J'autorise l'étudiant à faire le dépôt de son rapport de stage en vue d'une soutenance.

Encadrante de l'entreprise, **Imen BOUDABOUS**

Signature

J'autorise l'étudiant à faire le dépôt de son rapport de stage en vue d'une soutenance.

Encadrante académique, **Imene MBAZAA**

Signature

Dédicaces

A mes très chers parents, qui m'ont permis de devenir ce que je suis aujourd'hui,

A mes soeurs, qui ont été toujours à mes cotés

A mes collègues, qui ont rendu la période de stage agréable,

A mes amis,

Je dis merci et je leur dédie ce travail qui est l'aboutissement de plusieurs années de travail.

Ahmed IFA

Remerciements

En tout premier lieu, je remercie le bon Dieu, tout puissant, de m'avoir donné la force pour survivre, ainsi que l'audace pour dépasser toutes les difficultés.

Je tiens à adresser mes plus sincères remerciements à Madame MBAZAA Imene, mon encadrante à l'école supérieur privée d'ingénierie et de technologies, pour le temps qu'elle m'avez consacré, pour son aide précieuse et l'encadrement sérieux dont elle a fait preuve ainsi qu'à nos enseignants de l'ESPRIT à qui je dois ma formation.
Je tiens à exprimer ma profonde gratitude à Madame BOUDABOUS Imen, mon tuteur, pour m'avoir intégré rapidement au sein de l'entreprise et m'avoir accordé toute sa confiance ; pour le temps qu'il m' a consacré tout au long de cette période et de répondre à toutes mes interrogations sans oublier sa participation aux réalisations de ce rapport.

Je tiens à remercier tout particulièrement et à témoigner toute ma reconnaissance aux personnes suivantes, pour l'expérience enrichissante et pleine d'intérêt qu'elles m' ont fait vivre durant ces quatre mois au sein de Talan Tunisie Consulting.

Enfin, j' adresse mes remerciements aux membres du jury pour m'avoir honoré en acceptant d'évaluer ce travail.

Sans oublier tous ceux qui ont participé de prés ou de loin à l'accomplissement de ce travail.

Table des matières

Introduction générale **1**

1 Contexte général **3**

1.1 Organisme d'accueil 4

1.1.1 Talan Group 4

1.1.2 Secteurs d'activité 5

1.1.3 Talan Testing Factory 6

1.2 L'activité de Testing 7

1.2.1 Différents types de Tests 7

1.2.2 Recette informatique [2] 8

1.3 Étude et critique de l'existant 8

1.4 Solution proposée 11

1.5 Méthodologie adoptée 11

1.5.1 Méthode SCRUM 12

2 Sprint 0 : Analyse et spécification des besoins **15**

2.1 Analyse des besoins 16

2.1.1 Identification des acteurs 16

2.1.2 Besoins fonctionnels 16

2.1.3 Besoins non fonctionnels 17

2.1.4 Diagramme de cas d'utilisation 18

2.2 Backlog Produit 18

2.3 Diagramme de classe d'analyse 20

2.4 Choix des technologies et l'environnement de travail 21

2.4.1 Technologies 21

2.4.2 Environnement de travail 21

2.5 Architecture logique et physique globale 22
2.5.1 Architecture logique 22
2.5.2 Architecture physique 24
2.6 Planification des sprints 24

3 Sprint 1 : Gestion des utilisateurs 26

3.1 Backlog du sprint 27
3.2 Analyse 27
3.2.1 Diagramme de cas d'utilisation raffiné 28
3.2.2 Analyse du cas d'utilisation "S'authentifier" 29
3.2.3 Analyse du cas d'utilisation "Ajouter nouvel utilisateur" 31
3.3 Conception 32
3.3.1 Diagramme de composants 32
3.3.2 Diagramme de séquences objet 33
3.4 Réalisation 35
3.4.1 Interface d'authentification 35
3.4.2 Interface de demande d'accès 35
3.4.3 Interface de profile 37
3.4.4 Les interfaces de l'administrateur 37

4 Sprint 2 : Gestion de catalogue de template 41

4.1 Backlog du Sprint 42
4.2 Analyse 42
4.2.1 Diagramme de cas d'utilisation raffiné 43
4.2.2 Analyse du cas d'utilisation "Ajouter template" 43
4.2.3 Analyse du cas d'utilisation "Modifier template" 45
4.3 Conception 47
4.3.1 Diagramme de composants 47
4.3.2 Diagrammes de séquences objet 48

4.3.3 Diagramme de classes participantes 49
4.4 Réalisation 50

5 Sprint 3 : Gestion de scénario 53

5.1 Backlog Sprint 3 54
5.2 Analyse 54
5.2.1 Diagramme de cas d'utilisation raffiné 55
5.2.2 Analyse du cas d'utilisation "Ajouter Scénario" 55
5.2.3 Analyse du cas d'utilisation "Modifier Scénario" 57
5.3 Conception 59
5.3.1 Diagramme de composants 59
5.3.2 Diagrammes de séquences objet 60
5.3.3 Diagramme de classes participantes 61
5.4 Réalisation 61

6 Sprint 4 : Gestion et Exportation de cahier de recette 64

6.1 Backlog Sprint 4 65
6.2 Analyse 65
6.2.1 Diagramme de cas d'utilisation raffiné 66
6.2.2 Analyse du cas d'utilisation "Créer un cahier de recette" 67
6.2.3 Analyse du cas d'utilisation "Exporter un cahier de recette" 68
6.3 Conception 70
6.3.1 Diagramme de composants 70
6.3.2 Diagrammes de séquence objet 71
6.3.3 Diagrammes de classes participantes 72
6.4 Réalisation 72

Conclusion générale 79

Références 80

Table des figures

1.1 Logo Talan Tunisie 5

1.2 Répartition des activités de Talan 6

1.3 Modèle du fichier Excel entrée pour HP ALM 9

1.4 Exemple d'un brouillon de cahier de test rédigé à l'aide de Microsoft Excel 10

1.5 Processus de SCRUM [3] 13

1.6 SCRUM Board avec l'outil Trello 14

2.1 Diagramme de cas d'utilisation global 18

2.2 Diagramme de classe d'analyse 20

2.3 Architecture logique du BackEnd 23

2.4 Architecture logique du FrontEnd 23

2.5 Diagramme de déploiement 24

3.1 Diagramme de cas d'utilisation du Sprint 1 28

3.2 Diagramme de cas d'utilisation raffiné 28

3.3 Diagramme de séquence système du cas d'utilisation "S'authentifier" 30

3.4 Diagramme de séquence système du cas d'utilisation "Ajouter nouvel utilisateur" . . 32

3.5 Diagramme de composants du Sprint 1 33

3.6 Diagramme de séquence objet du cas d'utilisation "S'authentifier" 34

3.7 Diagramme de séquence objet du cas d'utilisation "Ajouter nouvel utilisateur" . . . 34

3.8 Interface d'authentification 35

3.9 Interface de demande d'accès aux administrateurs 36

3.10 Pop-up de succès de la demande d'accès 36

3.11 Le corps de l'email de demande d'accès 37

3.12 Interface de profile de l'utilisateur connecté 37

3.13 Interface de la liste des utilisateurs 38

3.14 Interface du formulaire d'ajout d'un nouvel utilisateur 38
3.15 Pop-up d'échec d'opération d'ajout d'un utilisateur 39
3.16 Pop-up de succès d'opération d'ajout d'un utilisateur 39
3.17 Pop-up de confirmation de suppression d'un utilisateur 39

4.1 Diagramme de cas d'utilisation du Sprint 2 42
4.2 Diagramme de cas d'utilisation raffiné du Sprint 2 43
4.3 Diagramme de séquence système du cas d'utilisation "Ajouter Template" 45
4.4 Diagramme de séquence système du cas d'utilisation "Modifier Template" 47
4.5 Diagramme de composants de Sprint 2 48
4.6 Diagramme de séquence objet du cas d'utilisation "Ajouter template" 48
4.7 Diagramme de séquence objet du cas d'utilisation "Ajouter template" 49
4.8 Diagramme de classes participantes du Sprint 2 49
4.9 Interface d'un exemple d'ajout d'un template de cahier de recette 50
4.10 Interface du catalogue de templates en mode de modification 51
4.11 Pop-up de confirmation de la suppression d'un template 51
4.12 Message d'information de la suppression d'un template 52

5.1 Diagramme de cas d'utilisation du Sprint 3 54
5.2 Raffinement du diagramme de cas d'utilisation du Sprint 3 55
5.3 Diagramme de séquence système du cas d'utilisation "Ajouter Scénario" 57
5.4 Diagramme de séquence système du cas d'utilisation "Modifier Scénario" 59
5.5 Diagramme de composants du sprint 3 59
5.6 Diagramme de séquence objet du cas d'utilisation "Ajouter Scénario" 60
5.7 Diagramme de séquence objet du cas d'utilisation "Modifier Scénario" 60
5.8 Diagramme de classes participantes du Sprint 3 61
5.9 Interface du choix de scénario à définir 62
5.10 Interface d'ajout du contenu d'un scénario 62
5.11 Interface de modification d'un scénario 63

6.1 Diagramme de cas d'utilisation du Sprint 4 66
6.2 Diagramme de cas d'utilisation raffiné 66
6.3 Diagramme de séquence système du cas d'utilisation "Créer un cahier de recette" . . 68
6.4 Diagramme de séquence système du cas d'utilisation "Exporter un cahier de recette" 69
6.5 Diagramme de composants du Sprint 4 70
6.6 Diagramme de de séquence objet du cas d'utilisation "Créer un cahier de recette" . . 71
6.7 Diagramme de séquence objet du cas d'utilisation "Exporter un cahier de recette" . . 71
6.8 Diagramme de classes participantes du Sprint 4 72
6.9 Interface de création du cahier de recette 73
6.10 Interface de l'arborescence des scénarios d'un cahier généré 74
6.11 Alerte d'erreur lors de la création du cahier 74
6.12 Interface de la liste et de gestion des cahiers de recette 75
6.13 Interface de l'aperçu du cahier de recette 75
6.14 Interface de l'arborescence du cahier 76
6.15 Interface de succès de téléchargement d'un cahier 76
6.16 Capture d'un cahier exporté en fichier .xlsx 77
6.17 Message d'alerte pour confirmer la suppression du cahier 77
6.18 Popup informant le succès de la suppression du cahier 77

Liste des tableaux

2.1 Backlog produit 19

2.2 Tableau de planification des sprints 24

3.1 Backlog du Sprint 1 27

3.2 Description textuelle du cas d'utilisation "S'authentifier" 29

3.3 Description textuelle du cas d'utilisation "Ajouter nouvel utilisateur" 31

4.1 Backlog du Sprint 2 42

4.2 Description textuelle du cas d'utilisation "Ajouter template" 44

4.3 Description textuelle du cas d'utilisation "Modifier template" 46

5.1 Backlog du Sprint 3 54

5.2 Description textuelle du cas d'utilisation "Ajouter Scénario" 56

5.3 Description textuelle du cas d'utilisation "Modifier Scénario" 58

6.1 Backlog du Sprint 4 65

6.2 Description textuelle du cas d'utilisation "Créer un cahier de recette" 67

6.3 Description textuelle du cas d'utilisation "Exporter un cahier de recette" 69

Introduction générale

En génie informatique, une anomalie est une défaillance, une erreur ou un défaut d'un programme ou logiciel informatique qui produit un résultat ou un comportement inattendu.
Généralement, la plupart des anomalies proviennent d'erreurs commises par les développeurs lors de la conception, de l'écriture des codes, de la saisie de données, etc. Cela se produit souvent à cause de la complexité du code source (des millions de lignes de code dans plusieurs cas).
Un logiciel qui ne fonctionne pas correctement peut poser de nombreux problèmes : pertes financières, de temps ou de réputation, et peut même mettre en danger les données personnelles d'une personne et même de sa vie.

Le test est également un bon moyen d'apprendre à vérifier et à valider le produit avant de le présenter au client. De plus, cela permet de mieux connaître ce produit et de voir comment il fonctionne dans des situations différentes. C'est dans ce contexte que s'inscrit ce projet de fin d'études mené au sein de la société Talan Tunisie. Il s'agit de concevoir et de mettre en œuvre un outil générateur de cas de test pour un projet de base de données dont le but est de minimiser les efforts requis par le testeur au cours de sa mission de test.

Ce rapport résume les étapes de la mise en œuvre de cette application et s'articule autour des six chapitres suivants :
Le premier chapitre **"Contexte général"** placera le projet dans son contexte général en présentant l'organisation hôte et les notions générales dont il dépend, ainsi qu'une étude de l'existant, étape primordiale pour poser la problématique appropriée et pour fournir une solution pertinente. Ensuite, nous présenterons nos choix en termes de méthodologie de développement, et d'architecture adoptées.
Le deuxième chapitre **"Sprint 0"** sera consacré à la spécification et l'analyse des besoins qui est la première phase du cycle de SCRUM durant laquelle nous allons préparer un plan du travail.
Le troisième chapitre **"Sprint 1"** sera consacré à la gestion et l'authentification des utilisateurs de notre application.
Le quatrième chapitre **"Sprint 2"** sera dédié à la gestion du catalogue de "template" dans lequel nous allons faire l'étude analytique, conceptuelle et enfin la partie de la réalisation du livrable de ce sprint.

Le cinquième chapitre **"Sprint 3"** qui va reposer sur l'analyse de la fonctionnalité du gestion des scénarios de test en mettant en évidence sa conception ainsi que les interfaces de réalisation du livrable.

Le sixième chapitre **"Sprint 4"** qui porte sur la création du cahier de recette et son exportation en format .xlsx.

Chapitre 1

Contexte général

Plan

1 Organisme d'accueil 4

2 L'activité de Testing 7

3 Étude et critique de l'existant 8

4 Solution proposée 11

5 Méthodologie adoptée 11

Introduction

Le contexte général du projet est la première étape dans laquelle nous allons d'abord présenter l'organisme d'accueil. Ensuite, nous allons présenter l'activité de testing et faire une étude et critique de l'existant en posant la problématique et en présentant la solution proposé et la méthode de travail adoptée.

1.1 Organisme d'accueil

Cette section est la présentation de la société mère Talan Group et le centre de développement d'accueil Talan Tunisie. Nous commençons par une présentation générale de cet organisme , puis, nous détaillons ses domaines d'activité ainsi que quelques informations sur l'équipe d'accueil.

1.1.1 Talan Group

Talan, fondée en 2002, est un acteur dynamique dans le secteur de conseils et de services. Ses activités sont principalement la refonte des processus métiers et l'intégration des nouvelles technologies.

Cette organisme concentre son expertise sectorielle sur les grands opérateurs de services à savoir : Finance, assurance, énergie utilitaires, services publiques, transport logistique et télécom. Elle compte plus de 3000 consultants pour un chiffre d'affaires de près de 310 millions d'euros, répartis entre une dix-huitaine de pôles de développement en France, au Canada, à l'Espagne, en Tunisie, au Luxembourg, au Royaume Uni, à New York et à Hong Kong... Talan propose un cadre novateur et motivant pour fidéliser ses collaborateurs.

Le Groupe Talan a devancé la tendance - et validé la faisabilité - de délocalisation des projets de maîtrise d'oeuvre par l'installation dès 2008 d'un centre de développement « Near-shore » en Tunisie. Ce centre regroupe à ce jour plus de 400 ingénieurs.

Talan offre ses services à l'échelle internationale et mise sur les valeurs partagées par tous ses membres dont la transparence,le respect,le travail d'équipe, l'esprit entrepreneurial et d'implication, pour plus dominer son domaine d'activité et satisfaire ses clients par des prestations à la hauteur de la renommée de Talan. [1]

La figure 1.1 représente le logo de Talan Tunisie.

Figure 1.1: Logo Talan Tunisie

1.1.2 Secteurs d'activité

L'activité de Talan se partage entre les métiers du conseil et de l'assistance à la maîtrise d'ouvrage, les métiers liés aux Systèmes d'Information et à l'Intégration des nouvelles technologies et les métiers de Support autour des Systèmes d'Information Les activités de Talan regroupe plusieurs prestations telles que :

- Support aux projets de transformation.
- Refonte et optimisation des processus métiers.
- Alignement des systèmes d'information aux changements d'organisation et accompagnement au changement.

Comme illustré dans la figure 1.2, ces activités s'étalent sur quatre grands secteurs qui sont :

— **Secteur de finance :** en collaborant avec les assurances, les banques, etc

— **Secteur de Télécom :** en travaillant sur des projets destinés aux opérateurs télécom et un ensemble de fournisseurs d'internet.

— **Secteur public :** à travers l'MOA et l'intégration des nouvelles technologies dans des différents projets.

— **Secteur d'énergie :** en travaillant sur des projets de développement destinés aux opérateurs de service d'eau, électricité, gaz naturel, etc

Expert dans l'accompagnement de la transformation agile, Talan a mené avec succès plusieurs missions auprès de grands opérateurs grâce à son expertise de plus de 20 ans dans le domaine des nouvelles technologies et des télécoms. [1]

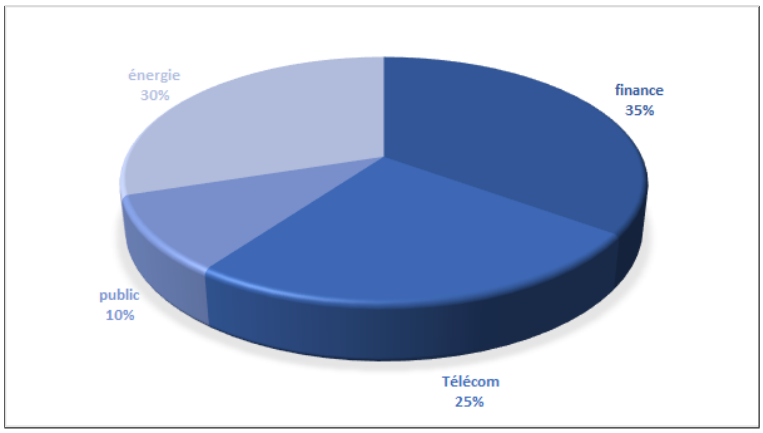


Figure 1.2: Répartition des activités de Talan

1.1.3 Talan Testing Factory

Notre projet de fin d'étude s'est déroulé dans l'équipe Talan Testing Factory de Talan Tunisie qui s'occupe des activités suivantes :

- Exécution des tests manuels.
- Tierce recette applicatif.
- Automatisation des test.
- Coaching / Accompagnement dans la mise en place de méthodologies de test.

Talan TESTING Factory comporte six équipes de test qui fournissent leurs services à plusieurs clients. Nous avons intégré une équipe de huit personnes travaillant dans le cadre de la TRA (Tierce recette applicatif).

Cette équipe fournit une activité de test sur un ensemble d'applications SI qui s'articulent autour de la même référence fonctionnelle.

Son activité couvre toutes les phases de test et intègre plusieurs compétences techniques autour de la gestion des IHM, de WS REST SOAP et de la base de données.

1.2 L'activité de Testing

Dans cette partie,nous allons présenter l'activité de testing et ses concepts de base pour rendre notre problématique facile à comprendre. Commençons par lister quelques définitions de notions fondamentales :

- **Anomalies :** Incompatibilité entre le résultat attendu et le resultat obtenu en phase de test.
- **Erreur :** Action humaine qui cause un résultat erroné différent de celui attendu.
- **Défaillance :** Capacité réduite d'un composant à remplir une fonction requise.
- **Bug (Défaut) :** Une imperfection dans un composant entraînant un problème de fonctionnement lors de l'exécution.

Créer un produit parfait est toujours impossible. Toute personne faisant partie d'une équipe de développement logiciel est susceptible de commettre des erreurs ou de négliger un détail, ce qui peut entraîner des anomalies et influer sur les résultats attendus.
Nous parlons donc d'une déception du client qui représente l'axe de l'activité de l'entreprise. De telles imperfections qui affectent la confiance du client peuvent automatiquement affecter la crédibilité et la réputation de l'entreprise sur le marché.
De plus, ces anomalies peuvent obliger l'entreprise à faire tout le travail pour satisfaire ses clients, et c'est très cher. La survenue de ces situations a fait de la phase de test une phase cruciale lors de la création d'un produit. La phase de test est essentielle tout au long du cycle de développement. Elle vous permet de détecter les anomalies avant de livrer un produit à l'utilisateur final et d'informer l'équipe chargée de la résolution des problèmes pouvant survenir avant la mise en production du logiciel.

1.2.1 Différents types de Tests

Il existe différents types de tests nous permettant d'avoir une vision globale de tous les défauts ou bugs existant dans un produit et couvrant tous ses aspects. Nous pouvons citer comme exemple :

- **Tests d'acceptation :** S'assure que le produit livré est opérationnel et conforme aux spécifications de l'utilisateur final.
- **Tests Unitaires :** ou des tests de composants qui consistent à vérifier le bon fonctionnement d'une partie bien spécifiée du logiciel.

- **Tests fonctionnels :** Vérifient les fonctionnalités d'un produit et vérifient s'ils répondent aux attentes de l'utilisateur.
- **Tests non-fonctionnels :** Ces tests mesurent et décrivent les caractéristiques d'un système.
- **Tests liés au changement (tests de confirmation et de régression) :** Chaque produit peut avoir plusieurs versions avec des modifications d'une version à l'autre. Après chaque modification, le risque de détérioration des fonctionnalités fournies précédemment est important. Ce type de test permet alors de vérifier le bon fonctionnement de ces fonctionnalités.

1.2.2 Recette informatique [2]

Pour faire référence à des réalités très souvent variées, nous parlons de recettes de tests d'essais qualificatifs au sens large. Dans un premier temps, une recette consiste en un contrôle et une vérification que le produit est conforme aux attentes formulées.

1.3 Étude et critique de l'existant

Dans cette section, nous présenterons une étude du processus de conception d'un scénario de test utilisé pour tracer et assurer une bonne couverture des tests qui seront planifiés et exécutés.
Ces cas de test seront écrits dans ce que nous appelons un cahier de recettes, où la création du cahier de recettes est considérée comme l'une des étapes les plus importantes de la phase de test. Un cahier de recettes est présenté sous la forme d'un rapport qui permet de clarifier toutes les exigences fonctionnelles requises par le client en décrivant tous les modules du produit et d'organiser correctement les cas de test à exécuter sous forme de scénarios. Chacun décrit le flux d'une fonctionnalité spécifique nécessitant la prise en compte d'une multitude de tests.
Actuellement, chez Talan, les processus d'écriture des tests dans le cahier sont gérés manuellement via des feuilles de calcul Excel.
Lors de la gestion des recettes d'un projet de base de données pour le compte de son client, l'équipe hôte passe par une étape de conception et entre des cas de test qui seront regroupés dans des cahiers de recettes.
Dans ce cas, un cahier de recettes n'est autre qu'un fichier Excel dans lequel le concepteur remplit les scénarios de test. Ce fichier servira d'entrée pour le logiciel HP ALM (Hewlett Packard Application

Lifecycle Management). Ce logiciel est un outil Web permettant de gérer l'ensemble du cycle de vie d'un projet, de la planification à la collecte des exigences, en passant par les tests et le déploiement, tâches fastidieuses.

Les neuf colonnes du modèle illustré par la figure 1.3 sont comme suit :

- **Type :** Contient le type d'exécution de test (manuel ou automatique).
- **Test Name :** Contient le titre du scénario précédé de l'acronyme du périmètre d'exécution de celui-ci et d'une numérotation du scénario dans le cahier.
- **Description :** Une description de l'objectif du test.
- **Step Name (Design Steps) :** Numérotation des étapes du scénario.
- **Description (Design Steps) :** Description de chaque étape.
- **Expected Result (Design Steps) :** Le résultat attendu après l'exécution du test.
- **Subject :** Le périmètre du cahier de tests.
- **Designer :** C'est une colonne dédiée pour le nom du concepteur du scénario.
- **Creation Date :** C'est la date de création du test

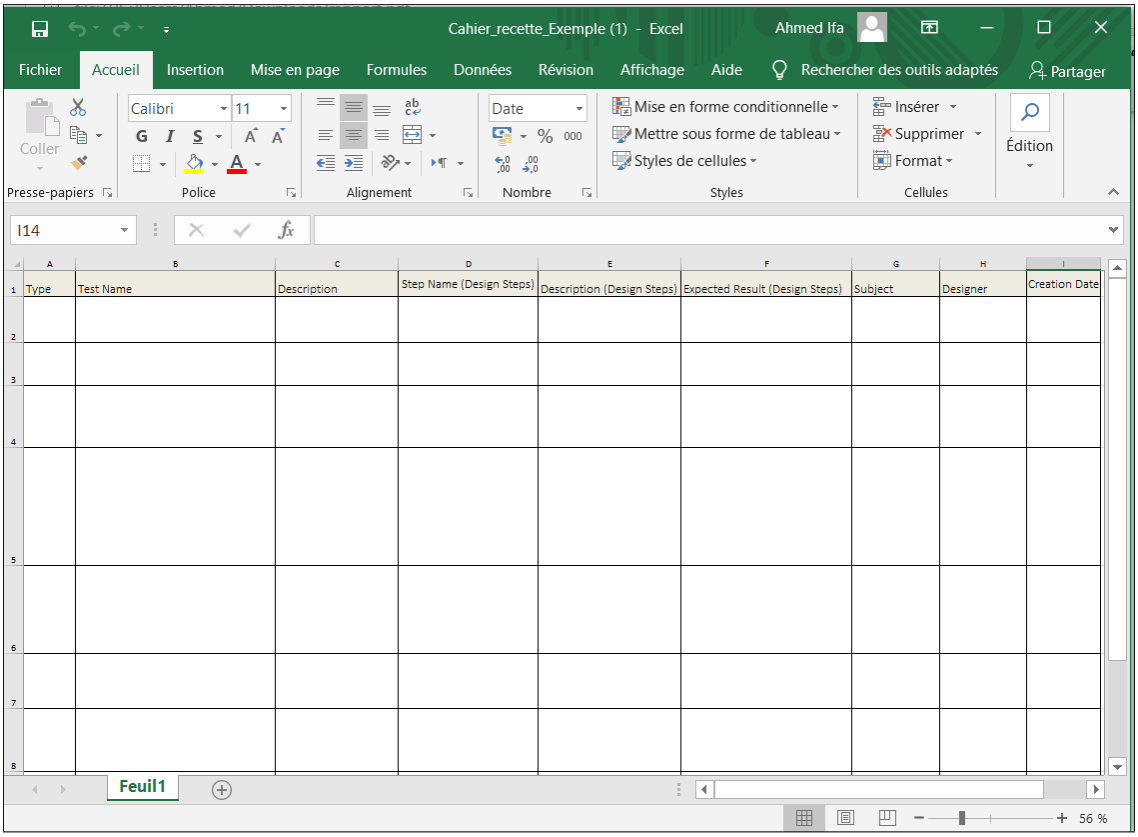


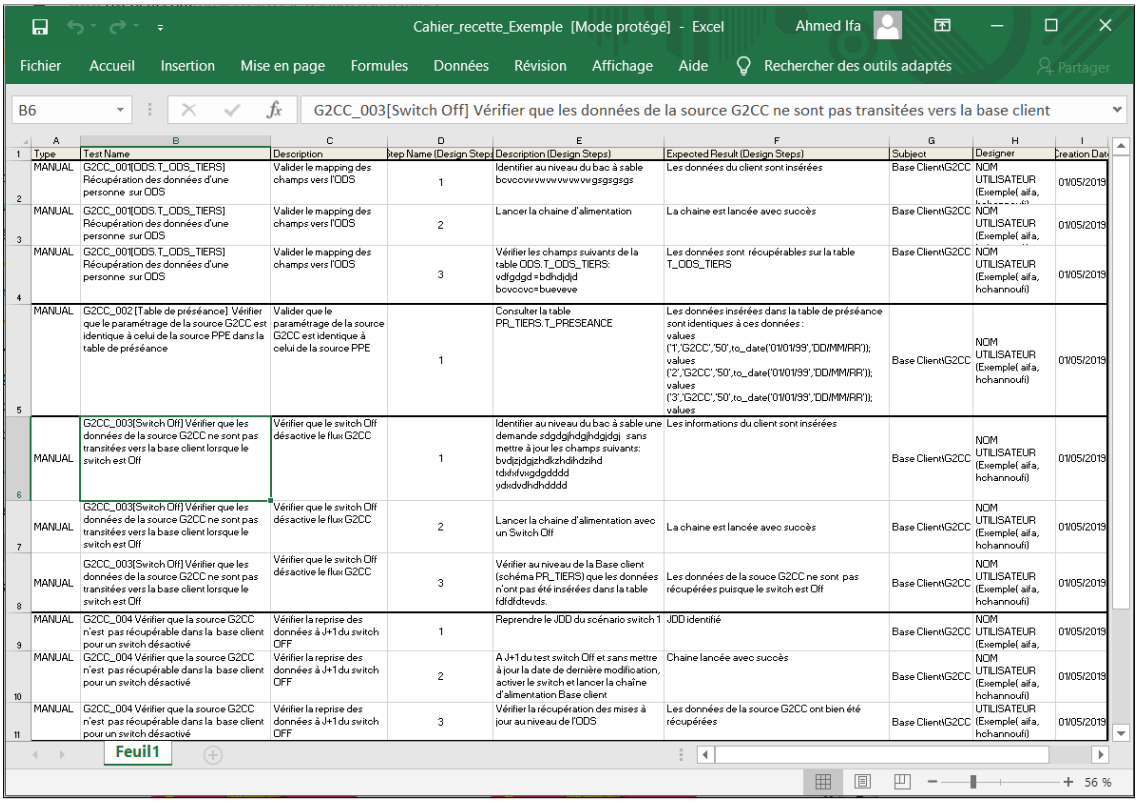
Figure 1.3: Modèle du fichier Excel entrée pour HP ALM

Une fois le cahier de recettes prêt et pour pouvoir l'importer dans l'outil HP ALM, le testeur utilise l'ALM Microsoft Excel Add-in.

Une fois ajouté à Micrsoft Excel, ce Add-in permet d'exporter le plan de test Microsoft Excel vers HP ALM. Ce plan de test exporté peut être affiché dans l'arborescence du plan de test de l'outil de gestion pour une utilisation ultérieure.

Ce processus de rédaction de cahiers de test présente des inconvénients et un certain nombre d'imperfections, comme :

- L'impossibilité de respecter les délais dans la plupart des cas.
- Perte d'énergie et de temps en remplissant chaque ligne avec des cellules redondantes.
- Le risque d'erreurs humaines augmente avec le nombre important de scripts de test à écrire et leur similarité.
- Une perte de temps pour classer et récupérer les fichiers Excel à chaque fois pour reprendre le remplissage.
- La gêne des utilisateurs peu familiers avec Excel.



	A	B	C	D	E	F	G	H	I
1	Type	Test Name	Description	Step Name (Design Steps)	Description (Design Steps)	Expected Result (Design Steps)	Subject	Designer	Creation Date
2	MANUAL	G2CC_001[ODS.T_ODS_TIERS] Récupération des données d'une personne sur ODS	Valider le mapping des champs vers l'ODS	1	Identifier au niveau du bac à sable bcvccvvvvvvvvvvgsgsgsgs	Les données du client sont insérées	Base Client\G2CC	NOM UTILISATEUR (Exemple(aifa,	01/05/2019
3	MANUAL	G2CC_001[ODS.T_ODS_TIERS] Récupération des données d'une personne sur ODS	Valider le mapping des champs vers l'ODS	2	Lancer la chaine d'alimentation	La chaine est lancée avec succès	Base Client\G2CC	NOM UTILISATEUR (Exemple(aifa,	01/05/2019
4	MANUAL	G2CC_001[ODS.T_ODS_TIERS] Récupération des données d'une personne sur ODS	Valider le mapping des champs vers l'ODS	3	Vérifier les champs suivants de la table ODS.T_ODS_TIERS: vdfgdgd=bdhdjdjd bcvccvc=bueveve	Les données sont récupérables sur la table T_ODS_TIERS	Base Client\G2CC	NOM UTILISATEUR (Exemple(aifa, hchannoufi)	01/05/2019
5	MANUAL	G2CC_002 [Table de préséance] Vérifier que le paramétrage de la source G2CC est identique à celui de la source PPE dans la table de préséance	Valider que le paramétrage de la source G2CC est identique à celui de la source PPE	1	Consulter la table PR_TIERS.T_PRESEANCE	Les données insérées dans la table de préséance sont identiques à ces données : values ('1','G2CC','50',to_date('01/01/99','DD/MM/RR')); values ('2','G2CC','50',to_date('01/01/99','DD/MM/RR')); values ('3','G2CC','50',to_date('01/01/99','DD/MM/RR')); values	Base Client\G2CC	NOM UTILISATEUR (Exemple(aifa, hchannoufi)	01/05/2019
6	MANUAL	G2CC_003[Switch Off] Vérifier que les données de la source G2CC ne sont pas transitées vers la base client lorsque le switch est Off	Vérifier que le switch Off désactive le flux G2CC	1	Identifier au niveau du bac à sable une demande sdgdgjhdgjhdgjdgj sans mettre à jour les champs suivants: bvdjzjdgjzhdkzhdlhdzlhd tdvlvlvlwgdgdddd ydvdvdhdhddddd	Les informations du client sont insérées	Base Client\G2CC	NOM UTILISATEUR (Exemple(aifa, hchannoufi)	01/05/2019
7	MANUAL	G2CC_003[Switch Off] Vérifier que les données de la source G2CC ne sont pas transitées vers la base client lorsque le switch est Off	Vérifier que le switch Off désactive le flux G2CC	2	Lancer la chaine d'alimentation avec un Switch Off	La chaine est lancée avec succès	Base Client\G2CC	NOM UTILISATEUR (Exemple(aifa, hchannoufi)	01/05/2019
8	MANUAL	G2CC_003[Switch Off] Vérifier que les données de la source G2CC ne sont pas transitées vers la base client lorsque le switch est Off	Vérifier que le switch Off désactive le flux G2CC	3	Vérifier au niveau de la Base client (schéma PR_TIERS) que les données n'ont pas été insérées dans la table fdfdfdtevds.	Les données de la souce G2CC ne sont pas récupérées puisque le switch est Off	Base Client\G2CC	NOM UTILISATEUR (Exemple(aifa, hchannoufi)	01/05/2019
9	MANUAL	G2CC_004 Vérifier que la source G2CC n'est pas récupérable dans la base client pour un switch désactivé	Vérifier la reprise des données à J+1 du switch OFF	1	Reprendre le JDD du scénario switch 1	JDD identifié	Base Client\G2CC	NOM UTILISATEUR (Exemple(aifa,	01/05/2019
10	MANUAL	G2CC_004 Vérifier que la source G2CC n'est pas récupérable dans la base client pour un switch désactivé	Vérifier la reprise des données à J+1 du switch OFF	2	A J+1 du test switch Off et sans mettre à jour la date de dernière modification, activer le switch et lancer la chaîne d'alimentation Base client	Chaine lancée avec succès	Base Client\G2CC	NOM UTILISATEUR (Exemple(aifa, hchannoufi)	01/05/2019
11	MANUAL	G2CC_004 Vérifier que la source G2CC n'est pas récupérable dans la base client pour un switch désactivé	Vérifier la reprise des données à J+1 du switch OFF	3	Vérifier la récupération des mises à jour au niveau de l'ODS	Les données de la source G2CC ont bien été récupérées	Base Client\G2CC	UTILISATEUR (Exemple(aifa, hchannoufi)	01/05/2019

Figure 1.4: Exemple d'un brouillon de cahier de test rédigé à l'aide de Microsoft Excel

1.4 Solution proposée

Pour simplifier la tâche de rédaction des cas de test, nous avons opté pour une solution Web qui répondra aux besoins des testeurs en matière de consultation, de configuration et de manipulation des cahiers de recettes, en respectant les tests d'un projet bien défini, de manière simple et rapide, en exploitant des interfaces conviviales qui garantissent une bonne expérience utilisateur.
La rédaction des nouveaux cahiers de recettes devrait être plus simple, plus rapide et plus rigoureuse en créant une solution pour exploiter la nature stéréotypée des scénarios de test qui se manifeste par la présence d'un catalogue de modèles prédéfinis.

Notre solution va permettre à l'utilisateur de créer puis exporter un cahier de recette d'une manière beaucoup plus facile que la méthode manuelle. L'application va générer automatiquement ce que le testeur avait réécrit de la même façon plusieurs fois ou parfois avec un petit changement dans un ou deux mots dans le script ou la description des scénarios de tests.
Sans avoir utilisé un fichier Excel ou naviguer entre un grand nombre de lignes et de colonnes du tableur, le testeur peut créer et compléter un cahier de recette à travers les interfaces claires et conviviales de notre solution et puis faire sortir un "output" final dans un fichier Excel prêt à être importé sur HP ALM.

Grâce à ces avantages, notre générateur de cahier de recette va garantir aux créateurs et manipulateurs de ces cahiers essentiellement le gain de beaucoup de temps et la diminution des possibilités des erreurs humaines lors de la rédaction des scripts.

1.5 Méthodologie adoptée

Suivre une méthodologie de conception est le premier garant d'une production de qualité. A Talan Tunisie, les méthodologies agiles et plus précisément la méthode SCRUM est la plus adoptée, étant donné la tolérance aux changements et la flexibilité, la transparence, ainsi que la courte durée des itérations. Dans cette partie, nous présenterons la méthode SCRUM que nous adopterons.
Une méthodologie agile est un processus basé sur la subdivision appropriée d'un projet en plusieurs parties dont chaque partie a un délai bien défini. Cela implique également une collaboration entre les différentes équipes inter-fonctionnelles afin de fournir dans les délais prédéfinis des versions du

produit, appelées itérations.

1.5.1 Méthode SCRUM

Scrum est une méthode agile dédiée à la "gestion de projet". Cette méthode, ou plutôt ce Framework de gestion de projet, vise à améliorer la productivité de son équipe.
Dans le cadre de notre projet, nous avons opté pour la méthode SCRUM. En fait, ce choix semble être le meilleur car la manière dont il organise les réunions chaque jour commence, favorise le climat de communication, pose mieux les problèmes, élimine rapidement les blocages et encourage l'entraide jusqu'à ce que les travaux soient terminés.
Les incréments se produisent successivement pendant des périodes de durée fixe au début du projet de une à quatre semaines, sous forme de sprints.
Défini par le chef de produit, chaque sprint a au préalable un objectif. À la fin d'un sprint, un produit partiel fonctionnel est livré.
SCRUM insiste sur la participation active du client pour définir les priorités des fonctionnalités du produit et pour choisir le sprint auquel chaque événement appartient. La liste des fonctionnalités à produire n'est jamais définitive, elle peut être modifiée à tout moment, mais jamais celles en cours de réalisation.
L'équipe SCRUM est composée principalement par : le propriétaire de produit(Product Owner), l'équipe de développement (Development Team), et d'un SCRUM Master.

- **Product Owner :** La personne responsable de la vision d'un produit et de la priorisation des exigences. Il définit les besoins du projet.
- **SCRUM Master :** Élimine les problèmes, assure le respect des règles SCRUM tout au long du projet et forme également l'équipe.
- **Development Team :** Elle n'as pas un rôle bien précis, elle est également composé de toutes les personnes nécessaires à la réalisation du produit telles que les développeurs, les architectes, les "business analysts", les testeurs, etc.

Pour notre cas et notre projet nous avons les responsabilités suivantes :

— **Product Owner :** L'encadrante technique Mme. GHERAB Olfa.

— **SCRUM Master :** L'encadrante technique Mme. BOUDABOUS Imen.

— **Team :** est composée de moi même IFA Ahmed.

La figure ci-dessous présente les caractéristiques de SCRUM ainsi que le déroulement du projet avec cette méthode.

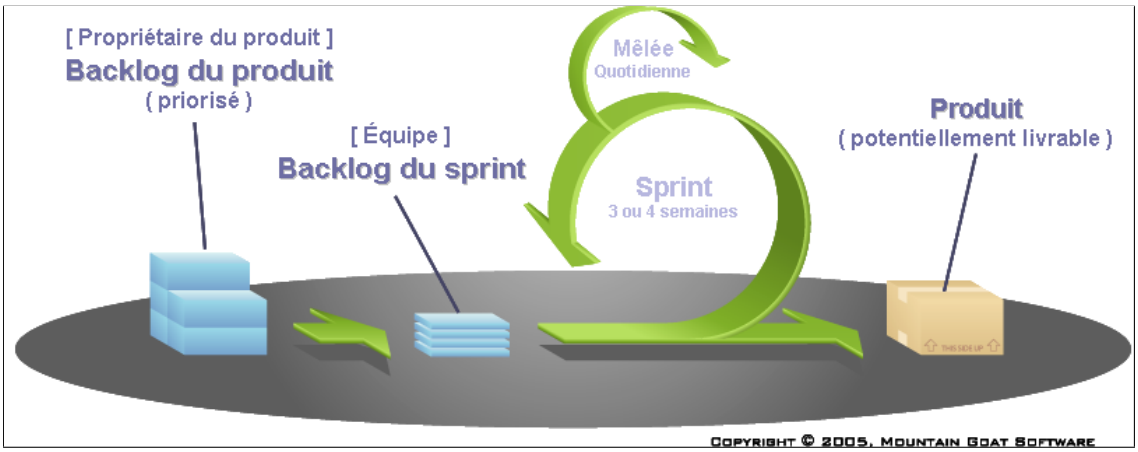


Figure 1.5: Processus de SCRUM [3]

— **SCRUM Board :**

Cet élément de la méthode SCRUM est totalement indispensable, il permettra de suivre la progression de la réalisation de différentes User Stories.

Il s'agit d'un tableau accessible et visible pour l'ensemble de l'équipe du projet et permettant un suivi en temps réel de l'évolution des tâches.

Le SCRUM Board est divisé en au moins trois parties : les tâches à effectuer, les tâches en cours et les tâches terminées. Pour une gestion plus précise et efficace, les équipes peuvent ajouter des colonnes supplémentaires. Les User Stories seront ensuite déplacées de gauche à droite et colonne par colonne.

Dans notre projet, nous avons utilisé l'outil Trello pour réaliser notre Scrum Board qui est composé de plusieurs listes pour mieux préciser l'état de chaque carte. Alors on trouve le Backlog Produit, le Backlog du sprint en cours de réalisation, les listes "ToDo", "doing" et "done", et les deux listes supplémentaires "Verify" pour faire un appel à la vérification d'une carte réalisée et "Blocked" pour réclamer qu'il y ait un blocage dans une tâche précise.

La figure 1.6 présente une capture du tableau de Scrum dans l'outil Trello.

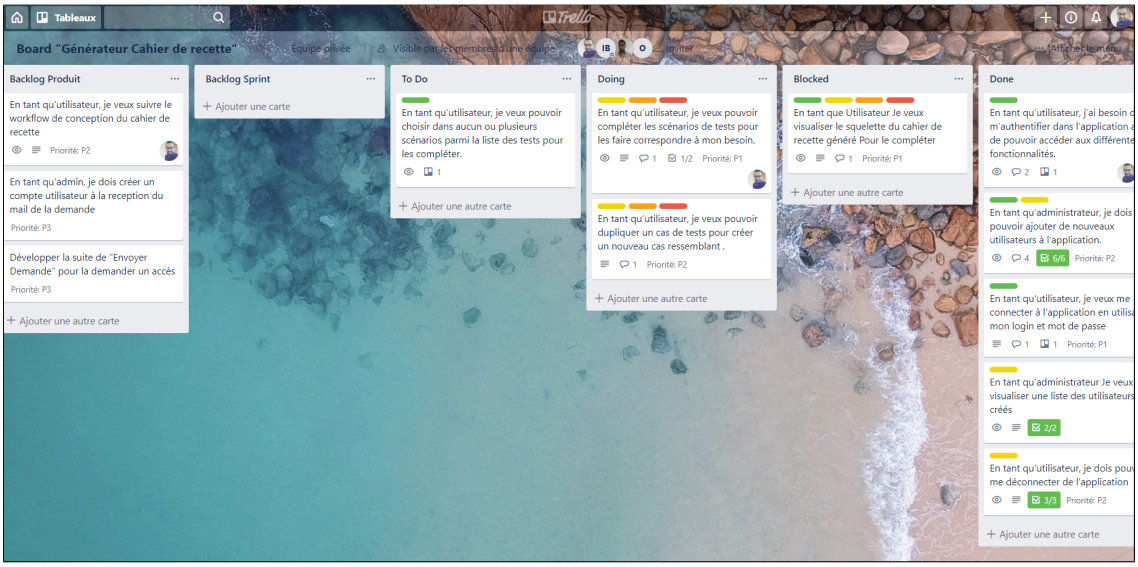


Figure 1.6: SCRUM Board avec l'outil Trello

Conclusion

Dans ce chapitre, nous avons présenté l'organisme d'accueil. Nous avons fait une étude de l'existant pour éclaircir ses limites et mettre en valeur notre solution. Dans ce chapitre aussi on a présenté des concepts de base liés au contexte du projet et on a mis en évidence le choix de la méthodologie de travail.

Chapitre 2

Sprint 0 : Analyse et spécification des besoins

Plan

1 Analyse des besoins 16

2 Backlog Produit 18

3 Diagramme de classe d'analyse 20

4 Choix des technologies et l'environnement de travail 21

5 Architecture logique et physique globale 22

6 Planification des sprints 24

Introduction

Au cours de ce chapitre, nous allons présenter le sprint 0 intitulé "Analyse et spécification des besoins". Dans ce sprint, nous allons commencer par l'analyse des besoins fonctionnels et non fonctionnels ainsi que l'identification des acteurs de notre système. Puis, nous allons réaliser le Backlog produit, le diagramme de cas d'utilisation et de classe d'analyse. Par la suite, nous présenterons nos choix de technologies et de l'environnement de travail et nous finirons avec la planification des sprints.

2.1 Analyse des besoins

Le but de l'étape de spécification des besoins est de mentionner en détail toutes les fonctionnalités fournies par le système. Les besoins collectés seront divisés en deux, les besoins fonctionnels et les besoins non fonctionnels.

2.1.1 Identification des acteurs

Les acteurs sont des entités, physiques ou autres, externes au système, qui interagissent avec ce dernier dans le but d'exploiter les services qu'il offre ou de fournir des services qui l'aident dans son fonctionnement.
Pour notre système, nous avons deux acteurs :

- **Utilisateur :** L'acteur principal du système qui créera un cahier de recettes à partir d'un stéréotype et complétera ses scénarios de test. Il est également responsable de la gestion des cahiers de recettes.
- **Administrateur :** Il est l'acteur secondaire responsable de l'approbation de l'accès aux utilisateurs de l'application et de leur gestion.

2.1.2 Besoins fonctionnels

Les besoins fonctionnels sont les besoins spécifiant les actions qu'un système doit effectuer, sans tenir compte de contraintes physiques.
Dans ce contexte, notre application doit permettre à ses utilisateurs d'exécuter les fonctions suivantes :

- Authentification et Gestion des utilisateurs de l'application (Administrateurs et testeurs) :

L'utilisateur ou l'administrateur peut s'authentifier en entrant son login et son mot de passe pour accéder aux différents fonctionnalités de l'application. L'administrateur peut ajouter de nouveaux utilisateurs, il a le pouvoir de supprimer des utilisateurs aussi.

- Gestion du profile personnel : L'utilisateur ou l'administrateur de l'application peut modifier ses informations personnelles telles que son nom, prénom ou mot de passe à partir de son profile personnel.
- Création de cahiers de recettes à partir d'un modèle : L'utilisateur peut créer un cahier de recette après le choix du modèle convenable à son besoin.
- Gestion de la liste des cahiers de recette : Cette liste permet à l'utilisateur de consulter l'historique des listes de cahiers des recettes déjà créées.
- Visualisation des cahiers de recette : Il s'agit d'un aperçu du cahier de recette qui permet à l'utilisateur de consulter le contenu d'un cahier.
- Exportation des cahiers de recettes sous format .xlsx : Le téléchargement du cahier de recette sous format d'un tableur Excel prêt à être importé sur HP ALM.
- Gestion du catalogue des modèles de cahier de recette : L'utilisateur peut gérer un catalogue de modèles de cahier de recette afin d'avoir des modèles qui répondent à son besoin.
- Gestion du catalogue des modèles de scénarios de test : L'utilisateur gère les modèles de scénarios pour rédiger des scénarios de tests qui répondent à son besoin pour remplir le cahier.

2.1.3 Besoins non fonctionnels

Les principaux besoins non fonctionnels de notre application sont résumés autour des points suivants :

— **Sécurité :** L'application doit assurer l'authentification des utilisateurs en sécurité. Pour accéder à l'application, un utilisateur doit d'abord s'authentifier en saisissant son login et son mot de passe. L'application doit respecter la confidentialité des données en utilisant le cryptage du mot de passe. Cela sera réalisé en utilisant Spring Security.

— **Ergonomie :** L'application doit afficher des interfaces conviviale et facile à utiliser. Les différents parties des interfaces de l'application doivent être claires et faciles à comprendre. La navigation entre les différentes fonctionnalités de l'application doit être possible à partir de n'importe quelle page.

— **Performance :** Le temps de réponse de l'application doit être optimisé. Pour assurer la performance de l'application on utilise par exemple le "lazy loading" des différents modules pour avoir un temps de réponse réduit.

2.1.4 Diagramme de cas d'utilisation

Dans cette section, on va présenter les besoins en utilisant le diagramme de cas d'utilisation global. Le diagramme ci-dessous présente les fonctionnalités globales offertes par l'application.
La figure 2.1 présente le diagramme de cas d'utilisation global.

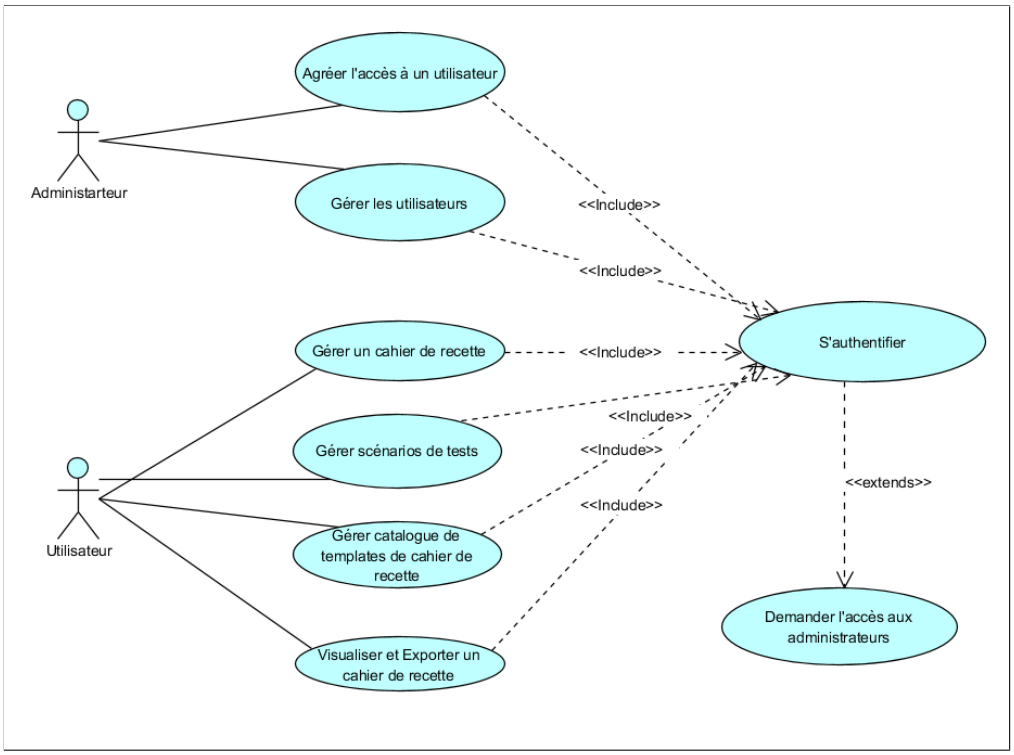


Figure 2.1: Diagramme de cas d'utilisation global

2.2 Backlog Produit

Le Backlog Produit de Scrum permet d'assembler tous les besoins fonctionnels ou techniques à réaliser pour le produit souhaité. Il est donc une liste des fonctionnalités appelées des histoires utilisateurs (User Story), ou techniques appelées des histoires techniques(Technical Story).
Le tableau ci-dessous présente le Backlog Produit de notre application. Il est à noter que ce backlog ne contient pas les histoires techniques comme la préparation des maquettes graphiques, la conceptions, etc.

ID	Feature	User Story	Priority
F	F.1 Gérer Utilisateurs	F.1.1 En tant qu'administrateur je veux ajouter un nouvel utilisateur	1
		F.1.2 En tant qu'administrateur je veux supprimer un utilisateur	2
		F.1.3 En tant qu'utilisateur/administrateur, j'ai besoin de m'authentifier dans l'application afin de pouvoir accéder aux différentes fonctionnalités.	1
		F.1.4 En tant qu'utilisateur/ administrateur, je veux modifier mes informations à partir de mon profil	2
		F.1.5 En tant qu'utilisateur je veux demander l'accès aux administrateurs pour avoir mon login et mot de passe de connexion à l'application.	3
	F.2 Gérer le catalogue de templates de cahier de recette	F.2.1 En tant qu'utilisateur je veux ajouter un template au catalogue pour enrichir ma base de référence.	1
		F.2.2 En tant qu'utilisateur je veux modifier un template du catalogue pour mettre à jour ma base de référence	1
		F.2.3 En tant qu'utilisateur je veux Supprimer un template du catalogue pour éviter l'encombrement des modèles inutiles.	2
	F.3 Gérer scénarios de tests	F.3.1 En tant qu'utilisateur je veux ajouter un scénario de test pour enrichir ma base de référence.	1
		F.3.2 En tant qu'utilisateur je veux modifier un scénario de test pour mettre à jour ma base de référence.	1
		F.3.3 En tant qu'utilisateur je veux Supprimer un scénario de test.	1
	F.4 Gérer un cahier de recette	F.4.1 En tant qu'utilisateur, je veux créer un cahier de recette à partir d'un template pour le pouvoir compléter.	1
		F.4.2 En tant qu'utilisateur, je veux Supprimer un cahier de recette pour éviter l'encombrement des cahiers périmés	1
		F.4.3 En tant qu'utilisateur, je veux modifier un cahier de recette.	2
		F.4.4 En tant qu'utilisateur, je veux visualiser un cahier de recette pour pouvoir consulter son contenu pour le vérifier.	2
	F.5 Exporter un cahier de recette	F.5.1 En tant qu'utilisateur, je veux exporter le cahier de recette préparé en format Excel pour l'intégrer dans ALM.	1

Tableau 2.1: Backlog produit

2.3 Diagramme de classe d'analyse

Après avoir cité les cas d'utilisation, nous passons au diagramme de classe d'analyse décrit par les relations suivantes :

Un utilisateur gère un ou plusieurs cahiers de recette. Un cahier de recette est géré par un ou plusieurs utilisateurs.

Un cahier de recette comporte un ou plusieurs template de scénarios. Un template appartient à un cahier de recette.

Un template contient un ou plusieurs scénarios de test et un scénario appartient à un template.

Un scénario comporte un ou plusieurs étapes et une étape appartient à un scénario.

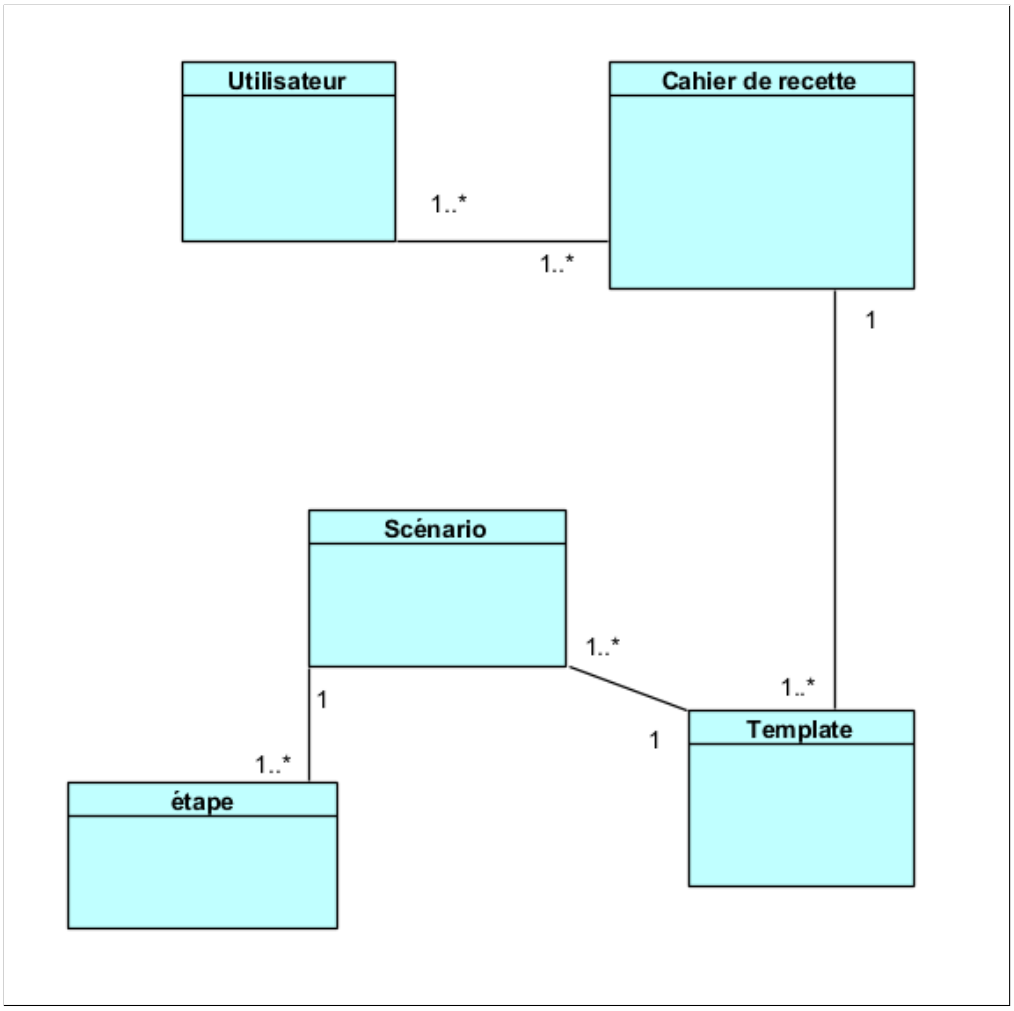


Figure 2.2: Diagramme de classe d'analyse

2.4 Choix des technologies et l'environnement de travail

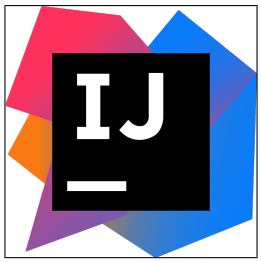
2.4.1 Technologies

— **Spring :** Spring est un Framework libre qui permet la construction et la définition d'une application Java. Il facilite le développement et les tests pour l'application.

— **Angular :** Angular est une plate-forme qui facilite la création des applications Web. Angular combine des modèles déclaratifs, une injection de dépendance, des outils de bout en bout et des modules intégrées afin de résoudre les problèmes de développement.

2.4.2 Environnement de travail

— **IntelIJ IDEA :** est un environnement de développement intégré (en anglais Integrated Development Environment - IDE) de technologie Java destiné au développement de logiciels informatiques. Il est développé par JetBrains [4]



— **WebStorm :** est un IDE pour les langages Web (HTML, CSS et JavaScript), développé par l'entreprise JetBrains et basé sur la plate-forme IntelliJ IDEA. [5]

— **WampServer :** est une plate-forme de développement Web de type WAMP, permettant de faire fonctionner localement (sans avoir à se connecter à un serveur externe) des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant trois serveurs (Apache, MySQL et MariaDB), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL. [6]

2.5 Architecture logique et physique globale

L'architecture logique est la décomposition logique du projet en couches, l'architecture physique qu'on a choisit pour atteindre notre application est l'architecture 3-tiers.

2.5.1 Architecture logique

Dans cette partie, nous décrirons l'architecture logique de notre application qui sera basée sur la division logique et sur la manière de regrouper les composants en fonction du type de fonction et des traitements qu'ils effectuent. Ainsi, nous avons opté pour une architecture multicouche illustrée par la figure 2.3 suivante composée de : Les couches de l'architecture logique :

— **La couche présentation :** Elle fournit la logique de navigation et est composé de composants Angular utilisés pour représenter graphiquement les données. Cette couche utilise la couche de service pour répondre aux demandes des clients.

— **La couche service :** Les traitements sont implémentés en tant que services REST et communiqués à la couche de présentation à partir de la couche de métier.

— **La couche métier :** Elle contient tous les traitements métier et communique avec la couche persistance pour répondre aux requêtes de la couche présentation à l'aide des services REST de la couche service.

— **La couche persistance :** Elle joue le rôle de persistance et stockage des données dans le SGBD (MySql) ainsi que la récupération des données quand nécessaire. Elle est aussi l'endroit

qui fait le lien (mapping) entre l'objet java et une table dans la base de données.

Le schéma ci-dessous présente l'architecture logique de la partie Back de notre application.

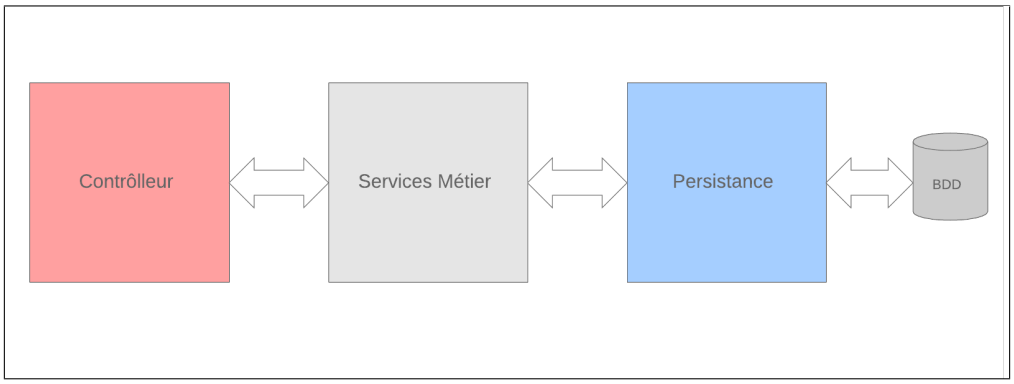


Figure 2.3: Architecture logique du BackEnd

Le schéma ci-dessous présente l'architecture logique de la partie Front de notre application. Ce schéma let en valeur la communication entre les composants de notre solution frontale implémenté avec Angular. L'utilisation des services de la part des composants et la présentation de ces opérations dans les vues du template de l'application (les interfaces) qui utilisent de leurs part les directives pour organiser le routage entre les pages.

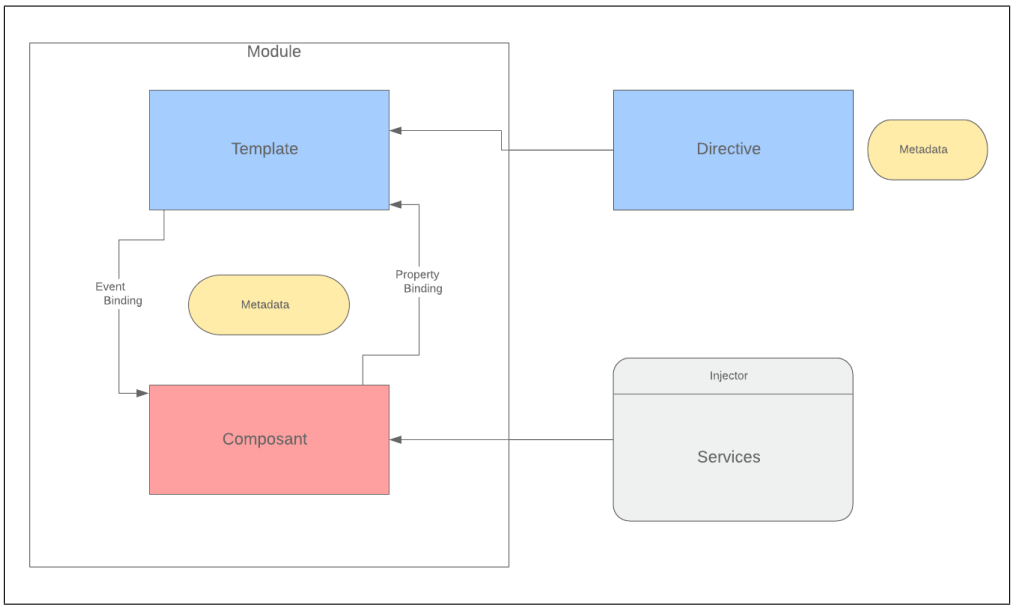


Figure 2.4: Architecture logique du FrontEnd

2.5.2 Architecture physique

La figure 2.5 présente le diagramme de déploiement de notre application qui met en valeur l'architecture physique de l'application. Le diagramme montre l'interaction entre la base de données et la partie BackEnd à travers les requêtes Sql, entre les parties BackEnd et FrontEnd en utilisant le Rest Api et enfin entre la partie frontale et le client (le navigateur) à travers des requêtes Http.

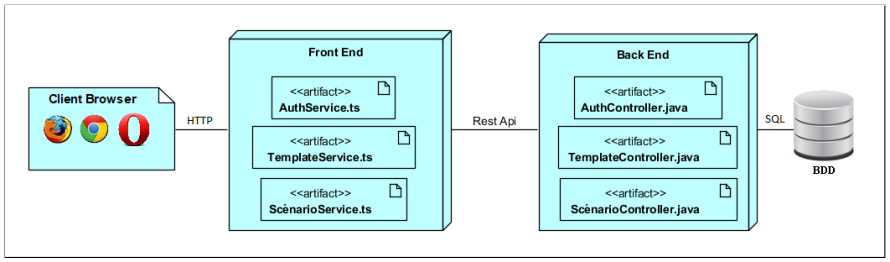


Figure 2.5: Diagramme de déploiement

2.6 Planification des sprints

Dans cette section, nous allons présenter la planification de notre travail en utilisant la méthode Scrum ; c'est à dire le découpage de la période de réalisation de notre produit en des sprints. Le tableau ci-dessous présente la planification des sprints. Chaque sprint s'étale sur 3 semaines.

Sprint	**Fonctionnalités**	**Durée**
Sprint 0	-Analyse -Spécification des besoins -Choix de la méthodologie et des technologies	De 19 Février à 11 Mars
Sprint 1	- Gestion d'utilisateur	De 12 Mars à 8 Avril
Sprint 2	-Gestion de catalogue de template	De 9 Avril à 30 Avril
Sprint 3	-Gestion de scénario de test	De 7 Mai à 29 Mai
Sprint 4	- Gestion de cahier de recette -Exportation de cahier de recette	De 3 Juin à 30 Juin

Tableau 2.2: Tableau de planification des sprints

Conclusion

Dans ce chapitre, nous avons fait une capture des besoins fonctionnels et non fonctionnels de notre solution ainsi que l'identification des acteurs et la réalisation du Backlog produit. Ensuite nous avons passé aux technologies et l'environnement de travail qu'on a choisi pour effectuer notre travail. Enfin, nous avons présenté l'architecture physique et logique de notre application et nous avons fait une planification des sprints du projet.

Chapitre 3

Sprint 1 : Gestion des utilisateurs

Plan

1 **Backlog du sprint** 27

2 **Analyse** 27

3 **Conception** 32

4 **Réalisation** 35

Introduction

Au cours de ce chapitre, nous présenterons le premier sprint intitulé "Gestion des utilisateurs" dans lequel nous allons faire une étude analytique et conceptuelle des cas d'utilisation de la gestion des utilisateurs . Enfin, nous terminerons avec la partie réalisation dans laquelle nous allons présenter les interfaces reliées à ce sprint tel que l'interface du login, ajout d'utilisateur, etc.

3.1 Backlog du sprint

Le tableau ci-dessous présente le backlog du sprint 1 contenant les User Stories reliés a ce sprint avec les taches effectués pour chaque User Story.

User story	Tâches	Estimation (en jours)
En tant qu'utilisateur/administrateur, j'ai besoin de m'authentifier dans l'application, pour pouvoir accéder aux différentes fonctionnalités.	Développement de l'interface d'authentification	3
	Développement du WS de l'ajout du template	
	Développement d'ajout de nouvel utilisateur.	
En tant qu'administrateur, je veux ajouter un nouvel utilisateur à l'application.	Développement d'ajout de nouvel utilisateur.	3
	Développement du WS de modification du template	
	Développement de l'interface de suppression de l'utilisateur	
En tant qu'administrateur, je veux supprimer un utilisateur existant.	Développement de l'interface de suppression de l'utilisateur	2
	Développement du WS de la suppression du template	
	Développement de l'interface du profil de l'utilisateur.	
En tant qu'utilisateur/administrateur, je veux modifier mes informations pour mettre mon profil à jour.	Développement de l'interface du profil de l'utilisateur.	3
	Préparation du WS de modification des informations de l'utilisateur.	
	Consommation du WS dans le FrontEnd.	
En tant qu'utilisateur je veux demander l'accès aux administrateurs pour avoir mon login et mot de passe de connexion à l'application.	Préparation de l'interface de demande d'accès.	3
	Développement de l'API de l'envoie de l'email et exposition du WS	
	Consommation du WS dans le FrontEnd.	

Tableau 3.1: Backlog du Sprint 1

3.2 Analyse

Ceci est le diagramme de cas d'utilisation du sprint 1, contenant les cas d'utilisation reliés a ce sprint.

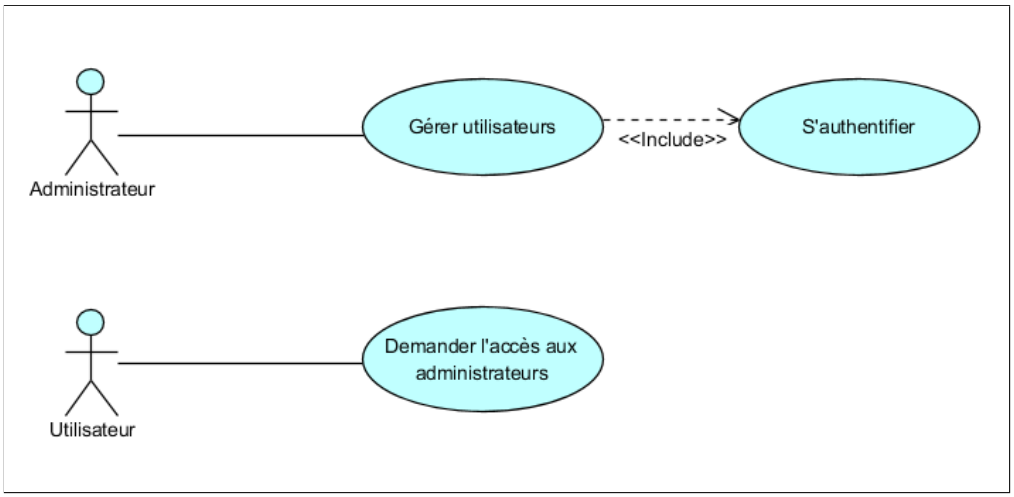


Figure 3.1: Diagramme de cas d'utilisation du Sprint 1

3.2.1 Diagramme de cas d'utilisation raffiné

La figure 3.2 présente le raffinement des cas d'utilisation du premier sprint dont l'authentification, l'ajout et la suppression d'un utilisateur, la modification des informations à partir du profile et la demande d'accès aux administrateurs.

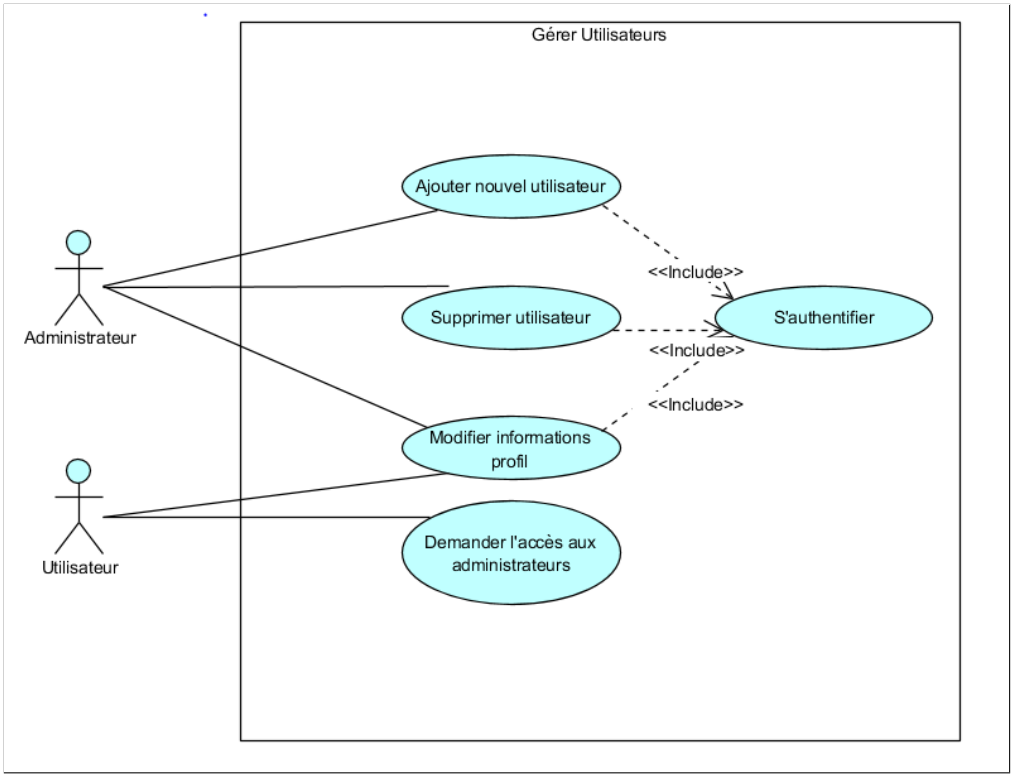


Figure 3.2: Diagramme de cas d'utilisation raffiné

3.2.2 Analyse du cas d'utilisation "S'authentifier"

- **Description textuelle du cas d'utilisation "S'authentifier" :**

Le tableau 3.2 montre la description textuelle du cas d'utilisation "s'authentifier". Dans ce tableau on trouve l'objectif et les acteurs de ce cas, on présente les préconditions et les postconditions pour que le cas d'utilisation soit vérifié. Enfin, dans la description textuelle, on rédige les deux scénarios : nominal (déroulement idéale des actions) et alternatif (description des étapes différentes liées aux choix de l'utilisateur).

Titre	S'authentifier
But	Permettre à l'utilisateur ou l'administrateur de s'authentifier
Acteurs	Administrateur, Utilisateur
Précondition	L'administrateur/utilisateur saisit son login et son mot de passe
Postcondition	Ouverture de la page d'accueil de l'administrateur/utilisateur
Scénario nominal	N.1- Le système affiche le formulaire d'identification N.2- L'administrateur/utilisateur saisit son login et son mot de passe dans le formulaire. N.3- Le système vérifie les informations saisies et renvoie vers la page d'accueil.
Scénario alternatif	A.1. L'utilisateur n'as pas saisie les bons identifiants. A.1.1 Le système renvoie un message d'erreur et signale à l'utilisateur de recommencer. A1.2 Retour à l'action N.1 du scénario nominal. A.2. L'utilisateur n'existe pas dans la base de données. A.2.1 Le système renvoie un message d'erreur. A2.2 Retour à l'action N.1 du scénario nominal.

Tableau 3.2: Description textuelle du cas d'utilisation "S'authentifier"

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

Dans la figure 3.3 on trouve le diagramme de séquence système du cas d'utilisation "s'authentifier".

Ce diagramme met en évidence l'interaction et les messages transmis entre l'utilisateur et le système, avec le scénario des événements effectués en cours de cette interaction.

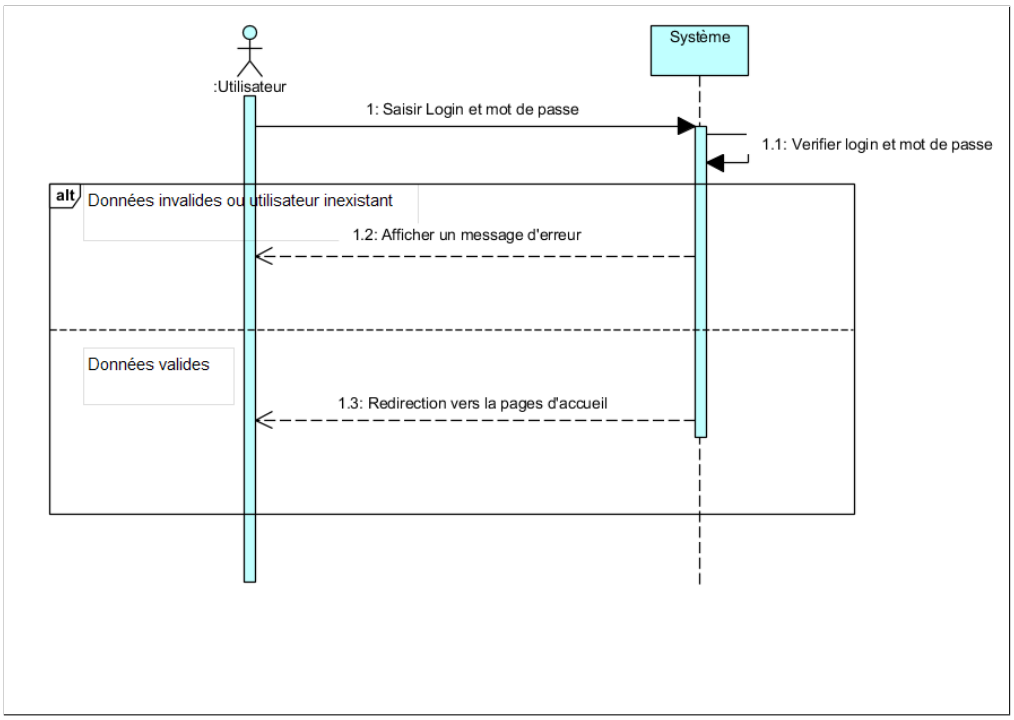


Figure 3.3: Diagramme de séquence système du cas d'utilisation "S'authentifier"

3.2.3 Analyse du cas d'utilisation "Ajouter nouvel utilisateur"

- **Description textuelle du cas d'utilisation "Ajouter nouvel utilisateur" :**

Le tableau 3.3 représente la description textuelle du cas d'utilisation "Ajouter nouvel utilisateur".

Titre	Ajouter nouvel utilisateur
But	Permettre à l'administrateur d'ajouter un utilisateur
Acteurs	Administrateur
Précondition	L'administrateur doit s'authentifier et la connexion doit être établie. L'administrateur doit posséder les droits nécessaires.
Postcondition	Utilisateur ajouté.
Scénario nominal	N.1-L'administrateur accède à la rubrique Gestion des utilisateurs et appuie sur le bouton ajouter. N.2-Le système affiche une page sous forme d'un formulaire à remplir N.3- L'administrateur remplit le formulaire et clique sur le bouton ajouter. N.4-Le système sauvegarde l'utilisateur ajouté dans la base de données et faire la redirection vers la page de liste des utilisateurs.
Scénario alternatif	A.1 Données vides ou erronées A.1.1 Le système détecte qu'il existe des champs vides ou invalides ou utilisateur déjà existant. A.1.2 Le système affiche un message d'erreur. A.1.3 Retour à l'action N.2 du scénario nominal.

Tableau 3.3: Description textuelle du cas d'utilisation "Ajouter nouvel utilisateur"

- **Diagramme de séquence système du cas d'utilisation "Ajouter nouvel utilisateur" :**

La figure 3.4 représente le diagramme de séquence système du cas d'utilisation "Ajouter nouvel utilisateur".

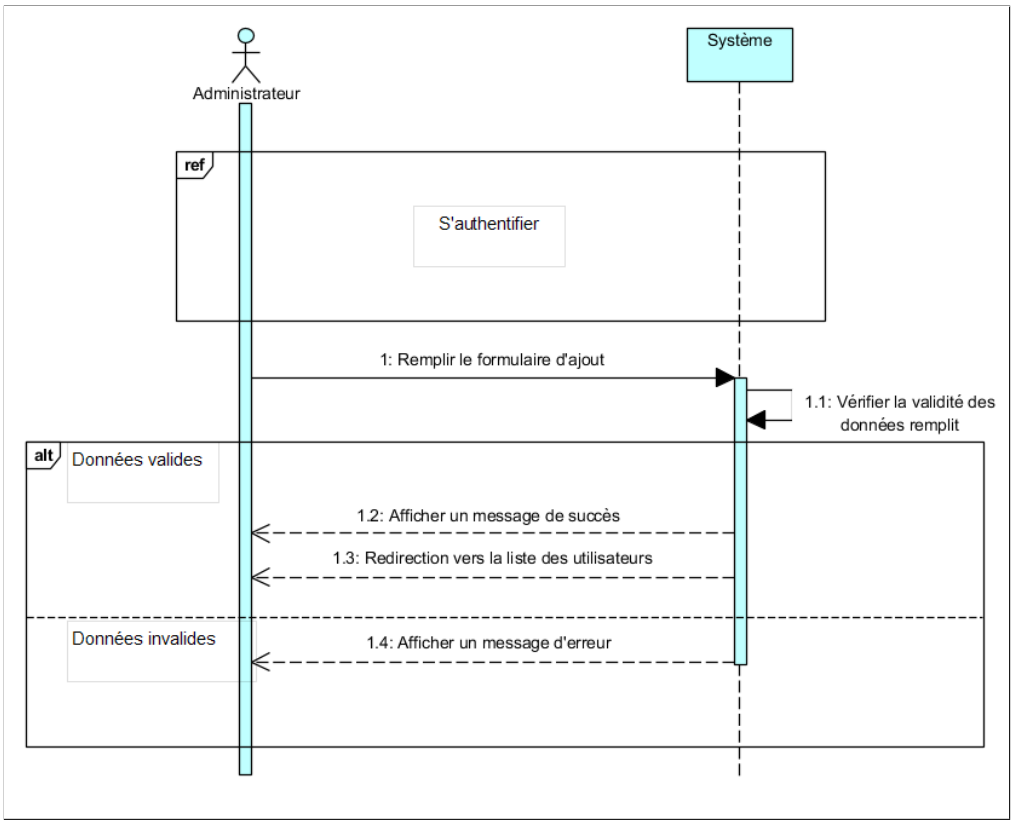


Figure 3.4: Diagramme de séquence système du cas d'utilisation "Ajouter nouvel utilisateur"

3.3 Conception

Au cours de cette section, on va présenter le diagramme de composants de sprint ainsi que les diagrammes de séquences objets de certains cas d'utilisation.

3.3.1 Diagramme de composants

Dans la figure 3.5, nous allons expliquer les interactions entre les différentes composants de notre application à savoir le model, Repository, Service, les contrôleurs.
Ces services seront par la suite exposés comme des Web services et vont être consommés par la partie Front de notre application. Nous allons prendre la partie d'authentification comme un exemple.

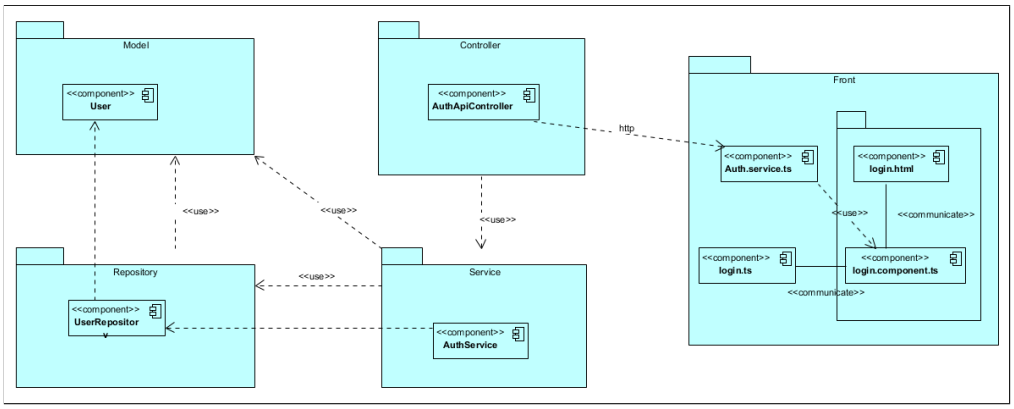


Figure 3.5: Diagramme de composants du Sprint 1

3.3.2 Diagramme de séquences objet

Le diagramme de séquence objet met en évidence l'interaction et les messages transmis entre l'utilisateur et les différents objets du système de la base de données à la partie BackEnd à la partie frontale et vice-versa.

- **Diagramme de séquence objet du cas d'utilisation "S'authentifier" :**

La figure 3.6 présente le diagramme de séquence objet du cas d'utilisation "s'authentifier". Ce diagramme montre le processus du login avec la vérification du mot de passe et la génération du token avec JWT (Json Web Token) pour l'utilisateur et enfin la redirection vers la page d'accueil de cet utilisateur connecté.

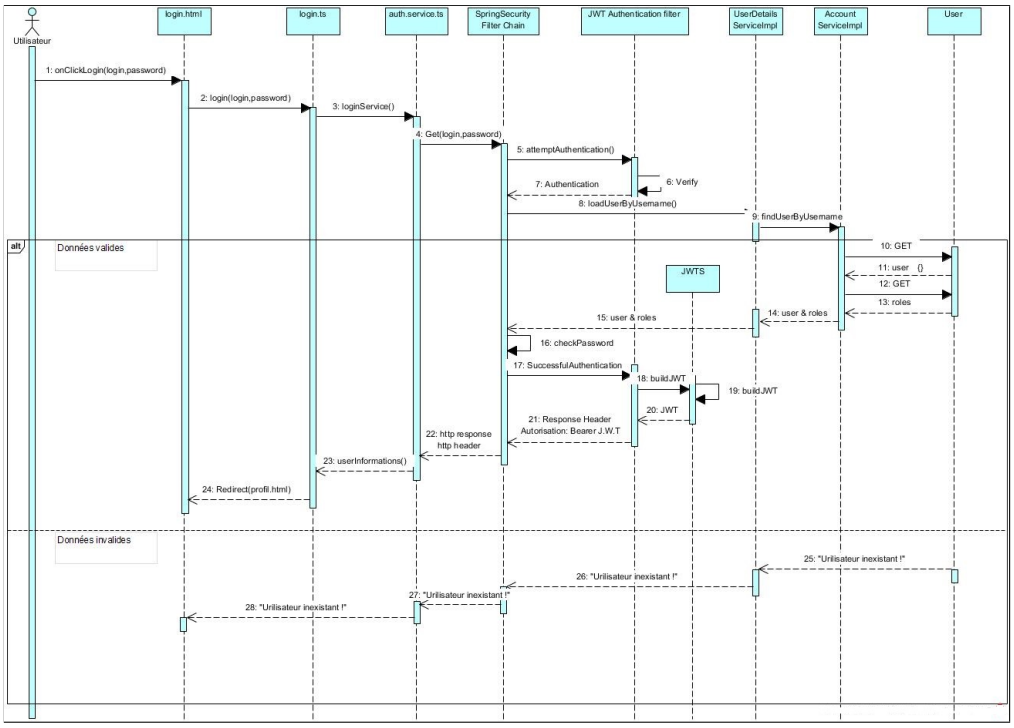


Figure 3.6: Diagramme de séquence objet du cas d'utilisation "S'authentifier"

- **Diagramme de séquence objet du cas d'utilisation "Ajouter nouvel utilisateur" :**

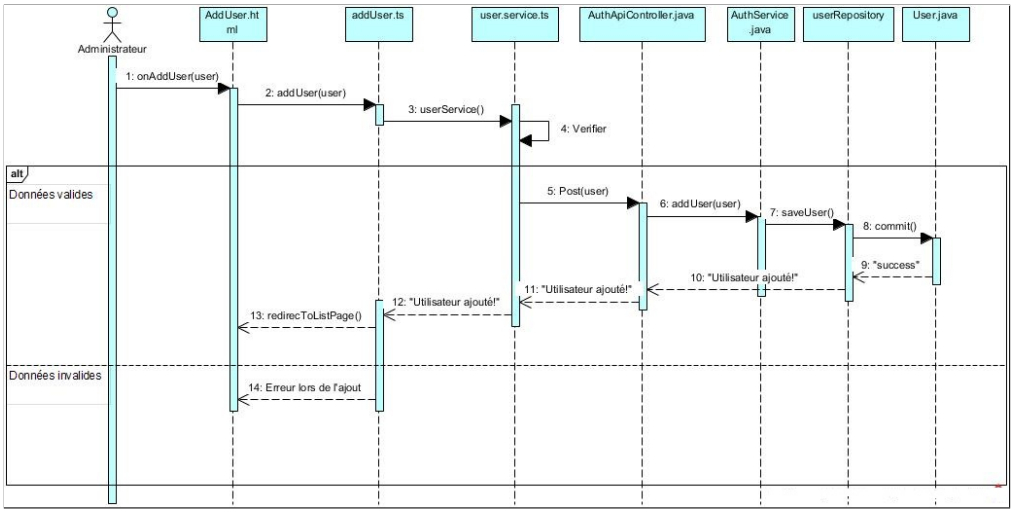


Figure 3.7: Diagramme de séquence objet du cas d'utilisation "Ajouter nouvel utilisateur"

3.4 Réalisation

Dans cette partie, nous présenterons les interfaces homme / machine afin de montrer la procédure d'utilisation de chaque interface reliée à ce sprint. Nous trouverons des interfaces liées à l'administrateur tel que l'ajout et la suppression des utilisateurs, et des interfaces liées à l'utilisateur et l'administrateur tel que l'interface d'authentification et le profile.

3.4.1 Interface d'authentification

Lorsque nous accédons à l'application pour la première fois, la page de connexion apparaît. Cette page fournit un mécanisme de sécurité qui empêche les utilisateurs non autorisés d'accéder au système. Pour aller au-delà de la page de connexion, le visiteur doit fournir un nom d'utilisateur / email et un mot de passe valides. Cette interface permet également à un utilisateur non encore enregistré d'être redirigé vers une autre interface pour demander l'accès.

Figure 3.8: Interface d'authentification

3.4.2 Interface de demande d'accès

Grâce à cette interface, un utilisateur non enregistré peut envoyer une notification par courrier électronique à tous les administrateurs.

Cette notification contient une demande d'autorisation d'accès accompagnée de l'adresse e-mail du destinataire requise pour la procédure d'autorisation. L'utilisateur peut être redirigé vers la page de connexion en utilisant le lien "Retour à la page Login".

Figure 3.9: Interface de demande d'accès aux administrateurs

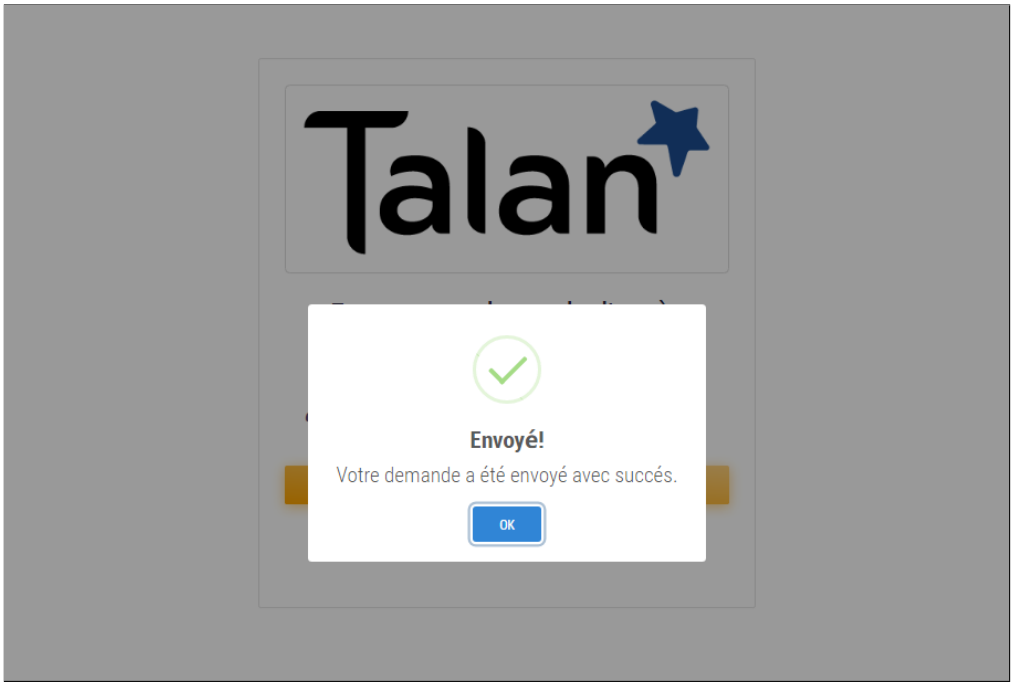


Figure 3.10: Pop-up de succès de la demande d'accès

La figure 3.11 montre un exemple de l'email qui sera envoyé aux administrateurs pour demander l'accès.

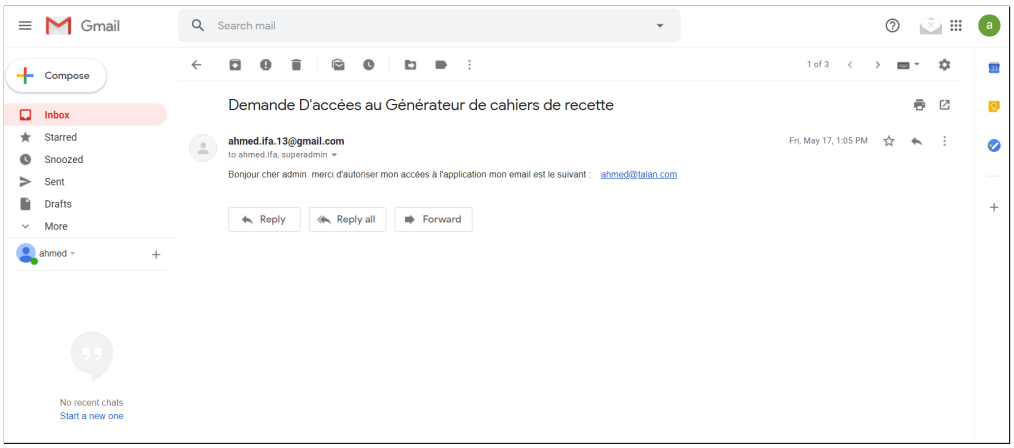


Figure 3.11: Le corps de l'email de demande d'accès

3.4.3 Interface de profile

Il s'agit d'une page contenant le profile personnel de l'utilisateur(nom, rôle, adresse, etc...). L'utilisateur peut changer son nom, son prénom ou son mot de passe à partir de cette interface.

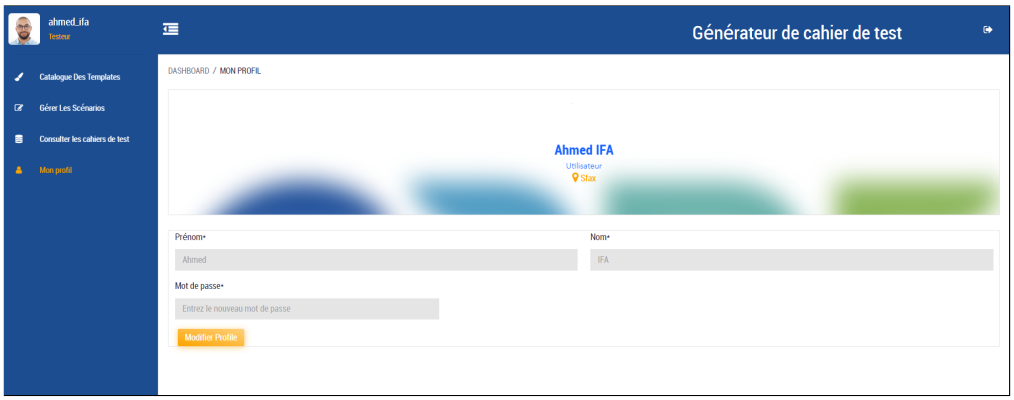


Figure 3.12: Interface de profile de l'utilisateur connecté

3.4.4 Les interfaces de l'administrateur

- **Gestion des utilisateurs :**

La figure 3.13 présente la liste des utilisateurs affichée dans un espace dédié à l'administrateur pour gérer ces derniers.

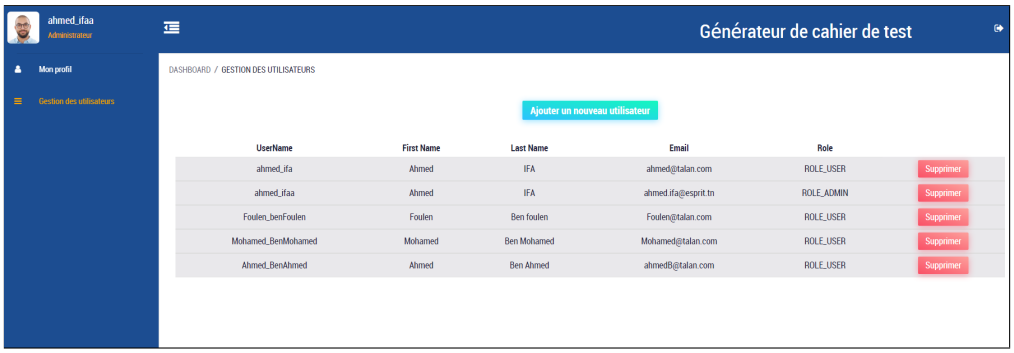


Figure 3.13: Interface de la liste des utilisateurs

Dans la figure ci-dessous on trouve le formulaire d'ajout d'un nouvel utilisateur en introduisant ses informations et puis en cliquant sur le bouton "Valider".

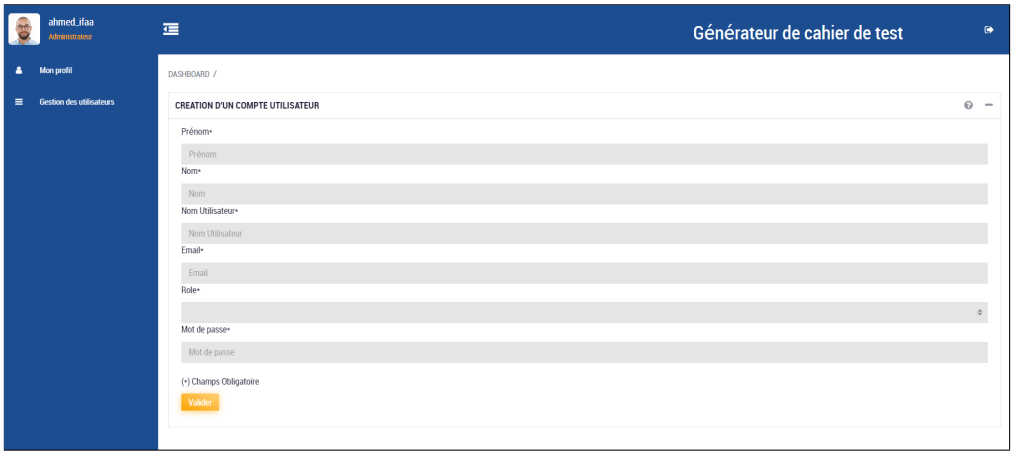


Figure 3.14: Interface du formulaire d'ajout d'un nouvel utilisateur

En cas ou l'administrateur oublie de remplir un ou plusieurs champs obligatoires, ou dans le cas ou il introduit des données déjà réservés à un autre utilisateur, un pop-up s'affiche pour l'informer.

La figure 3.15 montre le pop-up d'échec de l'opération d'ajout.

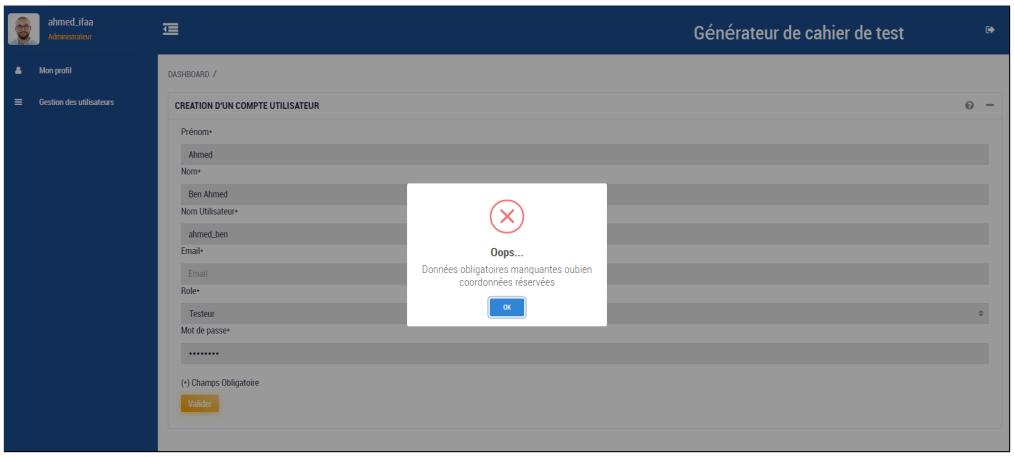


Figure 3.15: Pop-up d'échec d'opération d'ajout d'un utilisateur

La figure ci-dessous présente l'interface contenant le pop-up de succès de l'ajout d'un utilisateur.

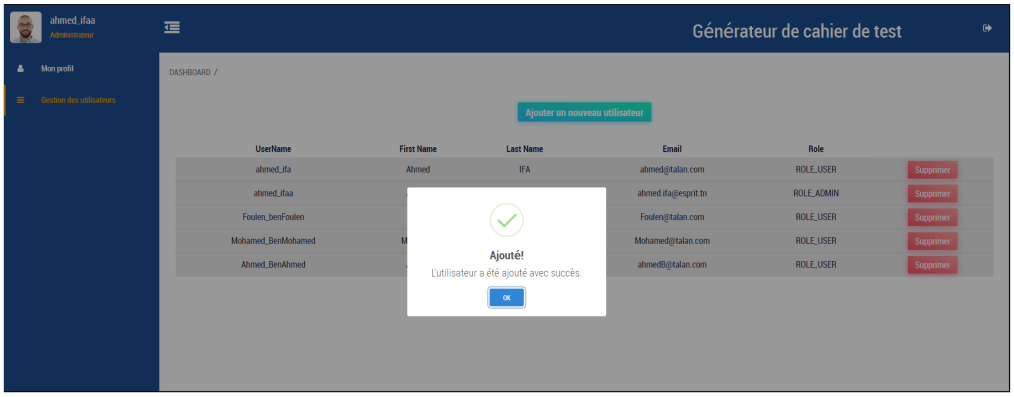


Figure 3.16: Pop-up de succès d'opération d'ajout d'un utilisateur

Pour supprimer un utilisateur, il suffit que l'administrateur clique sur le bouton "Supprimer" dans la liste des utilisateurs et un message de confirmation s'affiche pour approuver la suppression.

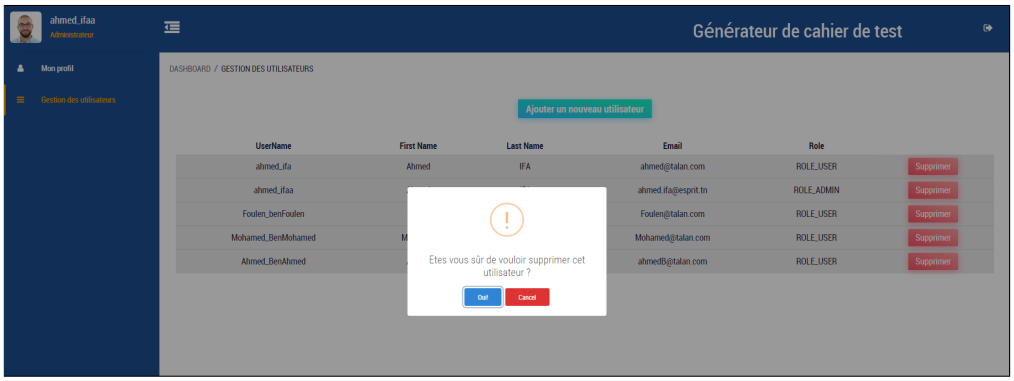


Figure 3.17: Pop-up de confirmation de suppression d'un utilisateur

Conclusion

Nous avons traité dans ce chapitre le Backlog du sprint de gestion des utilisateurs, son étude analytique et conceptuelle, nous avons détaillé ces parties au moyen de diagrammes statiques et dynamiques et nous avons terminé avec la partie réalisation où nous avons montré quelques captures des interfaces du livrable montrant le fonctionnement des cas d'utilisations traités.

Chapitre 4

Sprint 2 : Gestion de catalogue de template

Plan

1 **Backlog du Sprint** 42

2 **Analyse** 42

3 **Conception** 47

4 **Réalisation** 50

Introduction

Dans ce chapitre, et comme nous avons procédé au chapitre précédent, nous présenterons le sprint de la gestion de catalogue de template. Au début, nous allons préparer le Backlog. Ensuite, nous détaillerons la partie analyse avec les descriptions textuelles et les diagrammes de séquences analyse, puis la partie conception avec le diagramme de composants, les diagrammes de séquences objet et le diagrammes des classes participantes. enfin, nous terminerons avec la partie réalisation.

4.1 Backlog du Sprint

User story	Tâches	Estimation (en jours)
En tant qu'utilisateur je veux ajouter un template au catalogue pour enrichir ma base de référence.	Développement de l'interface d'ajout de modèle de cahier de recette	3
	Génération de la table dans la BDD	
	Développement du WS de l'ajout du template	
	Consommation du WS dans la partie Front	
En tant qu'utilisateur je veux modifier un template du catalogue pour mettre à jour ma base de référence	Développement de l'interface modification du template	3
	Développement du WS de modification du template	
	Consommation du WS dans la partie Front	
En tant qu'utilisateur je veux Supprimer un template du catalogue pour éviter l'encombrement des modèles inutiles.	Développement de l'interface de suppression du template	2
	Développement du WS de la suppression du template	
	Consommation du WS dans la partie Front	

Tableau 4.1: Backlog du Sprint 2

4.2 Analyse

Dans la figure 4.1, on trouve le diagramme de cas d'utilisation du deuxième sprint.

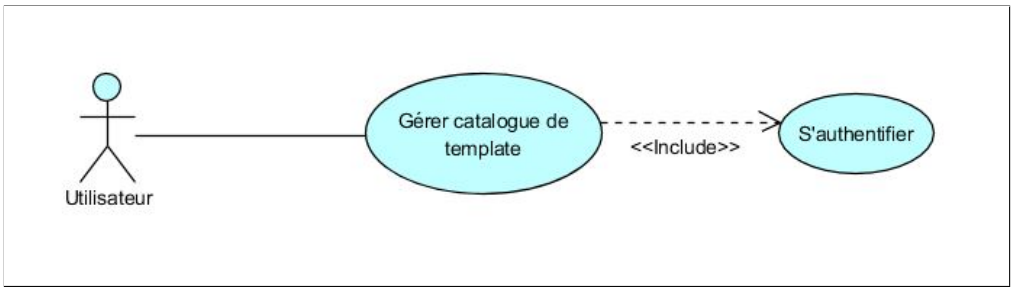


Figure 4.1: Diagramme de cas d'utilisation du Sprint 2

4.2.1 Diagramme de cas d'utilisation raffiné

La figure 4.2 montre le raffinement des cas d'utilisation du deuxième sprint dont le choix, l'ajout, la modification et la suppression d'un catalogue de template.

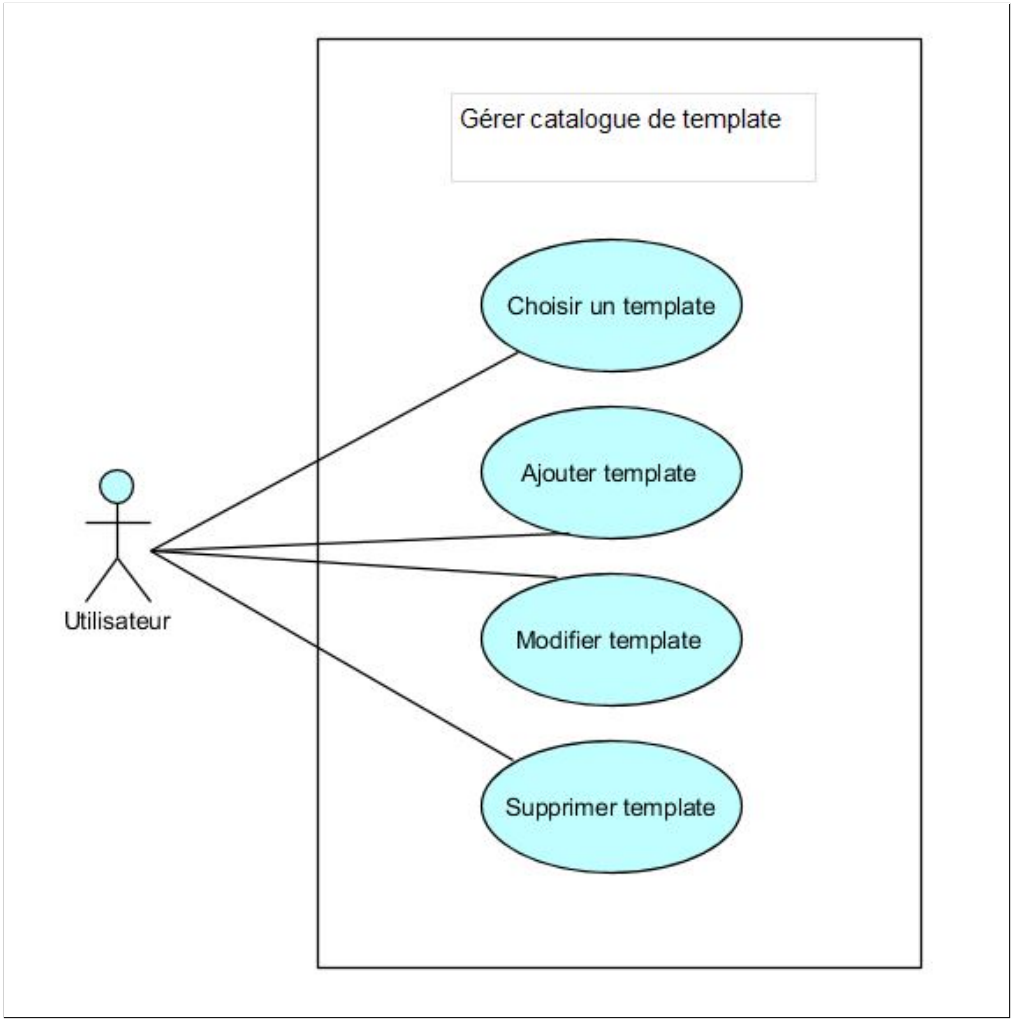


Figure 4.2: Diagramme de cas d'utilisation raffiné du Sprint 2

4.2.2 Analyse du cas d'utilisation "Ajouter template"

Dans cette section nous allons faire une étude d'analyse du cas d'utilisation "Ajouter Template".

- **Description textuelle du cas d'utilisation "Ajouter template" :**

Le tableau 4.2 présente la description textuelle du cas d'utilisation "Ajouter template" en précisant les acteurs, les conditions et les scénarios nominal et alternatif.

Titre	Ajouter template
But	Permettre à l'utilisateur d'ajouter un template
Acteurs	Utilisateur
Précondition	Utilisateur authentifié
Postcondition	Template ajouté et affiché dans la liste.
Scénario nominal	N.1. Le système affiche la liste des templates déjà existants. N.2. L'utilisateur saisit le nom du template dans la rubrique d'ajout. N.3. L'utilisateur ajoute un ou plusieurs modèles du template en cliquant sur le bouton de l'ajout de modèle. N.4. L'utilisateur clique sur le bouton ajouter pour ajouter le template. N.5. Le système affiche le template ajouté dans la liste.
Scénario alternatif	A.1. Données vides A.1.1 Le système affiche un message d'erreur. A.1.2 Retour à l'action N.1 du scénario nominal.

Tableau 4.2: Description textuelle du cas d'utilisation "Ajouter template"

- **Diagramme de séquence système du cas d'utilisation "Ajouter temlate" :**

Le diagramme de séquence système du cas d'utilisation "Ajouter template" est présenté dans la figure 4.3.

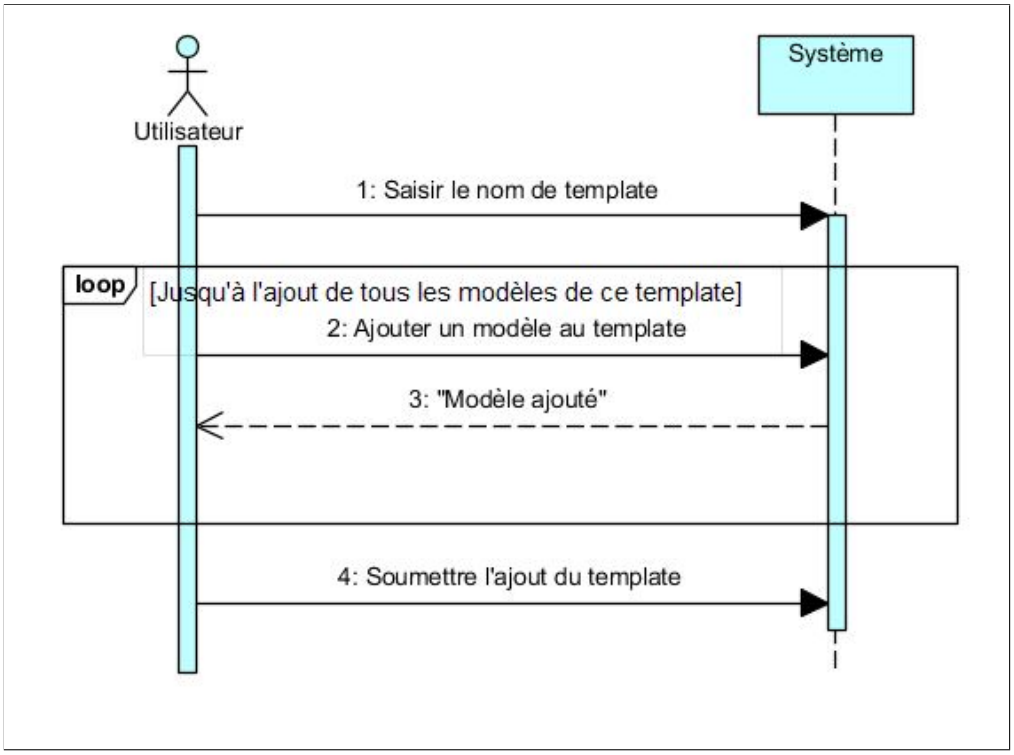


Figure 4.3: Diagramme de séquence système du cas d'utilisation "Ajouter Template"

4.2.3 Analyse du cas d'utilisation "Modifier template"

- **Description textuelle du cas d'utilisation "Modifier Template" :**

La description textuelle du cas d'utilisation "Modifier template" est présenté dans le tableau 4.3.

Titre	Modifier template
But	Permettre à l'utilisateur de modifier un template
Acteurs	Utilisateur
Précondition	Utilisateur authentifié et la liste des templates est affichée.
Postcondition	Template modifié et affiché après modification
Scénario nominal	N.1. L'utilisateur clique sur le bouton modifier N.2. Le système affiche les templates en mode modification N.3. L'utilisateur choisit et modifie le template à modifier. N.4. L'utilisateur clique sur le bouton synchroniser. N.5. Le système affiche le template modifié dans la liste des templates.
Scénario alternatif	A.1 Données vides A.1.1 Le système affiche un message d'erreur. A.1.2 Retour à l'action N.2 du scénario nominal.

Tableau 4.3: Description textuelle du cas d'utilisation "Modifier template"

- **Diagramme de séquence système du cas d'utilisation "Modifier template" :**

Les interactions entre l'utilisateur et le système sont présenté dans le diagramme de séquence système du cas d'utilisation "Modifier template" ci-dessous.

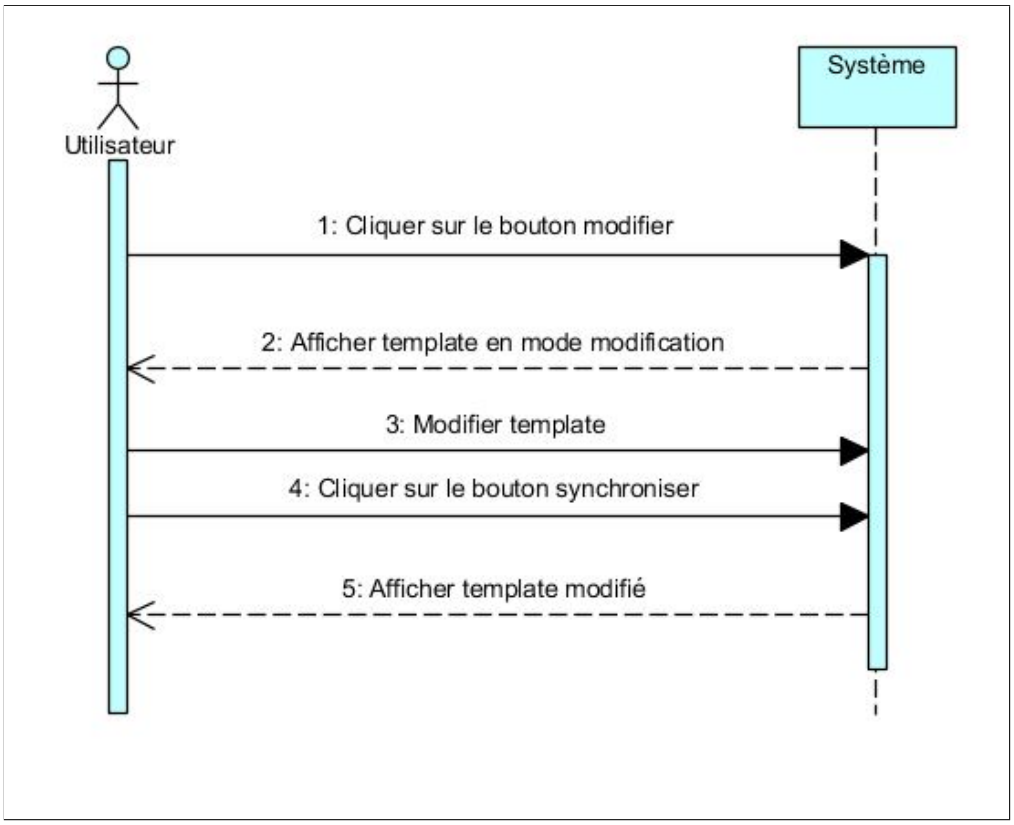


Figure 4.4: Diagramme de séquence système du cas d'utilisation "Modifier Template"

4.3 Conception

Dans cette section, nous allons présenter le diagramme de composant du deuxième sprint ainsi que les diagrammes de séquence objet de certains cas d'utilisation.

4.3.1 Diagramme de composants

Dans la figure 4.5, nous allons expliquer les interactions entre les différentes composants de notre application à savoir le model, Repository, Service, les contrôleurs.
Ces services seront par la suite exposés comme des Web services et vont être consommés par la partieFront de notre application.

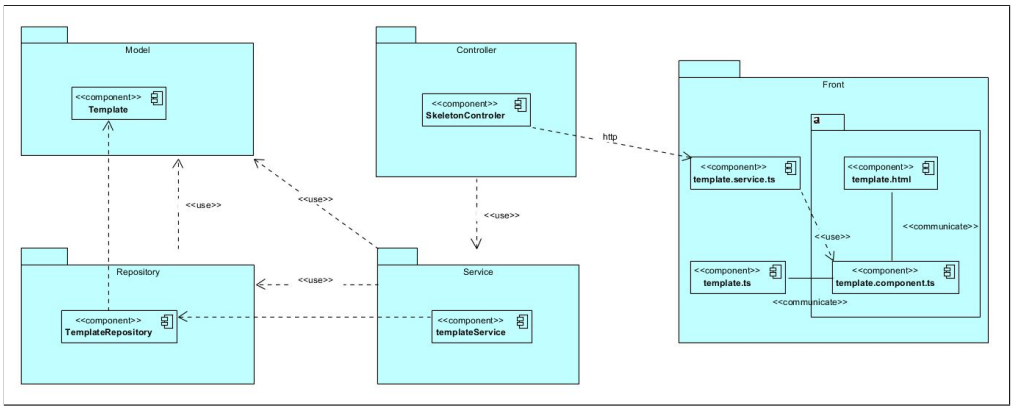


Figure 4.5: Diagramme de composants de Sprint 2

4.3.2 Diagrammes de séquences objet

- **Diagramme de séquence objet du cas d'utilisation "Ajouter Template" :**

La figure 4.6 présente le diagramme de séquence objet du cas d'utilisation "Ajouter Template".

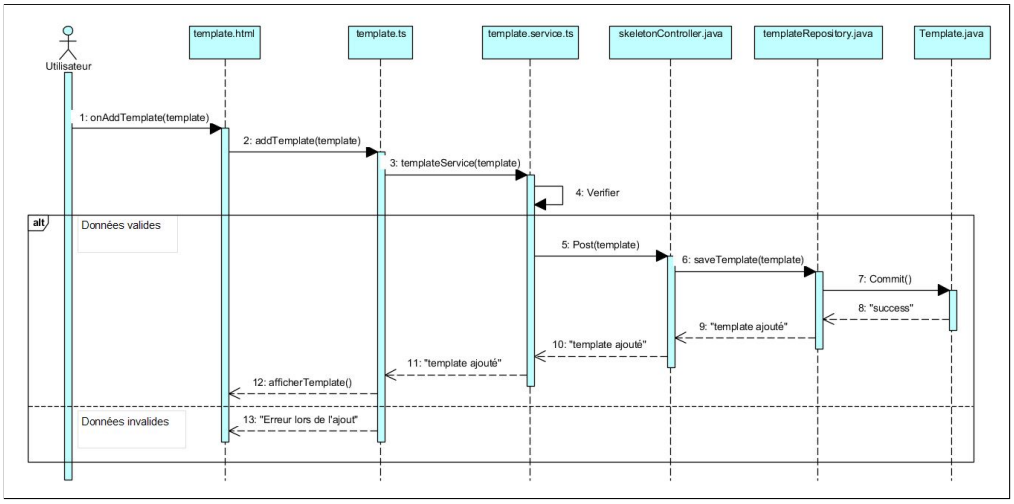


Figure 4.6: Diagramme de séquence objet du cas d'utilisation "Ajouter template"

- **Diagramme de séquence objet du cas d'utilisation "Modifier template" :**

Dans la figure 4.7 nous trouverons le diagramme de séquence objet du cas d'utilisation "Modifier Template".

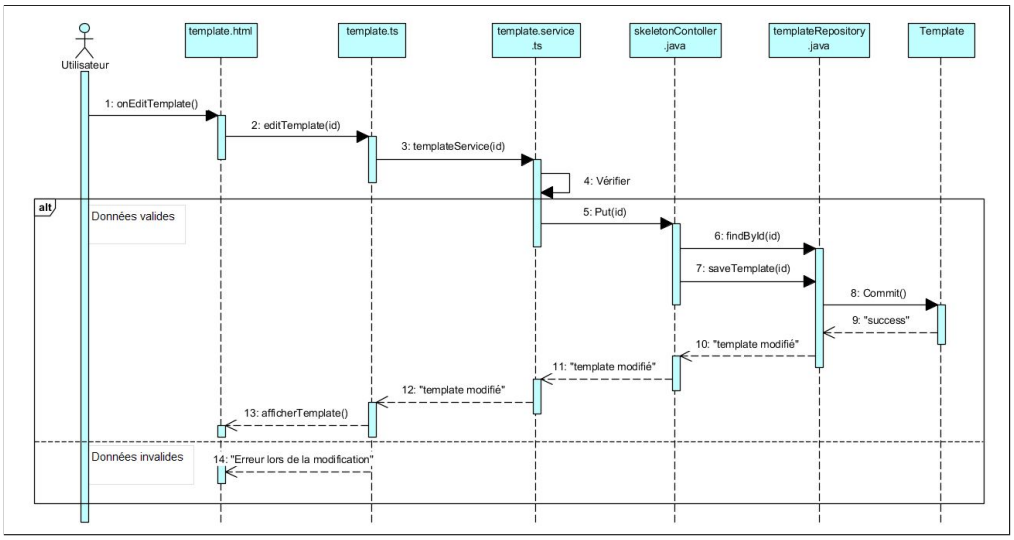


Figure 4.7: Diagramme de séquence objet du cas d'utilisation "Ajouter template"

4.3.3 Diagramme de classes participantes

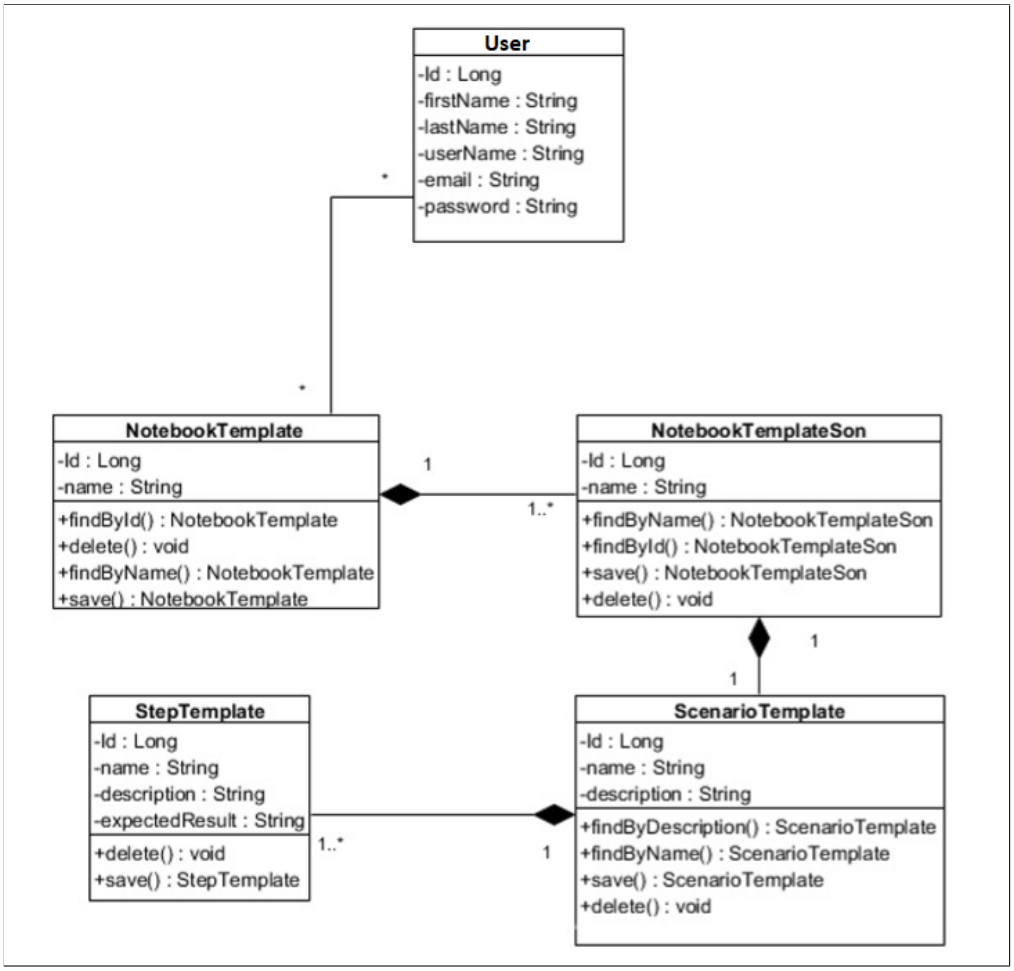


Figure 4.8: Diagramme de classes participantes du Sprint 2

4.4 Réalisation

- **Interfaces "Catalogue de templates"**

Cette interface permettra aux squelettes de ces templates d'être créés de manière simple et flexible au moyen de la carte nommée "Ajouter un nouveau template", comme illustré à la figure 4.9. Lors de la saisie de la liste des nom des scénarios, en cas d'erreur de frappe, le testeur peut revenir et en appuyant deux fois sur le nom, il pourra rectifier sa faute et modifier.

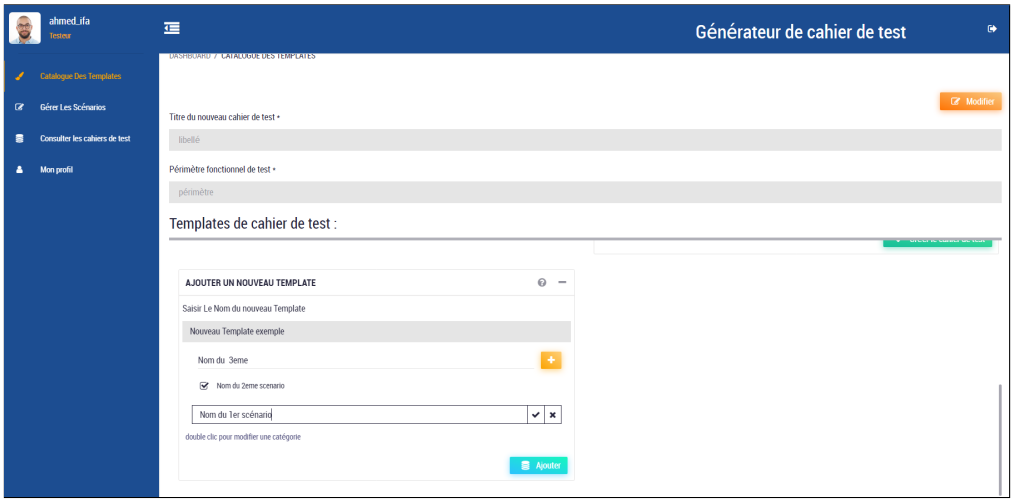


Figure 4.9: Interface d'un exemple d'ajout d'un template de cahier de recette

Après sa création, un modèle de cahier de recettes sera écrit sur une carte bien lisible qui pourra être réduite à l'aide d'un bouton situé en haut à droite pour conserver l'affichage de son nom, car il peut être étendu pour afficher la liste des noms de scénarios.

Notre interface admet comme prévu, en haut à droite un bouton nommé "Modifier" permettant de basculer entre deux modes :

— Un mode de consultation, activé par défaut, qui permet dans une prochaine étape de garantir la fonctionnalité de la création d'un carnet de recette à partir d'un modèle.

— Un mode de modification qui vous permet d'ajouter un modèle de scénario au squelette d'un modèle ou de le supprimer complètement de notre base de données de référence afin de garantir sa validité et d'éviter l'encombrement du catalogue ou de le modifier. ajouter un autre nom de modèle de scénario à la liste.

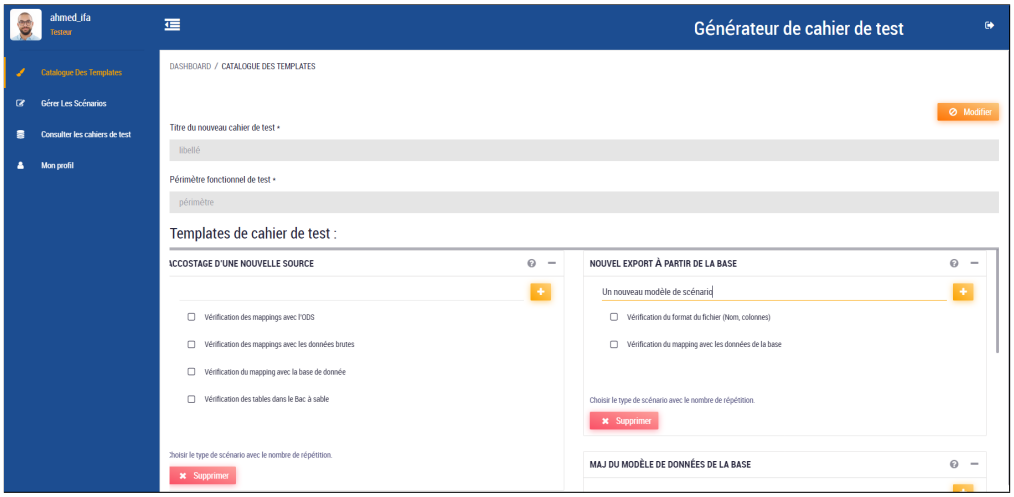


Figure 4.10: Interface du catalogue de templates en mode de modification

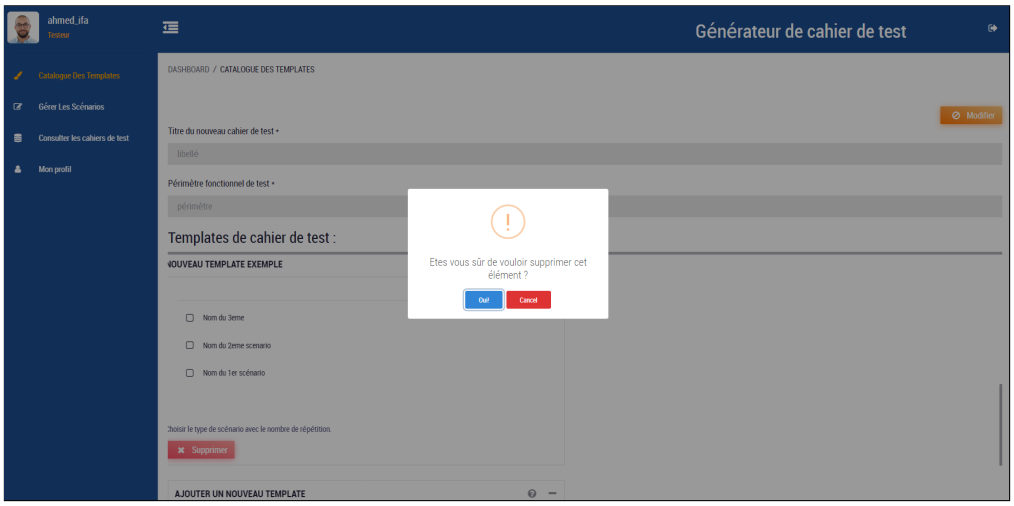


Figure 4.11: Pop-up de confirmation de la suppression d'un template

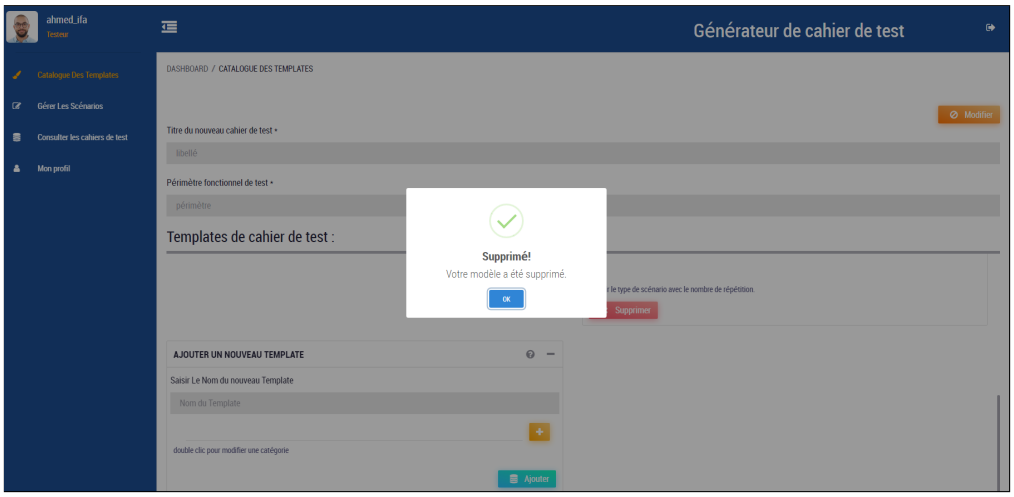


Figure 4.12: Message d'information de la suppression d'un template

Conclusion

Nous avons procédé dans ce chapitre de la même manière que le chapitre précédent, nous avons commencé par le Backlog du sprint de la gestion de catalogue de template, son étude analytique et conceptuelle, nous avons détaillé ces parties à travers des diagrammes statiques et dynamiques et nous avons terminé par la partie réalisation où nous avons montré quelques interfaces du livrable qui montrent le processus de la gestion de catalogue de template.

Chapitre 5

Sprint 3 : Gestion de scénario

Plan

1 **Backlog Sprint 3** 54

2 **Analyse** 54

3 **Conception** 59

4 **Réalisation** 61

Introduction

Durant ce chapitre, nous allons présenter l'étude du troisième sprint intitulé "Gestion de scénario". Dans un premier temps, nous allons préparer le Backlog de ce sprint. Ensuite, présenter la partie d'analyse et la partie conception. Enfin, nous allons terminer par la partie réalisation avec des interfaces du livrable du sprint.

5.1 Backlog Sprint 3

User story	Tâches	Estimation (en jours)
En tant qu'utilisateur je veux ajouter un scénario de test pour enrichir ma base de référence.	Développement de l'interface d'ajout de scénario de test	3
	Génération de la table dans la BDD	
	Développement du WS de l'ajout du template	
	Consommation du WS dans la partie Front	
En tant qu'utilisateur je veux modifier un scénario de test pour mettre à jour ma base de référence.	Développement de l'interface modification du scénario	3
	Développement du WS de modification du template	
	Consommation du WS dans la partie Front	
En tant qu'utilisateur je veux Supprimer un scénario de test.	Développement de l'interface de suppression du scénario	2
	Développement du WS de la suppression du template	
	Consommation du WS dans la partie Front	

Tableau 5.1: Backlog du Sprint 3

5.2 Analyse

Aux cours de cette section, nous allons faire l'étude analytique de ce sprint en présentant le diagramme de cas d'utilisation puis son raffinement, les description textuelles des cas d'utilisations de ce sprint. Enfin, nous allons présenter les diagrammes de séquence système de ces cas.
La figure 5.1 présente le diagramme de cas d'utilisation du troisième sprint.

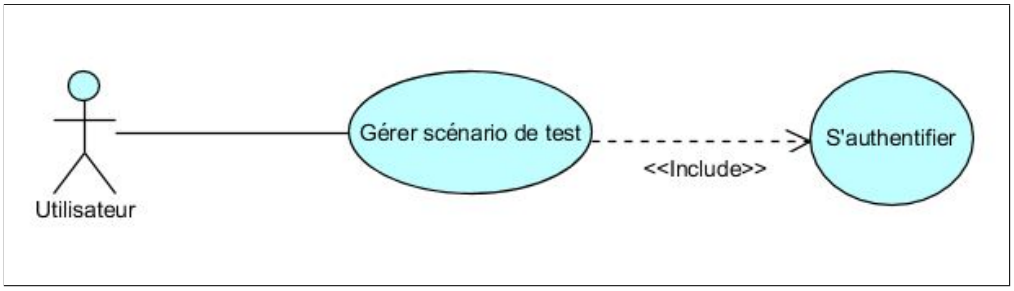


Figure 5.1: Diagramme de cas d'utilisation du Sprint 3

5.2.1 Diagramme de cas d'utilisation raffiné

La figure 5.2 présente le raffinement du diagramme de cas d'utilisation du troisième sprint en mettant en évidence les "use cases" de la gestion des scénarios de tests.

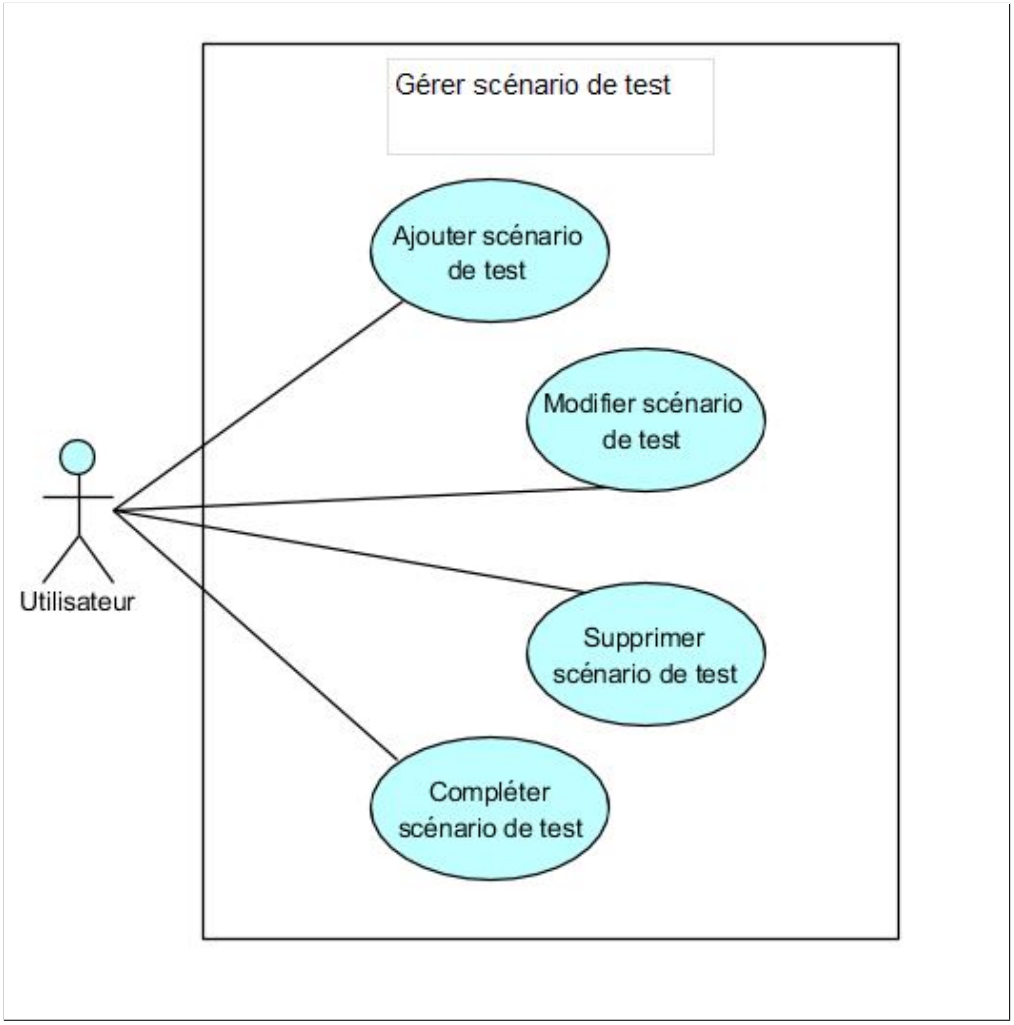


Figure 5.2: Raffinement du diagramme de cas d'utilisation du Sprint 3

5.2.2 Analyse du cas d'utilisation "Ajouter Scénario"

Dans cette section, nous allons faire l'étude analytique du cas d'utilisation "Ajouter Scénario" en présentant sa description textuelle et le diagramme de séquence système.

- **Description textuelle du cas d'utilisation "Ajouter Scénario" :**

Le tableau ci-dessous présente la description textuelle du cas d'utilisation "Ajouter Scénario".

Titre	Ajouter scénario
But	Permettre à l'utilisateur d'ajouter un scénario
Acteurs	Utilisateur
Précondition	Utilisateur authentifié
Postcondition	Le scénario est ajouté
Scénario nominal	N.1. L'utilisateur clique sur le bouton ajouter scénario. N.2. Le système affiche le formulaire d'ajout de scénario. N.3. L'utilisateur remplit le formulaire. N.4. L'utilisateur confirme l'ajout en cliquant sur le bouton ajouter.
Scénario alternatif	A.1. Données vides A.1.1 Le système affiche un message d'erreur. A.1.2 Retour à l'action N.2 du scénario nominal.

Tableau 5.2: Description textuelle du cas d'utilisation "Ajouter Scénario"

- **Diagramme de séquence système du cas d'utilisation "Ajouter Scénario" :**

Le diagramme de séquence système du cas d'utilisation "Ajouter scénario" est présenté dans la figure 5.3.

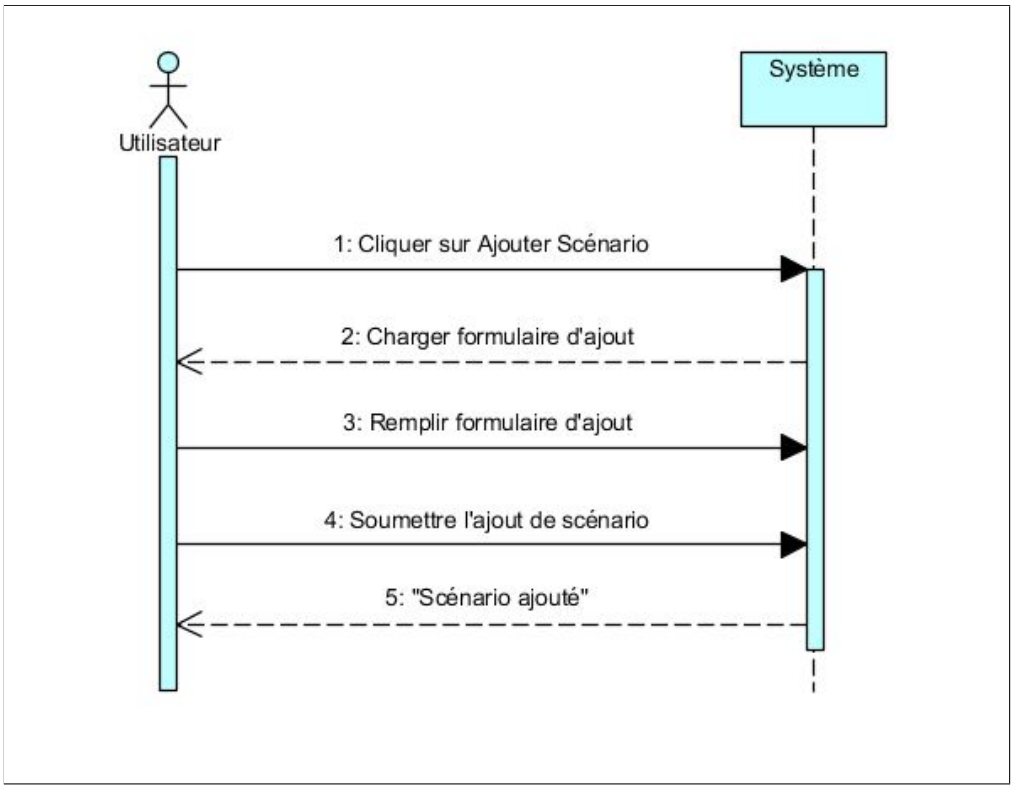


Figure 5.3: Diagramme de séquence système du cas d'utilisation "Ajouter Scénario"

5.2.3 Analyse du cas d'utilisation "Modifier Scénario"

Dans cette section, nous allons faire l'étude analytique du cas d'utilisation "Modifier Scénario" en présentant sa description textuelle et le diagramme de séquence système.

- **Description textuelle du cas d'utilisation "Modifier Scénario" :**

Dans le tableau 5.3, nous présentons la description textuelle du cas d'utilisation "Modifier Scénario".

Titre	Modifier scénario
But	Permettre à l'utilisateur de modifier un scénario.
Acteurs	Utilisateur
Pré-condition	Utilisateur authentifié.
Post-condition	Le scénario est modifié.
Scénario nominal	N.1. L'utilisateur choisit le scénario à modifier. N.1. Le système charge les informations du scénario sélectionné. N.3. L'utilisateur modifie le scénario. N.4. L'utilisateur confirme la modification en cliquant sur le bouton modifier. N.5. Le scénario est modifié.
Scénario alternatif	A.1 Données vides A.1.1 Le système affiche un message d'erreur. A.1.2 Retour à l'action N.2 du scénario nominal.

Tableau 5.3: Description textuelle du cas d'utilisation "Modifier Scénario"

- **Diagramme de séquence système du cas d'utilisation "Modifier Scénario" :**

La figure 5.4 présente le diagramme de séquence système du cas d'utilisation "Modifier Scénario".

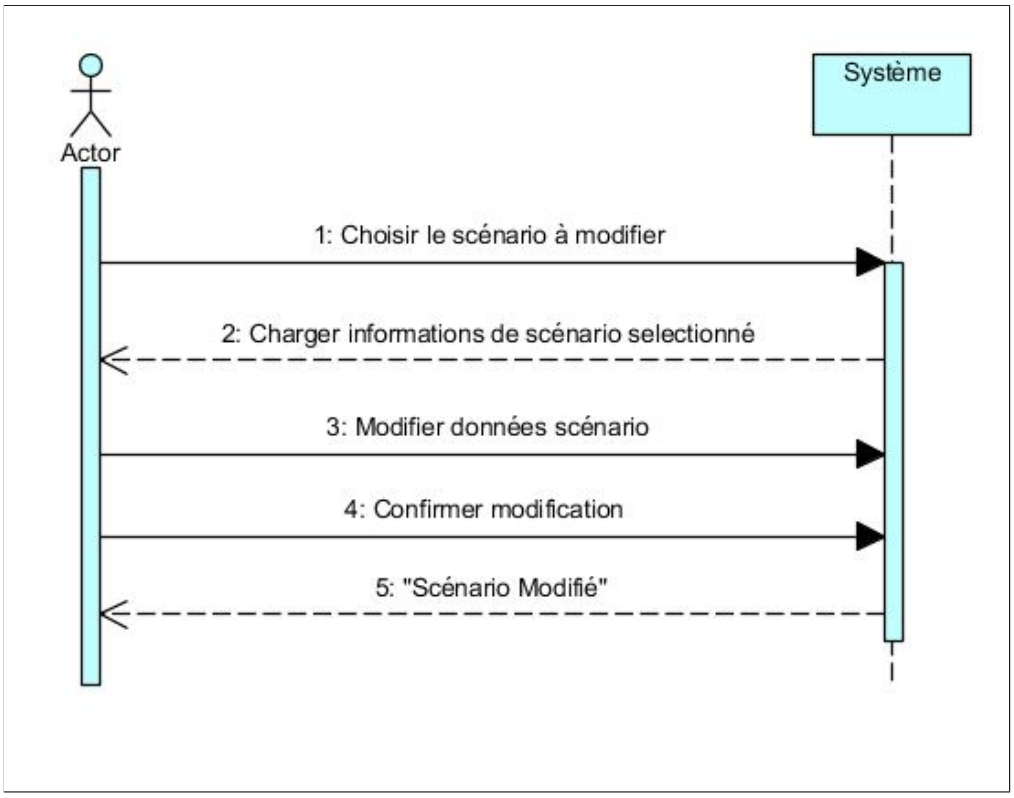


Figure 5.4: Diagramme de séquence système du cas d'utilisation "Modifier Scénario"

5.3 Conception

5.3.1 Diagramme de composants

Dans la figure 5.5, nous allons expliquer les interactions entre les différentes composants de notre application à savoir le model, Repository, Service, les contrôleurs.

Ces services seront par la suite exposés comme des Web services et vont être consommés par la partie Front de notre application.

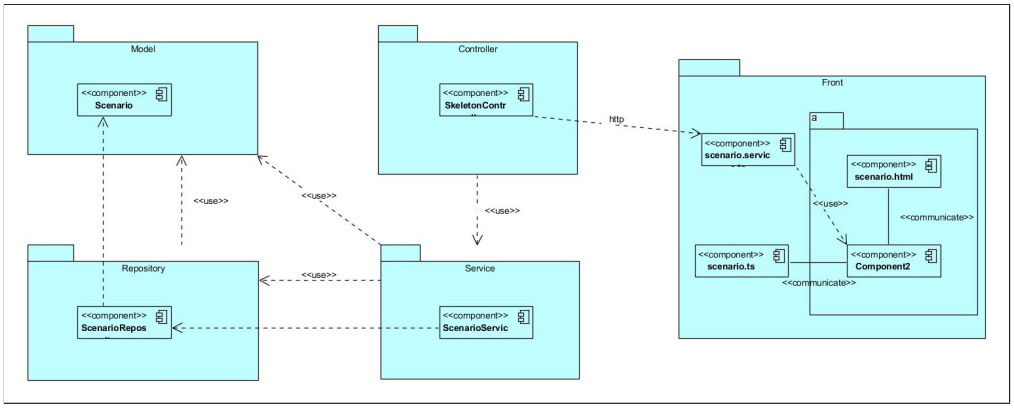


Figure 5.5: Diagramme de composants du sprint 3

5.3.2 Diagrammes de séquences objet

Au cours de cette section, nous allons présenter quelques diagrammes de séquence objet liés aux cas d'utilisation de ce sprint.

- **Diagramme de séquence objet du cas d'utilisation "Ajouter Scénario" :**

Ceci est le diagramme de séquence objet du cas d'utilisation "Ajouter Scénario" :

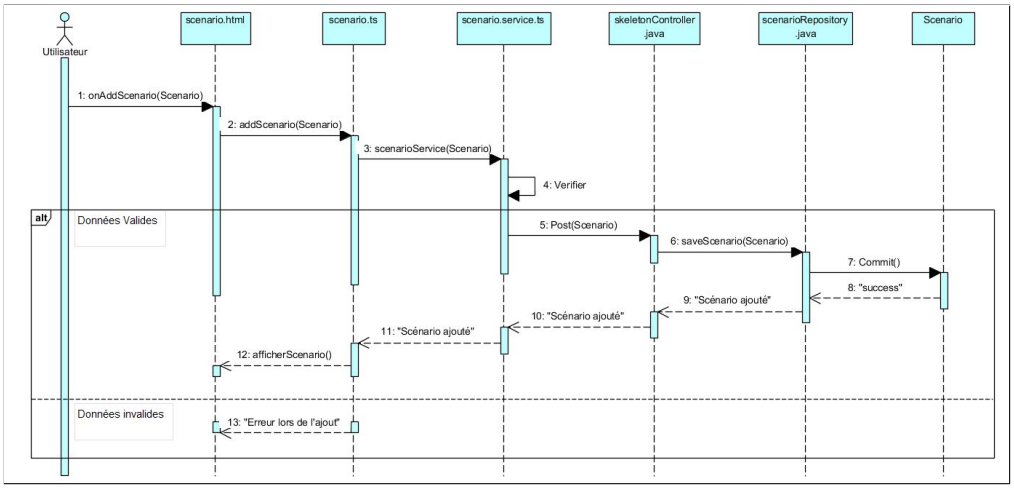


Figure 5.6: Diagramme de séquence objet du cas d'utilisation "Ajouter Scénario"

- **Diagramme de séquence objet du cas d'utilisation "Modifier Scénario" :**

La figure ci-dessous présente le diagramme de séquence objet du cas d'utilisation "Modifier Scénario".

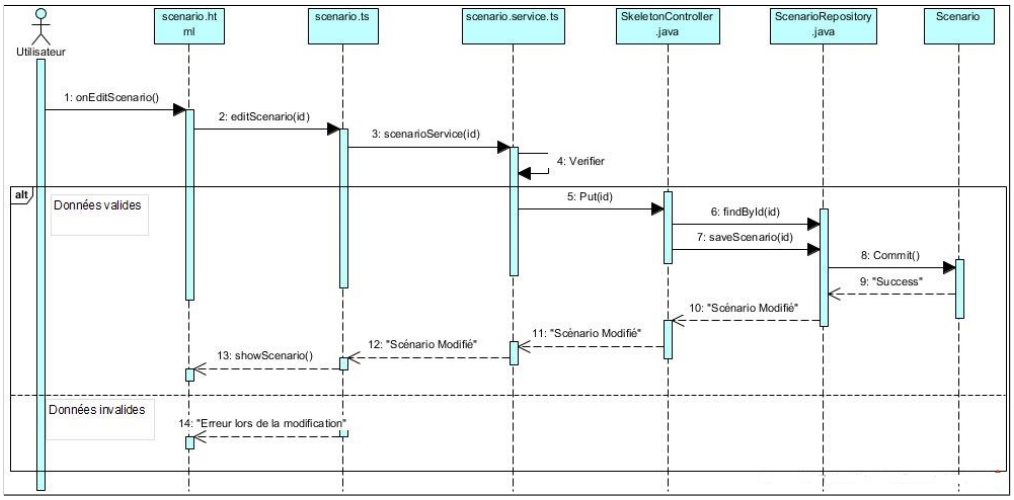


Figure 5.7: Diagramme de séquence objet du cas d'utilisation "Modifier Scénario"

5.3.3 Diagramme de classes participantes

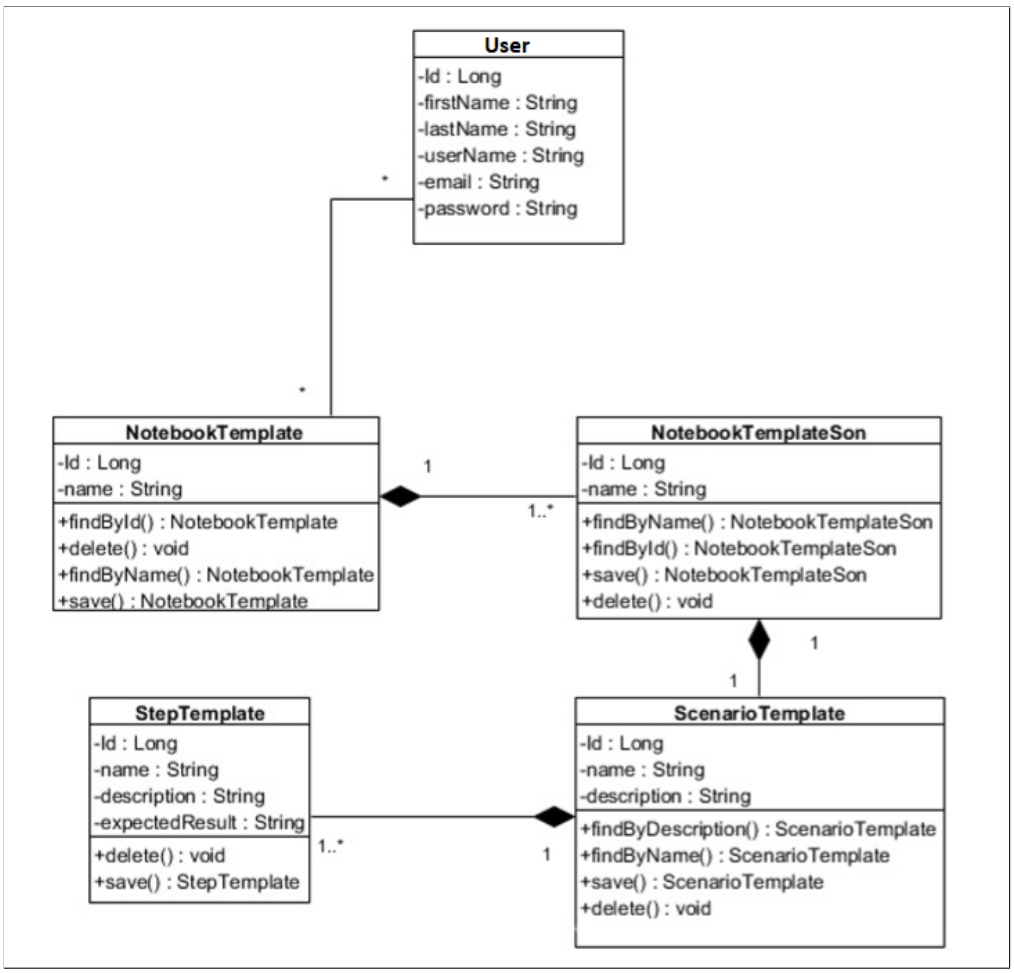


Figure 5.8: Diagramme de classes participantes du Sprint 3

5.4 Réalisation

Cette interface permet de définir le contenu d'un modèle de scénario après son inclusion dans un squelette de modèle de cahier de recette.

Inscrit dans une carte intitulée "Définir le contenu d'une brique de scénario", un formulaire définissant le contenu est précédé d'une liste déroulante actualisée chaque ajout pour ne contenir que les éléments non encore définis.

L'utilisateur renseigne un scénario et définit le modèle des étapes de ce stéréotype de scénario selon une syntaxe unifiée qui sera incluse dans la documentation de l'application.

Citons comme exemples :

$perimetre$: afin de marquer dans le modèle l'emplacement où le périmètre fonctionnel de la

recette sera remplacé lors de la génération d'un cahier.

$nomtable$: afin de montrer l'existence d'un nom de table.

$rangscenario$: pour montrer qu'une partie du contenu du modèle doit être remplacé par la numérotation d'un scénario dans un cahier de recette.

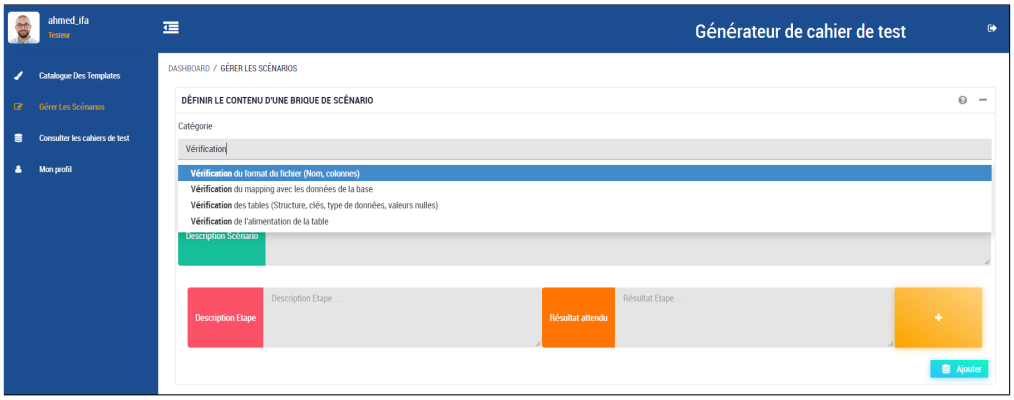


Figure 5.9: Interface du choix de scénario à définir

Dans la figure 5.10 nous présentons un exemple d'ajout d'un scénario et ses d'étapes.

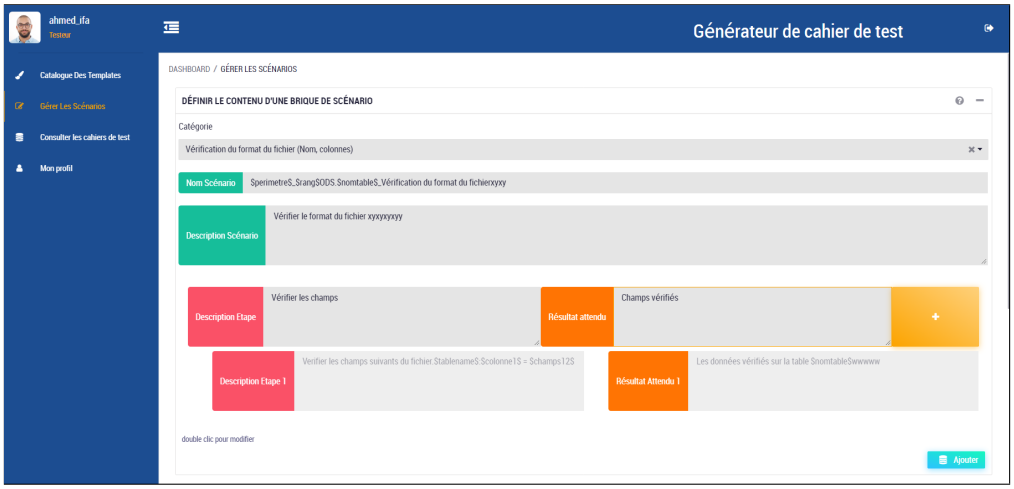


Figure 5.10: Interface d'ajout du contenu d'un scénario

Dans la Figure 5.11, nous présentons le chargement d'un modèle de scénario défini précédemment à l'aide de la carte "Modifier brique de scénario" afin d'apporter des modifications à son contenu. La persistance des modifications se fera par un simple appui sur le bouton "Valider".

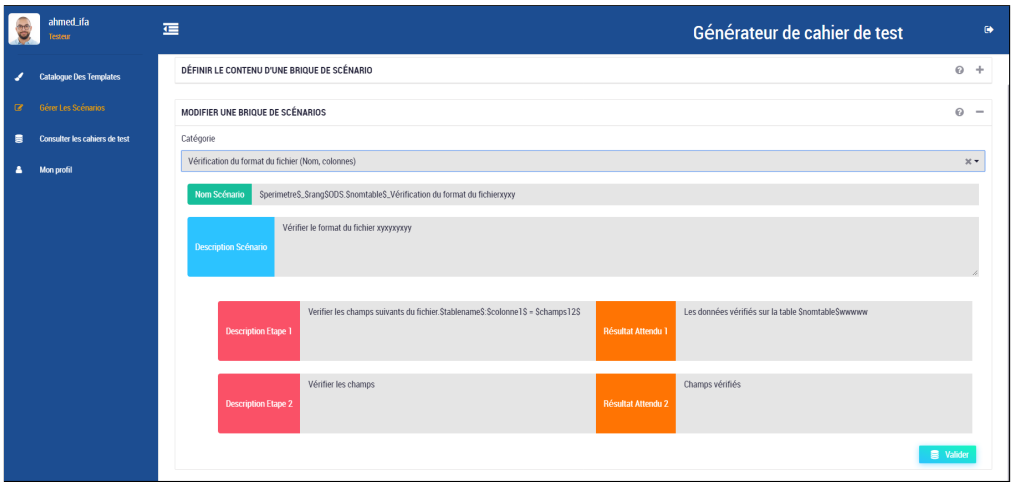


Figure 5.11: Interface de modification d'un scénario

Conclusion

Nous avons présenté dans ce chapitre, l'étude du sprint de la gestion des scénarios de test. Puis nous avons fini par la partie réalisation qui démontre les étapes à suivre pour gérer un scénario de test à travers des interfaces du livrable de ce sprint.

Chapitre 6

Sprint 4 : Gestion et Exportation de cahier de recette

Plan

1 **Backlog Sprint 4** **65**

2 **Analyse** **65**

3 **Conception** **70**

4 **Réalisation** **72**

Introduction

Durant ce chapitre, nous allons présenter l'étude du dernier sprint. Dans un premier temps, nous allons préparer le Backlog de ce sprint. Ensuite, présenter la partie d'analyse et la partie conception. Enfin, nous allons terminer par la partie réalisation avec des interfaces du livrable du sprint.

6.1 Backlog Sprint 4

User story	Tâches	Estimation (en jours)
En tant qu'utilisateur je veux créer un cahier de recette pour le pouvoir compléter.	Développement de l'interface d'ajout de cahier de recette	3
	Génération de la table dans la BDD	
	Développement du WS de l'ajout du cahier	
	Consommation du WS dans la partie Front	
En tant qu'utilisateur je veux modifier un cahier de recette.	Développement de l'interface de modification du cahier de recette.	3
	Développement du WS de modification du cahier	
	Consommation du WS dans la partie Front	
En tant qu'utilisateur je veux Supprimer un cahier de recette éviter l'encombrement des cahiers périmés.	Développement de l'interface de suppression du cahier de recette	2
	Développement du WS de la suppression du cahier de recette	
	Consommation du WS dans la partie Front	
En tant qu'utilisateur, je veux visualiser un cahier de recette pour pouvoir consulter son contenu pour le vérifier.	Développement de l'interface de consultation de la liste des cahiers de recette crées.	1
	Développement du WS de la consultation des cahiers de recette.	
	Consommation du WS dans la partie Front	
En tant qu'utilisateur, je veux Exporter un cahier de recette pour l'importer dans HP ALM.	Développement de la partie du téléchargement dans l'interface de la liste des cahiers de recette.	3
	Développement de l'API de génération du fichier .xlsx contenant le cahier de recette.	

Tableau 6.1: Backlog du Sprint 4

6.2 Analyse

Dans cette section, nous allons faire l'étude analytique de ce sprint en présentant le diagramme de cas d'utilisation puis son raffinement, les description textuelles des cas d'utilisations de ce sprint. Enfin, nous allons présenter les diagrammes de séquence système de ces cas.
La figure ci-dessous présente le diagramme de cas d'utilisation du quatrième sprint.

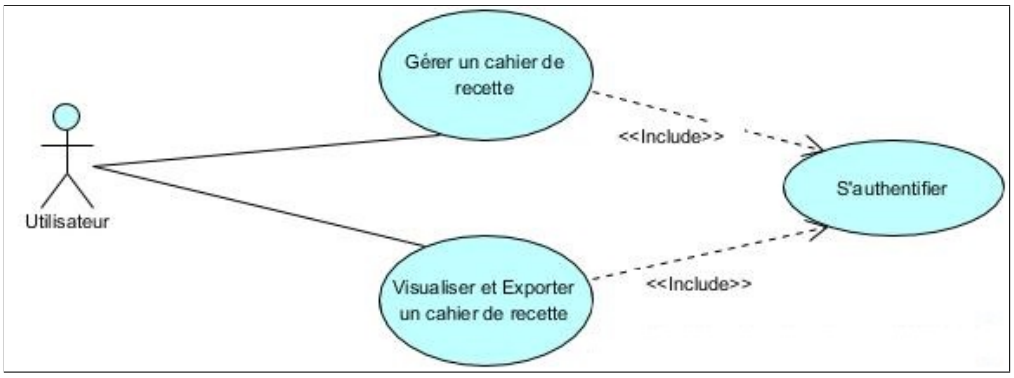


Figure 6.1: Diagramme de cas d'utilisation du Sprint 4

6.2.1 Diagramme de cas d'utilisation raffiné

La figure 6.2 présente le raffinement du diagramme de cas d'utilisation du sprint 4 en mettant en évidence les "use cases" de la gestion et l'exportation des cahier de recette.

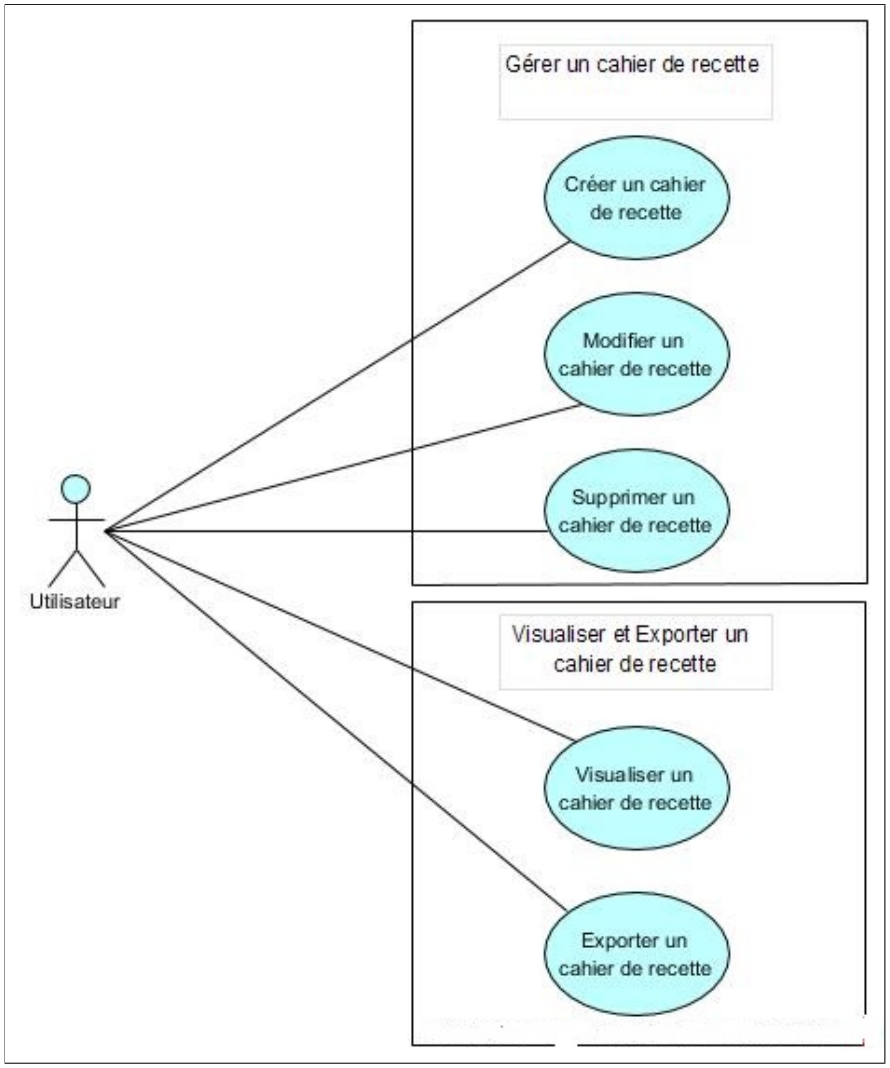


Figure 6.2: Diagramme de cas d'utilisation raffiné

6.2.2 Analyse du cas d'utilisation "Créer un cahier de recette"

Dans cette section, nous allons faire l'étude analytique du cas d'utilisation "Créer un cahier de recette" en présentant sa description textuelle et le diagramme de séquence système.

- **Description textuelle du cas d'utilisation "Créer un cahier de recette" :**

Le tableau 6.2 présente la description textuelle du cas d'utilisation "Créer un cahier de recette".

Titre	Créer un cahier de recette
But	Permettre à l'utilisateur de créer un cahier de recette.
Acteurs	Utilisateur
Précondition	Utilisateur authentifié.
Postcondition	Cahier de recette créé.
Scénario nominal	N.1. Le système affiche l'interface du catalogue de template N.2. L'utilisateur saisit le libellé et le périmètre du cahier à créer. N.3. L'utilisateur coche les scénarios du modèle qui va être inclus dans le cahier. N.4. L'utilisateur choisit le nombre d'occurrence de chaque scénario coché. N.5. L'utilisateur clique sur le bouton « Créer un cahier de test » N.6. Le système vérifie les données. N.7. Le système affiche l'arborescence du cahier créé.
Scénario alternatif	A.1 Données vides ou invalides A.1.1 Le système affiche un message d'erreur. A.1.2 Retour à l'action N.2 du scénario nominal.

Tableau 6.2: Description textuelle du cas d'utilisation "Créer un cahier de recette"

- **Diagramme de séquence système du cas d'utilisation "Créer un cahier de recette" :**

Dans la figure ci-dessous, nous trouvons le diagramme de séquence système du cas d'utilisation "Créer un cahier de recette".

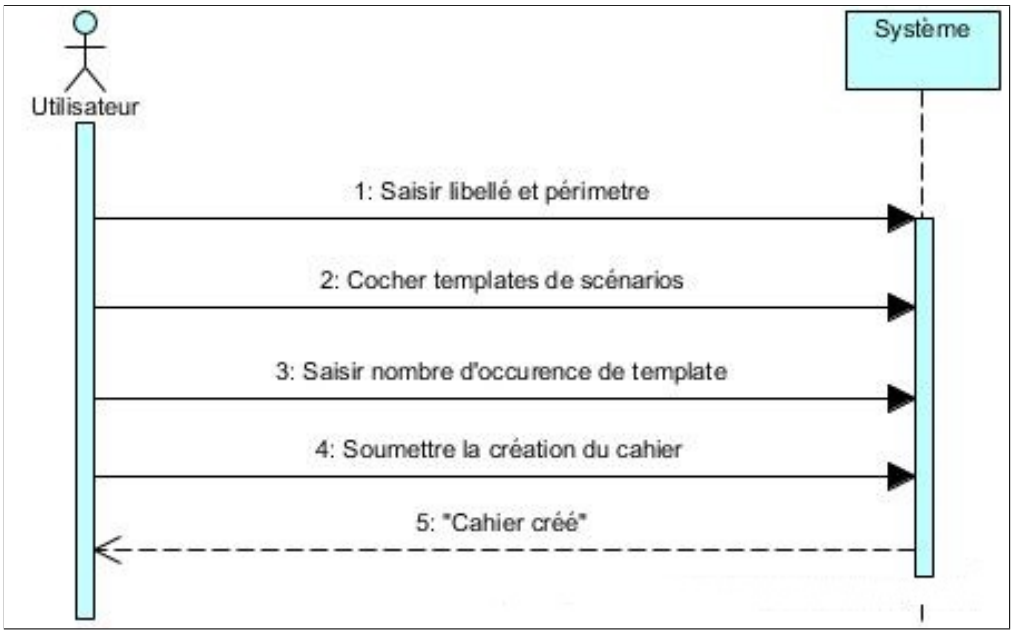


Figure 6.3: Diagramme de séquence système du cas d'utilisation "Créer un cahier de recette"

6.2.3 Analyse du cas d'utilisation "Exporter un cahier de recette"

Au cours de cette section, nous allons faire l'étude analytique du cas d'utilisation "Exporter un cahier de recette"en présentant sa description textuelle et le diagramme de séquence système.

- **Description textuelle du cas d'utilisation "Exporter un cahier de recette" :**

La description textuelle du cas d'utilisation "Exporter un cahier de recette" est présentée dans le tableau 6.3.

Titre	Exporter un cahier de recette
But	Permettre à l'utilisateur d'exporter un cahier de recette en format xlsx.
Acteurs	Utilisateur
Précondition	Utilisateur authentifié.
Postcondition	Cahier exporté.
Scénario nominal	N.1. Le système affiche l'interface contenant la liste des cahiers de recette. N.2. L'utilisateur clique sur le bouton de téléchargement devant les informations du cahier de recette à exporter. N.3. Le met le fichier contenant le cahier en téléchargement.
Scénario alternatif	A.1 Données vides ou invalides A.1.1 Le système affiche un message d'erreur. A.1.2 Retour à l'action N.2 du scénario nominal.

Tableau 6.3: Description textuelle du cas d'utilisation "Exporter un cahier de recette"

- **Diagramme de séquence système du cas d'utilisation "Exporter un cahier de recette" :**

La figure 6.4 présente le diagramme de séquence système du cas d'utilisation "Exporter un cahier de recette".

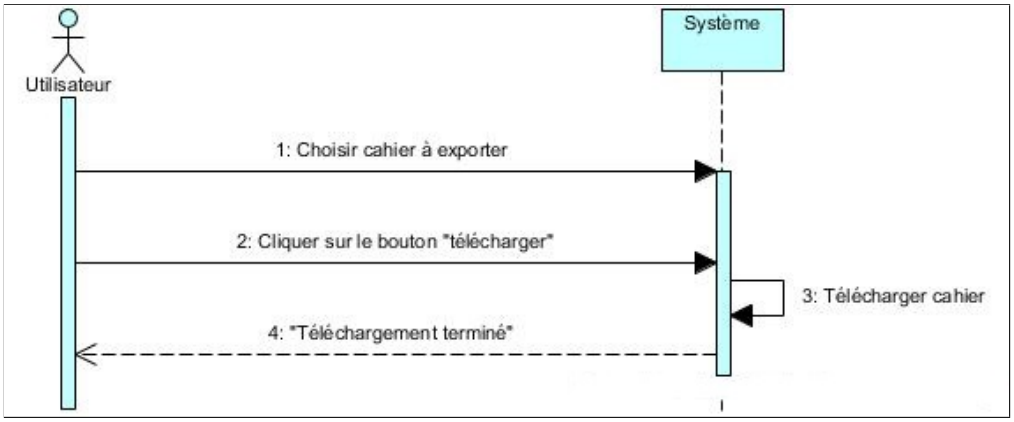


Figure 6.4: Diagramme de séquence système du cas d'utilisation "Exporter un cahier de recette"

6.3 Conception

Au cours de cette section, nous allons réaliser la conception reliée à ce sprint à travers quelques diagrammes(Compsants et séquences objet).

6.3.1 Diagramme de composants

Dans la figure 6.5, nous allons expliquer les interactions entre les différentes composants de notre application à savoir le model, Repository, Service, les contrôleurs.
Ces services seront par la suite exposés comme des Web services et vont être consommés par la partie Front de notre application.

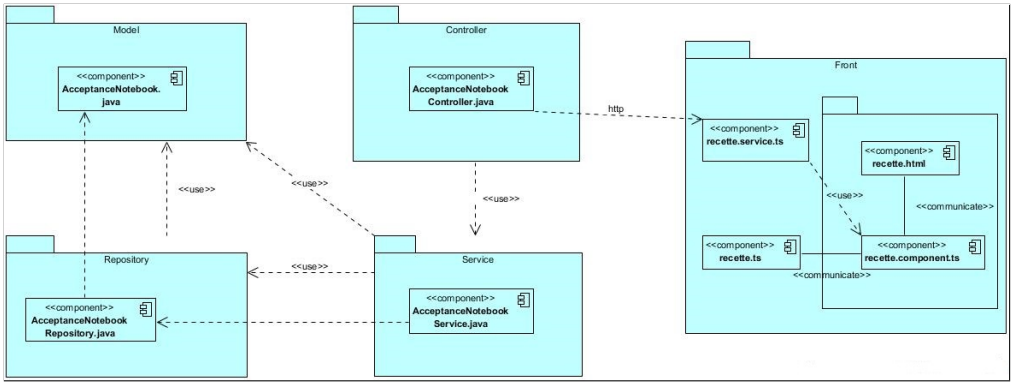


Figure 6.5: Diagramme de composants du Sprint 4

6.3.2 Diagrammes de séquence objet

- **Diagramme de séquence objet du cas d'utilisation "Créer un cahier de recette"**

La figure 6.6 présente le diagramme de séquence objet du cas d'utilisation "Créer un cahier de recette". Après avoir saisir les coordonnées de bases d'un cahier de recette, l'utilisateur clique sur le bouton de création du cahier pour le persister dans la base de données et puis il sera notifié du succès de la création comme le montre le diagramme suivant.

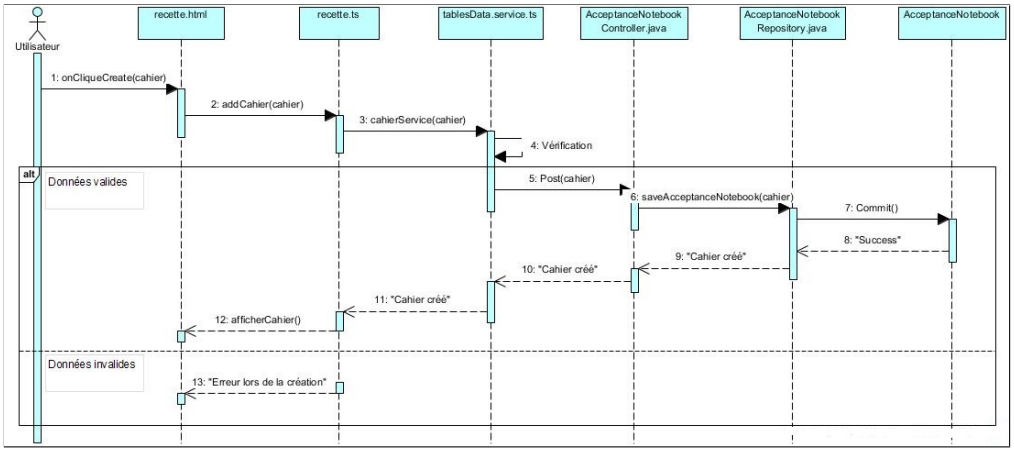


Figure 6.6: Diagramme de de séquence objet du cas d'utilisation "Créer un cahier de recette"

- **Diagramme de séquence objet du cas d'utilisation "Exporter un cahier de recette"**

La figure 6.7 présente le diagramme de séquence objet du cas d'utilisation "Exporter un cahier de recette". Pour exporter le cahier de recette il suffit que l'utilisateur clique sur le bouton de téléchargement pour récupérer le cahier et l'exporter tout de suite.

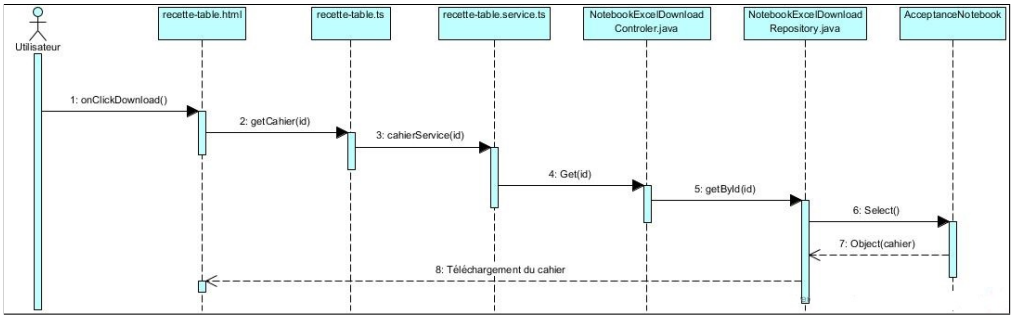


Figure 6.7: Diagramme de séquence objet du cas d'utilisation "Exporter un cahier de recette"

6.3.3 Diagrammes de classes participantes

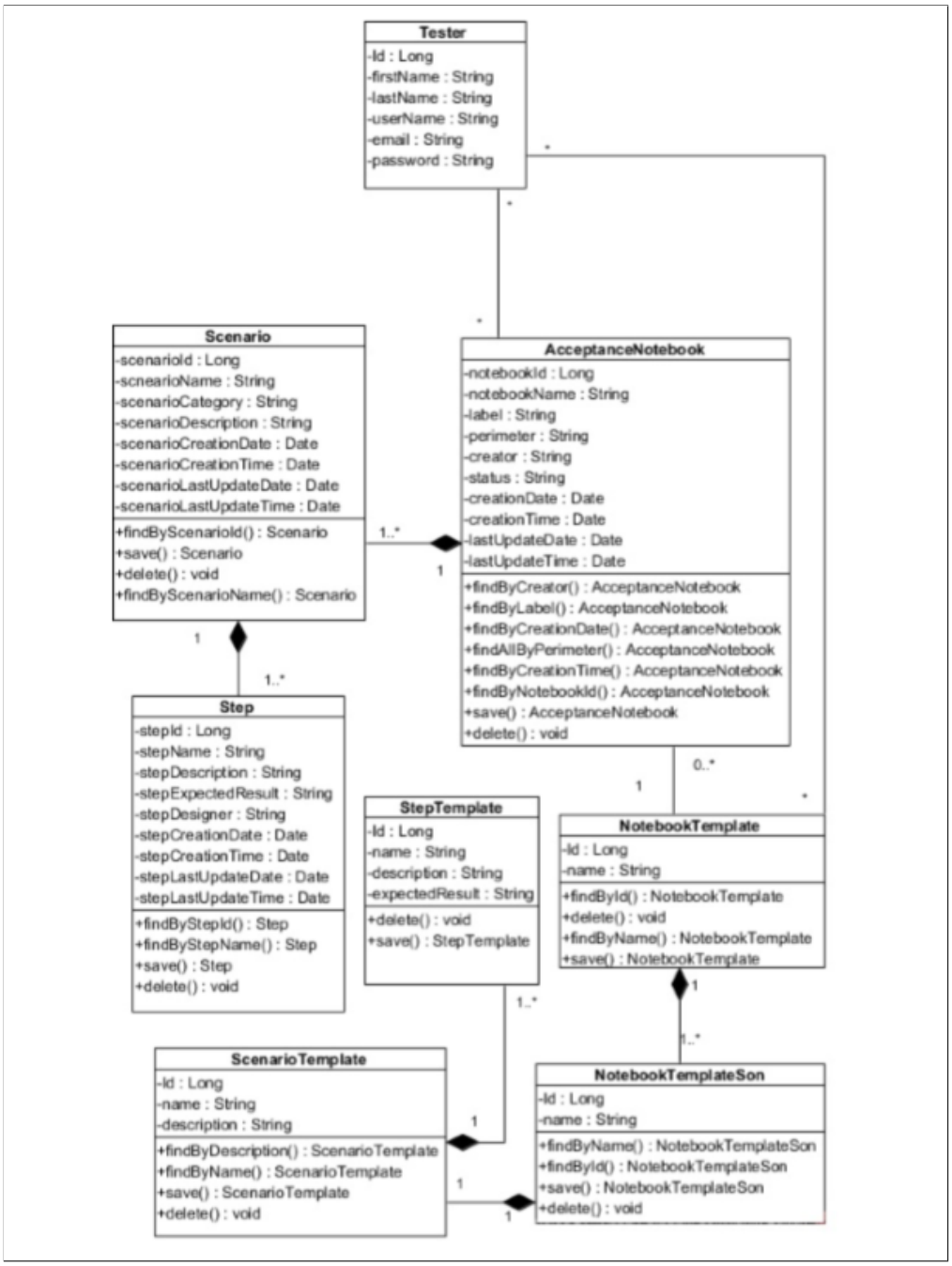


Figure 6.8: Diagramme de classes participantes du Sprint 4

6.4 Réalisation

La liste suivante indique en détail la contribution de la solution à la réduction de la quantité d'informations redondantes que le testeur doit écrire dans les cellules des neuf colonnes du modèle de feuille de calcul Excel :

- **Type :** Il faut que cette colonne contient la valeur "Manuel" toujours pour notre cas.
- **Test Name :** Pour chaque scénario, cette colonne contient l'acronyme du périmètre et son intitulé.
- **Description :** Une description de l'objectif du test.
- **Step Name (Design Steps) :** Il s'agit de la numérotation des étapes du scénario.
- **Description (Design Steps) :** Elle est conforme au modèle de scénario.
- **Expected Result (Design Steps) :** La ou les résultats attendu.
- **Subject :** Ceci est le périmètre fonctionnel du cahier de test.
- **Designer :** Le concepteur du cahier de recette
- **Creation Date :** La date de création du cahier de recette qui n'est plus au testeur de l'écrire, elle est générée automatiquement.

Dans la Figure 6.9, nous illustrons l'interface créée pour effectuer l'opération de création d'un cahier de recettes. Le testeur ne doit taper qu'un libellé et un périmètre fonctionnel du cahier. Ensuite, après avoir parcouru le catalogue et identifié le modèle souhaité, il coche les scénarios à inclure dans le cahier et choisit pour chacun son nombre d'occurrence. Enfin, en appuyant sur le bouton "Créer cahier de test", notre cahier sera généré.

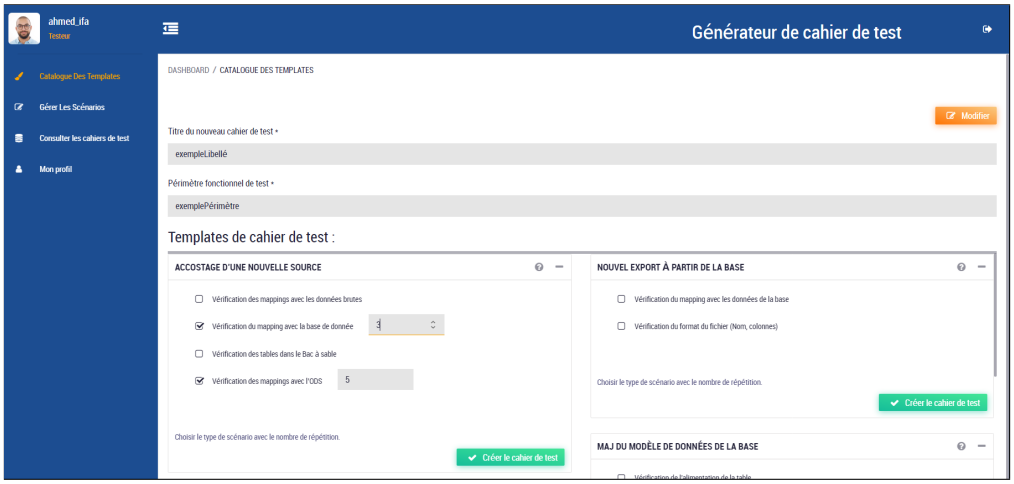


Figure 6.9: Interface de création du cahier de recette

Une fois le cahier généré, son squelette sera écrit dans une arborescence, comme illustré à la figure 6.10

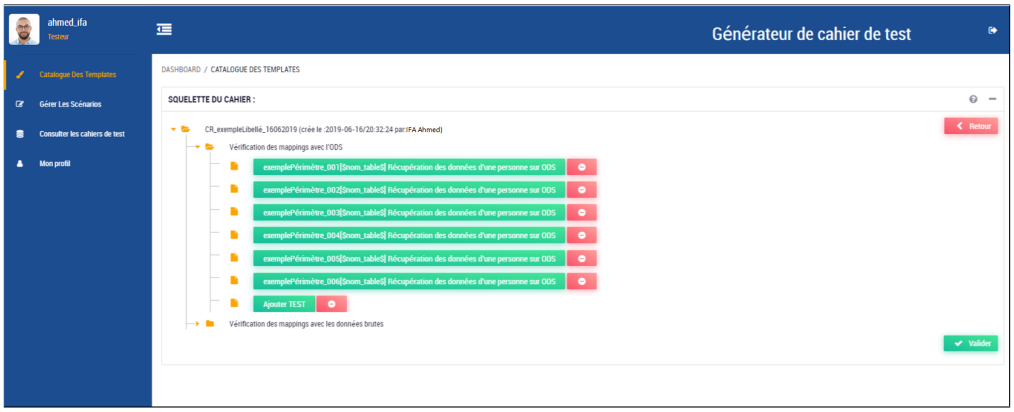


Figure 6.10: Interface de l'arborescence des scénarios d'un cahier généré

Toute action invalide effectuée par l'utilisateur doit être notifié, comme le montre la figure 6.11 et un message d'erreur s'affiche après laisser un champs obligatoire vide.

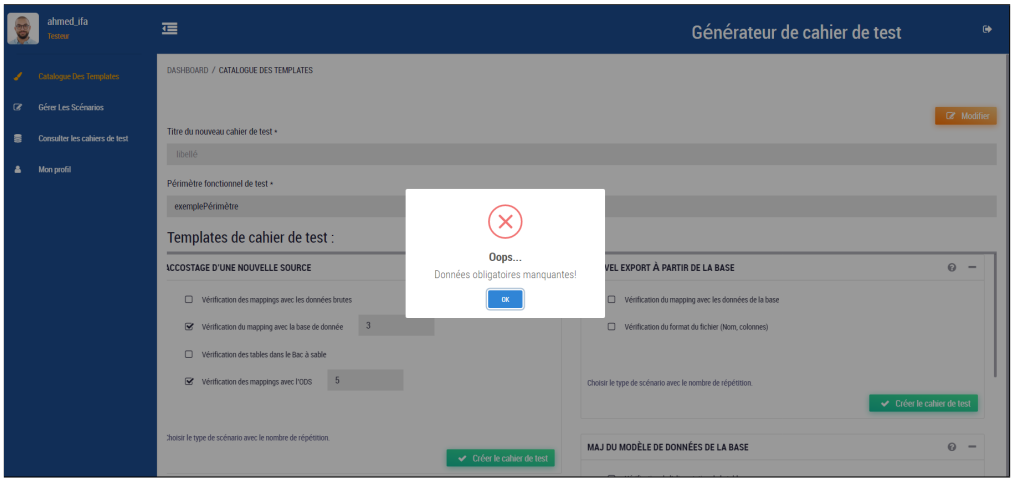


Figure 6.11: Alerte d'erreur lors de la création du cahier

Ceci est une barre de filtrage de la liste des livres de recettes affichés. Grâce à notre application, le testeur n'est plus obligé de gérer le stockage de ces livres de recettes, il peut toujours les trouver au même endroit.

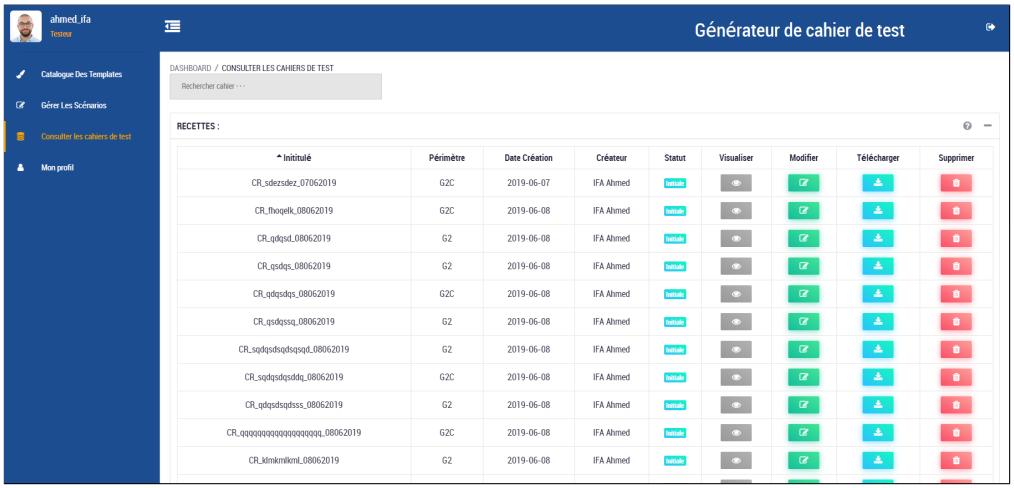


Figure 6.12: Interface de la liste et de gestion des cahiers de recette

Dans la figure précédente, nous trouvons les boutons pour gérer un cahier de recette déjà existant. Ces boutons sont :

— **Visualiser :** Ce bouton vous permet de visualiser la feuille de calcul Excel contenant le cahier en donnant à l'utilisateur une meilleure lisibilité en distinguant les scénarios de leurs étapes par changer le couleur d'un scénario à l'autre, comme montré dans la figure suivante.

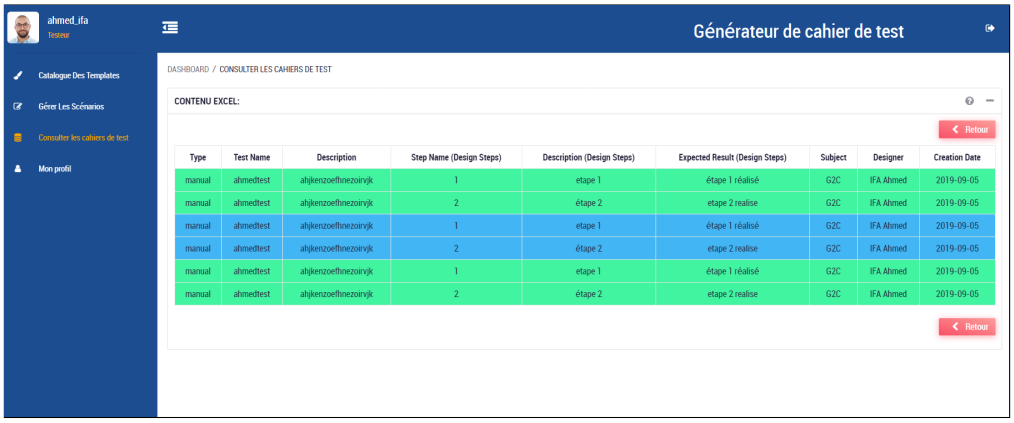


Figure 6.13: Interface de l'aperçu du cahier de recette

— **Modifier :** En cliquant sur ce bouton, l'utilisateur peut afficher les scénarios d'un cahier de recette dans une arborescence, classés par catégorie (selon nom du modèle de scénario). Dans chaque scénario, le testeur peut soit le supprimer, soit en modifier le contenu.

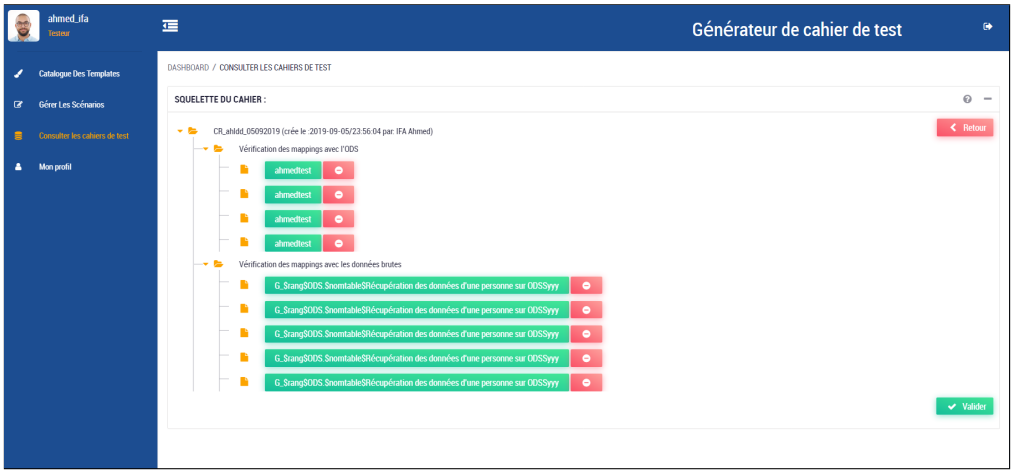


Figure 6.14: Interface de l'arborescence du cahier

— **Télécharger :** Ce bouton permet d'exporter le cahier de recette sous forme d'un fichier d'extension .xlsx (Tableur Excel) pour pouvoir l'importer dans l'outil de gestion HP ALM.

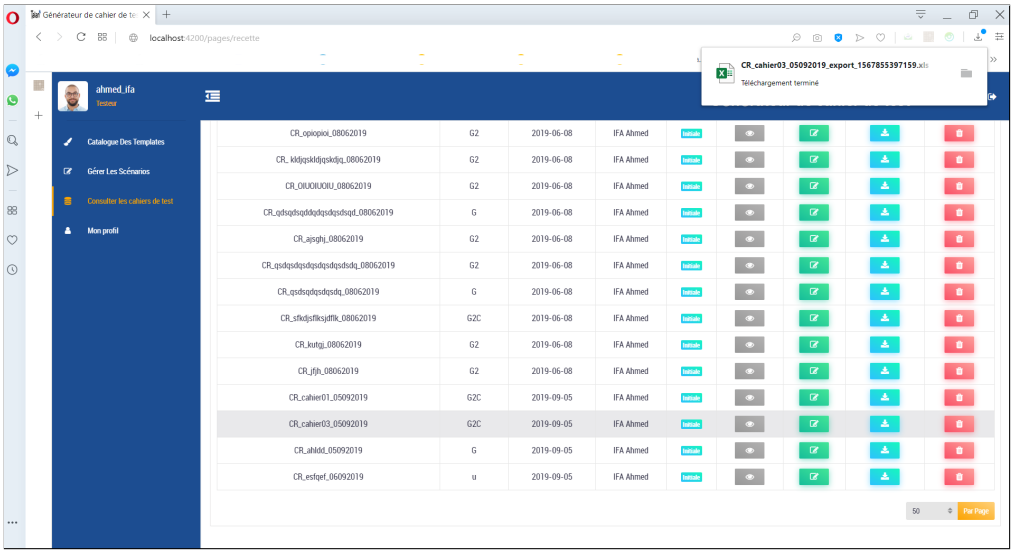


Figure 6.15: Interface de succès de téléchargement d'un cahier

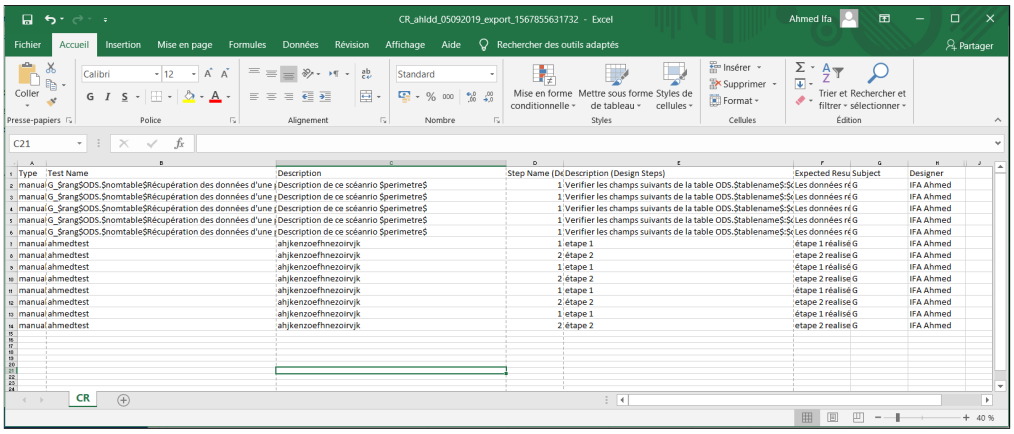


Figure 6.16: Capture d'un cahier exporté en fichier .xlsx

— **Supprimer :** C'est le bouton qui permet de supprimer un cahier de recette. En cliquant sur ce bouton, une demande de confirmation s'affiche pour que l'utilisateur confirme ou annule la suppression comme l'indique les deux figures suivantes.

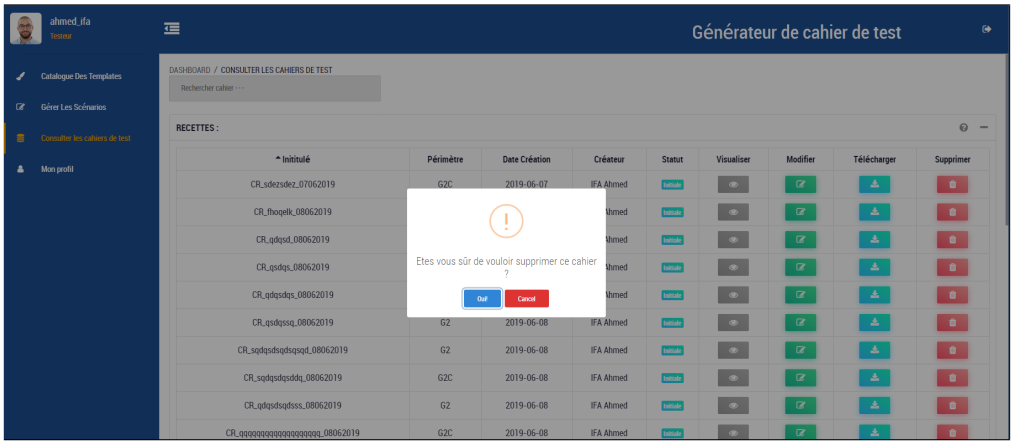


Figure 6.17: Message d'alerte pour confirmer la suppression du cahier

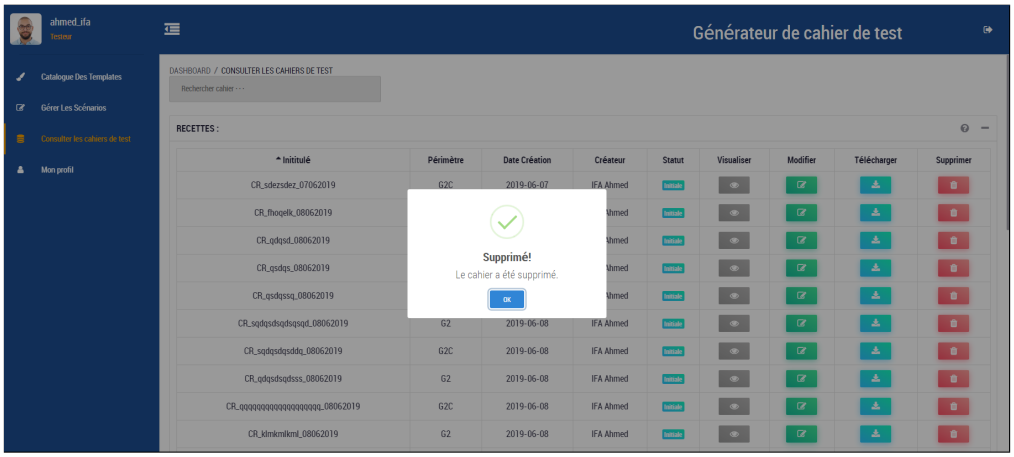


Figure 6.18: Popup informant le succès de la suppression du cahier

Conclusion

Nous avons traité dans ce chapitre le Backlog du sprint de la gestion et l'exportation de cahier de recette.Ensuite nous avons réalisé l'étude analytique et conceptuelle et nous avons terminé avec la partie réalisation qui met en valeur les interfaces de la gestion de cahier de recette.
En terminant ce sprint, nous pouvons dire que nous avons réussi à mener à bien le projet et à atteindre l'objectif de ce travail.

Conclusion générale

Ce stage de fin d'études chez "Talan Tunisie Consulting" a été une excellente occasion de mettre en oeuvre les acquis théoriques dans l'environnement pratique.

Notre projet consiste à la conception et la réalisation d'un générateur de cahiers de test pour une recette de base de données. Cette application sert à créer, renseigner et exporter un cahier de recette d'une façon simple et rapide.

Dans cette finalité, nous sommes passés par plusieurs itérations puisque nous avons utilisé la méthode Scrum. Notre rapport est alors constitué des chapitres suivants :
Dans le premier chapitre, nous avons fait une étude préalable du projet pour cerner la problématique puis nous avons présenté notre solution.
Dans le deuxième chapitre, nous avons étudié les besoins fonctionnels et non fonctionnels de notre solution et nous avons aussi mis en place son architecture physique et logique.
Dans le troisième chapitre, nous avons réalisé toutes les fonctionnalités liées à la gestion des utilisateurs.
Dans le quatrième chapitre, nous avons fait l'implémentions de la gestion de catalogue de template avec tous ses modèles des scénarios de test.
Dans le cinquième chapitre, nous nous sommes intéressés à la gestion des scénarios de test.
Enfin, dans le sixième et dernier chapitre, nous avons mis en place l'export d'un cahier de recette donné en format .xlsx.

Par ailleurs, le projet est ouvert à de nombreux aspects et perspectives. En effet, nous pouvons :

— Intégrer notre solution à l'ERP Byblos de Talan

— Intégrer des modules de messagerie instantanée entre les utilisateurs.

Enfin, ce projet nous a permis de vivre dans un environnement professionnel et de nous familiariser avec le travail en groupe en mettant en oeuvre la méthode Scrum dans un cycle de travail agile.

Références

[1] TALAN, *Site officiel*, [Visité le 20-Mars-2019], 2019. adresse : https://www.talan.com.

[2] TEST-RECETTE, *Test et recette informatique*, [Visité le 25-Juillet-2019], 2019. adresse : http://www.test-recette.fr/recette/acteurs.html.

[3] WIKIPEDIA, *Scrum*, [Visité le 28-Juillet-2019], 2019. adresse : https://fr.wikipedia.org/wiki/Scrum_(d%C3%A9veloppement).

[4] INTELIJ, *InteliJ*, [Visité le 20-Mai-2019], 2019. adresse : https://fr.wikipedia.org/wiki/IntelliJ_IDEA.

[5] WEBSTORM, *WebStorm*, [Visité le 20-Mars-2019], 2019. adresse : https://fr.wikipedia.org/wiki/WebStorm.

[6] WAMPSERVER, *WampServer*, [Visité le 20-Mars-2019], 2019. adresse : https://fr.wikipedia.org/wiki/WampServer.

ملخّص

يلخص هذا التقرير العمل المنجز في إطار مشروعيعرض هذا تقرير الأعمال المنجزة في إطار تربص ختم الدراسات للحصول على الشهادة الوطنية للمهندس في الاعلامية داخل مؤسسة تالان تونس. الهدف من هذا العمل هو تصميم وتطوير مولد حالات إختبار لفائدة مشروع إختبارات قبول قاعدة بيانات. قمنا بإنجاز هذا المشروع للإستجابة للحاجيات الوظيفية والوظيفية للمستخدم، وتعلم آخر التقنيات مثل انغلر وسبرينغ والرغبة في التكيف مع منهجية العمل داخل الفريق.

كلمات مفاتيح : حالة إختبار، إختبارات قبول، قاعدة بيانات، انغلر، سبرينغ، يو أم أل، سكروم

Résumé

Le présent rapport synthétise le travail eectué dans le cadre du projet de n d'études pour l'obtention du diplôme national d'ingénieur en informatique au sein de l'entreprise Talan Tunisie. L'objectif de ce travail est la conception et le développement d'un générateur de cas de test pour un projet de recette de Base de données. Nous avons réalisé ce projet avec l'ambition de répondre aux besoins fonctionnels et non fonctionnels d'un utilisateur nal, l'impatience d'apprendre des technologies récentes telles qu'Angular 4, Spring, et l'envie de s'adapter à une méthodologie de travail professionnel au sein d'une équipe.

Mots clés : Cas de test, recette, base de données, Angular, Spring, UML, SCRUM

Abstract

This report synthesizes the work done in the graduation project for the obtaining of the national Diploma of computer engineer in Talan Tunisie company. The objective of this work is the design and the development of a test cases generator for a database acceptance testing project. We realized this project with the aim to meet the functional and non-functional needs for the end user, impatience to learn the latest technologies such as Angular 4, Spring, and the desire to adapt a professional working methodology within a team.

Keywords : Test Case, acceptance, database, Angular, Spring, UML, SCRUM