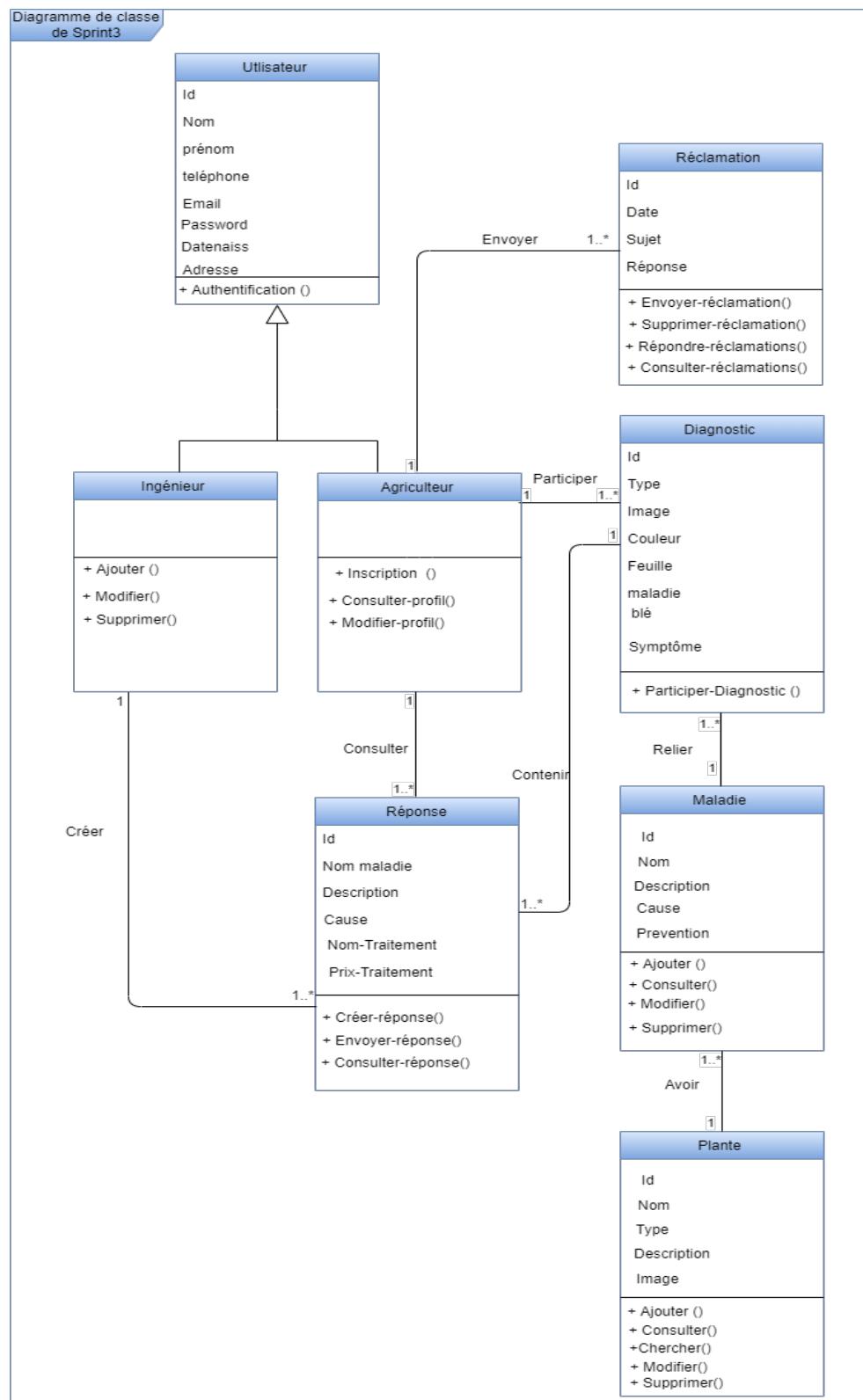


FIGURE 5.7 – Diagramme de classe de sprint 3

b) Dictionnaire de données :

Le dictionnaire de base de données est une définition des attributs de chaque classe de sprint trois.

Dictionnaire de données de la table Utilisateur

Utilisateur				
Attribut	Type	Taille	Obligatoire	Contrainte
Id	Int	50	Oui	Clé primaire
Nom	VARCHAR	20	Oui	
Prénom	VARCHAR	20	Oui	
Téléphone	Int	8	Oui	
Email	VARCHAR	30	Oui	

TABLE 5.5 – Dictionnaire de données de la table Utilisateur

Dictionnaire de données de la table Agriculteur

Agriculteur				
Attribut	Type	Taille	Obligatoire	Contrainte
Id	Int	50	Oui	Clé primaire
Nom	VARCHAR	20	Oui	
Prénom	VARCHAR	20	Oui	
Téléphone	Int	8	Oui	
Email	VARCHAR	30	Oui	
Password	varchar	50	Oui	
Datenaiss	VARCHAR	50	Oui	
Adresse	VARCHAR	50	Oui	

TABLE 5.6 – dictionnaire de données de la table Agriculteur

Dictionnaire de données de la table Ingénieur

Ingénieur				
Attribut	Type	Taille	Obligatoire	Contrainte
Id	Int	50	Oui	Clé primaire
Nom	VARCHAR	20	Oui	
Prénom	VARCHAR	20	Oui	
Téléphone	Int	8	Oui	
Email	VARCHAR	30	Oui	
Password	varchar	50	Oui	
Datenaiss	VARCHAR	50	Oui	
Adresse	VARCHAR	50	Oui	

TABLE 5.7 – dictionnaire de données de la table Ingénieur

Dictionnaire de données de la table Diagnostic

Diagnostic				
Attribut	Type	Taille	Obligatoire	Contrainte
Id	Int	50	Oui	Clé primaire
Type	Varchar		Oui	
Image	Varchar		Oui	
Couleur	Varchar		Oui	
Feuille	Varchar		Oui	
Nom maladie	Varcahar		Oui	
ble	Varcahar		Oui	
Symptôme	Varcahar		Oui	

TABLE 5.8 – dictionnaire de données de la table Diagnostic

Dictionnaire de données de la table Réponse

Réponse				
Attribut	Type	Taille	Obligatoire	Contrainte
Id	Int	50	Oui	Clé primaire
Nom maladie	Varchar	50	Oui	
Description	Varchar		Oui	
Cause	Varchar		Oui	
Nom-traitement	varchar	50	Oui	
Prix-traitement	double		Oui	

TABLE 5.9 – dictionnaire de données de la table Réponse

Dictionnaire de données de la table Réclamation

Réclamation				
Attribut	Type	Taille	Obligatoire	Contrainte
Id	Int	50	Oui	Clé primaire
Date	Date		Oui	
Sujet	Varchar		Oui	
Réponse	Varchar		Oui	

TABLE 5.10 – dictionnaire de données de la table Réclamation

5.6 Conception dynamique du troisième sprint

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

5.6.1 Les diagrammes de séquence

★ Diagramme de séquence de cas d'utilisation « participer au diagnostic »

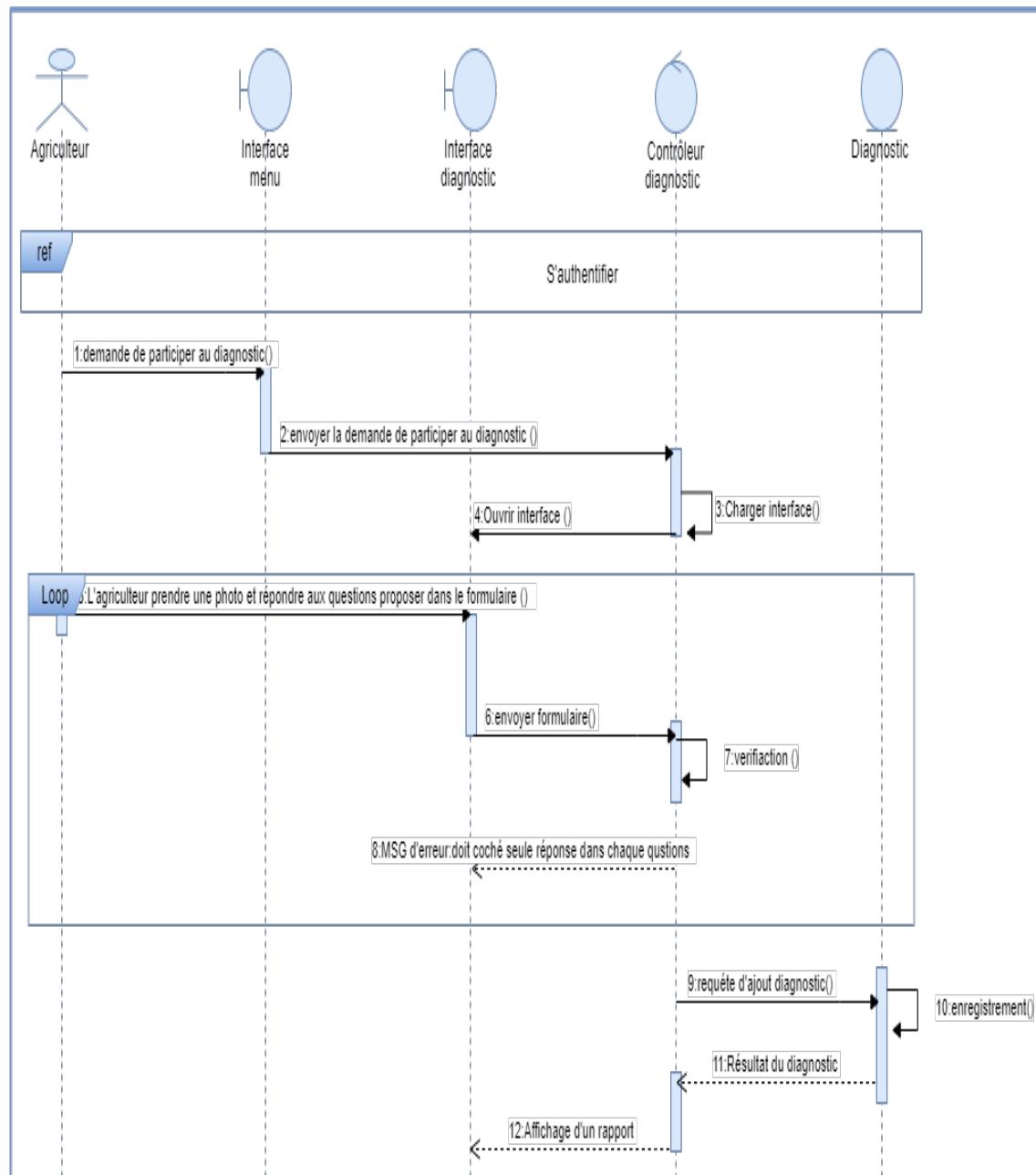


FIGURE 5.8 – Diagramme de séquence de cas d'utilisation « Participer au diagnostic»

★ Diagramme de séquence de cas d'utilisation «Créer rapport»

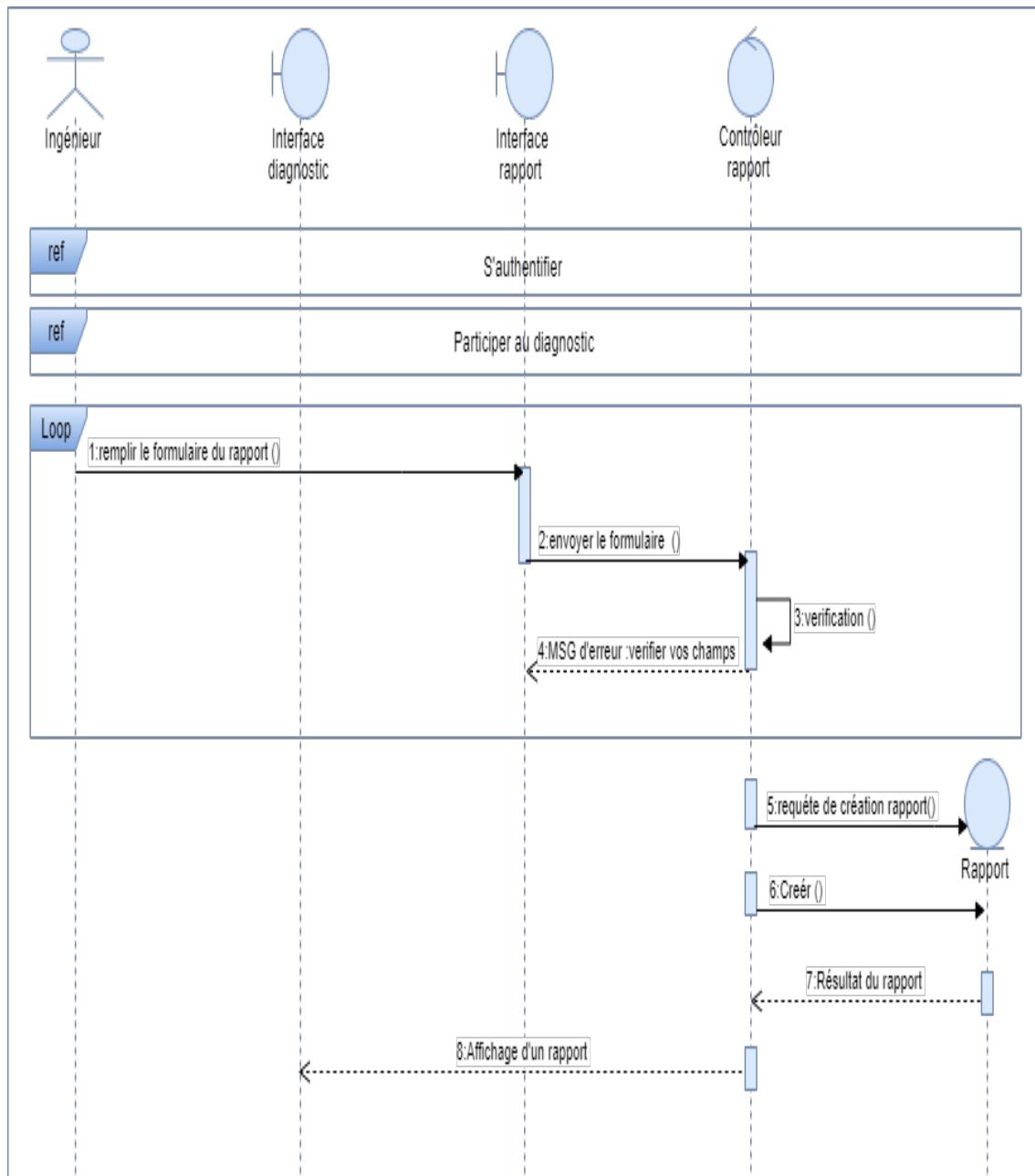


FIGURE 5.9 – Diagramme de séquence de cas d'utilisation « Crer rapport »

★ Diagramme de séquence de cas d'utilisation « Envoyer rclamation »

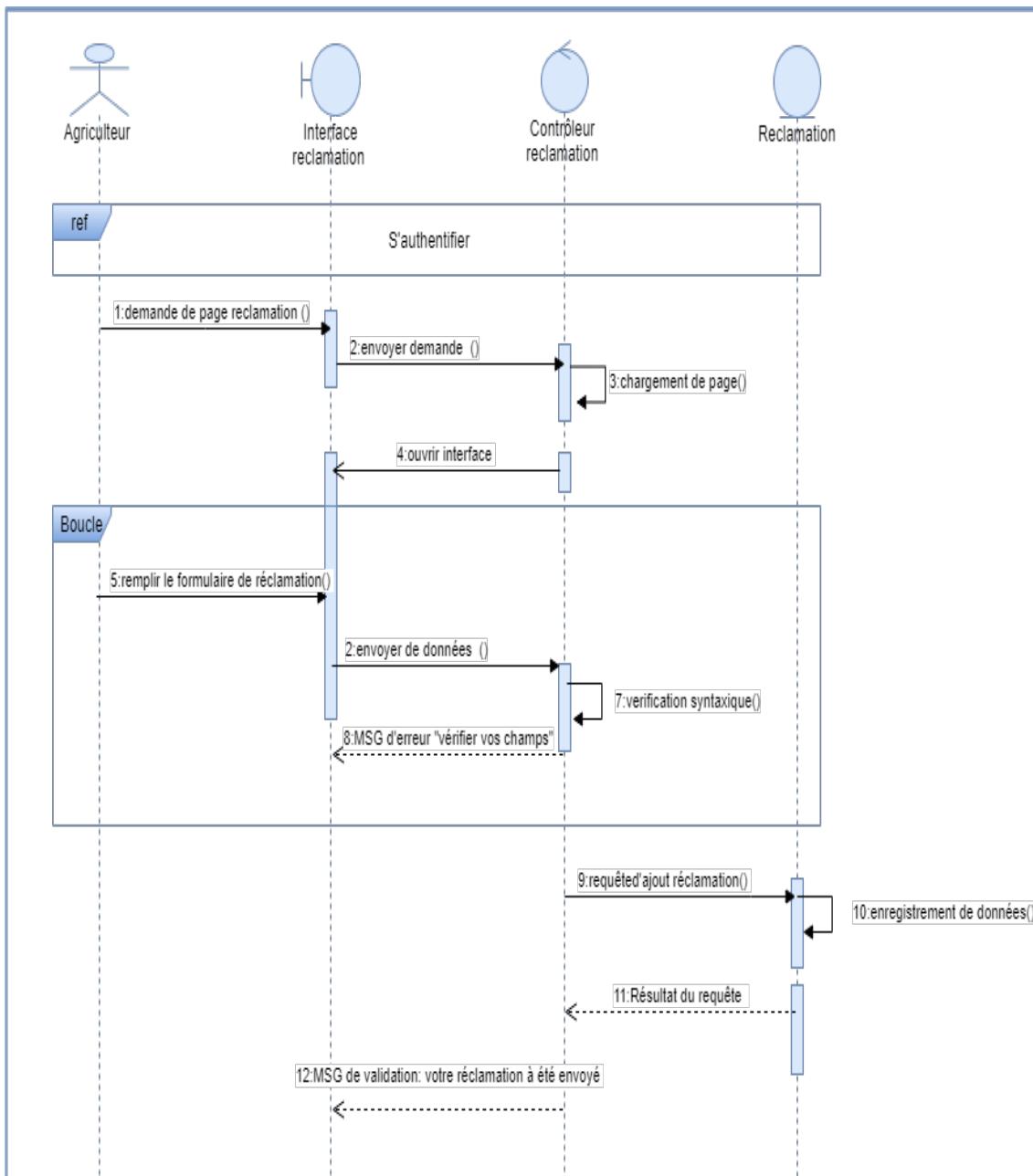


FIGURE 5.10 – Diagramme de séquence de cas d'utilisation « Envoyer réclamation»

5.7 Diagrammes d'activités :

★ Diagramme d'activité du cas d'utilisation "Participer au diagnostic"

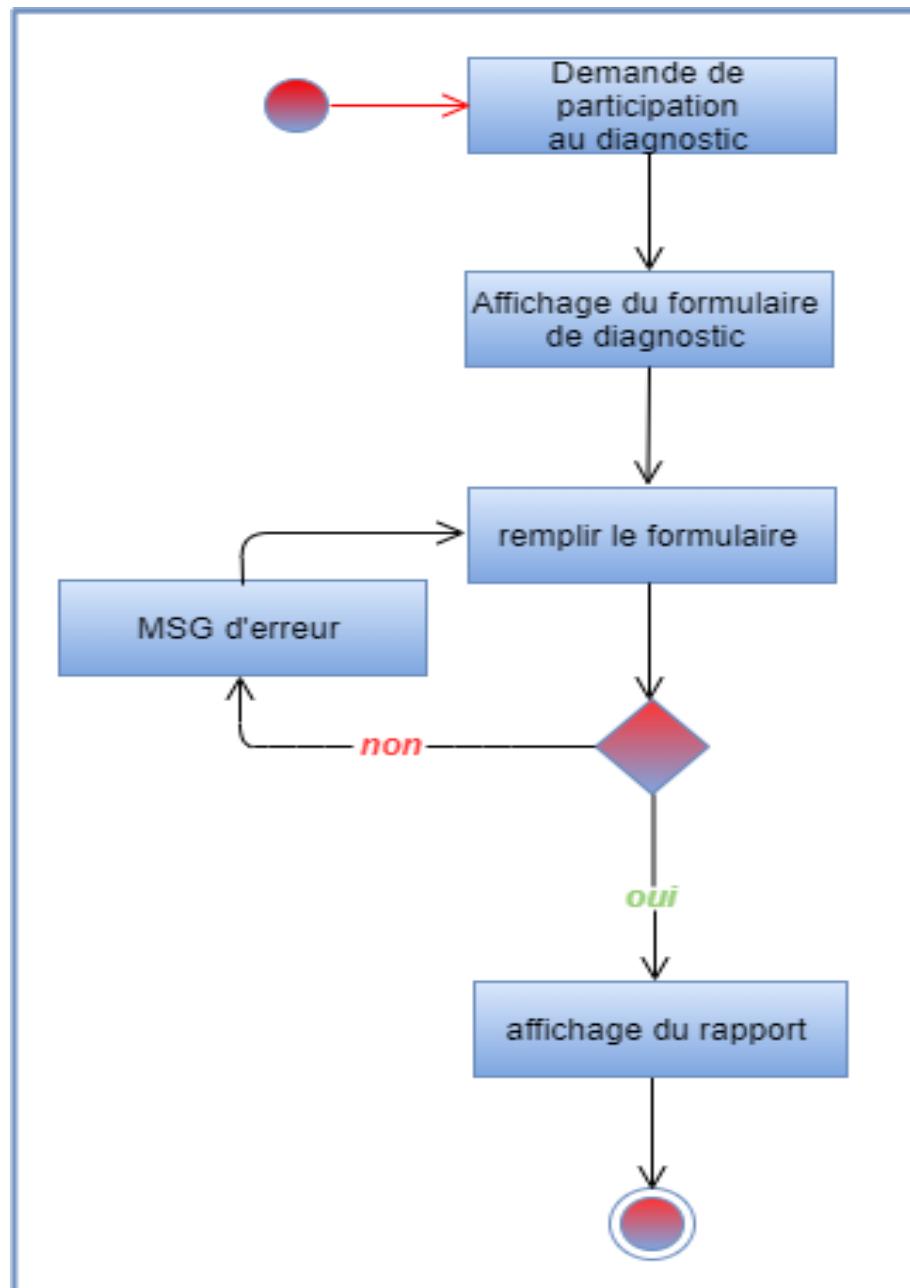


FIGURE 5.11 – Diagramme d'activité de cas d'utilisation « Participer au diagnostic»

★ Diagramme d'activité du cas d'utilisation "Envoyer réclamation"

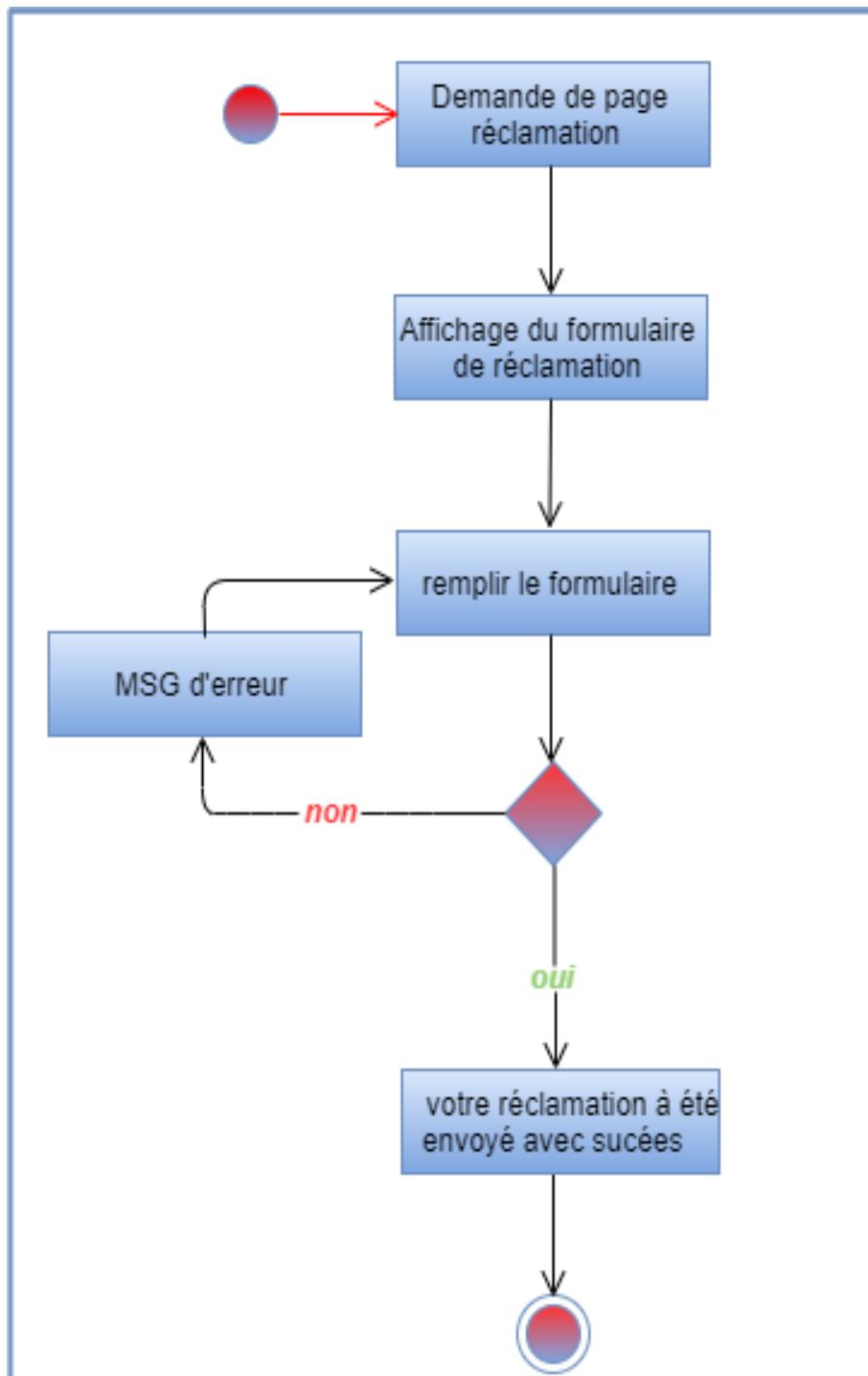


FIGURE 5.12 – Diagramme d’activité de cas d’utilisation « Envoyer réclamation »

5.8 Réalisation et tests :

Cette partie est consacrée à l'exposition du travail achevé à travers des captures d'écrans de différentes interfaces développées pendant ce sprint 3.

5.8.1 Les Interfaces :

5.8.1.1 Les interfaces mobile

La figure ci-dessous présente l'interface de demande diagnostic.

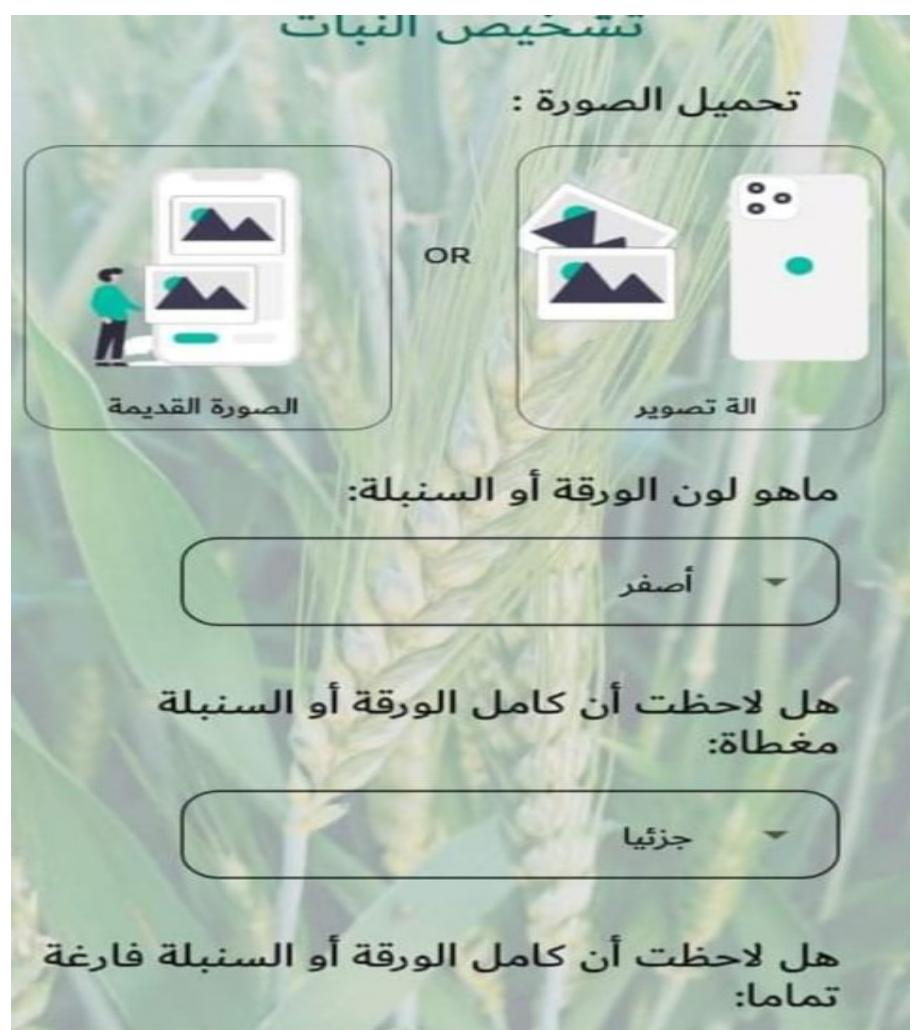


FIGURE 5.13 – Interface Demande aux diagnostic

La figure ci-dessous présente l'interface de Envoyer réclamation



FIGURE 5.14 – Interface envoyer réclamation

5.8.2 Les tests :

- i. Le IHM, l'utilisateur doit remplir un formulaire pour qu'il communique avec son SGBD :
 - Participer au diagnostic

Le tableau ci-dessous présente la fiche test fonctionnel de l'interface Diagnostic :

ID Participer au diagnostic
Type : Fonctionnel
Testeur : Asma Ouertani
Object attendu : l'agriculteur doit prendre une photo de son plante et répond aux les questions affichées pour consulte son rapport de diagnostic
Description : Après le développement de l'interface de participer au diagnostic on va répondre aux les questions dans le formulaire et on va faire des tests. (Test sur chaque champ)
Règles métier Règle 1 : l'agriculteur accède à l'interface diagnostic Règle 2 : l'agriculteur doit prendre une photo de son plante et remplir le formulaire de diagnostic Règle 3 : l'agriculteur doit respecter les contrôles de saisis
Tests d'acceptation sur l'entité : Diagnostic
Nom de scénario 1 : Contrôle syntaxique des champs lorsque l'agriculteur remplit le formulaire il doit respecter les champs suivants : <ul style="list-style-type: none">✓ Choisir une plante : l'agriculteur doit sélectionner une plante qui affiche dans la liste✓ Prendre une photo de son plante✓ Répondre à la question : l'agriculteur doit répondre à toutes les questions dans le formulaire.

5.9 Conclusion :

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

Chapitre 6

Mise en œuvre du sprint 4

6.1 Introduction

Ce chapitre présente les détails du quatrième Sprint. Nous commençons par la planification du Sprint en précisant les cas d'utilisations associés. Puis, Nous passons par une phase d'analyse, ensuite nous présenterons quelques interfaces réalisées au cours de ce sprint. Nous détaillons chaque cas en présentant les diagrammes d'analyse ,séquences, et la réalisation.

Cette étape est réalisée pour présenter le sprint 4 ainsi les cas d'utilisations qu'elle compose.

Ce schéma montre la planification de sprint 4 :

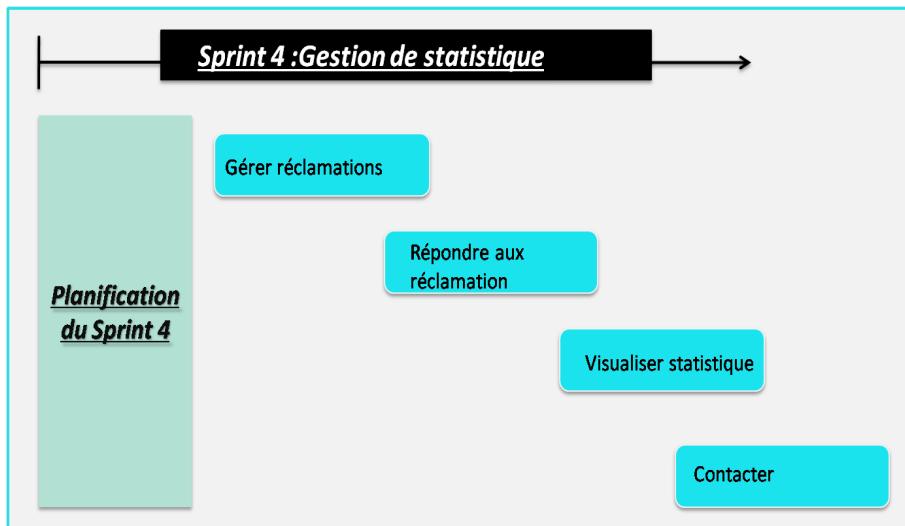


FIGURE 6.1 – Planification du sprints4

6.1.1 Backlog de sprint

En partant du même principe que le sprint précédent, nous commençons par définir le but de notre quatrième sprint. Le tableau6.1 décrit le Backlog du sprint.

ID	Histoire	Date début	Date fin	Responsable
1	Répondre aux réclamations	06/10/2021	10/10/2021	Ouertani Asma
2	Gérer réclamations	01/10/2021	05/10/2021	
3	Visualiser statistique	12/10/2021	20/10/2021	
3	Contacter	21/10/2021	30/10/2021	

TABLE 6.1 – Backlog de sprint4

6.2 Tableau des tâches

Les tâches relatives au troisième sprint (A T= 10/08/2021) sont illustrées dans le tableau des tâches suivant.

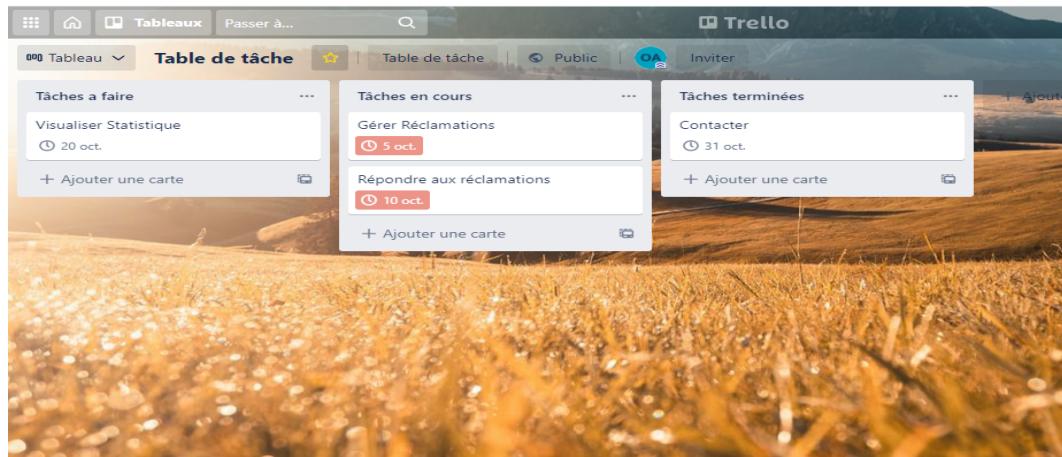


FIGURE 6.2 – planification de sprint 4

6.3 Phase d'analyse

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

a) Définition des acteurs et des fonctionnalités de sprint 4

Le principal acteur de ce sprint est :

- Ingénieur : c'est l'acteur à pour rôle de répondre aux réclamations de l'agriculteur.
- Administrateur : c'est l'acteur qui a le rôle de gérer les réclamations et de visualiser les statistiques de notre application.
- Agriculteur : c'est l'acteur à pour rôle de contacter l'administrateur.

Le diagramme de cas d'utilisation global de sprint 4 est montré dans la figure ci-dessous.

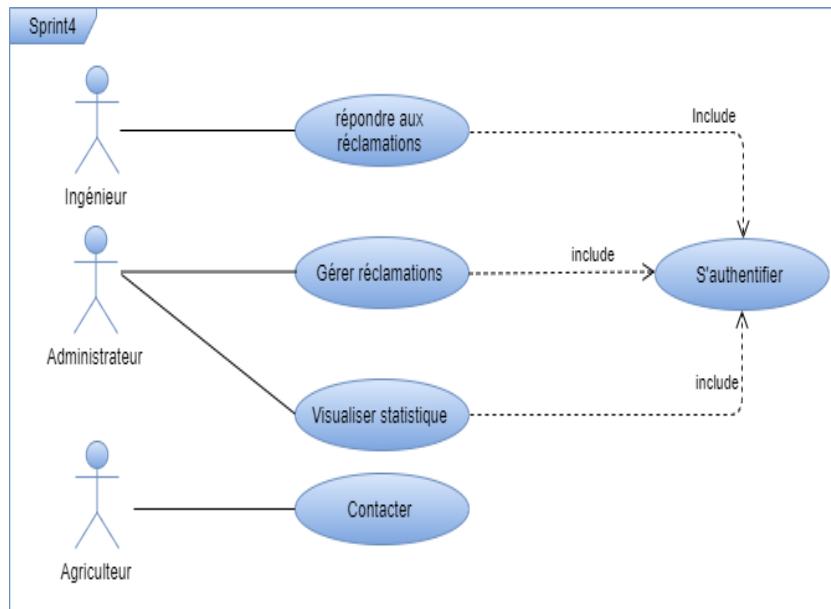


FIGURE 6.3 – Diagramme de cas d'utilisation de sprint 4

b) Raffinement des cas d'utilisation de sprint 4

★ **Raffinement de cas d'utilisation «Répondre aux réclamations»**

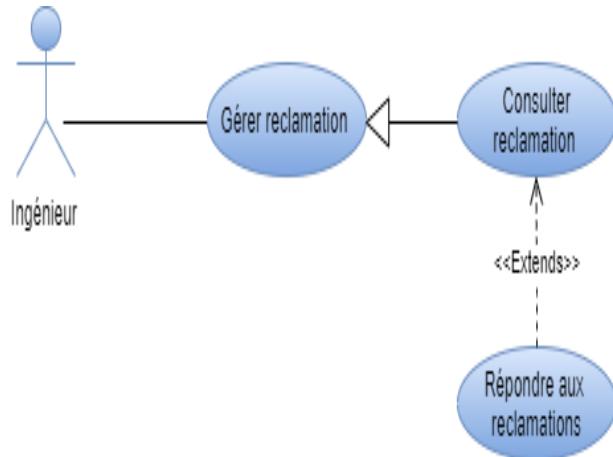


FIGURE 6.4 – Raffinement de cas d'utilisation « Répondre aux réclamations »

★ **Description textuelle de cas d'utilisation "Répondre aux réclamations"**

Titre	Répondre aux réclamations
Acteur	Ingénieur
Objectif	L'ingénieur répond les réclamations
Pré condition	L'ingénieur doit être authentifié.
Post condition	L'ingénieur répond aux réclamations des agriculteurs .
Scénario Principal	1- L'ingénieur demande de consulte la liste des réclamations 2- Le système affiche la liste des réclamations 3- L'ingénieur choisit une réclamation et clique sur répondre 4- Le système affiche la zone de réponse 5- L'ingénieur remplit cette zone et clique sur "envoyer" 6- Le système enregistre le réponse et le renvoyer au agriculteur
Scénario alternatif et d'exception :	1- L'ingénieur quitte la page de réclamation : • Le cas d'utilisation se termine avec échec. 2- Mauvaise connexion avec la base de données : • Se termine avec échec

TABLE 6.2 – Description textuelle de cas d'utilisation "Répondre aux réclamations "

★ Raffinement de cas d'utilisation « Gérer réclamation »

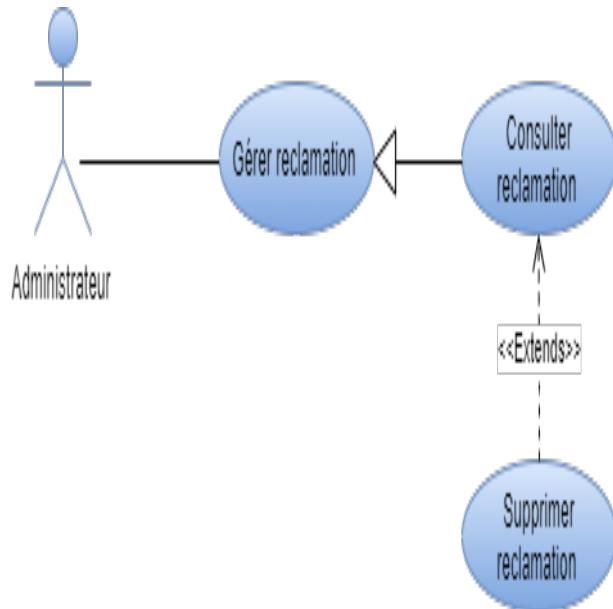


FIGURE 6.5 – Raffinement de cas d'utilisation « Gérer réclamation»

★ Description textuelle de cas d'utilisation " Gérer réclamations "

Titre	Gérer réclamations
Acteur	Administrateur
Objectif	L'administrateur à l'accès de consulte les réclamations
Pré-condition	L'administrateur doit être authentifié.
Post condition	L'administrateur consulte les réclamations des agriculteurs
Scénario Principal	<p>1- L'administrateur demande de consultes la liste des réclamations</p> <p>2- Le système consulte la liste des réclamations. Dans se cas d'utilisation l'administrateur peut supprimer les réclamations</p> <p>3- L'administrateur choisit une réclamation de la liste affichée (liste des réclamations) et clique sur le bouton « supprimer »</p> <p>4- Le système supprime la réclamation de la base de données .</p>
Scénario alternatif et d'exception :	<p>1- L'administrateur quitte la page de réclamation</p> <ul style="list-style-type: none"> • Le cas d'utilisation se termine avec échec. <p>2- Mauvaise connexion avec la base de données :</p> <ul style="list-style-type: none"> • Se termine avec échec

TABLE 6.3 – Description textuelle de cas d'utilisation "Gérer réclamations "

★ Raffinement de cas d'utilisation « Visualiser statistique »

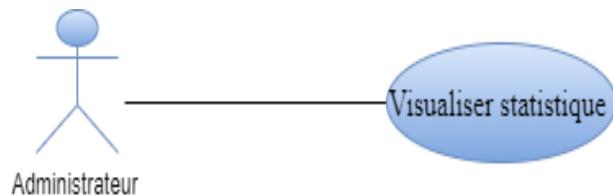


FIGURE 6.6 – Raffinement de cas d'utilisation « Visualiser statistique»

★ Description textuelle de cas d'utilisation " Visualiser statistique "

Titre	Visualiser statistique
Acteur	Administrateur
Objectif	L'administrateur visualise les statistiques de la maladie
Pré-condition	L'administrateur doit être authentifié.
Post-condition	L'administrateur accède à l'interface de visualisation.
Scénario Principal	<p>1- L'administrateur visualiser statistique dans la page d'accueil</p> <p>2- Le système envoie la requête à la base de données</p> <p>3- Le Système sélection la maladie les plus popularisé du SGBD</p> <p>4- Le système ouvre la page et affiche les statistiques avec les noms les plus popularisées de la maladie sous forme d'un histogramme.</p>

TABLE 6.4 – Description textuelle de cas d'utilisation "Visualiser statistique "

★ Raffinement de cas d'utilisation « Contacter »



FIGURE 6.7 – Raffinement de cas d'utilisation « Contacter »

★ Description textuelle de cas d'utilisation "Contacter "

Titre	Contacter
Acteur	Agriculteur
Objectif	Envoyer un message à l'administration
Précondition	L'agriculteur doit être à L'application .
Postcondition	Message envoyé.
Scénario Principal	1- L'agriculteur demande l'interface de contact 2- Le système affiche l'interface de contact 3- L'agriculteur remplit son message 4- L'agriculteur envoie un message 5- Le système affiche "message envoyé"

TABLE 6.5 – Description textuelle de cas d'utilisation "Contacter "

6.4 Phase de conception

Au niveau de cette section, nous présentons le diagramme de classe, les diagrammes de séquences ainsi que les diagrammes d'activités de ce sprint..

6.4.1 Conception statique de sprint

a) Diagramme de classe de sprint 4

La figure ci-dessous représente le diagramme de classes utilisé pour le développement du sprint 4

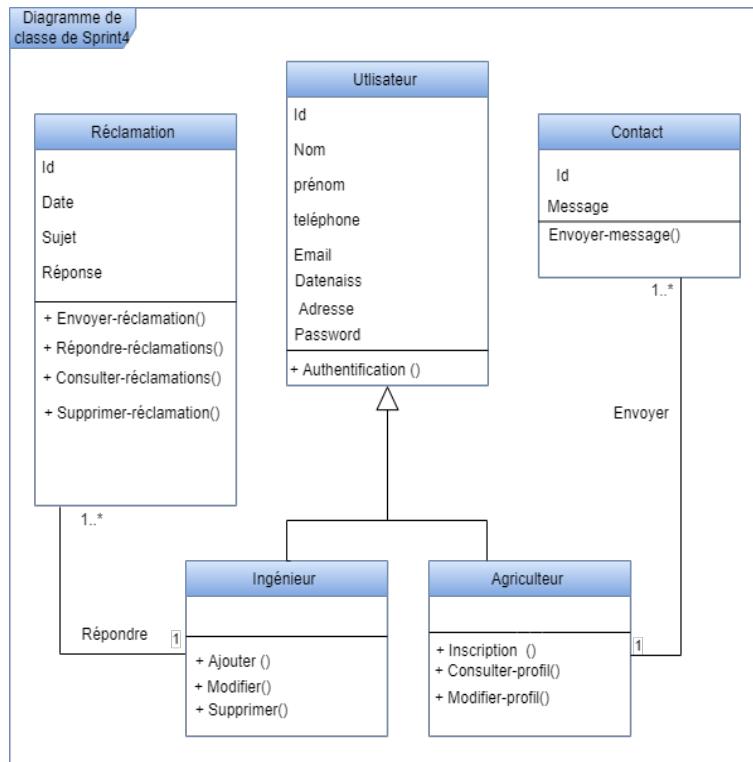


FIGURE 6.8 – Diagramme de classe de sprint4

b) Dictionnaire de données :

Le dictionnaire de base de données est une définition des attributs de chaque classe du quatrième sprint.

Dictionnaire de données de la table Ingénieur

Ingénieur				
Attribut	Type	Taille	Obligatoire	Contrainte
Id	Int	50	Oui	Clé primaire
Nom	Varchar	20	Oui	
Prénom	Varchar	20	Oui	
Téléphone	Int	8	Oui	
Email	Varchar	30	Oui	

TABLE 6.6 – dictionnaire de données de la table Ingénieur

Dictionnaire de données de la table Réclamation

Réclamation				
Attribut	Type	Taille	Obligatoire	Contrainte
Id	Int	50	Oui	Clé primaire
Date	Date		Oui	
Sujet	Varchar		Oui	
Réponse	Varchar		Oui	

TABLE 6.7 – dictionnaire de données de la table Réclamation

Dictionnaire de données de la table Contact

Contact				
Attribut	Type	Taille	Obligatoire	Contrainte
Id	Int	50	Oui	Clé primaire
message	String		non	

TABLE 6.8 – dictionnaire de données de la table contact

6.5 Conception dynamique du quatrième sprint

6.5.1 Les diagrammes de séquence

★ Diagramme de séquence de cas d'utilisation «Répondre aux réclamations»

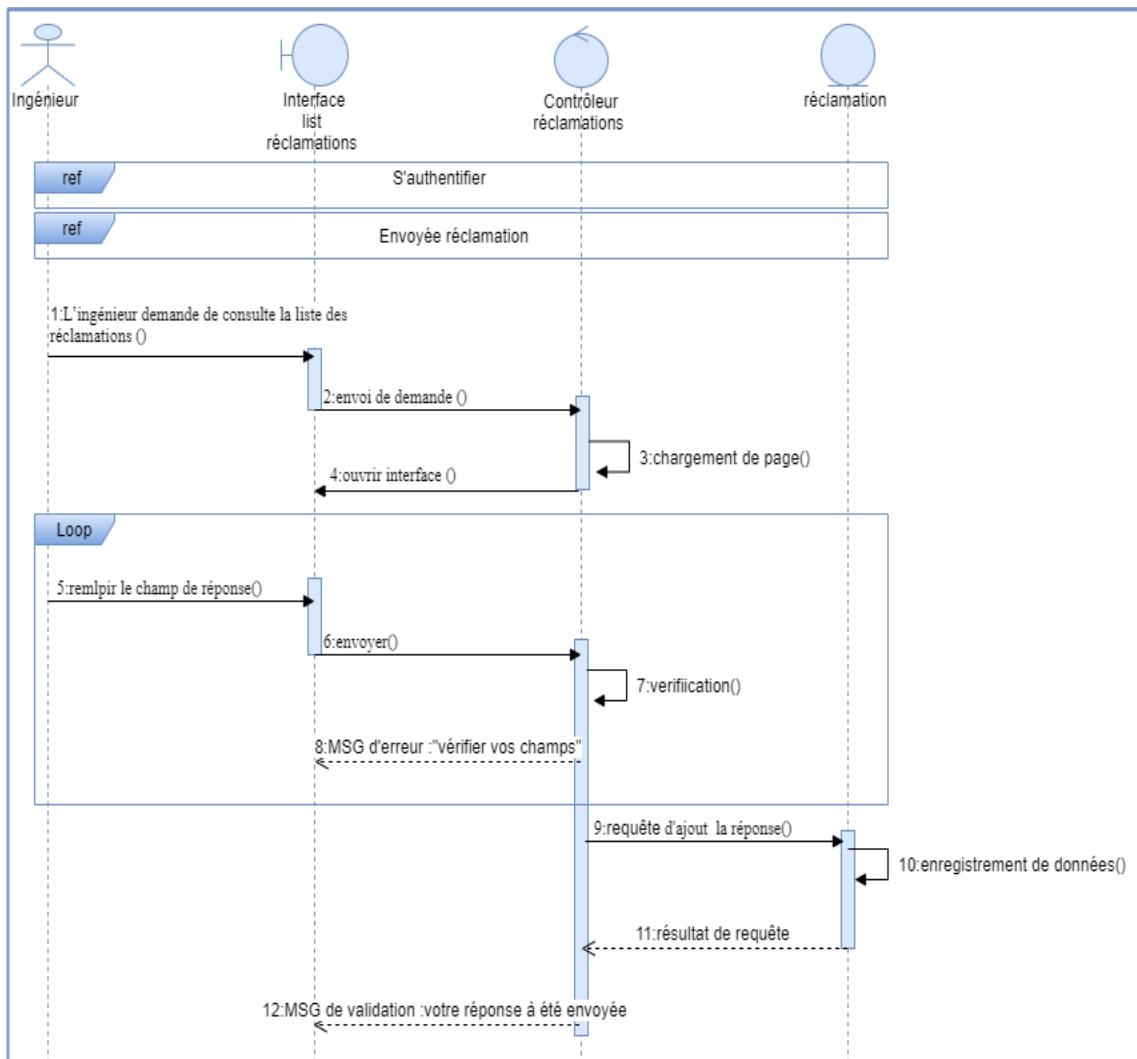


FIGURE 6.9 – Diagramme de séquence de cas d'utilisation «Répondre aux réclamations»

★ Diagramme de séquence de cas d'utilisation «Visualiser statistique»

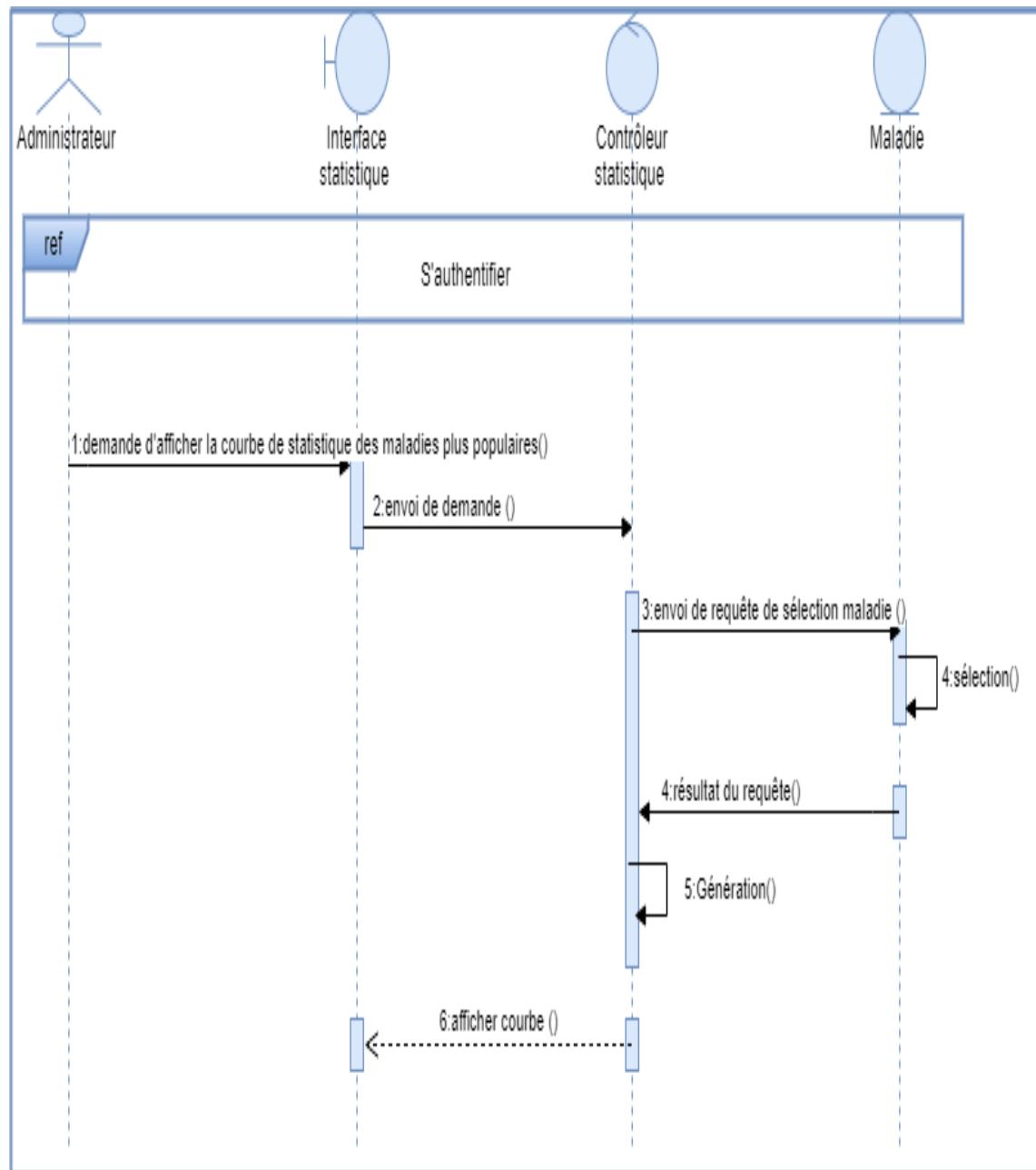


FIGURE 6.10 – Diagramme de séquence de cas d'utilisation «Visualiser statistique»

★ Diagramme de séquence de cas d'utilisation «Contacter»

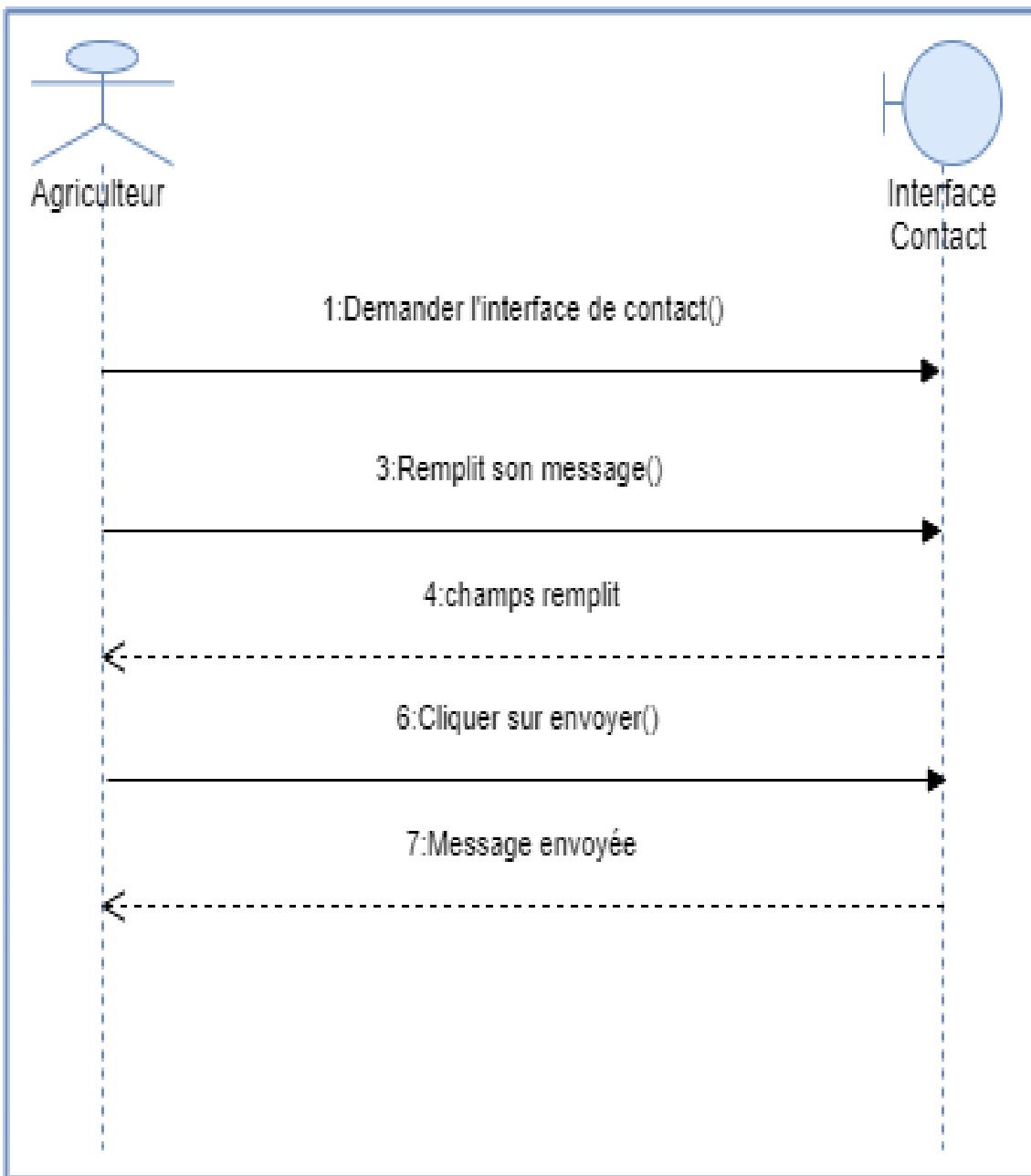


FIGURE 6.11 – Diagramme de séquence de cas d'utilisation « Contacter »

6.6 Diagramme de classe global

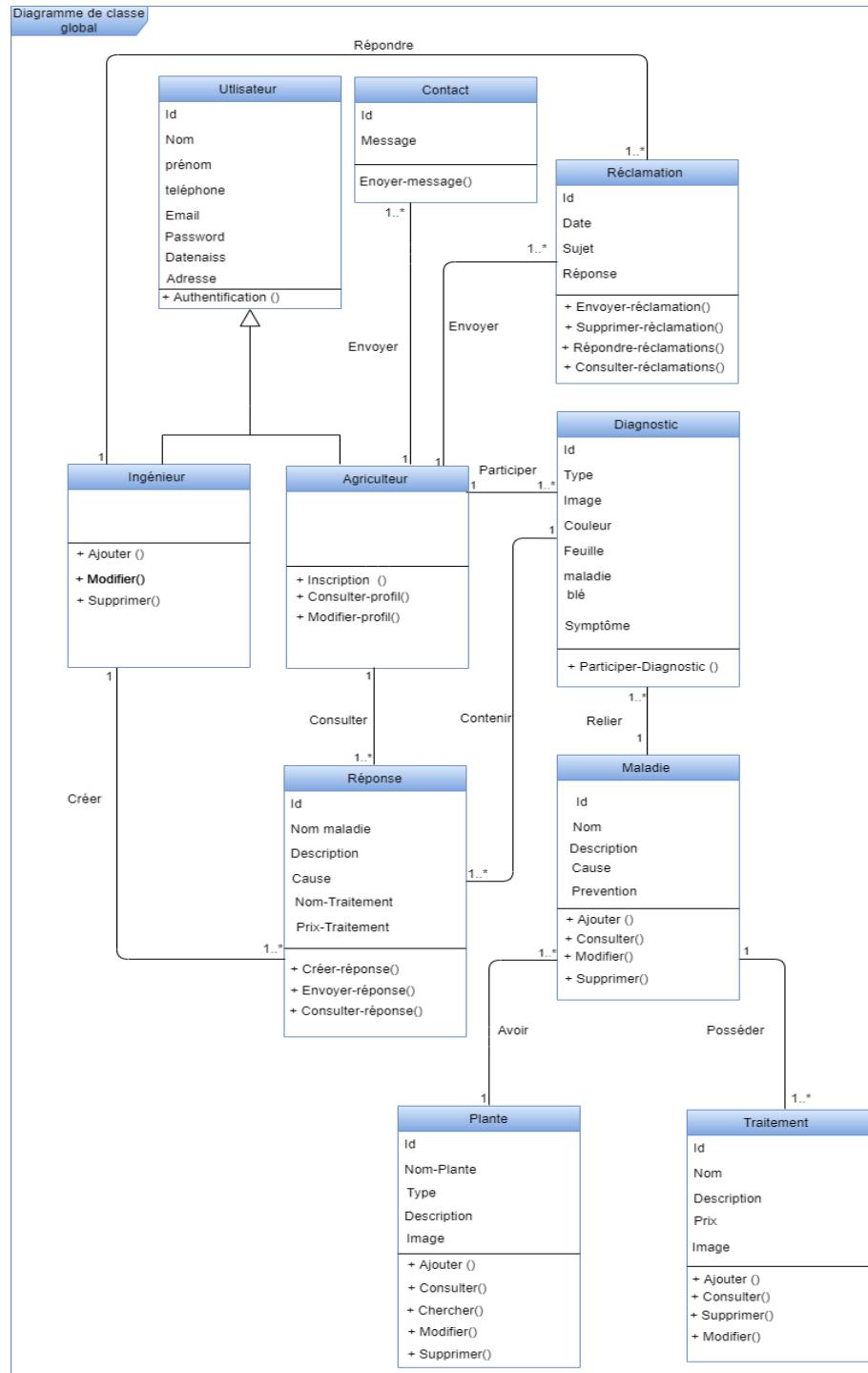


FIGURE 6.12 – Diagramme de classe global

6.7 Réalisation et tests :

Dans cette partie, on va présenter des imprimés écrans de quelques interfaces réalisées au cours ce dernier sprint.

6.7.1 Les Interfaces :

6.7.2 Les Interfaces mobiles

La figure ci-dessous présente l'interface de consultation des réclamations



FIGURE 6.13 – Interface consulter réclamation

La figure ci-dessous présente l'interface de contact



FIGURE 6.14 – Interface consulter réclamation

6.7.3 Les Interfaces web

La figure ci-dessous présente l'interface de consultation des réclamations

CHAPITRE 6. MISE EN ŒUVRE DU SPRINT 4

Date	Sujet	Réponce	Action
2022/1/4	و ماهي (P205) كيفية استعمال الفسفرور وضائقة	empty	
2022/1/4	عذري حسب الظروف الجوية و عدد الجرائم المرجوحة و نسبة الأصداف الحسلي		
2022/1/4	و ماهي (P205) كيفية استعمال الفسفرور وضائقة	empty	
2022/1/4	و ماهي (P205) كيفية استعمال الفسفرور وضائقة	empty	
2022/1/4	متى الوقت المناسب لإستعمال البوتاسيوم	empty	
2022/1/4	لا يوجد علاج لمرض التحتم السائب بعد ظهوره ، اللهم إلا بطريقه الجمع اليدوي للدمابل المصابة ، ووضعها مباشرة داخل أكياس مغلقة		
2022/1/4	متى الوقت المناسب لإستعمال البوتاسيوم	empty	
2022/1/4	متى الوقت المناسب لإستعمال البوتاسيوم	empty	
		Activer Windows	

FIGURE 6.15 – Interface consulter réclamation

La figure ci-dessous présente l'interface de suppression d'une réclamation

Date	Sujet	Réponce	Action
2022/1/4	عن نقص الفسفرور	عذري حسب الظروف الجوية و عدد الج	
2022/1/4	(P205) كيفية استعمال الفسفرور	empty	
2022/1/4	(P205) كيفية استعمال الفسفرور	empty	
2022/1/4	للمتاسب لإستعمال البوتاسيوم	Réclamation bien supprimer	
2022/1/4	لمرض التحتم	Success!	
2022/1/4	متى الوقت المناسب لإستعمال البوتاسيوم	empty	
2022/1/4	متى الوقت المناسب لإستعمال البوتاسيوم	empty	
		Activer Windows	
		Accéder aux paramètres po	
		1-5 of 5 < >	

FIGURE 6.16 – Interface supprimer réclamation

6.7.4 Les tests :

- i. Le IHM, l'utilisateur doit remplir un formulaire pour qu'il communique avec son SGBD :
 - Répondre aux réclamations

Le tableau ci-dessous présente le fiche test fonctionnel de répondre aux réclamations :

ID Répondre aux réclamations
Type : Fonctionnel
Testeur : Asma Ouertani
Object attendu : ingénieur doit consulter les réclamations pour remplir le formulaire de réponse.
Description : Après le développement de l'interface Répondre aux réclamations on va introduire la réponse dans le formulaire et on va faire le test. (Test sur la syntaxe de champs de réponse)
Règles métier Règle 1 : l'ingénieur accède à l'interface Répondre aux réclamations Règle 2 : l'ingénieur doit remplir le formulaire de réponse de répondre aux réclamations (remplir les champs réponse) Règle 3 : l'ingénieur doit respecter les contrôles de saisies
Tests d'acceptation sur l'entité : Réclamation Nom de scénario 1 : contrôle syntaxique des champs lorsque l'agriculteur remplit le formulaire il doit respecter les champs suivants : <ul style="list-style-type: none">✓ Ajouter une réponse : l'ingénieur doit écrire une réponse dans une zone de texte Nom de scénario 1 : l'utilisateur doit compléter le processus de Répondre aux réclamations : <ul style="list-style-type: none">✓ Ajouter une réponse : l'ingénieur doit écrire une réponse dans une zone de texte✓ Confirmer son réponse : l'ingénieur doit cliquer sur « envoyée »

6.8 Conclusion :

A ce stade nous avons réussi donc à développer le quatrième et le dernier sprint de notre application pour arriver à un produit complet et fonctionnel. Notre application Web et mobile est donc prête à être exploiter en offrant aux agriculteurs la possibilité de gérer convenablement toutes les fonctionnalités de la plateforme.

Conclusion générale

Ce stage s'inscrit dans le cadre du Projet de master effectuées au sein de Faculté des Sciences Juridiques, Economiques et de Gestion de Jendouba FSJEGJ en vue de l'obtention du diplôme en Master professionnel en informatique spécialité commerce électronique. Dans ce contexte, nous avons eu l'occasion de concevoir et développer une application web et mobile sous la plate-forme React js et React Native intitulée « Diagnostic de maladie des plantes des grandes cultures ».

Durant les six mois de notre projet, nous avons pu réaliser les objectifs fixés au début de ce projet. Nous avons développé l'application « Diagnostic de maladie des plantes des grandes cultures » avec toutes ses fonctionnalités. Cette application est actuellement en phase finale de test et qui sera déployée très prochainement par l'institut national des grandes cultures INGC. Cependant, nous avons eu quelques dérives en termes de délai par rapport au planning prévu.

Il est important de noter que la réalisation de ce projet nous a été bénéfique sur tous les plans. Sur le plan technique, ce projet nous a été une bonne occasion pour mettre en œuvre et d'approfondir les techniques et le savoir académique acquis durant notre cursus à la FSJEGJ.

Sur le plan professionnel, ce projet nous a également permis d'améliorer nos aptitudes à communiquer et à travailler dans un cadre professionnel ce qui va sans doute nous aider à bien s'intégrer dans la vie professionnelle.

Pour conclure, nous pouvons affirmer que les besoins du cahier des charges ont été satisfait, malgré les difficultés rencontrées au cours de la réalisation. Plus encore, notre travail peut être sujet d'extensions futures dans la mesure où nous pourrons comprendre des améliorations. En effet, nous envisageons d'ajouter notre application sur les boutiques en ligne et la développer sous d'autres plateformes comme iphone.

Bibliographie

Ouvrages :

- (a) P. Roques, UML 2, Mod'éliser une application web, édition Eyrolles, 2006 page 21.
- (b) P. Roques et F. Vallee, UML 2 en action : De l'analyse des besoins à la conception. Eyrolles, 2007.
- (c) Mod'élisation de systèmes complexes avec SysML Pascal Roques Préface de Jean-Michel Bruel 2013.
- (d) Le Guide Scrum Le guide définitif de Scrum : les règles du jeu : Développé et maintenu par Ken Schwaber et Jeff Sutherland Juillet 2013.
- (e) Jean-Pierre Vickoff, Méthodes agile les meilleures pratiques compréhension et mise en œuvre, 2009.

Webgraphie :

- (f) <http://ingc.com.tn/index.php/ar/>
- (g) <https://www.snhf.org/hortiquid-le-savoir-au-jardin/>
- (h) <http://ephytia.inra.fr/fr/CMS/74/Diagnoplant>
- (i) <https://www.cliniquedesplantes.fr/>
- (j) <https://www.botaniqueeditions.com/content/20-utilisation-etiquettes-jardin>
- (k) <https://play.google.com/store/apps/details?id>
- (l) <https://www.academiepro.com/uploads/cours>
- (m) <http://agilemanifesto.org/iso/fr manifesto.html>
- (n) <https://www.marine-guyot.ovh/methode-agile-ou-classique/>
- (o) <https://Méthode-meilleures-pratiques-Compréhension-œuvre/dp/2912843073>
- (p) <https://Méthode-meilleures-pratiques-Compréhension-œuvre/dp/2912843073>
- (q) <https://blog-gestion-de-projet.com/agilité-et-scrum-fondamentaux/roles-scrum/>
- (r) <https://blog-gestion-de-projet.com/agilité-et-scrum-fondamentaux/artefacts-scrum/>
- (s) <https://www.mongodb.com/fr-fr>
- (t) <https://fr.wikipedia.org/wiki/Node.js>
- (u) <https://fr.wikipedia.org/wiki/React>
- (v) <https://fr.wikipedia.org/wiki/ReactNative>
- (w) <https://fr.wikipedia.org/wiki/VisualStudioCode>

BIBLIOGRAPHIE

- (x) <http://www.dicodunet.com/definitions/developpement/latex.htm>
- (y) [https://fr.wikipedia.org/wiki/Bootstrap_\(framework\)](https://fr.wikipedia.org/wiki/Bootstrap_(framework))
- (z) <https://www.tice-education.fr/tous-les-articles-er-ressources/articles-internet/819-draw-io-un-outil-pour-dessiner-des-diagrammes-en-ligne> : text=Draw.io est une application XML puis de l'exp