Esprit

Département Technologies du Web et de l’Internet)

Projet de Fin d’étude

Intitulé

Conception et réalisation d’une application de la gestion des assurances automobile

Organisme

DSI Esprit

Réalisé par

zakhama olfa

**Encadrants Internes** **Encadrant externe**

Yosra Ouerghi Chokri Chaaraoui

Yassine Amri

Mars 2021

# Dédicaces

Je dédie ce travail

**A la mémoire de mon père**

Décédé trop tôt, qui m’a toujours poussé et motivé dans mes études.

Ses conseils ont balisé le chemin de ma vie,je continue aujourd’hui à appliquer ses acquis.

J’éspère que, du monde qui est sien maintenant , il aprécie cette humble geste comme preuve de reconnaissance de la part de sa fille qui a toujours prié pour le salut de son âme .

Puisse Dieu le tout puissant , l’avoir en sa sainte miséricorde .

**A ma très chère Maman**

Aucune dédicace très chère maman, ne pourrait exprimer la profondeur des sentiments que j’éprouve pour toi , tes sacrifices innombrables et ton dévouement firent pour moi un encouragement.

tu as guetté mes pas, et tu a couvé de tendresse, et ta bénédiction m’a été d’un grand secours pour mener à bien mes études.

tu m’as aidé et soutenu pendant de nombreuses années avec à chaque fois une attention renouvelée.

Puisse Dieu, tout puissant te combler de santé, de bonheur et te procurer une longue vie.

**A mon très cher frère Chiheb et ma très chère soeur Asma**

Je vous souhaite un avenir plein de joie, de bonheur, de réussite et de sérénité.

# Remerciements

Je tiens à remercier en tout premier lieu tous les responsables du département informatique « DSI » qui ont collaboré, de près ou de loin, à l’élaboration de ce travail.

Je suis également reconnaissant à tous mes professeurs à qui je dois ma formation actuelle.

Mon profond respect s’éteint particulièrement à M. Chokri Chaaraoui qui m’a accueilli, guidé tout au long de mon projet.

Un grand "Merci" et une profonde gratitude à mes superviseurs, Yosra Ouerghi et M. Yassine Amri pour leurs précieux conseils, leurs encouragements constants, et pour m’avoir suivi et guidé pour mener efficacement ce travail

Je n’oublierai pas non plus de remercier tous les membres du jury qui m’ont fait l’honneur de bien vouloir étudier mon travail.

**Zakhama Olfa**

I

**Table des matières**

**Dédicaces 1**

**Remerciements I**

**Introduction Générale 1**

1. **Étude préalable 3**
   1. Présentation du cadre du stage . . . . . . . . . . . . . . . . . . . . . . 4
      1. Présentation de la DSI . . . . . . . . . . . . . . . . . . . . . . . 4
      2. Présentation de la société cliente . . . . . . . . . . . . . . . . . 4
   2. Modélisation du contexte . . . . . . . . . . . . . . . . . . . . . . . . . 5
      1. Métier de l’assurance auto : . . . . . . . . . . . . . . . . . . . . 5
      2. Les différents catégories d’assurances auto . . . . . . . . . . . . 5
      3. La lutte contre la fraude . . . . . . . . . . . . . . . . . . . . . . 5
   3. Problématique´ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
   4. Étude de l’existant . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
      1. Description de l’existant . . . . . . . . . . . . . . . . . . . . . . 6
      2. Critique de l’existant . . . . . . . . . . . . . . . . . . . . . . . . 7
   5. Solution proposée . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
      1. L’innovation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
   6. Choix de la méthodologie de développement . . . . . . . . . . . . . . . 8 1.6.1 La méthode Agile . . . . . . . . . . . . . . . . . . . . . . . . . . 8
2. **Analyse et spécification des besoins 12**
   1. Capture des besoins . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13

II

* + 1. Identification des acteurs . . . . . . . . . . . . . . . . . . . . . . 13
    2. Besoins fonctionnels . . . . . . . . . . . . . . . . . . . . . . . . 14
    3. Besoins non fonctionnels . . . . . . . . . . . . . . . . . . . . . . 15
    4. Besoins non fonctionnels . . . . . . . . . . . . . . . . . . . . . . 15

1. **Gestion des données métiers 16**
   1. Sprint Backlog . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17
      1. Les histoires à réaliser . . . . . . . . . . . . . . . . . . . . . . . 17
      2. planification des sprints . . . . . . . . . . . . . . . . . . . . . . 22
   2. Conception . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23
      1. Diagramme de classe général . . . . . . . . . . . . . . . . . . . 23
      2. Diagramme des cas d’utilisation général . . . . . . . . . . . . . 23
      3. Diagramme de composants . . . . . . . . . . . . . . . . . . . . 25
   3. Environnement de travail . . . . . . . . . . . . . . . . . . . . . . . . . . 28
      1. Environnement logiciel . . . . . . . . . . . . . . . . . . . . . . . 28
   4. Spécification Architectural . . . . . . . . . . . . . . . . . . . . . . . . . 31
      1. Architecture physique . . . . . . . . . . . . . . . . . . . . . . . 31
      2. Architecture logi que . . . . . . . . . . . . . . . . . . . . . . . . 31
   5. Environnement de travail . . . . . . . . . . . . . . . . . . . . . . . . . . 32 3.5.1 Environnement matériel . . . . . . . . . . . . . . . . . . . . . . 32
2. **Rôle Utilisateur 33**

4.0.1 Sprint Backlog . . . . . . . . . . . . . . . . . . . . . . . . . . . 34

* 1. Analyse . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35
  2. Conception . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 42
  3. Implémentation de premier module . . . . . . . . . . . . . . . . . . . . 43 4.3.1 Interface d’authentification : . . . . . . . . . . . . . . . . . . . . 43

1. **Sprint Deux : Rôle Manager 45**
   1. Sprint Backlog . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46
      1. Histoires à réaliser . . . . . . . . . . . . . . . . . . . . . . . . . 46
      2. Objectif du sprint . . . . . . . . . . . . . . . . . . . . . . . . . . 46
   2. Analyse . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 47
   3. Conception . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 52
   4. Implémentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 54 5.4.1 Interface Manager . . . . . . . . . . . . . . . . . . . . . . . . . 54
2. **Sprint Trois : Rôle Admin 56**
   1. Sprint Backlog . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 57
      1. Histoires à réaliser . . . . . . . . . . . . . . . . . . . . . . . . . 57
      2. Objectif du sprint . . . . . . . . . . . . . . . . . . . . . . . . . . 59
   2. Analyse . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 59
      1. Diagramme de Cas d’utilisation "Admin" . . . . . . . . . . . . . 59
   3. Conception . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 65
   4. Implémentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 66
      1. Interface Compagnie . . . . . . . . . . . . . . . . . . . . . . . . 66
      2. Interface type contrat . . . . . . . . . . . . . . . . . . . . . . . 67
      3. Interface Garantie . . . . . . . . . . . . . . . . . . . . . . . . . 69
      4. Interface Clause . . . . . . . . . . . . . . . . . . . . . . . . . . 70
      5. Interface fréquence de priodicité . . . . . . . . . . . . . . . . . 71
      6. Interface mode de paiement . . . . . . . . . . . . . . . . . . . . 72
3. **Sprint quatre : Rôle Souscripteur et Agent d’assurance 74**
   1. Sprint Backlog . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 75
      1. Histoires à réaliser . . . . . . . . . . . . . . . . . . . . . . . . . 75
      2. Objectif du sprint . . . . . . . . . . . . . . . . . . . . . . . . . . 76
   2. Analyse . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 76
   3. Conception . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 85
   4. Implémentation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 87

**Conclusion Générale et Perspectives 89**

**Liste des tableaux**

1.1 Rôles associés aux acteurs de SCRUM . . . . . . . . . . . . . . . . . . 11

3.1 Product Backlog . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22

4.1 Product Backlog . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35

5.1 Liste des tâches du deuxième Sprint . . . . . . . . . . . . . . . . . . . . 46

6.1 Liste des tâches du troixième Sprint . . . . . . . . . . . . . . . . . . . . 58

7.1 Liste des tâches du quatrième Sprint . . . . . . . . . . . . . . . . . . . 76

V

**Table des figures**

* 1. la logo de l’organisme . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
  2. Logo CGA . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
  3. Principe de la méthodologie Scrum . . . . . . . . . . . . . . . . . . . . 9
  4. Planification des sprints . . . . . . . . . . . . . . . . . . . . . . . . . . 23
  5. Diagramme de classe général . . . . . . . . . . . . . . . . . . . . . . . 23
  6. Diagramme de cas d’utilisation général . . . . . . . . . . . . . . . . . . 24
  7. Diagramme de composants . . . . . . . . . . . . . . . . . . . . . . . . 25
  8. Prototype de l’interface "authentification" . . . . . . . . . . . . . . . . . 26
  9. Prototype de l’interface "demande souscription" . . . . . . . . . . . . . 27
  10. Staruml logo . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28
  11. Staruml logo . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29
  12. Visual studio Code logo . . . . . . . . . . . . . . . . . . . . . . . . . . 29
  13. XAMP Control Panel . . . . . . . . . . . . . . . . . . . . . . . . . . . 30
  14. ShareLaTeX logo . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30
  15. architecture 3 tiers . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 31
  16. architecture MVC . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 32
  17. Diagramme d’activité "Inscription" . . . . . . . . . . . . . . . . . . . . 38
  18. Diagramme d’activité "Authentification" . . . . . . . . . . . . . . . . . 39
  19. Diagramme de séquence système "s’inscrire" . . . . . . . . . . . . . . . 40
  20. Diagramme de séquence système "s’authentifier" . . . . . . . . . . . . . 41
  21. Diagramme de classe "Utilisateur" . . . . . . . . . . . . . . . . . . . . . 42
  22. Diagramme d’activité "Authentification" . . . . . . . . . . . . . . . . . 43

VI

* 1. Interface authentification . . . . . . . . . . . . . . . . . . . . . . . . . 44
  2. Diagramme d’activité "Authentification" . . . . . . . . . . . . . . . . . 47
  3. Diagramme d’activité "Ajout utilisateur" . . . . . . . . . . . . . . . . . 49
  4. Diagramme d’état de transition "Ajout utilisateur" . . . . . . . . . . . . 50
  5. Diagramme de séquence système "Ajout utilisateur" . . . . . . . . . . . 51
  6. Diagramme de classe "Manager" . . . . . . . . . . . . . . . . . . . . . . 52
  7. Diagramme de séquence objet "Ajout utilisateur" . . . . . . . . . . . . . 53
  8. Interface liste des utilisateurs . . . . . . . . . . . . . . . . . . . . . . . 54
  9. Interface Créer utilisateur . . . . . . . . . . . . . . . . . . . . . . . . . 55
  10. Diagramme de cas d’utilisateur "Admin" . . . . . . . . . . . . . . . . . 60
  11. Diagramme d’activité "Ajout compagnie" . . . . . . . . . . . . . . . . 62
  12. Diagramme d’état de transition "Ajout compagnie" . . . . . . . . . . . 63
  13. Diagramme de séquence système "Ajout compagnie" . . . . . . . . . . . 64
  14. Diagramme de classe "Admin" . . . . . . . . . . . . . . . . . . . . . . . 65
  15. Diagramme de séquence objet "Ajout compagnie" . . . . . . . . . . . . 65
  16. Interface Liste des compagnies . . . . . . . . . . . . . . . . . . . . . . 66
  17. Interface Ajout compagnie . . . . . . . . . . . . . . . . . . . . . . . . . 67
  18. Interface Liste des types de contrats . . . . . . . . . . . . . . . . . . . 67
  19. Interface Ajout type contrat . . . . . . . . . . . . . . . . . . . . . . . . 68
  20. Interface Liste des garanties . . . . . . . . . . . . . . . . . . . . . . . . 69
  21. Interface Ajout garantie . . . . . . . . . . . . . . . . . . . . . . . . . . 70
  22. Interface Liste des clauses . . . . . . . . . . . . . . . . . . . . . . . . . 70
  23. Interface Ajout clause . . . . . . . . . . . . . . . . . . . . . . . . . . . 71
  24. Interface Liste des fréquence . . . . . . . . . . . . . . . . . . . . . . . 71
  25. Interface Ajout fréquence . . . . . . . . . . . . . . . . . . . . . . . . . 72
  26. Interface Liste des modes de paiement . . . . . . . . . . . . . . . . . . 73
  27. Interface Ajout mode de paiement . . . . . . . . . . . . . . . . . . . . 73
  28. Diagramme de cas d’utlisateur "Souscripteur et agent d’assurance " . . 76
  29. Diagramme d’activité création contrat . . . . . . . . . . . . . . . . . . 79
  30. Diagramme d’activité création constat . . . . . . . . . . . . . . . . . . 80
  31. Diagramme état de tansition création contrat . . . . . . . . . . . . . . 81
  32. Diagramme état de tansition création constat . . . . . . . . . . . . . . 82
  33. Diagramme séquence système création contrat . . . . . . . . . . . . . . 83
  34. Diagramme séquence système création constat . . . . . . . . . . . . . . 84
  35. Diagramme de classe "souscripteur et agent d’assurance " . . . . . . . . 85
  36. Diagramme de séquence objet "Création Constat" . . . . . . . . . . . . 86
  37. Interface formulaire "Demande souscription" . . . . . . . . . . . . . . . 87
  38. affichage Interface "Demande souscription" . . . . . . . . . . . . . . . . 87
  39. Interface formulaire "Création contrat" . . . . . . . . . . . . . . . . . . 87
  40. affichage de l’état des demande de souscriptions d’un souscripteur . . . 88
  41. affichage de la liste des contrats . . . . . . . . . . . . . . . . . . . . . . 88
  42. Formulaire de création constat . . . . . . . . . . . . . . . . . . . . . . . 88

**Esprit 2021**

# Introduction Générale

Considérée comme une innovation singulière , la science de l’informatique joue un rôle fondamental dans notre société , servant à faciliter et améliorer nos mode de vie, notamment, le domaine des assurances .

Etant le pivot de l’économie , ce secteur rend tant de services de développement tout en s’accommodant au système digital.

Dans le cadre de notre stage de fin d’études, DSI nous a confié l’étude, la conception et le développement d’un module Web.

Ce module est une application appelée «gestion des assurances automobile» qui lutte contre toute forme de fraude.

Notre candidature est destinée au comité CGA,

Son objectif est de produire des contrats pour les assurés afin d’avoir un accès permanent à l’ensemble de leurs services dans un environnement très sécurisé.

Afin de détailler notre travail lors de ce stage, nous présentons une approche générale de notre rapport.

Ce rapport est organisé en sept chapitres :

Dans le premier chapitre, nous présentons le cadre général du projet y compris l’organisme d’accueil, le contexte du projet, la problématique à résoudre, l’étude de l’existant, la contribution apportée ainsi que la méthodologie adoptée .

Le second chapitre sont consacrés pour l’analyse et les spécifications des besoins fonctionnels et non fonctionnels afin de préciser les objectifs à atteindre.

Le troisième chapitre est dédié à la gestion des données métiers tels que l’implémentation du sprint backlog , la réalisation de la conception général de l’application , la spécification architectural du projet ainsi que les environnements matériels et logiciels utilisés lors du développement .

Le quatrième chapitre, le cinquième chapitre, le sixième chapitre et le septième chapitre présentent la phase du développement de notre application tout en respectant les principes fondamentaux de la méthode Agile Scrum.

Enfin, nous achevons ce rapport par une conclusion générale et quelques perspectives .

**Chapitre 1**

**Étude préalable**

3

## Introduction

Dans ce chapitre, nous présenterons le cadre général du projet, en commençant par une présentation détaillée de la gestion des systèmes d’information DSI suivie d’une description générale du contexte du projet, enfin nous proposerons une solution adéquate aux problèmes cités dans l’analyse. de l’existant.

## 1.1 Présentation du cadre du stage

### 1.1.1 Présentation de la DSI

Esprit (Ecole Supérieure Privée d’Ingénieurs et de Technologie) est l’école d’ingénieurs qui, lors de sa création, s’est fixé comme objectif principal de former des ingénieurs «autrement» pour se rapprocher du monde du travail.

En conséquence, "Esprit" a créé un département de recherche-développement-innovation (RDI) nommé Esprit-Tech a été créé en 2010 pour former des ingénieurs totalement en phase avec les besoins des entreprises en rencontrant des professionnels pour faire le point sur certains programmes pour les adapter. aux besoins réels du marché.

Nous avons eu la chance d’effectuer trois stages en entreprise pendant toute la durée du cursus universitaire dont fait partie mon projet de fin d’études



FIGURE 1.1 – la logo de l’organisme

### 1.1.2 Présentation de la société cliente

Notre société cliente est le Comité Général des Assurances CGA.

Conformément à la loi n 2008-8 du 13 février 2008, ce comité a été mis en place, son objectif étant de modifier et compléter le code des assurances.

Elle est une autorité de contrôle du secteur des assurances, elle veille à la protection des droits des assurés et des bénéficiaires de contrats d’assurance, à la solidité de la base financière des sociétés d’assurance et de réassurance et à leur capacité et à honorer leurs engagements.



FIGURE 1.2 – Logo CGA

## 1.2 Modélisation du contexte

### 1.2.1 Métier de l’assurance auto :

L’assurance est une technique basée sur l’esprit de solidarité, cette solidarité est particulièrement nécessaire dans le domaine de l’assurance automobile étant évident que chaque usager pris isolément ne peut faire face à des dommages corporels et / ou matériels très importants. que ces appareils sont susceptibles de provoquer.

C’est en ce sens que la réforme de l’assurance automobile a eu lieu pour nous offrir une certaine protection financière en cas de sinistre en ayant une responsabilité juridique envers autrui pour dommages corporels ou matériels et en assurant les indemnisations des

sinistresraires

### 1.2.2 Les différents catégories d’assurances auto

Afin de réduire le coût de notre assurance auto, il faut opter pour des formules de couverture plus fiables telles que :

Assurance automobile au tiers : qui couvre la responsabilité civile automobile, afin de réparer les dommages corporels et matériels en cas d’accident responsable. Il peut également couvrir les garanties d’options telles que le vol, l’incendie, le verre brisé ou l’assistance.

Assurance automobile tous risques : qui comprend des garanties supplémentaires telles que la protection personnelle ainsi que le vandalisme

Assurance automobile au kilomètre : Si nous prenons notre voiture pour de courts trajets, nous sommes considérés comme un petit cavalier, dans ce cas, l’assurance kilométrage automobile nous concerne directement.

## 1.2 Modélisation du contexte

### 1.2.1 Métier de l’assurance auto :

L’assurance est une technique basée sur l’esprit de solidarité, cette solidarité est particulièrement nécessaire dans le domaine de l’assurance automobile étant évident que chaque usager pris isolément ne peut faire face à des dommages corporels et / ou matériels très importants. que ces appareils sont susceptibles de provoquer.

C’est en ce sens que la réforme de l’assurance automobile a eu lieu pour nous offrir une certaine protection financière en cas de sinistre en ayant une responsabilité juridique envers autrui pour dommages corporels ou matériels et en assurant les indemnisations des sinistres .

### 1.2.2 Les différents catégories d’assurances auto

Afin de réduire le coût de notre assurance auto, il faut opter pour des formules de couverture plus fiables telles que :

Assurance automobile au tiers : qui couvre la responsabilité civile automobile, afin de réparer les dommages corporels et matériels en cas d’accident responsable. Il peut également couvrir les garanties d’options telles que le vol, l’incendie, le verre brisé ou l’assistance.

Assurance automobile tous risques : qui comprend des garanties supplémentaires telles que la protection personnelle ainsi que le vandalisme

Assurance automobile au kilomètre : Si nous prenons notre voiture pour de courts trajets, nous sommes considérés comme un petit cavalier, dans ce cas, l’assurance kilométrage automobile nous concerne directement.

### 1.2.3 La lutte contre la fraude

La fraude est un élément essentiel des activités des compagnies d’assurance .

On avait autant de fraudes que des matières assurés ( automobiles ,santé ..)

Il y’avait des acteurs professionnels de la fraude notamment la fraude présidente ,la fraude à la TVA et la fraude à l’assurance .

En 2015 , plus de 46.000 fraudes ont été recensés par l’ALFA .

La lutte contre ce phénomène est un sujet brûlant dans le monde de l’assurance

## 1.3 Problématique´

De nombreux cas de fraude à l’assurance automobile sont détectés chaque année.

Notamment :

* De fausses déclarations des sinistres.
* Une souscription frauduleuse d’un contrat d’assurance .
* Une fausse réclamation intentionnelle pour indemnisation .
* Un conducteur malintentionné confronté à un conducteur provocant un accident .

En conséquence, les assureurs redoublent de vigilance en entamant des sanctions telle que la résiliation du contrat accompagnée d’un préavis .

Les questions que nous pouvons nous poser à ce stade sont les suivantes :

-Que devons-nous faire pour éviter la fraude?

-Quel sera le nouveau mode de fonctionnement de l’assurance pour lutter contre toutes tentative de tricherie?

## 1.4 Étude de l’existant

### 1.4.1 Description de l’existant

Plus de deux millions de voitures circulant en Tunisie et ce chiffre augmente d’année en année.

Selon le président de l’Association tunisienne des droits d’assurances (ATDA), Abdellatif Mamoghli, les actes frauduleux des assurés sont en augmentation.

Afin de lutter contre ce fléau, les experts ont mis en place un système de détection des fraudes.

Cet outil est basé sur l’isolation de façon systématique des indicateurs de fraudes pour prévoir qu’un document soit falsifié.

En plus, ils ont préparé un logiciel de calcul des probabilités de ces arnaques.

### 1.4.2 Critique de l’existant

Plus le nombre des véhicules progressent plus les pratiques des fraudes émergent

Puisque la quantité des données demeure énorme la tâche d’indicateurs de fraude devient compliquée.

À ce jour, aucune statistique ne permet de mesurer l’étendue réelle du phénomène décrit.

Cela signifie simplement que nous n’avons pas d’indicateurs fiables .

## 1.5 Solution proposée

Pour éviter d’être victime d’une arnaque en assurance-automobile , On a décidé de mettre en œuvre une application Web «Gestion des assurances automobile» sécurisée au assureur afin de mieux gérer les assurances et faciliter la vie de l’assuré .

Ce projet permet de gérer la complexité spécifique du processus de souscription en assurance automobile.

### 1.5.1 L’innovation

Nous proposons de viser la modernisation les plateformes d’assurance automobile et à mettre fin à l’assurance traditionnelle.

En effet, grâce au fichier informatisé, il n’y a pas de perte de temps ni de document.

L’assuré a accès à son dossier quand il le souhaite et le gestionnaire peut répondre à ses demandes de manière rapide et fiable.

De plus, le client peut souscrire à n’importe quelle compagnie d’assurance enregistrée dans le système sans avoir à se déplacer.

Enfin, l’un des avantages majeurs est la sécurité et la transparence du contrat.

## 1.6 Choix de la méthodologie de développement

## 1.6.1 La méthode Agile

Pour le bon déroulement du projet, nous avons besoin d’une méthodologie efficace.

En effet, la méthode agile est une méthode incrémentale itérative, qui est réalisée dans un esprit collaboratif et le formalisme nécessaire. Il génère des produits de haute qualité tout en tenant compte de l’évolution des besoins des clients.

L’adhésion à d’une méthodologie spécifique peut garantir le succès du projet et l’adéquation des résultats et des objectifs initialement déterminés.

Nous choisissons d’utiliser la méthode Scrum, qui permet de gagner beaucoup de temps, d’améliorer la qualité du développement et la satisfaction des utilisateurs, tout en prenant en compte les évolutions des besoins clients tout au long du projet.

Cette méthodologie est suivie par la DSI pour la gestion de ses projets.

Entièrement développé et testé pour de courtes itérations. Simplicité des processus. de la productivité Permet d’adapter le logiciel créé en fonction de l’évolution du projet.

#### 1.6.1.1 Présentation de la méthodologie Scrum

Scrum est un processus agile qui permet de fournir la plus grande valeur commerciale dans les plus brefs délais.

Avec Scrum, le développement produit est rythmé par une série d’itérations qui prennent 2 à 4 semaines appelées sprints.

Les tâches à mettre en œuvre lors des différents sprints (Backlog) sont définies par l’équipe en tenant compte des priorités et capacités. À partir de ce contenu, l’équipe identifie les tâches nécessaires et s’engage à atteindre l’objectif du sprint.

À la fin de chaque sprint, tous les acteurs SCRUM peuvent voir travailler ce qui a déjà été réalisé et ce qui a été ajouté au produit pendant le sprint en cours et décider de le livrer tel quel ou de continuer à le faire. s’améliorer lors d’un sprint supplémentaire.

La figure 1.3 explique le principe de la méthode Scrum décrite ci-dessus :

#### 1.6.1.2 Les acteurs principaux de Scrum

* Product owner (Représentant clients et utilisateurs).
* Scrum Master (Protecteur de l’équipe).

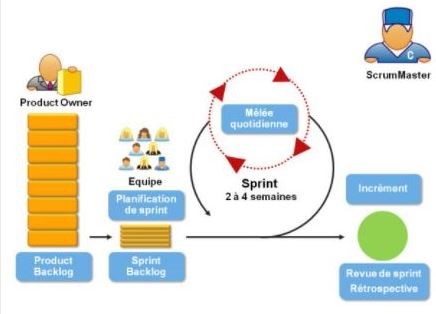


FIGURE 1.3 – Principe de la méthodologie Scrum

* Scrum Team (L’équipe de développement).
* Stakeholders (Les utilisateurs finaux).

**Product Owner**

Cet acteur porte la vision du produit à réaliser et travaille en interaction avec l’équipe de développement. Il s’agit généralement d’un expert dans le domaine d’activité du projet.

.

**Scrum Master**

Il doit maîtriser Scrum et s’assurer qu’il est correctement appliqué. Il a donc un rôle de coach à la fois auprès du Product Owner et auprès de l’équipe de développement. Il doit donc faire preuve de pédagogie. Il est également chargé de s’assurer que l’équipe de développement est pleinement productive. Habituellement, le candidat idéal pour le rôle de Scrum Master est le chef de projet. Cependant, il devra abandonner le style de gestion «commandement et contrôle» pour adopter un style de gestion participative.

|  |  |
| --- | --- |
| Rôle | Acteur |
| Product owner | Chaaraoui Chokri |
| Scrum Master | Ouerghi Yosra |
| Scrum Team | Zakhama Olfa |

Dans le tableau 1.1, nous présentons les rôles associés aux acteurs de SCRUMnous présentons les rôles associés aux acteurs de SCRUM

TABLE 1.1 – Rôles associés aux acteurs de SCRUM

## Conclusion

Dans ce chapitre, nous avons identifié les besoins de notre application afin de définir nos missions et objectifs. Puis nous avons effectué l’analyse de l’existant pour bien concevoir notre base de données.

Ensuite, nous avons proposé un solution pour démarrer l’étude de concept.

A la fin, nous avons expliqué la méthodologie du projet «SCRUM» en argumentant notre choix.

Dans le chapitre suivant, Dans le prochain chapitre, nous présenterons l’analyse et les spécifications des besoins

**Chapitre 2**

**Analyse et spécification des besoins**

12

## Introduction

Après avoir replacé le projet dans son contexte et expliqué les différentes problématiques présentes, la compréhension et l’analyse des différentes fonctionnalités offertes par notre solution sont devenues cruciales. Nous analysons, au cours de cette partie, les besoins en question et nous les expliquons pour mieux comprendre l’utilité de notre application.

Nous commençons dans cette partie en identifiant les différents acteurs impliqués et en comptant les besoins fonctionnels. Nous citons ensuite les contraintes placées sur le système. Nous terminons par exposer les éléments du Product Backlog

## 2.1 Capture des besoins

### 2.1.1 Identification des acteurs

Un acteur représente une personne ou un autre système informatique qui attend un ou plusieurs services proposés par une interface d’accès.

Il interagit avec le système en envoyant ou en recevant des messages.

De plus, votre candidature impliquera des acteurs primaires et des acteurs secondaires.

#### 2.1.1.1 Acteurs

#### Manager :

Il gère les utilisateurs .

**Souscripteur :**

IL est la personne qui envoie une demande de souscription à l’administrateur et rédige un constat en cas d’un sinistre .

Il doit avoir un compte pour se connecter à l’application

**Administrateur :**

Il est la personne qui s’occupe de la gestion totale de l’application : la gestion des données métiers tels que :les compagnies d’assurance , les garanties d’assurance ,les types de contrat

,les clauses , les fréquences de périodicité , les modes de paiement

Il doit avoir un compte pour se connecter à l’application

**Agent d’assurance :**

Il gère les demandes de souscription envoyés par les souscripteurs et crée par la suite des contrats individuels

IL doit avoir un compte pour se connecter à l’application

### 2.1.2 Besoins fonctionnels

IL doit avoir un compte pour se connecter à l’application

### 2.1.2 Besoins fonctionnels

Notre application doit répondre aux besoins fonctionnels suivants :

— Manager :

— Gérer les utilisateurs

— Administrateur :

— Gérer les compagnies d’assurance

— Gérer les types de contrats

— Gérer les garanties d’assurance

— Gérer les modes de paiement — Gérer les fréquences de périodicité

— Souscripteur :

— Créer des demandes de souscription

— Consulter ses demandes de souscription

— Consulter ses contrats individuels

— Créer des constats

— Créer des réclamations

— Agent d’assurance :

— Consulter les demandes de souscriptions

### 2.1.3 Besoins non fonctionnels

Les exigences non fonctionnelles représentent les contraintes implicites que le système doit respecter.

### 2.1.4 Besoins non fonctionnels

Les exigences non fonctionnelles représentent les contraintes implicites que le système doit respecter.

Parmi celles-ci nous citons :

— Fiabilité : Les applications doivent s’exécuter sur la base de sources de données cohérentes.

— Sécurité : son exigence constitue l’aspect technique le plus important. Il est respecté grâce au principe de contrôle d’accès appliqué par notre système (gestion des rôles et autorisations : un rôle étant un profil auquel nous accordons un ensemble de fonctionnalités, modules, sous-modules et tâches. Déterminé), mais aussi grâce à le principe de cryptage des données que nous appliquerons aux mots de passe des comptes utilisateurs

De cette manière, nous garantissons :

* La confidentialité des données
* L’intégrité des données
* La fiabilité des données.

— Ergonomie :En tenant compte de toutes les interactions possibles sur l’écran , l’application doit fournir une interface conviviale et ergonomique à l’usage des utilisateurs item textbf Reuse : l’application doit être portable pour pouvoir la réutiliser.

— Les tableaux de bord ne doivent pas être surchargés et ils doivent être bien présentés afin de mettre en évidence les informations pertinentes.

## Conclusion

Tout au long de ce chapitre, nous avons d’abord déterminé les acteurs et les besoins fonctionnels et non fonctionnels de notre solution.

Dans ce qui suit, nous détaillerons chaque sprint en déterminant leur backlog produit..

**Chapitre 3**

**Gestion des données métiers**

16

## Introduction

Le sprint est le cœur de Scrum. Il s’agit d’un bloc de temps pendant lequel un incrément de produit sera effectué. Tous les sprints d’une version ont une durée constante et ne se chevauchent jamais, c’est-à-dire qu’un sprint ne peut pas démarrer tant que le précédent n’est pas terminé.

Tout au long de ce chapitre, nous traiterons de ses utilisateurs de nos sprints pour produire un incrément potentiellement livrable.

## 3.1 Sprint Backlog

Avant d’entamer le premier sprint, nous avons un sprint zéro dédié à la préparation de ce qui est nécessaire pour lancer les sprints dans de bonnes conditions.

### 3.1.1 Les histoires à réaliser

Avant de se lancer dans un sprint, l’équipe Scrum doit définir l’objectif de ce dernier qui doit être un tableau descriptif précisant la charge de travail pour chaque tâche. Le tableau

3.1 décrit les histoires à réaliser lors de ce sprint

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sprint | id fea-  tures | features | id story | User Story | Priorité |
| Utilisateur | 1 | Inscription | 1.1 | En tant que manager, je veux pouvoir m’inscrire à Assurance Auto  . | 100 |
| 1.2 | En tant qu’admin, je veux pouvoir m’inscrire à Assurance Auto. | 99 |
| 1.3 | En tant que souscripteur, je veux pouvoir m’inscrire à Assurance  Auto. | 98 |
| 1.4 | En tant qu’agent d’assuarnce, je veux pouvoir m’inscrire à Assurance Auto | 97 |
| 2 | Authen-  tification | 2.1 | En tant que manager, je veux pourvoir me connecter à Assurance AUto avec tous les drois. | 96 |
| 2.2 | En tant qu’admin, je veux pourvoir me connecter à Assurance Auto avec tous les  drois | 95 |
| 2.3 | En tant qu’agent d’assurance, je veux pouvoir me connecter à Assurance Auto avec tous les droits. | 94 |
| 2.4 | En tant ue souscripteur, je veux pourvoir me connecter à Assurance AUto avec tous  les drois | 93 |
| Manager | 3 | Gestion ut-  lisateur | 3.1 | En tant que manager, je veux ajouter un compte utlisateur . | 92 |
| 3.2 | En tant que manager, je veux modifier un compte utilisateur. | 91 |
| 3.3 | En tant que manager, je veux consulter la liste des comptes utilisateurs. | 90 |
| 3.4 | En tant que manager, je veux supprimer un compte utilisateur. | 89 |
| Admin | 4 | Gestion compagnie d’assurance | 4.1 | En tant qu’admin, je veux ajouter une compagnie . | 88 |
| 4.2 | En tant qu’admin, je veux modifier une compagnie. | 87 |
| 4.3 | En tant qu’admin, je veux consulter la liste des compagnies. | 86 |
| 4.4 | En tant qu’admin, je veux supprimer une compagnie. | 85 |
| 5 | Gestion des garan-  ties | 5.1 | En tant qu’admin, je veux ajouter une garantie. | 84 |
| 5.2 | En tant qu’admin, je veux modifier une garantie. | 83 |
| 5.3 | En tant qu’admin, je veux consulter une liste des garanties. | 82 |
| 5.4 | En tant qu’admin, je veux supprimer une garantie. | 81 |

Gestion

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6 | des  types des | 6.1 | En tant qu’admin, je veux ajouter un type de contrat. | 80 |

contrats

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | 6.2 | En tant qu’admin, je veux modifier un type de contrat. | 79 |
| 6.3 | En tant qu’admin, je veux consulter la liste des types de contrat. | 78 |
| 6.4 | En tant qu’admin, je veux supprimer un type de contrat. | 77 |
| 7 | Gestion des clauses | 7.1 | En tant qu’admin, je veux ajouter une clause. | 76 |
| 7.2 | En tant qu’admin, je veux modifier une clause . | 75 |
| 7.3 | En tant qu’admin, je veux consulter la liste des clauses. | 74 |
| 7.4 | En tant qu’admin, je veux supprimer une clause. | 73 |
| 8 | Gestion  des fréquences de  périodicités | 8.1 | En tant qu’admin, je veux ajouter une fréquence de périodicité. | 72 |
| 8.2 | En tant qu’admin, je veux modifier une fréquence de périodicité . | 71 |
| 8.3 | En tant qu’admin, je veux consulter la liste des fréquences de périodicité. | 70 |
|  |  |  | 8.4 | En tant qu’admin, je veux supprimer une fréquence de périodicité. | 69 |
| 9 | Gestion  des modes de paie-  ment | 9.1 | En tant qu’admin, je veux ajouter un mode de paiement. | 68 |
| 9.2 | En tant qu’admin, je veux modifier un mode de paiement . | 67 |
| 9.3 | En tant qu’admin, je veux consulter la liste des modes de paiement. | 66 |
| 9.4 | En tant qu’admin, je veux supprimer un mode de paiement. | 65 |
| Souscripteur  et Agent  d’assuarnce | 10 | Gestion  des demande de  souscription | 10.1 | En tant que souscripteur, je veux créer une demande de souscription. | 64 |
| 10.2 | En tant qu’agent d’assurance , je veux accepter une demande de souscription | 63 |
| 10.3 | En tant qu’agent d’assurance , je veux refuser une demande de  souscription | 62 |
| 10.4 | En tant que souscripteur , je veux consulter mes demandes de sous-  cription | 61 |
| 10.5 | En tant que souscripteur , je veux consulter toutes les demandes de  souscription | 60 |
|  | 11 | Gestion des réclamations | 11.1 | En tant que souscripteur , je veux créer une réclamation | 59 |
| 12 | Gestion des contrats individuels | 12.1 | En tant qu’agent d’assurance, je veux créer un contrat individuel. | 58 |
| 12.2 | En tant que souscripteur je veux consulter mes contrats individuels | 57 |
| 12.3 | En tant qu’agent d’assurance , je veux consulter tous les  contrats individels | 56 |
| 13 | Gestion des constats | 13.1 | En tant que souscripteur , je veux créer un constat. | 55 |
| 13.2 | En tant qu’agent d’assurance je veux consulter tous les constats | 54 |
| 13.3 | En tant que souscripteur , je veux consulter tous mes constats | 53 |

TABLE 3.1: Product Backlog

### 3.1.2 planification des sprints

Le projet a duré six mois et a duré entre septembre 2020 et février 2021.

Nous avons choisi, pour planifier notre projet, d’utiliser le diagramme de GANTT qui est l’un des outils les plus efficaces pour représenter visuellement l’avancement des différentes activités (tâches) qui constituent un projet.

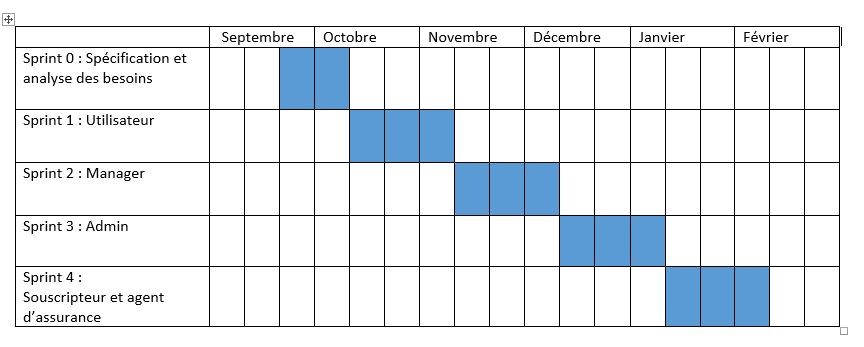


FIGURE 3.1 – Planification des sprints

## 3.2 Conception

### 3.2.1 Diagramme de classe général

Les diagrammes de classes sont l’un des types de diagrammes UML les plus utiles car ils décrivent clairement la structure d’un système particulier en modélisant ses classes, ses attributs, ses opérations et les relations entre ses objets.

Le

diagramme

3.2

modélise

le

diagramme

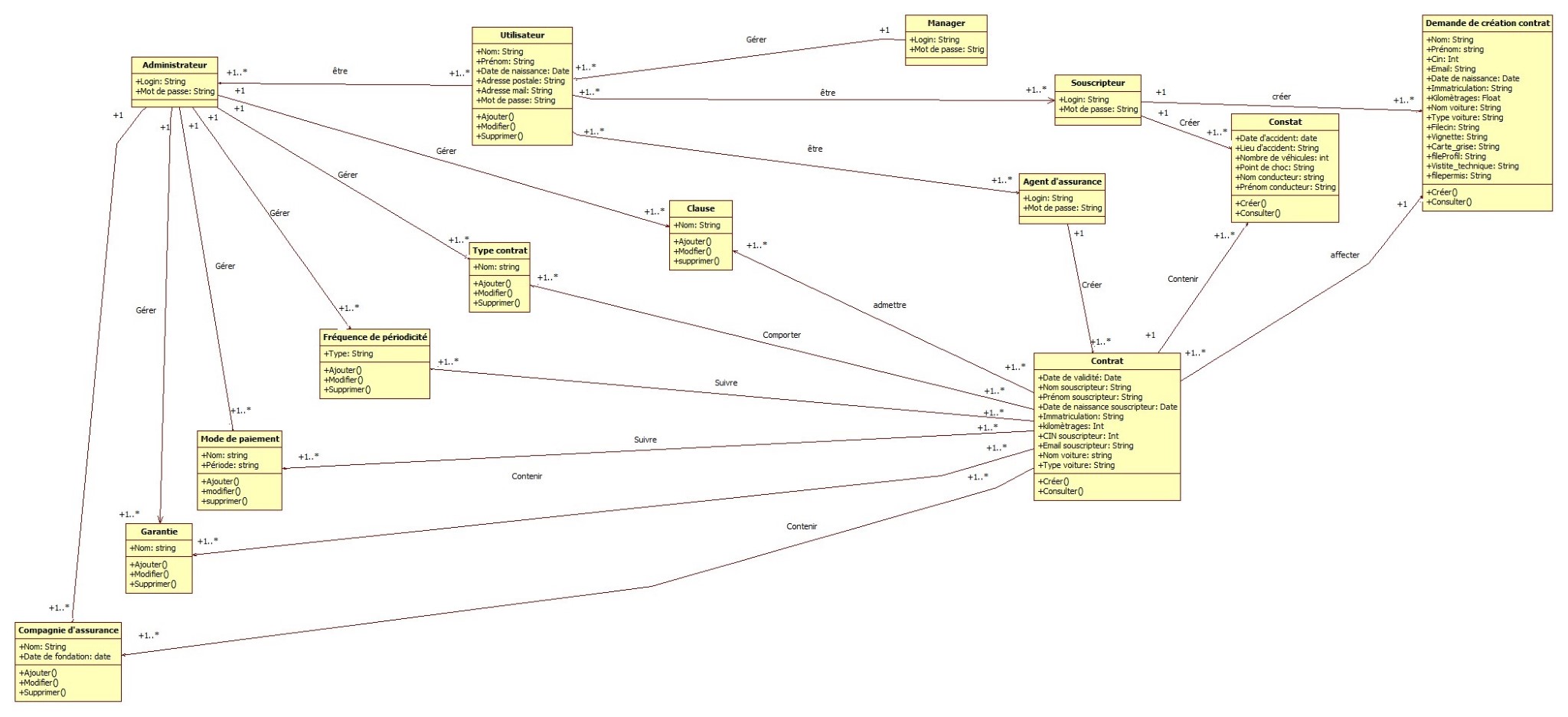
de

classes

général

de

l’application



F

IGURE

3.2

–

Diagramme

de

classe

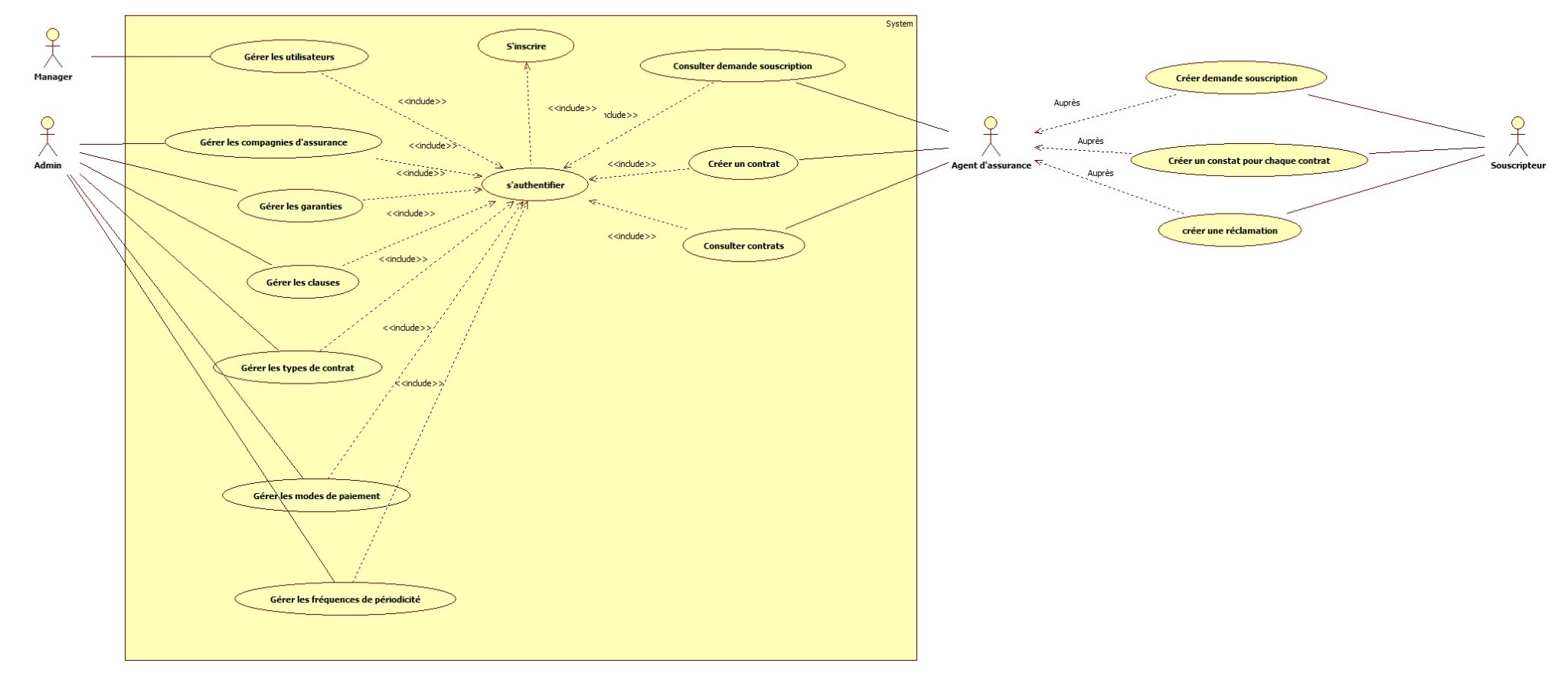
général

### 3.2.2 Diagramme des cas d’utilisation général

Les diagrammes de cas d’utilisation sont des diagrammes UML utilisés pour une représentation du comportement fonctionnel d’un système logiciel.

La figure 3.3 présente le diagramme de cas d’utilisation général .

FIGURE 3.3 – Diagramme de cas d’utilisation général

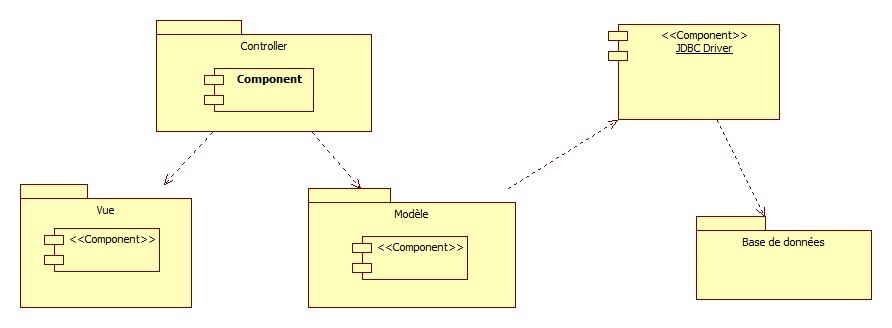


### 3.2.3 Diagramme de composants

Les diagrammes de composants UML représentent les relations entre les différents composants d’un système dans une vue d’ensemble statique.

Ils peuvent inclure à la fois des aspects de modélisation logiques et physiques

FIGURE 3.4 – Diagramme de composants



Le contrôleur désigne le répertoire : /Controller

Le modèle désigne le répertoire : /Entity

La vue désigne le répertoire : /templates

#### 3.2.3.1 Prototype des IHM

Le maquettage prototypage est une étape nécessaire,il consiste à préparer quelques interfaces du futur produit à l’aide de l’outil de maquettage Balsamiq. L’objectif principal de cette technique est d’obtenir des informations sur l’interaction des utilisateurs avec le futur produit.

* La maquette de la figure 3.5 3.6 représente l’interface authentification .



FIGURE 3.5 – Prototype de l’interface "authentification"

* La maquette de la figure 3.6 représente l’interface detaille de la demande création contrat .

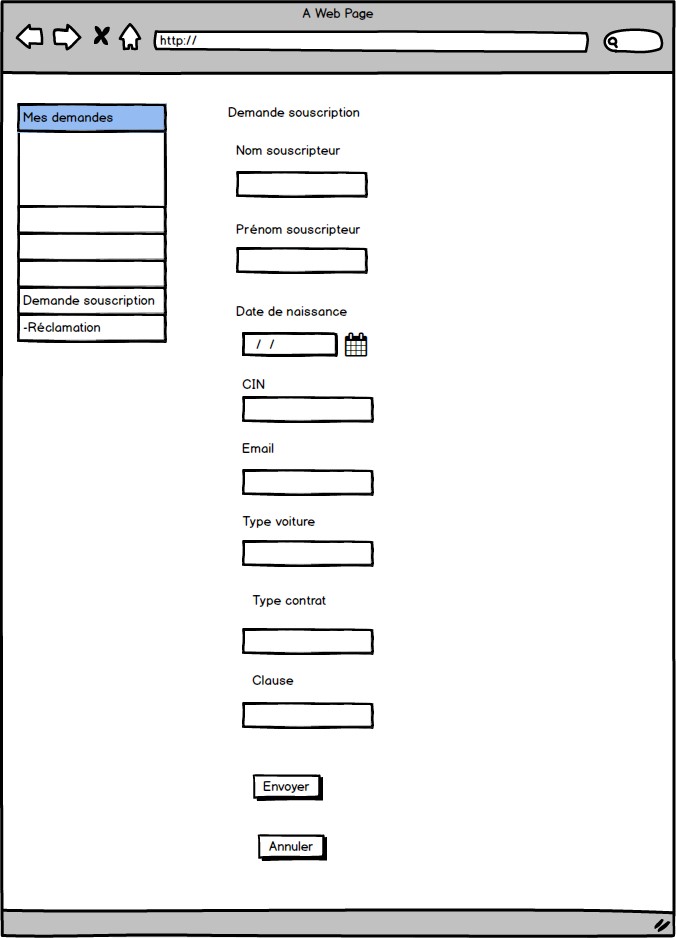


FIGURE 3.6 – Prototype de l’interface "demande souscription"

## 3.3 Environnement de travail

### 3.3.1 Environnement logiciel



FIGURE 3.7 – Staruml logo

Pour accomplir notre mission, nous avons utilisé plusieurs ressources logicielles.

#### Gitlab

GIT est un outil de versionnage pratique pour :

* Travaillez avec des équipes dispersées et distantes. Dans le cas où l’équipe et le PO sont au même endroit, il est sûrement facile d’utiliser les posts-it pour gérer les tâches. item Gère automatiquement les versions de code et les modifications du contenu de l’arborescence.

Voici quelques commandes de base telles que :

* "Git init" pour ajouter le référentiel Git sous le dossier de travail.
* "Git status" pour afficher les modifications sur le référentiel.
* "Git add" pour confirmer toutes les modifications (localement uniquement).
* "Git commit -m NOM–COMMIT" pour enregistrer les branches confirmées sous le nom "NOM–COMMIT"
* ""Git push" : Après avoir enregistré une copie locale d’un référentiel et modifié ses fichiers sur notre ordinateur, nous pouvons télécharger les modifications sur GitHub. C’est ce qu’on appelle pousser vers Github
* ""Git pull" : Lorsque le référentiel distant change, notre copie locale sera derrière lui. Nous pouvons le mettre à jour avec les nouveaux changements dans le référentiel distant. Cela s’appelle l’extraction de Github, comme cela est fait par cette commande

#### StarUML

StarUML est un logiciel de modélisation UML, qui a été transféré en "open source" par son éditeur ", à la fin de son exploitation commerciale (qui se poursuit évidemment ...) , sous une licence modifiée de la GNU GPL.

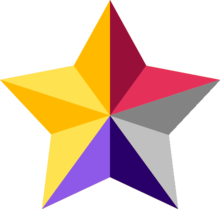


FIGURE 3.8 – Staruml logo

#### PHp Storm

PhpStorm est un environnement de développement intégré (IDE) innovant basé sur Java conçu par JetBrains pour les développeurs PHP et Web.

Il prend en charge PHP 5.3 / 5.4 / 5.5 / 5.6 / 7.0 / 7.1 / 7.2 / 8.0



FIGURE 3.9 – Visual studio Code logo



FIGURE 3.10 – XAMP Control Panel

#### XAMPP

XAMPP est un ensemble de logiciels permettant de configurer un serveur Web local, un serveur FTP et un serveur de messagerie. Il s’agit d’une distribution de logiciels libres ( X (cross) Apache MariaDB Perl PHP) offrant une bonne souplesse d’utilisation, connue pour son installation simple et rapide.

#### ShareLaTeX



FIGURE 3.11 – ShareLaTeX logo

ShareLaTeX est un éditeur LaTeX et un compilateur PDF en ligne, collaboratif et en temps réel. De plus, ShareLaTeX est sorti en février 2014. Le 20 juillet 2017, ShareLatex a été racheté par Overleaf5 qui a réuni les services d’Overleaf et de Sharelatex sur une seule plateforme. La fusion des deux services au sein d’Overleaf v2 s’est achevée en septembre 2018.

## 3.4 Spécification Architectural

### 3.4.1 Architecture physique

Notre application web est basée sur une architecture physique appelée 3eres archietcture

L’architecture à trois niveaux est un modèle en couches, c’est-à-dire que chaque couche ne communique qu’avec ses couches adjacentes (haut et bas) et le flux de la commande parcourt le système de haut en bas; les couches supérieures contrôlent le les couches inférieures, c’est-à-dire les couches supérieures sont toujours des sources d’interaction ( clients ) tandis que les couches inférieures ne répondent qu’aux demandes (les serveurs).

Dans l’architecture à 3 niveaux, il existe un niveau intermédiaire, c’est-à-dire qu’il existe généralement une architecture partagée entre :

\*Le consommateur le demandeur de ressources

* Le serveur d’application

(également appelé middleware) le serveur chargé de fournir la ressource mais en appelant un autre serveur

* Le serveur secondaire

(généralement un serveur de base de données), fournissant un service au premier serveur

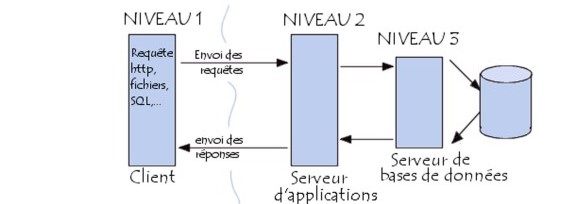


FIGURE 3.12 – architecture 3 tiers

### 3.4.2 Architecture logique

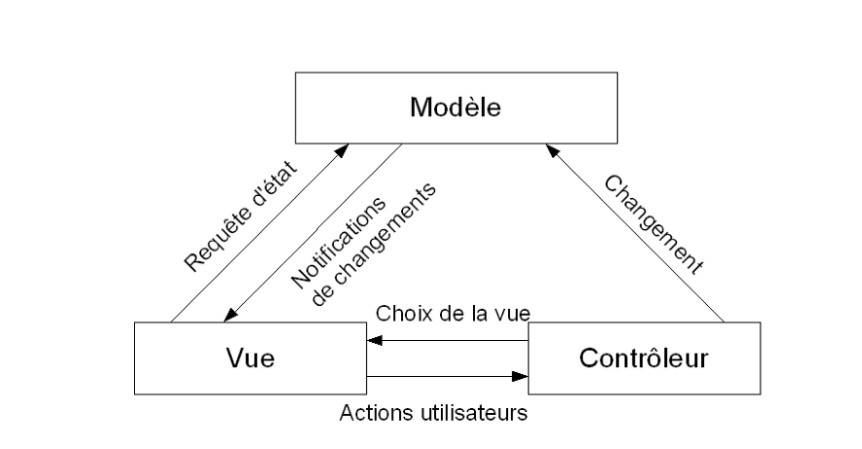
l’architecture à 3 niveaux utilisée par notre application web est l’architecture MVC

Cette dernière est l’une des architectures logicielles les plus utilisées pour les applications web, elle se compose de 3 modules :

Modèle : cœur de l’application qui gère les données, récupère les informations de la base de données, les organise pour qu’elles puissent ensuite être traitées par le contrôleur.

Vue : composant GUI utilisé pour présenter les données du modèle à l’utilisateur.

Contrôleur : composant responsable de la prise de décision, gère la logique du code qui prend les décisions, il est l’intermédiaire entre le modèle et la vue.



F

IGURE

3.13

–

architecture

MVC

## 3.5 Environnement de travail

### 3.5.1 Environnement matériel

Durant notre projet, j’ai mis une ordinateur portatif "hp".

Qui possède les caractéristiques suivantes :

— Processeur : Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz 2.90 GHz.

— Mémoire installée (RAM) : 8.00 Go.

— Type du systéme : Windows 10 professionnel 64 bits.

## Conclusion

Dans ce chapitre, nous avons présenté le backlog général de l’application , les diagrammes de conceptions et l’environnement de travail utlisé durant la phase de développement .

Dans le chapitre suivant, nous présentons le sprint zéro de l’application consacré à tous les utilisateurs .

**Chapitre 4**

Rôle Utilisateur

33

## Introduction

Après avoir terminé le sprint zéro de notre projet, nous pouvons maintenant passer au premier sprint qui se focalise sur le module "authetification" , c’est la partie primoridiale de toute application web . nous allons parler de l’authentification et de l’autorisation, puis nous verrons comment permettre aux utlisateurs de s’inscrire sur notre site à travers un formulaire.

### 4.0.1 Sprint Backlog

#### 4.0.1.1 Utilisateur

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sprint | id fea-  tures | features | id story | User Story | Priorité |
| Utilisateur | 1 | Inscription | 1.1 | En tant que manager, je veux pouvoir m’inscrire à Assurance Auto. | 100 |
| 1.2 | En tant qu’admin, je veux pouvoir m’inscrire Assurance Auto | 99 |
| 1.3 | En tant que souscripteur, je veux pouvoir m’inscrire Assurance Auto . | 98 |
| 1.4 | En tant qu’agent d’assurance , je veux pouvoir m’inscrire à Assurance Auto | 97 |
| 2 | Authen-  tification | 2.1 | En tant que manager, je veux pourvoir me connecter à Assurance AUto avec tous les drois. | 96 |
|  |  |  | 2.2 | En tant qu’admin, je veux pourvoir me connecter à Assurance Auto avec tous les  drois | 95 |
| 2.3 | En tant qu’agent d’assurance, je veux pouvoir me connecter à Assurance Auto avec tous les droits. | 94 |
| 2.4 | En tant ue souscripteur, je veux pourvoir me connecter à Assurance AUto avec tous  les drois | 93 |

TABLE 4.1: Product Backlog

#### 4.0.1.2 L’objectif du sprint

L’objectif de ce premier sprint est de savoir utiliser le composant Security Bundle pour créer l’authentification et l’enregistrement

## 4.1 Analyse

#### 4.1.0.1 Description textuelle de cas d’utilisation "Inscription"

Les tableau ci-dessous montrent des descriptions textuelles des cas d’utilisation raffinés "Inscription".

|  |  |
| --- | --- |
| Acteur | Utilisateur ( Soit Manager, administrateur , agent d’assurances ou souscripteur) |
| Objectif | S’inscrire. |
| Pré-condition | : Disponiblité d’accès au réseau internet et navigateur en bon état de fonctionnement |
| Post-condition | : L’utilisateur s’est inscrit . |
| Scénario nominal | : L’UC commence lorsque l’utilisateur demande de s’inscrire et le processus se poursuit selon les étapes suivantes 1 L’utilisateur demande à s’inscrire :   1. Le système affiche le formulaire d’inscription. 2. L’utilisateur remplie les champs du formulaire. 3. Le système choisie le role . 4. L’utlisateur choisie son role dans l’application puis il valide 5. Le système vérifie si un utilisateur n’est pas déja inscrit avec la même adresse email ou le même nom puis il enregistre le nouveau compte |
| Scénario d’échec | Lors de la vérification , si le système trouve que l’utlisateur est déja inscrit , il peut créer un compte avec la même adresse mail ou le même nom |

#### 4.1.0.2 Description textuelle de cas d’utilisation "Authentifica-

#### tion"

Le tableau ci-dessous montrent des descriptions textuelles des cas d’utilisation raffinés "Authentification".

|  |  |
| --- | --- |
| Acteur | Utilisateur ( Soit Manager, administrateur , agent d’assurances ou souscripteur) |
| Objectif | S’authentifier. |
| Pré-condition | : Inscription |
| Post-condition | : L’utilisateur s’est authentifié et l’interface adéquate se charge. |
| Scénario nominal | : L’UC commence lorsque l’utilisateur demande de s’authentifier et le processus se poursuit selon les étapes suivantes 1 L’utilisateur demande l’accès à l’application. :   1. Le système affiche à l’utilisateur un formulaire à remplir dans lequel il saisit son login et son mot de passe.. 2. L’utilisateur saisit le login et le mot de passe. 3. Le système vérifie l’existence des données saisies dans la base de données et lance l’application en cas du succès. |
| Scénario d’échec | Une alternative d’échec est lorsque le login ou le mot de passe sont erronés.. |

#### 4.1.0.3 Diagramme d’activité "Inscription"

représente le cycle de vie de l’inscription d’un utilisateur, montre la séquence d’actions et de décisions au sein d’une activité.

La figure 4.1 présente le diagramme d’activité "inscription"

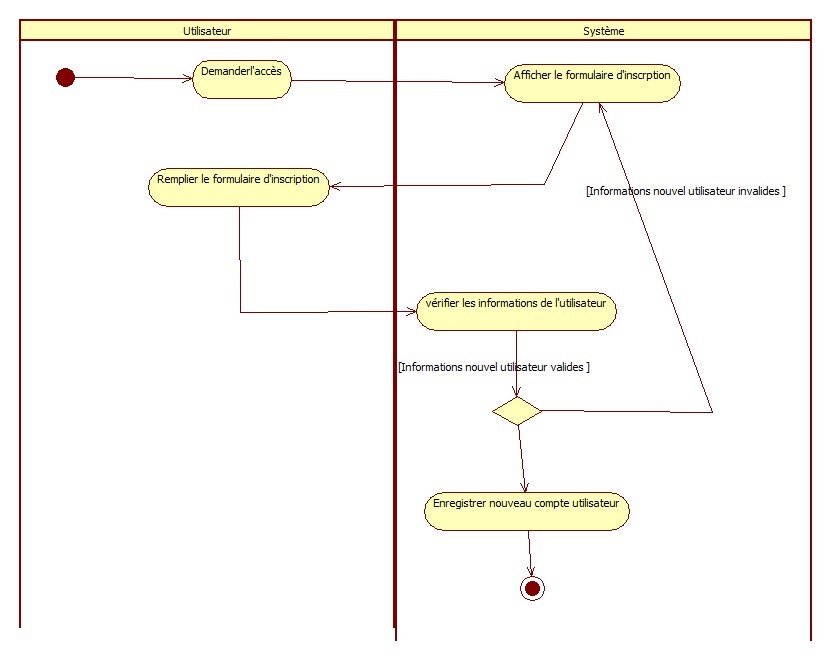


FIGURE 4.1 – Diagramme d’activité "Inscription"

#### 4.1.0.4 Diagramme d’activité "Authentification"

La figure 4.2 présente le diagramme d’activité "Authentification"

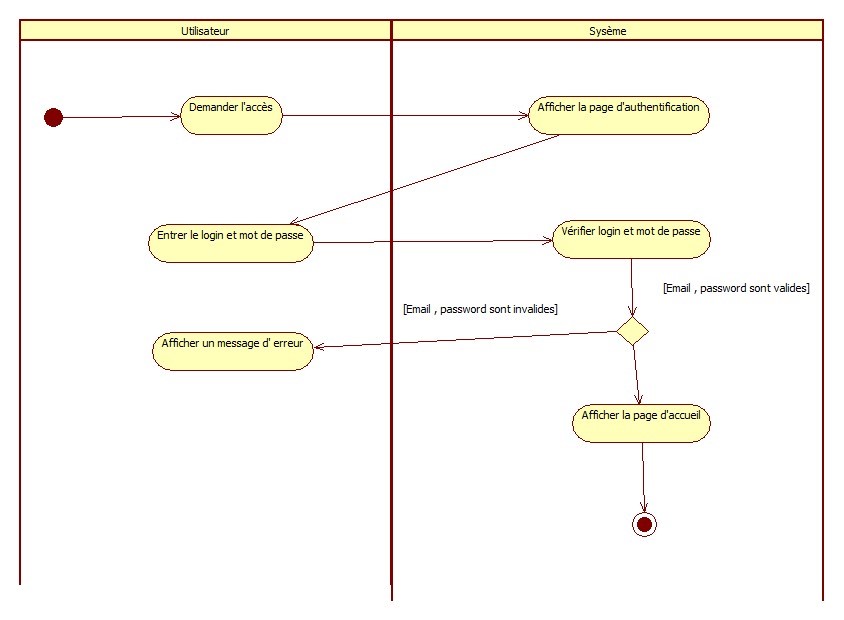


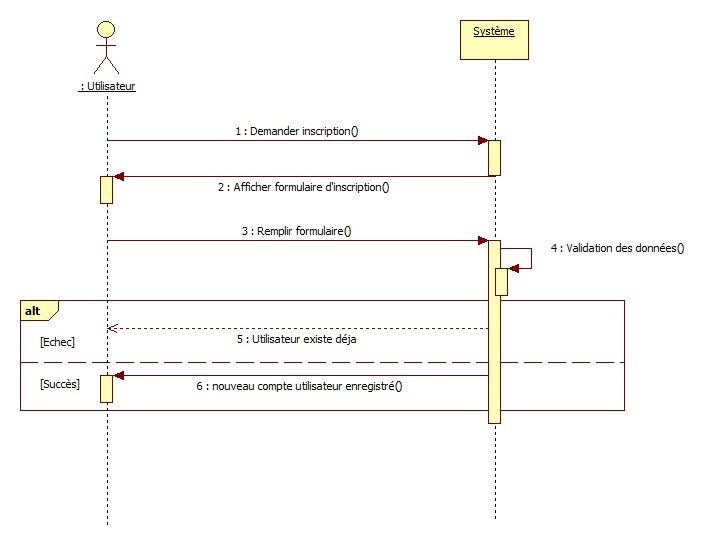
FIGURE 4.2 – Diagramme d’activité "Authentification"

#### 4.1.0.5 Diagramme de séquence système "Inscription"

Pour shématiser la vie comportemental de notre système , nous faisans recours aux diagrammes de séquence .

Ces diagrammes permettent de présenter l’interactions entre l’acteur et le système avec des messages présentés dans un ordre chronologiques .

FIGURE 4.3 – Diagramme de séquence système "s’inscrire"



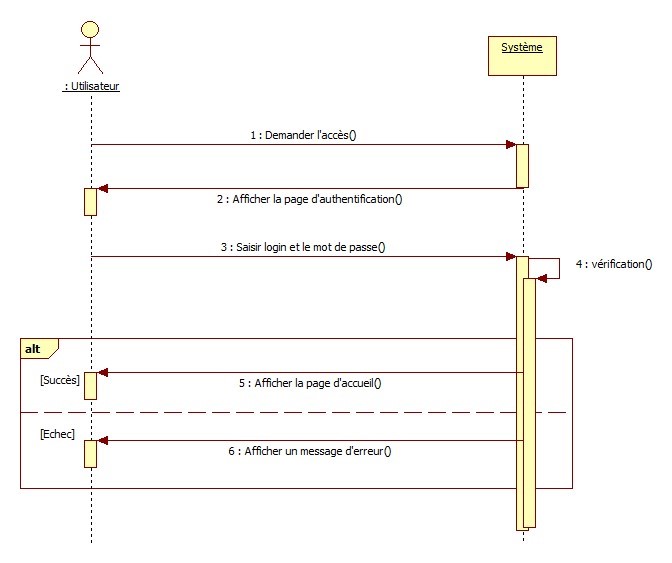
#### 4.1.0.6 Diagramme de séquence système "Authentification"

Avant d’atteindre la phase d’authentification, chaque utilisateur doit saisir son identifiant et son mot de passe.

Ensuite, tout au long de sa navigation, il ne peut accéder qu’aux services pour lesquels il est autorisé.

Le diagramme 4.4 montre les séquences à effectuer pour démarrer la phase d’authentification

FIGURE 4.4 – Diagramme de séquence système "s’authentifier"

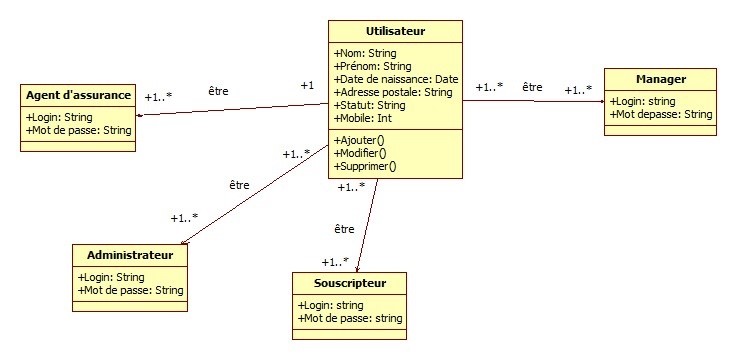


## 4.2 Conception

#### 4.2.0.1 Diagramme de classe "Utilisateur"

La figure 4-5 ci-dessous présente le diagramme de classe " Utilisateur"

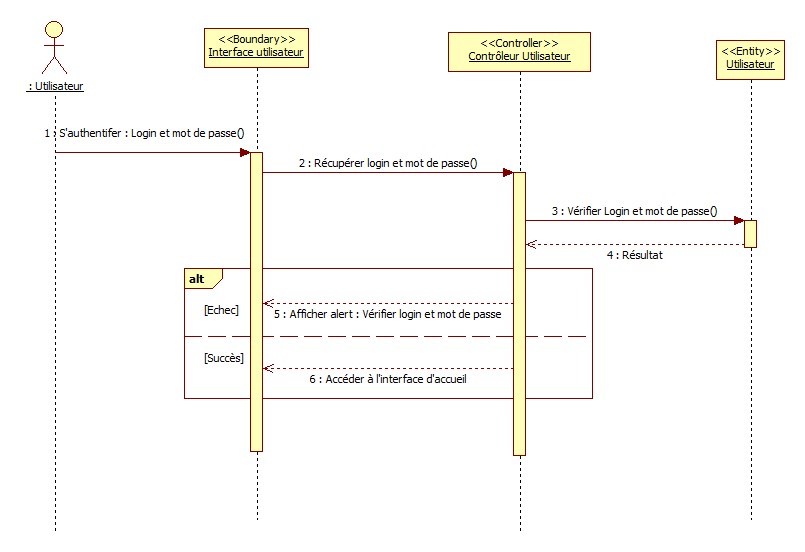
FIGURE 4.5 – Diagramme de classe "Utilisateur"



#### 4.2.0.2 Diagramme de séquence objet "Authentification"

Par rapport aux diagrammes de séquence du système, nous remplacerons le système vu comme une boîte noire par un ensemble d’objets en interaction

La figure 4-6 représente le diagramme de séquence objet« Authentification ».



F

IGURE

4.6

–

Diagramme

d’activité

"Authentification"

## 4.3 Implémentation de premier module

### 4.3.1 Interface d’authentification :

L’utilisateur doit saisir son login et son mot de passe, puis appuyer sur le bouton "se connecter".

En effet, le processus approprié pour accéder à l’application, l’utilisateur doit saisir les paramètres d’un compte valide dans l’interface d’authentification représentée sur la figure 4-7, sinon, l’interface empêchera l’accès à l’application.

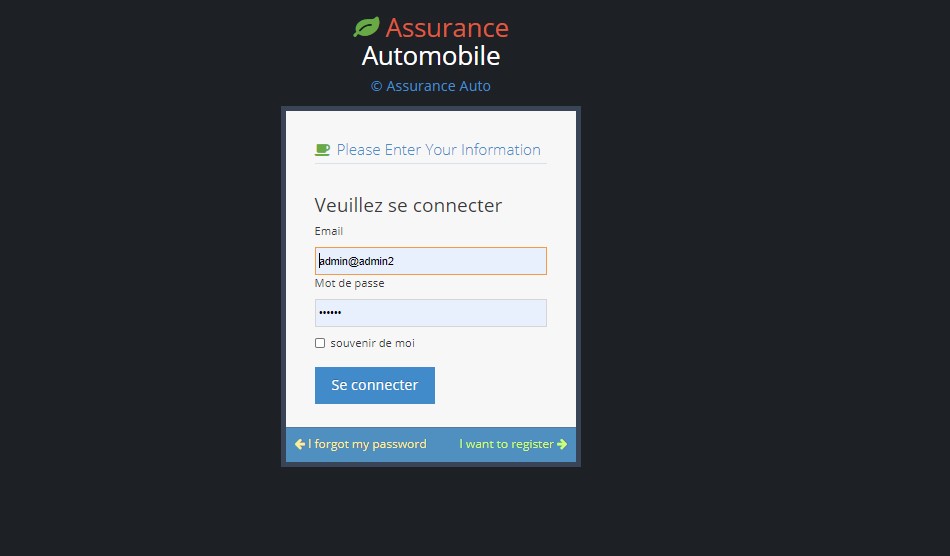


FIGURE 4.7 – Interface authentification

## Conclusion

Au cours de ce chapitre, nous avons réussi à produire le premier livrable du projet qui implique l’authentification.

Dans le chapitre suivant, tous les efforts seront consacrés à la production d’un nouveau livrable couvrant le rôle du Manager.

**Chapitre 5**

Sprint Deux : Rôle Manager

45

## Introduction

Nous pouvons maintenant passer au deuxième Sprint. Nous allons définir les tâches à réaliser .

l’objectif est de nous concentrer sur le développement de la partie consacrée au role du manager .

## 5.1 Sprint Backlog

### 5.1.1 Histoires à réaliser

La liste des tâches à réaliser dans le sprint deux est décrite dans le tableau ??.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id story | User story | id tâche | tâche | estimation(H) |
| 3 | Gestion utlisateur | 3.1 | En tant que manager que je veux ajouter un compte utilisateur | 8 |
| 3.2 | En tant que manager je veux modfier un compte utilisateur | 8 |
| 3.3 | En tant que manager je veux consulter la liste comptes utlisateurs | 8 |
| 3.4 | En tant qu’admin que je veux supprimer un compte utilisateur | 8 |

TABLE 5.1: Liste des tâches du deuxième Sprint

### 5.1.2 Objectif du sprint

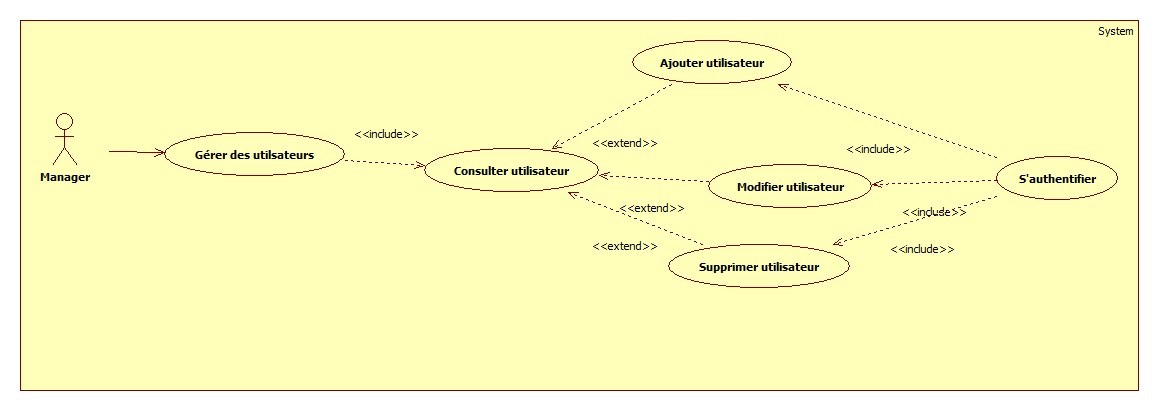
L’objectif de ce deuxième sprint est la gestion des comptes des utlisateurs .

## 5.2 Analyse

Nous choisissons de présenter tout d’abord le diagramme de cas d’utilisation détaillés afin de limiter la complexité de ce modèle.

#### 5.2.0.1 Diagramme de cas d’utilisation "Gestion utilisateur"

La figure suivantes présente les diagrammes de cas d’utilisation de la partie applicative qui présente l’action : gestion des comptes utlisateurs



F

IGURE

5.1

–

Diagramme

d’activité

"Authentification"

#### 5.2.0.2 Description textuelle de cas d’utilisation "Gestion utilisateur"

|  |  |  |
| --- | --- | --- |
| Acteur | Manager | |
| Objectif | L’UC est utilisé pour ajouter, modifier des informations d’un utilisateur. | |
| Pré-condition | : S’authentifier | |
| Post-condition | :La liste des utilisateurs est mise à jour selon la demande de l’utilisateur ( Manager ). | |
| Scénario nominal | :Le UC commence lorsque le manager accède au menu "Utilisateurs" depuis le menu principal et le processus se poursuit selon les étapes suivantes : a Demande d’ajout : :  1.L’utilisateur demande l’ajout d’un nouvel utilisateur en cliquant sur le bouton "Ajouter".   1. Le système affiche le formulaire d’ajout d’utilisateurs et donne un coup de main au gestionnaire pour entrer les informations d’un nouvel utilisateur. 2. Le manager d’application saisit les données nécessaires concernant l’utilisateur : il insère le nom, prénom, login, mot de passe, adresse email, choisissez le profil dans la liste et validez.   4.Le système vérifie que le formulaire est rempli.  b Validation de l’ajout :  Scénario de succès :Le système ajoute le nouvel utilisateur dans la base de données  Scénario d’échec : le système n’autorise pas l’ajout du nouvel utilisateur.  Scénario alternatif :  Modifier les informations d’un utilisateur : c) Demande de modification :   1. Le système affiche la liste des utilisateurs. 2. Le responsable demande à modifier les informations d’un utilisateur en cliquant sur le bouton «Modifier» correspondant à cet utilisateur. 3Le système affiche le formulaire de modification et donne la main au gestionnaire pour modifier les informations relatives à cet utilisateur.. d Validation de la modification :   Scénario de succès : Le système affiche la modification apportée à l’utilisateur dans la liste des utilisateurs.  Scénario d’échec : Le système ne permet pas la modification de l’utilisateur (Modification impossible).  Supprimer un utilisateur : e Demande de suppression :   1. Le système affiche la liste des utilisateurs 2. Le manager demande de supprimer un utilisateur en cliquant sur le bouton « Supprimer » correspondant à cet utilisateur. 3. Un message de vérification s’affiche au manager pour confirmer la suppression de l’utilisateur. 4. Le manager confirme la suppression.. e Validation de la suppression :   Le système affiche la liste des utilisateurs après la mise à jour effectuée. | |
|  | | Scénario d’échec possibles :  Le système affiche « Utilisateur existe déjà » et donne la main au manager de ressaisir les informations personelles du nouvel utilisateur Le système affiche le message « Champ vide » en cas d’un champ non rempli et retourne à la position du champ manquant.  Le système affiche le message « Champ vide » en cas d’un champ non rempli et retourne à la position du champ manquant. | |
| Scénario d’échec | | Une alternative d’échec est lorsque le login ou le mot de passe sont erronés.. | |

#### 5.2.0.3 Diagramme d’activité "Ajout utilisateur"

Comme illustré par la figure ci-dessous, le diagramme d’activité « Ajouter utilisateur » représente le cycle de vie de l’ajout d’un utilisateur, montre l’enchaînement des actions et

FIGURE 5.2 – Diagramme d’activité "Ajout utilisateur"

décisions

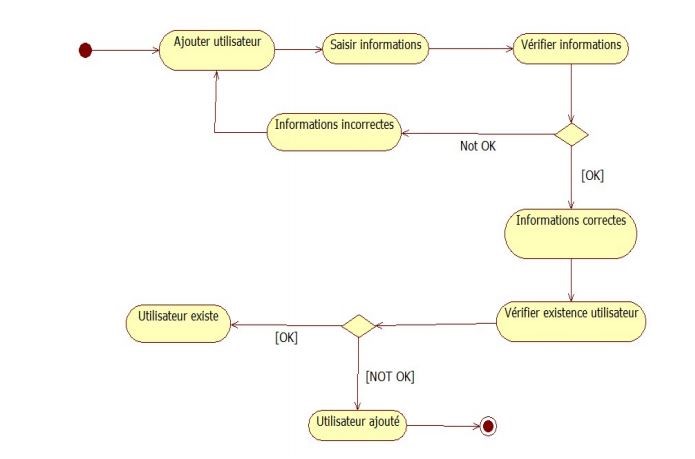
au

sein

d’une

activité

.



#### 5.2.0.4 Diagramme d’états-transitions

LLes diagrammes d’états UML décrivent le comportement interne d’un objet utilisant une machine à états finis. Ils présentent les séquences d’états possibles et actions qu’une instance de classe peut traiter au cours de son cycle de vie en réponse à événements discrets En effet, le diagramme d’état décrit toute l’existence d’un objet, donc que toutes ses actions liées à la transmission et à la réception de messages. Un graphique d’états représente donc la logique interne associée à chaque objet. Les diagrammes de transition d’état sont basés sur 3 concepts :

* Un état : se caractérise par sa durée et sa stabilité, il représente une conjonction instantané des valeurs d’attribut d’un objet.
* Une transition : représente le mouvement qu’un objet effectue pendant les états c’est-àdire le mécanisme qui pousse un objet à quitter un état pour passer à un nouveau Etat.
* Expressions d’action : correspondent aux activités effectuées par un objet. Le les expressions d’action peuvent être associées à des états ou des transitions

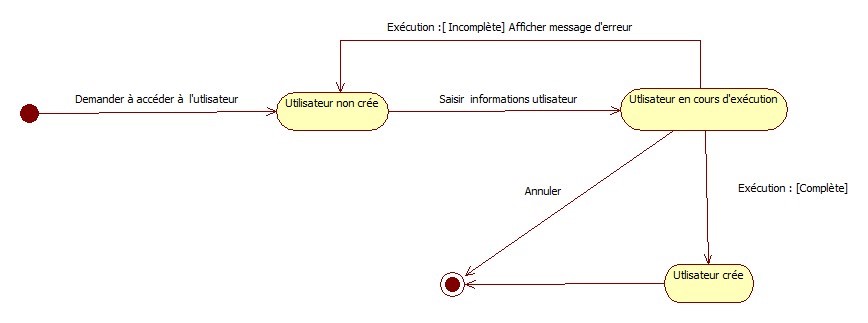
\* Diagramme d’état "Ajouter un utilisateur" :

Comme illustré par la figure ci-dessous, le diagramme de transition d’état "Utilisateur" représente le cycle de vie de la classe d’utilisateurs, ainsi que ses états possibles et les tran-

FIGURE 5.3 – Diagramme d’état de transition "Ajout utilisateur"

sitions

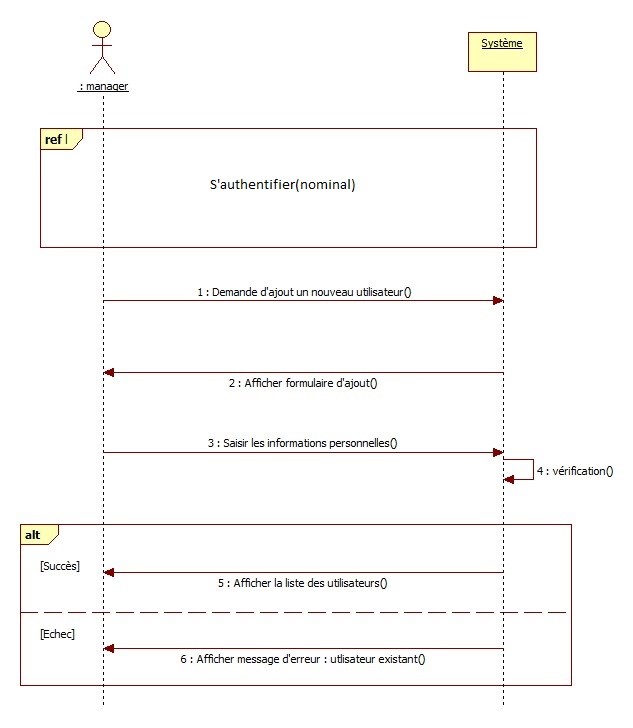
nécessaires.



#### 5.2.0.5 Diagramme de séquence système "Ajout utilisateur"

Dans la figure 5-4, nous traiterons le cas où le manager ajoute un nouvel utilisateur pour l’insérer dans la base de données. Le processus commence lorsque le manager demande l’accès au formulaire d’ajout d’utilisateur, puis le système affiche ce formulaire et donne la main au manager pour ajouter un nouvel utilisateur, ce dernier entre les données nécessaire qui sera vérifié par le système. À la fin de ce processus, le nouvel utilisateur ajouté sera affiché dans la liste des utilisateurs.

FIGURE 5.4 – Diagramme de séquence système "Ajout utilisateur"

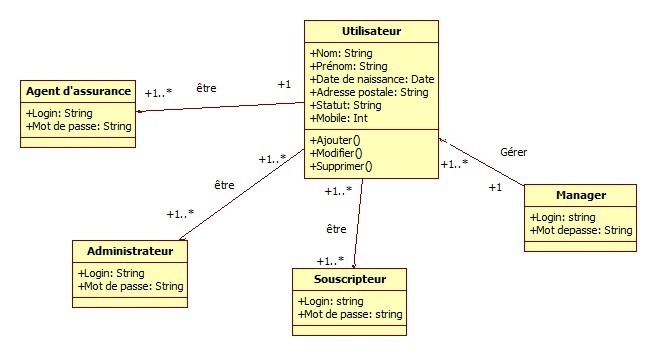


## 5.3 Conception

#### 5.3.0.1 Diagramme de classe "Manager"

La figure 5-5 ci-dessous présente le diagramme de classe " Manager

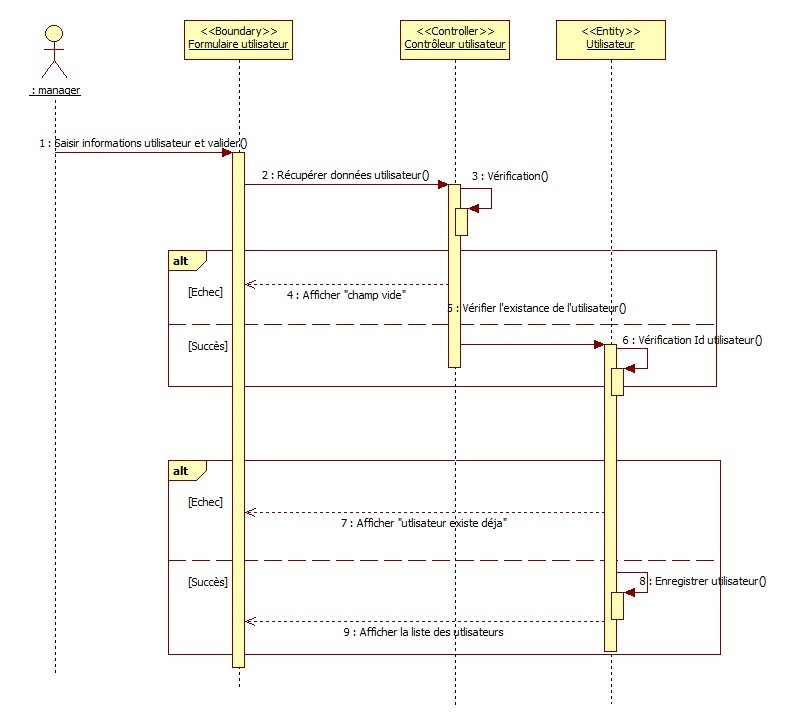
FIGURE 5.5 – Diagramme de classe "Manager"



#### 5.3.0.2 Diagramme de séquence objet "Ajout utilisateur"

La figure 5-6 présente les étapes qui permettent d’ajouter un utilisateur

FIGURE 5.6 – Diagramme de séquence objet "Ajout utilisateur"



## 5.4 Implémentation

Dans cette partie, on va présenter l’interface Homme-Machine implémenté au niveau du deuxième Sprint.

### 5.4.1 Interface Manager

1. Interface "Liste utilisateurs" :

L’interface de la figure 5-7 s’ouvre lorsque le manager clique sur le menu « Gestion Users

».

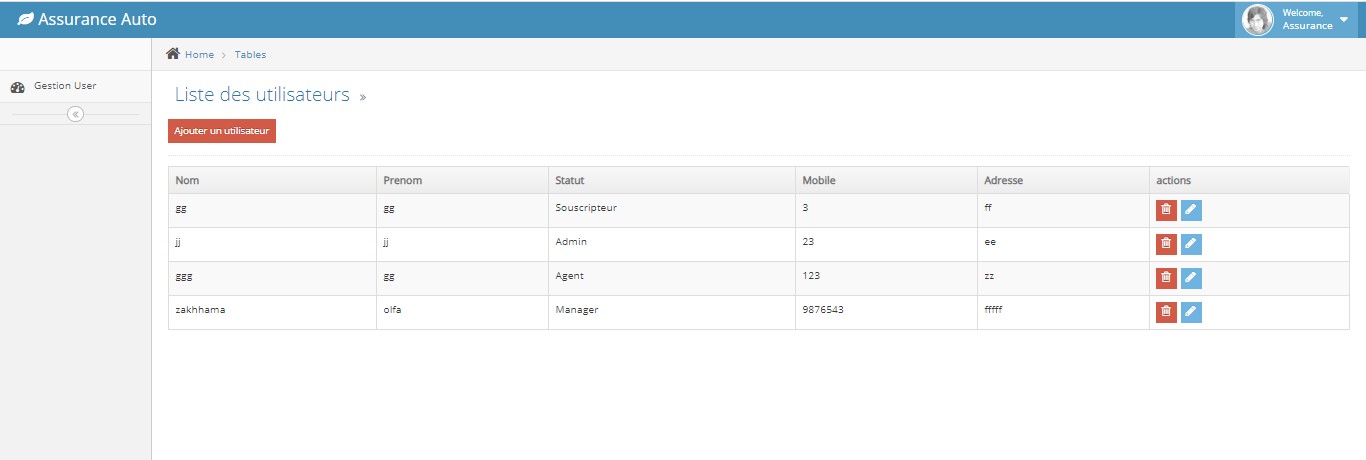


FIGURE 5.7 – Interface liste des utilisateurs

Cette interface contient la liste de tous les utilisateurs, nous pouvons les gérer dans le but de la modification ou de la désactivation ou de l’activation qui peuvent être consultés.

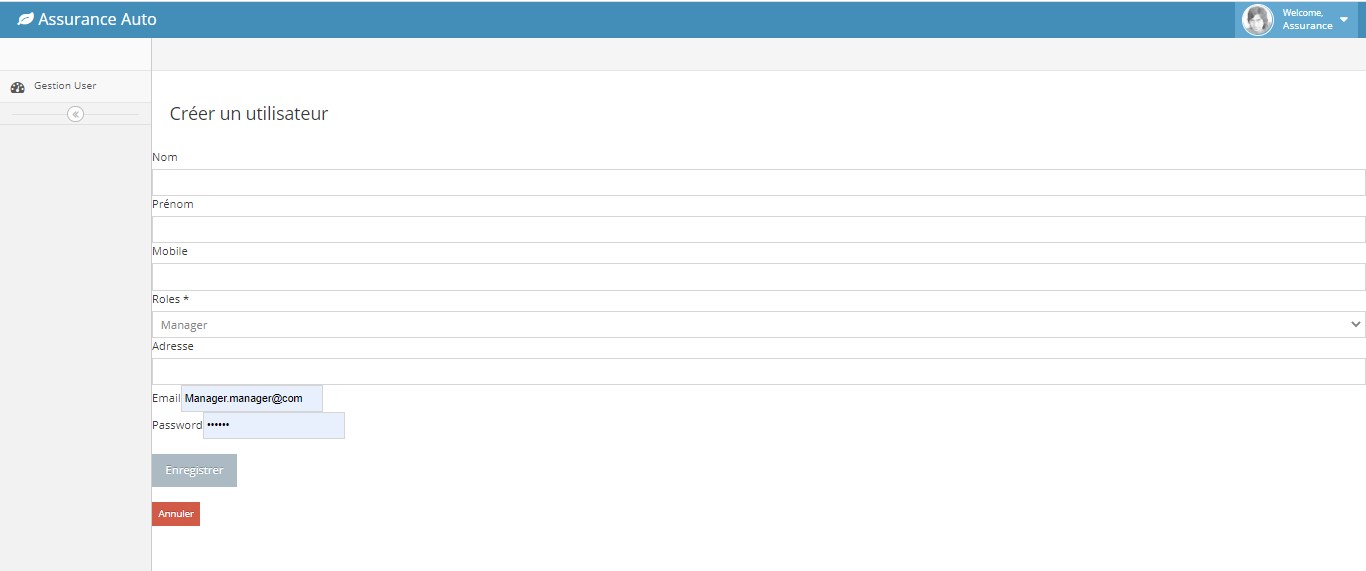
Le manager peut donc :

* Ajouter un utilisateur : en cliquant sur le bouton "Ajouter", le "Formulaire user »s’affichera comme illustré à la Figure 5-8.
* Modifier un utilisateur : s’il clique sur le bouton "Modifier" les informations relatifs à l’utilisateur choisi seront affichés dans l’interface du formulaire utilisateur, le responsable doit les modifier et cliquer sur valider pour les enregistrer.

2 . Interface "Créer utilisateur" :

Lorsque le manager clique sur le bouton « Ajouter un utilisateur » l’interface présentée de la figure 5-7 ci-dessous sera affichée.

FIGURE 5.8 – Interface Créer utilisateur



Cette interface donne la main pour ajouter un nouvel utilisateur qui est défini obligatoirement par son nom, prénom, email, login, mot de passe et en l’affectant un profil.

## Conclusion

Au cours de ce chapitre, nous avons réussi à produire le deuxième livrable du projet qui comporte la gestion des utilisateurs .

**Chapitre 6**

**Sprint Trois : Rôle Admin**

56

## Introduction

Nous pouvons maintenant passer au troisième Sprint. Nous allons définir les tâches à réaliser .

l’objectif est de nous concentrer sur le développement de la partie dédiée au rôle de l’admin de l’application .

## 6.1 Sprint Backlog

### 6.1.1 Histoires à réaliser

La liste des tâches à réaliser dans le sprint trois est décrite dans le tableau ??.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id story | User story | id tâche | tâche | estimation(H) |
| 4 | Gestion compagnie | 4.1 | En tant qu’admin que je veux ajouter une compagnie | 8 |
| 4.2 | En tant qu’admin que je veux modifier une compagnie | 8 |
| 4.3 | En tant qu’admin que je veux supprimer une compagnie | 8 |
| 4.4 | En tant qu’admin que je veux consulter la liste des compagnies | 8 |
| 5 | Gestion type de contrat | 5.1 | En tant qu’admin que je veux ajouter un type de contrat | 8 |
| 5.2 | En tant qu’admin que je veux modifier un type de contrat | 8 |
| 5.3 | En tant qu’admin que je veux supprimer un type de  contrat | 8 |
| 5.4 | En tant qu’admin que je veux consulter la liste des types de contrat | 8 |
| 6 | Gestion garantie | 6.1 | En tant qu’admin que je veux ajouter une garantie | 8 |
| 6.2 | En tant qu’admin que je veux modifier une garantie | 8 |
|  |  | 6.3 | En tant qu’admin que je veux supprimer une garantie | 8 |
| 6.4 | En tant qu’admin que je veux consulter la liste des garanties | 8 |
| 7 | Gestion clause | 7.1 | En tant qu’admin que je veux ajouter une clause | 8 |
| 7.2 | En tant qu’admin que je veux modifier une fréquence | 8 |
| 7.3 | En tant qu’admin que je veux supprimer une fréquence | 8 |
| 7.4 | En tant qu’admin que je veux consulter la liste des fréquences de périodictés | 8 |
| 8 | Gestion fréquence de périodcité | 8.1 | En tant qu’admin que je veux ajouter une fréquence de pé-  riodicité | 8 |
| 8.2 | En tant qu’admin que je veux modifier une fréquence | 8 |
| 8.3 | En tant qu’admin que je veux supprimer une fréquence | 8 |
| 8.4 | En tant qu’admin que je veux consulter la liste des fréquences de périodictés | 8 |
| 9 | Gestion mode de paiement | 9.1 | En tant qu’admin que je veux ajouter un mode de paiement | 8 |
| 9.2 | En tant qu’admin que je veux modifier un mode de paiement | 8 |
| 9.3 | En tant qu’admin que je veux supprimer un mode de paiement | 8 |
| 9.4 | En tant qu’admin que je veux consulter la liste des modes de paiement | 8 |

TABLE 6.1: Liste des tâches du troixième Sprint

### 6.1.2 Objectif du sprint

L’objectif de ce deuxième sprint est la réalisation du module gérer par l’admin

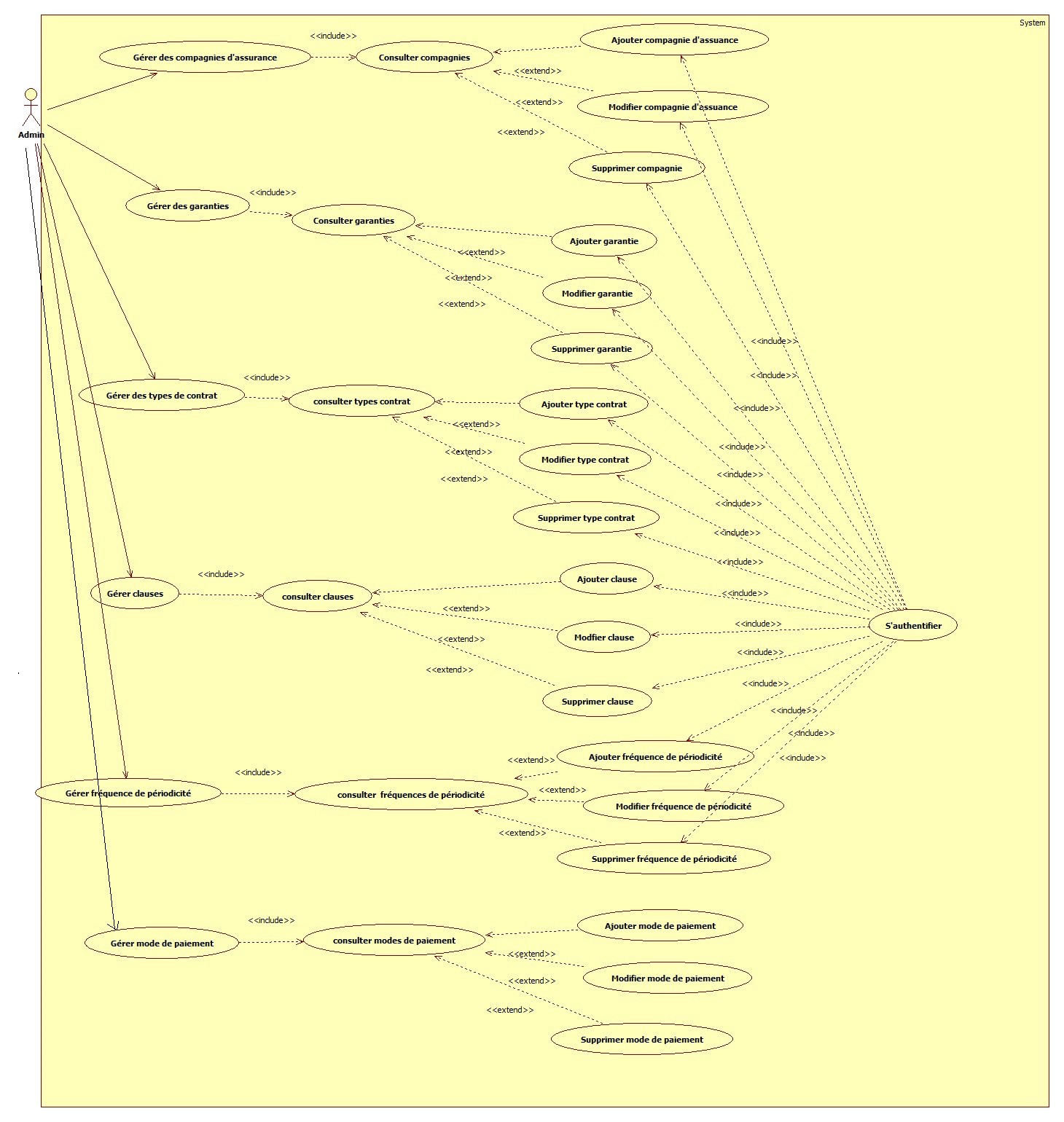
## 6.2 Analyse

Nous choisissons de présenter tout d’abord le diagramme de cas d’utilisation détaillés afin de limiter la complexité de ce modèle.

### 6.2.1 Diagramme de Cas d’utilisation "Admin"

La figure 6.1 présente le diagramme de cas d’utilisation de l’admin .

FIGURE 6.1 – Diagramme de cas d’utilisateur "Admin"



#### 6.2.1.1 Description textuelle de "Gérer les compagnies d’assurance "

|  |  |
| --- | --- |
| Titre | Gérer les compagnies |
| Acteur | Admin |
| Pré-condition | L’admin s’est authentifié |
| Post-condition | La liste des compagnies est mise à jour selon la demande de l’admin |
| Scénario nominal | Le diagramme de cas d’utilisation commence lorsque l’administrateur demande l’accès au volet «Gestion de l’entreprise» à partir du menu principal et le processus se poursuit avec les étapes suivantes a)Demande d’ajout :  1. L’utilisateur demande l’ajout d’une nouvelle compagnie d’assurance 2. Le système affiche le formulaire d’ajout des compagnies d’assurance et donne la main à l’administrateur pour saisir les informations d’une nouvelle compagnie  3 L’admin saisie le type de la compagnie b) validation de l’ajout |
| Scénario alternatif | Modifier le nom ou la date de fondation d’une compagnie c) Demande de modification :   1. Le système affiche la liste des compagnies 2. L’admin demande de modifer les informations d’une compagnie en cliquant sur le bouton "modifier" correspondant à cette compagnie 3 Le système affiche le formulaire de modification et donne la main à l’admin pour modifier les informations relatives à cette compagnie 4 l’admin saisit les nouvelles informations puis il clique sur le bouton   "valider"   1. validation de la modification   Le système affiche la modification effectuée  Supprimer une compagnie   1. Demande de suppression : 2. Le système affiche la liste des compagnie 3. L’admin demande de supprimer une compagnie en cliquant sur le bouton supprimer correspondant à cette compagnie |
| Scénario d’échec | Le système affiche " Format incorrect" au cas où le formats des champs sont incorrectes  Le système affiche " champ vide" en cas d’un champ nom rempli |

#### 6.2.1.2 Diagramme d’activité "Ajout compagnie"

Comme illustré par la figure ci-dessous, le diagramme d’activité "Ajouter compagnie " représente les étapes de l’ajout d’une compagnie d’assurance ainsi que la séquence d’actions

FIGURE 6.2 – Diagramme d’activité "Ajout compagnie"

et

de

décisions

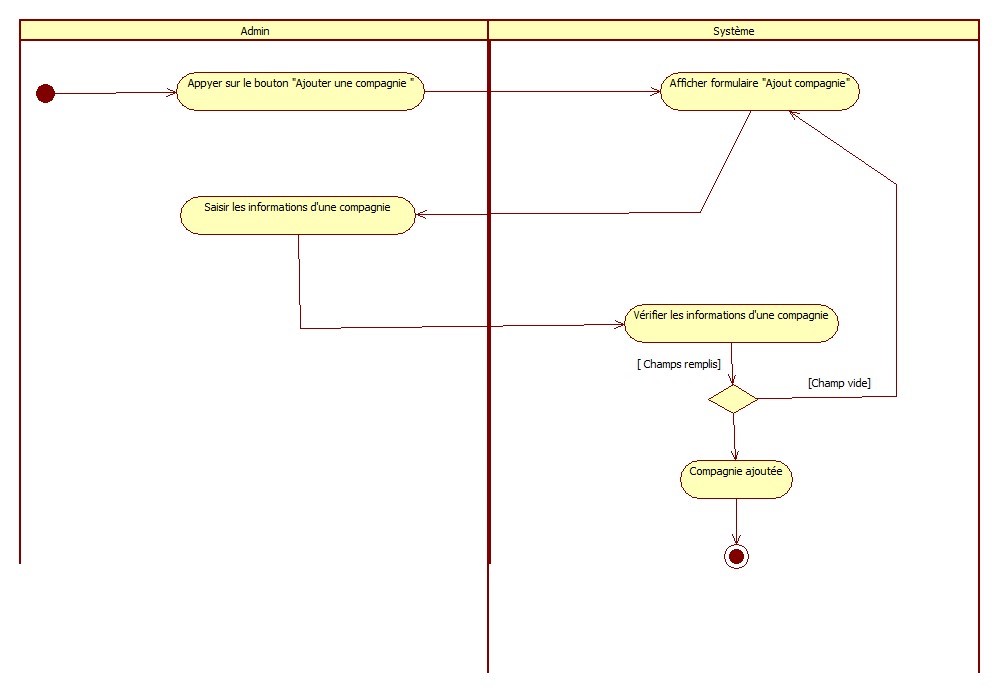
au

sein

de

cette

activité

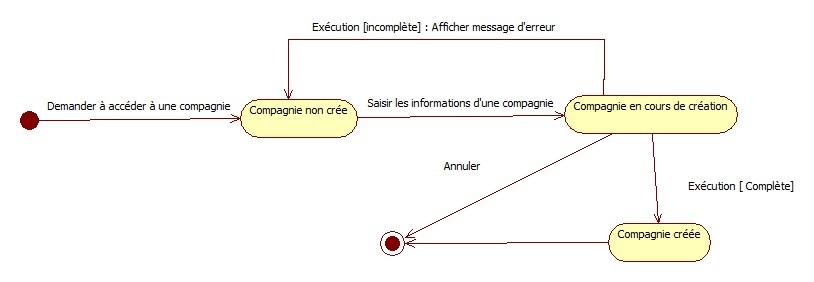


#### 6.2.1.3 Diagramme d’états-transitions

Comme illustré par la figure 6-3, le diagramme d’état-transition « AJout compagnie » représente le cycle de vie de la classe compagnie , ainsi ses états possibles et les transitions

FIGURE 6.3 – Diagramme d’état de transition "Ajout compagnie"

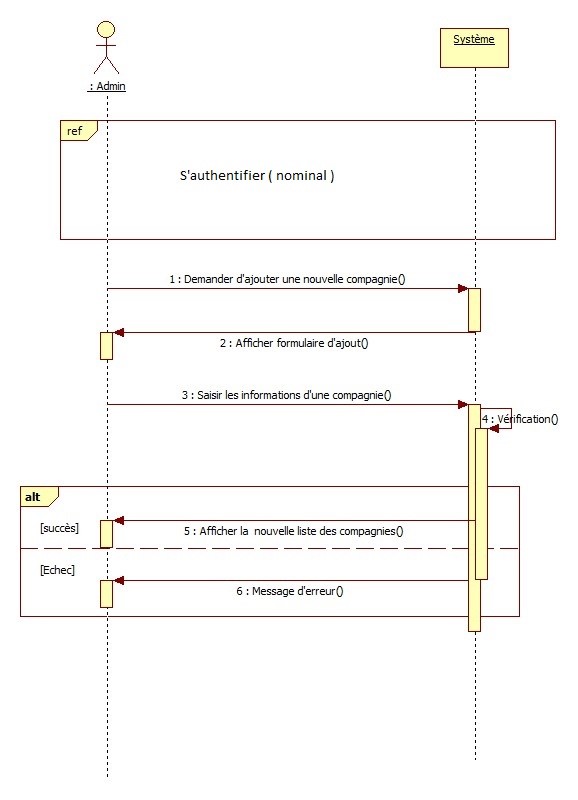
nécessaires.



#### 6.2.1.4 Diagramme de séquence système "Ajout compagnie"

Dans la figure 6-4, nous traiterons le cas où l’administrateur ajoute une nouvelle compagnie d’assurance pour l’insérer dans la base de données. Le processus commence lorsque l’administrateur demande l’accès au formulaire d’ajout d’entreprise, puis le système affiche ce formulaire et passe la main à l’administrateur pour ajouter une nouvelle société, l’administrateur entre les données nécessaire qui sera vérifié par le système. Au terme de ce processus, la nouvelle société ajoutée sera affichée dans la liste des compagnies d’assurance.

FIGURE 6.4 – Diagramme de séquence système "Ajout compagnie"

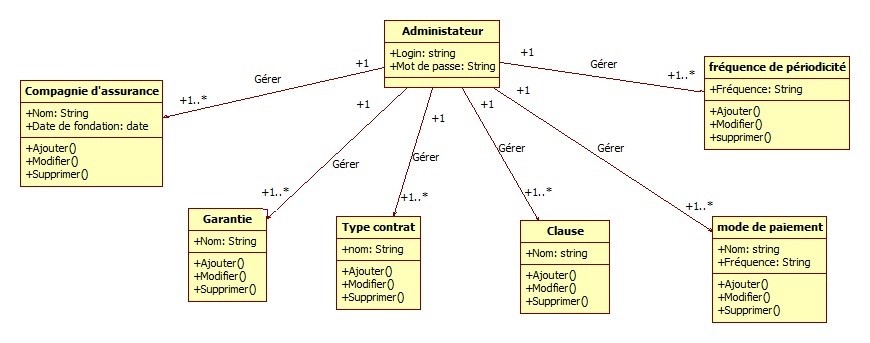


## 6.3 Conception

#### 6.3.0.1 Diagramme de classe "Admin"

La figure 6.5 présente le diagramme de classe de l’admininstrateur .

FIGURE 6.5 – Diagramme de classe "Admin"



#### 6.3.0.2 Diagramme de séquence objet "Ajout compagnie"

La figure 6-6 présente les étapes qui permettent d’ajouter une nouvelle compagnie d’assu-

rance.

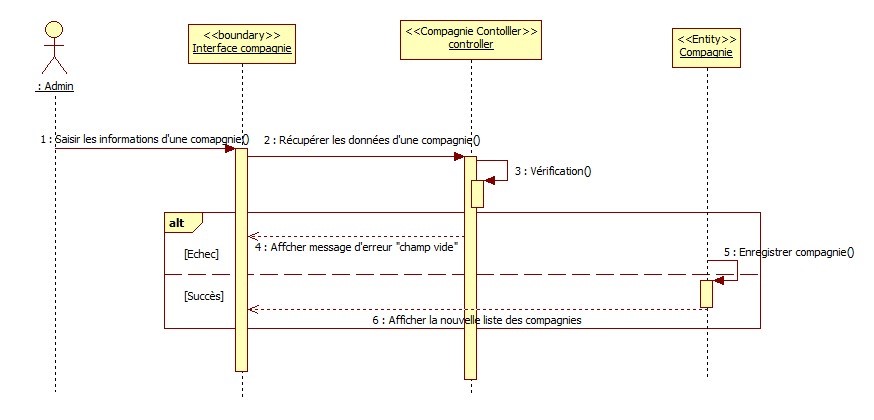


FIGURE 6.6 – Diagramme de séquence objet "Ajout compagnie"

## 6.4 Implémentation

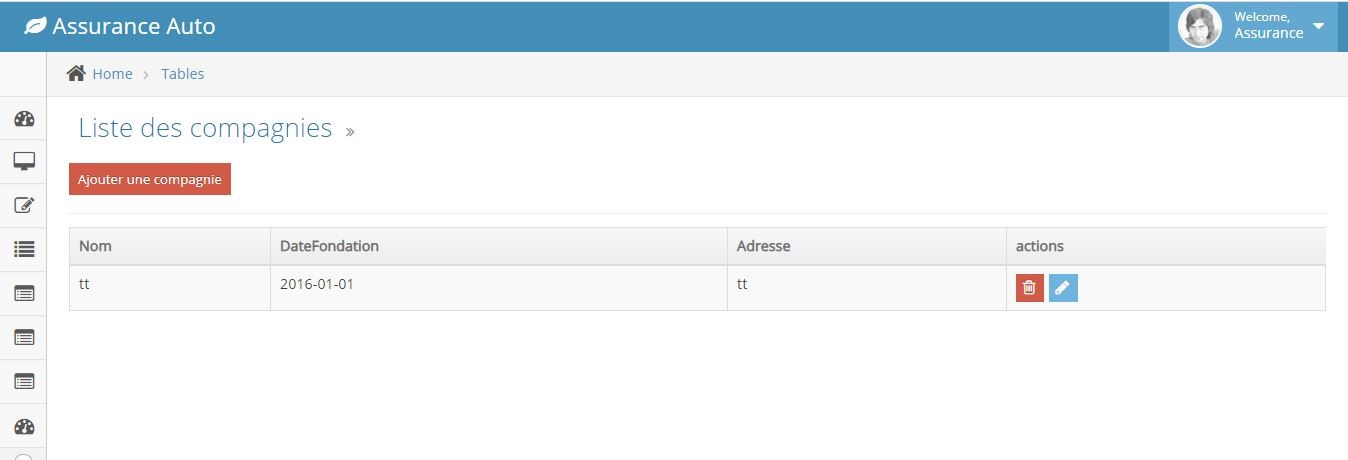
Dans cette partie, on va présenter l’interface Homme-Machine implémenté au niveau du troixième Sprint.

### 6.4.1 Interface Compagnie

1) Interface liste des compagnies :

Cette interface de la figure 6-7 s’ouvre lorsque l’administrateur clique sur le menu "Compagnies".

FIGURE 6.7 – Interface Liste des compagnies



Cette interface contient la liste de toutes les compagnies d’assurances, nous pouvons les gérer soit en les modifiant soit en les modifiant.

L’administrateur peut alors :

* Ajouter une agence : en cliquant sur le bouton « Ajouter », le « Formulaire compagnie » sera affiché (la figure 6-8).
* Modifier une compagnie : s’il clique sur le bouton « Modifier » les informations relatives à la compagnie d’assurance choisie seront affichées dans l’interface formulaire compagnie, l’admin doit les modifier et cliquer sur le bouton « Enregistrer » pour les enregistrer.

2) Interface Ajout compagnie :

Lorsque l’administrateur clique sur le bouton « Ajouter compagnie » l’interface présentée dans la figure 6-8 sera affichée.

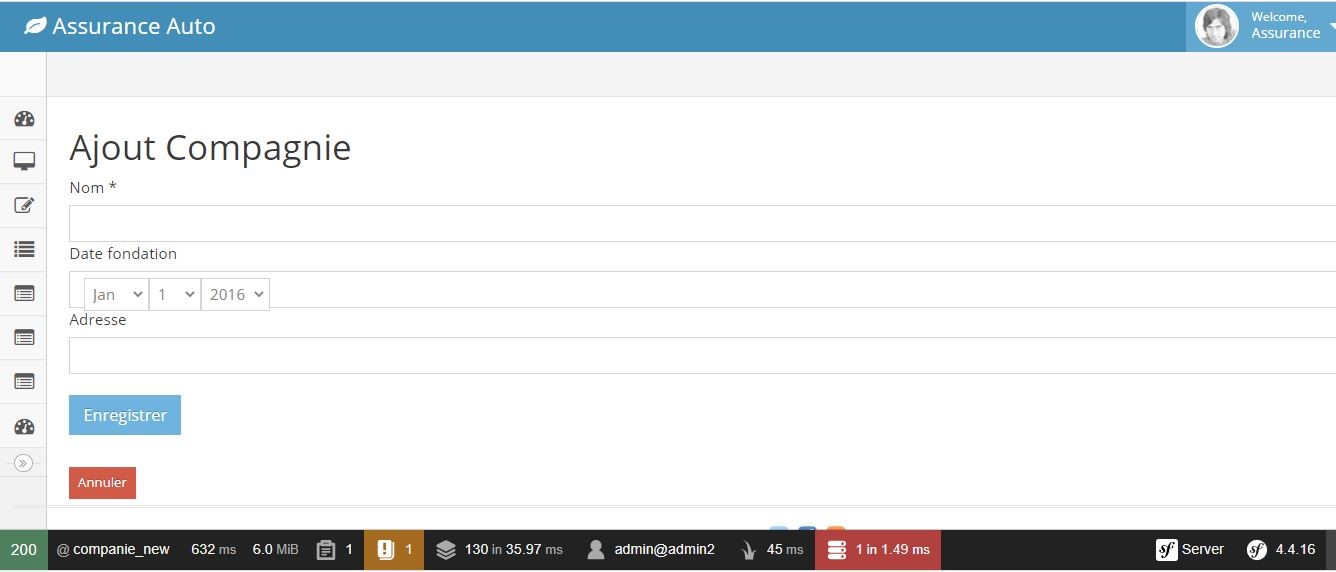


FIGURE 6.8 – Interface Ajout compagnie

### 6.4.2 Interface type contrat

1) Interface liste des types de contrat :

L’interface de la figure 6-9 s’ouvre lorsque l’administrateur clique sur le menu «Type de contrat ».

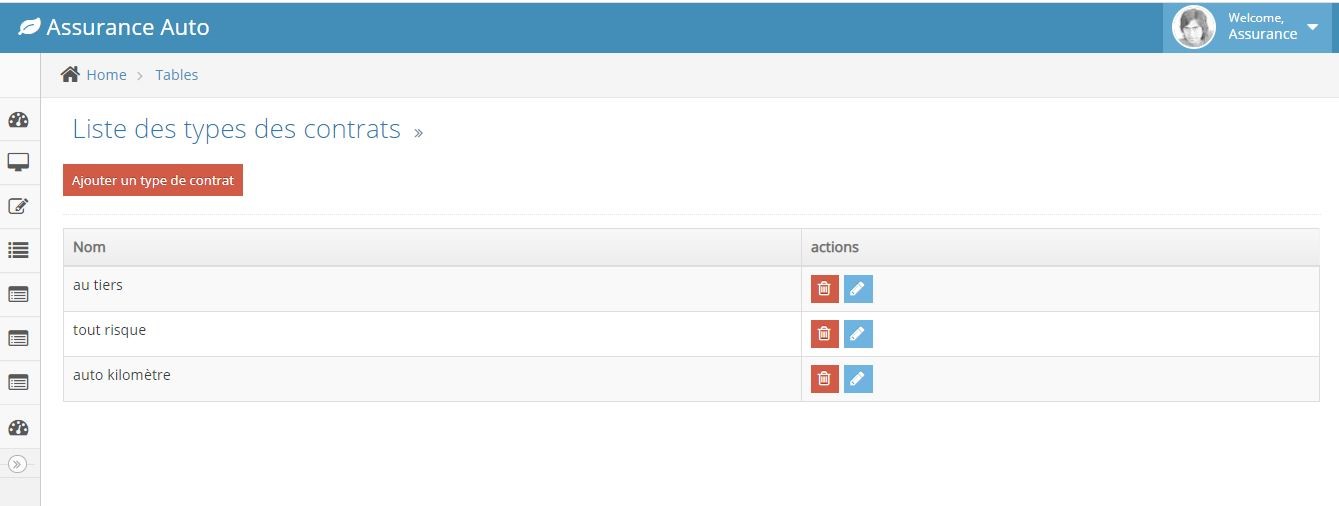


FIGURE 6.9 – Interface Liste des types de contrats

Cette interface contient la liste de tous les types de contrat saisies, nous pouvons les gérer soit pour les modifier ou les consulter.

L’administrateur peut aussi :

\*Ajouter un type de contrat : en cliquant sur le bouton « Ajouter », le « Formulaire type de contrat » sera affiché comme illustré dans la figure 6-10.

\* Modifier un type de contrat : s’il clique sur le bouton « Modifier » les informations relatives au type de contrat choisi seront affichées dans l’interface formulaire "type contrat", l’admin doit les modifier et cliquer sur le bouton « Enregistrer » pour les enregistrer.

2) Interface Ajout type contrat :

Lorsque l’administrateur clique sur le bouton « Ajouter » l’interface présentée de la figure

ci-dessous

sera

affichée.

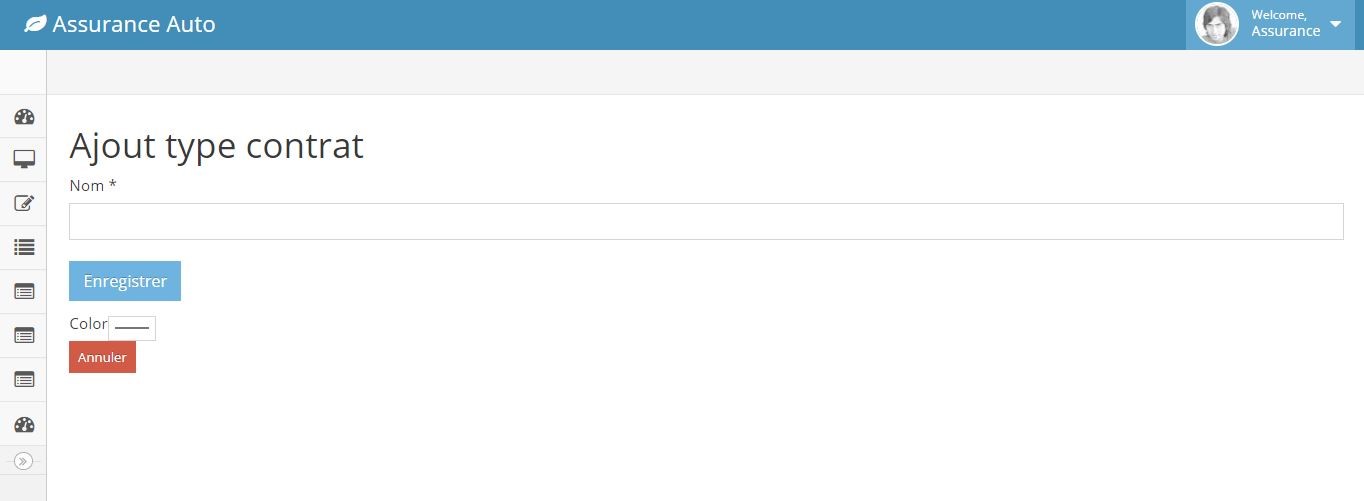


FIGURE 6.10 – Interface Ajout type contrat

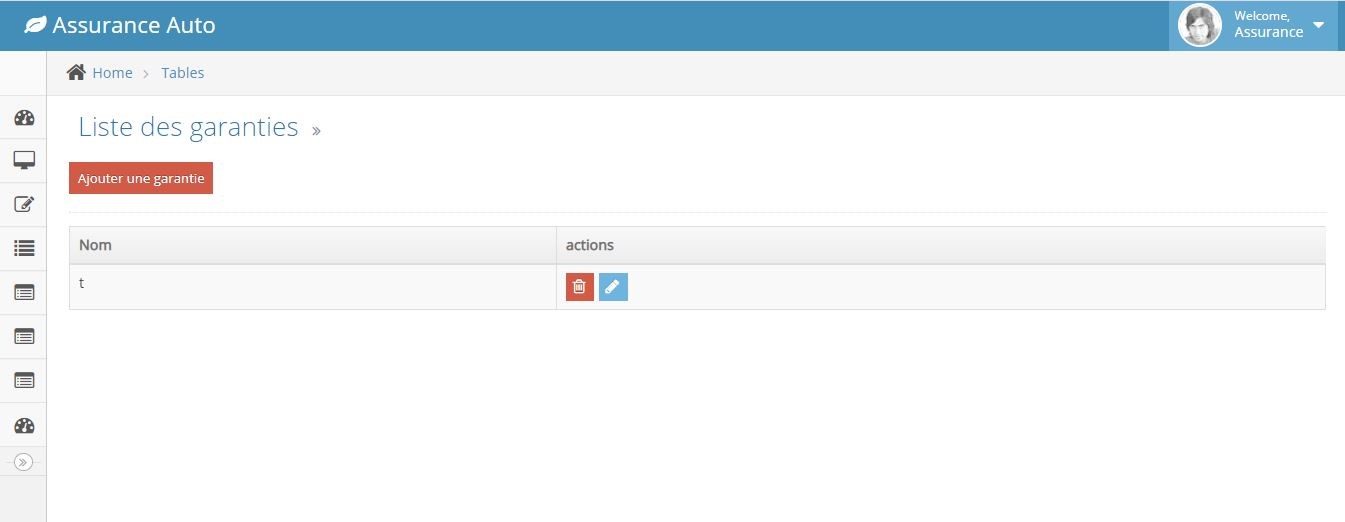
Cette interface donne la main au admin pour ajouter un nouveau type de contrat qui est défini par son nom.

### 6.4.3 Interface Garantie

1) Interface Liste des garanties :

Cette interface de la figure 6.11 s’ouvre lorsque l’admin clique sur le menu « Garantie ».

FIGURE 6.11 – Interface Liste des garanties



Cette interface contient la liste de toutes les garanties, nous pouvons les modifier , les supprimer et les consulter.

L’administrateur peut :

1) Ajouter une garantie : en cliquant sur le bouton « Ajouter », le « Formulaire garantie» sera affiché comme illustré dans la figure ci-dessous.

2 ) Modifier une garantie : s’il clique sur le bouton « Modifier » les informations relatives à la garantie sélectionnée seront affichées dans l’interface formulaire garantie, l’admin doit alors les modifier en saisisant les nouvelles informations liées à la nouvelle garantie .

\*Supprimer une garantie : en cliquant sur le bouton « Supprimer » après avoir sélectionné la garantie choisie.

2) Interface "Ajout garantie" :

Lorsque l’administrateur clique sur le bouton « Ajouter » l’interface présentée de la figure

6.12

sera

affichée

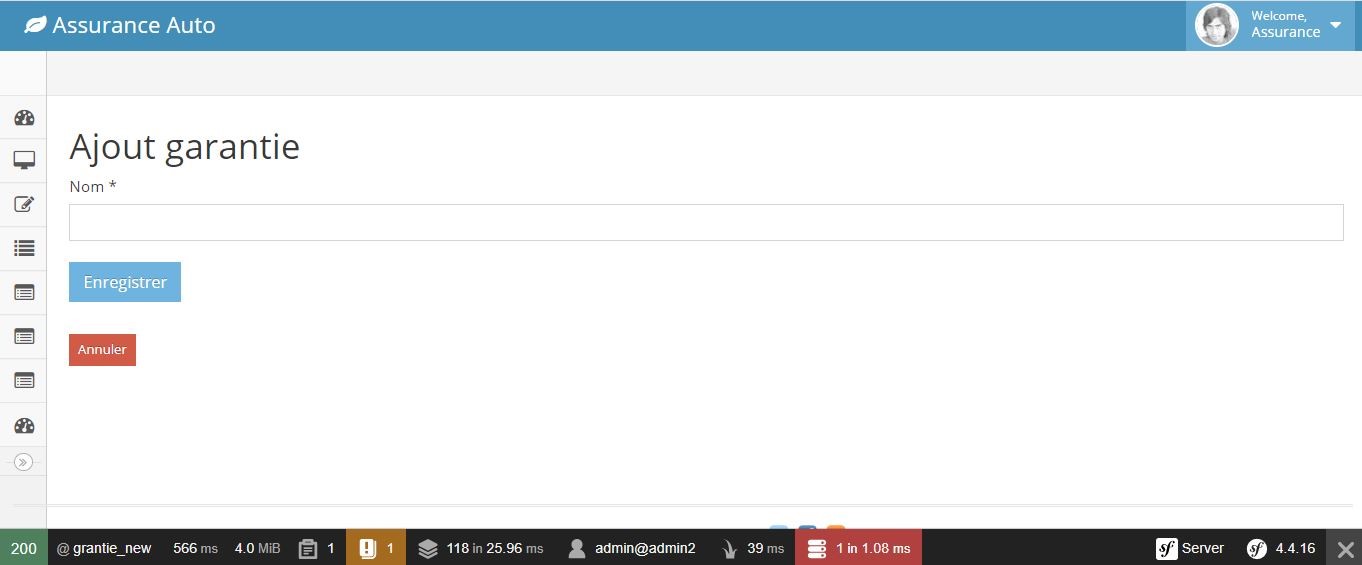


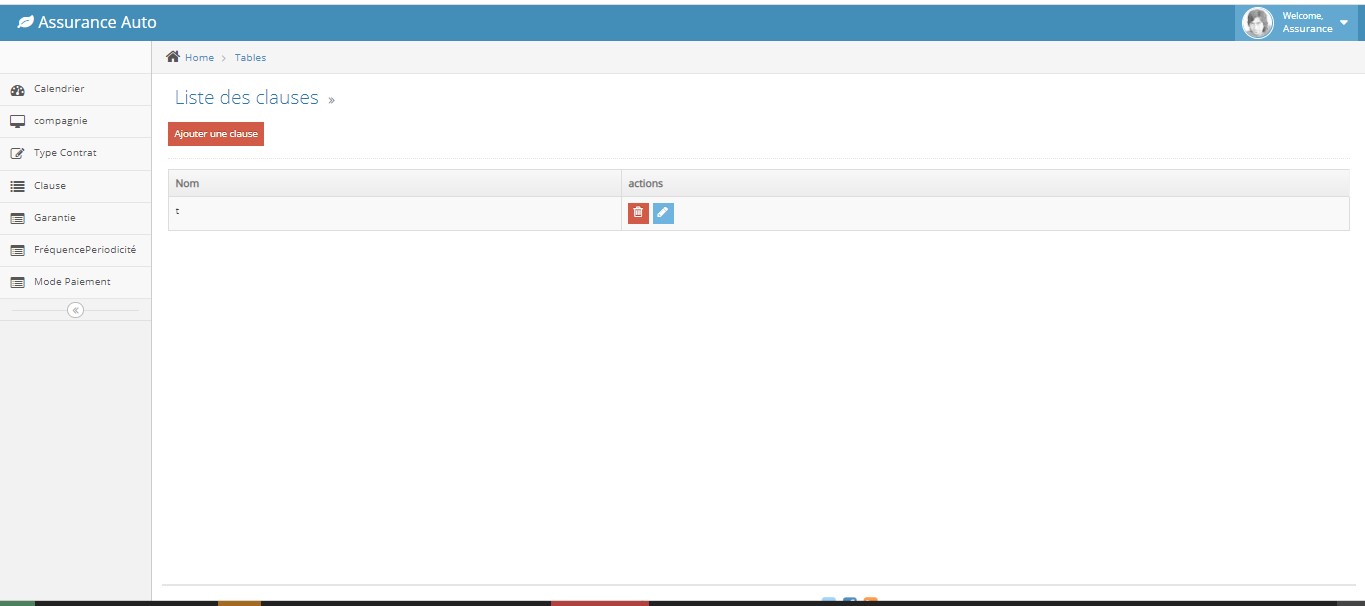
FIGURE 6.12 – Interface Ajout garantie

### 6.4.4 Interface Clause

1) Interface liste des clauses :

L’interface de la figure 6-13 s’ouvre lorsque l’administrateur clique sur le menu «Clause ».

FIGURE 6.13 – Interface Liste des clauses



Cette interface contient la liste de tous les clauses saisies, nous pouvons les gérer pour les modifier et les consulter.

L’administrateur peut aussi :

\*Ajouter une clause : en cliquant sur le bouton « Ajouter », le « Formulaire Clause » sera affiché comme illustré dans la figure 6-14.

\* Modifier une clause : s’il clique sur le bouton « Modifier » les informations relatives au clause choisi seront affichées dans l’interface formulaire clause , l’admin doit les modifier et cliquer sur le bouton « Valider » pour les enregistrer.

2) Interface Ajout clause :

Lorsque l’administrateur clique sur le bouton « Ajouter » l’interface présentée de la figure

6-14

sera

affichée.

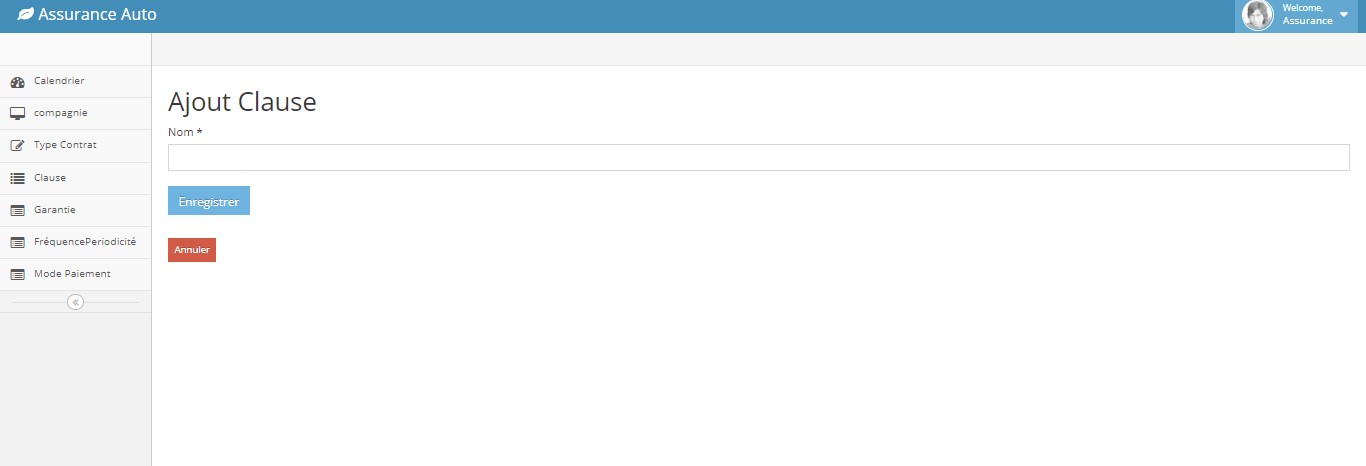


FIGURE 6.14 – Interface Ajout clause

Cette interface donne la main pour ajouter une nouvelle qui est défini par son nom.

### 6.4.5 Interface fréquence de priodicité

1) Interface liste des fréquences :

L’interface de la figure 6-15 s’ouvre lorsque l’administrateur clique sur le menu «Fréquence

».



FIGURE 6.15 – Interface Liste des fréquence

Cette interface contient la liste de tous les types de contrat saisies, nous pouvons les gérer pour les modifier et les consulter.

L’administrateur peut aussi :

\*Ajouter une fréquence : en cliquant sur le bouton « Ajouter », le « Formulaire Fréquence » sera affiché comme illustré dans la figure 6-16.

\* Modifier une fréquence : s’il clique sur le bouton « Modifier » les informations relatives au fréquence choisi seront affichées dans l’interface formulaire fréquence , l’admin doit les modifier et cliquer sur le bouton « Enregistrer » pour les enregistrer.

2) Interface Ajout fréquence :

Lorsque l’administrateur clique sur le bouton « Ajouter » l’interface présentée de la figure

6-14

sera

affichée.



FIGURE 6.16 – Interface Ajout fréquence

Cette interface donne la main pour ajouter une nouvelle fréquence qui est défini par son nom.

### 6.4.6 Interface mode de paiement

1) Interface liste des modes de paiement :

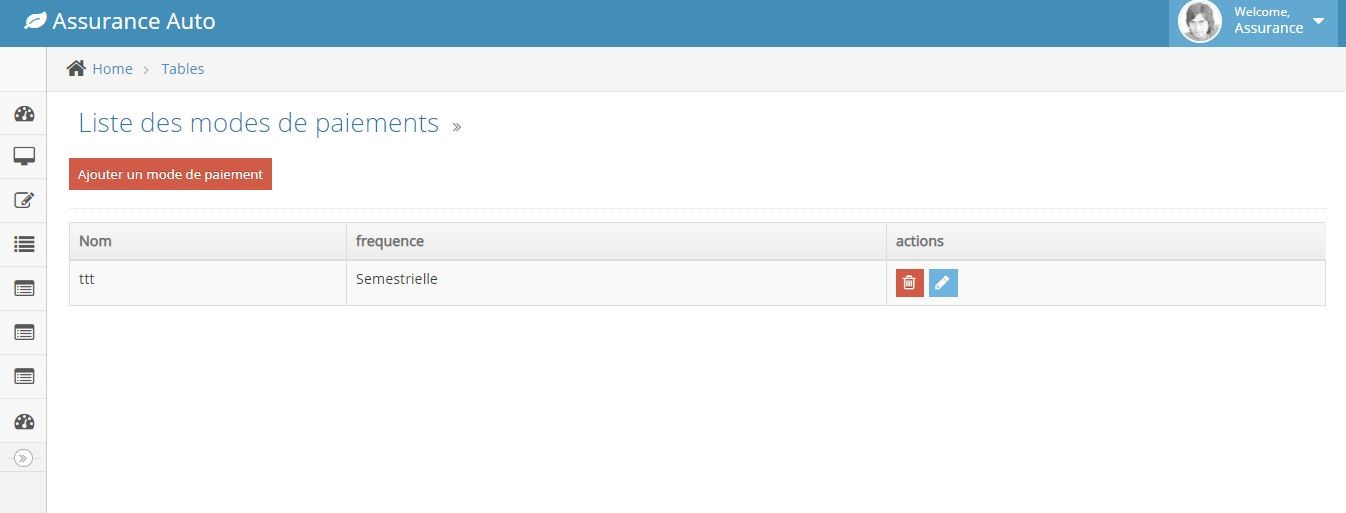
L’interface de la figure 6-17 s’ouvre lorsque l’administrateur clique sur le menu «mode de paiement ».

Cette interface contient la liste de tous les modes de paiement saisis, nous pouvons les gérer pour les modifier et les consulter.

L’administrateur peut aussi :

\*Ajouter un mode de paiement : en cliquant sur le bouton « Ajouter », le « Formulaire Mode de paiement » sera affiché comme illustré dans la figure 6-18.

FIGURE 6.17 – Interface Liste des modes de paiement



\* Modifier un mode de paiement : s’il clique sur le bouton « Modifier » les informations relatives au fréquence choisi seront affichées dans l’interface formulaire mode de paiement , l’admin doit les modifier et cliquer sur le bouton « Valider » pour les enregistrer.

2) Interface Ajout mode de paiement :

Lorsque l’administrateur clique sur le bouton « Ajouter » l’interface présentée de la figure

6-18

sera

affichée.

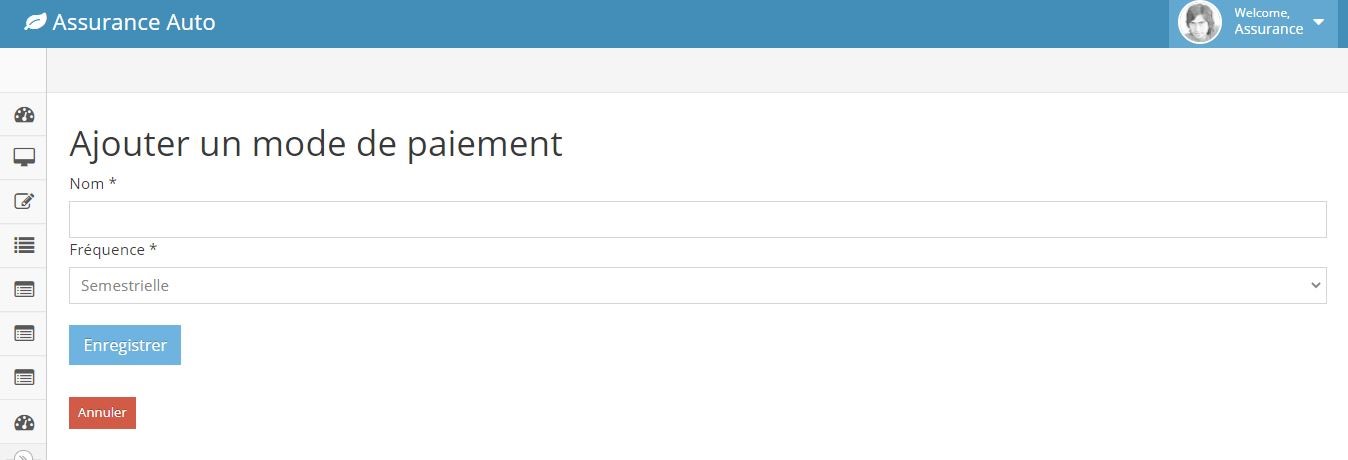


FIGURE 6.18 – Interface Ajout mode de paiement

Cette interface donne la main pour ajouter une nouveeau mode de paiement qui est défini par son nom et sa fréquence.

## Conclusion

Au cours de ce chapitre, nous avons réussi à produire le troisième livrable du projet qui se focalise sur le role de l’admin .

**Chapitre 7**

**Sprint quatre : Rôle Souscripteur et Agent d’assurance**

74

## Introduction

Nous pouvons maintenant passer au quatrième Sprint. Nous allons définir les tâches à réaliser .

l’objectif est de nous concentrer sur le développement de la partie consacrée à l’agent d’assurance et au souscripteur

## 7.1 Sprint Backlog

### 7.1.1 Histoires à réaliser

La liste des tâches à réaliser dans le sprint trois est décrite dans le tableau ??.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id story | User story | id tâche | tâche | estimation(H) |
| 10 | Demandes de souscription | 10.1 | En tant que souscripteur que je veux créer une demande de souscription | 8 |
| 10.2 | En tant qu’agent que je veux accepter la demande | 8 |
| 10.3 | En tant qu’agent que je veux refuser la demande | 8 |
| 10.4 | En tant que souscripteur je veux consulter mes demandes de souscription | 8 |
| 10.5 | En tant qu’agent d’assurance que je veux consulter toutes les demandes de souscription | 8 |
| 11 | Réclamations | 11.1 | En tant que souscripteur je veux créer une réclamation | 8 |
| 12 | Contrats individuels | 12.1 | En tant qu’agent d’asurance que je veux créer un contrat | 8 |
| 12.2 | En tant que souscripteur je veux consutler mes contrats | 8 |
| 12.3 | En tant qu’agent que je  consulter tous les contrats | 8 |
| 13 | Constats | 13.1 | En tant que souscripteur que je veux créer un constat pour chaque voiture | 8 |
| 13.2 | En tant qu’agent que je veux consulter tous les constats | 8 |
| 13.3 | En tant que souscripteur que je veux consuter mes constats | 8 |

TABLE 7.1: Liste des tâches du quatrième Sprint

### 7.1.2 Objectif du sprint

L’objectif de ce quatrième sprint est la réalisation du module consacré au role du souscripteur et de l’agent d’assurance .

## 7.2 Analyse

Nous choisissons de présenter tout d’abord le diagramme de cas d’utilisation détaillés afin de limiter la complexité de ce modèle.

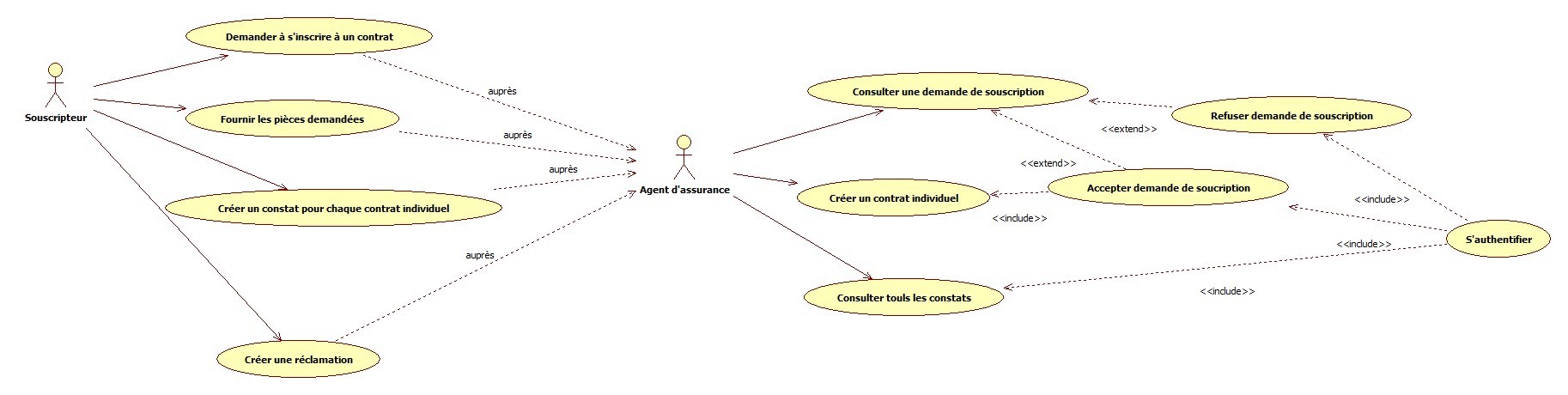


FIGURE 7.1 – Diagramme de cas d’utlisateur "Souscripteur et agent d’assurance "

#### 7.2.0.1 Description textuelle de "Création contrat "

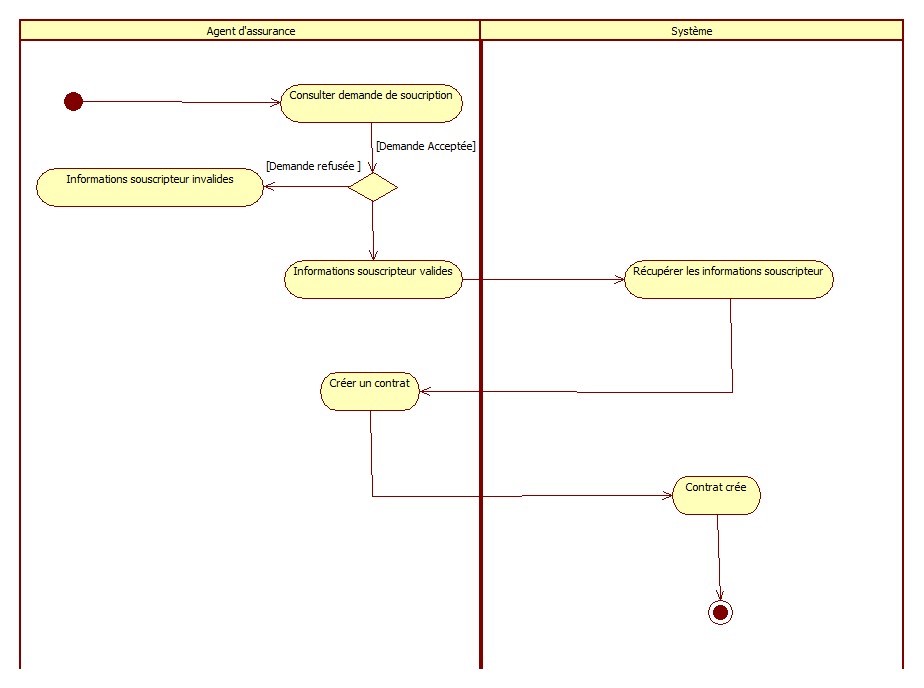
|  |  |
| --- | --- |
| Titre | Création contrat |
| Acteur principal | Agent d’assurance |
| Pré-condition | L’agent s’est authentifié |
| Post-condition | Le contrat est crée |
| Scénario nominal | L’UC débute quand l’agent d’assurances accède à l’acte « Créer un contrat » à partir du menu principal et le processus se poursuit selon les étapes suivantes :  a) Demande de soucription :   1. Le système affiche tout d’abord le formulaire de création de demande de souscription . 2. Au premier lieu, le souscripteur saisi toutes ses informations personelles et toutes les pièces demandés 3. Dès que , l’agent d’assurance consulte la demande de soucription crée par le souscripteur il doit la valider a) Création contrat individuel : 4. En cas d’acceptation de la demande de soucription , l’agent d’assurance crée un contrat en récupérant toutes les informations et les pièces saisies par le souscripteur. 5. L’agent d’assurances clique sur le bouton « enregistrer ». 6. L’agent d’assurances télécharge le contrat en format PDF. 7. Le souscripteur consulte ses contrats. |
| Scénario alternatif | Modifier la demande de souscription a) Demande de modification :  1) Le système affiche la liste des demandes de chaque souscripteur 2) Le souscripteur demande de modifer les informations d’une demande en cliquant sur le bouton "modifier" correspondant à cette demande  3) Le système affiche le formulaire de modification et donne la main au souscripteur pour modifier les informations relatives à cette demande 4) le souscripteur saisit les nouvelles informations puis il clique sur le bouton "valider"  b) validation de la modification  Le système affiche la modification effectuée |
| Scénario d’échec | En cas de non validation des documents l’agent d’assurance peut refuser la demande de souscription crée par le souscripteur et le contrat ne sera pas crée  Le système affiche le message « Format incorrecte » en cas où une ou plusieurs valeurs saisies sont incorrectes |

#### 7.2.0.2 Description textuelle de "Création constat "

|  |  |
| --- | --- |
| Titre | Création constat |
| Acteur principal | Souscripteur |
| Pré-condition | Le souscripteur s’est authentifié |
| Post-condition | Le constat est crée |
| Scénario nominal | L’UC débute quand le souscripteur accède à l’acte « Espace contrat » à partir du menu principal et le processus se poursuit selon les étapes suivantes :  a) Demande de consultation :  1 le souscripteur demande à consulter ses demandes de souscription pour vérifier leurs états a) Création constat :  1 En cas d’acceptation de la demande de soucription , la demande de souscription sera tranformée à un contrat . 2 Le souscripteur consulte les détails d’un contrat .   1. Le souscripteur clique sur le bouton " Ajouter un constat pour chaque voiture " 2. Le souscripteur saisie les infomrations d’un constat . 3. Le souscripteur clique sur le bouton enregistrer pour créer un constat . |
| Scénario alternatif | Modifier le constat  a) Demande de modification :  1) Le système affiche la liste de ses constat pour chaque voiture 2) Le souscripteur demande de modifer les informations d’un constat en cliquant sur le bouton "modifier" correspondant à ce constat 3) Le système affiche le formulaire de modification et donne la main au souscripteur pour modifier les informations relatives à ce constat 4) le souscripteur saisit les nouvelles informations puis il clique sur le bouton "enregistrer"  b) validation de la modification  Le système affiche la modification effectuée |
| Scénario d’échec | Le souscripteur consulte ses demandes validés par l’agent d’assurance pour vérifier l’état de chaque demande  Toute demande ayant comme état : En cours ou refusé : ne sera pas tranformé à un contrat |

#### 7.2.0.3 Diagramme d’activité "Création contrat"

FIGURE 7.2 – Diagramme d’activité création contrat



#### 7.2.0.4 Diagramme d’activité "Création constat"

la figure 7-3 représente le diagramme d’activité "création constat" comportant les étapes de la création d’un constat d’assurance ainsi que la séquence d’actions et de décisions au

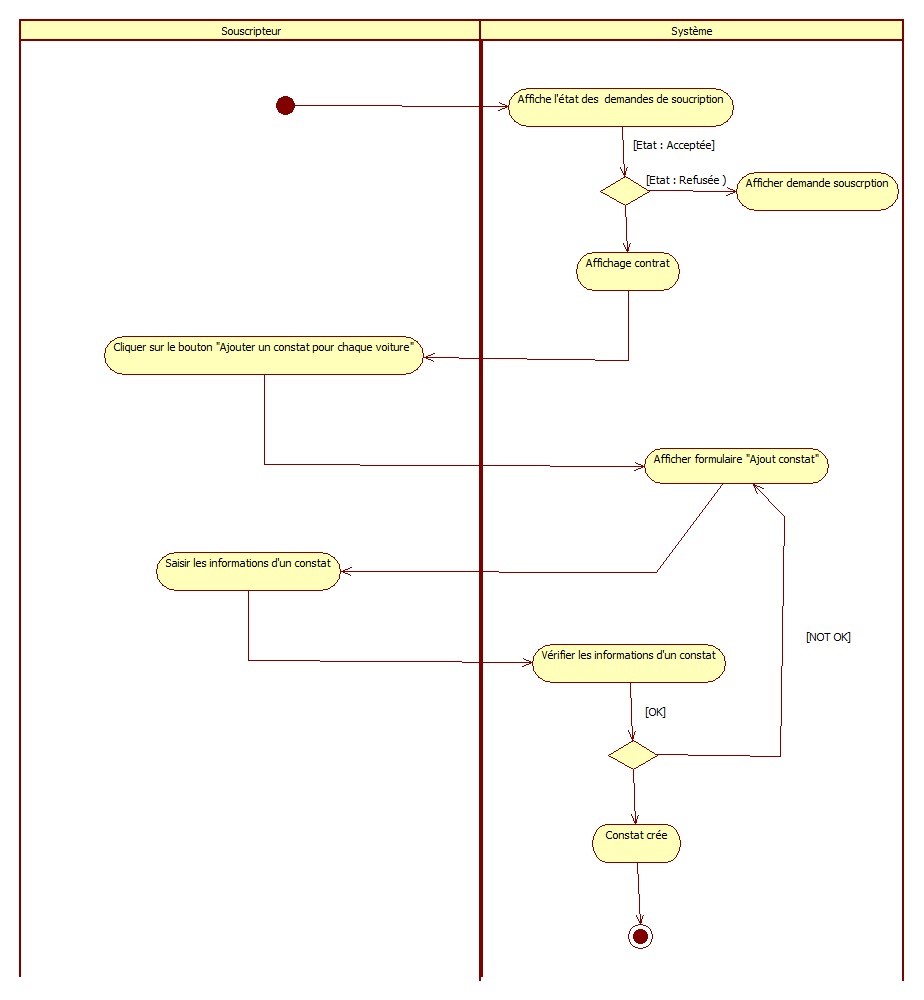
FIGURE 7.3 – Diagramme d’activité création constat

sein

de

cette

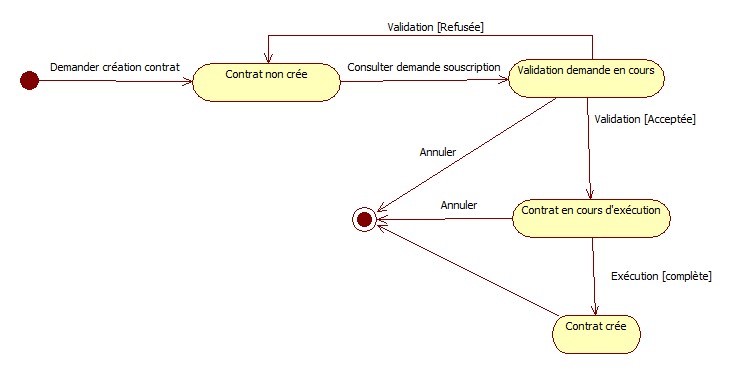
activité



#### 7.2.0.5 Diagramme état de transition "Création contrat "

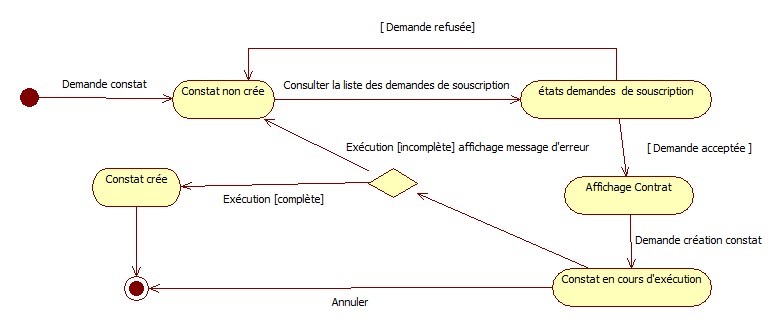
Comme illustré dans la figure7-4, le diagramme d’état-transition « Création contrat » représente le cycle de vie de la classe contrat , ainsi ses états possibles et les transitions nécessaires.

FIGURE 7.4 – Diagramme état de tansition création contrat



#### 7.2.0.6 Diagramme état de transition "Création constat "

FIGURE 7.5 – Diagramme état de tansition création constat

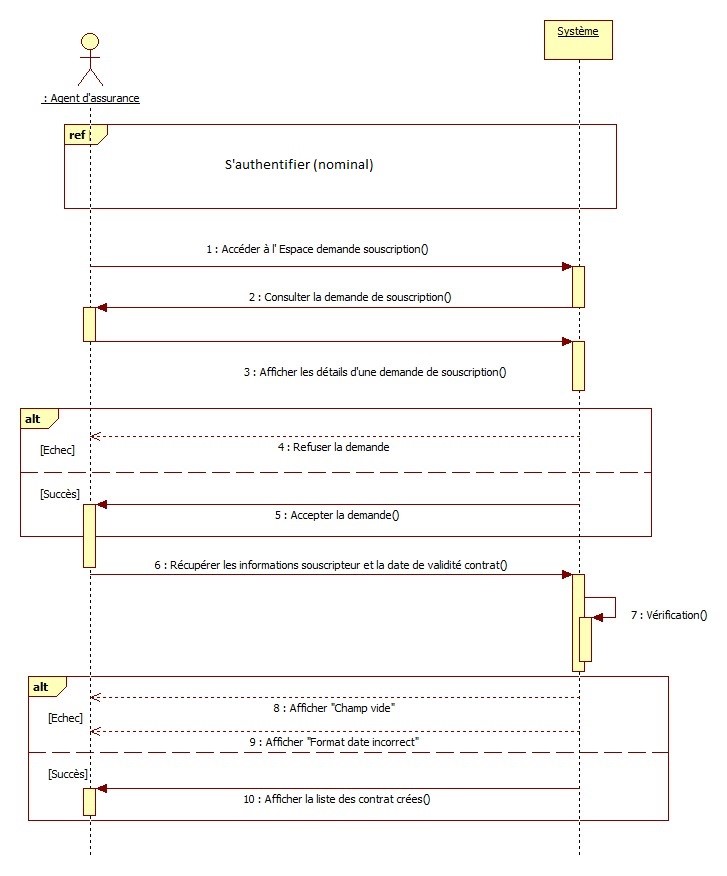


#### 7.2.0.7 Diagramme de séquence système "Création contrat"

Le processus de la création d’un contrat pour un souscripteur se réalise quand l’agent d’assurance passe par des étapes ou des écrans commençant par la « validatin de la demande de souscription » jusqu’à la «création d’un contrat" en récupérant les données du souscripteur, la garantie , la compagnie d’assurance , la clause , la fréquence de périodicité et le mode de paiement choisis et saisisant la date de validation de contrat.

Pendant l’étape "Demande de soucription" l’agent d’assurance doit valider la demande En cas de non validation ou en manque de documents , ce dernier refuse la demande , En cas d’acceptation , l’agent crée un contrat pour le souscripteur .

FIGURE 7.6 – Diagramme séquence système création contrat



#### 7.2.0.8 Diagramme de séquence système "Création constat"

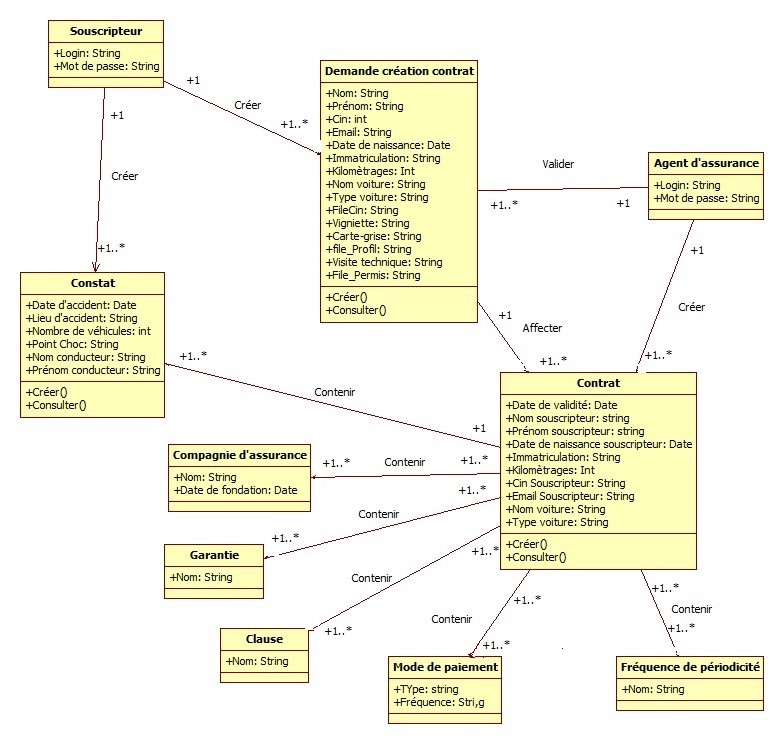
La figure 7.7 représente le processus de création constat qui début par la création de contrat . Une fois que la demande de soucription a été acceptée par l’agent d’assurance , le contrat se crée en récupérant toutes les données de soucripteur qui est à son tour ajoute un constat pour ce contrat .

FIGURE 7.7 – Diagramme séquence système création constat



## 7.3 Conception

#### 7.3.0.1 Diagramme de classe "souscripteur et agent d’assurance "



F

IGURE

7.8

–

Diagramme

de

classe

"souscripteur

et

agent

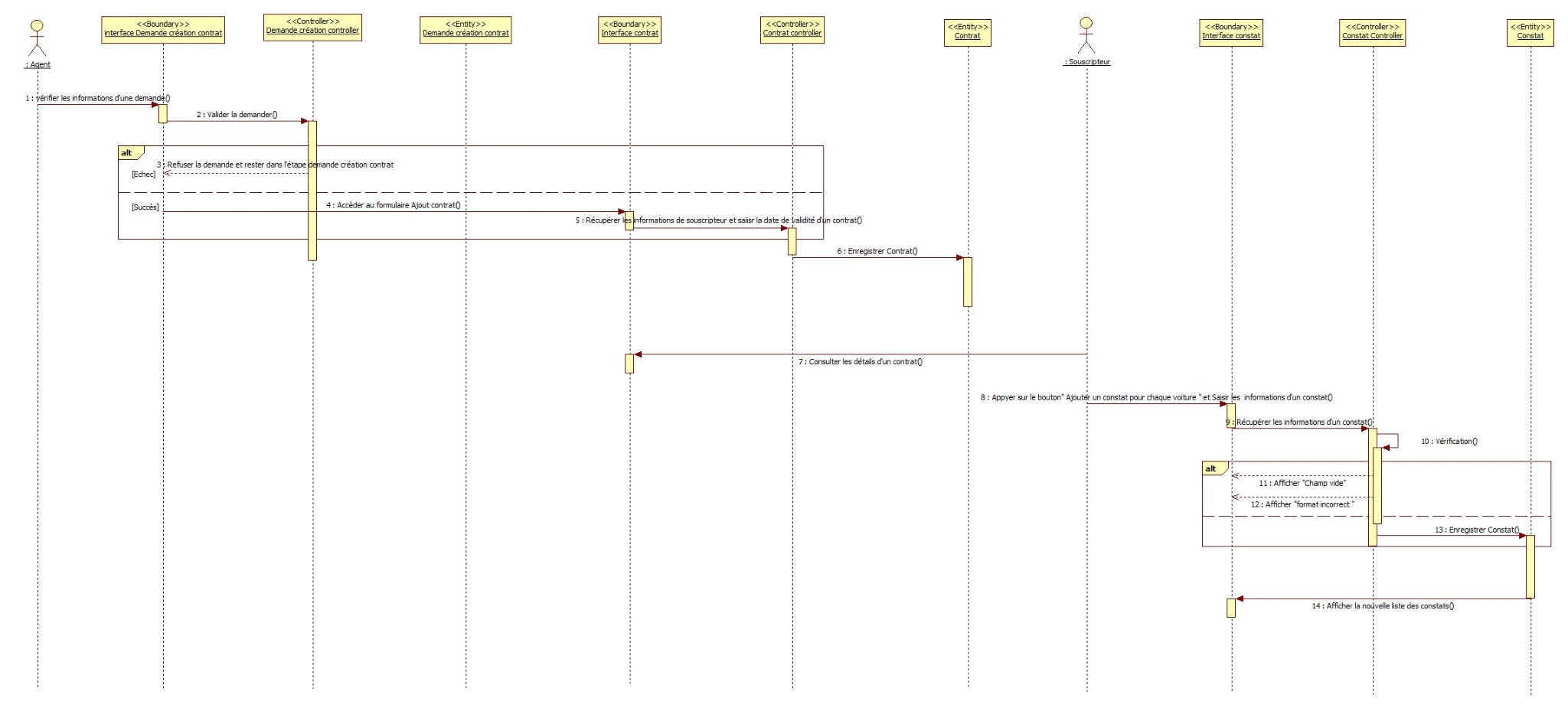
d’assurance

"

#### 7.3.0.2 Diagramme de séquence objet "Agent et souscripteur"

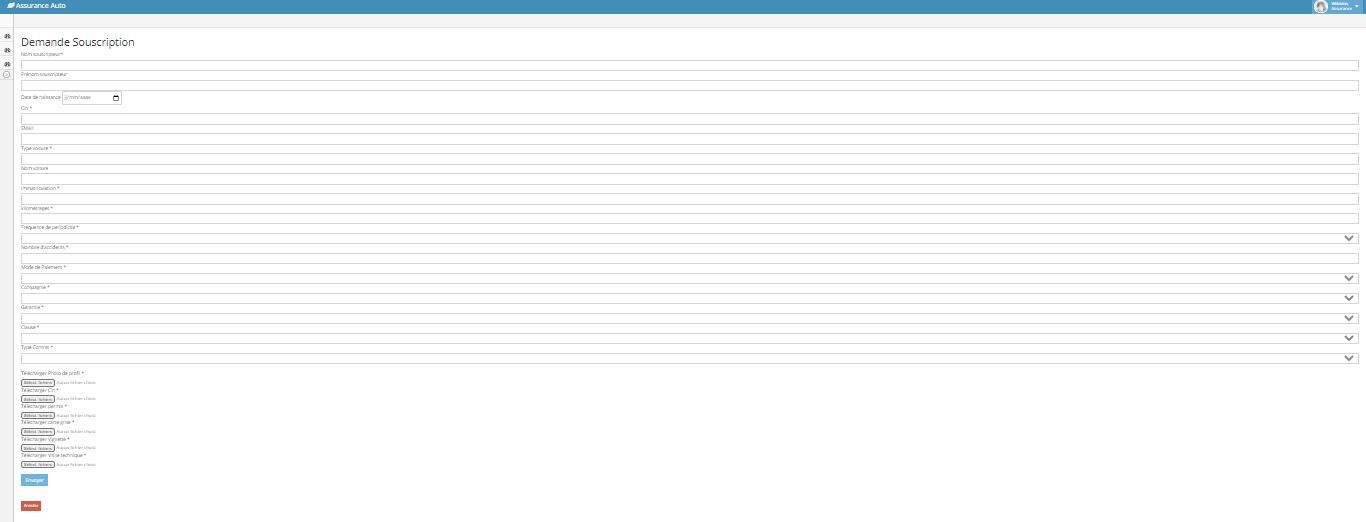
La figure présente les étapes que fassent l’agent d’assurance et le souscripteur pour créer un constat pour chaque voiture . En effet, la figure détaille les étapes et les interactions entre les interfaces de l’application pour finir cet acte .

FIGURE 7.9 – Diagramme de séquence objet "Création Constat"



## 7.4 Implémentation

Dans cette partie, on va présenter l’interface Homme-Machine implémenté au niveau du dernier Sprint. la figure 7.10 présente le formulaire de demande de soucription



F

IGURE

7.10

–

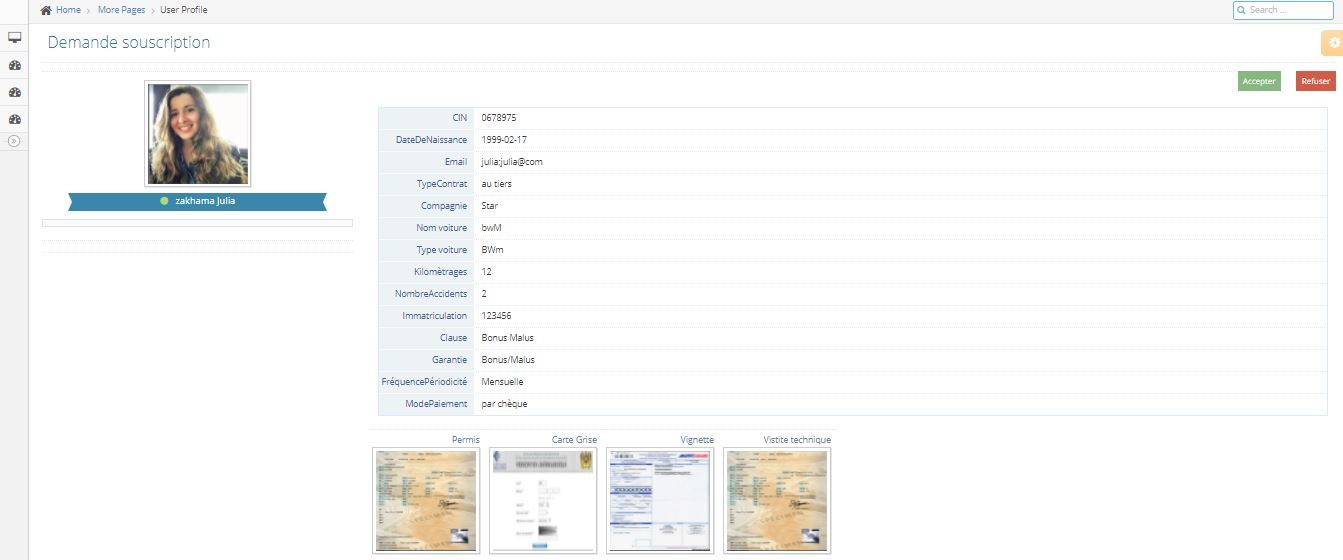
Interface

formulaire

"Demande

souscription"

la figure 7.11 affiche les détails de chaque demande .



F

IGURE

7.11

–

affichage

Interface

"Demande

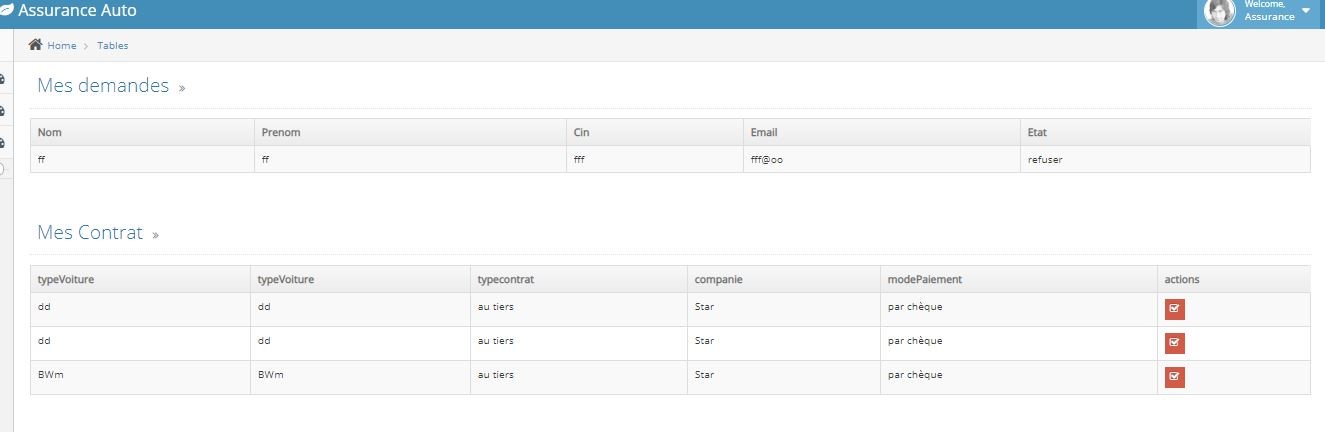
souscription"

La figure 7.12 présente le formulaire de création de contrat qui se crée quand la demande est acceptée par l’agent d’assurance .



FIGURE 7.12 – Interface formulaire "Création contrat"

La figure 7.13 affiche l’état des demandes de souscriptions d’un souscripteur.



F

IGURE

7.13

–

affichage

de

l’état

des

demande

de

souscriptions

d’un

souscripteur

La figure 7.14 présnte la nouvelle liste des contrats .



FIGURE 7.14 – affichage de la liste des contrats

La figure 7.14 présente le formulaire de création de constat .

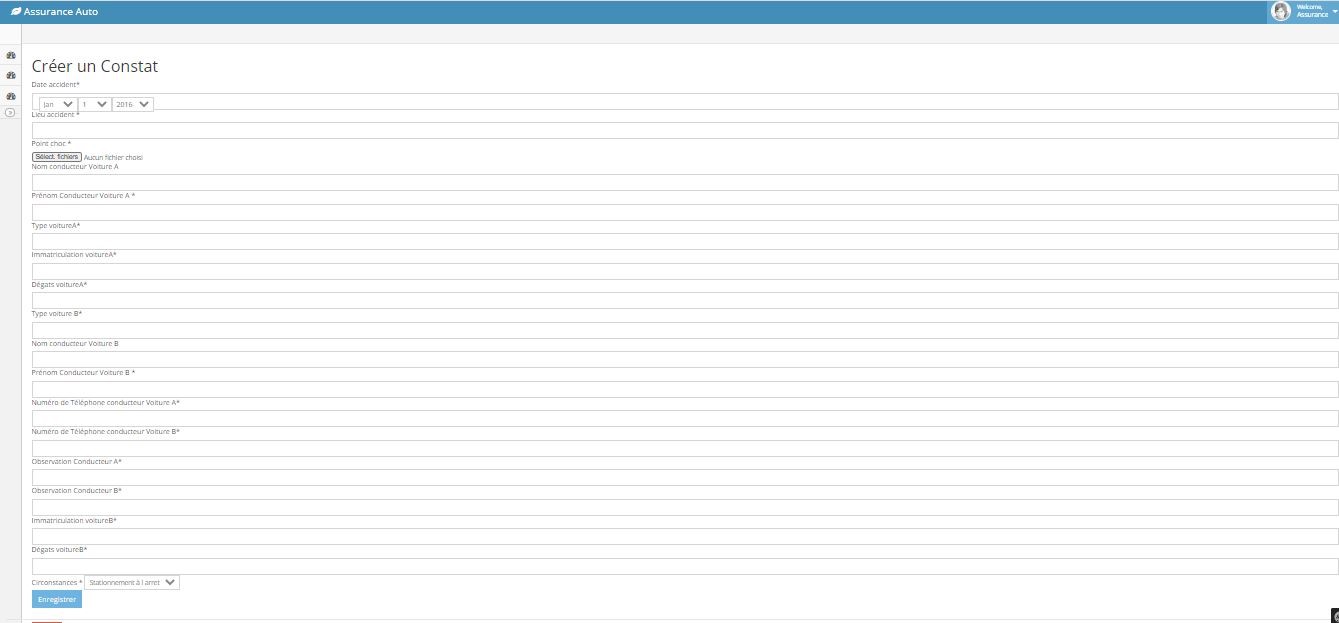


FIGURE 7.15 – Formulaire de création constat

## Conclusion

Au cours de ce chapitre, nous avons réussi à produire le dernier livrable du projet qui comporte la gestion des demandes et des réclamation. Comme ça on peut dire que nous avons atteint tout les objectifs de cahier de charge de projet.

# Conclusion Générale et Perspectives

Le travail présenté dans ce rapport de stage consiste à la conception et le développement d’une application sécurisée qui découpe une souscription en plusieurs étapes de l’information jusqu’à la rédaction d’un constat.

Tout d’abord, nous avons fait une pré-étude. Ensuite, nous avons détaillé la modélisation des besoins et modélisation conceptuelle, en présentant les acteurs du système et en développant les diagrammes de cas d’utilisation, le diagramme de classes, le diagrammes de séquence et diagrammes d’état. Enfin, nous avons présenté les interfaces de notre application.

Cette application est développé sous l’environnement "PhpStorm", communiquant avec une base de données de données "MySQL".

Ce stage que j’ai du effectué au sein de la « DSI » m’a permis de me familiariser avec les différents services web et d’avoir une approche réelle du monde du travail. J’ai pu faire le rapprochement entre ce que j’avais appris en cours et ce qui se passe vraiment dans l’ entreprise .

Concernant les perspectives de ce projet, nous prévoyons d’améliorer la fonctionnalités de l’application pour l’adapter davantage aux besoins des futurs clients.

Des nouvemmes fonctionnalités peuvent être intégrés à la solution pour la création d’un contrat et d’un constat.

tels que : L’ajout des actes de gestion comme la demande d’une avance ,la gestion des primes et la couverture des dommages matériels .

90