
MEMOIRE

Présenté à

Institut Supérieur d'Informatique de Mahdia

En vue de l'obtention

De la Licence en informatique de gestion

Spécialité : Electronic Business(EB)

Spécialité : Business Intelligence(BI)

Par

Rania Souissi & Chiraz Bahri

DEVELOPPEMENT D'UNE APPLICATION WEB MOBILE GMP INSPECTION

Soutenu le 28/06/2022, devant la commission d'examen :

Mr. Soufien Gdaiem

Président

Mr. Hafedh Boukthir

Encadrant

Mme Takwa Ben Aicha

Examinateur



DÉDICACES

C'est avec profonde gratitude et des mots sincères, je dédie ce modeste travail.

La Mémoire de Ma chère Mère

À cette femme exceptionnelle qui m'inspire depuis toujours et à qui je dois tout, Que ce travail soit un hommage à votre mémoire, Que Dieu vous bénisse et vous accorde sa miséricorde.

À mon cher Père Khaled

Aucune dédicace ne peut exprimer mon profond amour, ma sincère gratitude et ma profonde reconnaissance pour les sacrifices que tu as consentis. Que ce travail soit le témoignage de ma reconnaissance et de mon infinie gratitude. Merci pour tout ce que tu as fait pour moi et que Dieu te protège et te procure santé et te préserve longue vie.

À mes chères sœurs Amal et Khouloud

Vous étiez toujours pour moi mes amis sincères et des confidents dévoué prêt à m'aider et me soutenir au long de mon parcours
Que ce travail soit le témoignage d'un grand amour et d'une affection profonde.

À mes chères cousines Basma et Nada

À qui j'éprouve tous mes sentiments d'amour et de tendresses. Nulle Dédicace ne saurait exprimer l'intensité de mon amour et la profondeur de mon attachement.

À ma chère meilleure amie et binôme Chiraz

Pour sa patience, sympathie et tout ce qu'elle a avancé afin de réussir ce projet.

À mes amis

Pour tous les instants inoubliables que j'ai passés avec vous.

R ania



DÉDICACES

Je dédie le fruit des efforts investis dans ce travail à tous ceux qui me sont chères :

Ma chère Mère Khadija

Tu m'as donné la vie, la tendresse et le courage pour réussir. Tout ce que je peux t'offrir ne pourra exprimer mon amour éternel et ma considération pour les sacrifices que vous avez consentis et pour l'affection dont tu m'as toujours entourée. Je ferai toujours de mon mieux pour rester ta fierté et ne jamais te décevoir.

Mon cher père Hadi

L'épaule solide, l'œil attentif, compréhensif. Je te dédie ce travail pour ta patience, ton soutien et tes encouragements. Aucune dédicace ne saurait exprimer mes sentiments. Que Dieu le tout-puissant te préserve, t'accorde santé, bonheur, et te protège de tout mal.

À mes chers et adorables frères Khalil et Mohammed

Merci pour leur encouragement et leur soutien moral, je vous aime beaucoup ! Que Dieu, le tout-puissant, vous protège et vous garde.

À ma chère amie et binôme Rania

Je ne peux pas trouver les mots justes et sincères pour exprimer mon affection et mes pensées. Nous avons passé des agréables moments ensemble qui vont rester gravés dans mon mémoire.

À toute personne chère à mon cœur.

C hiraz



Remerciements

Nous remercions Dieu le tout-puissant de nous avoir donné la santé et la volonté d'entamer et de terminer ce rapport

Nous exprimons, aussi, notre sincère gratitude et remerciements à notre encadrant à l'ISIMA **M. Hafedh Boukthir** pour l'aide et les conseils concernant l'acheminement du projet et l'architecture du rapport aussi pour le temps qu'il a consacré et l'ensemble des informations qu'il nous a fournies avec tendresse et compréhension, nous voulons le remercier pour leur patience et leur amabilité.

Nos remerciements **Madame Amani koobi** pour sa pédagogie et pour la confiance et le support qu'elle nous a accordé tout au long de ce projet au sein de l'EPI D'OR.

Nous remercions aussi notre encadrant professionnel **M. Mahdi Hattab** pour son accueil chaleureux et le partage de son expertise avec nous.

Nous tenons de même à remercier tous les enseignants de l'ISIMA qui nous ont enseigné pendant ces trois années successives et qui par leurs compétences nous ont soutenu dans la poursuite de nos études.

Enfin, nous présenterons notre profond remerciement et respect pour les jurys qui ont acceptés de juger notre travail.

TABLE DES MATIÈRES

DEDICACES	ii
DEDICACES	iii
REMERCIEMENTS	iv
LISTE DES FIGURES	x
LISTE DES TABLEAUX	x
LISTE DES ABRÉVIATIONS	xi
INTRODUCTION GÉNÉRALE	xii
1 Étude du projet	2
1.1 Introduction	3
1.2 Cadre et contexte du sujet	3
1.2.1 Cadre du projet	3
1.2.2 Présentation du sujet	3
1.2.3 Objectifs à atteindre	3
1.3 Présentation de l'organisme d'accueil	4
1.4 Problématique	4
1.4.1 Étude de l'existant	5
1.4.2 Critique de l'existant	5
1.4.3 Solution proposée :	8
1.5 Méthodologie adoptée	9
1.5.1 Méthodologie agile Scrum :	9
1.5.2 5 raisons pour lesquelles Scrum est idéal pour les développeurs	9
1.6 Langage de modélisation	11
1.7 Conclusion	11
2 Planifications et spécifications des besoins	12

TABLE DES MATIÈRES

2.1	Introduction	13
2.2	Analyse et spécification des besoins	13
2.2.1	Spécification des besoins fonctionnels :	13
2.2.2	Spécification des besoins non fonctionnels :	14
2.3	Identification et structuration de cas d'utilisation	14
2.3.1	Identifications des acteurs :	14
2.3.2	Diagramme de cas d'utilisation global :	14
2.4	Pilotage de projet avec SCRUM	15
2.4.1	Équipe et rôles :	15
2.4.2	Le Backlog du produit :	15
2.4.3	Structure et Découpage des sprints :	17
2.4.4	Diagramme de Gantt :	18
2.5	Environnement de travail	19
2.5.1	Environnement matériel :	19
2.5.2	Environnements logiciels :	19
2.5.3	Technologies et langages utilisées :	21
2.6	Architecture de l'application	22
2.6.1	Architecture physique :	22
2.6.2	Architecture logique :	23
2.7	Conclusion	24
3	Sprint 1 : Authentification, Gérer les inscriptions et Gestion des comptes	25
3.1	Introduction	26
3.2	Backlog du sprint 1	26
3.3	Spécifications fonctionnelles	27
3.3.1	Classification des cas d'utilisation par acteur :	27
3.3.2	Diagramme des cas d'utilisation :	27
3.4	Conception	30
3.4.1	Diagramme de séquence :	30
3.4.2	Diagramme de classes global du premier Sprint :	31
3.5	Réalisation	32
3.5.1	Interface d'authentification :	32
3.5.2	Interface d'inscription :	34
3.5.3	Interface ajouter, modifier, supprimer un compte dans la base de données :	36
3.5.4	Test fonctionnel	36
3.5.5	Logo de l'application :	37
3.6	Conclusion	37

TABLE DES MATIÈRES

4 Sprint 2 : Gestion des check-lists	38
4.1 Introduction	39
4.2 Backlog du sprint 2	39
4.3 Spécifications fonctionnelles	40
4.3.1 Classification des cas d'utilisation par acteur :	41
4.3.2 Diagramme des cas d'utilisation :	41
4.4 Conception	43
4.4.1 Diagramme de séquence :	43
4.4.2 Diagramme de classes global du deuxième Sprint :	46
4.5 Réalisation	46
4.5.1 Interface d'accueil :	46
4.5.2 Interface choisir check-list :	47
4.5.3 Interface remplir check-list :	49
4.5.4 Interface consulter check-list :	51
4.5.5 Interface imprimer check-list :	52
4.5.6 Test fonctionnel	53
4.6 Conclusion	54
5 Sprint 3 : Gestion des Scores, Gestion du Tableau de bord	55
5.1 Introduction	56
5.2 Backlog du sprint 3	56
5.3 Spécifications fonctionnelles	56
5.3.1 Classification des cas d'utilisation par acteur :	57
5.3.2 Diagramme des cas d'utilisation :	57
5.4 Conception	59
5.4.1 Diagramme de séquence :	59
5.4.2 Diagramme de classes global du troisième Sprint :	61
5.5 Test	61
5.5.1 Interface consulter score :	61
5.5.2 Interface consulter tableau de bord :	62
5.6 Conclusion	63
CONCLUSION GÉNÉRALE	64
NETOGRAPHIE	64

LISTE DES FIGURES

1.1	Logo de L'EPI D'OR	4
1.2	Page d'accueil de iAuditor	6
1.3	Page d'accueil de GoAudits	7
1.4	Page d'accueil de first audit	8
1.5	cycle de vie scrum	10
1.6	Logo uml	11
2.1	Digramme de cas d'utilisation global	15
2.2	Structure du projet	17
2.3	Découpage du projet en sprints	18
2.4	Diagramme de Gantt	18
2.5	Logo Visual Studio Code	19
2.6	Logo Drawo.io	20
2.7	Logo Postman	20
2.8	Logo GitHub	20
2.9	Logo Overleaf	21
2.10	Logo Flutter	21
2.11	Logo MongoDB	21
2.12	Logo Node.js	22
2.13	Logo Dart	22
2.14	l'architecture de l'application	23
2.15	Schéma MVC	24
3.1	Raffinement de cas d'utilisation « S'authentifier »	28
3.2	Raffinement de cas d'utilisation « Gérer les inscriptions »	29
3.3	diagramme de séquence « s'authentifier »	30
3.4	Diagramme de séquence « Créer un compte »	31
3.5	Diagramme de classes global du premier sprint	32
3.6	interface d'authentification web	33
3.7	interface d'authentification mobile	34
3.8	interface d'inscription	35

LISTE DES FIGURES

3.9	interface ajouter, modifier et supprimer	36
3.10	Test d'authentification	36
3.11	Test créer un compte	37
3.12	logo de l'application	37
4.1	Raffinement de cas d'utilisation « Gérer les check-lists »	41
4.2	Diagramme de séquence « remplir et sauvegarder check-list »	43
4.3	Diagramme de séquence « consulter check-list »	44
4.4	Diagramme de séquence « Imprimer check-list »	45
4.5	Diagramme de classes global du deuxième Sprint	46
4.6	Interface d'accueil	47
4.7	Interface pour les Check-lists hygiène des équipements et locaux	48
4.8	Interface matricule	49
4.9	Interface check-list personnelle	49
4.10	Interface remplir check-list	50
4.11	Interface choisir nom et date de check-list	51
4.12	Interface consulter check-list	51
4.13	Interface imprimer check-list	52
4.14	Test remplir check-list	53
4.15	Test consulter check-list	54
5.1	Raffinement de cas d'utilisation « Gérer les Scores et le tableau de bord»	57
5.2	Diagramme de séquence « Consulter score »	59
5.3	Diagramme de séquence du cas « Consulter le tableau de bord »	60
5.4	Diagramme de classes global du troisième Sprint	61
5.5	Interface consulter score	62
5.6	Interface consulter tableau de bord	63

LISTE DES TABLEAUX

2.1	<i>Backlog du Produit</i>	16
2.2	<i>Tableau comparatif des différentes API de cartographie</i>	19
3.1	Backlog du sprint 1.	26
3.2	<i>Classification des cas d'utilisations par acteur</i>	27
3.3	Description textuelle du cas « S'authentifier »	28
3.4	Description textuelle du cas d'utilisation « créer un compte »	29
4.1	Backlog du sprint 2.	39
4.2	<i>Classification des cas d'utilisations par acteur</i>	41
4.3	Description textuelle de cas d'utilisation « remplir et sauvegarder »	42
4.4	Description textuelle de cas d'utilisation« consulter et imprimer »	42
5.1	Backlog du sprint 3.	56
5.2	<i>Classification des cas d'utilisations par acteur</i>	57
5.3	Description textuelle de cas d'utilisation « gérer les scores »	58
5.4	Description textuelle de cas d'utilisation « consulter le tableau de bord »	58



LISTE DES ABRÉVIATIONS

API interface de programmation d’application

GMP Good Manufacturing Practice

HTTP Hypertext Transfer Protocol

JSON JavaScript Objet Notation

SMSDA Système de Management de la Sécurité des Denrées Alimentaires

TIC Technologies de l’information et de la communication

UML Unified Modeling Language

UNPA Usine nouvelle des pates alimentaires

SCRUM Systematic Customer Resolution Unraveling Meeting



INTRODUCTION GÉNÉRALE

De nos jours, les applications web et mobiles couvrent une large gamme de besoin non seulement pour les individus, mais aussi pour les entreprises. En effet, elles sont d'avantage utilisé dans les multiples domaines tels que le marketing, la santé, la finance, commerce, l'industrie et l'agroalimentaire.

Le système informatique est devenu essentiel dans notre vie. Son objectif est d'automatiser le traitement de l'information et ayant pour finalité d'élaborer, stocker, acheminer, présenter ou détruire les données. Personne ne peut nier qu'il est une révolution innovante et fondamentale qui a touché considérablement la vie humaine depuis l'apparition des systèmes informatique.

N'étant pas une tendance passagère, le système informatique s'est imposé peu à peu et a envahi tous les domaines de la vie quotidienne pour minimiser et faciliter la mémorisation et la réalisation des tâches de personnes.

Parmi les sujets les plus curieux suscitant le plus des commentaires à l'heure actuel, on trouve l'impact de la nouvelle technologie sur le monitoring des données. Pour bien structurer les données, le monitoring est aujourd'hui un passage obligatoire. Ce type d'outil est capable de récupérer et corrélérer différents indicateurs au sein des bases de données et serveurs de logiciels et applications métier, dans le but d'informer l'état de santé de ce système en temps réel.

C'est dans ce cadre que s'inscrit notre projet de fin d'études, réalisé au sein de la société L'EPI D'OR, dans lequel nous nous étions intéressées à la conception et à la réalisation d'une application web mobile intitulé GMP inspection pour la vérification et le monitoring des programmes prérequis du système de management de la sécurité des aliments.

Notre rapport se subdivisera en cinq chapitres :

– Le premier chapitre « **Étude du projet** » présente le cadre de notre projet commençant par la présentation du projet, ainsi que les objectifs attendus. Ensuite l'étude préalable composée de la problématique, étude de l'existant, critique de l'existant. Et finissant par la présentation de solution

INTRODUCTION GÉNÉRALE

proposée et la Méthodologie, ainsi que le choix du langage de la modélisation.

- Le deuxième chapitre « **Planifications et spécifications des besoins** » consiste à présenter les besoins fonctionnels et non fonctionnels, description des tâches, le diagramme de cas d'utilisation globale de notre application, pilotage de projet avec Scrum, environnement de travail ainsi que l'architecture de notre application.
- Le troisième chapitre Sprint 1 « **Authentification, Gérer les inscriptions et Gestion des comptes** » contiendra la première itération de notre projet.
- Le quatrième chapitre Sprint 2 « **Gestion des check-lists** » contiendra la partie la plus importante de notre projet.
- Le cinquième chapitre Sprint 3 « **Gestion des Scores, Gestion du Tableau de bord** » est le dernier chapitre.

Étude du projet

Sommaire

1.1	Introduction	3
1.2	Cadre et contexte du sujet	3
1.2.1	Cadre du projet	3
1.2.2	Présentation du sujet	3
1.2.3	Objectifs à atteindre	3
1.3	Présentation de l'organisme d'accueil	4
1.4	Problématique	4
1.4.1	Étude de l'existant	5
1.4.2	Critique de l'existant	5
1.4.3	Solution proposée :	8
1.5	Méthodologie adoptée	9
1.5.1	Méthodologie agile Scrum :	9
1.5.2	5 raisons pour lesquelles Scrum est idéal pour les développeurs	9
1.6	Langage de modélisation	11
1.7	Conclusion	11

1.1 Introduction

Ce chapitre introductif sera le départ pour la compréhension du sujet. Nous commençons la première partie par le cadre et le contexte du notre sujet. Ensuite, nous étudions le système actuel afin de cibler les insuffisances et l'embauche de nouvelles solutions.

1.2 Cadre et contexte du sujet

1.2.1 Cadre du projet

Ce travail s'inscrit dans le cadre d'un stage de fin d'études en vue de l'obtention d'un diplôme de fin d'études en licence en informatique de gestion spécialité business intelligence et E-business. Durant un stage de quatre mois effectués à la société L'EPI D'OR, notre mission consiste à concevoir et développer une application web et mobile pour la vérification et le monitoring des programmes requis du système de management de la sécurité des aliments.

1.2.2 Présentation du sujet

La révolution technologique et l'émergence des TIC poussent les secteurs public et privé à intégrer de nouveaux outils visant à optimiser la productivité. Dans cette optique, la société L'EPI D'OR nous a confié la réalisation d'une application web et mobile de vérification et de suivi des programmes prérequis du système de management de la sécurité alimentaire. Ainsi, chaque utilisateur dispose de son propre login et mot de passe. C'est une application d'accès sécurisé.

1.2.3 Objectifs à atteindre

Notre mission consiste à :

- Aider l'équipe qualité à éliminer la saisie manuelle des données.
- Élaboration des statistiques pour consulter des informations précises sur ses produits.
- Réduire les erreurs humaines et augmenter l'efficacité.

1.3 Présentation de l'organisme d'accueil

L'Usine Nouvelle des Pâtes Alimentaires L'EPI D'OR, installée à Sousse, dans le centre du pays, est la plus ancienne des unités de production de pâtes alimentaires en Tunisie. Elle fut créée par feu Habib MELLOULI, un pionnier de l'industrie de fabrication des pâtes alimentaires en Tunisie. Habib MELLOULI évoluait, dans la branche d'activité de pastification, depuis 1943, marquant ainsi une carrière d'un demi-siècle.



FIGURE 1.1 – Logo de L'EPI D'OR

L'entreprise exploite actuellement un complexe industriel de transformation comprenant :

- 1- Une unité de transformation de blé : Minoterie et semoulerie.
- 2- Une unité de production des pâtes (longues et courtes).
- 3- Une unité de production des pâtes spéciales et de couscous.
- 4- Une unité d'exportation.
- 5- Une unité de transport.
- 6- Une unité de distribution.

L'entreprise est dirigée par Monsieur Jaleleddine MELLOULI, de formation universitaire et pastier de père en fils. Elle emploie 800 personnes dont une centaine de cadres diplômés. Malgré la concurrence très vive, l'entreprise arrive à bien se positionner et se maintient dans le peloton de tête.

1.4 Problématique

L'application Accès intitulé « Hazard Monitoring » est exploitée par l'équipe qualité de l'UNPA depuis 2009. Cependant, l'équipe qualité était dans l'obligation de faire des mises à jour dont elle ne

pouvait pas les introduire au niveau de la base Accès. De ce fait, ils ont recopié toutes les données sur un fichier Excel dont ils utilisent depuis plus d'un an.

L'usage du fichier Excel est jugé non pratique à cause de la lourdeur des fichiers ainsi que l'impossibilité d'avoir des statistiques. De même pour l'ancienne l'application, l'équipe qualité ne pourrait pas avoir des statistiques. .

1.4.1 Étude de l'existant

L'étude de l'existant appartient à la phase d'analyse du projet. Cette étape primordiale sert à mettre en route le projet et permet de définir un cadre, d'éviter les redites, d'aiguiser l'esprit critique et de favoriser la créativité. Cette étape permet aussi de détecter les points forts et les points faibles d'une application. Actuellement, il existe plusieurs types d'applications classées comme, iAuditor, GoAudits, First audit...

1.4.2 Critique de l'existant

Pour assurer notre objectif, il est nécessaire que nous commençons par présenter les critiques de l'existant qui permet de déterminer les points forts et les points faibles d'une application.

Dans ce qui suit, nous présentons une analyse de quelques exemples d'application :

iAuditor

iAuditor est une application mobile d'inspection de listes de contrôle utilisée pour créer des listes de contrôle, effectuer des inspections et générer des rapports à partir d'un appareil ou d'une tablette.[1]

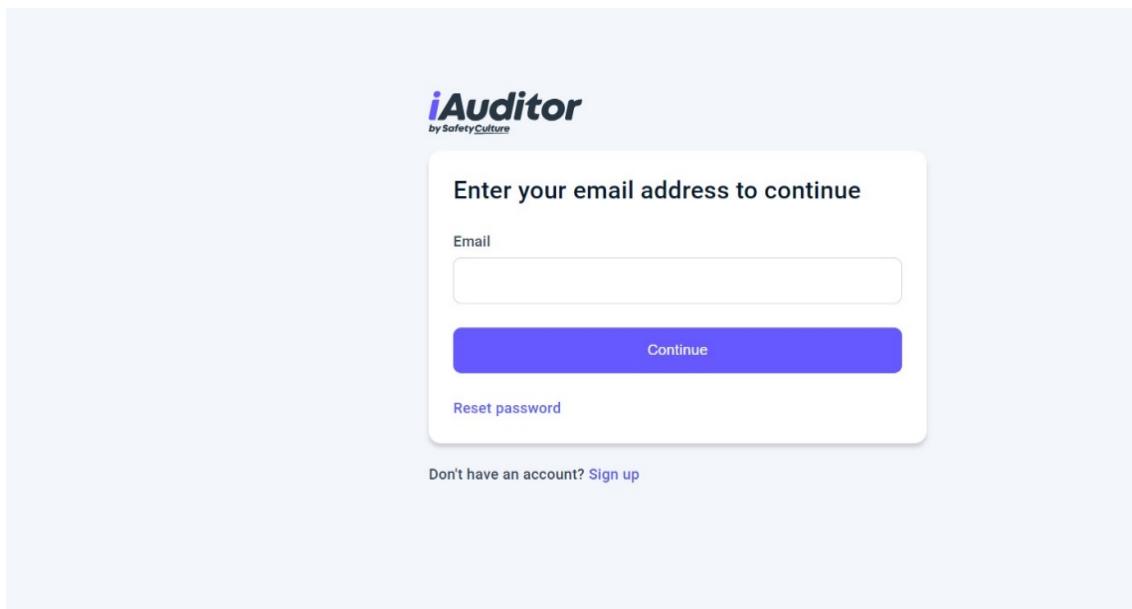


FIGURE 1.2 – Page d'accueil de iAuditor

Avantages :

- Enregistrement automatique, édition facile et rappel des inspections précédentes.
- Permet d'envoyer les résultats d'inspection du terrain aux clients.
- Facile de créer et de personnaliser des audits et des listes de contrôle et d'obtenir exactement ce que vous voulez.

Inconvénients :

- Ils ont parfois du mal à synchroniser leurs inspections.
- Le téléchargement d'un modèle que vous créez en version PDF nécessite beaucoup d'efforts.
- Gestion multi-documents non disponibles.

GoAudits

Plateforme mobile intelligente pour l'audit et les inspections. Un moyen efficace d'améliorer la qualité, la sécurité, les normes opérationnelles et la conformité...[2]

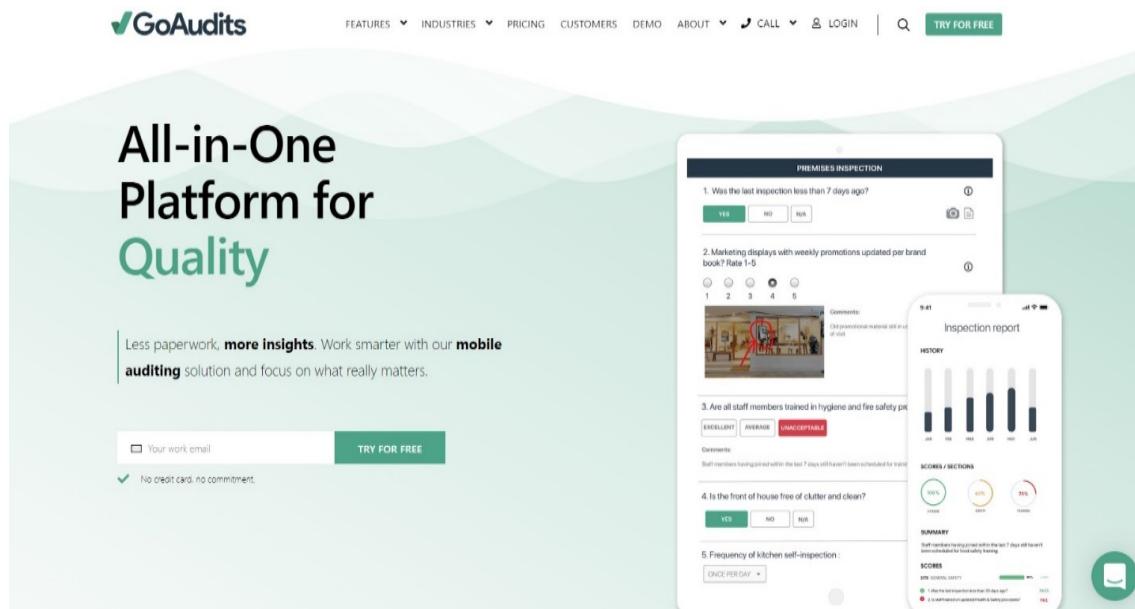


FIGURE 1.3 – Page d'accueil de GoAudits

Avantages :

- Facile à utiliser.
- L'application fournit des rapports détaillés avec des pourcentages, des graphiques et met en évidence les critères communs manquants.
- Possibilité de créer et de modifier vos audits ou de choisir parmi une gamme de modèles.

Inconvénients :

- La navigation dans le tableau de bord et le système arrière était difficile.
- Les fonctionnalités ne sont pas bien fixées avec les listes de contrôle.
- Difficile à utiliser pour créer des listes de contrôle longues et détaillées.

First audit

First audit est une application permettant de créer des listes de contrôle pour les inspections, la maintenance, les contrôles de maintenance, les journaux et les enquêtes...[3]



FIGURE 1.4 – Page d'accueil de first audit

Avantages :

- Facile à utiliser.
- Assistance rapide et efficace en cas de difficultés.
- Satisfaction des utilisateurs à tous les niveaux.

Inconvénients :

- Les champs personnalisés sont très utiles, mais une liste de valeurs prédéfinie ne peut pas être utilisée.
- Difficile à utiliser pour créer plusieurs listes de contrôle.
- Il manque une certaine flexibilité.

1.4.3 Solution proposée :

Pour remédier aux problèmes précédemment cités, L'EPI D'OR nous a confié la mission de créer une application web mobile qui permettrait l'accès par les responsables qualité pour consolider et organiser correctement les données, et la bonne gestion du temps. L'application fournira aux utilisateurs des processus de recherche et de suivi et obtiendra des statistiques.

1.5 Méthodologie adoptée

La méthodologie est nécessaire pour la gestion de n'importe quel type du projet, mais, il faut choisir la plus adoptée pour mieux gérer le projet. Puisque ce dernier est un projet de niveau évaluatif, alors, la meilleure méthode qui respecte cette demande est la méthode « Scrum ».

1.5.1 Méthodologie agile Scrum :

La méthode Scrum est considérée comme un Framework de gestion de projet, c'est la méthode la plus utilisée parmi les autres méthodes agiles existantes. En effet, le terme Scrum est apparu pour la première fois en 1986. Il a été publié par "Hirotaka Takeuchi" et "Ikujiro Nonaka" et il décrit une nouvelle approche plus rapide et plus flexible du développement de nouveaux produits. Cette méthode a été comparée à une union de rugby, dont le principe de base, est que toute l'équipe avance ensemble et est prête à se réorienter au fur et à mesure de l'avancement. C'est le bon concept de l'avancement de développement du projet.[4]

1.5.2 5 raisons pour lesquelles Scrum est idéal pour les développeurs

- SCRUM réalise la compréhension du travail en divisant le projet en des tâches et en des petites parties qui peuvent être implémentées de manière itérative. Ces itérations sont appelées « sprints ».
- L'équipe de développement contrôle la quantité de travail qu'elle effectue et obtient rapidement des "Feedback" pour comprendre ce qu'il faut faire ensuite.
- Concentration claire, favorisant le travail d'équipe et limitant les distractions.
- Une plus grande capacité à intégrer les changements au fur et à mesure qu'ils se produisent
- Avec Scrum, le travail est effectué par l'équipe de développement simultanément plutôt que séquentiellement. Les programmeurs codent « à la volée » et n'attendent pas que toutes les

questions aient reçu une réponse et que tout soit parfaitement clair avant de commencer à programmer.

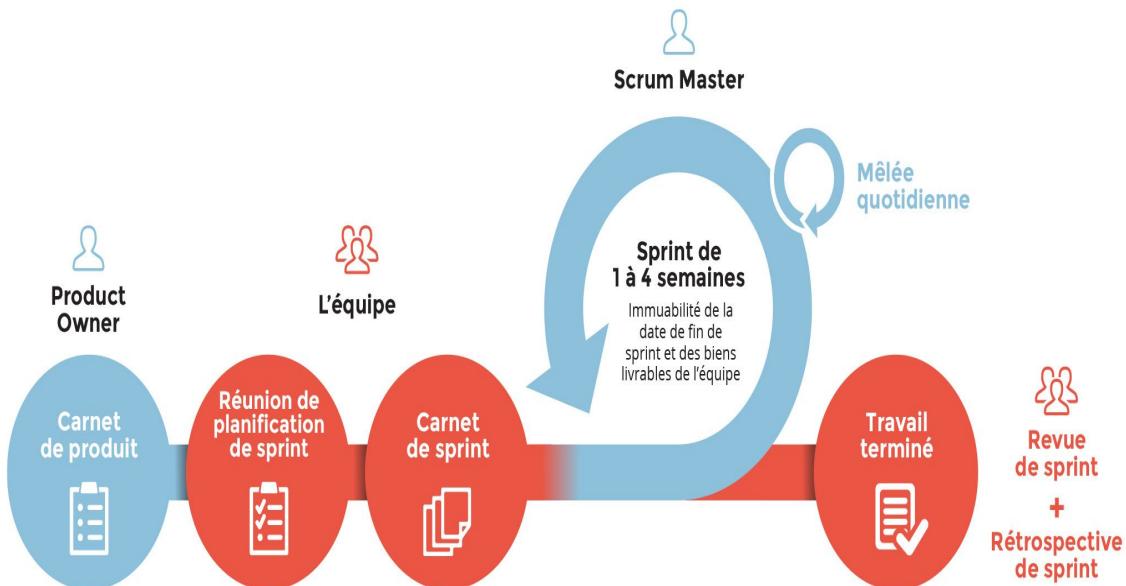


FIGURE 1.5 – cycle de vie scrum

Cette méthode est articulée autour de trois acteurs clés :

Product Owner : le seul qui appelle le client. Il travaille en interaction avec l'équipe de développement.

Il s'agit généralement d'un expert du domaine métier du projet.

L'équipe de développement : L'équipe de développement rassemble les personnes qui réalisent le travail. Elle est pluridisciplinaire et peut donc encapsuler d'autres rôles tels que développeur, architecte logiciel, analyste fonctionnel, graphiste/ergonome, ingénieur système.

Scrum master : qui doit maîtriser Scrum et s'assurer que ce dernier est correctement appliqué. Il a donc un rôle de coach à la fois auprès du Product Owner et auprès de l'équipe de développement. Il doit donc faire preuve de pédagogie. Il est également chargé de s'assurer que l'équipe de développement est pleinement productive.

1.6 Langage de modélisation

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



FIGURE 1.6 – Logo uml

1.7 Conclusion

Dans le premier chapitre, nous avons non seulement présenté notre projet de manière générale, mais aussi, nous avons défini les principaux objectifs de l'application et indiqué les méthodes qui seront utilisées pour accomplir notre projet. Dans le deuxième chapitre, nous allons spécifier les besoins fonctionnels et non fonctionnels, le Backlog produit et présentation des parties prenantes SCRUM. Aussi, nous allons présenter l'architecture adoptée pour notre application.

Planifications et spécifications des besoins

Sommaire

2.1	Introduction	13
2.2	Analyse et spécification des besoins	13
2.2.1	Spécification des besoins fonctionnels :	13
2.2.2	Spécification des besoins non fonctionnels :	14
2.3	Identification et structuration de cas d'utilisation	14
2.3.1	Identifications des acteurs :	14
2.3.2	Diagramme de cas d'utilisation global :	14
2.4	Pilotage de projet avec SCRUM	15
2.4.1	Équipe et rôles :	15
2.4.2	Le Backlog du produit :	15
2.4.3	Structure et Découpage des sprints :	17
2.4.4	Diagramme de Gantt :	18
2.5	Environnement de travail	19
2.5.1	Environnement matériel :	19
2.5.2	Environnements logiciels :	19
2.5.3	Technologies et langages utilisées :	21
2.6	Architecture de l'application	22
2.6.1	Architecture physique :	22
2.6.2	Architecture logique :	23
2.7	Conclusion	24

2.1 Introduction

Dans ce chapitre, nous commencerons par analyser et préciser les besoins en expliquant les besoins fonctionnels et non fonctionnels de notre application. Ensuite, nous présenterons le diagramme de cas d'utilisation global en précisant les différents acteurs. Et enfin, nous détaillerons la gestion du projet avec la méthodologie Scrum en présentant l'architecture adoptée pour notre application.

2.2 Analyse et spécification des besoins

L'expression des besoins a pour objectif d'assurer une compréhension des besoins, des acteurs et de leurs exigences.

2.2.1 Spécification des besoins fonctionnels :

Les besoins fonctionnels nous permettent de présenter les actions fournies par le système et les attentes pour les applications futures.

Notre application doit répondre aux exigences suivantes :

Administrateur :

- Authentifié avec un email et mot de passe
- Gérer les comptes : Ajouter, Modifier, supprimer

Responsable :(SMSDA ou Hygiène) :

- Authentifié avec un email et mot de passe
- Remplir les check-lists
- Consulter les check-lists
- Choisir les check-lists.
- Sauvegarder les check-lists
- Imprimer les check-lists
- Consulter les scores des check-lists
- Consulter les tableaux de bord

2.2.2 Spécification des besoins non fonctionnels :

Ces besoins décrivent toutes les contraintes auxquelles est soumis le système pour sa réalisation et son bon fonctionnement :

- **Ergonomie de l'interface :** L'interface doit être conviviale et Facile à utiliser.
- **Maintenance :** Le code doit être clair pour permettre des évolutions ou améliorations futures.
- **Sécurité :** L'utilisateur de l'application doit être authentifié pour accéder différents tableaux de bord.
- **Fiabilité :** le système doit fonctionner de manière fiable et sécurisée en garantissant l'intégrité des données.
- **Performance :** l'application doit rester efficace quelles que soient les contraintes.
- **Responsive :** Compatible à toutes les résolutions.

2.3 Identification et structuration de cas d'utilisation

2.3.1 Identifications des acteurs :

Un acteur peut être une personne, un matériel ou un logiciel qui interagit avec le système pour exécuter une ou plusieurs fonctions liées au cas d'utilisation. Pour notre système, nous avons identifié les acteurs suivants :

- **L'administrateur :** L'acteur principal qui a le droit d'accès à tous.
- **Responsable :** l'utilisateur responsable de la vérification des check-lists. Il peut remplir les données.

2.3.2 Diagramme de cas d'utilisation global :

Les diagrammes de cas d'utilisation modélisent le comportement d'un système et permettent de capturer ses exigences. Ces diagrammes identifient aussi les interactions entre le système et ses acteurs.

La figure 2.1 présente le diagramme de cas d'utilisation global de notre application :

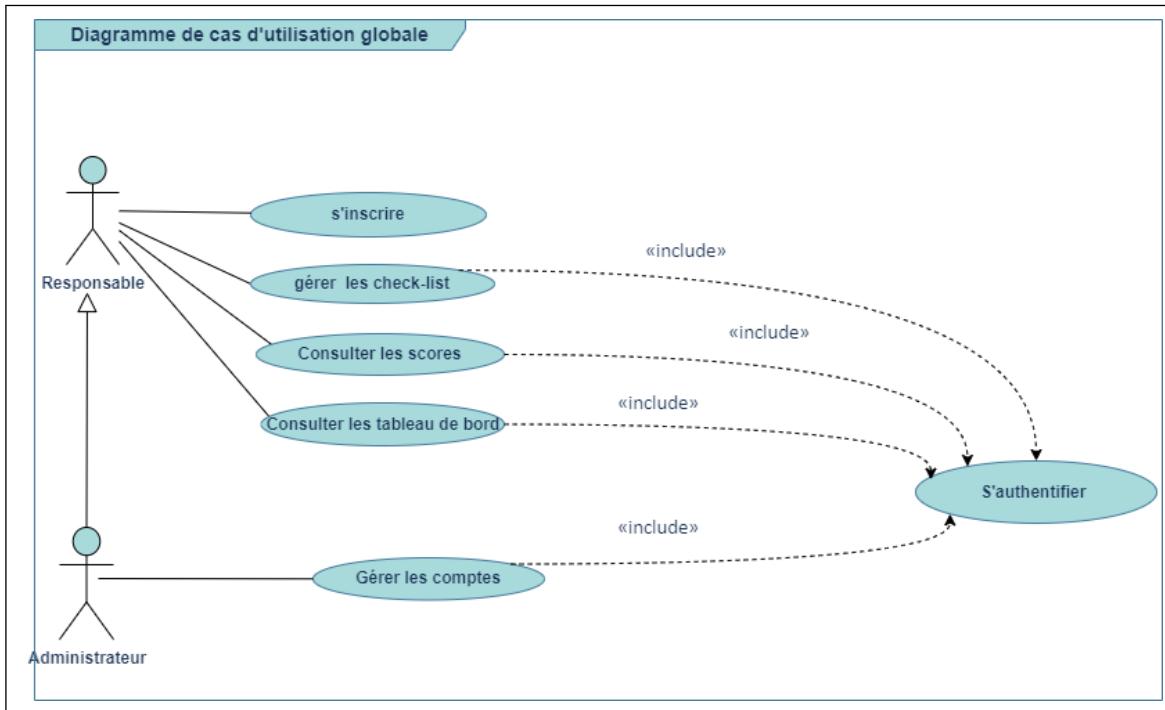


FIGURE 2.1 – Diagramme de cas d'utilisation global

L’administrateur aura toutes les possibilités qui ont été accordés aux responsables. Il peut par exemple gérer les check-lists les scores le tableau de bord. Il doit pouvoir vérifier que tout fonctionne bien Et toutes les fonctionnalités sont bien faites.

2.4 Pilotage de projet avec SCRUM

2.4.1 Équipe et rôles :

Scrum master : M. Hattab Mehdi

Product owner : Mme. Ameni koobi, M. Boukthir Hafedh

Team Member : Bahri Chiraz et Souissi Rania

2.4.2 Le Backlog du produit :

Pour réaliser un projet Scrum, il faut définir le produit Backlog qui est une liste priorisée des tâches.

Le tableau 2.1 résume les différentes fonctions à réaliser :

TABLE 2.1 – Backlog du Produit

Thème	ID	User story	Estimation
Authentification	1	En tant qu'administrateur ou responsable, je peux m'authentifier afin d'accéder à mon espace.	13 sp
Gérer les inscriptions	2	En tant qu'utilisateur, je peux créer un compte d'une manière sécurisé	8 sp
Gestion des comptes	3	En tant qu'administrateur, je peux gérer les comptes afin de modifier, ajouter et supprimer un compte dans les bases de données.	5 sp
Gestion des check-list	4.1	En tant que responsable, je peux gérer les check-lists afin de remplir une check-list.	21 sp
	4.2	En tant que responsable, je peux gérer les check-lists afin de consulter une check-list.	
	4.3	En tant que responsable, je peux gérer les check-lists afin de choisir une check-list.	
	4.4	En tant que responsable, je peux gérer les check-lists afin sauvegarder une check-list.	
	4.5	En tant que responsable hygiène ou responsable SMSDA, je peux gérer les check-lists afin d'imprimer une check-list.	
Gestion des scores	5	En tant que responsable, je peux gérer les scores de tous les check-lists afin de le consulter.	13 sp
Gestion du Tableau de bord	6	En tant que responsable, je peux gérer le tableau de bord de toutes les check-lists afin de le consulter.	13 sp

Story point (sp) : Un story point est une mesure abstraite de l'effort requis pour mettre en œuvre une user story. En termes simples, c'est un nombre qui indique à l'équipe le niveau de difficulté de l'histoire. La difficulté pourrait être liée aux complexités, aux risques et aux efforts impliqués.

2.4.3 Structure et Découpage des sprints :

Une structure de projet fournit le cadre dans lequel l'effort de développement sera accompli. Bien que simple et de haut niveau, il fournit des lignes directrices et un flux qui couvrent la topographie du paysage du développement local.

La figure 2.2 montre la structure de notre projet :

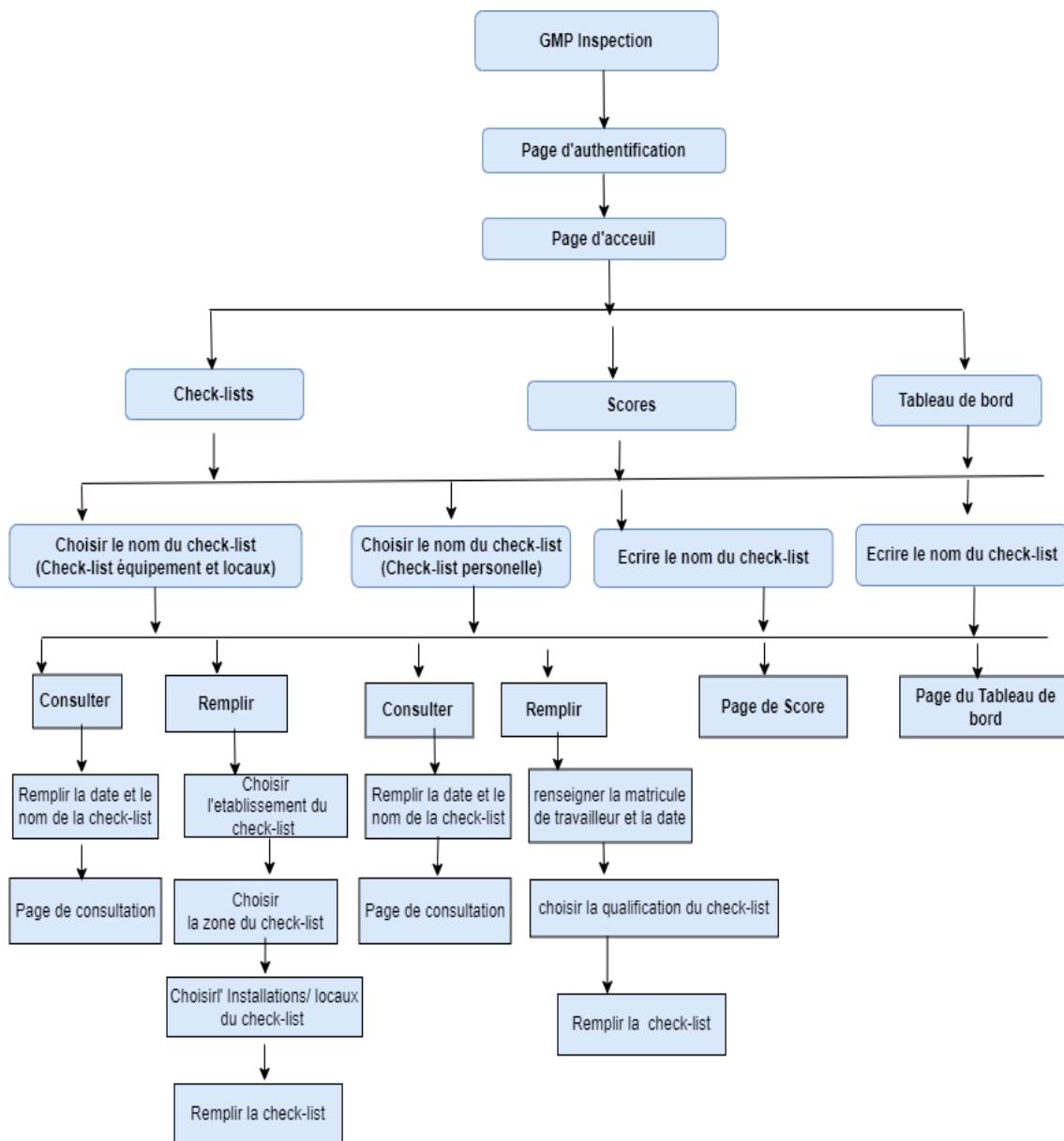


FIGURE 2.2 – Structure du projet

Après avoir extrait les différentes tâches à faire, nous avons discuté avec le client et réaliser le découpage des sprints selon leurs priorités comme il est représenté dans la figure 2 :

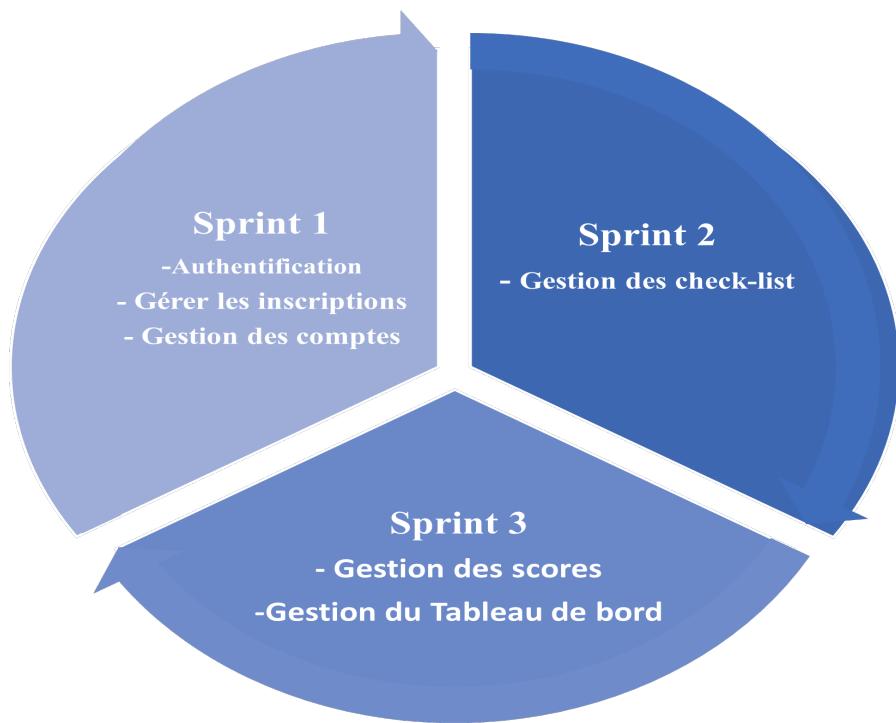


FIGURE 2.3 – Découpage du projet en sprints

2.4.4 Diagramme de Gantt :

Un diagramme de Gantt est un outil de gestion de projet qui aide à la planification et à l'ordonnancement de projets de toutes tailles, bien qu'il soit particulièrement utile pour simplifier des projets complexes.

La figure 2.4 montre le diagramme de Gantt de notre projet :

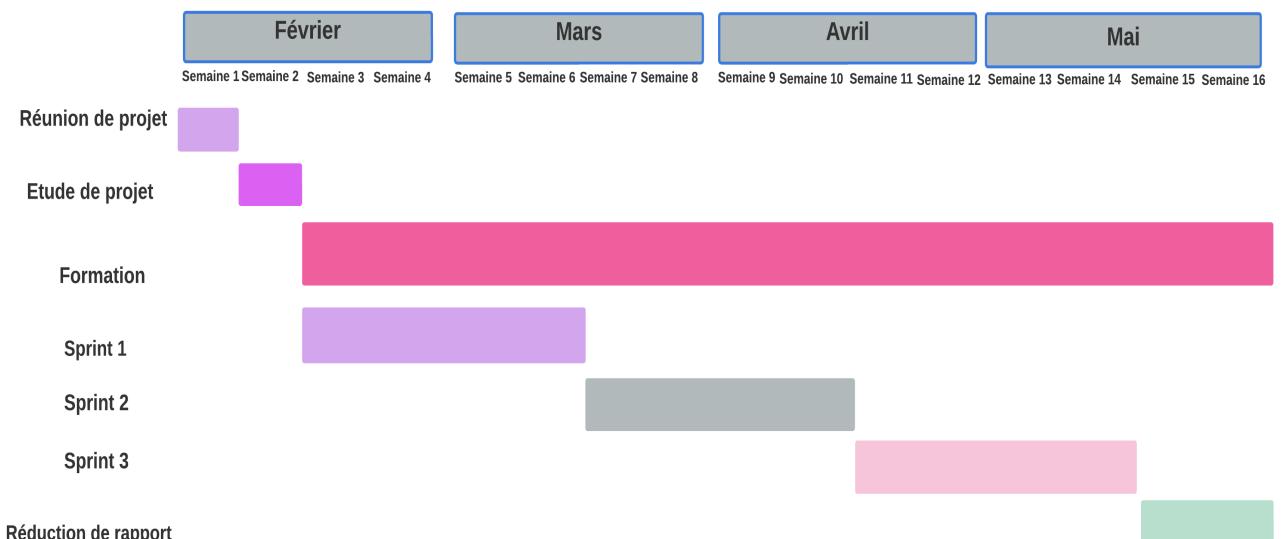


FIGURE 2.4 – Diagramme de Gantt

2.5 Environnement de travail

2.5.1 Environnement matériel :

Nous mentionnons dans le tableau 2.2 les caractéristiques des ordinateurs dans lesquels nous avons développé notre application :

TABLE 2.2 – Tableau comparatif des différentes API de cartographie

Pc Portable	Asuc	Lenovo
Propriétaire	Bahri Chiraz	Souissi Rania
Ram	8 Go	12 Go
Processeur	Core I5	Core I3
Disque Dur	1 To	1 To
Système d'exploitation	Windows 10 Professionnel	Windows 10 Professionnel

2.5.2 Environnements logiciels :

Dans cette partie, nous allons présenter et détailler les différents outils informatiques et langages utilisé dans notre travail :

- **Visual Studio Code**

Visuel studio code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS.

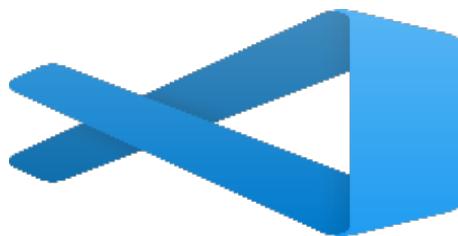


FIGURE 2.5 – Logo Visual Studio Code

- **Draw.io**

Draw.io est un logiciel propriétaire permettant de créer des diagrammes et des graphiques.



FIGURE 2.6 – Logo Drawo.io

- **Postman**

Postman est une application utilisée pour les tests d’API. Il s’agit d’un client HTTP qui teste les requêtes HTTP, en utilisant une interface utilisateur graphique, à travers laquelle nous obtenons différents types de réponses qui doivent ensuite être validées.



FIGURE 2.7 – Logo Postman

- **GitHub**

Il vise à coordonner le travail de programmeur, mais peut être utilisé pour suivre modifications de tous les fichiers.



FIGURE 2.8 – Logo GitHub

- **LaTeX**

Overleaf est un outil d’écriture et de publication collaboratif LaTeX et Rich Text en ligne qui rend l’ensemble du processus d’écriture, d’édition et de publication de documents scientifiques beaucoup plus rapide et plus facile.

LaTeX, logiciel utilisé pour la composition de documents techniques. LaTeX est un logiciel libre créé en 1985 par l’informaticien américain Leslie Lamport en complément du système de

composition TeX. LaTeX a été créé pour faciliter la production de livres et d'articles à usage général dans TeX.



FIGURE 2.9 – Logo Overleaf

2.5.3 Technologies et langages utilisées :

- **Flutter**

Flutter est un kit de développement logiciel (SDK) d'interface utilisateur open-source créé par Google. Il est utilisé pour développer des applications pour Android, iOS, Linux, Mac, Google, Fuchsia et le web à partir d'une seule base de code. [5]



FIGURE 2.10 – Logo Flutter

- **MongoDB**

MongoDB est un système de gestion de base de données NO SQL. Les données sont modélisées sous forme de document sous un style JSON.



FIGURE 2.11 – Logo MongoDB

- **Node.js**

Node.js est un environnement d'exécution JavaScript multiplateforme open source, créé par Ryan Dahl. Il permet à l'infrastructure de créer et d'exécuter du code JavaScript côté serveur.

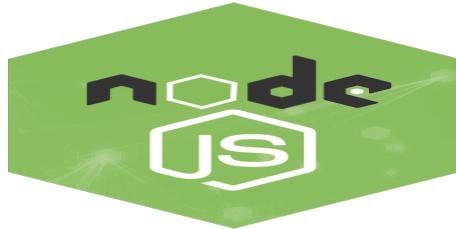


FIGURE 2.12 – Logo Node.js

- **Dart**

Dart est un langage de programmation Open Source côté client. Il est développé par Google et peut également être utilisé pour créer des applications mobiles, de bureau, de serveur et web.



FIGURE 2.13 – Logo Dart

2.6 Architecture de l'application

2.6.1 Architecture physique :

L'architecture d'un système est un modèle conceptuel qui définit son éléments et leurs relations et décrire leurs propriétés. le choix L'architecture est très sensible dans la conception des projets, car elle affecte les performances de l'application et sa maintenance et ses mises à jour, son cycle de vie. La figure ci-dessous montre l'architecture de notre application :

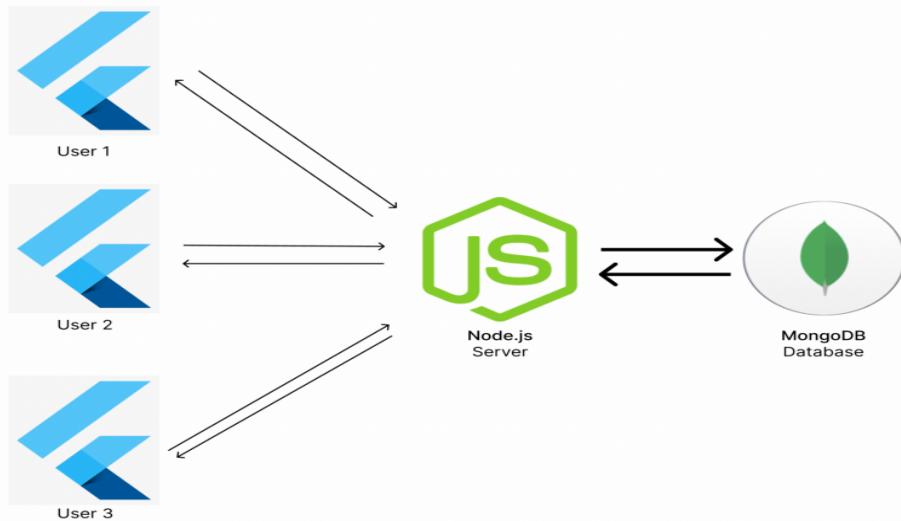


FIGURE 2.14 – l’architecture de l’application

C'est ainsi que les données sont transmises via notre application. Donc, disons que l'utilisateur 1 se connecte, ce qui signifie que les données vont à Node.js, et si tout est correct, il obtiendra les données de MongoDB. Lorsque cela sera fait, Node.js renverra un message à notre utilisateur.

2.6.2 Architecture logique :

MVC est un modèle architectural, ce qui signifie qu'il régit l'ensemble de l'architecture des applications. Même s'il est souvent connu sous le nom de modèle de conception, nous pouvons nous tromper si nous ne le référons qu'en tant que modèle de conception, car les modèles de conception sont utilisés pour résoudre un problème technique spécifique, alors que le modèle d'architecture est utilisé pour résoudre des problèmes architecturaux. Il affecte donc l'ensemble architecture de notre application.

Il a trois composants principaux :

- **Modèle :** Le modèle est en fait connecté à la base de données, donc tout ce que vous faites avec les données. L'ajout ou la récupération de données se fait dans le composant de modèle.
- **Vue :** La représentation des données est effectuée par le composant de vue.
- **Contrôleur :** le contrôleur est le composant qui permet l'interconnexion entre les vues et le modèle, il agit donc comme un intermédiaire.

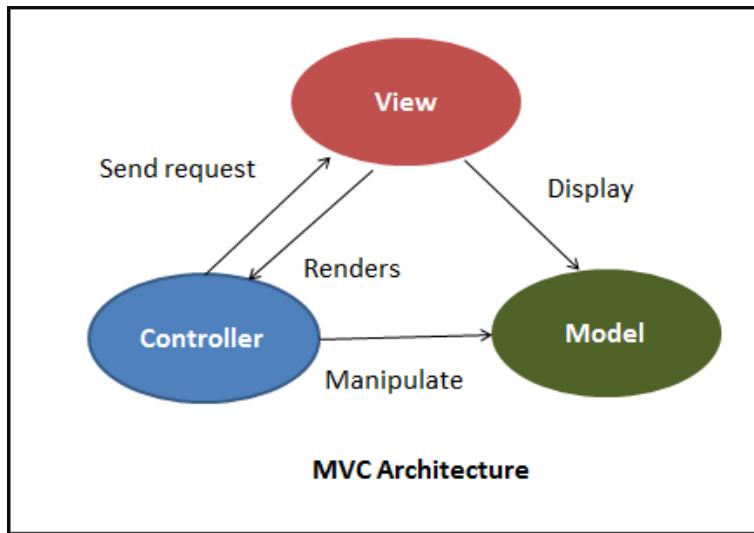


FIGURE 2.15 – Schéma MVC

2.7 Conclusion

Durant ce chapitre, nous avons identifié les exigences fonctionnelles et non fonctionnelles du système, ainsi que les principaux acteurs et leurs rôles. Par la suite, nous avons identifié le Backlog produit et les planifications des sprints. Nous avons terminé le chapitre par l'explication des architectures de notre application.

Dans le chapitre suivant, nous commençons le développement du Sprint 1.

Sprint 1 : Authentification, Gérer les inscriptions et Gestion des comptes

Sommaire

3.1	Introduction	26
3.2	Backlog du sprint 1	26
3.3	Spécifications fonctionnelles	27
3.3.1	Classification des cas d'utilisation par acteur :	27
3.3.2	Diagramme des cas d'utilisation :	27
3.4	Conception	30
3.4.1	Diagramme de séquence :	30
3.4.2	Diagramme de classes global du premier Sprint :	31
3.5	Réalisation	32
3.5.1	Interface d'authentification :	32
3.5.2	Interface d'inscription :	34
3.5.3	Interface ajouter, modifier, supprimer un compte dans la base de données :	36
3.5.4	Test fonctionnel	36
3.5.5	Logo de l'application :	37
3.6	Conclusion	37

3.1 Introduction

Après avoir spécifié les différents besoins de l'application et le cas d'utilisation globale dans le chapitre précédent, ce chapitre nous présenterons la partie d'authentification et gérer les inscriptions, gestion des comptes avec ses diagrammes en suivant le Framework scrum.

3.2 Backlog du sprint 1

Pour bien réaliser les tâches on a effectué le tableau ci-dessous qui représente le Backlog du sprint 1 :

TABLE 3.1 – Backlog du sprint 1.

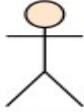
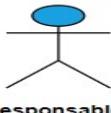
Sprint1			
ID	User Story	ID tâche	Tâche
Authentification			
1	En tant qu'utilisateur ou Administrateur je peux m'authentifier afin d'accéder à l'application.	1.1	Réaliser un diagramme de cas d'utilisation un diagramme de séquence . de la fonctionnalité « S'authentifier » .
		1.2	Développer le cas « S'authentifier »
		1.3	Tester « S'authentifier »
Gérer les inscriptions			
2	En tant qu'utilisateur je peux Créer un compte d'une manière sécurisé	2.1	Réaliser un diagramme de cas d'utilisation, de séquence de la fonctionnalité « Créer un compte »
		2.2	Développer le cas « créer un compte ».
		2.3	Tester « créer un compte »
Gestion des comptes			
3	En tant qu'administrateur, je peux gérer les comptes afin de les ajouter dans les bases de données.	3.1	À travers MongoDB Compass ou Atlas, l'administrateur peut ajouter les comptes
		3.2	Tester le cas « ajouter un compte »
4	En tant qu'administrateur, je peux gérer les comptes afin de les modifier dans les bases de données.	4.1	À travers MongoDB Compass ou Atlas, l'administrateur peut modifier les comptes
		4.2	Tester le cas « modifier un compte »
5	En tant qu'administrateur je peux gérer les comptes afin de les supprimer dans les bases de données.	5.1	À travers MongoDB Compass ou Atlas, l'administrateur peut supprimer les comptes
		5.2	Tester le cas « supprimer un compte »

3.3 Spécifications fonctionnelles

Dans cette partie, nous présentons le diagramme de cas d'utilisations puis une Description textuelle pour chacun d'entre eux.

3.3.1 Classification des cas d'utilisation par acteur :

TABLE 3.2 – *Classification des cas d'utilisations par acteur*

Acteurs	Cas d'utilisation
 Administrateur	— S'authentifier — Gérer les comptes
 Responsable	— S'inscrire

3.3.2 Diagramme des cas d'utilisation :

- Raffinement du cas d'utilisation "Authentification" :

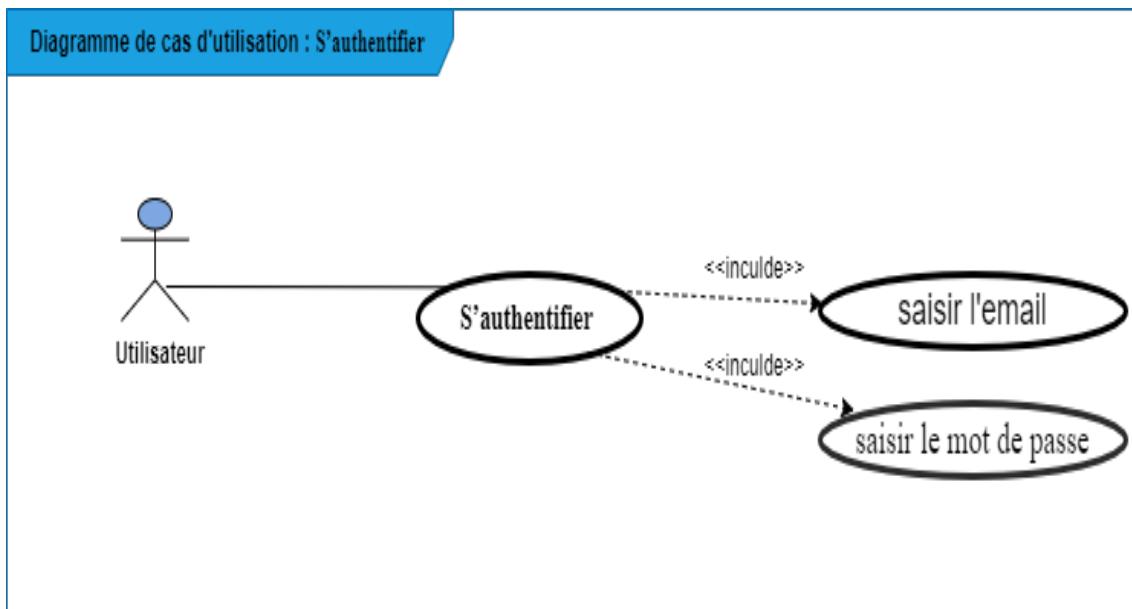


FIGURE 3.1 – Raffinement de cas d'utilisation « S'authentifier »

Description textuelle du cas d'utilisation : « s'authentifier » :

TABLE 3.3 – Description textuelle du cas « S'authentifier »

Acteur	L'utilisateur
Objectif	Permet aux Utilisateurs d'application d'accéder à son espace .
Pré-condition	- L'utilisateur doit exister dans la base de données et doit obligatoirement être connecté à Internet
Post-condition	Utilisateur Authentifié
Scénario nominatif	<ul style="list-style-type: none"> — L'utilisateur saisit son email et son mot de passe., — L'utilisateur clique sur le bouton "Connexion" pour confirmer. — Vérifier si le champ est vide et donner un message d'erreur, — Les paramètres d'accès sont envoyés vers la base de données pour la vérification des données, — L'interface utilisateur apparaîtra.
Scénario alternatif	En cas d'échec un message d'erreur s'afficher

- Raffinement du cas d'utilisation « Gérer les inscriptions » :

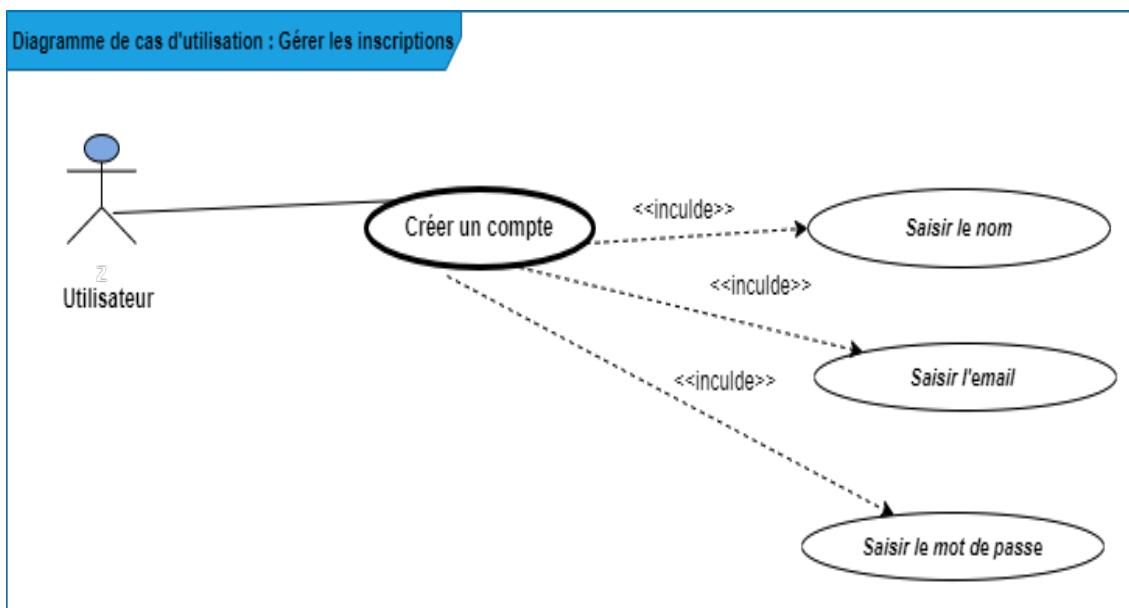


FIGURE 3.2 – Raffinement de cas d'utilisation « Gérer les inscriptions »

Description textuelle du cas d'utilisation : « créer un compte » :

TABLE 3.4 – Description textuelle du cas d'utilisation « créer un compte »

Acteur	Utilisateurs
Objectif	L'utilisateur doit créer un compte pour accéder à l'application.
Pré-condition	- Tous les champs doivent être remplis et corrects.
Post-condition	Utilisateur Authentifié
Scénario nominatif	<ul style="list-style-type: none"> — L'utilisateur remplit tous les champs, — Il confirme en cliquant sur le bouton « créer un compte », — Le système envoie les coordonnées à la base de données pour vérification, — Inscription effectuée.
Scénario alternatif	Le système affiche un message d'erreur indiquant qu'il y a un champ vide.

3.4 Conception

Dans cette partie, nous allons réaliser les diagrammes de séquences détaillés et le diagramme de classe global.

3.4.1 Diagramme de séquence :

- Diagramme de séquence du cas « s'authentifier » :

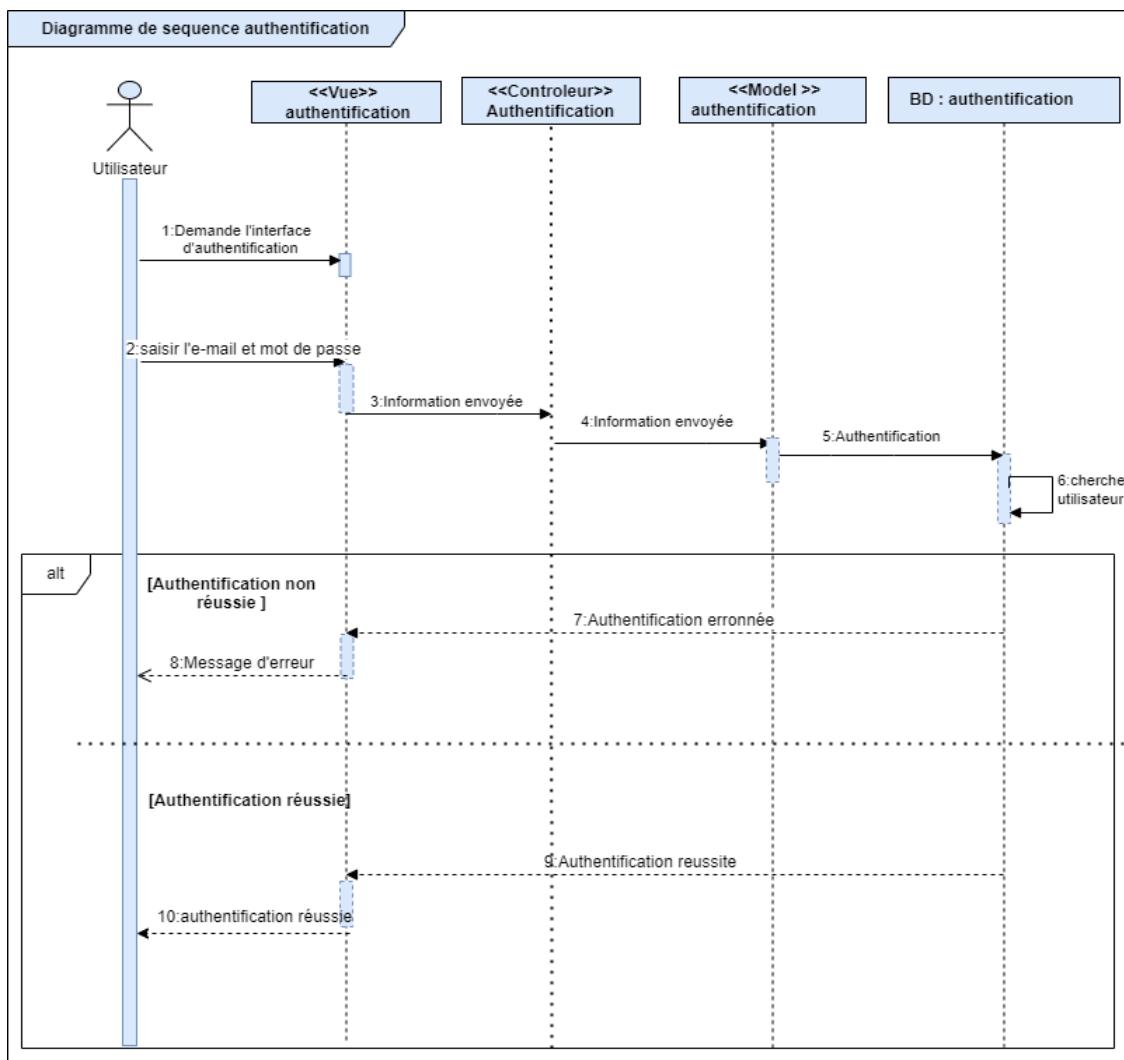


FIGURE 3.3 – diagramme de séquence « s'authentifier »

SPRINT 1 : AUTHENTIFICATION, GÉRER LES INSCRIPTIONS ET GESTION DES COMPTES

- Diagramme de séquence du cas « Crée un compte » :

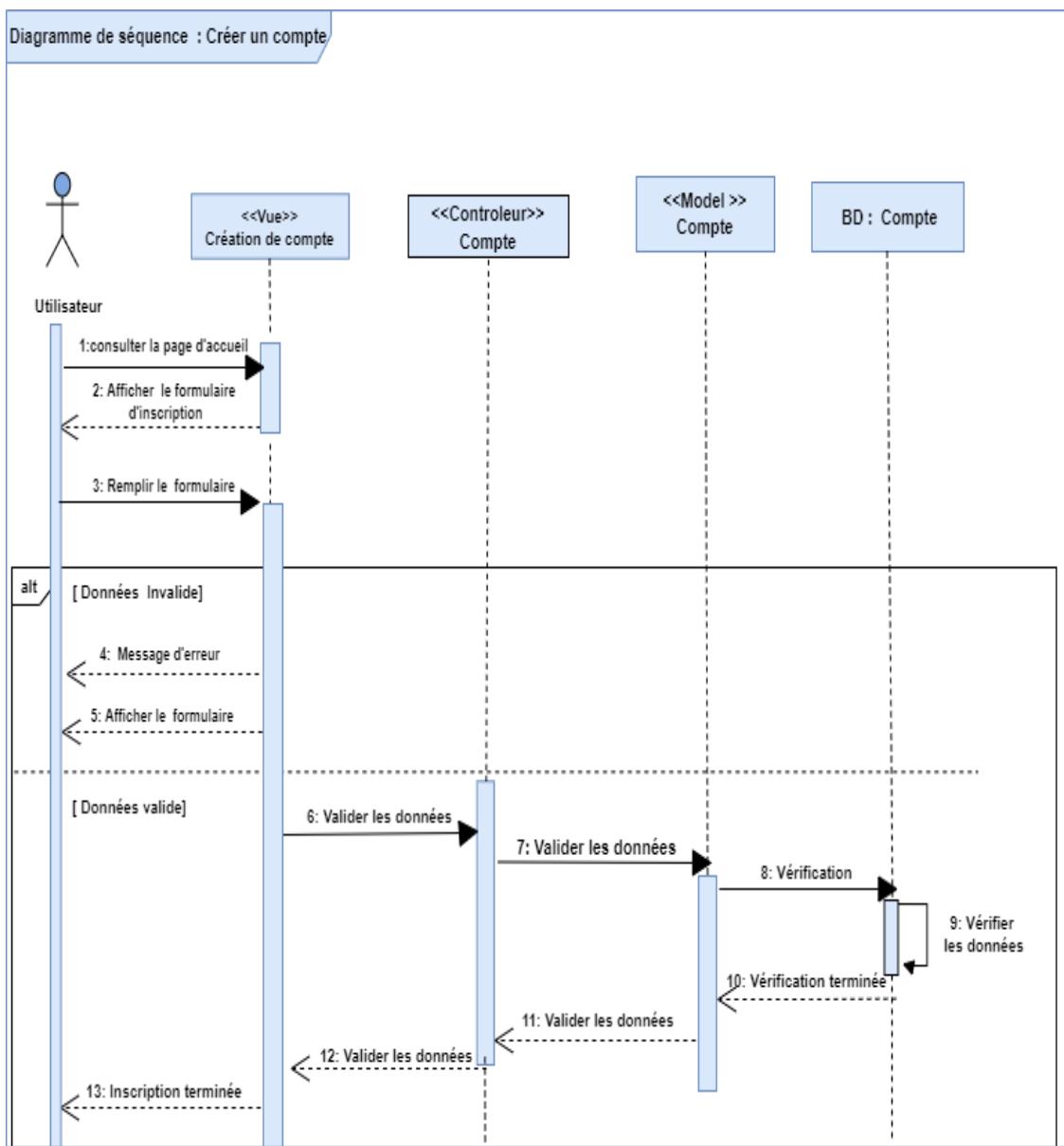


FIGURE 3.4 – Diagramme de séquence « Crée un compte »

3.4.2 Diagramme de classes global du premier Sprint :

Le diagramme de classes est une description statique du système centrée sur le concept d’association et de classe. Une classe représente un ensemble d’objets qui ont des propriétés similaires et des comportements communs décrivant en termes d’attributs et d’opérations.

Dans cette section, nous allons présenter le diagramme de classes global du premier sprint :

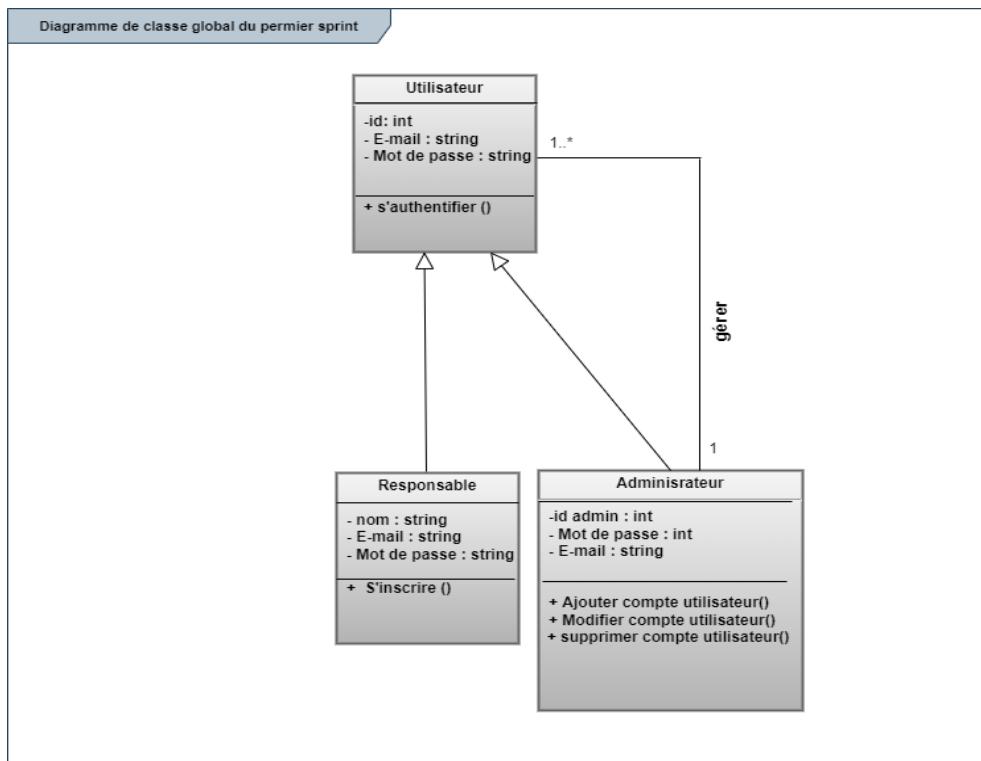


FIGURE 3.5 – Diagramme de classes global du premier sprint

3.5 Réalisation

Dans cette partie, nous allons exposer quelques scénarios d'exécution à travers des captures d'écran.

3.5.1 Interface d'authentification :

L'interface d'authentification permet l'utilisateur d'accéder à l'application après la validation de son email et mot de passe.

SPRINT 1 : AUTHENTIFICATION, GÉRER LES INSCRIPTIONS ET GESTION DES COMPTE

- Web

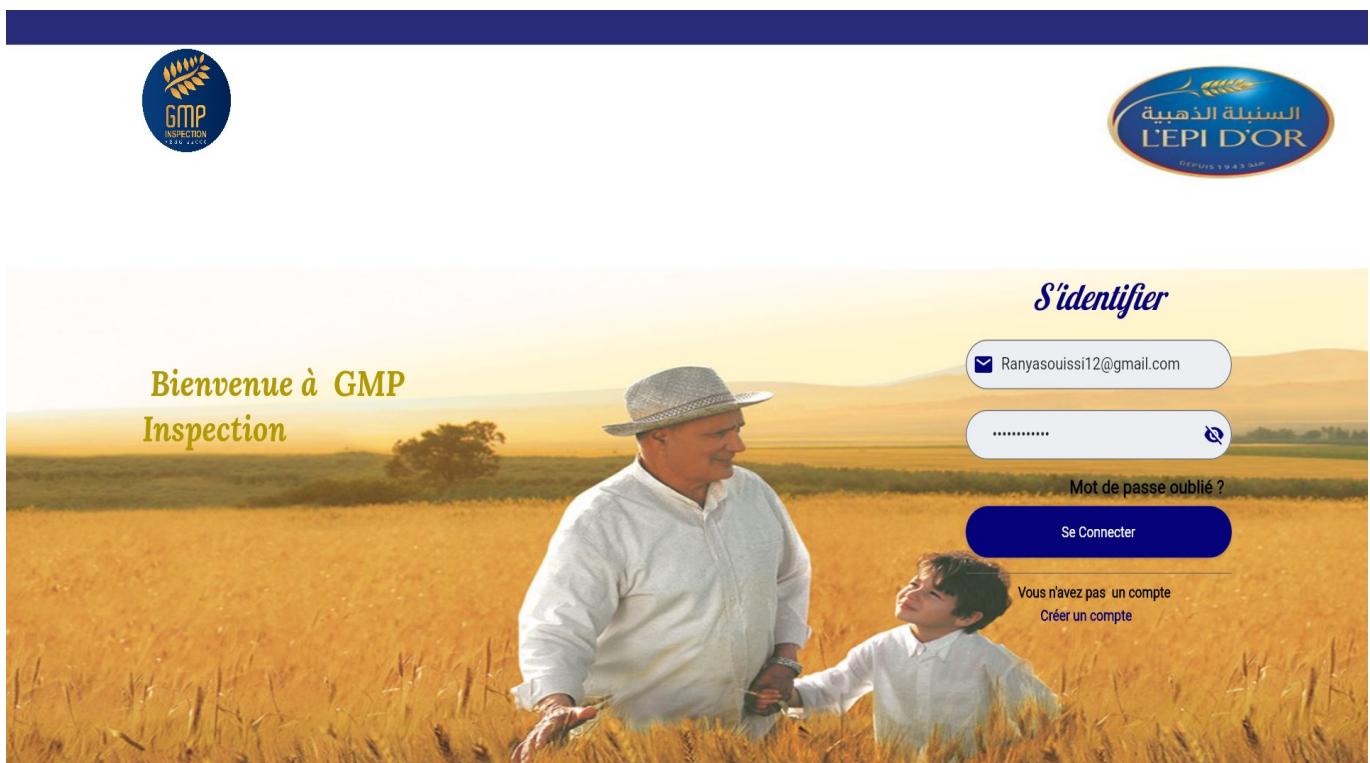


FIGURE 3.6 – interface d’authentification web

- Mobile



FIGURE 3.7 – interface d'authentification mobile

3.5.2 Interface d'inscription :

L'interface d'inscription permet à l'utilisateur de créer un compte.



FIGURE 3.8 – interface d'inscription

3.5.3 Interface ajouter, modifier, supprimer un compte dans la base de données :

```

_id: ObjectId('627e7c02535c8209aab8b50b')
username: "chiraz"
email: "chirazbahri@gmail.com"
password: "$2b$10$GFeBcgOis7ijINly0aW8.kRXIcwff5frvVIULzf2Gi6nJ7rs9gh0"
isAdmin: false
createdAt: 2022-05-13T15:40:50.343+00:00
updatedAt: 2022-05-13T15:40:50.343+00:00
__v: 0

_id: ObjectId('627e7c3e595c8209aab8b50f')
username: "ranya"
email: "ranyasouissi12@gmail.com"
password: "$2b$10$yz3owSzdrZw3BL2afFWWadeHUI.g6/cYmgSt22yTYB5fBWX5r05g6."
isAdmin: false
createdAt: 2022-05-13T15:41:50.394+00:00
updatedAt: 2022-05-13T15:41:50.394+00:00
__v: 0

```

FIGURE 3.9 – interface ajouter, modifier et supprimer

3.5.4 Test fonctionnel

la pratique de test représente la dernière étape du cycle de développement d'un sprint. Ils permettent de vérifier les résultats obtenus afin d'assurer et de garantir une bonne qualité de développement.

```

POST /api/auth/login
{
  "username": "ranya",
  "password": "14000"
}

```

```

{
  "username": "ranya",
  "isAdmin": false,
  "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZC16IjYyN2U3YzN1NTM1YzgyMDlhYWI4YjUwZiIsImIzQWRtaW10mZhbHN1LCJpYXQiOjE2NTUzODIzMTR9.GGwUJwz5f0efk1g8PAIBCbwy6zJx9dda5Vv-DhXPU"
}

```

FIGURE 3.10 – Test d'authentification

SPRINT 1 : AUTHENTIFICATION, GÉRER LES INSCRIPTIONS ET GESTION DES COMPTES

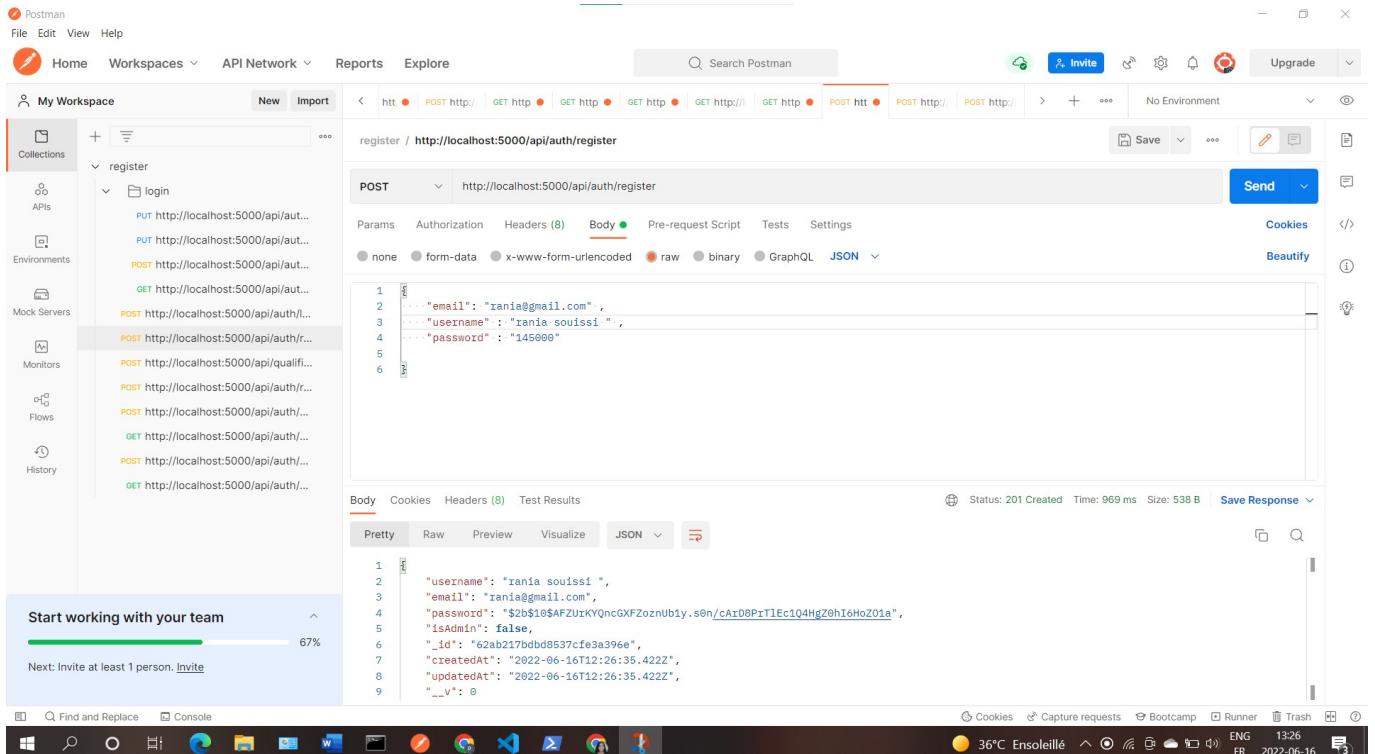


FIGURE 3.11 – Test créer un compte

3.5.5 Logo de l'application :



FIGURE 3.12 – logo de l'application

3.6 Conclusion

Pendant ce chapitre, nous avons présenté les différentes phases de la conception de notre application, nous avons commencé par le Backlog de sprint, la conception puis la partie réalisation. La partie suivante sera consacrée à la réalisation de notre deuxième sprint intitulé « Gestion des check-lists ».

Sprint 2 : Gestion des check-lists

Sommaire

4.1	Introduction	39
4.2	Backlog du sprint 2	39
4.3	Spécifications fonctionnelles	40
4.3.1	Classification des cas d'utilisation par acteur :	41
4.3.2	Diagramme des cas d'utilisation :	41
4.4	Conception	43
4.4.1	Diagramme de séquence :	43
4.4.2	Diagramme de classes global du deuxième Sprint :	46
4.5	Réalisation	46
4.5.1	Interface d'accueil :	46
4.5.2	Interface choisir check-list :	47
4.5.3	Interface remplir check-list :	49
4.5.4	Interface consulter check-list :	51
4.5.5	Interface imprimer check-list :	52
4.5.6	Test fonctionnel	53
4.6	Conclusion	54

4.1 Introduction

Dans le chapitre précédent, nous avons présenté notre premier sprint « **Authentification, Gérer les inscriptions et Gestion des comptes** ».

Dans ce chapitre, nous réaliserons le deuxième sprint « **Gestion des check-lists** », ce sprint présente la partie la plus importante de notre application.

4.2 Backlog du sprint 2

Pour bien réaliser les tâches, on a effectué le tableau ci-dessous qui représente le Backlog du sprint 2 :

TABLE 4.1 – Backlog du sprint 2.

Sprint 2			
ID	User Story	ID tâche	Tâche
Gérer les check-lists			
1	En tant que responsable, je peux gérer les check-lists afin de remplir une check-list	1.1	Réaliser un diagramme de cas d'utilisation, de séquence de la fonctionnalité « Remplir une check-list »
		1.2	Développer le cas « remplir une check-list ».
		1.3	Tester « remplir une check-list »
2	En tant que responsable, je peux gérer les check-lists afin de consulter une check-list	2.1	Réaliser un diagramme de cas d'utilisation, de séquence de la fonctionnalité « Consulter une check-list »
		2.2	Développer le cas « Consulter une check-list ».
		2.3	Tester le cas « Consulter une check-list »

Sprint2			
ID	User Story	ID tâche	Tâche
Gérer les check-list			
3	En tant que responsable, je peux gérer les check-lists afin de choisir une check-list.	3.1	Réaliser un diagramme de cas d'utilisation, de séquence de la fonctionnalité « Choisir une check-list »
		3.2	Développer le cas « Choisir une check-list»
		3.3	Tester le cas « Choisir une check-list »
4	En tant que responsable, je peux gérer les check-lists afin de sauvegarder une check-list.	3.1	Réaliser un diagramme de cas d'utilisation, de séquence de la fonctionnalité « Sauvegarder une check-list »
		4.2	Développer le cas « Sauvegarder une check-list»
		4.3	Tester le cas « Sauvegarder un compte »
5	En tant que responsable, je peux gérer les check-lists afin de imprimer une check-list.	5.1	Réaliser un diagramme de cas d'utilisation, de séquence de la fonctionnalité « Imprimer une check-list »
		5.2	Développer le cas « Imprimer une check-list»
		5.3	Tester le cas « Imprimer une check-list»

4.3 Spécifications fonctionnelles

Dans cette partie, nous présentons le diagramme de cas d'utilisations puis une Description textuelle pour chacun d'entre eux.

4.3.1 Classification des cas d'utilisation par acteur :

TABLE 4.2 – *Classification des cas d'utilisations par acteur*

Acteurs	Cas d'utilisation
 Responsable	— Gérer les check-lists

4.3.2 Diagramme des cas d'utilisation :

- Raffinement du cas d'utilisation "Gérer les check-lists" :

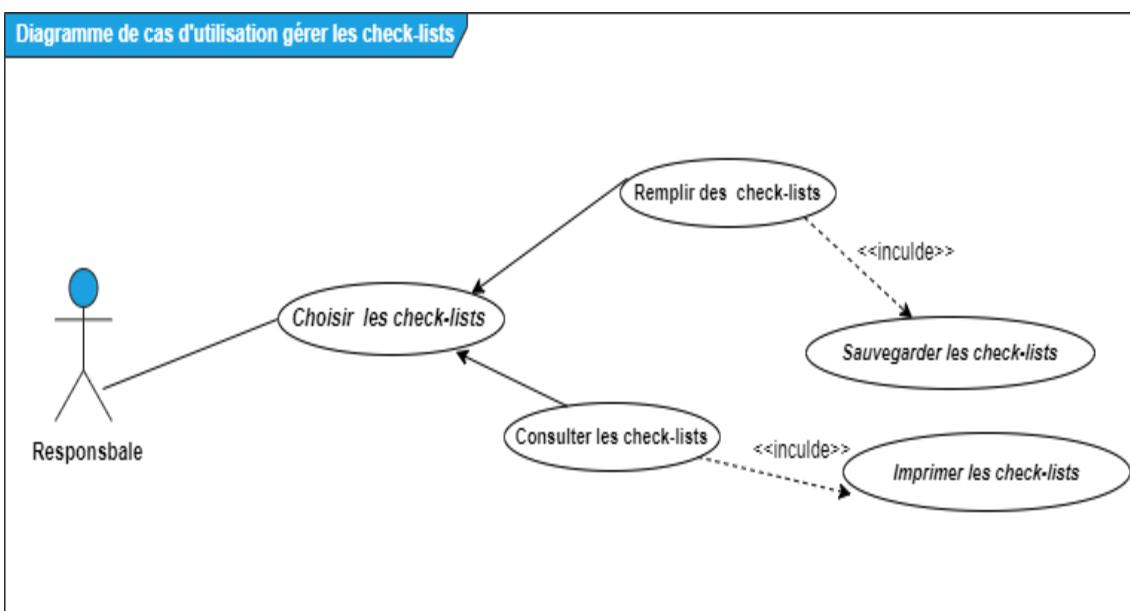


FIGURE 4.1 – Raffinement de cas d'utilisation « Gérer les check-lists »

Description textuelle du cas d'utilisation « remplir et sauvegarder les check-lists » :

TABLE 4.3 – Description textuelle de cas d'utilisation « remplir et sauvegarder »

Acteur	Responsable
Objectif	Permet à le responsable de gérer les check-lists.
Pré-condition	Le responsable doivent être authentifié
Post-condition	Opération terminée et validée
Scénario nominatif	<ul style="list-style-type: none"> — Le responsable ouvre l'interface check-list, — Le responsable clique sur le bouton « remplir une check-list », — une liste des check-lists s'affichera, — Le responsable choisir la check-list, — L'interface de l'utilisateur s'affichera
Scénario alternatif	En cas d'échec, un message d'erreur s'affiche

Description textuelle du cas d'utilisation « Consulter et imprimer les check-lists» :

TABLE 4.4 – Description textuelle de cas d'utilisation« consulter et imprimer »

Acteur	Responsable
Objectif	Permet au responsable de gérer les check-lists.
Pré-condition	Le responsable doivent être authentifié
Post-condition	Opération terminée et validée
Scénario nominatif	<ul style="list-style-type: none"> — Le responsable ouvre l'interface check-list, — Le responsable clique sur le bouton « consulter une check-list », — Le responsable remplir la date et le nom de check-list à consulter, — une liste des check-lists s'affichera, — Le responsable clique sur le Bouton imprimer, — impression avec succès.
Scénario alternatif	En cas d'échec, un message d'erreur s'affiche

4.4 Conception

Dans cette partie, nous allons réaliser les diagrammes de séquences détaillés et le diagramme de classe global.

4.4.1 Diagramme de séquence :

- Diagramme de séquence du cas « Remplir et sauvegarder check-list » :

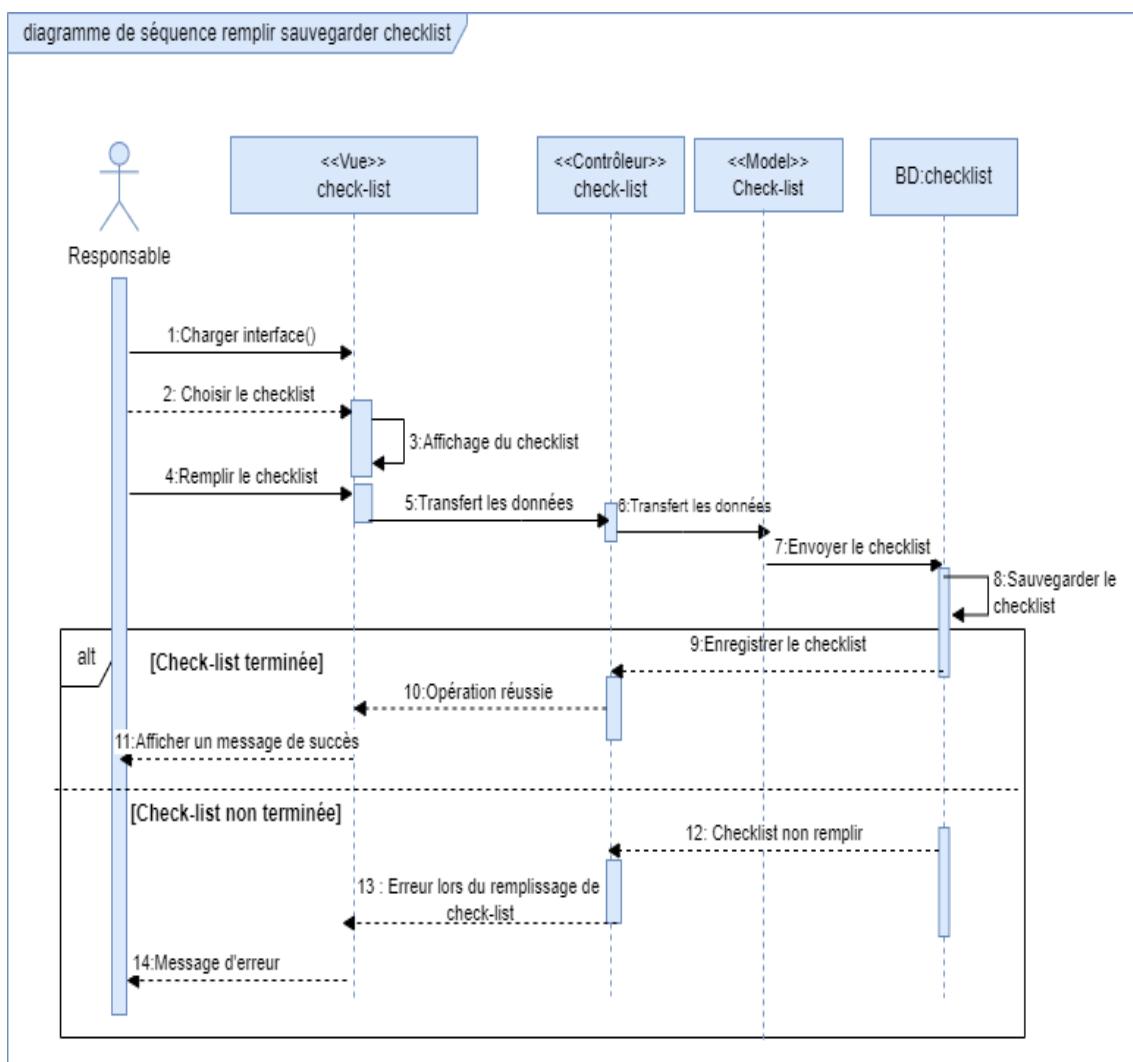


FIGURE 4.2 – Diagramme de séquence « remplir et sauvegarder check-list »

- Diagramme de séquence du cas « Consulter check-list » :

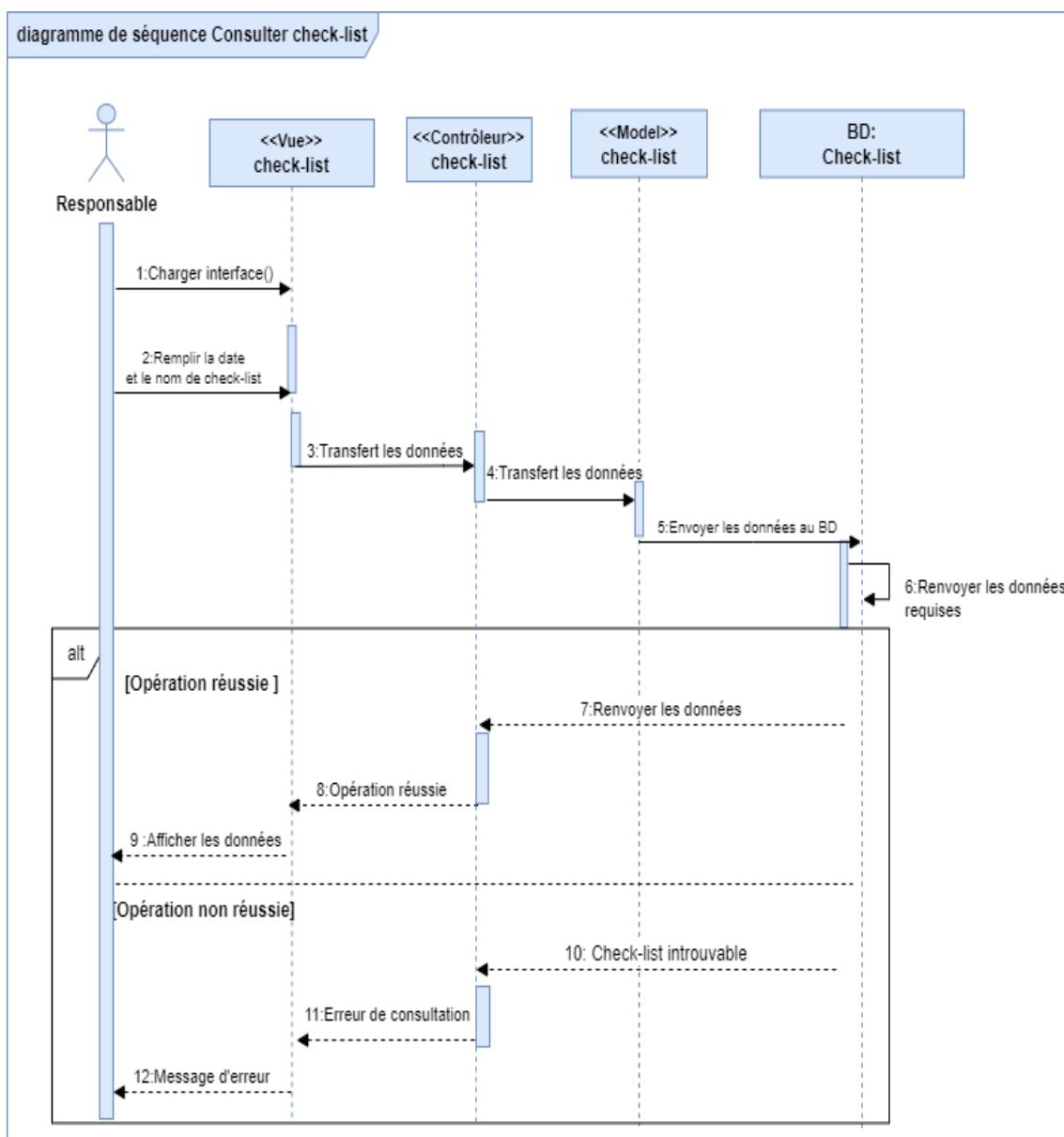


FIGURE 4.3 – Diagramme de séquence « consulter check-list »

- Diagramme de séquence du cas « Imprimer check-list » :

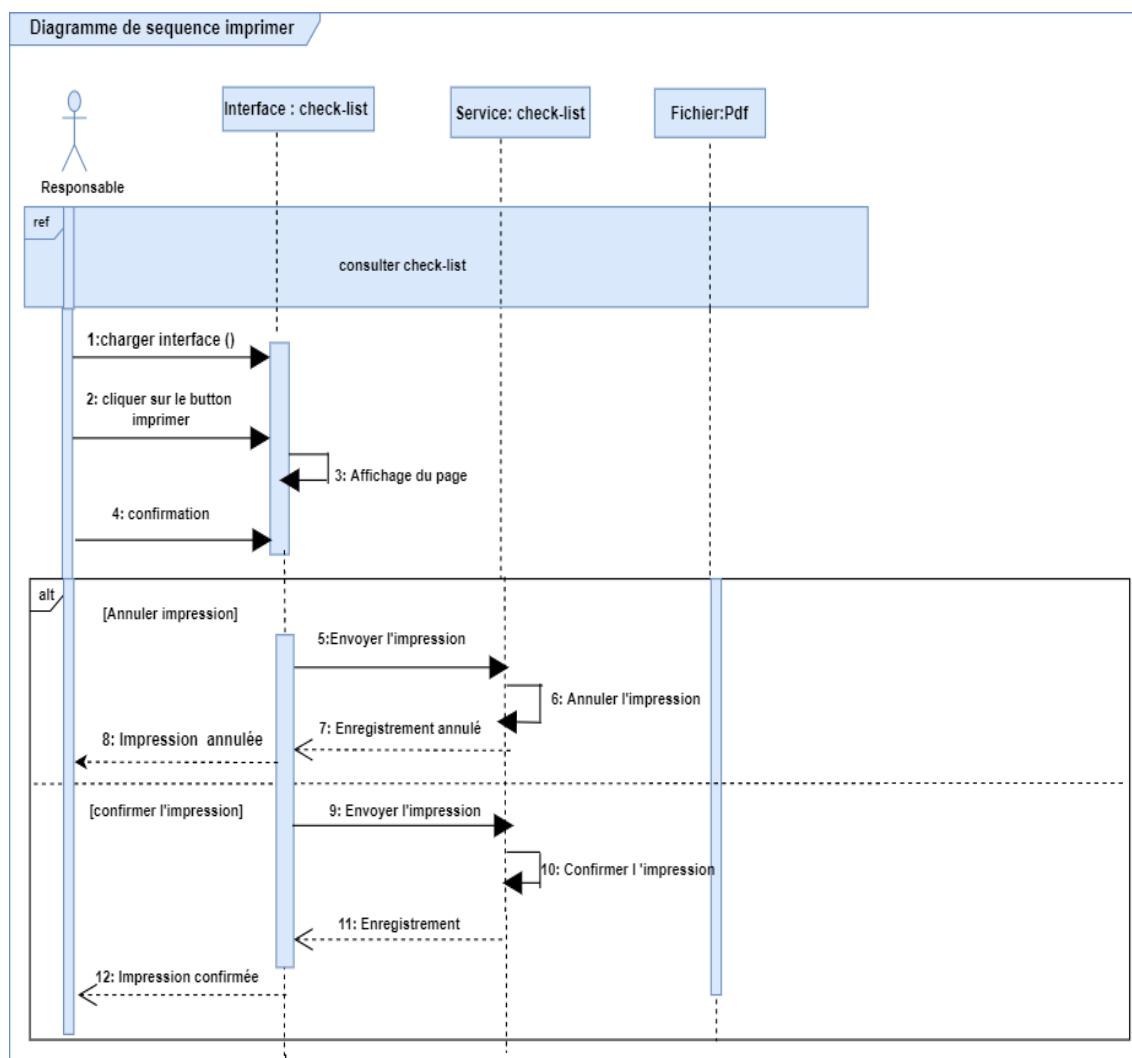


FIGURE 4.4 – Diagramme de séquence « Imprimer check-list »

4.4.2 Diagramme de classes global du deuxième Sprint :

La figure ci-dessous décrire le diagramme de classe global relatif au deuxième sprint

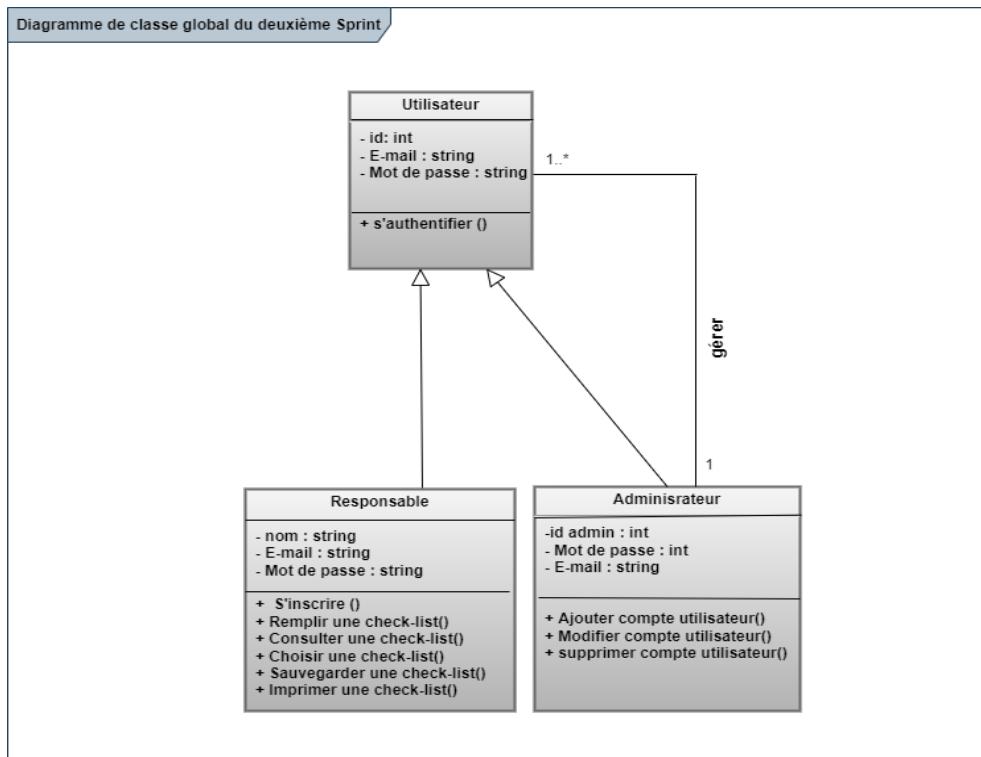


FIGURE 4.5 – Diagramme de classes global du deuxième Sprint

4.5 Réalisation

4.5.1 Interface d'accueil :

Après la validation de l'authentification, une interface sera affichée au responsable qui comprend la check-list, les scores, les tableaux de bord



FIGURE 4.6 – Interface d'accueil

4.5.2 Interface choisir check-list :

Cette interface permet au responsable de choisir la check-list à remplir

- Interface pour les Check-lists hygiène des équipements et locaux :



FIGURE 4.7 – Interface pour les Check-lists hygiène des équipements et locaux

- Interface pour la check-list personnelle :

Avant de choisir la check-list, le responsable doit renseigner la matricule de travailleur et la date.



FIGURE 4.8 – Interface matricule



FIGURE 4.9 – Interface check-list personnelle

4.5.3 Interface remplir check-list :

Cette interface permet au responsable de remplir la check-list :

- Le numéro 1 sera envoyé à la base de données, lorsqu'on clique sur le bouton vert, contrairement lorsqu'on clique sur bouton rouge, le zéro sera envoyé à la base de données.



FIGURE 4.10 – Interface remplir check-list

4.5.4 Interface consulter check-list :

Cette interface permet au responsable d'accéder aux check-lists déjà remplies après avoir choisi le nom et la date de la check-list :



FIGURE 4.11 – Interface choisir nom et date de check-list

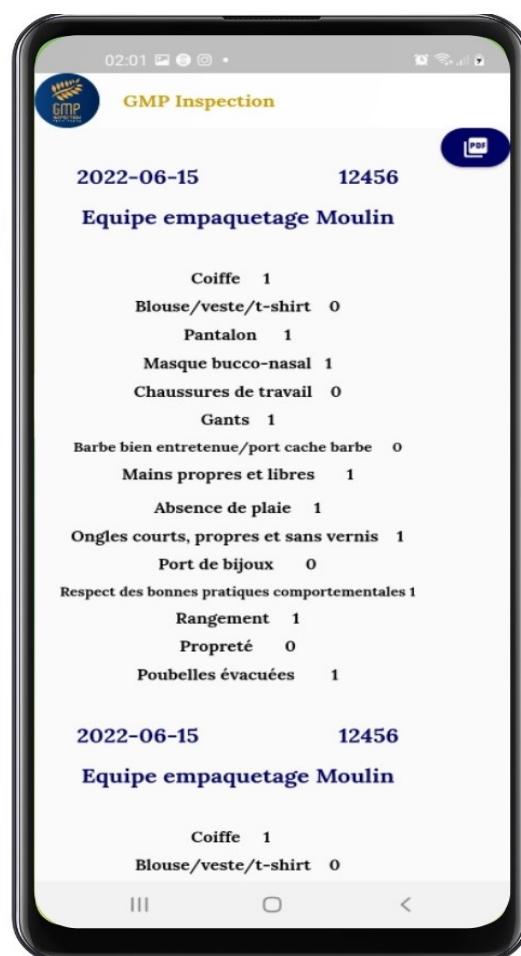


FIGURE 4.12 – Interface consulter check-list

4.5.5 Interface imprimer check-list :

Cette interface permet au responsable d'imprimer la check-list à consulter :

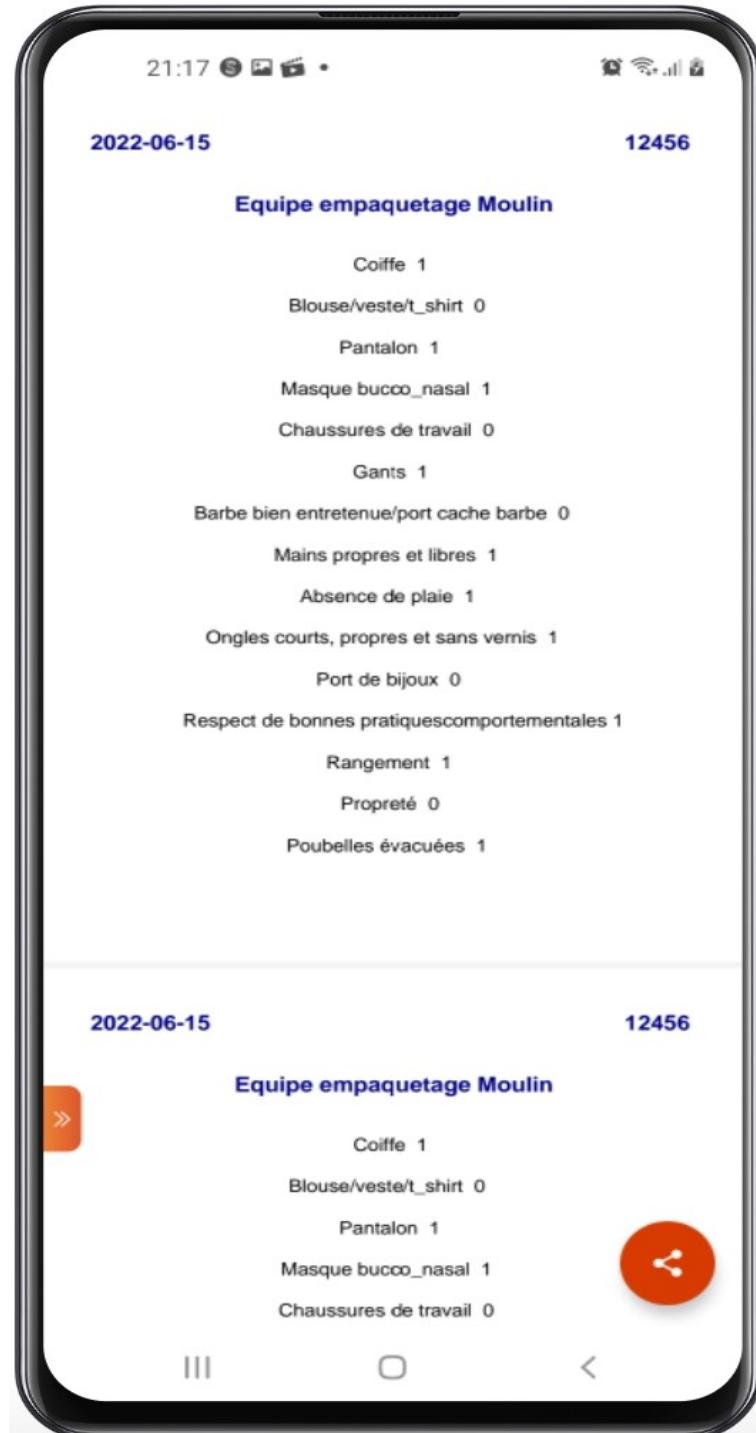


FIGURE 4.13 – Interface imprimer check-list

4.5.6 Test fonctionnel

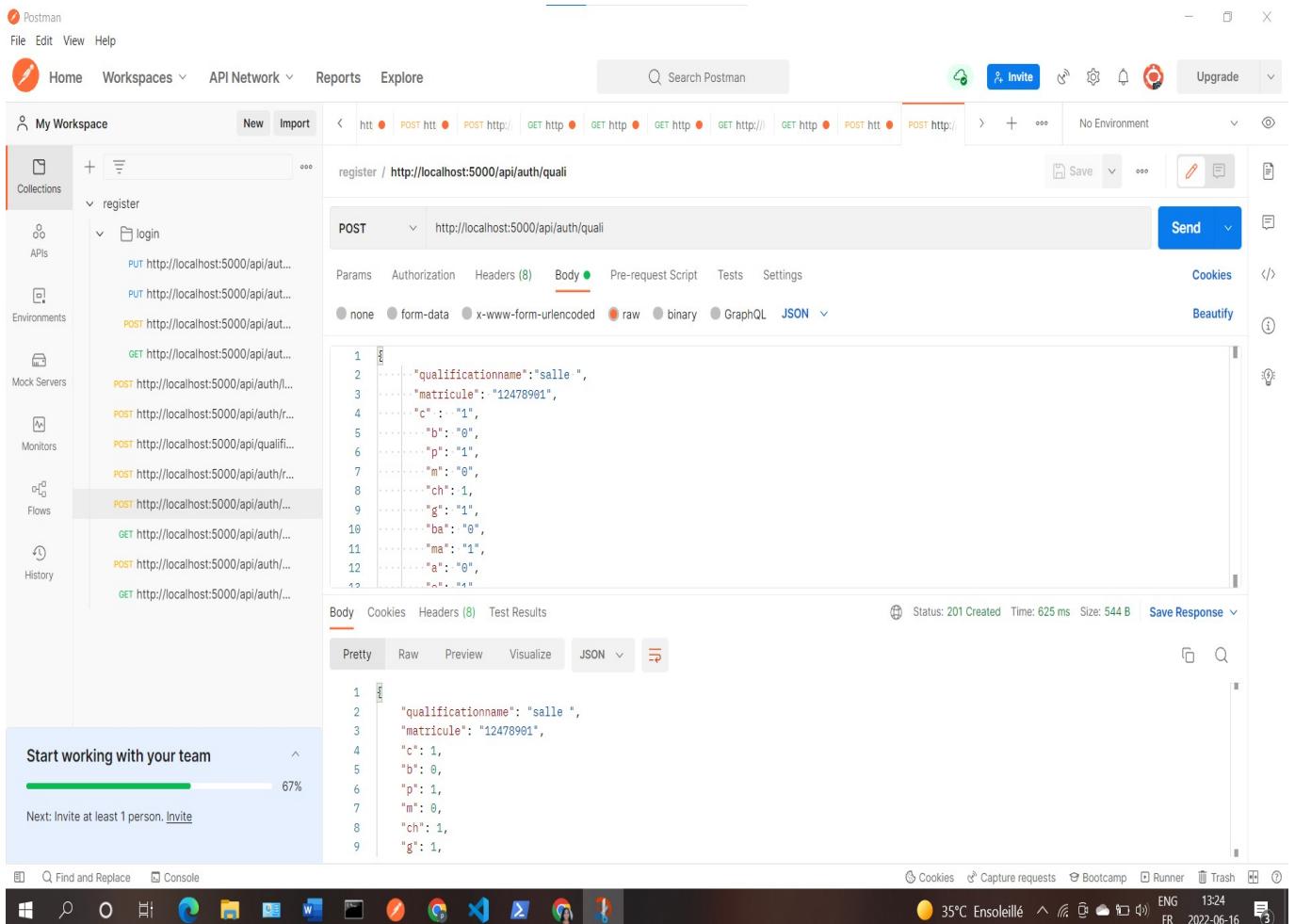


FIGURE 4.14 – Test remplir check-list

SPRINT 2 : GESTION DES CHECK-LISTS

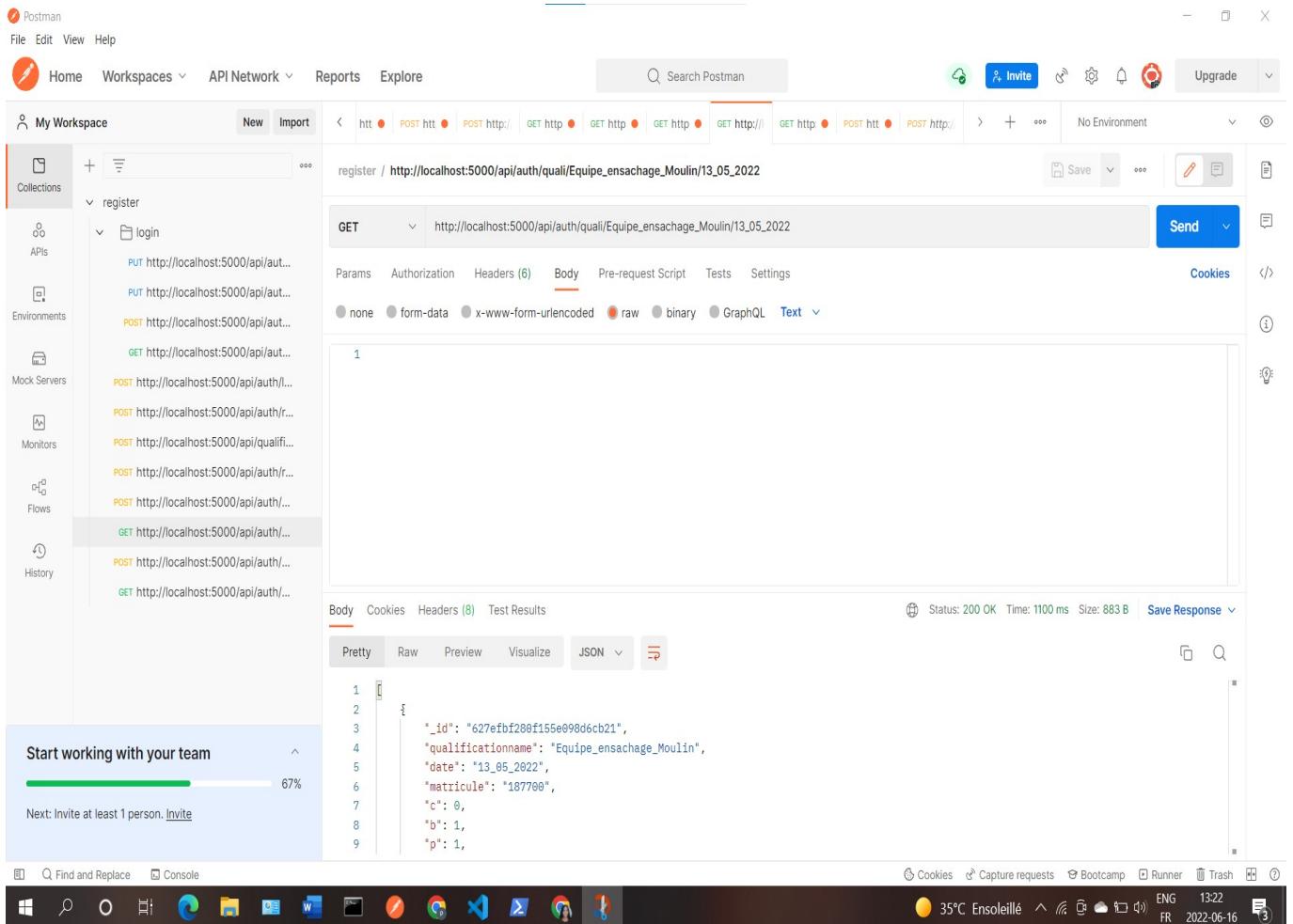


FIGURE 4.15 – Test consulter check-list

4.6 Conclusion

À la fin de ce quatrième chapitre, nous avons réussi à réaliser notre deuxième sprint En commençant par le Backlog de sprint, la conception puis la partie réalisation. Dans le chapitre qui suit, nous allons développer le troisième sprint.

Sommaire

5.1	Introduction	56
5.2	Backlog du sprint 3	56
5.3	Spécifications fonctionnelles	56
5.3.1	Classification des cas d'utilisation par acteur :	57
5.3.2	Diagramme des cas d'utilisation :	57
5.4	Conception	59
5.4.1	Diagramme de séquence :	59
5.4.2	Diagramme de classes global du troisième Sprint :	61
5.5	Test	61
5.5.1	Interface consulter score :	61
5.5.2	Interface consulter tableau de bord :	62
5.6	Conclusion	63

5.1 Introduction

Dans le chapitre précédent, nous avons présenté notre deuxième sprint « **Gestion des check-list** ». Dans ce chapitre nous réaliserons le troisième sprint « **Gestion des Scores, Gestion du Tableau de bord** ». Ce sprint présente la dernière partie de notre application.

5.2 Backlog du sprint 3

Le tableau (5.1) résume les différentes fonctions à effectuer dans ce sprint :

TABLE 5.1 – Backlog du sprint 3.

Sprint 1			
ID	User Story	ID tâche	Tâche
Gérer les scores			
1	En tant que responsable, je peux gérer les scores afin de le consulter.	1.1	Réaliser un diagramme de cas d'utilisation un diagramme de séquences de la fonctionnalité « Consulter les scores » .
		1.2	Développer le cas « Consulter les scores »
		1.3	Tester « Consulter les scores »
Gérer le tableau de bord			
2	En tant que responsable, je peux gérer le tableau de bord afin de le consulter.	2.1	Réaliser un diagramme de cas d'utilisation, de séquence de la fonctionnalité « Consulter le tableau de bord »
		2.2	Développer le cas « Consulter le tableau de bord »
		2.3	Tester « Consulter le tableau de bord »

5.3 Spécifications fonctionnelles

Dans cette partie, nous présentons le diagramme de cas d'utilisations puis une description textuelle pour chacun d'entre eux.

5.3.1 Classification des cas d'utilisation par acteur :

TABLE 5.2 – Classification des cas d'utilisations par acteur

Acteurs	Cas d'utilisation
 Responsable	<ul style="list-style-type: none"> — Gérer les scores — Gérer le tableau de bord

5.3.2 Diagramme des cas d'utilisation :

- Raffinement du cas d'utilisation « Gérer les Scores et le tableau de bord »

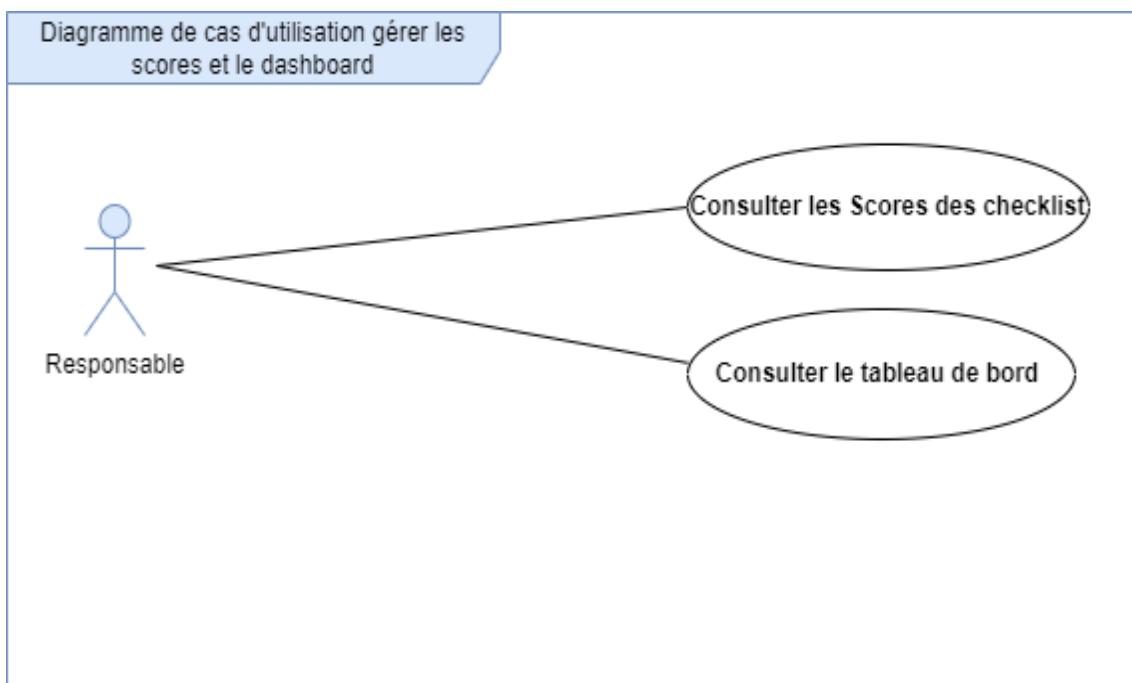


FIGURE 5.1 – Raffinement de cas d'utilisation « Gérer les Scores et le tableau de bord»

Description textuelle du cas d'utilisation « gérer les scores »

TABLE 5.3 – Description textuelle de cas d'utilisation « gérer les scores »

Acteur	Responsable
Objectif	Permet au responsable de gérer les scores.
Pré-condition	Le responsable doivent être authentifié
Post-condition	Opération terminée et validée
Scénario nominatif	<ul style="list-style-type: none"> — Le responsable ouvre l'interface d'accueil, — Le responsable clique sur le bouton « Score », — Le responsable renseigne le nom de la checklist pour consulter le score, — le responsable consulte ses scores.
Scénario alternatif	En cas d'échec, un message d'erreur s'affiche

Description textuelle du cas d'utilisation « consulter le tableau du bord » :

TABLE 5.4 – Description textuelle de cas d'utilisation « consulter le tableau de bord »

Acteur	Responsable
Objectif	Permet au responsable de consulter le tableau du bord.
Pré-condition	Le responsable doivent être authentifié
Post-condition	Opération terminée et validée
Scénario nominatif	<ul style="list-style-type: none"> — Le responsable ouvre l'interface d'accueil, — Le responsable clique sur le bouton « Tableau du bord », — Le responsable renseigne le nom de la checklist pour consulter le Tableau de bord, — consulter le tableau de bord.
Scénario alternatif	En cas d'échec, un message d'erreur s'affiche

5.4 Conception

Dans cette partie, nous allons réaliser les diagrammes de séquences détaillés et le diagramme de classe global.

5.4.1 Diagramme de séquence :

- Diagramme de séquence du cas « Consulter score » :

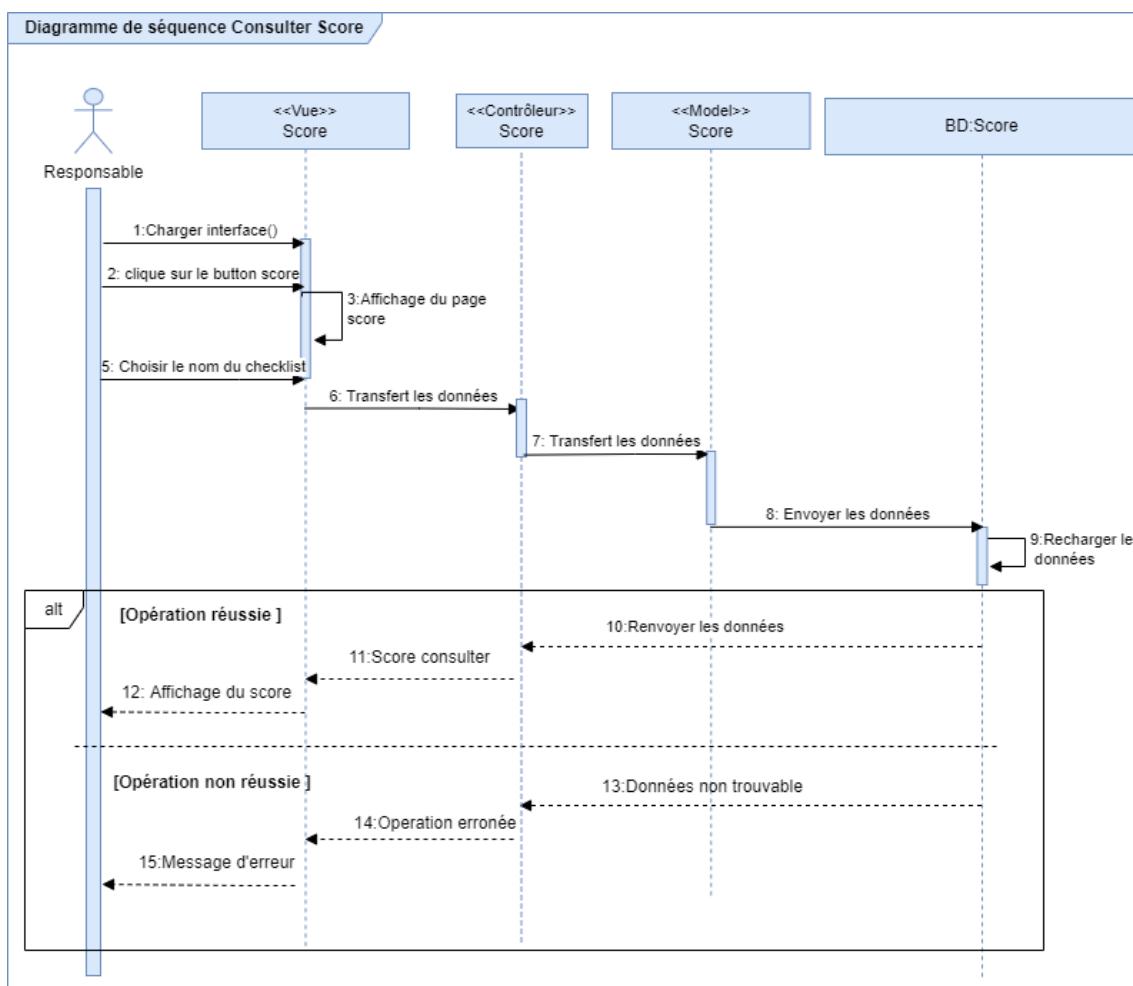


FIGURE 5.2 – Diagramme de séquence « Consulter score »

- Diagramme de séquence du cas « Consulter le tableau de bord » :

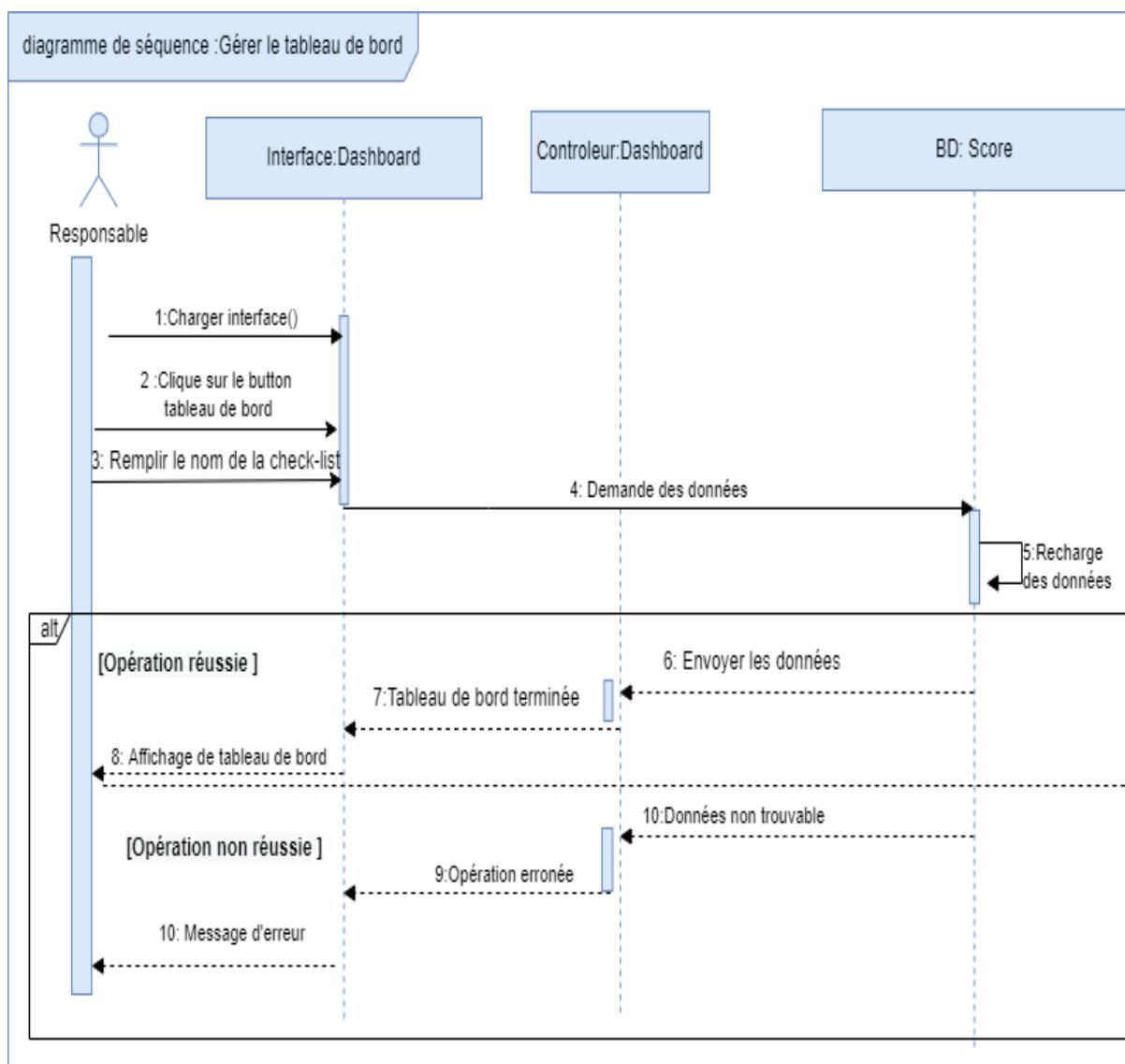


FIGURE 5.3 – Diagramme de séquence du cas « Consulter le tableau de bord »

5.4.2 Diagramme de classes global du troisième Sprint :

La figure ci-dessous décrire le diagramme de classe global relatif au troisième sprint :

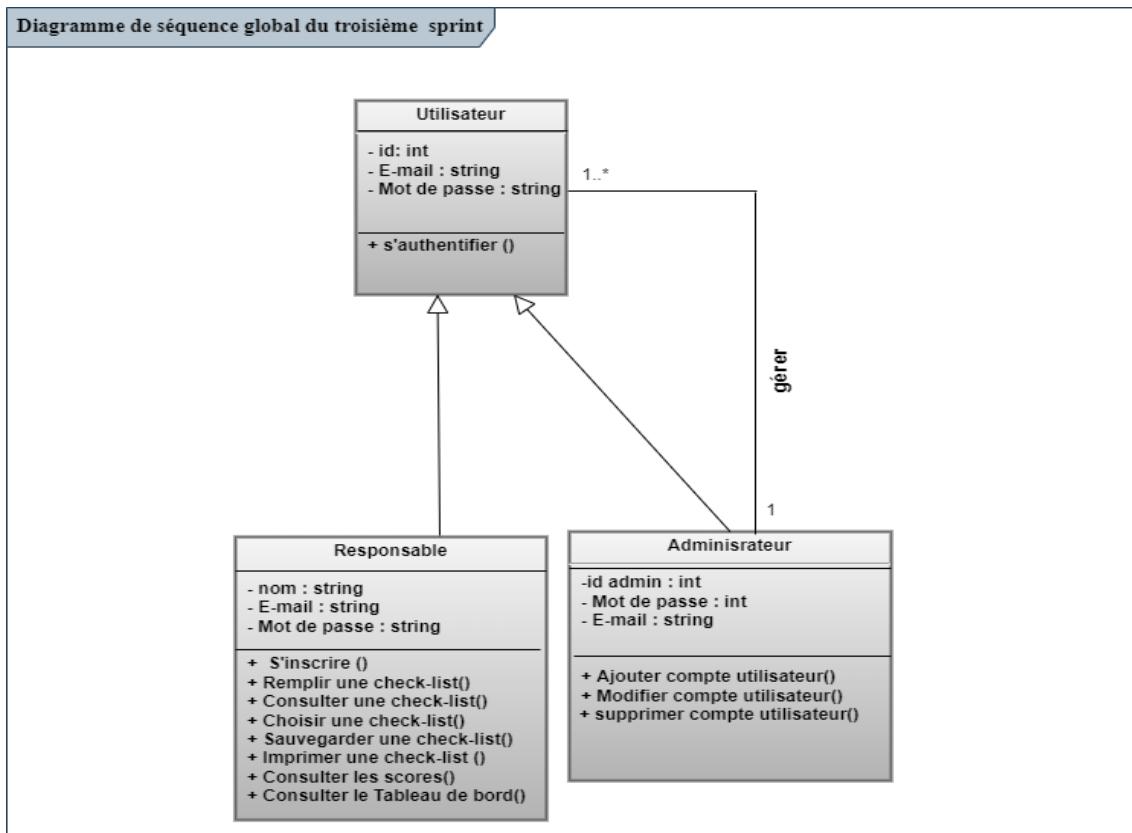


FIGURE 5.4 – Diagramme de classes global du troisième Sprint

5.5 Test

Dans cette partie, nous allons exposer quelques scénarios d'exécution à travers des captures d'écran.

5.5.1 Interface consulter score :

Cette interface permet à l'utilisateur de consulter les scores de check-list :

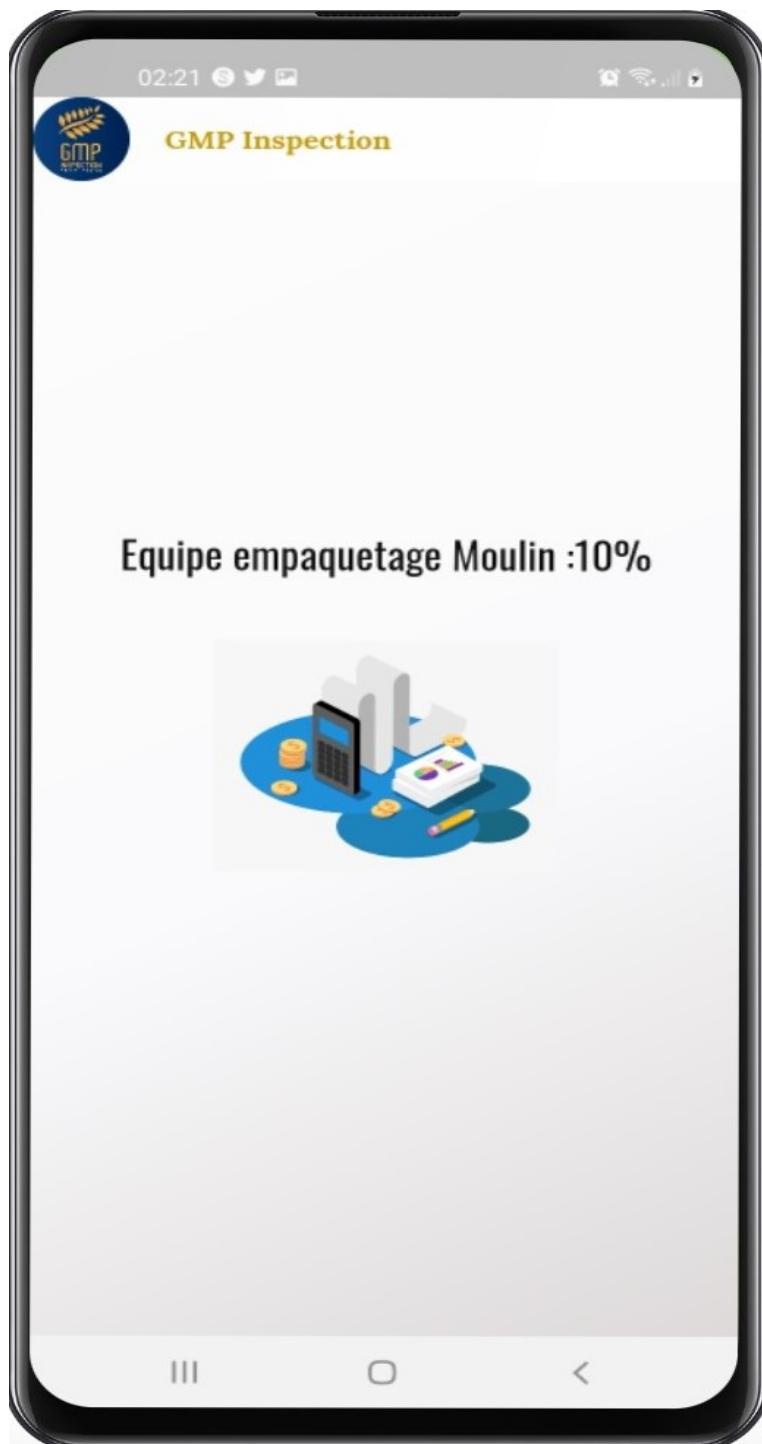


FIGURE 5.5 – Interface consulter score

5.5.2 Interface consulter tableau de bord :

Cette interface permet à l'utilisateur de consulter le tableau de bord de check-list :



FIGURE 5.6 – Interface consulter tableau de bord

5.6 Conclusion

À la fin de ce chapitre, nous avons finalisé notre troisième et dernier sprint, En commençant par le Backlog de sprint, la conception puis la partie réalisation.



CONCLUSION GÉNÉRALE

Ce rapport récapitule le fruit du travail réalisé dans le cadre du projet de fin d'études en licence en informatique de gestion. L'objectif de ce projet était de concevoir une application web mobile intitulé GMP inspection pour la vérification et le monitoring des programmes requis du système de management de la sécurité des aliments selon la méthodologie Scrum en utilisant les technologies Flutter et NodeJs.

En premier lieu, nous avons commencé par le contexte général de notre stage qui s'est déroulé à l'EPI D'OR Sousse. Puis, nous avons identifié les critiques de l'existant et leurs solutions proposées, nous avons aussi focalisé sur la méthodologie appliquée par la bonne gestion de notre projet et langage UML comme langage de modélisation.

Par la suite, nous avons procédé à la planification de notre travail, par l'application de la méthode Scrum, en listant toutes les tâches à effectuer et en attribuant les degrés d'importance à chacune d'entre elles.

Ensuite, nous avons appliqué le même enchainement pour chaque Dprint, en commençant par la phase de spécification des besoins, en passant par la conception, puis le développement, pour couronner le tout par le test de chaque incrément.

À la fin, nous avons pu fournir trois sprints, chaque sprint complétant son précédent au sein de notre application web mobile. Ce travail nous a permis d'approfondir nos connaissances dans le domaine du développement. En effet, cela nous a apporté beaucoup de motivation et d'inspiration dans l'acquisition de nouvelles connaissances lors du développement de notre application.

Pour conclure, nous considérons ce projet comme un premier noyau pour la réalisation de notre objectif. En effet, nous aimerais continuer à développer cette application et ajouter plusieurs fonctionnalités telles que l'ajout, la modification, la suppression de check-lists pour enfin la publier à Play Store.



NETOGRAPHIE

[1] **iAuditor.** Application mobile d'inspection de listes de contrôle utilisée pour effectuer des audits [en ligne]. Disponible sur :

<https://www.getapp.fr/software/102403/iauditor?fbclid=IwAR0L12g15PTdxe-8ijyi\IVNztPzSK-qJ14AHKvMjuNsh5PdFc7YrKaREU8s>
(visité le 18/02/2022).

[2] **capterra.** aider les entreprises luxembourgeoises à choisir le meilleur logiciel [en ligne]. Disponible sur :

https://www.capterra.lu/search/?q=goaudit&fbclid=IwAR3BG05PPvD2vl4pC80U5_EYG5_n\HdNr9RKY0-cOuB3FT_mJ8HMvbOP3Rtw
(visité le 18/02/2022).

[3] **Firstaudit.** les méthodes agiles [en ligne]. Disponible sur :

https://www.check-liste.fr/?fbclid=IwAR3pI6dlpP7npgCg_DkgfR7_YYdVH-Z5\xGuU-ip2AOlMEgypAHEFuiq98mI
(visité le 18/02/2022).

[4] **Planzone.** Disponible sur :

<https://www.planzone.fr/blog/quest-ce-que-la-methodologie-scrum?fbclid=IwAR1-apK-e22\oHaE-jXIQkPzGIwH8MTYwIh9BW2jTd8ocAjL5qrBybj6PVwk>
(visité le 28/02/2022).

[5] **Wikipédia.** Disponible sur :

[https://fr.wikipedia.org/wiki/Flutter_\(logiciel\)?fbclid=IwAR331O9GFPVnWZQM\OaqNxWSjvpu7yKALByUdLKNjkq04XQe2SMPPCapyK5k](https://fr.wikipedia.org/wiki/Flutter_(logiciel)?fbclid=IwAR331O9GFPVnWZQM\OaqNxWSjvpu7yKALByUdLKNjkq04XQe2SMPPCapyK5k)
(visité le 16/03/2022).