

Année Universitaire : 2018-2019

Ministère de l'Enseignement Supérieur

Et de la Recherche Scientifique

Université de Tunis

Institut Supérieur de gestion de Tunis



Rapport

De projet de fin d'études

Présenté en vue de l'obtention du diplôme de

Licence Fondamentale en informatique appliquée à la gestion

Sujet :

Conception et développement d'une plateforme web et mobile de gestion des auto-écoles

Élaboré par :

Chebbi Amira

Ben Othman Oumayma

Organisme d'accueil :

Formalab

Encadré par

ISG Mme Ajroud Henda

Société M. Ayoub Ghazzi

Dédicace

À MES CHERS PARENTS

Aucune dédicace ne saurait exprimer mon respect, mon amour éternel et ma considération pour les sacrifices que vous avez consenti pour mon instruction et mon bien être. Que ce modeste travail soit l'exaucement de vos vœux tant formulés, le fruit de vos innombrables sacrifices, bien que je ne vous en acquitte jamais assez.

À MES CHERS FRERES ET SŒURS

Pour m'avoir soutenu et encouragée à toujours progresser et faire de mon mieux, merci infiniment d'être toujours à mes côtés.

À mes chers neveux et nièces

Meriam, Youssef, Melek, Yasmine : Aucune dédicace ne peut exprimer tout l'amour que j'ai pour vous. Puisse Dieu vous garder, éclairer votre route et vous aider à réaliser à votre tour vos vœux les plus chers.

À TOUS MES AMIS

Pour l'amitié sincère qui nous a liées et des bons moments passés ensemble, je vous dédie ce travail en vous souhaitons un avenir radieux et plein de bonnes promesses. Spécialement aux personnes qui m'ont toujours aidé et encouragé, qui étaient toujours à mes côtés, et qui m'ont accompagné durant mon chemin d'études supérieures, mes aimables amis, collègues à vous Dora, Iheb, Wafa, Nour, Abir et bien évidemment mon binôme Oumayma.

À tous les membres du club ISG GOOGLE CLUB

Ils m'ont permis de dépasser mes limites et de croire en moi. J'ai relevé des défis et j'ai vécue des expériences que les mots ne peuvent exprimer. Ce club m'a aussi permis de rencontrer des personnes exceptionnelles. Je serai à jamais reconnaissante et je chérirai ces moments éternellement.

Chebbi Amira

Dédicace

A la mémoire de ma chère mère Faïza qui est toujours dans mon cœur et mon esprit, qui a souhaité vivre longtemps pour me voir réussir. Je te dédie aujourd'hui ma réussite.

Que dieu t'accueille dans son paradis éternel

A mon père Belhassen l'épaule solide de ma vie et la personne la plus digne de mon estime et de mon respect. Aucune dédicace ne saurait exprimer mes sentiments.

Que dieu te préserve et te procure santé et longue vie.

A mon frère Hedi, en témoignage de l'amour et de l'attachement que je te port. Je ne pourrai jamais te remercier assez pour tout ce que tu as fait pour moi.

Je te dédie ce modeste travail avec mes vœux de réussite, de bonheur et de santé

A mon mari Mohamed Yassine, aucun mot ne saurait t'exprimer mon profond attachement et ma reconnaissance pour l'amour, la tendresse et le soutien dont tu m'as toujours entouré.

Aux personnes dont j'ai bien aimé la présence, à ma sœur Amira, à mes amis Jihen, Eya, Fatma, Nour et bien évidemment mon binôme Amira. Vous êtes pour moi des sœurs et des amies sur qui je peux compter.

Ben Othman Oumayma

Remerciements

On remercie d'abord le tout puissant de nous avoir donné la santé et la volonté de mener à bien ce projet et de clôturer cette année par ce dernier.

Nous tenons à remercier toute personne ayant contribuer à la réalisation de ce projet de prêt ou de loin, et c'est avec ces quelques lignes que nous vous témoignons notre gratitude :

Tout d'abord, ce travail ne serait pas aussi riche et n'aurait pas pu avoir le jour sans l'aide et l'encadrement de Mme AJROUD HENDA. Tout d'abord on la remercie pour avoir cru en nous. On la remercie également pour la qualité de son encadrement exceptionnel, pour sa patience, sa rigueur et sa disponibilité durant notre préparation de ce projet de fin d'étude.

M. Ayoub Ghazzi et M. Gouia Nadim ainsi que toute l'équipe de FORMALAB nous vous remercions pour nous avoir fourni cette opportunité d'effectuer notre stage au sein de l'entreprise Formalab, merci également pour votre soutien continu et pour vos recommandations et conseils précieux.

Nous exprimons notre profonde reconnaissance à tout le cadre professionnel de L'ISG pour la formation d'excellence qu'ils nous ont offert, pour leurs conseils et leurs contributions pour que nous entamons notre carrière professionnelle avec des bases solides.

Finalement, on saisit cette occasion pour remercier les membres du jury en espérant qu'ils trouvent dans ce rapport les qualités de clarté et de motivation qu'ils attendent.

Amira et Oumayma

Table des matières

Introduction Générale	1
Introduction.....	3
I. Environnement du projet	3
1.1 Présentation du cadre du stage	3
1.2 Présentation de l'organisme d'accueil.....	3
II. Présentation du projet	3
2.1 Etude de l'existant.....	3
2.2 Critique de l'existant.....	9
2.3 Solution proposée	9
2.4 Spécification des besoins.....	10
2.4.1 Identification des acteurs	10
2.4.2 Besoins fonctionnels	10
2.4.2 Les besoins non fonctionnels	11
2.5 Méthodologie adaptée	12
2.5.1 Comparaison entre les différentes méthodologies agile	12
2.5.2 Choix de la méthodologie	13
2.6 Environnement de développement	15
2.6.1 Environnement matériel.....	15
2.6.2 Environnement logiciel.....	16
2.7 Langages utilisés.....	19
2.7.1 Langage de développement	19
2.7.2 Langage de modélisation	20
2.7.3 Architecture.....	20
III. Découpage du projet	22
3.1 Pilotage du projet avec SCRUM	22
3.2 Backlog du produit.....	23
3.3 Planification des sprints.....	25
3.4 Le planning du projet	26
3.5 Outil de Gestion de projet : Trello.....	27
3.6 Prototypes d'interfaces	27
IV. Conclusion	31

Introduction.....	33
I. Spécification des besoins	33
1.1 Le sprint backlog.....	33
II. Analyse des cas du sprint 1	37
2.1. Classification des cas d'utilisation par acteur	37
2.2. Diagramme de cas d'utilisation du premier sprint	37
2.3. Description des cas d'utilisation du premier sprint	38
2.3.1. Analyse du cas d'utilisation « Consulter la liste des moniteurs ».....	38
2.3.2. Analyse du cas d'utilisation « Consulter la liste des véhicules »	41
2.3.2. Analyse du cas d'utilisation « Géolocaliser les véhicules »	41
2.3.3. Analyse du cas d'utilisation « Consulter la liste des candidats avec le nombre de rapport »	43
2.3.4. Analyse du cas d'utilisation « Afficher le planning »	45
III. Conception	47
3.1 Conception de cas ‘Consulter la liste des moniteurs’	47
3.2 Conception de cas ‘Ajouter un moniteur’	48
3.3 Conception de cas ‘Modifier un moniteur’	48
3.4 Conception de cas ‘Supprimer un moniteur’	49
3.5 Conception de cas ‘Consulter un moniteur’	49
3.6 Conception de cas ‘Géolocaliser les véhicules’	50
3.7 Conception des cas ‘Consulter informations candidat et moniteur’	50
3.8 Conception de cas ‘Consulter la liste des candidats avec le nombre de rapports ‘	50
3.9 Conception de cas ‘Ajouter un rapport’	51
3.10 Conception de cas ‘Consulter le rapport’	51
3.11 Conception de cas ‘Modifier le rapport’	52
3.12 Conception de cas ‘Afficher le planning’	52
3.13 Conception de cas ‘Ajouter une séance’	52
3.14 Conception de cas ‘Consulter une séance’	53
3.15 Conception de cas ‘Supprimer une séance’	53
3.16 Conception de cas ‘Modifier une séance’	53
3.17 Conception de cas ‘consulter liste des véhicules’	53
3.18 Conception de cas ‘Ajouter un véhicule’	53
3.19 Conception de cas ‘Modifier un véhicule’	53

3.20 Conception de cas ‘Supprimer un véhicule’	53
3.21 Conception de cas ‘Consulter un véhicule’	53
IV. Diagramme de classe globale du premier sprint	54
V. Structure de la base de données	54
IV. Implémentation	57
4.1 Diagramme de composants	57
V. Test	63
5.1 Les tests des web services	63
5.2 Les tests unitaires	65
VI. Conclusion	67
Introduction	69
I. Spécification des besoins	69
1.1 Le Sprint backlog	69
II. Analyse des cas du deuxième sprint	74
2.1 Classification des cas d’utilisations par acteur	74
2.2 Diagramme de cas d’utilisation de deuxième sprint	74
2.3 Description des cas d’utilisation de l’application web du deuxième sprint	75
2.3.1 Analyse du cas d’utilisation « S’authentifier »	75
2.3.2 Analyse du cas d’utilisation « Gérer comptabilité »	76
2.3.3 Analyse du cas d’utilisation « Gérer les cours »	79
2.3.4 Analyse du cas d’utilisation « Consulter liste des candidats »	80
2.3.5 Analyse du cas d’utilisation « Consulter compte »	80
2.3.6 Analyse du cas d’utilisation « Consulter le planning »	81
2.3.7 Analyse de cas d’utilisation « Passer un test en ligne »	81
2.3.8 Analyse de cas d’utilisation « Consulter les cours »	82
2.4 Description des cas d’utilisation de l’application mobile du deuxième sprint ...82	82
2.4.1 Description textuelle du cas d’utilisation « S’authentifier »	82
2.4.2 Analyse du cas d’utilisation « Gérer compte »	82
2.4.3 Analyse du cas d’utilisation « Consulter le planning »	84
2.4.4 Analyse du cas d’utilisation « Consulter liste des candidats »	86
III. Conception	87
3.1 Conception de l’application web	87
3.1.1 Conception du cas « S’authentifier »	87
3.1.2 Conception du cas « Consulter la liste des dépenses »	87

3.1.3	Conception du cas « Ajouter une dépense »	88
3.1.4	Conception du cas « Modifier une dépense »	89
3.1.5	Conception du cas « Supprimer une dépense ».....	89
3.1.6	Conception du cas « Consulter statistiques »	89
3.1.7	Conception du module « Consulter la liste des gains »	90
3.1.8	Conception du module « Consulter la liste des factures ».....	90
3.1.9	Conception du module « Gérer candidat ».....	90
3.1.10	Conception du cas « Consulter cours ».....	90
3.1.11	Conception du cas « Consulter le planning »	90
3.1.12	Conception du cas « Consulter compte ».....	90
3.1.13	Conception du cas « Modifier compte ».....	90
3.1.14	Conception du module « Consulter la liste des cours »	90
3.1.15	Conception de cas « Passer les tests en ligne »	91
3.2	Conception de l'application mobile	91
3.2.1	Conception du cas « Consulter le planning »	91
3.2.2	Conception du cas « Consulter compte ».....	92
3.2.3	Conception du cas « Modifier compte ».....	93
3.2.4	Conception du cas « Réinitialiser mot de passe »	93
3.2.5	Conception de cas « S'authentifier ».....	93
IV.	Diagramme de classe globale du deuxième sprint	94
V.	Structure de la base de données.....	94
VI.	Implémentation	96
6.1	Diagramme de composants.....	96
VII.	Les web services	104
VIII.	Test	104
8.1	Les Tests des web service.....	104
8.2	Les tests unitaires	105
8.2.1	Le test unitaire du cas « Ajouter une dépense »	105
IX.	Conclusion	107
	Conclusion générale et perspectives	108
	Bibliographie.....	109
	Annexe 1.....	111
	Diagramme de Gantt	111
	Annexe 2.....	112

Table de Figures

Figure 1 Interface d'accueil d'Auto-plus	4
Figure 2 Interface d'inscription pour un candidat	4
Figure 3 Interface du planning	5
Figure 4 Interface qui représente la page d'accueil de l'application.....	6
Figure 5 Interfaces d'accueil des différentes applications mobile et web	7
Figure 6 Interface du site vitrine	8
Figure 7 L'application mobile sur le Play Store	8
Figure 8 Vue global Scrum	14
Figure 9 Architecture du projet.....	20
Figure 10 architecture MVVM	21
Figure 11 architecture MVC	22
Figure 12 Hiérarchie de l'équipe	23
Figure 13 Découpage des sprints	26
Figure 14 Planning du déroulement du projet.....	26
Figure 15 Capture de Trello	27
Figure 16 Prototype de l'interface agenda	27
Figure 17 Prototype de l'interface gestion des moniteurs	28
Figure 18 Prototype de l'interface ajouter moniteur	28
Figure 19 Prototype de l'interface géolocalisation des véhicules	29
Figure 20 Prototype de l'interface ajouter rapport dans l'application mobile	29
Figure 21 Prototype de l'interface consulter planning	29
Figure 22 Prototype de l'interface s'authentifier	29
Figure 23 Prototype de l'interface d'accueil 2	30
Figure 24 Prototype de page d'accueil 1	30
Figure 25 Prototype de l'interface de contact du site vitrine.....	31
Figure 26 Diagramme de cas d'utilisation du Sprint 1	38
Figure 27 Diagramme de cas d'utilisation « Consulter la liste des moniteurs »	38
Figure 28 Diagramme de cas d'utilisation « Consulter la liste des véhicules »	41
Figure 29 Diagramme de cas d'utilisation « géolocaliser les véhicules ».....	41
Figure 30 Diagramme de cas d'utilisation « Consulter la liste des candidats avec le nombre de rapport ».....	43
Figure 31 Diagramme de cas d'utilisation « Afficher le planning »	45
Figure 32 Diagramme de séquence de l'opération « Consulter la liste des moniteurs »	47
Figure 33 Diagramme de séquence de l'opération « Ajouter un moniteur ».....	48
Figure 34 Diagramme de séquence de l'opération « Modifier un moniteur ».....	48
Figure 35 Diagramme de séquence de l'opération « Supprimer un moniteur »	49
Figure 36 Diagramme de séquence de l'opération « Consulter un moniteur »	49
Figure 37 Diagramme de séquence de l'opération « Géolocaliser les véhicules ».....	50
Figure 38 Diagramme de séquence du cas « Consulter informations candidat et moniteur »..	50
Figure 39 Diagramme de séquence de l'opération « Consulter la liste des candidats avec le nombre de rapports »	50

Figure 40 Diagramme de séquence de l'opération « Ajouter un rapport ».....	51
Figure 41 Diagramme de séquence de l'opération « Consulter le rapport »	51
Figure 42 Diagramme de séquence de l'opération « Modifier le rapport ».....	52
Figure 43 Diagramme de séquence de l'opération « Afficher le planning ».....	52
Figure 44 Diagramme de classe globale du sprint 1	54
Figure 45 Diagramme de composants de la fonctionnalité : « Gestion moniteur ».....	57
Figure 46 Interface du cas "Gestion moniteurs"	58
Figure 47 Diagramme de composants de la fonctionnalité : « Consulter le planning » (mobile)	58
Figure 48 Interfaces du cas consulter le planning (mobile)	59
Figure 49 Diagramme de composants de la fonctionnalité : « Gestion planning » (web)	59
Figure 50 Interface du cas "Gestion planning " (web)	60
Figure 51 Diagramme de composants de la fonctionnalité : « Géolocaliser les véhicules » /2	60
Figure 52 Diagramme de composants de la fonctionnalité : « Géolocaliser les véhicules » /1	60
Figure 53 Diagramme de composants de la fonctionnalité : « Géolocaliser les véhicules » /3	60
Figure 54 Diagramme de composants de la fonctionnalité : « Géolocaliser les véhicules » partie 4	61
Figure 55 Diagramme de composants de la fonctionnalité : « Géolocaliser les véhicules » partie 5	61
Figure 56 Interfaces du cas "Géolocaliser les véhicules"	61
Figure 57 Diagramme de composants de la fonctionnalité : « Gestion rapport » partie 1	62
Figure 58 Diagramme de composants de la fonctionnalité : « Gestion rapport » partie 2	62
Figure 59 Interfaces de la fonctionnalité « Gestion rapport »	63
Figure 60 Test du web service "Consulter les moniteurs"	64
Figure 61 Test du web service "Consulter les véhicules"	64
Figure 62 Test du web service "Consulter les séances"	65
Figure 63 méthode de test unitaire ajouter un moniteur	66
Figure 64 Résultat du test ajout d'un moniteur.....	66
Figure 65 Résultat d'erreur d'ajout d'un moniteur	67
Figure 66 Diagramme de cas d'utilisation du deuxième sprint	75
Figure 67 Diagramme de cas d'utilisation “Gérer comptabilité“	76
Figure 68 Diagramme de cas d'utilisation “Gérer les cours “	79
Figure 69 Diagramme de cas d'utilisation “Consulter compte“	80
Figure 70 Diagramme de cas d'utilisation “Gérer compte”	82
Figure 71 Diagramme de cas d'utilisation “Consulter le planning”	84
Figure 72 Diagramme de cas d'utilisation “Consulter liste des candidats”.....	86
Figure 73 Diagramme de séquence du cas d'utilisation "S'authentifier"	87
Figure 74 Diagramme de séquence de cas d'utilisation "Consulter la liste des dépenses"	87
Figure 75 Diagramme de séquence du cas d'utilisation "Ajouter une dépense"	88
Figure 76 Diagramme de séquence du cas d'utilisation "Modifier une dépense"	89
Figure 77 Diagramme de séquence du cas d'utilisation "Supprimer une dépense"	89
Figure 78 Diagramme de séquence de "Passer les tests en ligne"	91
Figure 79 Diagramme de séquence de "Consulter le planning"	92
Figure 80 Diagramme de séquence du cas d'utilisation "Modifier compte".....	93

Figure 81 Diagramme de classe globale du deuxième sprint	94
Figure 82 Diagramme de composants de la fonctionnalité : « S'authentification » version web	96
Figure 83 Interfaces web du cas "S'authentifier"	97
Figure 84 Diagramme de composants de la fonctionnalité : « S'authentification » version mobile	97
Figure 85 Interfaces mobiles du cas "S'authentifier"	97
Figure 86 Diagramme de composants de la fonctionnalité : « Gestion comptabilité » Partie 2	98
Figure 87 Diagramme de composants de la fonctionnalité : Gestion comptabilité Partie 1	98
Figure 88 Interfaces de la fonctionnalité « Gestion comptabilité »	99
Figure 89 Diagramme de composants de la fonctionnalité : « Gestion cours »	99
Figure 90 Interface de la fonctionnalité "Gestion cours".....	99
Figure 91 Diagramme de composants de la fonctionnalité : « Consulter cours »	100
Figure 92 Diagramme de composants de la fonctionnalité : « Consulter planning » Partie web	100
Figure 93 Interface du cas "Consulter planning"	100
Figure 94 Diagramme de composants de la fonctionnalité : « Consulter planning » Mobile Partie 1	101
Figure 95 Diagramme de composants de la fonctionnalité : « Consulter planning » mobile partie 2	101
Figure 96 Interface consulter planning -mobile.....	102
Figure 97 Diagramme de composants de la fonctionnalité : « Gérer compte ».....	102
Figure 98 Diagramme de composants de la fonctionnalité : « Gérer compte » web	102
Figure 99 Interface du cas « Gérer compte » web	103
Figure 100 Interface du cas « Gérer compte » -mobile.....	103
Figure 101 Diagramme de composants de la fonctionnalité : « Passer les tests en ligne »	103
Figure 102 Interface du cas " Passer test en ligne "	104
Figure 103 Collection "dépense"	105
Figure 104 Modifier compte -Postman	105
Figure 105 méthode de test d'ajouter une dépense.....	106
Figure 106 Résultat du test ajout dépense.....	106
Figure 107 Résultat d'échec lors du test ajout dépense	106

Liste des tableaux

Tableau 1 Comparaison des méthodologies.....	13
Tableau 2 Backlog du produit.....	25
Tableau 3 Backlog du sprint 1	37
Tableau 4 Classification des cas d'utilisation par acteur.....	37
Tableau 5 Description textuelle du cas d'utilisation "Consulter la liste des moniteurs"	39
Tableau 6 Description textuelle du cas d'utilisation "Ajouter un moniteur".....	39
Tableau 7 Description du cas d'utilisation « Modifier un moniteur »	40
Tableau 8 Description textuelle du cas d'utilisation "Supprimer un moniteur"	40
Tableau 9 Description textuelle du cas d'utilisation "Consulter un moniteur"	40
Tableau 10 Description textuelle du cas d'utilisation "Géolocaliser les véhicules".....	42
Tableau 11 Description textuelle du cas d'utilisation "Consulter informations du candidat et du moniteur"	42
Tableau 12 Description textuelle du cas d'utilisation « Consulter liste de candidats avec le nombre de rapport "	43
Tableau 13 Description textuelle du cas d'utilisation "Ajouter un rapport".....	44
Tableau 14 Description textuelle du cas d'utilisation "Consulter la liste des rapports "	44
Tableau 15 Description textuelle du cas d'utilisation "Afficher le planning"	45
Tableau 16 Description textuelle du cas d'utilisation "Ajouter une séance".....	46
Tableau 17 Description textuelle du cas d'utilisation "Consulter une séance"	46
Tableau 18 Description textuelle du cas d'utilisation "Supprimer une séance"	47
Tableau 19 Description textuelle du cas d'utilisation « Modifier une séance »	47
Tableau 20 Table administrateurs	55
Tableau 21 Table moniteurs.....	55
Tableau 22 Table véhicules.....	56
Tableau 23 Table Séances.....	56
Tableau 24 Table rapports	56
Tableau 25 Table rapports	57
Tableau 26 Table résultats	57
Tableau 27 Backlog sprint 2	74
Tableau 28 Classification des cas d'utilisation par acteur du deuxième sprint	74
Tableau 29 Description textuelle du cas d'utilisation « S'authentifier »	75
Tableau 30 Description textuelle du cas d'utilisation "Consulter la liste des dépenses"	76
Tableau 31 Description textuelle du cas d'utilisation "Ajouter une dépense"	77
Tableau 32 Description textuelle du cas d'utilisation "Modifier une dépense"	77
Tableau 33 Description textuelle du cas d'utilisation "Supprimer une dépense"	78
Tableau 34 Description textuelle du cas d'utilisation "Consulter un cours"	79
Tableau 35 Description textuelle du cas d'utilisation "Consulter compte"	80
Tableau 36 Description textuelle du cas d'utilisation " Modifier compte "	81
Tableau 37 Description textuelle du cas d'utilisation "Consulter le planning"	81
Tableau 38 Description textuelle du cas d'utilisation "Passer un test en ligne"	81
Tableau 39 Description textuelle du cas d'utilisation "Consulter les cours"	82
Tableau 40 Description textuelle du cas d'utilisation" Consulter compte".....	83

Tableau 41 Description textuelle du cas d'utilisation "Modifier compte"	83
Tableau 42 Description textuelle du cas d'utilisation "Réinitialiser mot de passe"	84
Tableau 43 Description textuelle du cas d'utilisation "Consulter le planning"	85
Tableau 44 Description textuelle du cas d'utilisation "Consulter compte candidat"	85
Tableau 45 Description textuelle du cas d'utilisation "Consulter compte moniteur"	85
Tableau 46 Description textuelle du cas d'utilisation "Consulter la liste des candidats"	86
Tableau 47 Description textuelle du cas d'utilisation "Consulter compte candidat"	87
Tableau 48 Table des dépenses	94
Tableau 49 Table des gains	95
Tableau 50 Table des cours	95
Tableau 51 Table des factures	95
Tableau 52 Table des candidats	96
Tableau 53 Table des administrateurs	96
Tableau 54 tableau des WEB SERVICE	121

Introduction Générale

Dans un monde actif et continuellement évolutif, la possession des moyens efficaces et performants de digitalisation devient primordiale. En effet, loin d'être un phénomène de mode ou de tendance, la digitalisation vient nous apporter plein d'avantages. De ce fait, elle est devenue la stratégie à laquelle plusieurs entreprises ont recours. Cette transformation digitale est devenue alors un signe de progrès et de maturité grâce aux services offerts pour l'entreprise ou l'administration ainsi que pour le personnel.

Selon un rapport réalisé par Capgemini Consulting¹, les leaders du marché en matière de rentabilité, de croissance et de satisfaction sont les entreprises les plus avancées dans la transformation digitale. Ces entreprises sont considérées plus performantes que leurs consœurs selon différents critères comme la satisfaction client, la capacité d'innovation, la rentabilité ou la croissance [1].

Dans le cadre du projet de fin d'études pour l'obtention de la licence fondamentale en informatique de gestion, la société Formalab nous a confié l'étude, la conception et le développement d'une application qui s'intitule « Gestion des auto-écoles ». Ce rapport présente les différentes étapes suivies ainsi que les technologies et outils utilisés pour réaliser notre application. Il se compose de trois chapitres.

Le premier chapitre, « Etude préliminaire », englobe la présentation du projet, de l'environnement de développement ainsi que de la méthodologie adoptée pour sa réalisation. Il inclut aussi la spécification des besoins qui présente les différentes fonctionnalités ainsi que les acteurs intervenants.

Par la suite, un deuxième chapitre, « Sprint 1 », présente le premier sprint, ainsi que le sprint backlog. Il contient les différents modules de gestion des moniteurs, des véhicules, du planning et des rapports.

Nous terminons par le troisième chapitre, « Sprint 2 », qui présente le deuxième sprint de notre projet. Il englobe, d'une part, les modules de gestion de la comptabilité, des cours en ligne, des candidats et des comptes, et, d'autre part, les fonctionnalités liées aux tests en ligne et à l'élaboration du site vitrine présentant notre produit.

Nous clôturons notre rapport par une conclusion générale et les perspectives de notre travail.

¹ Capgemini Consulting est une entreprise qui offre des services aux entreprises

Chapitre 1

Etude préliminaire

Introduction

Ce premier chapitre concerne la présentation de notre projet de fin d'études qui va nous offrir une vision claire de ce dernier. Il englobe, dans un premier lieu, la mise en place du projet dans son cadre général, l'introduction de l'environnement du projet en présentant l'entreprise d'accueil et son domaine d'activité. Dans un second lieu, on présente une étude de l'existant et on propose une solution. Nous établissons une vision globale de notre produit tout en dégageant ses fonctionnalités et en identifiant les différents acteurs qui interagissent avec notre système. Nous présentons de même la méthodologie de réalisation et le langage de modélisation qu'on a adoptés ainsi que notre environnement de développement. Pour finir on procède à une élaboration de la planification de réalisation du projet. Cette phase de planification et architecture est consacrée essentiellement à la préparation de l'environnement de développement.

I. Environnement du projet

1.1Présentation du cadre du stage

Dans le cadre de notre formation au sein de l'Institut supérieur de gestion de Tunis, notre projet de fin d'études est intitulé « Conception et développement d'une plateforme web et une application mobile de gestion d'une auto-école » effectués au sein de la société Formalab, dans le but d'obtenir le diplôme de licence fondamental en informatique appliquée à la gestion.

1.2Présentation de l'organisme d'accueil

Formalab, fondée en 2016 par Ayoub Ghozzi et Nadim Gouia, est une entreprise de services informatiques et un laboratoire de formation continue en informatique. Sa gamme de cours en modules et filières proposée est conçue pour couvrir les besoins de l'initiation à la maîtrise des qualifications professionnelles. Formalab est sollicitée aussi pour accompagner plusieurs entreprises dans leur transition numérique avec de nouveaux projets en développement spécifique et la création de nouveaux sites web.

II. Présentation du projet

2.1Etude de l'existant

Cette partie est considérée comme étant une étape primordiale et essentielle lors de la réalisation de notre projet. Dans le but d'avoir une idée sur les produits existants nous avons effectué, au cours de cette phase, des recherches de produits similaires. L'étude de l'existant effectuée sur une double échelle (nationale et internationale) nous a permis de faire une sélection des solutions existantes. Nos résultats sont comme suit :

- Sur l'échelle nationale :**

- Auto-plus² :**

Une Application de gestion desktop consacrée aux auto-écoles et qui permet de gérer les activités administratives et financières des auto-écoles.

² <http://www.ksoft-tunisie.com/logiciel-fiche.php?ref=AUTO%20PLUS>

Voici ci-dessous les figures 1,2 et 3 présentant quelques captures des interfaces de cette application.



Figure 1 Interface d'accueil d'Auto-plus



Figure 2 Interface d'inscription pour un candidat

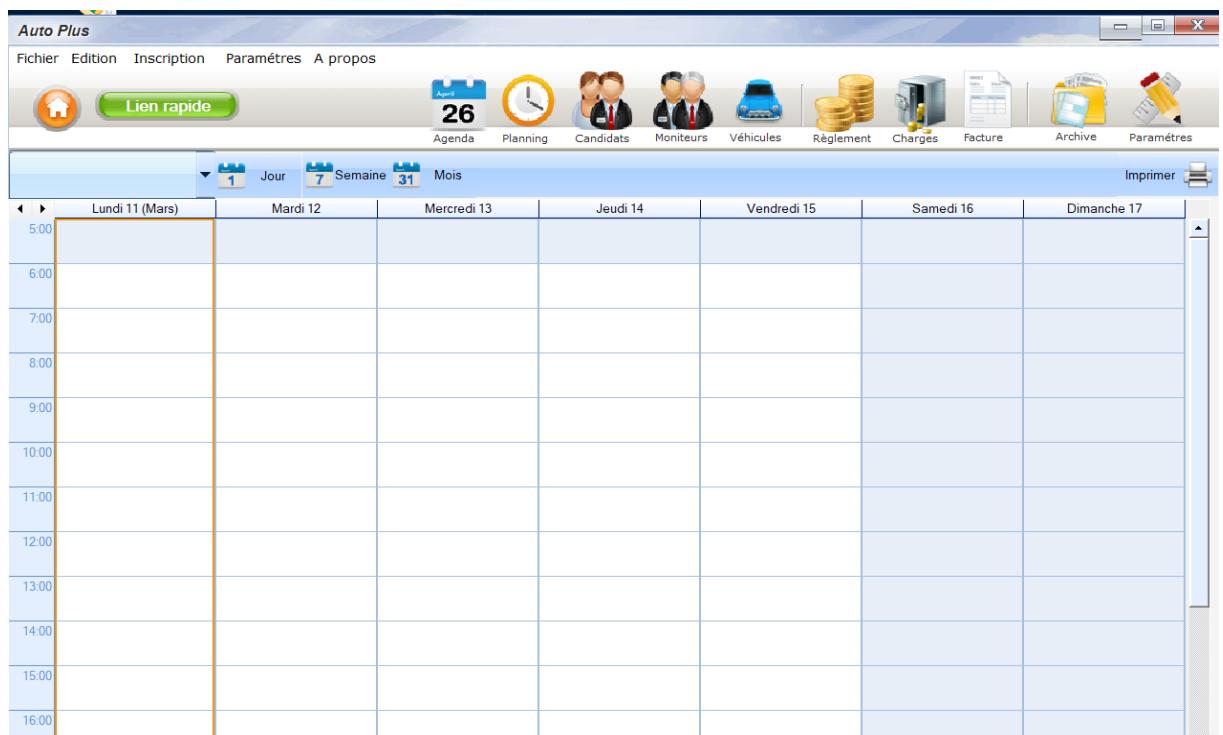


Figure 3 Interface du planning

- **Les points forts :**

- Le logiciel permet d'imprimer les différents fichiers existants.
- Le logiciel présente une interface simple et claire.

- **Les points faibles :**

- Auto Plus présente une application desktop seulement : Une absence de version web ou mobile.
- Les services de cette application sont dédiés à l'administrateur seulement.
- L'ergonomie n'est pas moderne.

- **IT4IT³ :**

Offre un logiciel pour la gestion d'une auto-école ainsi que des candidats et des moniteurs.

Voici ci-dessous la figure 4 qui présente l'interface de ce logiciel.

³ <http://www.it4it.com.tn/>

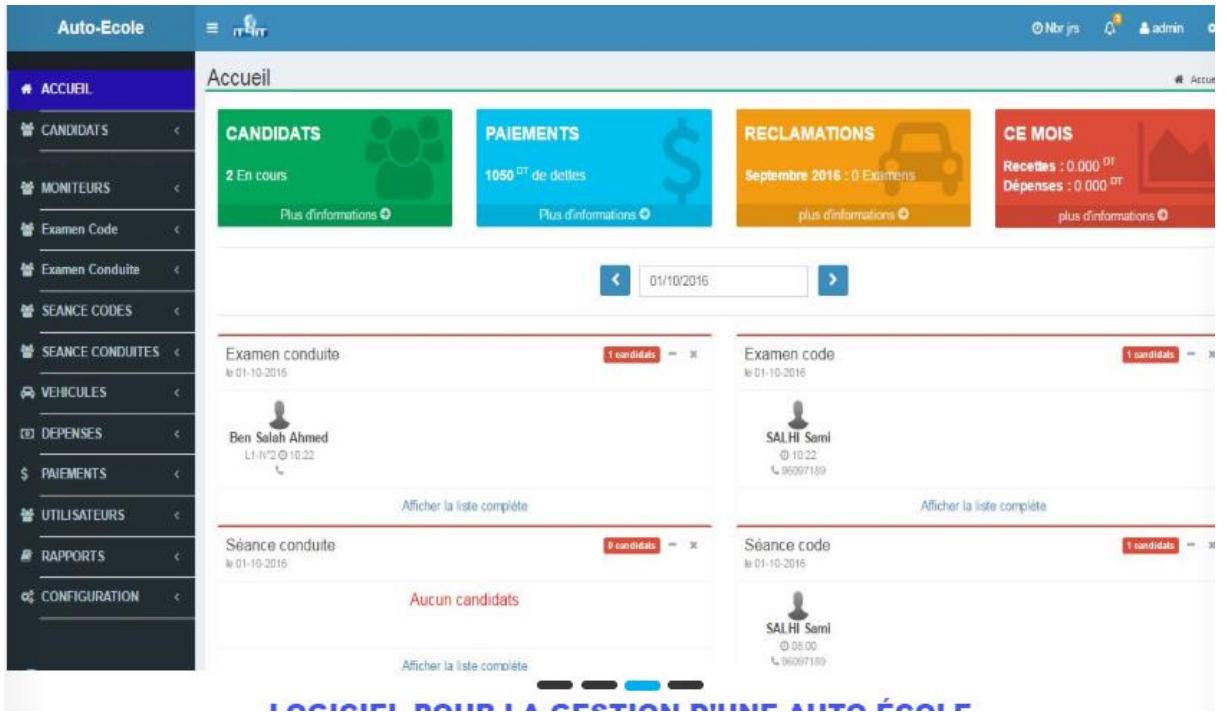


Figure 4 Interface qui représente la page d'accueil de l'application

- **Les points forts :**

- L'application est simple à gérer et facile à comprendre pour les non connaisseurs.
- Diversité des fonctionnalités : Cette application offre plusieurs fonctionnalités telles que la gestion des différents acteurs, la gestion des rapports des séances, la comptabilité, etc.

- **Les points faibles :**

- L'absence d'une démo (il n'existe que des captures ainsi qu'une description)
- Absence d'une version mobile.
- Cette application n'est dédiée qu'à l'administrateur et non au moniteur ou au candidat.

- **Sur l'échelle internationale :**

- **Drivescout⁴ :**

C'est un site web et une application mobile qui permettent de faciliter les différentes tâches d'une auto-école.

Voici ci-dessous la figure 5 qui présente des captures des interfaces web et mobile de cette application.

⁴ <https://drivescout.com/>

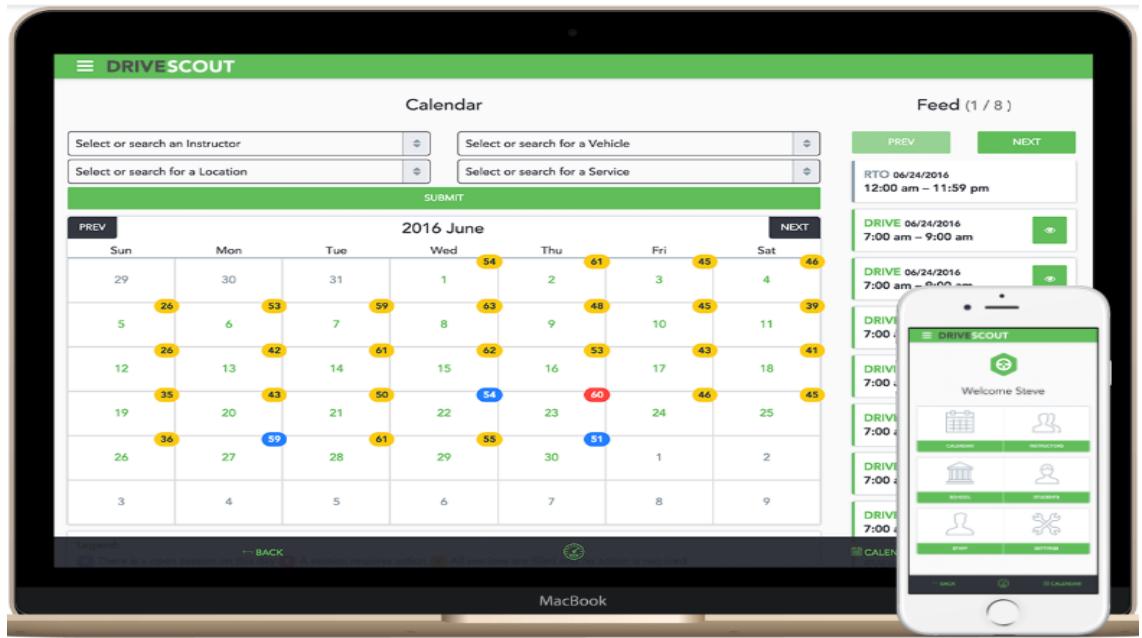


Figure 5 Interfaces d'accueil des différentes applications mobile et web

- **Les points forts :**

- Cette application facilite la gestion du temps tout en garantissant l'élimination de la duplication des données grâce à un système intégré.
- La possibilité d'effectuer des transactions de paiement en ligne en toute sécurité.
- La plateforme possède une très bonne ergonomie.
- Diversité des fonctionnalités : Cette application offre des fonctionnalités différentes telles que la gestion des différents acteurs, la gestion des plannings, la comptabilité, etc.

- **Les points faibles :**

- Absence de la géolocalisation des différents véhicules.
- L'expérience de l'utilisateur est un peu difficile : Si l'utilisateur n'est pas un connaisseur en informatique, il aura des difficultés à visualiser le site.

- **RDV 360 auto-école⁵:**

C'est un site web de gestion d'une auto-école. Il est dédié aux différents acteurs (Clients, Moniteurs, Directeurs, secrétaire).

Voici ci-dessous la figure 6 qui présente le site web de cette solution.

⁵ <https://logiciel-auto-ecole.rdv360.com/>

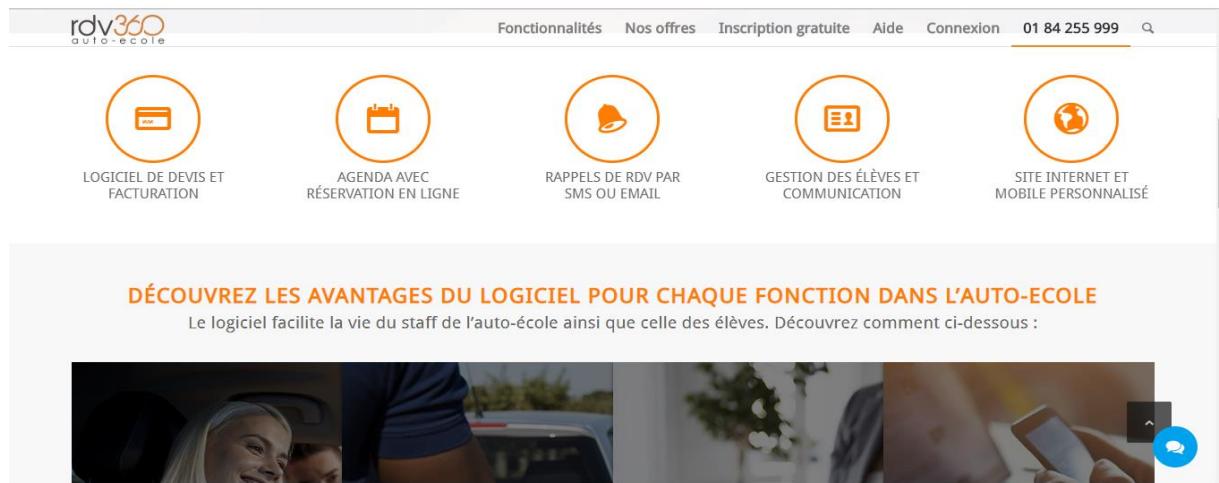


Figure 6 Interface du site vitrine

- **Les points forts :**

- Cette solution offre une démonstration gratuite.
- Le design est simple, sans surcharge, les éléments importants sont mis en valeur.
- Le site dispose d'un moteur de recherche interne.
- L'existence de deux versions (version gratuite et une version premium).

- **Les points faibles :**

- Limitation des services : Il n'y a pas des services pour la géolocalisation des véhicules ni des services de facturation et de comptabilité.

- **Driving Instructor Software⁶ :**

L'application “Driving Instructor Software” est une application mobile pour la gestion des auto-écoles.

Voici ci-dessous la figure 8 qui présente cette solution mobile.



Figure 7 L'application mobile sur le Play Store

⁶ <https://play.google.com/store/apps/details?id=driving.instructor.app>

- **Les points forts :**
 - Interface facile à comprendre.
 - Diversité des fonctionnalités : Cette application offre des fonctionnalités différentes telles que la gestion des différents acteurs, gestion des plannings, etc.

- **Les points faibles :**
 - Limitation des services : Il n'y a pas des services pour la géolocalisation des véhicules ni des services de facturation et de comptabilité aussi bien qu'une absence des services pour les moniteurs afin de gérer leurs calendriers et contacter les clients.
 - L'absence d'un site web ou d'une application desktop reliée.
 - L'interface utilisateur ne sont pas modernes.
 - L'application n'est pas pratique à utiliser par exemple il n'y a pas un menu pour sélectionner les fonctionnalités.

2.2 Critique de l'existant

Après une observation approfondie des applications pour les auto-écoles, nous avons pu dégager les points faibles suivants :

- Une négligence des espaces dédiés aux candidats et aux moniteurs. En effet la plupart des applications existantes sur le marché sont dédiées aux administrateurs ou bien à tout type d'utilisateur sans distinction de rôle.
- L'absence d'interactions entre les différents types d'utilisateurs.
- Manque de gestion des véhicules et leurs états.
- Absence d'emploi du temps des moniteurs.

Cependant, certains sites web ou applications se caractérisent par :

- Une très bonne ergonomie.
- Le traitement rapide et facile des paiements
- Les réservations en ligne.
- La gestion des différents acteurs.
- La planification et rappels des rendez-vous.

2.3 Solution proposée

Après les recherches réalisées et une étude de l'existant, on a pu constater que les activités des auto-écoles se multiplient jour après jour et que leurs tâches deviennent de plus en plus complexes.

Pour remédier aux problèmes rencontrés, notre projet consiste en :

- La création d'une plateforme web qui englobe toutes les fonctionnalités nécessaires pour la gestion d'une auto-école. Parmi ces fonctionnalités on peut citer le système de gestion de statistiques, de planning, des moniteurs, des candidats, etc.
- La création d'une application mobile qui permet la gestion du compte et la consultation du planning pour le moniteur et l'administrateur. Le moniteur aura aussi la possibilité de gérer les rapports des séances. D'une autre part l'administrateur aura la possibilité de géolocaliser les différents véhicules au cours des séances de conduite.
- La création d'un site vitrine contenant la présentation de notre produit.

2.4 Spécification des besoins

Dans cette phase nous allons présenter les besoins fonctionnels et non fonctionnels ainsi que les différents acteurs du projet.

2.4.1 Identification des acteurs

Notre application de gestion d'une auto-école fait intervenir 4 types d'utilisateurs :

- L'administrateur : C'est l'acteur principal ayant un rôle primordial qui est la gestion des moniteurs, des candidats, des véhicules ainsi que du planning.
- Les moniteurs : Ce sont ceux qui enseignent théoriquement et pratiquement la conduite d'un véhicule.
- Les candidats : sont ceux qui vont suivre des cours de code ou également de conduite afin de pouvoir passer un examen final pour avoir le permis de conduire.
- Le visiteur : le visiteur du site vitrine proposant notre solution.

2.4.2 Besoins fonctionnels

Les besoins fonctionnels d'une application présentent les fonctionnalités offertes du système et pour répondre aux besoins des auto-écoles notre application doit offrir les fonctionnalités suivantes :

Consultation du planning :

- Agenda adapté aux auto-écoles : Cette fonction permet au candidat de consulter son planning, au moniteur de consulter en temps réel ce dernier depuis son mobile et en permettant à l'administrateur de consulter les plannings des moniteurs et des candidats.

Gestion des séances :

- Cette fonction permet à l'administrateur d'ajouter, de modifier, de supprimer et de modifier les séances.

La géolocalisation des véhicules en temps réel :

- Cette fonction a pour but de connaître en temps réel, sur une carte géographique, la position de chaque voiture.

Gestion des rapports des séances :

- Un service qui permet aux moniteurs de gérer les rapports de chaque séance de conduite de chaque candidat ainsi que les commentaires inscrits par le moniteur.

Gestion des cours et passation des tests en ligne :

- Notre application présente des cours et des tests de code en ligne pour les candidats.

L'édition d'une Factures, Gestion de la comptabilité :

- Une fonction de comptabilité et de gestion des encassemens permet d'éditionner des factures sur l'application.

- Des tableaux de bords paramétrables et statistiques permettent d'avoir un aperçu fiable de la performance de l'auto-école.

Gestion des Candidats :

- Grace à un tableau de bord, l'administrateur peut consulter la liste des candidats tout en ajoutant, modifiant ou supprimant un candidat.

Gestion des Moniteurs :

- Notre système permet à l'administrateur de consulter la liste des moniteurs tout en ajoutant, modifiant, supprimant et consultant un moniteur.

Gestion des véhicules :

- Notre application fournie à l'administrateur la possibilité de gérer et consulter rapidement les informations relatives aux véhicules de l'auto-école.

Gestion du compte :

- Cette fonction permet aux différents acteurs la possibilité de consulter leurs comptes et de modifier leurs informations.

Authentification :

- Notre application permet aux différents acteurs de se connecter et par la suite d'accéder à leurs espaces privés.

Visiter site vitrine :

- Pour avoir une vue globale sur notre application et son mode d'utilisation, notre système dispose d'un site vitrine pour les visiteurs.

Gestion des cours :

- Notre système permet à l'administrateur d'ajouter, de consulter et de supprimer des cours.

Consultation des informations des candidats :

- Notre application offre aux moniteurs la possibilité de consulter les informations des candidats.

Organisation du planning :

- Notre application permet à l'administrateur d'affecter une séance à un moniteur et un candidat de plus de modifier ou supprimer cette dernière.

2.4.2 Les besoins non fonctionnels

Les besoins non fonctionnels représentent les caractéristiques du système. Pour notre projet voici, nous décrivons les besoins non fonctionnels suivantes :

Contrainte ergonomique :

- Le design reste le point majeur qui influence l'utilisateur. Pour cette raison, toutes les interfaces de l'application doivent être simples, homogènes, conviviales et voire même apte à aider l'utilisateur à mieux gérer son espace de travail tout en offrant une navigation rapide en seulement quelques clics.

Contraintes de performance :

- Le temps de chargement des interfaces de toute l'application doit être acceptable pour une utilisation en temps réel.
- L'application doit garantir la sécurité des informations affichées en spécifiant les droits d'accès de chaque utilisateur.

Maintenabilité :

- Les différents modules de l'application doivent être faciles à maintenir. Pour ce faire, le code doit être bien structuré. Nous devrons respecter les règles de codage qui permettent d'assurer une meilleure lisibilité du code comme par exemple le nommage et l'organisation des fichiers du code source et l'utilisation des commentaires, et l'utilisation d'un style architectural.

Sécurité :

- Laravel contribue à sécuriser l'application Web en la protégeant contre les plus graves failles de sécurité. Il utilise l'algorithme de hachage Bcrypt pour générer une représentation cryptée d'un mot de passe.

Agilité :

- Facilité de modification : Reprendre plus facilement aux changements.

2.5 Méthodologie adaptée

Afin de réaliser un résultat efficace et dans le but de réaliser un produit de qualité une méthode de gestion de projet doit être mise et suivie. Ces méthodes sont diverses et différentes dont on peut citer les méthodes traditionnelles et les méthodes agiles.

Les méthodes traditionnelles sont des méthodes qui exigent l'achèvement de chaque partie pour commencer une autre. Ce type ne permet ni l'adaptation aux changements de l'environnement ni l'interaction avec le client.

Les méthodes agiles se distinguent par un cycle de développement itératif, incrémental et adaptatif. Ce type de méthodes implique le client tout au long du projet pour une meilleure visibilité qui aide à la réalisation. D'où la possibilité pour le client d'ajouter de nouveaux besoins pour éviter les risques potentiels.

Les principes des méthodes agiles et les valeurs offertes nous ont poussé à opter pour ce type de méthodes.

2.5.1 Comparaison entre les différentes méthodologies agile

Pour choisir une méthodologie qui soit le plus adaptée à notre projet, nous avons fait une comparaison entre les méthodologies agile les plus utilisées dans les entreprises.

Méthodologie	Description	Points faibles	Points forts
XP ⁷	<ul style="list-style-type: none"> Une méthodologie rapide de développement de logiciel. Son équipe de développeurs sont proches les uns des autres. 	<ul style="list-style-type: none"> Assez floue dans sa mise en œuvre. 	<ul style="list-style-type: none"> Itérative. Une communication directe entre l'équipe de développement et le client.
RUP ⁸	<ul style="list-style-type: none"> Une méthodologie de développement par itérations. 	<ul style="list-style-type: none"> Peu de place pour le code et la technologie. 	<ul style="list-style-type: none"> Itérative Incrémentale Éliminer les risques du projet (financière et technique).
SCRUM ⁹	<ul style="list-style-type: none"> Une méthodologie basée sur le découpage des projets en itérations appelées sprint. 	<ul style="list-style-type: none"> Un processus rapide qui exerce une forte pression sur l'équipe. 	<ul style="list-style-type: none"> Simple Qualité du produit mise en avant.

Tableau 1 Comparaison des méthodologies

2.5.2 Choix de la méthodologie

Après cette étude comparative, nous avons choisi la méthodologie Scrum vu sa simplicité. Scrum permet au projet de s'adapter aux besoins changeants des clients.

• La méthode agile Scrum

« Scrum est la méthodologie la plus utilisée parmi les méthodes Agiles existantes. Le terme Scrum (qui signifie mêlée) qui décrit une nouvelle approche plus rapide et flexible pour le développement de nouveaux produits. Ils comparent alors cette nouvelle méthode au rugby, le principe de base étant que l'équipe avance ensemble et soit toujours prête à réorienter le projet au fur-et-à-mesure de sa progression ». [2]

⁷ Extrême Programming c'est une des méthodes de développement agile.

⁸ Rational Unified Process est une méthode de développement promue par la société Rational Software.

⁹ Scrum est une méthode agile de gestion de projet.

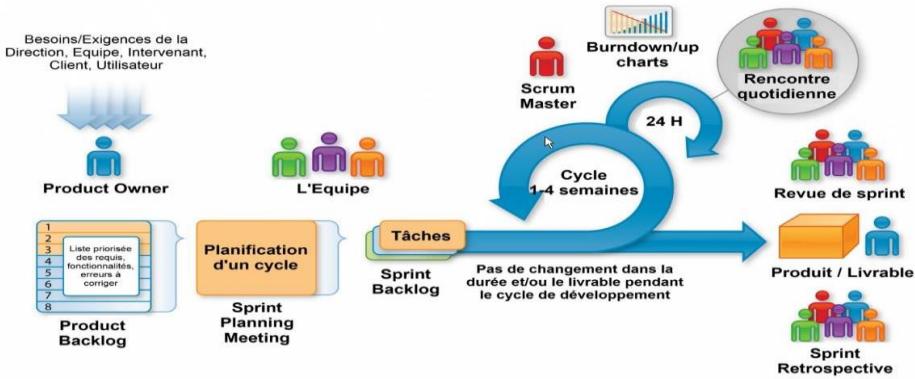


Figure 8 Vue global Scrum

- **Caractéristique de la méthode Agile Scrum**

La méthodologie Scrum se caractérise essentiellement par un cycle de développement itératif et incrémentale et se base sur l'esprit d'équipe en maximisant la coopération et la communication entre les acteurs du projet. Elle se caractérise également par les trois piliers suivants :

La transparence : Pour mieux comprendre le projet, les différents participants doivent adopter un langage commun entre eux.

L'inspection : Dans le but de détecter les déviations, des vérifications sont effectuées sur les artefacts produit régulièrement tout au long du projet.

L'adaptation : Pour corriger les déviations rencontrées pendant l'inspection, Scrum doit être adaptative par ses mélées quotidiennes ainsi que par ses réunions de sprint.

- **Les acteurs**

La méthodologie SCRUM définit trois rôles principaux :

- **Le Product Owner**

Le Product Owner est responsable d'augmenter la valeur du produit résultant du travail de l'équipe de développement. C'est la seule personne qui prend les décisions importantes sur le projet et définit l'ordre des fonctionnalités.

- **Le Scrum Master**

Le Scrum Master est responsable d'assurer la compréhension des différents membres de la théorie, des pratiques, des règles ainsi que des valeurs de Scrum.

- **L'Equipe de Développement**

L'Equipe de Développement est chargée de transformer les besoins exprimés par le Product Owner en fonctionnalités utilisables. Elle englobe plusieurs rôles tels que le développeur, l'architecte logiciel, l'analyste fonctionnel etc.

- **Les artefacts de Scrum**

- **Backlog du produit**

Le backlog de produit est une liste des fonctionnalités attendues d'un produit. Il englobe la totalité des éléments d'un travail dont l'équipe a besoin. Cette liste est classée par priorité

définit par le Product Owner. Le backlog du produit se trouve dans la phase de préparation appelée aussi Sprint 0.

- **Sprint backlog**

Le sprint backlog est une partie de backlog du produit qui doit être produit dans un sprint donné. Il se présente sous la forme d'un tableau tout comme le backlog du produit.

- **L'incrément**

L'incrément est constitué par les éléments du produit terminés durant un sprint et le sprint précédent.

- **Les évènements**

- **Le sprint**

Le sprint est une durée de temps d'une semaine à un mois au maximum. C'est la période que l'équipe SCRUM délivre un incrément.

- **Sprint planification meeting**

La planification du sprint se passe en deux phases. Dans la première phase, l'équipe détermine ce qui sera développé durant ce sprint ainsi que la durée nécessaire pour la réalisation de ce dernier. Dans la deuxième phase, l'équipe précise comment sera la réalisation de ce sprint.

- **Dailly meeting**

Dailly meeting sont des réunions quotidiennes qui ne durent pas plus que 15 min.

- **Sprint Review Meeting**

La revue de sprint sont des réunions à la fin de chaque sprint. Ces réunions ont pour but de valider l'incrément de produit et d'essayer de faire des améliorations pour le fonctionnement de l'équipe.

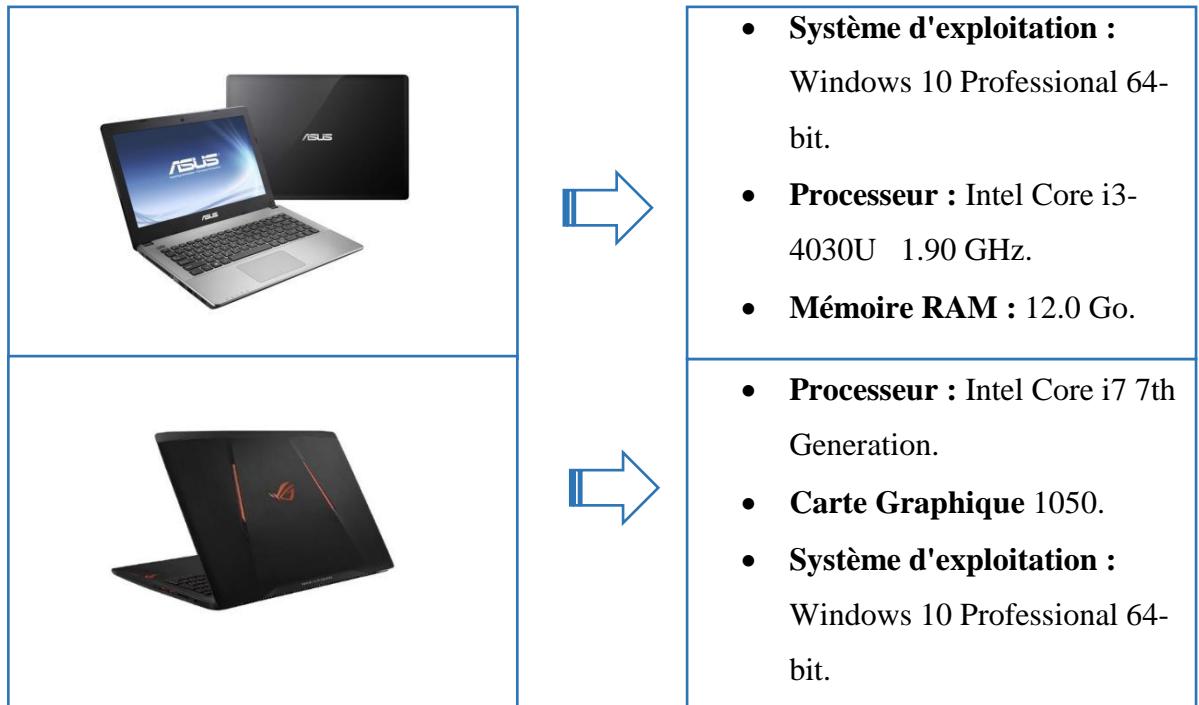
- **Sprint Rétrospective Meeting**

La rétrospective du sprint a pour objectif l'adaptation des changements survenus au cours d'un sprint. L'équipe de développement essaie d'améliorer les éléments du processus de développement lors du sprint suivant.

2.6 Environnement de développement

2.6.1 Environnement matériel

Notre application a été développée sur deux ordinateurs possédant les caractéristiques suivantes :



2.6.2 Environnement logiciel

- **Laravel :**

Laravel est un Framework web open-source écrit en PHP. Il est disponible sous Windows, Linux et Mac os. En 2016, Laravel est devenu le projet PHP le mieux noté de GitHub. [3]



- **Angular**

Angular est un Framework open-source géré par Google et basé sur Type Script qui permet un développement plus stable et rapide. Ce dernier est une réécriture complète du Framework AngularJS. Il permet de développer des applications dynamiques et immersives. [4]



- **Android Studio**

Android Studio est un environnement de développement développé par google et écrit en java, pour créer des applications mobiles Android. Il est disponible sous Windows, Linux et Mac os. Pour tester ses applications, il présente un émulateur intégré permettant de faire marcher un système Android virtuel sur un ordinateur. [5]



- **Adobe XD**

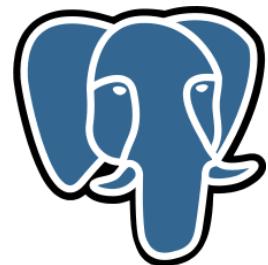
Adobe XD est un outil pour la conception et le prototypage d'interface animée pour les applications web et mobiles. Adobe XD est développé et publié par Adobe Inc.

Il présente plusieurs fonctionnalités permettant un design créatif. En plus il est livré avec des kits déjà intégrés d'interface pour Apple iOS, Microsoft Windows et Google Material. [6]



- **PostgreSQL 11**

PostgreSQL est un système libre de gestion de bases de données relationnelles et objet. La principale caractéristique de ce SGBDRO est l'utilisation des types de données composés. PostgreSQL peut stocker plus de types de données que les types simples. De plus, il laisse aux utilisateurs la possibilité de créer des types, des fonctions et d'utiliser l'héritage. [7]



- **Firebase**

Firebase est une plateforme de développement d'application web et mobile. Il offre l'hébergement NoSQL, des bases de données, des notifications, des services, etc. [8]



- **Microsoft Word**

Microsoft Word est un logiciel de traitement de texte développé et publié par Microsoft. C'est l'un des logiciels les plus utilisés au monde, tant par le grand public que par le monde professionnel. Il permet aux utilisateurs de personnaliser leurs documents et même de corriger la grammaire et l'orthographe automatiquement. [9]



- **Trello**

Trello est un outil de gestion de projets en ligne gratuit, simple et modulable. Deux versions de Trello sont disponibles, une version basique gratuite et une version qui est payante qui permet d'obtenir des services supplémentaires. Il permet l'organisation des projets par un tableau composé de trois colonnes qui sont to do, doing et enfin done ce qui permet de voir le déroulement du projet. [10]



- **Slack**

Slack est une plateforme de communication collaborative ainsi qu'un logiciel de gestion de projets. Slack dispose de plusieurs clients natifs sur les plateformes mobile iOS et Android ainsi que sur MacOs, Windows, linux et via un navigateur internet. [11]



- **Microsoft Visio**

Microsoft Visio est un logiciel de diagrammes développé par Windows. Ce dernier ne peut s'exécuter que dans les machines avec un système d'exploitation Windows. Les graphiques utilisés par Visio pour les diagrammes sont vectoriels. [12]



- **Postman**

Postman est un outil qui permet de créer et de tester les requêtes http, de les stocker dans un historique afin de pouvoir les rejouer, mais surtout de les organiser en collections. [13]



- **GitLab**

GitLab est une plateforme de développement collaboratif qui permet de stocker de code en ligne et le développement collaboratif de projet volumineux. Il offre la possibilité aux utilisateurs d'inspecter le code précédent et d'y revenir en cas de problèmes imprévus. [14]



- **Bootstrap**

Bootstrap est une collection d'outils utiles à la création du design de sites et d'application web. C'est un ensemble qui contient des codes HTML, CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions Javascript en option. [15]



2.7 Langages utilisés

Dans cette partie, nous allons présenter les langages que nous avons utilisés pour notre projet.

2.7.1 Langage de développement

- HTML 5, CSS, TypeScript



HTML5 est un langage utilisé pour le développement web. CSS 3 et HTML travaillent en parallèle puisque le CSS permet de faire l'animation des éléments HTML.

TypeScript est un langage de développement qui a pour but d'améliorer et de sécuriser le code JavaScript.

- **Java**

Java est un langage informatique orienté objets rapide, sécurisé et fiable.



- **PHP**

PHP est un langage de programmation utilisé pour développer des pages web dynamiques via un serveur HTTP.



2.7.2 Langage de modélisation

Pour la modélisation de notre projet nous avons choisi UML¹⁰ comme langage de modélisation. UML est un langage riche de modélisation commun dans le monde. Son apparition est due à certaines limites rencontrées par les méthodes de modélisation classiques comme Merise.

2.7.3 Architecture

Notre projet se compose d'une partie web développée avec Angular et d'une partie mobile développée avec Android. Ces deux dernières consomment des web services développés avec Laravel.

La Figure ci-dessous représente l'architecture de notre projet

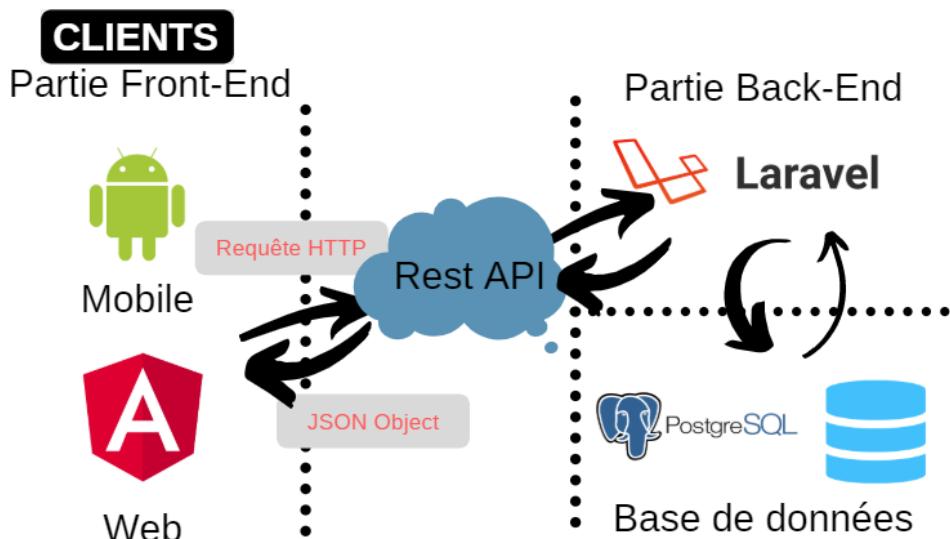


Figure 9 Architecture du projet

¹⁰ Unified Modeling Language est un langage de modélisation.

- **Partie web**

- **Architecture MVVM**

L'architecture MVVM (Model View ViewModel) est un design pattern. Il s'agit d'un modèle conceptuel qui offre à ses utilisateurs une structuration du code. Ce modèle précise des fortes contraintes de développement afin de rendre le code plus clair et maintenable. Ce modèle permet, de séparer la vue de la logique et de l'accès aux données.

On peut définir alors les trois parties du modèle MVVM comme suit :

➤ **Modèle**

Le modèle est un ensemble de classes qui va convertir les données de l'application qui proviennent le plus souvent d'une base de données ou d'un service web. Il ne se charge que du stockage des données. C'est la vue qui se charge de l'affichage.

➤ **Vue**

La vue est la partie chargée de la disposition et de l'apparence de ce que l'utilisateur voit. Il se charge principalement de la représentation du modèle et reçoit les interactions des utilisateurs.

➤ **Modèle de vue**

La vue-modèle est l'intermédiaire entre le modèle et la vue. Elle s'occupe de récupérer les données depuis le modèle. Elle présente une abstraction de la vue qui expose des propriétés ainsi que des commandes publiques. MVVM possède un classeur (binder) automatisant la communication entre la vue et ses propriétés liées dans le modèle de vue.

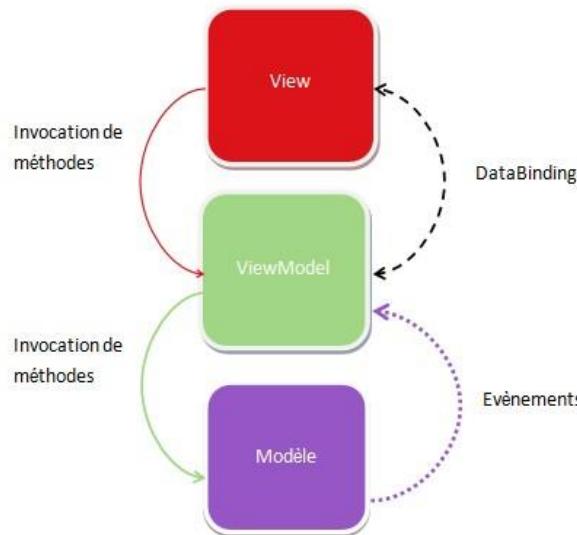


Figure 10 architecture MVVM

- **Partie Mobile**

- **Architecture MVC**

Nous utiliserons le pattern architectural Modèle Vue Contrôleur (MVC) dans notre architecture choisie, présenté par la figure suivante.

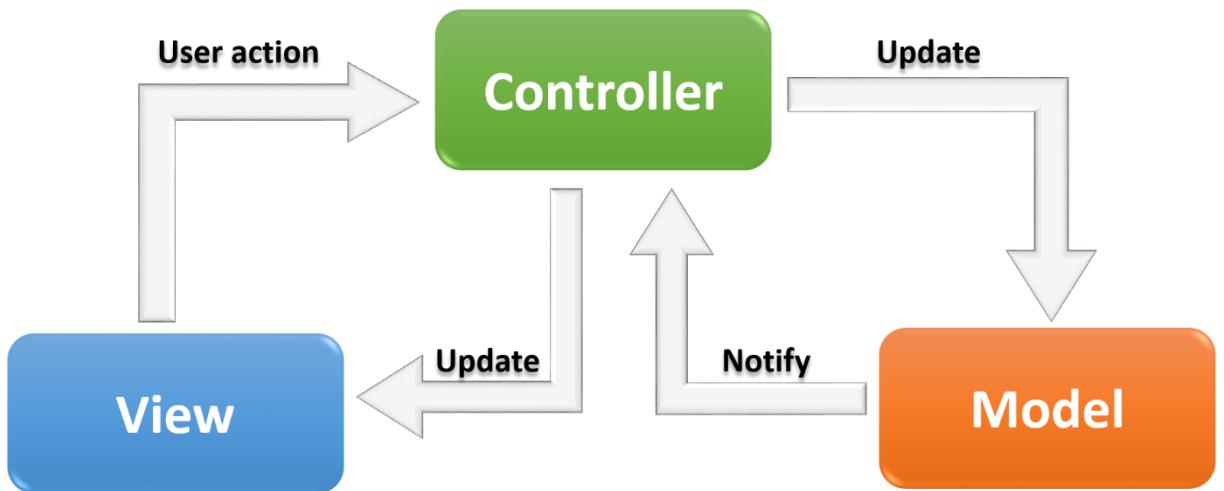


Figure 11 architecture MVC

Le modèle MVC se décompose en trois couches qui sont comme suit :

➤ **Modèle**

Il contient les données à afficher, c'est le lien entre le contrôleur et la base de données où les données sont stockées. Dans le cas d'une programmation orienté objet l'exploitation des données se fait à travers des classes.

➤ **Vue**

Cela correspond aux fichiers Layout en .XML. C'est la vue qui est chargée de tout ce qui est visible.

➤ **Contrôleur**

Il correspond aux Activities (ou Fragments). Il se charge du traitement des actions des utilisateurs et de la mise à jour.

III. Découpage du projet

3.1 Pilotage du projet avec SCRUM

Notre projet est constitué de plusieurs membres qui sont :

Mr Ghozzi Ayoub qui est le « Product Owner », Mlle Fatnassi Oumayma qui se présente comme « Scrum Master » et enfin deux membres, Ben Othman Oumayma et Chebbi Amira, qui sont chargés de la réalisation du projet.

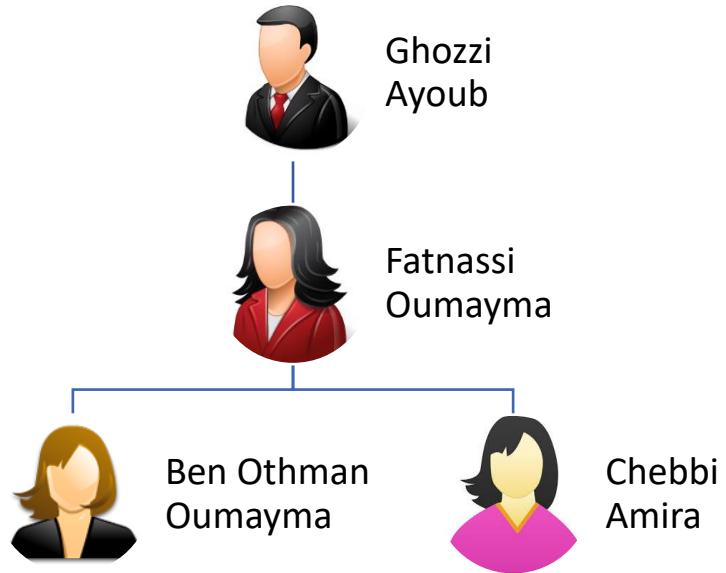


Figure 12 Hiérarchie de l'équipe

3.2 Backlog du produit

Id	Fonctionnalité	User Story ID	User Story			Priorité
			En tant que	Je souhaite		
1	Gérer les véhicules	1.1	Administrateur	Ajouter un véhicule	un	Fort
		1.2	Administrateur	Modifier un véhicule	un	Fort
		1.3	Administrateur	Supprimer un véhicule	un	Fort
		1.4	Administrateur	Consulter un véhicule	un	Fort
		1.5	Administrateur	Consulter la liste des véhicules		Fort
2	Gérer le planning	2.1	Administrateur	Ajouter une séance	une	Fort
		2.2	Administrateur	Modifier une séance	une	Fort
		2.3	Administrateur	Supprimer une séance	une	Fort
		2.4	Administrateur	Consulter une séance	une	Fort
		2.5	Administrateur	Consulter la liste des séances		Fort
3	Gérer les moniteurs	3.1	Administrateur	Ajouter un moniteur	un	Fort
		3.2	Administrateur	Modifier un moniteur	un	Fort

		3.3	Administrateur	Supprimer un moniteur	Fort
		3.4	Administrateur	Consulter un moniteur	Fort
		3.5	Administrateur	Consulter la liste des moniteurs	Fort
4	Gérer les rapports des séances	4.1	Moniteur	Ajouter un rapport	Fort
		4.2	Moniteur	Modifier un rapport	Fort
		4.3	Moniteur	Consulter un rapport	Fort
		4.4	Moniteur	Consulter la liste des rapports	Fort
5	Géolocaliser les véhicules	5.1	Administrateur	Géolocaliser les véhicules	Fort
6	Gérer la comptabilité	6.1	Administrateur	Enregistrer un gain	Moyen
		6.2	Administrateur	Modifier un gain	Moyen
		6.3	Administrateur	Supprimer un gain	Moyen
		6.4	Administrateur	Afficher des statistiques	Moyen
		6.5	Administrateur	Enregistrer une dépense	Moyen
		6.6	Administrateur	Modifier une dépense	Moyen
		6.7	Administrateur	Supprimer une dépense	Moyen
		6.8	Administrateur	Etablir une facture	Moyen
		6.9	Administrateur	Supprimer une facture	Moyen
		6.10	Administrateur	Modifier une facture	Moyen
7	Consulter planning	7.1	Moniteur, candidat	Consulter planning	Moyen
8	S'authentifier	8.1	Administrateur, Moniteur et Candidat	M'authentifier pour accéder à mon espace	Moyen
9	Consulter informations des candidats	9.1	Moniteur	Consulter informations des candidats	Moyen

10	Consulter le site vitrine	10.1	Visiteur	Consulter le site vitrine	Moyen
11	Gérer les candidats	11.1	Administrateur	Ajouter un candidat	Moyen
		11.2	Administrateur	Modifier un candidat	Moyen
		11.3	Administrateur	Supprimer un candidat	Moyen
		11.4	Administrateur	Consulter un candidat	Moyen
		11.5	Administrateur	Consulter la liste des candidats	Moyen
12	Gérer les cours	12.1	Administrateur	Ajouter un cours	Moyen
		12.2	Administrateur	Supprimer un cours	Moyen
		12.3	Administrateur	Consulter un cours	Moyen
		12.4	Administrateur	Consulter la liste des cours	Moyen
13	Gérer mon compte	13.1	Administrateur, Moniteur et Candidat	Modifier mon compte	Moyen
		13.2	Administrateur, Moniteur et Candidat	Consulter mon compte	Moyen
14	Consulter cours	14.1	Candidat	Consulter un cours	Faible
15	Passer les tests en ligne	15.1	Candidat	Passer les tests en ligne	Faible

Tableau 2 Backlog du produit

3.3 Planification des sprints

Après avoir élaboré le Product Backlog, un découpage en sprint est nécessaire pour faire tenir l'ordre du travail. Pour ce fait, nous avons opté pour deux sprints successifs.

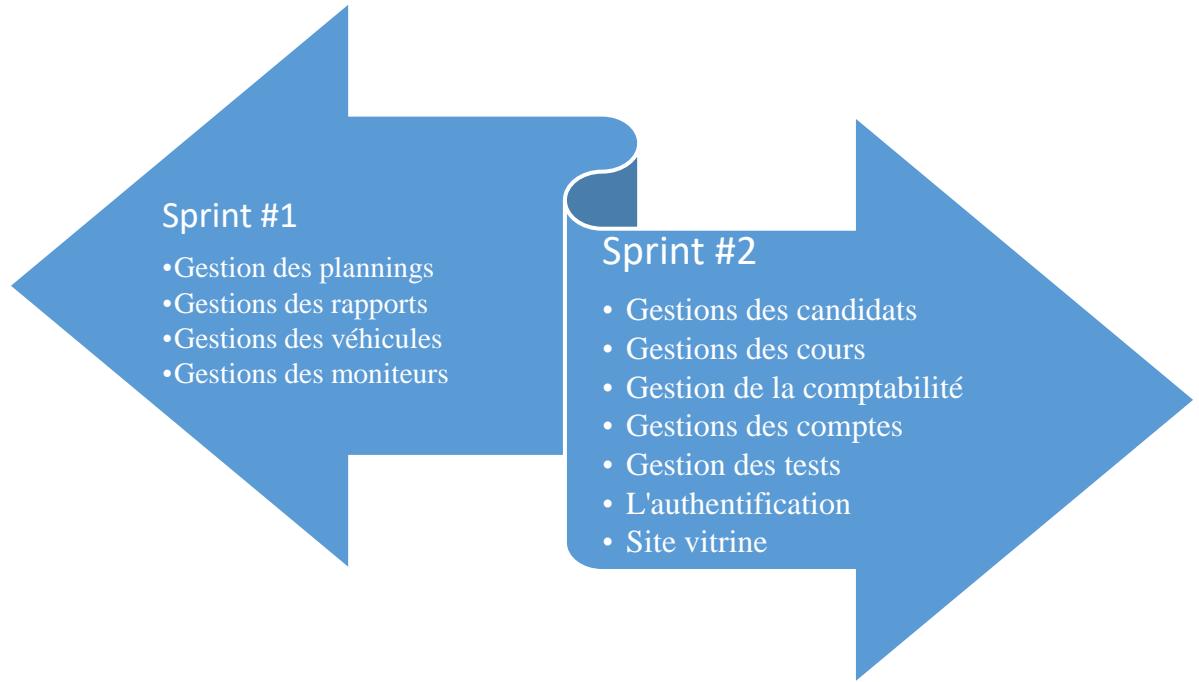


Figure 13 Découpage des sprints

3.4 Le planning du projet

Pour la planification de notre projet, voici le planning que nous avons suivi :

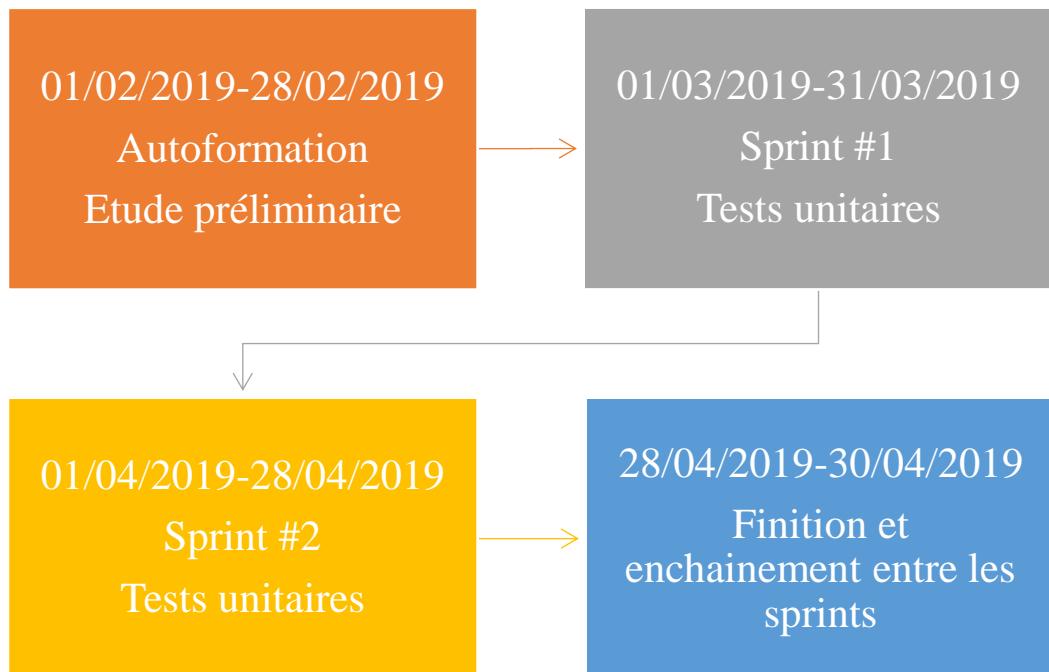


Figure 14 Planning du déroulement du projet

3.5 Outil de Gestion de projet : Trello¹¹

Pour la gestion de notre projet nous avons choisi Trello comme logiciel de gestion de projet.
Voici une capture de notre tableau Trello.

La capture d'écran montre un tableau Trello avec les suivantes colonnes :

- To Do**: Sprint 2 - Angular (à 31 mars), Sprint 2 - Android (à 25 avr.), Sprint 1 - Test Mobile (à 17 mars), Sprint 1 - Angular (à 14 avr.), Sprint 2 - Test Mobile (à 28 avr.), Sprint 3 - Back-office.
- Doing**: Sprint 2 - Test Web (à 7 avr.), Sprint 1 - Test Web (à 21 mars), Sprint 2 - Design (à 17 mars), Sprint 1 - Laravel (à 17 mars).
- Done**: Sprint 1 - Design (terminé le 3 mars), Sprint 1 - Test Web (terminé le 21 mars), Sprint 2 - Design (terminé le 17 mars), Sprint 1 - Laravel (terminé le 28/28), Sprint 2 - Laravel (terminé le 31 mars), Sprint 2 - Laravel (terminé le 52/52).

Le menu déroulant "Menu" à droite propose des options telles que "À propos de ce tableau", "Changer le fond d'écran", "Filtrer les cartes", "Stickers", "Plus", "Power-Ups", "Activité" et "Ajouter un Power-up...".

Figure 15 Capture de Trello

3.6 Prototypes d'interfaces

Au cours de cette section, nous allons présenter quelques prototypes des interfaces graphiques qui nous ont permis à bien comprendre les besoins du client et donc atteindre sa satisfaction. En effet, ces interfaces graphiques permettent au client de mieux s'exprimer et d'être plus interactif. Ci-dessous quelques interfaces réalisées avec l'outil Adobe XD¹².

Le prototype d'interface agenda affiche le mois de Novembre 2019. Il comprend :

- Un menu en haut avec "Agenda", "Candidatq", "Caisse", "Cours" et "Déconnexion".
- Une photo de profil sur la gauche.
- Un calendrier pour Novembre 2019 avec les dates 1 à 31.
- Une grille horizontale pour l'affichage des événements, divisée en heures (08:00, 09:00, 10:00, 11:00) et jours (Lundi à Dimanche).
- Des boutons pour naviguer entre "Jour" et "Semaine".
- Des boutons pour "Aujourd'hui" et pour naviguer dans le mois.
- Une section "Notes" à droite.

Figure 16 Prototype de l'interface agenda

¹¹ Trello est un outil de gestion en ligne.

¹² Adobe XD est un outil d'Adobe Systems pour la conception et la création de prototypes.

<https://www.adobe.com/products/xd.html>

Figure 17 Prototype de l'interface gestion des moniteurs

Figure 18 Prototype de l'interface ajouter moniteur

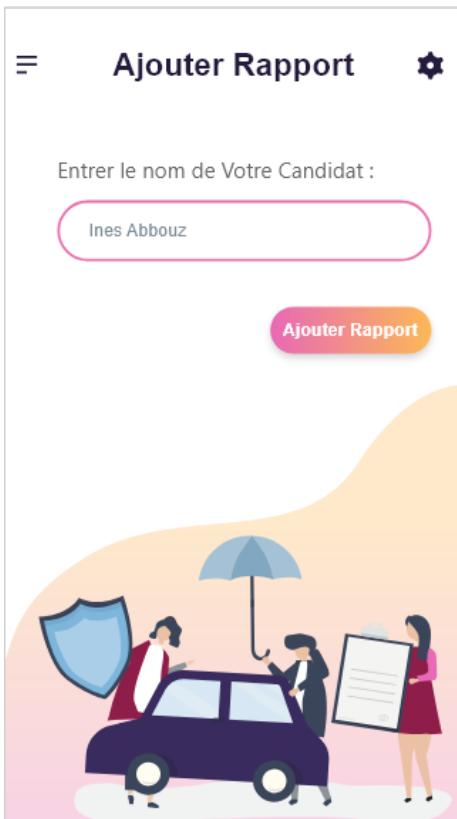


Figure 20 Prototype de l'interface ajouter rapport dans l'application mobile

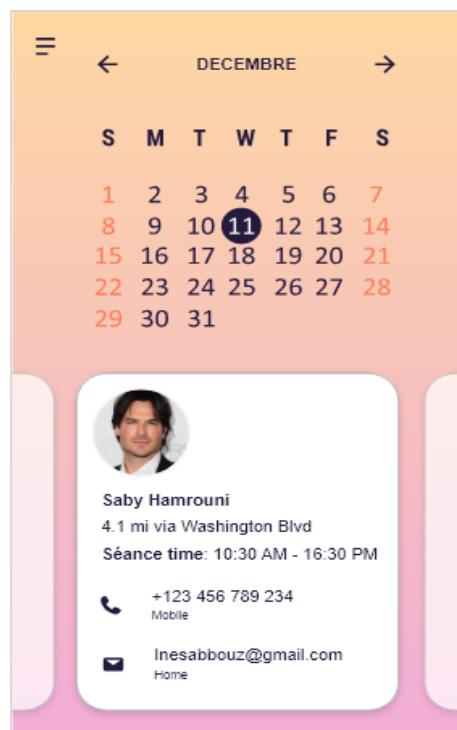


Figure 21 Prototype de l'interface consulter planning

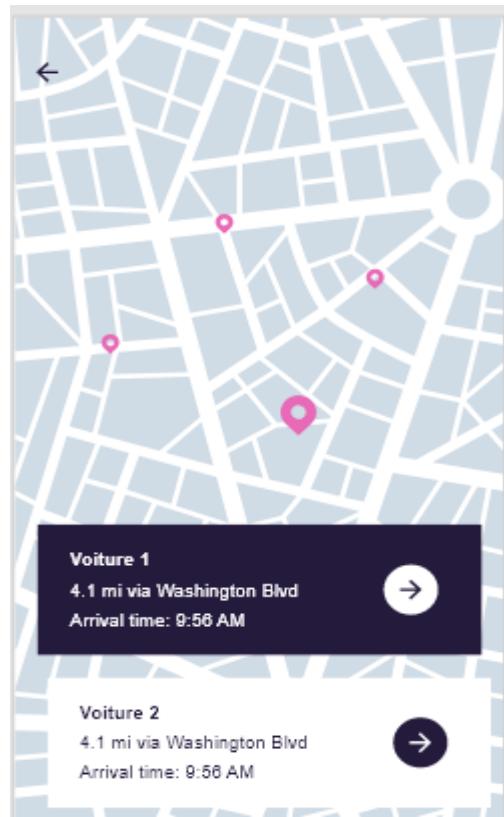


Figure 19 Prototype de l'interface géolocalisation des véhicules

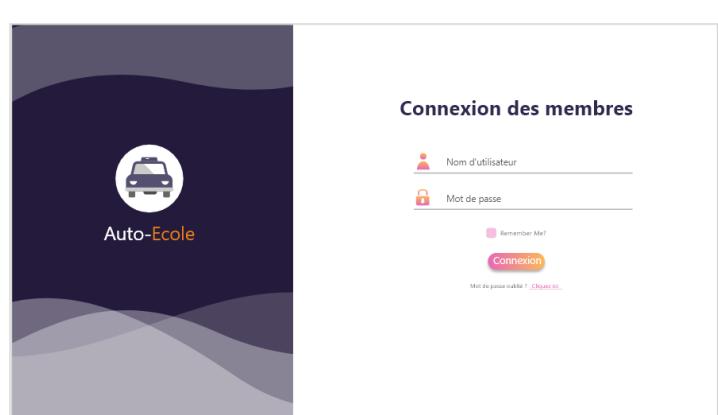


Figure 22 Prototype de l'interface s'authentifier

➤ Prototypes d'interface site vitrine

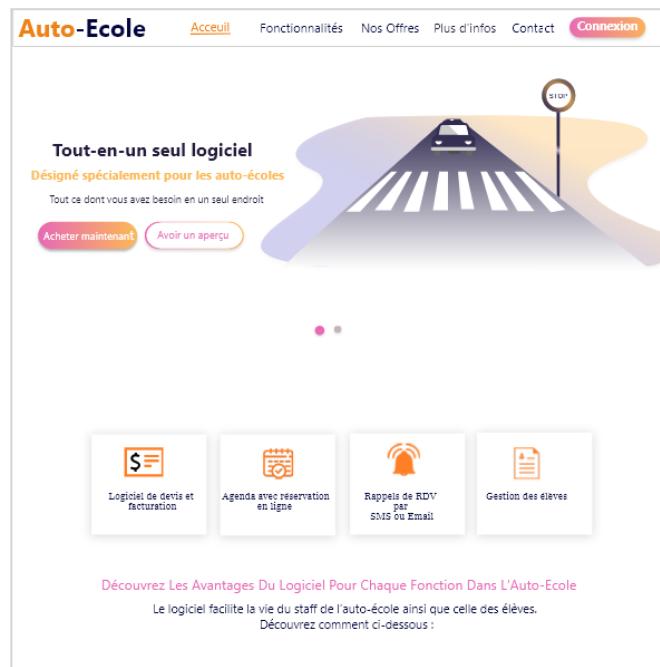


Figure 24 Prototype de page d'accueil 1



Figure 23 Prototype de l'interface d'accueil 2

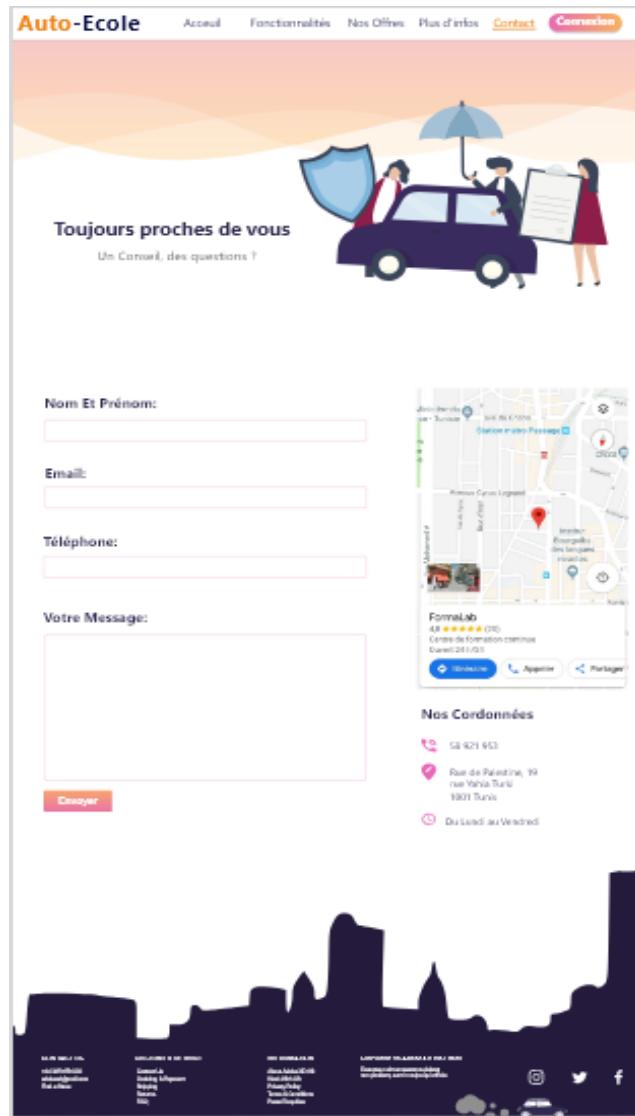


Figure 25 Prototype de l'interface de contact du site vitrine

IV. Conclusion

Dans ce chapitre, nous avons présenté en premier lieu l'environnement où nous avons effectué notre stage de fin d'études. Ensuite, nous avons fait une étude de l'existant et nous avons présenté la solution proposée. Au sein de ce chapitre, nous avons choisi et présenté la méthodologie adéquate ainsi que l'environnement de développement. Il s'agit d'une étude préliminaire qui nous a permis de bien le comprendre. Nous avons présenté les différents acteurs du système ainsi que les besoins fonctionnels et non fonctionnels suivis par une élaboration du Backlog produit. Pour finir, nous avons élaboré le découpage du projet et la planification des sprints. Dans le chapitre suivant, nous présenterons le premier sprint.

Chapitre 2

Premier Sprint

Introduction

Après avoir terminé l'étape de l'analyse des besoins de notre système informatique, nous pouvons maintenant commencer le développement de la première partie de notre projet qui consiste à réaliser le premier sprint.

Dans ce chapitre, nous allons présenter le backlog du premier sprint ainsi que l'analyse et la conception de cette partie de l'application.

I. Spécification des besoins

1.1 Le sprint backlog

Le backlog du premier sprint, présenté dans le tableau ci-dessous, énonce les différentes fonctionnalités de notre projet.

User story ID	User Story	Id Tache	Tache
3.1	En tant qu'administrateur je souhaite ajouter un moniteur.	3.1.1	Réaliser la modélisation et la conception de la fonctionnalité “Ajouter moniteur”.
		3.1.2	Développement du cas “Ajouter moniteur”.
		3.1.3	Test du cas “Ajouter moniteur”.
3.2	En tant qu'administrateur je souhaite modifier un moniteur.	3.2.1	Réaliser la modélisation et la conception de la fonctionnalité “Modifier moniteur”.
		3.2.2	Développement du cas “Modifier moniteur”.
		3.2.3	Test du cas “Modifier moniteur”.
3.3	En tant qu'administrateur je souhaite supprimer un moniteur.	3.3.1	Réaliser la modélisation et la conception de la fonctionnalité “Supprimer moniteur”.
		3.3.2	Développement du cas “Supprimer moniteur”.
		3.3.3	Test du cas “Supprimer moniteur”.
3.4	En tant qu'administrateur je souhaite consulter un moniteur.	3.4.1	Réaliser la modélisation et la conception de la fonctionnalité “Consulter moniteur”.

		3.4.2	Développement du cas “Consulter moniteur”.
		3.4.3	Test du cas “Consulter moniteur”.
3.5	En tant qu'administrateur je souhaite consulter la liste des moniteurs.	3.5.1	Réaliser la modélisation et la conception de la fonctionnalité “Consulter la liste des moniteurs”.
		3.5.2	Développement du cas “Consulter la liste des moniteurs”.
		3.5.3	Test du cas “Consulter la liste des moniteurs”.
1.1	En tant qu'administrateur je souhaite ajouter un véhicule.	1.1.1	Réaliser la modélisation et la conception de la fonctionnalité “Ajouter un véhicule”.
		1.1.2	Développement du cas “Ajouter un véhicule”.
		1.1.3	Test du cas “ Ajouter un véhicule”.
1.2	En tant qu'administrateur je souhaite modifier un véhicule.	1.2.1	Réaliser la modélisation et la conception de la fonctionnalité “Modifier un véhicule”.
		1.2.2	Développement du cas “Modifier un véhicule”.
		1.2.3	Test du cas “ Modifier un véhicule”.
1.3	En tant qu'administrateur je souhaite supprimer un véhicule.	1.3.1	Réaliser la modélisation et la conception de la fonctionnalité “Supprimer un véhicule”.
		1.3.2	Développement du cas “Supprimer un véhicule”.
		1.3.3	Test du cas “ Supprimer un véhicule”.
1.4	En tant qu'administrateur je souhaite consulter un véhicule.	1.4.1	Réaliser la modélisation et la conception de la fonctionnalité “Consulter un véhicule”.

		1.4.2	Développement du cas “Consulter un véhicule”.
		1.4.3	Test du cas “ Consulter un véhicule”.
5.1	En tant qu'administrateur je souhaite géolocaliser les véhicules.	5.1.1	Réaliser la modélisation et la conception de la fonctionnalité “Géolocaliser les véhicules ”.
		5.1.2	Développement du cas “Géolocaliser les véhicules ”.
		5.1.3	Test du cas “ Géolocaliser les véhicules ”.
1.5	En tant qu'administrateur je souhaite consulter la liste des véhicules.	1.5.1	Réaliser la modélisation et la conception de la fonctionnalité “Consulter la liste des véhicules”.
		1.5.2	Développement du cas “Consulter la liste des véhicules”.
		1.5.3	Test du cas “ Consulter la liste des véhicules”.
4.1	En tant que moniteur je souhaite ajouter un rapport.	4.1.1	Réaliser la modélisation et la conception de la fonctionnalité “Ajouter un rapport”.
		4.1.2	Développement du cas “Ajouter un rapport”.
		4.1.3	Test du cas “ Ajouter un rapport”.
4.2	En tant que moniteur je souhaite modifier un rapport.	4.2.1	Réaliser la modélisation et la conception de la fonctionnalité “Modifier un rapport”.
		4.2.2	Développement du cas “Modifier un rapport”.
		4.2.3	Test du cas “ Modifier un rapport”.
4.3	En tant que moniteur je souhaite consulter un rapport.	4.3.1	Réaliser la modélisation et la conception de la fonctionnalité “Consulter un rapport”.

		4.3.2	Développement du cas “Consulter un rapport”.
		4.3.3	Test du cas “ Consulter un rapport”.
4.4	En tant que moniteur je souhaite consulter la liste des rapports.	4.4.1	Réaliser la modélisation et la conception de la fonctionnalité “Consulter la liste des rapports”.
		4.4.2	Développement du cas “Consulter la liste des rapports”.
		4.4.3	Test du cas “ Consulter la liste des rapports”.
2.1	En tant qu'administrateur je souhaite ajouter une séance.	2.1.1	Réaliser la modélisation et la conception de la fonctionnalité “Ajouter une séance”.
		2.1.2	Développement du cas “ Ajouter une séance ”.
		2.1.3	Test du cas “Ajouter une séance”.
2.2	En tant qu'administrateur je souhaite modifier une séance.	2.2.1	Réaliser la modélisation et la conception de la fonctionnalité “Modifier une séance”.
		2.2.2	Développement du cas “ Modifier une séance ”.
		2.2.3	Test du cas “Modifier une séance ”.
2.3	En tant qu'administrateur je souhaite supprimer une séance.	2.3.1	Réaliser la modélisation et la conception de la fonctionnalité “Supprimer une séance”.
		2.3.2	Développement du cas “ supprimer une séance ”.
		2.3.3	Test du cas “ supprimer une séance ”.
2.4	En tant qu'administrateur je souhaite consulter une séance.	2.4.1	Réaliser la modélisation et la conception de la fonctionnalité “Consulter une séance”.

		2.4.2	Développement du cas “Consulter une séance”.
		2.4.3	Test du cas “Consulter une séance”.
2.5	En tant qu'administrateur je souhaite consulter le planning.	2.5.1	Réaliser la modélisation et la conception de la fonctionnalité “Consulter le planning”.
		2.5.2	Développement du cas “Consulter le planning”.
		2.5.3	Test du cas “Consulter le planning”.

Tableau 3 Backlog du sprint 1

II. Analyse des cas du sprint 1

2.1. Classification des cas d'utilisation par acteur

Le tableau ci-dessous présente la classification des cas d'utilisation par acteur.

Acteur	Cas d'utilisation
Administrateur	Gérer les moniteurs
Administrateur	Gérer les véhicules
Administrateur	Gérer le planning
Administrateur	Géolocaliser les véhicules
Moniteur	Gérer les rapports

Tableau 4 Classification des cas d'utilisation par acteur

2.2. Diagramme de cas d'utilisation du premier sprint

La figure ci-dessous représente le diagramme de cas d'utilisation global du premier sprint. L'administrateur aura le droit, après l'utilisation d'un mot de passe et d'un identifiant, de gérer le planning, les véhicules ainsi que les moniteurs. Ces derniers peuvent aussi grâce à l'application mobile gérer les rapports des séances.

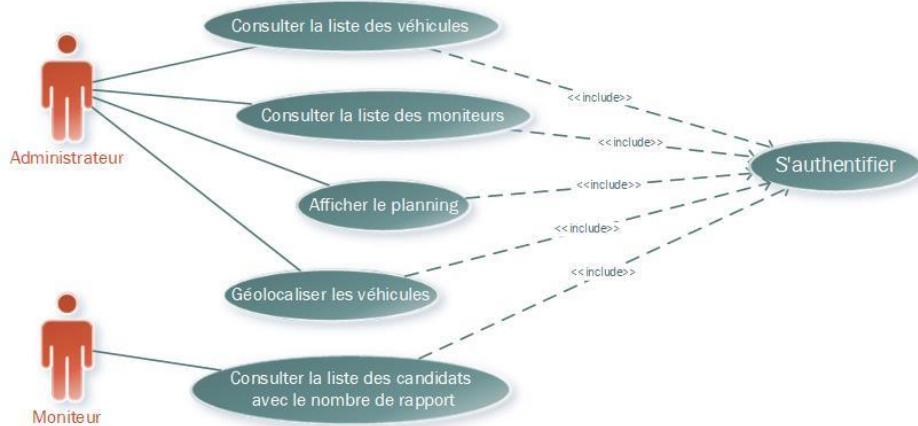


Figure 26 Diagramme de cas d'utilisation du Sprint 1

2.3. Description des cas d'utilisation du premier sprint

Au cours de cette partie nous allons mettre en place le raffinement de nos cas d'utilisation ainsi qu'une description des différents scénarios qui peuvent se présenter afin qu'ils soient plus clairs et compréhensibles.

2.3.1. Analyse du cas d'utilisation « Consulter la liste des moniteurs »

- Raffinement du cas d'utilisation « Consulter la liste des moniteurs »

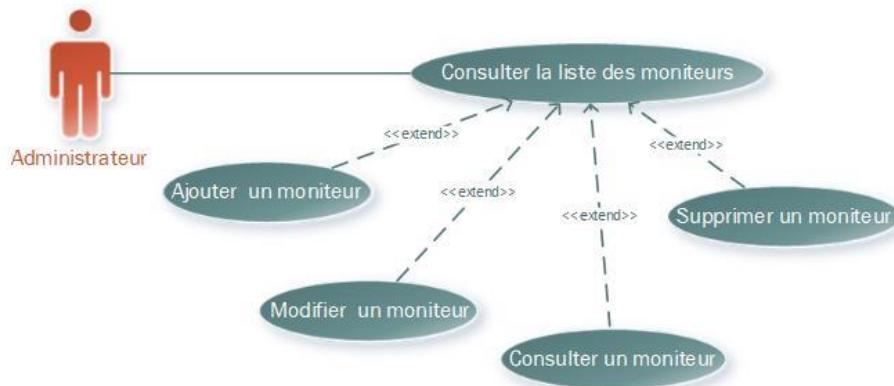


Figure 27 Diagramme de cas d'utilisation « Consulter la liste des moniteurs »

- Description textuelle du cas d'utilisation « Consulter la liste des moniteurs »

Cas D'utilisation

Consulter la liste des moniteurs.

Acteur	Administrateur.
Pré condition	L'administrateur s'est authentifié.
Post condition	Liste des moniteurs consultée.
Scénario principal	<ol style="list-style-type: none"> 1. L'administrateur clique sur le bouton "Moniteurs". 2. Le système affiche la liste des moniteurs.
Extension	- L'administrateur peut modifier un moniteur.

- L'administrateur peut supprimer un moniteur.
- L'administrateur peut ajouter un moniteur.
- L'administrateur peut consulter un moniteur.

Tableau 5 Description textuelle du cas d'utilisation "Consulter la liste des moniteurs"

- **Description textuelle du cas d'utilisation « Ajouter un moniteur »**
- | Cas D'utilisation | Ajouter un moniteur. |
|--------------------------|--|
| Acteur | Administrateur. |
| Pré condition | L'administrateur s'est authentifié et il a affiché la liste des moniteurs. |
| Post condition | L'ajout est effectué avec succès. |
| Scénario principal | <ol style="list-style-type: none"> 1. L'administrateur clique sur le bouton "Ajouter". 2. Le système affiche l'interface de l'ajout. 3. L'administrateur remplit les informations nécessaires pour l'ajout d'un moniteur. 4. L'administrateur clique sur le bouton "Valider". 5. Le système effectue une vérification de données saisies. 6. Le système effectue l'ajout du moniteur et affiche "moniteur ajouté avec succès". |
| Scénario alternatif | Le système affiche un message d'erreur s'il existe des informations non valides ou manquantes. |

Cas D'utilisation	Ajouter un moniteur.
Acteur	Administrateur.
Pré condition	L'administrateur s'est authentifié et il a affiché la liste des moniteurs.
Post condition	L'ajout est effectué avec succès.
Scénario principal	<ol style="list-style-type: none"> 1. L'administrateur clique sur le bouton "Ajouter". 2. Le système affiche l'interface de l'ajout. 3. L'administrateur remplit les informations nécessaires pour l'ajout d'un moniteur. 4. L'administrateur clique sur le bouton "Valider". 5. Le système effectue une vérification de données saisies. 6. Le système effectue l'ajout du moniteur et affiche "moniteur ajouté avec succès".
Scénario alternatif	Le système affiche un message d'erreur s'il existe des informations non valides ou manquantes.

Tableau 6 Description textuelle du cas d'utilisation "Ajouter un moniteur"

- **Description textuelle du cas d'utilisation « Modifier un moniteur »**
- | Cas D'utilisation | Modifier un moniteur. |
|--------------------------|--|
| Acteur | Administrateur. |
| Pré condition | L'administrateur s'est authentifié et il a affiché la liste des moniteurs. |
| Post condition | Le moniteur choisi est modifié. |
| Scénario principal | <ol style="list-style-type: none"> 1. L'administrateur choisit le moniteur à modifier. 2. Le système affiche le formulaire de modification. 3. L'administrateur saisit les données et valide ses modifications. |

Cas D'utilisation	Modifier un moniteur.
Acteur	Administrateur.
Pré condition	L'administrateur s'est authentifié et il a affiché la liste des moniteurs.
Post condition	Le moniteur choisi est modifié.
Scénario principal	<ol style="list-style-type: none"> 1. L'administrateur choisit le moniteur à modifier. 2. Le système affiche le formulaire de modification. 3. L'administrateur saisit les données et valide ses modifications.

	4. Le système effectue une vérification de données saisies. 5. Le système effectue la modification et affiche un message de succès.
Scénario alternatif	Le système affiche un message d'erreur informant l'administrateur que certaines informations sont manquantes ou non valides.

Tableau 7 Description du cas d'utilisation « Modifier un moniteur »

- **Description textuelle du cas d'utilisation « Supprimer un moniteur »**

Cas D'utilisation	Supprimer un moniteur.
Acteur	Administrateur.
Pré condition	L'administrateur s'est authentifié et a affiché la liste des moniteurs.
Post condition	Le moniteur choisi est supprimé.
Scénario principal	1. L'administrateur choisit le moniteur à supprimer. 2. Le système affiche un message de validation. 3. L'administrateur valide son choix. 4. Le système effectue la suppression.
Scénario alternatif	L'administrateur annule la suppression.

Tableau 8 Description textuelle du cas d'utilisation “Supprimer un moniteur”

- **Description textuelle du cas d'utilisation « Consulter un moniteur »**

Cas D'utilisation	Consulter un moniteur.
Acteur	Administrateur.
Pré condition	L'administrateur s'est authentifié et a affiché la liste des moniteurs
Post condition	Le moniteur choisi est consulté.
Scénario principal	1. L'administrateur choisit le moniteur à consulter. 2. Le système affiche les informations du moniteur.
Scénario alternatif	Le système affiche un message d'erreur si les données ne sont pas chargées correctement.

Tableau 9 Description textuelle du cas d'utilisation “Consulter un moniteur”

2.3.2 Analyse du cas d'utilisation « Consulter la liste des véhicules »

- Raffinement du cas d'utilisation « Consulter la liste des véhicules »

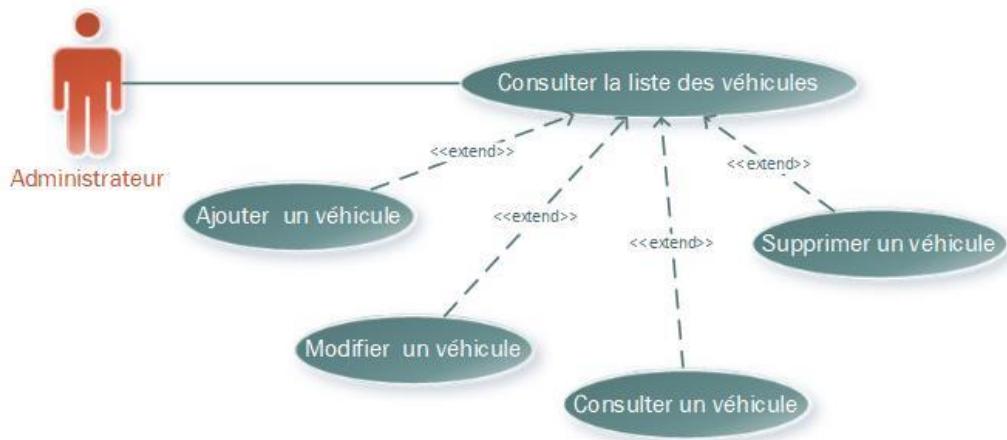


Figure 28 Diagramme de cas d'utilisation « Consulter la liste des véhicules »

- **Description textuelle du cas d'utilisation « Consulter la liste des véhicules »**

La description textuelle du cas d'utilisation consulter la liste des véhicules se fait de la même façon que consulter la liste des moniteurs. (Voir Tableau 5 page 39)

- **Description textuelle du cas d'utilisation « Ajouter un véhicule »**

La description textuelle du cas d'utilisation ajouter un véhicule se fait de la même façon qu'ajouter un moniteur. (Voir Tableau 6 page 39)

- **Description textuelle du cas d'utilisation « Modifier un véhicule »**

La description textuelle du cas d'utilisation modifier un véhicule se fait de la même façon que modifier un moniteur. (Voir Tableau 7 page 40)

- **Description textuelle du cas d'utilisation « Supprimer un véhicule »**

La description textuelle du cas d'utilisation supprimer un véhicule se fait de la même façon que supprimer un moniteur. (Voir Tableau 8 page 40)

- **Description textuelle du cas d'utilisation « Consulter un véhicule »**

La description textuelle du cas d'utilisation consulter un véhicule se fait de la même façon que consulter un moniteur. (Voir Tableau 9 page 40)

2.3.2. Analyse du cas d'utilisation « Géolocaliser les véhicules »

- Raffinement du cas d'utilisation « Géolocaliser les véhicules »

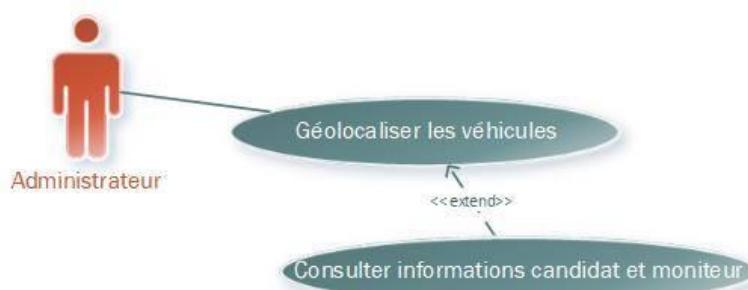


Figure 29 Diagramme de cas d'utilisation « géolocaliser les véhicules »

- **Description de cas d'utilisation « Géolocaliser les véhicules »**

Cas D'utilisation	Géolocaliser les véhicules.
Acteur	Administrateur.
Pré condition	L'administrateur s'est authentifié
Post condition	Véhicules géolocalisés.
Scénario principal	<ol style="list-style-type: none"> 1. L'administrateur clique sur le bouton “Géolocaliser”. 2. Le système affiche la carte de géolocalisation.
Scénario alternatif	Le système affiche un message d'erreur si les données ne sont pas chargées.
Extension	- L'administrateur peut consulter informations du candidat et du moniteur.

Tableau 10 Description textuelle du cas d'utilisation “Géolocaliser les véhicules”

- **Description de cas d'utilisation « Consulter informations du candidat et du moniteur »**

Cas D'utilisation	Consulter informations d'un candidat et du moniteur.
Acteur	Administrateur.
Pré condition	L'administrateur s'est authentifié, il se trouve dans l'interface de géolocalisation.
Post condition	Les informations consultées.
Scénario principal	<ol style="list-style-type: none"> 1. L'administrateur clique sur l'icône “Voiture”. 2. Le système affiche toutes les informations relatives au candidat et au moniteur.

Tableau 11 Description textuelle du cas d'utilisation “Consulter informations du candidat et du moniteur”

2.3.3. Analyse du cas d'utilisation « Consulter la liste des candidats avec le nombre de rapport »

- Raffinement du cas d'utilisation « Consulter la liste des candidats avec le nombre de rapport »

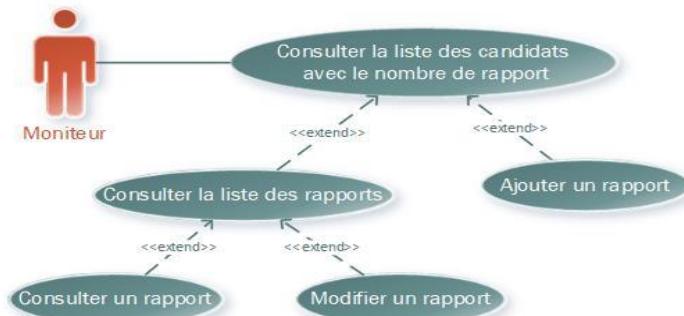


Figure 30 Diagramme de cas d'utilisation « Consulter la liste des candidats avec le nombre de rapport »

- Description textuelle du cas d'utilisation « Consulter la liste des candidats avec le nombre de rapport »

Cas D'utilisation	Consulter la liste des candidats avec le nombre de rapport
-------------------	--

Acteur	Moniteur.
Pré condition	Le moniteur s'est authentifié
Post condition	La liste des candidats avec le nombre de rapport affichée
Scénario principal	<ol style="list-style-type: none"> 1. Le moniteur clique sur le bouton "Evaluation". 2. Le système affiche une liste de candidats avec le nombre de rapport de chacun.
Scénario alternatif	Le système affiche une liste vide si les données ne sont pas chargées.
Extension	<ul style="list-style-type: none"> - Le moniteur peut consulter la liste des rapports d'un candidat. - Le moniteur peut ajouter un rapport.

Tableau 12 Description textuelle du cas d'utilisation « Consulter liste de candidats avec le nombre de rapport »

- Description textuelle du cas d'utilisation « Ajouter un rapport »

Cas D'utilisation	Ajouter rapport.
-------------------	------------------

Acteur	Moniteur.
--------	-----------

Pré condition	Le moniteur s'est authentifié et il a consulté le nombre de rapport des candidats.
Post condition	Rapport ajouté.
Scénario principal	<ol style="list-style-type: none"> 1. Le moniteur clique sur le bouton "Ajouter un Rapport". 2. Le système affiche le nom du candidat avec les détails de la séance et demande une confirmation d'ajout. 3. Le moniteur clique sur le bouton « Ajouter ». 4. Le système affiche un formulaire à remplir. 5. Le moniteur remplit le formulaire. 6. Le moniteur clique sur le bouton "Valider". 7. Le système affiche « rapport ajouté avec succès »
Scénario alternatif	<ul style="list-style-type: none"> - Le système affiche un message d'erreur si les données ne sont pas chargées de la base de données. - Le moniteur annule l'ajout. - Le moniteur ne valide pas le formulaire.

Tableau 13 Description textuelle du cas d'utilisation "Ajouter un rapport"

- **Description textuelle du cas d'utilisation « Consulter la liste des rapports »**

Cas D'utilisation	Consulter la liste des rapports.
Acteur	Moniteur.
Pré condition	Le moniteur s'est authentifié et a consulté la liste des candidats avec le nombre de rapports.
Post condition	Liste des rapports d'un candidat consultée.
Scénario principal	<ol style="list-style-type: none"> 1. Le moniteur clique sur le bouton "consulter". 2. Le système affiche une liste des rapports du candidat sélectionné.
Scénario alternatif	Le système affiche une page vide si les données ne sont pas chargées.
Extension	<ul style="list-style-type: none"> - Le moniteur consulter un rapport. - Le moniteur modifier après la consultation d'un rapport (avec une contrainte du temps).

Tableau 14 Description textuelle du cas d'utilisation "Consulter la liste des rapports "

- **Description textuelle du cas d'utilisation « Consulter un rapport »**
La description textuelle du cas d'utilisation consulter un rapport se fait de la même façon que consulter un moniteur. (Voir tableau 9 page 40)
- **Description textuelle du cas d'utilisation « Modifier un rapport »**
La description textuelle du cas d'utilisation modifier un rapport se fait de la même façon que modifier un moniteur. (Voir tableau 7 page 40)

2.3.4. Analyse du cas d'utilisation « Afficher le planning »

Dans ce cas d'utilisation on doit prendre en compte que le planning est constitué d'un ensemble de séances.

- **Raffinement du cas d'utilisation « Afficher le planning »**

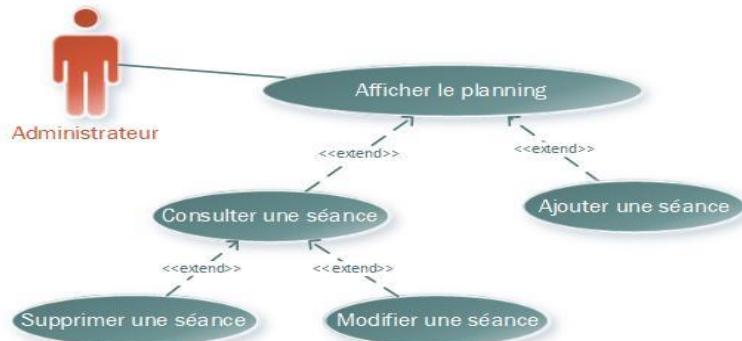


Figure 31 Diagramme de cas d'utilisation « Afficher le planning »

- **Description textuelle du cas d'utilisation « Afficher le planning »**

Cas D'utilisation	Afficher le planning.
Acteur	Administrateur.
Pré condition	L'administrateur s'est authentifié
Post condition	Ensembles des informations du planning affichées.
Scénario principal	<ol style="list-style-type: none"> 1. L'Administrateur clique sur le lien « Planning ». 2. Le système affiche le planning des séances.
Extension	<ul style="list-style-type: none"> -L'administrateur peut consulter une séance. - L'administrateur peut ajouter une séance.

Tableau 15 Description textuelle du cas d'utilisation “Afficher le planning”

- **Description textuelle du cas d'utilisation « Ajouter une séance »**

Cas D'utilisation	Ajouter une séance.
Acteur	Administrateur.
Pré condition	L'administrateur s'est authentifié et il a affiché son planning.

Post condition	Séance ajoutée.
Scénario principal	<ol style="list-style-type: none"> 1. L'administrateur sélectionne un horaire. 2. Le système affiche le formulaire d'ajout d'une séance. 3. L'administrateur remplit le formulaire. 4. L'administrateur clique sur le bouton « valider ». 5. Le système vérifie les informations saisies. 6. Le système affiche un message indiquant que la séance est ajoutée.
Exception	<ul style="list-style-type: none"> - Un message d'erreur est affiché si l'un des champs obligatoires est vide et/ou invalide. - Le moniteur annule l'ajout de la séance.

Tableau 16 Description textuelle du cas d'utilisation “Ajouter une séance”

- **Description textuelle du cas d'utilisation « Consulter une séance »**

Cas D'utilisation	Afficher le planning.
Acteur	Administrateur.
Pré condition	L'administrateur s'est authentifié et a affiché son planning.
Post condition	Ensembles des informations d'une séance sont affichées.
Scénario principal	<ol style="list-style-type: none"> 1. L'Administrateur clique sur une séance. 2. Le système affiche les informations de la séance sélectionnée.
Extension	<ul style="list-style-type: none"> - L'administrateur peut modifier la séance. - L'administrateur peut supprimer la séance.

Tableau 17 Description textuelle du cas d'utilisation “Consulter une séance”

- **Description textuelle du cas d'utilisation « Supprimer une séance »**

Cas D'utilisation	Supprimer une séance.
Acteur	Administrateur.
Pré condition	L'administrateur s'est authentifié et il est entrain de consulter une séance.
Post condition	Séance supprimée.
Scénario principal	<ol style="list-style-type: none"> 1. L'administrateur clique sur le bouton supprimer.

2. Le système demande une validation de la suppression.
 3. L'administrateur valide la suppression.
 4. Le système affiche un message indiquant que la séance est supprimée.

Tableau 18 Description textuelle du cas d'utilisation “Supprimer une séance”

- Description textuelle du cas d'utilisation « Modifier une séance. »

Acteur	Administrateur.
Pré condition	L'administrateur s'est authentifié et il est entrain de consulter une séance.
Post condition	Séance modifiée.
Scénario principal	<ol style="list-style-type: none">1. L'administrateur clique sur le bouton modifier.2. Le système affiche le formulaire de mordication.3. L'administrateur saisit les données et valide les modifications.4. Le système effectue une vérification de données saisies.5. Le système effectue la modification et affiche un message de succès.
Exception	<ul style="list-style-type: none">- Un message d'erreur est affiché si l'un des champs obligatoires est vide et/ou invalide.- L'administrateur annule la modification

Tableau 19 Description textuelle du cas d'utilisation « Modifier une séance »

III. Conception

3.1 Conception de cas ‘Consulter la liste des moniteurs’

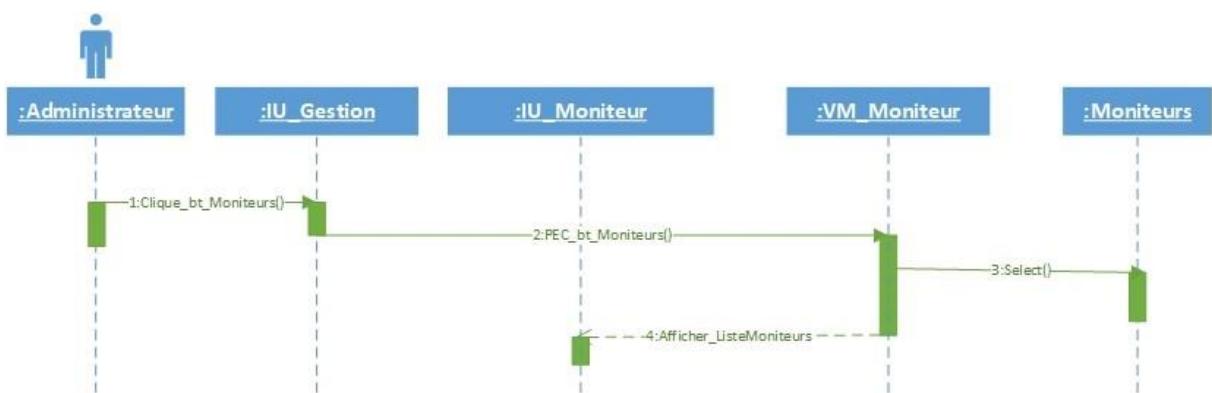


Figure 32 Diagramme de séquence de l'opération « Consulter la liste des moniteurs »

3.2 Conception de cas ‘Ajouter un moniteur’

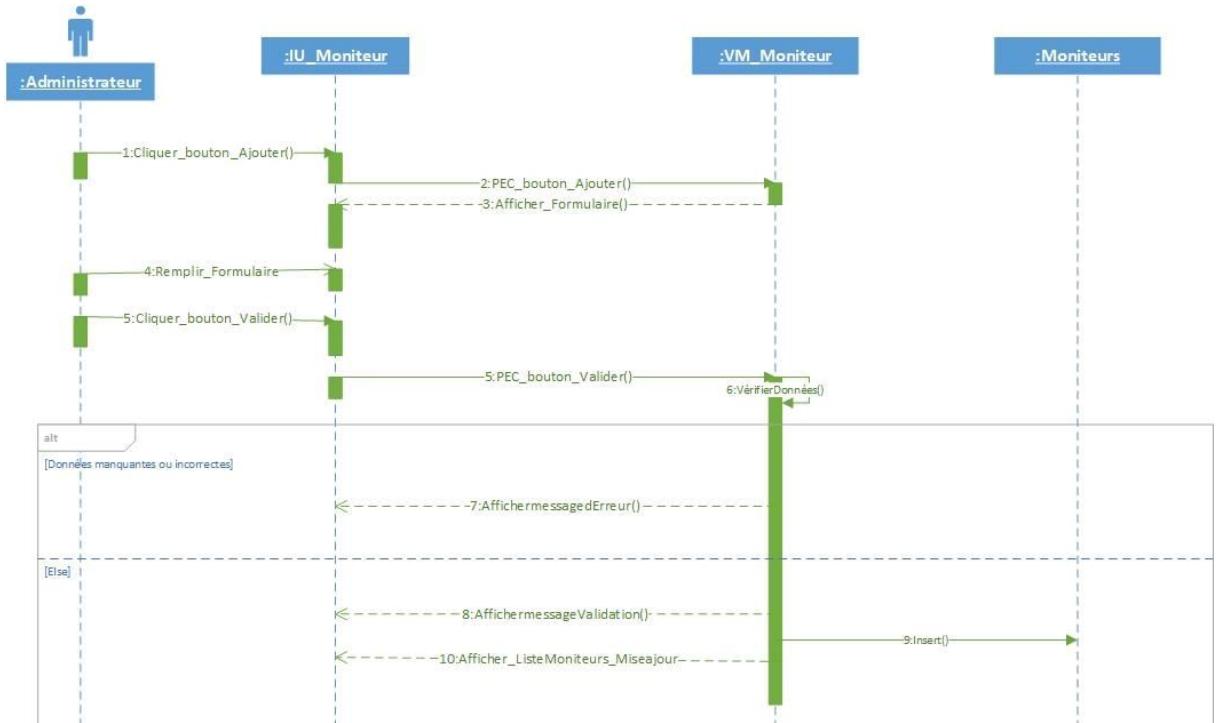


Figure 33 Diagramme de séquence de l’opération « Ajouter un moniteur »

3.3 Conception de cas ‘Modifier un moniteur’

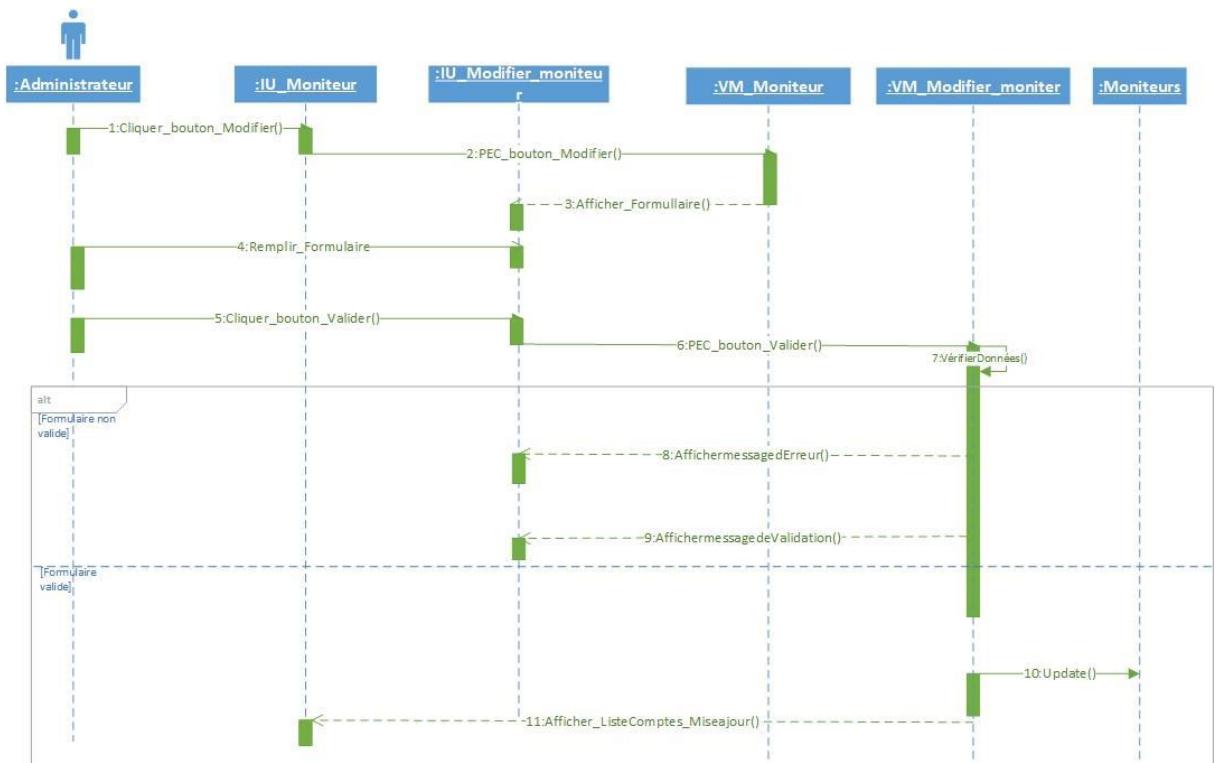


Figure 34 Diagramme de séquence de l’opération « Modifier un moniteur »

3.4 Conception de cas ‘Supprimer un moniteur’



Figure 35 Diagramme de séquence de l’opération « Supprimer un moniteur »

3.5 Conception de cas ‘Consulter un moniteur’

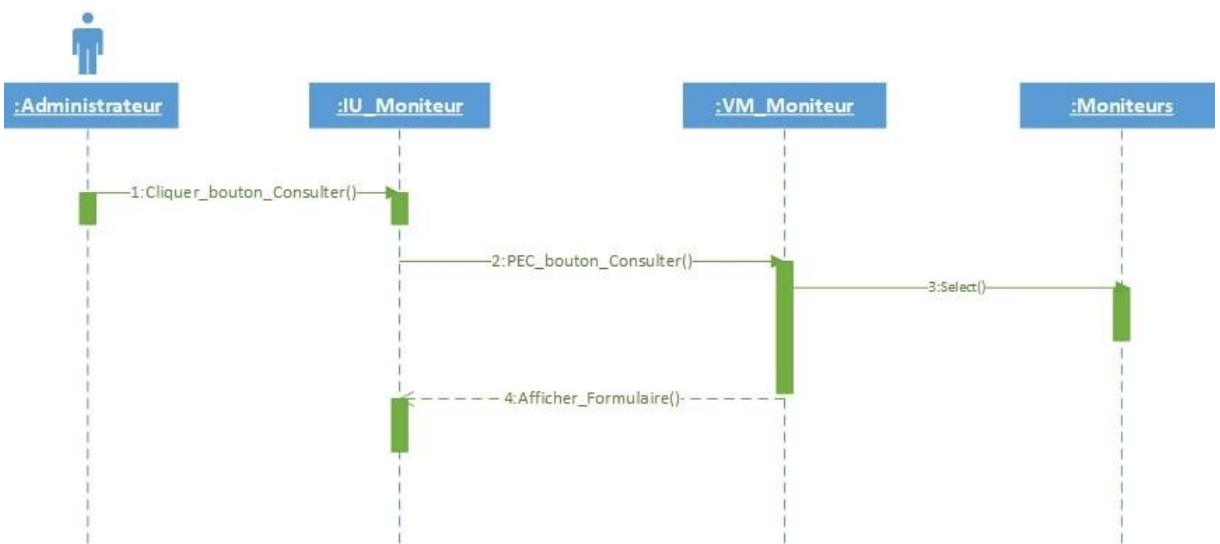


Figure 36 Diagramme de séquence de l’opération « Consulter un moniteur »

3.6 Conception de cas ‘Géolocaliser les véhicules’

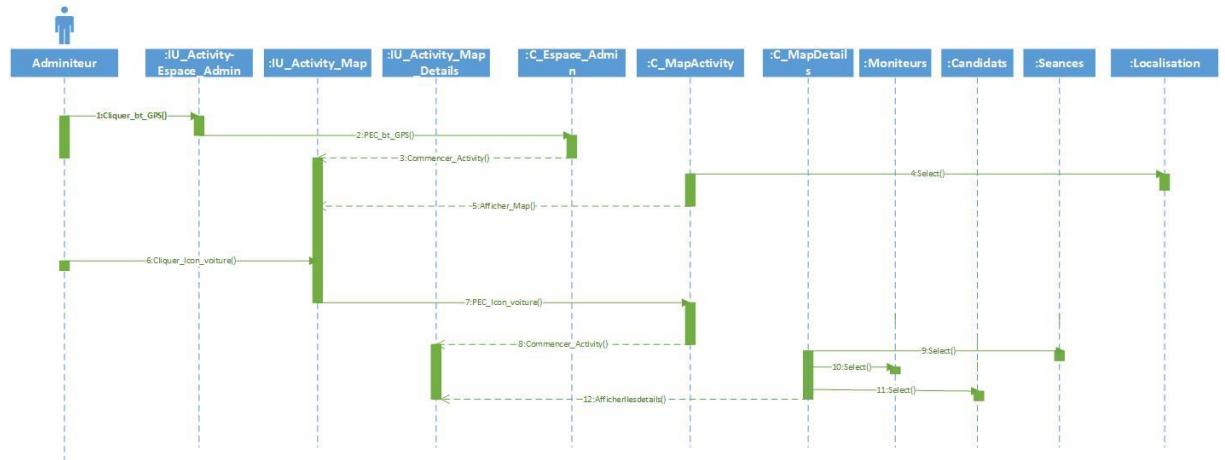


Figure 37 Diagramme de séquence de l’opération « Géolocaliser les véhicules »

3.7 Conception des cas ‘Consulter informations candidat et moniteur’



Figure 38 Diagramme de séquence du cas « Consulter informations candidat et moniteur »

3.8 Conception de cas ‘Consulter la liste des candidats avec le nombre de rapports’

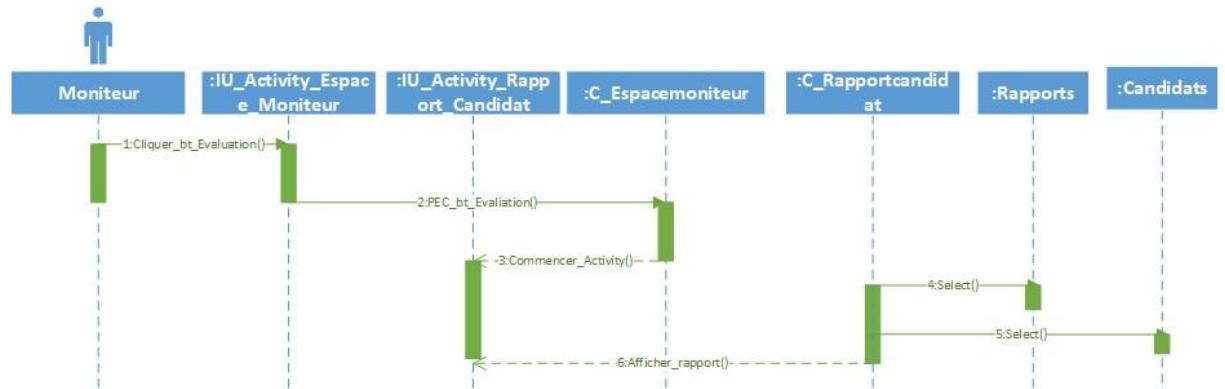


Figure 39 Diagramme de séquence de l’opération « Consulter la liste des candidats avec le nombre de rapports »

3.9 Conception de cas ‘Ajouter un rapport’

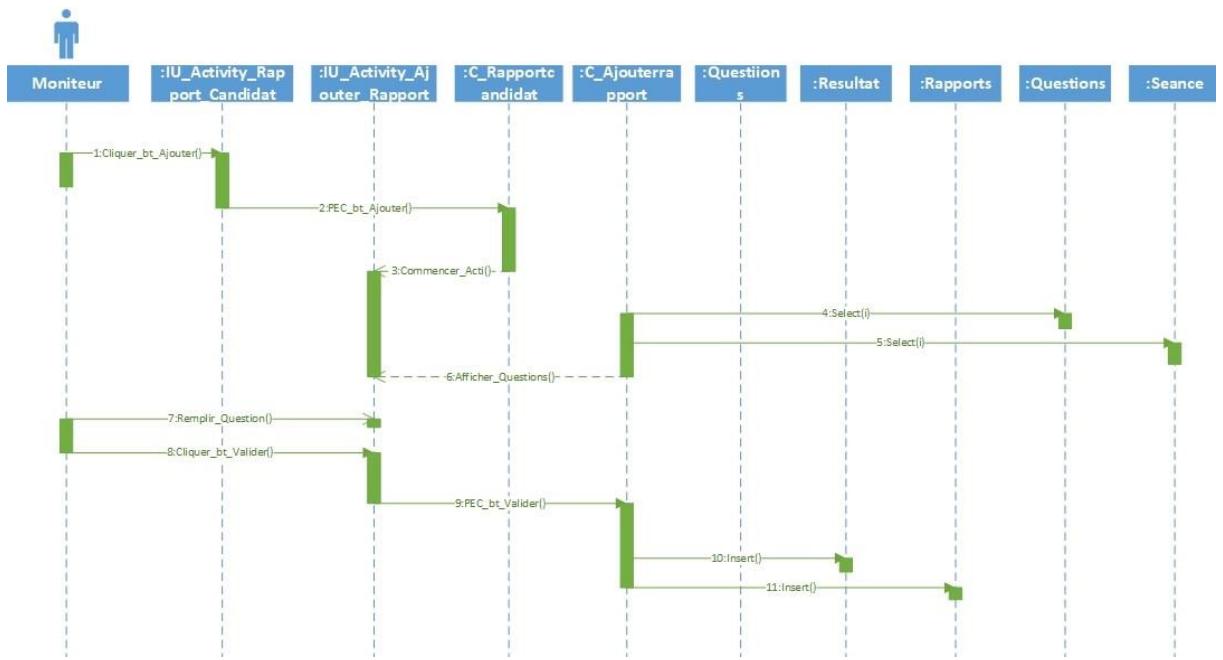


Figure 40 Diagramme de séquence de l’opération « Ajouter un rapport »

3.10 Conception de cas ‘Consulter le rapport’

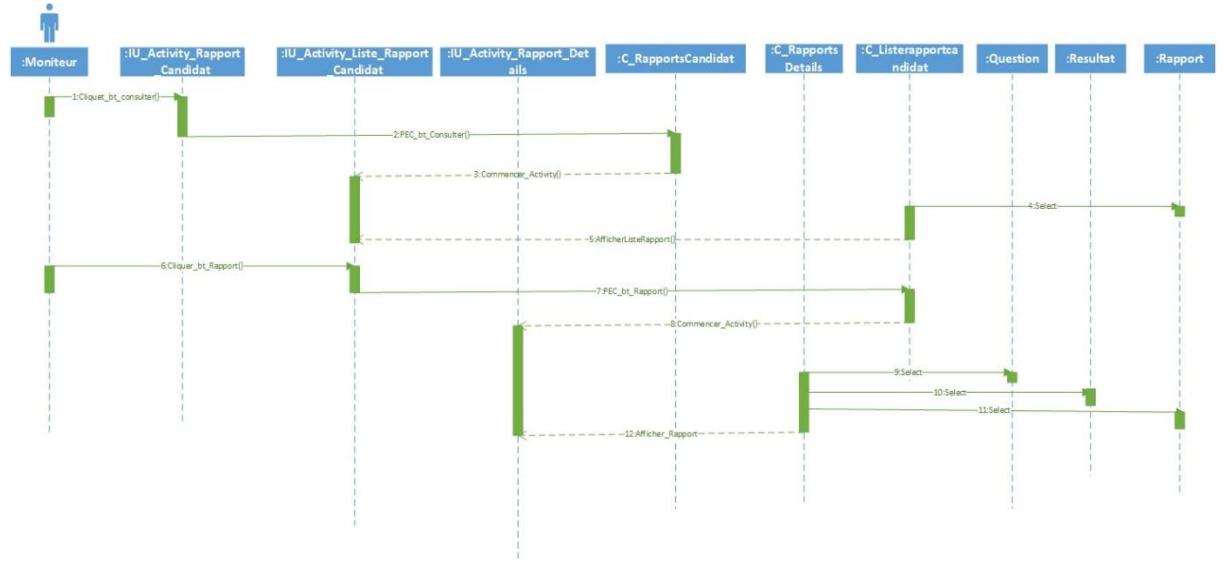


Figure 41 Diagramme de séquence de l’opération « Consulter le rapport »

3.11 Conception de cas ‘Modifier le rapport’

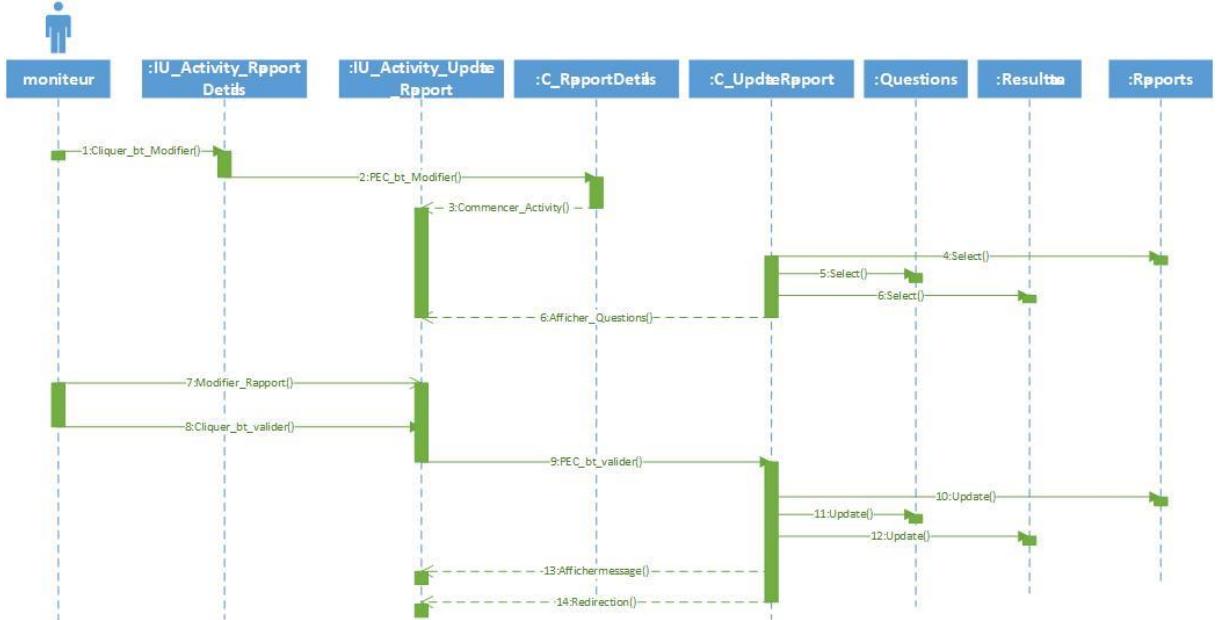


Figure 42 Diagramme de séquence de l’opération « Modifier le rapport »

3.12 Conception de cas ‘Afficher le planning’

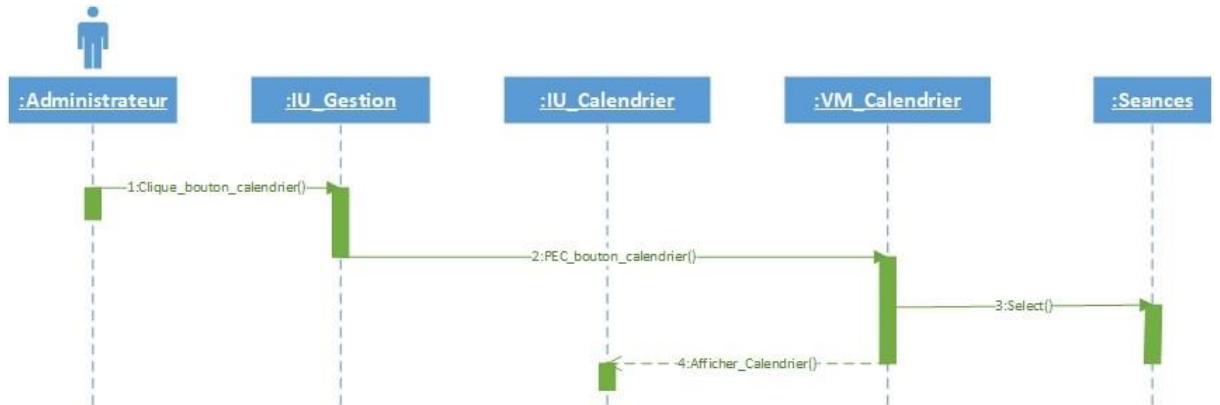


Figure 43 Diagramme de séquence de l’opération « Afficher le planning »

3.13 Conception de cas ‘Ajouter une séance’

Pour la conception de cas d’utilisation ajouter une séance se fait de la même façon qu’ajouter un moniteur (figure 33, page 48).

3.14 Conception de cas ‘Consulter une séance’

Pour la conception de cas d'utilisation consulter une séance se fait de la même façon que consulter un moniteur (figure 36 page 49).

3.15 Conception de cas ‘Supprimer une séance’

Pour la conception de cas d'utilisation supprimer une séance se fait de la même façon que supprimer un moniteur (page 35, page 49).

3.16 Conception de cas ‘Modifier une séance’

Pour la conception de cas d'utilisation modifier une séance se fait de la même façon que modifier un moniteur (figure 34, page 48).

3.17 Conception de cas ‘consulter liste des véhicules’

Pour la conception de cas d'utilisation consulter liste des véhicules se fait de la même façon que consulter liste des moniteurs (figure 32, page 47).

3.18 Conception de cas ‘Ajouter un véhicule’

Pour la conception de cas d'utilisation ajouter un véhicule se fait de la même façon qu'ajouter un moniteur (figure 33, page 48).

3.19 Conception de cas ‘Modifier un véhicule’

Pour la conception de cas d'utilisation modifier un véhicule se fait de la même façon que modifier un moniteur (figure 34, page 48).

3.20 Conception de cas ‘Supprimer un véhicule’

Pour la conception de cas d'utilisation supprimer un véhicule se fait de la même façon que supprimer un moniteur (page 35, page 49).

3.21 Conception de cas ‘Consulter un véhicule’

Pour la conception de cas d'utilisation consulter un véhicule se fait de la même façon que consulter un moniteur (figure 36 page 49).

IV. Diagramme de classe globale du premier sprint

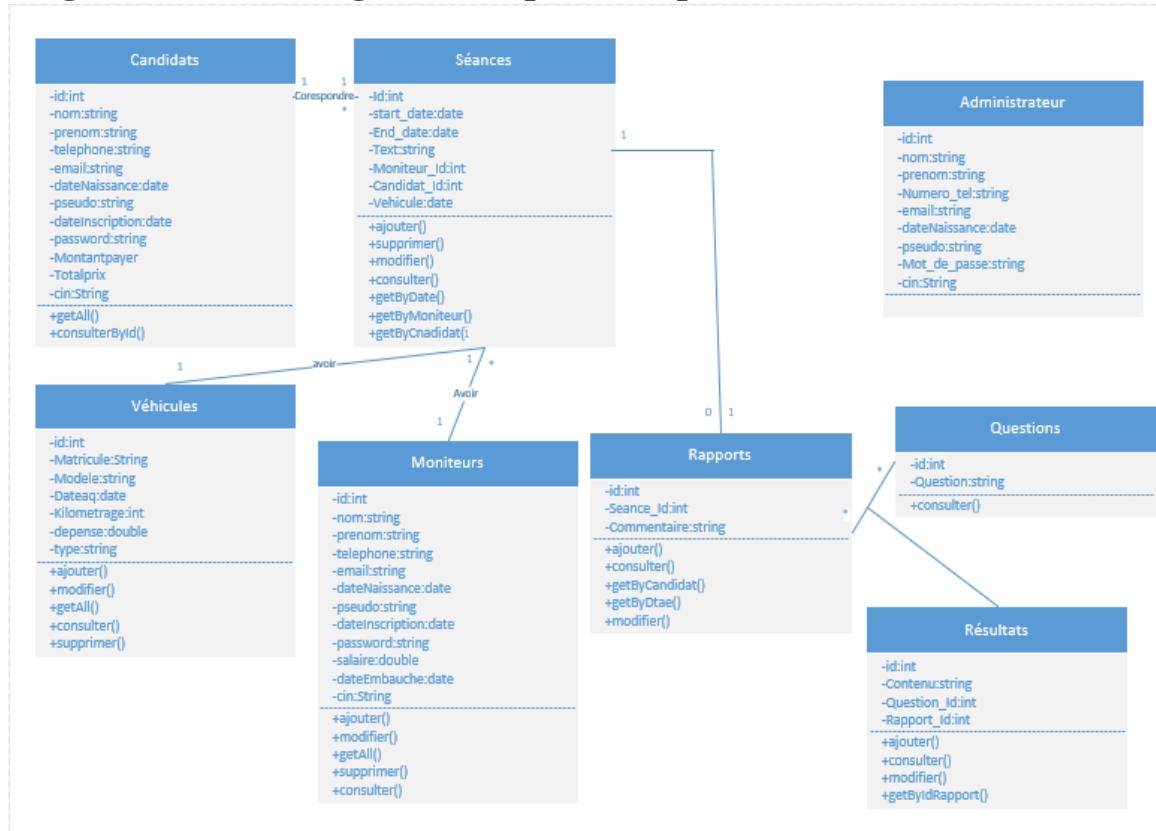


Figure 44 Diagramme de classe globale du sprint 1

V. Structure de la base de données

Dans cette partie, nous présentons les tables de la base de données utilisées lors de ce premier sprint, tout en respectant le règles de passage du modèle conceptuel et modèle objet vers le modèle relationnel.

- **Table « Administrateurs »**

Champs	Types	Contraintes
Id	Bigint	Primary key
Numero_tel	Character varying	Not null
Cin	Character varying	Not null
Nom	Character varying	Not null
Prenom	Character varying	Not null
Email	Character varying	Not null

Date_naissance	Date	Not null
Pseudo	Character varying	Not null
Mot_de_passe	Character varying	Not null
Photo	Character varying	---

Tableau 20 Table administrateurs

- **Table « Moniteurs »**

Champs	Types	Contraintes
Id	Bigint	Primary key
Nom	Character varying	Not null
Telephone	Character varying	Not null
Prenom	Character varying	Not null
Email	Character varying	Not null
DateNaissance	Date	Not null
Pseudo	Character varying	Not null
Password	Character varying	Not null
Salaire	Character varying	Not null
DateEmbauche	Date	Not null
Cin	Character varying	Not null
Image	Character varying	---

Tableau 21 Table moniteurs

- **Table « Véhicules »**

Champs	Types	Contraintes
Id	Bigint	Primary key
Matricule	Character varying	Not null
DateAcq	Date	Not null
Modele	Character varying	Not null

Kilometrage	Integer	Not null
Depense	Double	Not null
Type	Character varying	Not null

Tableau 22 Table véhicules

- **Table « Séances »**

Champs	Types	Contraintes
Id	Bigint	Primary key
Start_date	Character varying	Not null
End_date	Character varying	Not null
Text	Character varying	Not null
Moniteur_Id	Bigint	Not null
Véhicule_Id	Bigint	Not null
Candidat_Id	Bigint	Not null

Tableau 23 Table Séances

- **Table « Rapports »**

Champs	Types	Contraintes
Id	Bigint	Primary key
Commentaire	String	---
Seance_Id	Bigint	Not null

Tableau 24 Table rapports

- **Table « Questions »**

Champs	Types	Contraintes
Id	Bigint	Primary key
Question	String	Not null

Tableau 25 Table rapports

- **Table « Resultats »**

Champs	Types	Contraintes
Id	Bigint	Primary key
Contenu	String	---
Question_Id	Bigint	Not null
Rapport_Id	Bigint	Not null

Tableau 26 Table résultats

IV. Implémentation

4.1 Diagramme de composants

Le diagramme de composant permet de représenter l'architecture physique et statique d'une application : Les fichiers de code source, les fichiers exécutables, les bases de données physiques, etc.

- **Diagramme de composant de la fonctionnalité « Gestion Moniteur »**

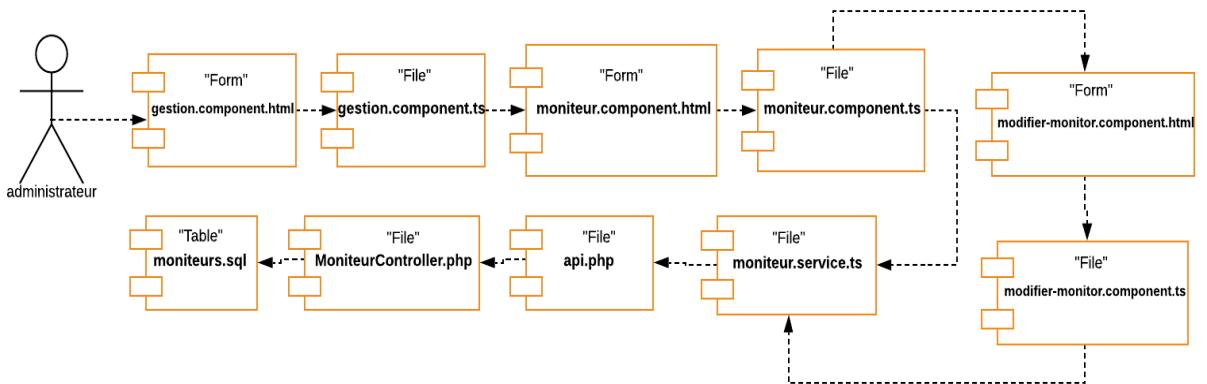


Figure 45 Diagramme de composants de la fonctionnalité : « Gestion moniteur »

➤ Interface de la fonctionnalité « Gestion moniteur »

The screenshot shows a web application interface for managing monitors. On the left, a sidebar menu includes 'ACCUEIL', 'GESTION' (with 'Moniteur' highlighted in pink), 'COMPTABILITÉ', and 'Candidat'. The main content area is titled 'Liste des moniteurs' and displays a table with one row:

Nom	Prénom	Email	Modifier	Supprimer
abbouz	ines	inesabbouz@gmail.com	<button>Modifier</button>	<button>Supprimer</button>

A pink button 'Ajouter moniteur' is located in the top right corner of the main content area.

Figure 46 Interface du cas "Gestion moniteurs"

- **Diagramme de composant de la fonctionnalité « Consulter le planning » (mobile)**

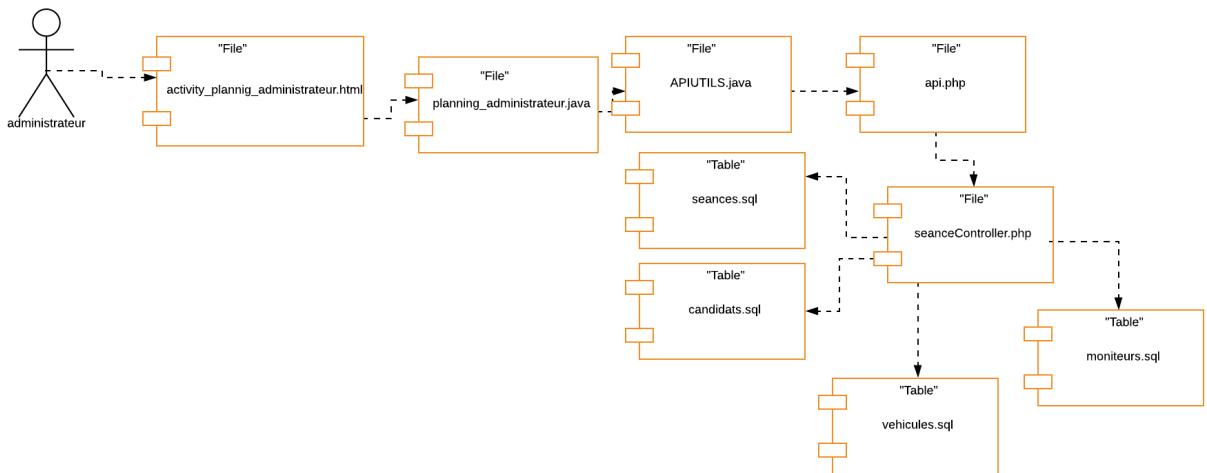


Figure 47 Diagramme de composants de la fonctionnalité : « Consulter le planning »

➤ Interface du cas « Consulter le planning » (mobile)

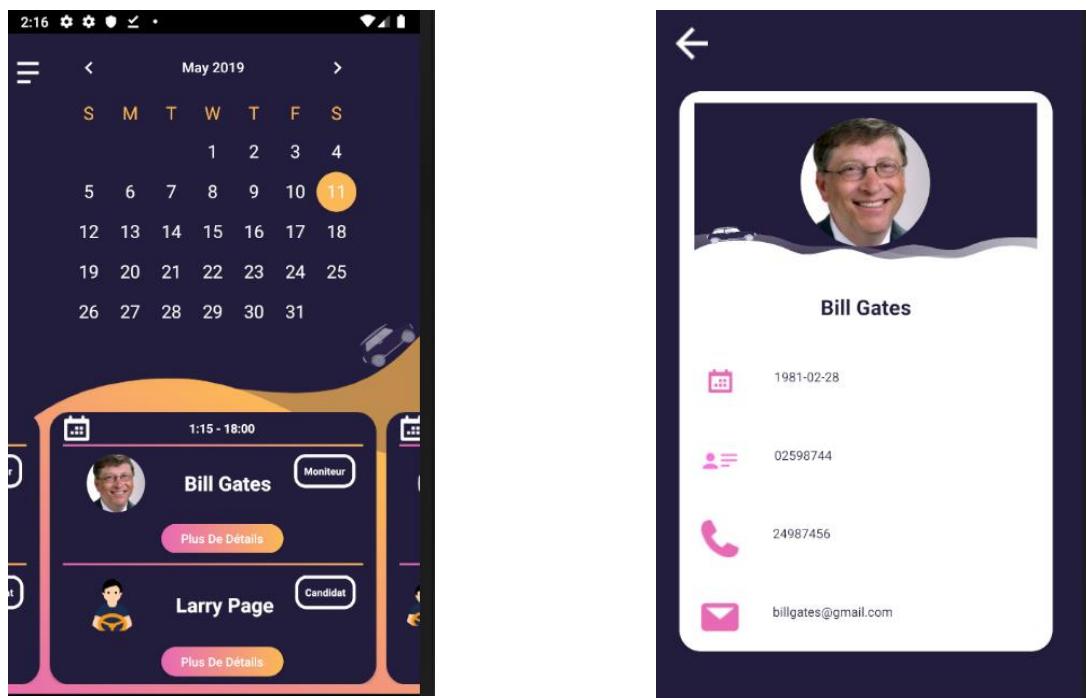


Figure 48 Interfaces du cas consulter le planning (mobile)

- **Diagramme de composant de la fonctionnalité « Gestion planning » (web)**

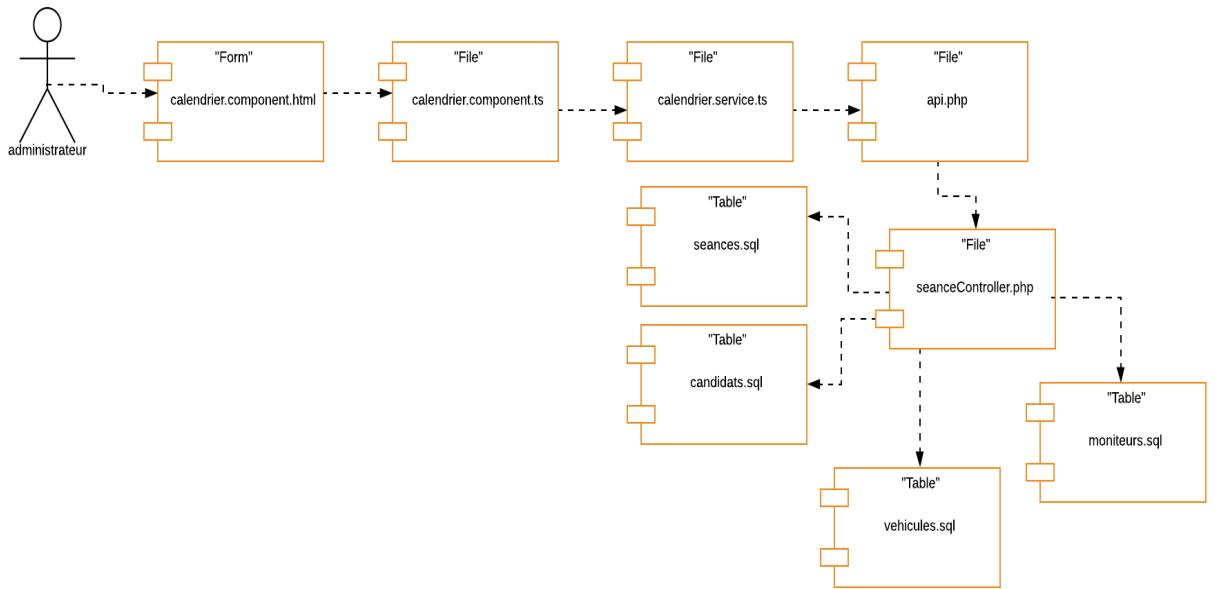


Figure 49 Diagramme de composants de la fonctionnalité : « Gestion planning » (web)

➤ Interfaces du cas « Gestion planning » (web)

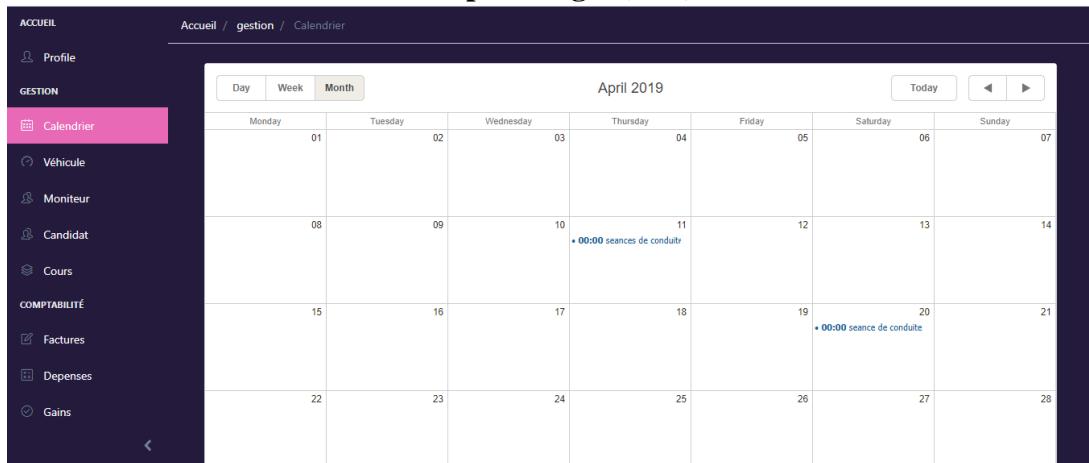


Figure 50 Interface du cas "Gestion planning " (web)

- **Diagramme de composant de la fonctionnalité « Géolocaliser les véhicules »**

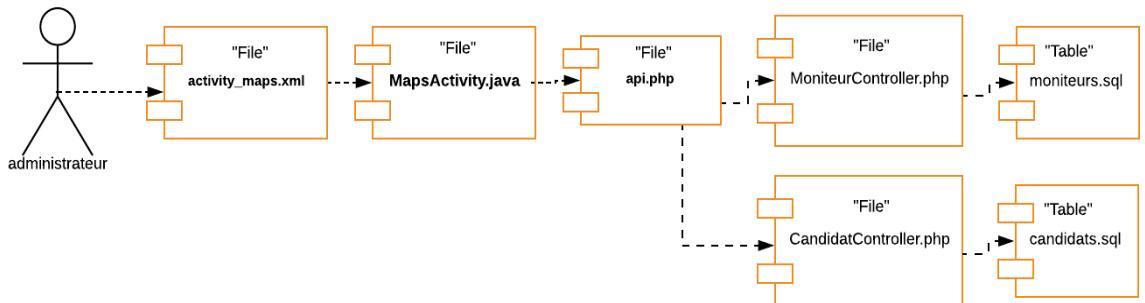


Figure 52 Diagramme de composants de la fonctionnalité : « Géolocaliser les véhicules » /1

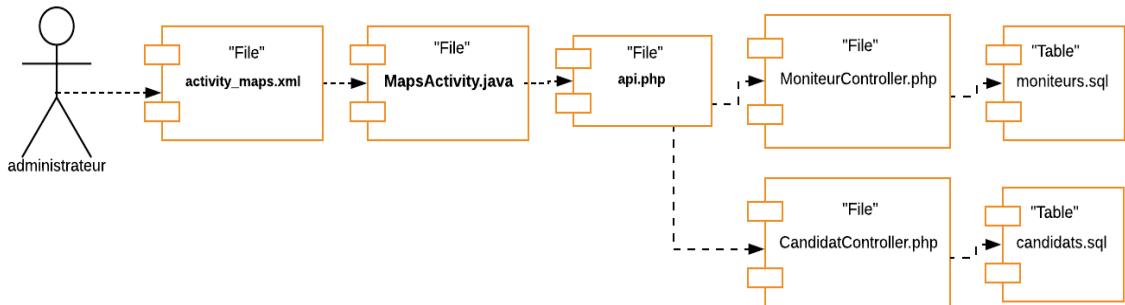


Figure 51 Diagramme de composants de la fonctionnalité : « Géolocaliser les véhicules » /2

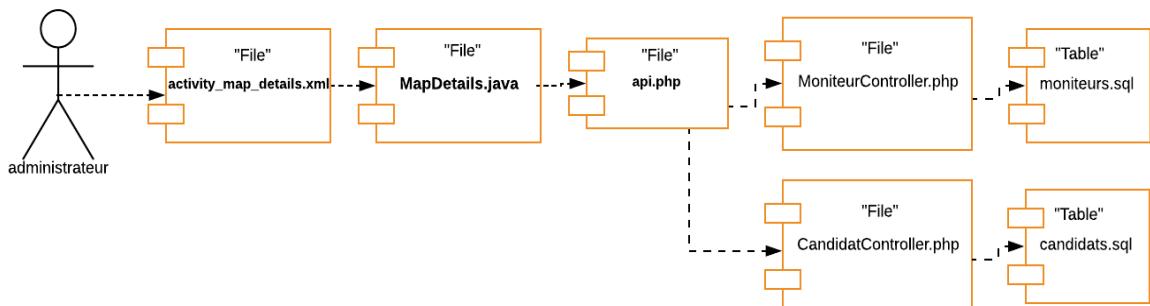


Figure 53 Diagramme de composants de la fonctionnalité : « Géolocaliser les véhicules » /3

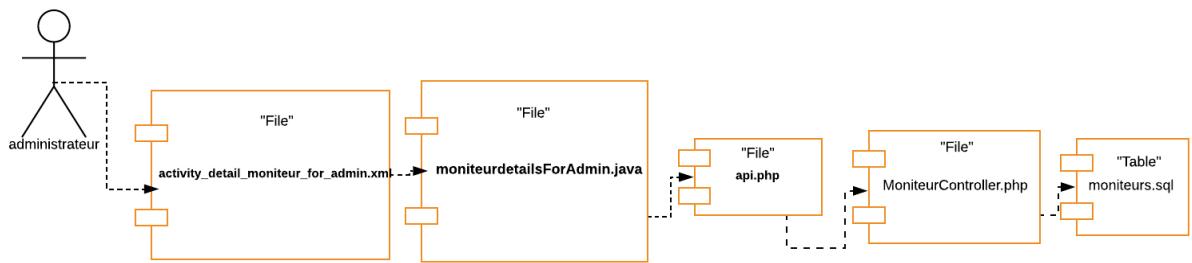


Figure 54 Diagramme de composants de la fonctionnalité : « Géolocaliser les véhicules » partie 4

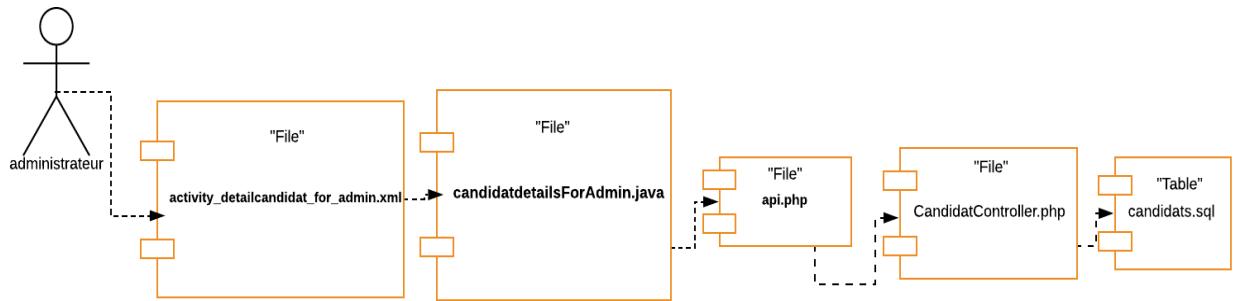


Figure 55 Diagramme de composants de la fonctionnalité : « Géolocaliser les véhicules » partie 5

➤ Interfaces du cas « Géolocaliser les véhicules »

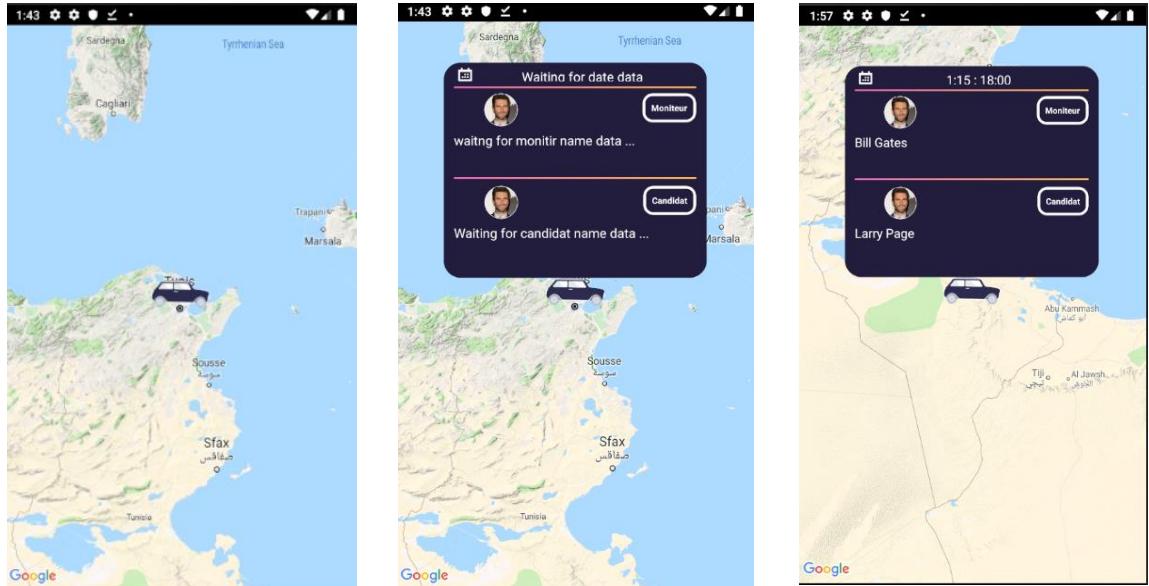


Figure 56 Interfaces du cas "Géolocaliser les véhicules"

- Diagramme de composant de la fonctionnalité « Gestion rapport »

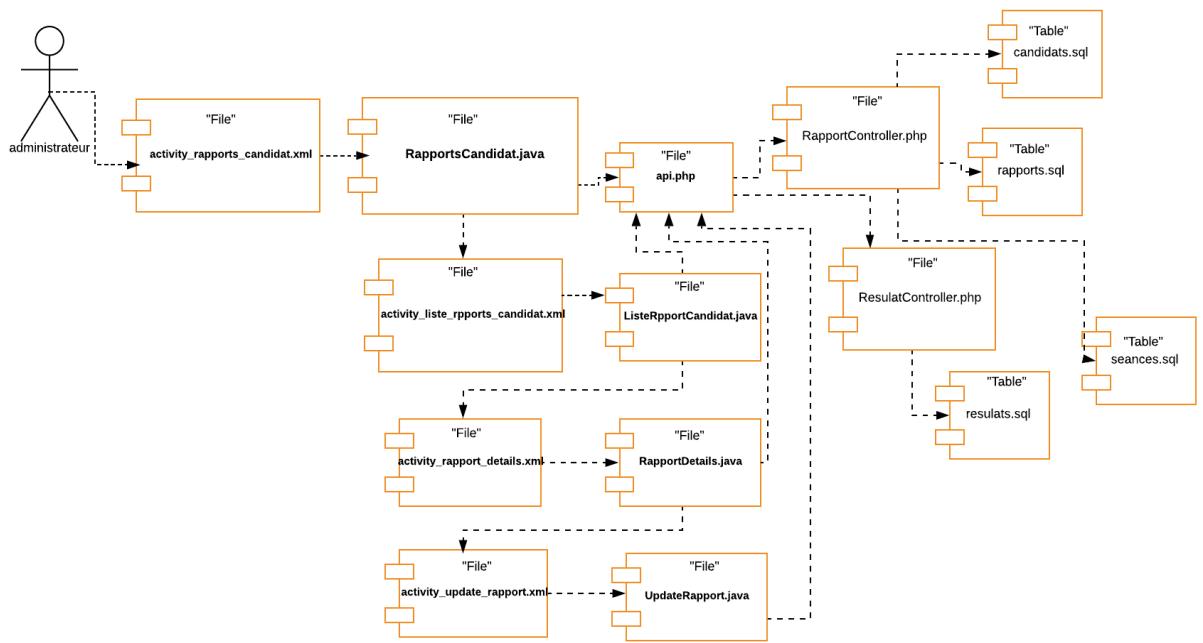


Figure 57 Diagramme de composants de la fonctionnalité : « Gestion rapport » partie 1

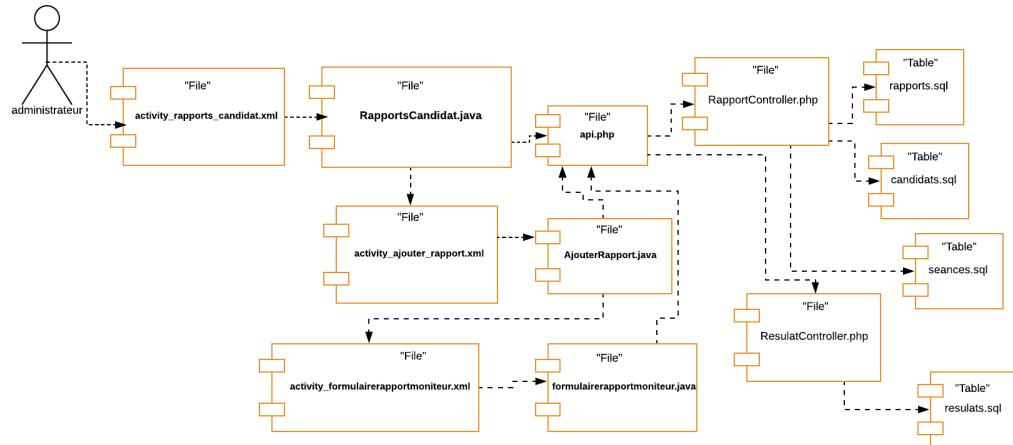


Figure 58 Diagramme de composants de la fonctionnalité : « Gestion rapport » partie

➤ Interfaces de la fonctionnalité « Gestion rapport »

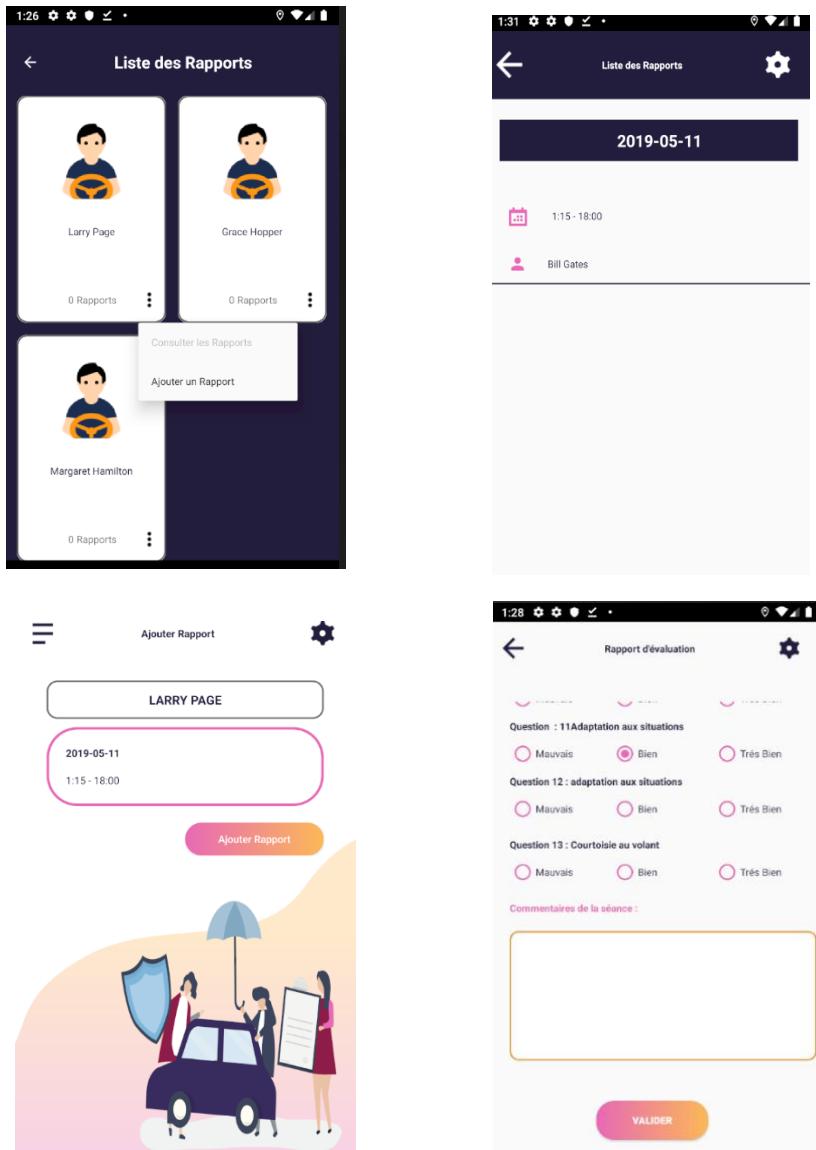


Figure 59 Interfaces de la fonctionnalité « Gestion rapport »

V. Test

Afin de produire une application de qualité et dans le but d'assurer le bon fonctionnement des méthodes implémentées précédemment, il nécessaire de passer par une phase de test. Cette étape consiste principalement à vérifier que le résultat obtenu est conforme au résultat attendu pour chaque route du serveur.

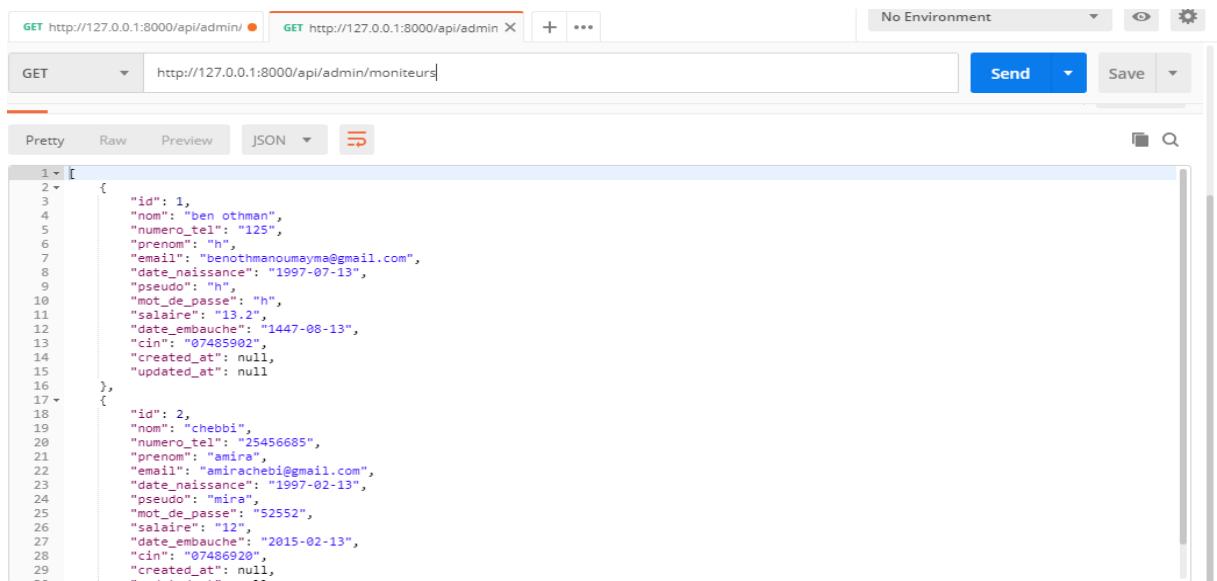
5.1 Les tests des web services

Pour tester les web services, nous avons utilisé l'outil Postman¹³.

Voici quelques captures de nos tests.

¹³ Postman est une application pour tester les web services et les API.

➤ Test du web service consulter les moniteurs



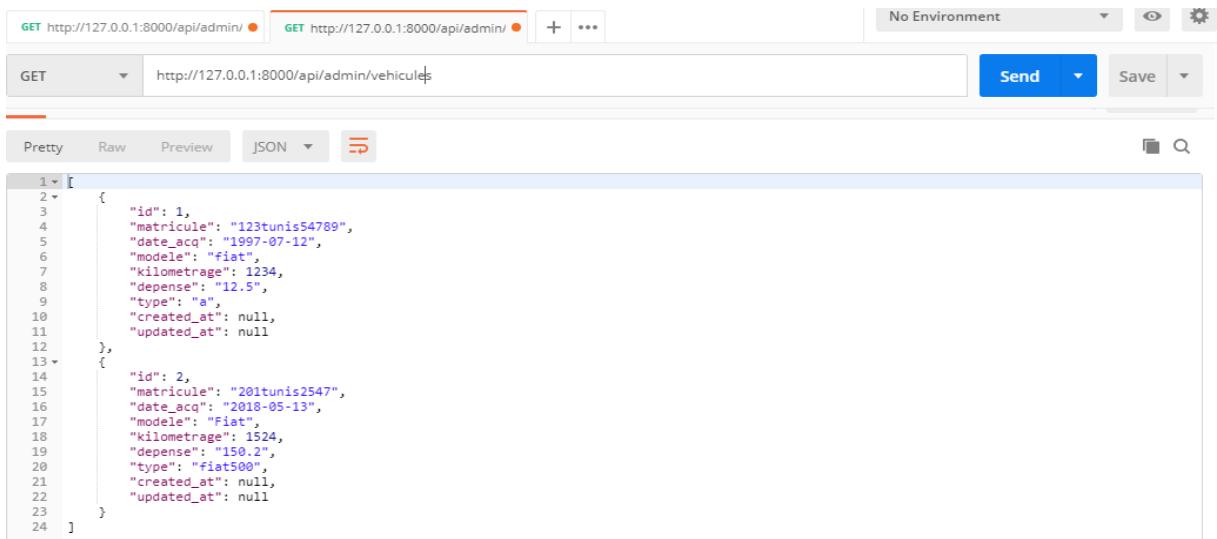
```

GET http://127.0.0.1:8000/api/admin/moniteurs
[{"id": 1, "nom": "ben_othman", "numero_tel": "125", "prenom": "", "email": "benothmanoumayma@gmail.com", "date_naissance": "1997-07-13", "pseudo": "", "mot_de_passe": "h", "salaire": "13.2", "date_embauche": "1447-08-13", "cin": "07485902", "created_at": null, "updated_at": null}, {"id": 2, "nom": "chebbi", "numero_tel": "25456685", "prenom": "amira", "email": "amirachebbi@gmail.com", "date_naissance": "1997-02-13", "pseudo": "mira", "mot_de_passe": "52552", "salaire": "12", "date_embauche": "2015-02-13", "cin": "07486930", "created_at": null, "updated_at": null}]

```

Figure 60 Test du web service "Consulter les moniteurs"

➤ Test du web service consulter les véhicules



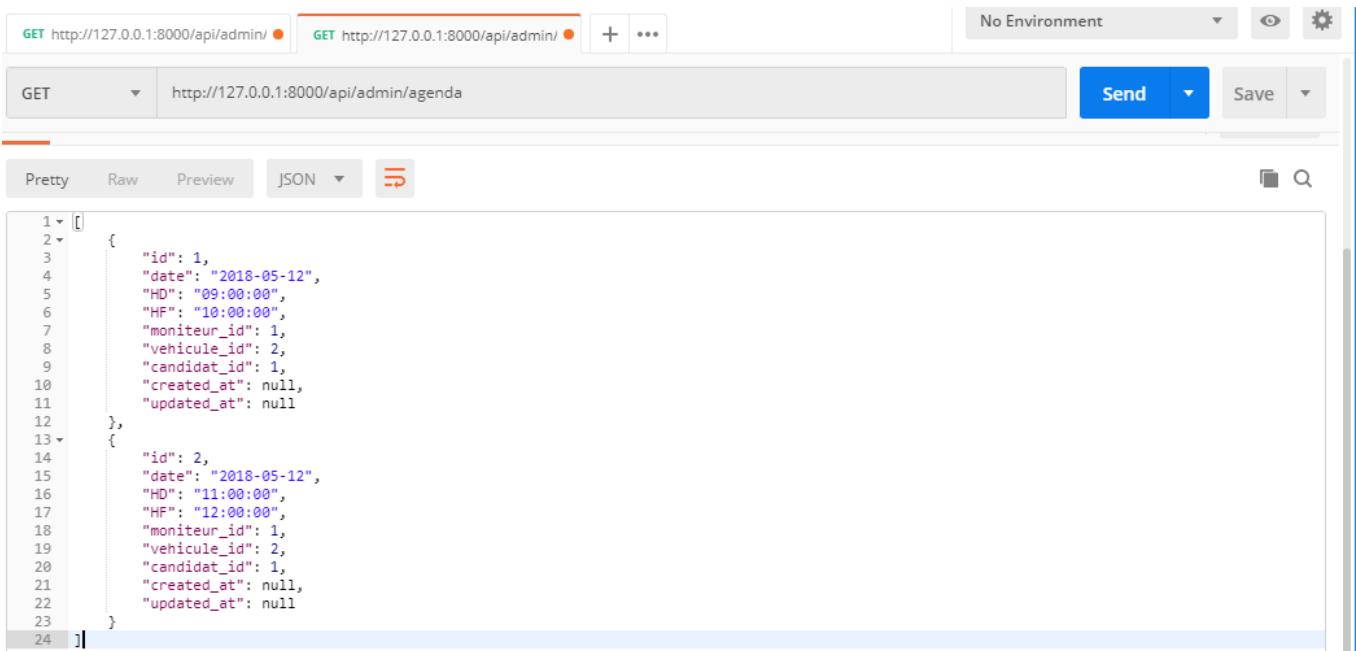
```

GET http://127.0.0.1:8000/api/admin/vehicule
[{"id": 1, "matricule": "123tunis54789", "date_acq": "1997-07-12", "modele": "fiat", "kilometrage": 1234, "depense": "12.5", "type": "a", "created_at": null, "updated_at": null}, {"id": 2, "matricule": "201tunis2547", "date_acq": "2018-05-13", "modele": "Fiat", "kilometrage": 1524, "depense": "150.2", "type": "fiat500", "created_at": null, "updated_at": null}]

```

Figure 61 Test du web service "Consulter les véhicules"

➤ Test du web service consulter les séances



The screenshot shows a REST client interface with two tabs at the top: 'GET http://127.0.0.1:8000/api/admin/' and 'GET http://127.0.0.1:8000/api/admin/agenda'. The second tab is selected, displaying a JSON response for querying sessions. The JSON data is as follows:

```
1 [ ]  
2 {  
3     "id": 1,  
4     "date": "2018-05-12",  
5     "HD": "09:00:00",  
6     "HF": "10:00:00",  
7     "moniteur_id": 1,  
8     "vehicule_id": 2,  
9     "candidat_id": 1,  
10    "created_at": null,  
11    "updated_at": null  
12 },  
13 {  
14     "id": 2,  
15     "date": "2018-05-12",  
16     "HD": "11:00:00",  
17     "HF": "12:00:00",  
18     "moniteur_id": 1,  
19     "vehicule_id": 2,  
20     "candidat_id": 1,  
21     "created_at": null,  
22     "updated_at": null  
23 }  
24 ]|
```

Figure 62 Test du web service "Consulter les séances"

5.2 Les tests unitaires

Nous allons présenter dans cette partie quelques cas de tests unitaires réalisés ainsi que le raisonnement adopté.

- **Le test unitaire du cas « Ajouter moniteur »**
 - Raisonnement

Dans le but de tester le bon fonctionnement de l'ajout d'un moniteur on a suivi le raisonnement suivant :

1. La création d'un objet JSON ayant comme paramètres la méthode adaptée, la route du contrôleur correspondant ainsi que l'objet JSON du moniteur à créer.
2. Le résultat fourni par la méthode d'ajout est donné par « assertJson () ».

On peut considérer que le test est réussi si le résultat fournit est conforme au résultat attendu. Dans cette situation la méthode « assertEquals () » prend la valeur 'true'.

➤ **Code source de la méthode du cas d'utilisation « Ajouter moniteur »**

```
class MoniteurTest extends TestCase
{
    use RefreshDatabase;
    /**
     * A basic feature test example.
     *
     * @return void
     */
    public function testAjout_Moniteur_is_true()
    {
        $response = $this->json('POST', 'api/admin/moniteurs/store',
        [
            'nom' => 'Ali',
            'telephone'=> '26939696' ,
            'prenom' =>'hattab' ,
            'password' =>'hatba',
            'email' => 'AliHattab@gmail.com' ,
            'dateNaissance'=> '28/11/1998' ,
            'pseudo' => '@ali' ,
            'salaire' => '2533',
            'dateEmbauche' =>'20/11/2020' ,
            'cin' => '2356896' ,
        ]);
        $response
            ->assertStatus(200)
            ->assertJson([
                "result"=>true,
                "message"=> "success"
            ]);
    }
}
```

Figure 63 méthode de test unitaire ajouter un moniteur

➤ **Capture d'écran représentant le résultat du test du cas « Ajouter moniteur »**

```
PS C:\Users\ASUS\Desktop\Auto_backend> vendor\bin\phpunit --filter testAjout_Moniteur_is_true
PHPUnit 7.5.10 by Sebastian Bergmann and contributors.

.

1 / 1 (100%)

Time: 5.49 seconds, Memory: 18.00 MB

OK (1 test, 2 assertions)
```

Figure 64 Résultat du test ajout d'un moniteur

- Capture d'écran représentant le résultat du test du cas « Ajouter moniteur » en violant certaines contraintes

```
PS C:\Users\ASUS\Desktop\Auto_backend> vendor\bin\PHPUnit --filter testAjout_Moniteur_is_true
PHPUnit 7.5.10 by Sebastian Bergmann and contributors.

F
1 / 1 (100%)

Time: 7.73 seconds, Memory: 18.00 MB

There was 1 failure:

1) Tests\Feature\MoniteurTest::testAjout_Moniteur_is_true
Expected status code 200 but received 500.
Failed asserting that false is true.
```

Figure 65 Résultat d'erreur d'ajout d'un moniteur

VI. Conclusion

Dans ce chapitre qui concerne le premier sprint, nous avons analysé, conçu et testé les « user story » du premier sprint. Dans le chapitre suivant nous allons nous intéresser à la mise en œuvre de deuxième sprint.

Chapitre 3

Le deuxième sprint

Introduction

Après avoir terminé le premier sprint, et en suivant le même principe, nous allons présenter le deuxième sprint. Tout d'abord nous allons présenter le deuxième sprint backlog. Nous poursuivons par une analyse, une conception et enfin une implémentation et un test du deuxième sprint. Pour terminer par une conclusion.

I. Spécification des besoins

1.1 Le Sprint backlog

La figure ci-dessous présente le sprint backlog du deuxième sprint.

User story ID	User Story	Id Tache	Tache
8.1	En tant qu'utilisateur je souhaite je dois m'authentifier pour accéder à mon espace personnel.	8.1.1	Réaliser la modélisation et la conception de la fonctionnalité “S’authentifier”.
		8.1.2	Développement du cas “S’authentifier”.
		8.1.3	Test du cas “S’authentifier”.
11.5	En tant qu'administrateur je souhaite consulter la liste des candidats.	11.5.1	Réaliser la modélisation et la conception de la fonctionnalité “Consulter la liste des candidats”.
		11.5.2	Développement du cas “Consulter la liste des candidats”.
		11.5.3	Test du cas “Consulter la liste des candidats”.
11.1	En tant qu'administrateur je souhaite ajouter un candidat.	11.1.1	Réaliser la modélisation et la conception de la fonctionnalité “Ajouter candidat”.
		11.1.2	Développement du cas “Ajouter candidat”.
		11.1.3	Test du cas “Ajouter candidat”.
11.2	En tant qu'administrateur je souhaite modifier un candidat.	11.2.1	Réaliser la modélisation et la conception de la fonctionnalité “Modifier candidat”.
		11.2.2	Développement du cas “Modifier candidat”.
		11.2.3	Test du cas “Modifier candidat”.

11.3	En tant qu'administrateur je souhaite supprimer un candidat.	11.3.1	Réaliser la modélisation et la conception de la fonctionnalité “Supprimer un candidat”.
		11.3.2	Développement du cas “Supprimer un candidat”.
		11.3.3	Test du cas “Supprimer un candidat ”.
11.4	En tant qu'administrateur je souhaite consulter un candidat.	11.4.1	Réaliser la modélisation et la conception de la fonctionnalité “Consulter un candidat”.
		11.4.2	Développement du cas “Consulter un candidat”.
		11.4.3	Test du cas “Consulter un candidat ”.
13.2	En tant qu'utilisateur je souhaite consulter mon compte.	13.2.1	Réaliser la modélisation et la conception de la fonctionnalité “Consulter mon compte ”.
		13.2.2	Développement du cas “Consulter mon compte ”.
		13.2.3	Test du cas “Consulter mon compte ”.
13.1	En tant qu'utilisateur je souhaite modifier mon compte.	13.1.1	Réaliser la modélisation et la conception de la fonctionnalité “Modifier mon compte ”.
		13.1.2	Développement du cas “ Modifier mon compte ”.
		13.1.3	Test du cas “ Modifier mon compte ”.
14.1	En tant que candidat je souhaite consulter les cours.	14.1.1	Réaliser la modélisation et la conception de la fonctionnalité “Consulter les cours”.
		14.1.2	Développement du cas “Consulter les cours”.
		14.1.3	Test du cas “Consulter les cours”
15.1	En tant que candidat je souhaite passer les tests en ligne.	15.1.1	Réaliser la modélisation et la conception de la fonctionnalité “Passer les tests”.

		15.1.2	Développement du cas “Passer les tests”.
		15.1.3	Test du cas “Passer les tests”.
9.1	En tant que moniteur je souhaite consulter les informations des candidats.	9.1.1	Réaliser la modélisation et la conception de la fonctionnalité “Consulter informations des candidats”.
		9.1.2	Développement du cas “Consulter information des candidats”.
		9.1.3	Test du cas “Consulter information des candidats”.
7.1	En tant que moniteur et candidat, je souhaite consulter planning.	7.1.1	Réaliser la modélisation et la conception de la fonctionnalité “Consulter planning”.
		7.1.2	Développement du cas “Consulter planning”.
		7.1.3	Test du cas “Consulter planning”.
12.1	En tant qu’administrateur je souhaite ajouter un cours.	12.1.1	Réaliser la modélisation et la conception de la fonctionnalité “Ajouter un cours”.
		12.1.2	Développement du cas “Ajouter un cours”.
		12.1.3	Test du cas “Ajouter un cours”.
12.2	En tant qu’administrateur je souhaite supprimer un cours.	12.2.1	Réaliser la modélisation et la conception de la fonctionnalité “Supprimer un cours”.
		12.2.2	Développement du cas “Supprimer un cours”.
		12.2.3	Test du cas “Supprimer un cours”.
12.3	En tant qu’administrateur je souhaite consulter un cours.	12.3.1	Réaliser la modélisation et la conception de la fonctionnalité “Consulter un cours”.
		12.3.2	Développement du cas “Consulter un cours”.
		12.3.3	Test du cas “Consulter un cours”.

	12.4	En tant qu'administrateur je souhaite consulter la liste des cours.	12.4.1	Réaliser la modélisation et la conception de la fonctionnalité “Consulter la liste des cours”.
			12.4.2	Développement du cas “Consulter la liste des cours”.
			12.4.3	Test du cas “Consulter la liste des cours”.
	6.1	En tant qu'administrateur je souhaite enregistrer les gains.	6.1.1	Réaliser la modélisation et la conception de la fonctionnalité “Enregistrer les gains”.
			6.1.2	Développement du cas “Enregistrer les gains”.
			6.1.3	Test du cas “Enregistrer les gains”.
	6.2	En tant qu'administrateur je souhaite modifier un gain.	6.2.1	Réaliser la modélisation et la conception de la fonctionnalité “Modifier un gain”.
			6.2.2	Développement du cas “Modifier un gain”.
			6.1.3	Test du cas “Modifier un gain”.
	6.3	En tant qu'administrateur je souhaite supprimer un gain.	6.3.1	Réaliser la modélisation et la conception de la fonctionnalité “Supprimer un gain”.
			6.3.2	Développement du cas “Supprimer un gain”.
			6.3.3	Test du cas “Supprimer un gain”.
	6.4	En tant qu'administrateur je souhaite afficher les statistiques.	6.4.1	Réaliser la modélisation et la conception de la fonctionnalité “Afficher statistiques”.
			6.4.2	Développement du cas “Afficher statistiques”.
			6.4.3	Test du cas “Afficher statistiques”.
	6.5	En tant qu'administrateur je souhaite enregistrer les dépenses.	6.5.1	Réaliser la modélisation et la conception de la fonctionnalité “Enregistrer les dépenses”.
			6.5.2	Développement du cas “Enregistrer les dépenses”.

		6.5.3	Test du cas “Enregistrer les dépenses”.
6.6	En tant qu'administrateur je souhaite modifier une dépense.	6.6.1	Réaliser la modélisation et la conception de la fonctionnalité “Modifier une dépense”.
		6.6.2	Développement du cas “Modifier une dépense”.
		6.6.3	Test du cas “Modifier une dépense”.
6.7	En tant qu'administrateur je souhaite supprimer une dépense.	6.7.1	Réaliser la modélisation et la conception de la fonctionnalité “Supprimer une dépense”.
		6.7.2	Développement du cas “Supprimer une dépense”.
		6.7.3	Test du cas “Supprimer une dépense”.
6.8	En tant qu'administrateur je souhaite éditer une facture.	6.8.1	Réaliser la modélisation et la conception de la fonctionnalité “Editer une facture”.
		6.8.2	Développement du cas “Editer une facture”.
		6.8.3	Test du cas “Editer une facture”.
6.9	En tant qu'administrateur je souhaite modifier une facture.	6.9.1	Réaliser la modélisation et la conception de la fonctionnalité “Modifier une facture”.
		6.9.2	Développement du cas “Modifier une facture”.
		6.9.3	Test du cas “Modifier une facture”.
6.10	En tant qu'administrateur je souhaite supprimer une facture.	6.10.1	Réaliser la modélisation et la conception de la fonctionnalité “Supprimer une facture”.
		6.10.2	Développement du cas “Supprimer une facture”.
		6.10.3	Test du cas “Supprimer une facture”.
10.1	En tant que visiteur je souhaite consulter le site vitrine.	10.1.1	Réaliser la modélisation et la conception de la fonctionnalité “Consulter le site vitrine”.

	10.1.2	Développement du cas “ Consulter le site vitrine”.
	10.1.3	Test du cas “Consulter le site vitrine ”.

Tableau 27 Backlog sprint 2

II. Analyse des cas du deuxième sprint

2.1Classification des cas d'utilisations par acteur

Le tableau ci-dessous présente la classification des cas d'utilisation par acteur.

Acteur	Cas d'utilisation
Administrateur, moniteur, candidat	S'authentifier
Administrateur	Gérer la comptabilité
Administrateur	Gérer les candidats
Administrateur	Gérer les cours
Administrateur, moniteur, candidat	Gérer compte
Candidat, moniteur	Consulter planning
Moniteur	Consulter information des candidats
Candidat	Consulter cours
Candidat	Passer test en ligne
Visiteur	Consulter le site vitrine

Tableau 28 Classification des cas d'utilisation par acteur du deuxième sprint

2.2Diagramme de cas d'utilisation de deuxième sprint

La figure ci-dessous représente le diagramme de cas d'utilisation global du deuxième sprint. L'administrateur aura le droit, après l'utilisation d'un mot de passe et d'un identifiant, de gérer les plannings les véhicules ainsi que les moniteurs. Ces derniers peuvent aussi grâce à l'application mobile gérer les rapports des séances.

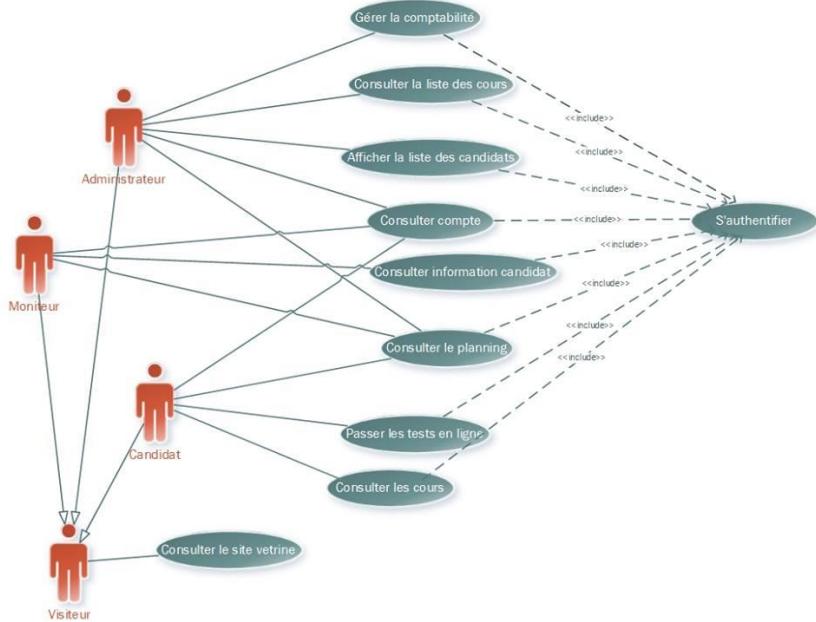


Figure 66 Diagramme de cas d'utilisation du deuxième sprint

2.3 Description des cas d'utilisation de l'application web du deuxième sprint

Au cours de cette partie nous allons mettre en place le raffinement des cas d'utilisation de notre application web ainsi qu'une description des différents scénarios qui peuvent se présenter afin qu'ils soient plus clairs et compréhensibles.

2.3.1 Analyse du cas d'utilisation « S'authentifier »

- Description textuelle du cas d'utilisation « S'authentifier »

Cas D'utilisation	S'authentifier.
Acteur	Administrateur, candidat.
Pré condition	L'utilisateur possède des identifiants de connexion.
Post condition	Acteur authentifié
Scénario principal	<ol style="list-style-type: none"> 3. Le système affiche le formulaire de connexion 4. L'utilisateur saisit ses identifiants (login et mot de passe) dans les champs appropriés et valide le formulaire. 5. Le système vérifie les informations saisies par l'administrateur. 6. Le système affiche l'interface appropriée.
Scénario alternatif	Un message d'erreur est affiché si le login et/ou le mot de passe sont erronés.

Tableau 29 Description textuelle du cas d'utilisation « S'authentifier »

2.3.2 Analyse du cas d'utilisation « Gérer comptabilité »

- Raffinement du cas d'utilisation « Gérer comptabilité »

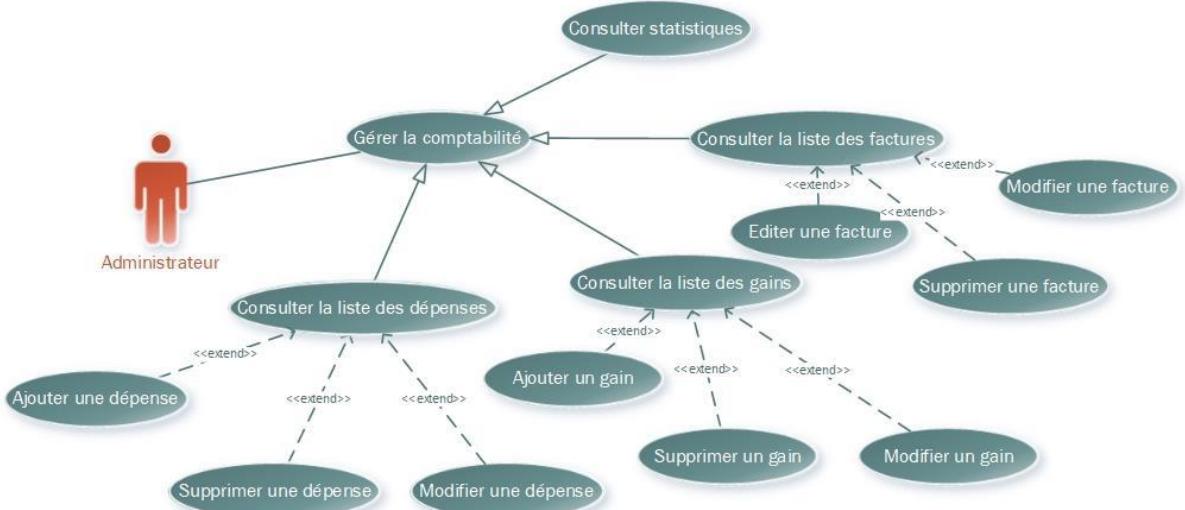


Figure 67 Diagramme de cas d'utilisation “Gérer comptabilité”

- Description textuelle « Consulter la liste des dépenses »

Cas D'utilisation

Consulter la liste des dépenses.

Acteur	Administrateur.
Pré condition	L'administrateur s'est authentifié
Post condition	Dépenses consultées.
Scénario principal	<ol style="list-style-type: none"> 1. L'administrateur clique sur le bouton “Dépenses”. 2. Le système affiche la liste des dépenses.
Scénario alternatif	Le système affiche une page vide.
Extension	<ul style="list-style-type: none"> - L'administrateur peut modifier une dépense. - L'administrateur peut supprimer une dépense. - L'administrateur peut ajouter une dépense.

Tableau30 Description textuelle du cas d'utilisation “Consulter la liste des dépenses”

- Description textuelle « Ajouter une dépense »

Cas d'utilisation

Ajouter une dépense

Acteur	Administrateur
Pré condition	Administrateur authentifié et a affiché la liste des dépenses

Post condition	<p>La dépense est ajoutée.</p> <ol style="list-style-type: none"> 1. L'administrateur clique sur le bouton « ajouter ». 2. Le système affiche un formulaire d'ajout. 3. L'administrateur remplit le formulaire d'ajout. 4. Le système vérifie les données saisies par l'administrateur. 5. Le système affiche un message indiquant que la dépense est ajoutée.
Scénario alternatif	<p>Un message d'erreur est affiché si l'un des champs obligatoires est vide et/ou invalide.</p>

Tableau 31 Description textuelle du cas d'utilisation “Ajouter une dépense”

- **Description textuelle « Modifier une dépense »**

Cas d'utilisation	Modifier une dépense
Acteur	Administrateur
Pré condition	L'administrateur s'est authentifié et a affiché la liste des dépenses.
Post condition	Dépense modifiée.
Description du scénario principale	<ol style="list-style-type: none"> 1. L'administrateur clique sur « Modifier ». 2. Le système affiche le formulaire de modification d'une dépense. 3. L'administrateur modifie le formulaire. 4. L'administrateur valide le formulaire. 5. Le système vérifie les informations saisies par l'administrateur. 6. Le système affiche un message indiquant que les dépenses sont mises à jour
Scénario alternatif	<p>Un message d'erreur est affiché si l'un des champs obligatoires est vide et/ou invalide</p>

Tableau 32 Description textuelle du cas d'utilisation “Modifier une dépense”

- **Description textuelle « Supprimer une dépense »**

Cas d'utilisation	Supprimer une dépense
Acteur	Administrateur

Pré condition	L'administrateur s'est authentifié et a affiché la liste des dépenses.
Post condition	Dépense supprimée.
Description du scénario principale	<ol style="list-style-type: none"> 1. L'administrateur clique sur le bouton supprimer 2. Le système affiche un message de demande de confirmation. 3. L'administrateur valide la suppression. 4. Le système affiche un message indiquant que la dépense est supprimée.

Tableau 33 Description textuelle du cas d'utilisation “Supprimer une dépense”

- **Description textuelle « Consulter la liste des gains »**

La description textuelle du cas d'utilisation consulter la liste des gains se fait de la même façon que consulter la liste des dépenses. (Tableau 30, page 77).

- **Description textuelle « Ajouter un gain »**

La description textuelle du cas d'utilisation ajouter un gain se fait de la même façon qu'ajouter une dépense. (Tableau 31, page 77).

- **Description textuelle « Modifier un gain »**

La description textuelle du cas d'utilisation modifier un gain se fait de la même façon que modifier une dépense. (Tableau 32, page 78).

- **Description textuelle « Supprimer un gain »**

La description textuelle du cas d'utilisation supprimer un gain se fait de la même façon que supprimer une dépense. (Tableau 33, page 78).

- **Description textuelle « Consulter la liste des factures »**

La description textuelle du cas d'utilisation consulter la liste des factures se fait de la même façon que consulter la liste des dépenses. (Tableau 30, page 77).

- **Description textuelle « Editer une facture »**

La description textuelle du cas d'utilisation éditer une facture se fait de la même façon qu'ajouter une dépense. (Tableau 31, page 77).

- **Description textuelle « Modifier une facture »**

La description textuelle du cas d'utilisation modifier une facture se fait de la même façon que modifier une dépense. (Tableau 32, page 78).

- **Description textuelle « Supprimer une facture »**

La description textuelle du cas d'utilisation supprimer une facture se fait de la même façon que supprimer une dépense. (Tableau 33, page 78).

- **Description textuelle « Consulter statistiques »**

La description textuelle du cas d'utilisation consulter statistiques se fait de la même façon que consulter la liste des dépenses. (Tableau 30, page 77).

2.3.3 Analyse du cas d'utilisation « Gérer les cours »

- Raffinement du cas d'utilisation « Gérer les cours »

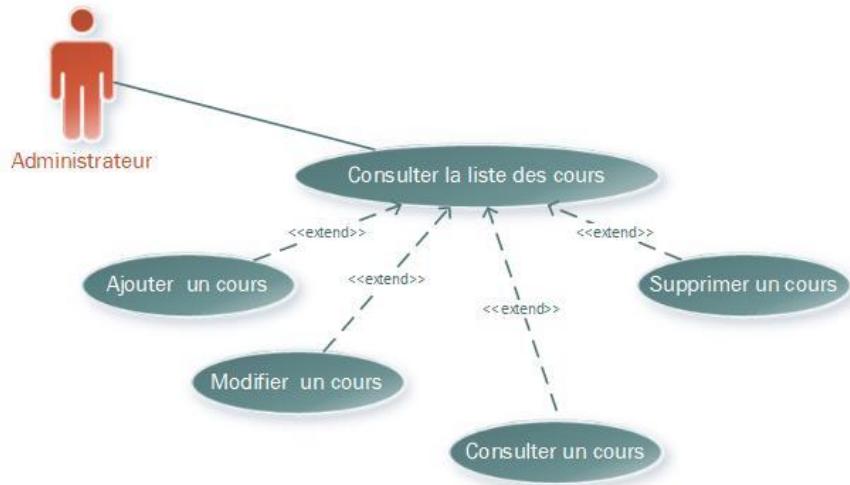


Figure 68 Diagramme de cas d'utilisation “Gérer les cours”

- **Description textuelle du cas d'utilisation « Consulter la liste des cours »**
La description textuelle du cas d'utilisation consulter la liste des cours se fait de la même façon que consulter la liste des dépenses. (Tableau 30, page 77).
- **Description textuelle du cas d'utilisation « Modifier un cours »**
La description textuelle du cas d'utilisation modifier un cours se fait de la même façon que modifier une dépense. (Tableau 32, page 78).
- **Description textuelle du cas d'utilisation « Ajouter un cours »**
La description textuelle du cas d'utilisation ajouter un cours se fait de la même façon qu'ajouter une dépense. (Tableau 31, page 77).
- **Description textuelle du cas d'utilisation « Supprimer un cours »**
La description textuelle du cas d'utilisation supprimer un cours se fait de la même façon que supprimer une dépense. (Tableau 33, page 78).
- **Description textuelle du cas d'utilisation « Consulter un cours »**

Cas D'utilisation	Consulter un cours.
Acteur	Administrateur.
Pré condition	L'administrateur s'est authentifié et a affiché la liste des cours.
Post condition	Cours consulté.
Scénario principal	<ol style="list-style-type: none"> 1. L'administrateur clique sur le bouton “Consulter”. 2. Le système affiche les détails correspondants au cours sélectionné.

Tableau 34 Description textuelle du cas d'utilisation “Consulter un cours”

2.3.4 Analyse du cas d'utilisation « Consulter liste des candidats »

L'analyse du cas d'utilisation « consulter liste des candidats » est similaire au cas d'utilisation « consulter la liste des dépenses » (page 74 ->76)

2.3.5 Analyse du cas d'utilisation « Consulter compte »

- Raffinement du cas d'utilisation « Consulter compte »

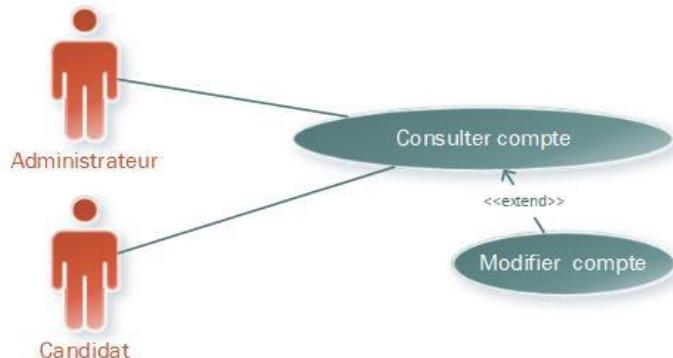


Figure 69 Diagramme de cas d'utilisation “Consulter compte”

- Description textuelle du cas d'utilisation « Consulter compte »

Cas d'utilisation	Consulter compte
Acteur	Administrateur, candidat
Pré condition	Acteur authentifié.
Post condition	Ensembles des informations affichées
Description du scénario principale	<ol style="list-style-type: none"> 1. L'acteur clique sur le lien « Profil » 2. Le système affiche les informations liées à l'acteur.

Tableau 35 Description textuelle du cas d'utilisation “Consulter compte”

- Description textuelle du cas d'utilisation « Modifier compte »

Cas d'utilisation	Modifier compte
Acteur	Administrateur, candidat
Pré condition	L'acteur s'est authentifié, il se trouve dans l'interface de son profil.
Post condition	Ensembles des informations modifiées
Description du scénario principale	<ol style="list-style-type: none"> 1. L'utilisateur clique sur « Modifier ». 2. Le système affiche le formulaire de modification des informations.

Scénario alternatif	3. L'utilisateur modifie le formulaire.
	4. L'utilisateur valide le formulaire.
	5. Le système vérifie les informations saisies par l'utilisateur.
	6. Le système affiche un message indiquant que les informations sont mises à jour
	Un message d'erreur est affiché si l'un des champs obligatoires est vide et/ou invalide.

Tableau 36 Description textuelle du cas d'utilisation “Modifier compte”

2.3.6 Analyse du cas d'utilisation « Consulter le planning »

- Description textuelle du cas d'utilisation « Consulter le planning »

Cas d'utilisation	Consulter le planning
Acteur	Administrateur, candidat
Pré condition	Acteur authentifié.
Post condition	Ensembles des informations relatives au planning sont affichées
Description du scénario principale	<ol style="list-style-type: none"> 1. L'acteur clique sur le lien « Planning » 2. Le système affiche le planning

Tableau 37 Description textuelle du cas d'utilisation “Consulter le planning”

2.3.7 Analyse de cas d'utilisation « Passer un test en ligne »

- Description textuelle du cas d'utilisation « Passer un test en ligne »

Cas d'utilisation	Passer un test en ligne
Acteur	Candidat
Pré condition	Candidat authentifié
Post condition	Test passé
Description du scénario principale	<ol style="list-style-type: none"> 1. Le candidat clique sur le bouton ‘commencer le test’. 2. Le système affiche plusieurs questions. 3. Le candidat répond aux questions du test. 4. Le système vérifie les informations saisies par le candidat. 5. Le système affiche le résultat du test.

Tableau 38 Description textuelle du cas d'utilisation “Passer un test en ligne”

2.3.8 Analyse de cas d'utilisation « Consulter les cours »

- Description textuelle du cas d'utilisation « Consulter les cours »

Cas d'utilisation	Consulter les cours
Acteur	Candidat
Pré condition	Candidat authentifié
Post condition	Ensembles des cours affichés
Description du scénario principale	<ol style="list-style-type: none"> 1. Le candidat clique sur le lien « cours ». 2. Le système affiche une liste des cours. 3. Le candidat choisit le cours souhaité 4. Le système affiche le cours

Tableau 39 Description textuelle du cas d'utilisation “Consulter les cours”

2.4 Description des cas d'utilisation de l'application mobile du deuxième sprint

Au cours de cette partie nous allons mettre en place le raffinement de nos cas d'utilisation de notre application mobile ainsi qu'une description des différents scénarios qui peuvent se présenter afin qu'ils soient plus clairs et compréhensibles.

On rappelle que seul le moniteur et l'administrateur auront la possibilité d'utiliser la version mobile.

2.4.1 Description textuelle du cas d'utilisation « S'authentifier »

La description textuelle du cas d'utilisation s'authentifier de l'application mobile se fait de la même façon que la fonctionnalité s'authentifier de l'application web.
(Tableau 29, Page 76)

2.4.2 Analyse du cas d'utilisation « Gérer compte »

- Raffinement du cas d'utilisation « Gérer compte »

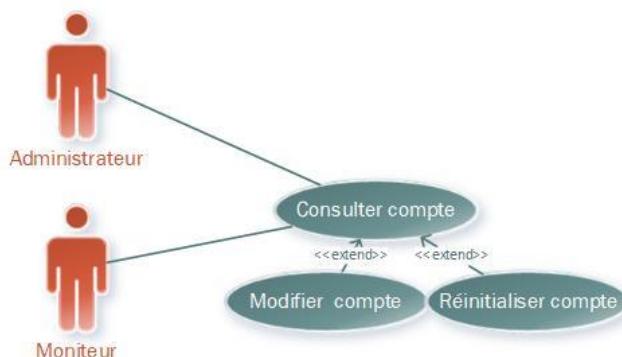


Figure 70 Diagramme de cas d'utilisation “Gérer compte”

- **Description textuelle du cas d'utilisation « Consulter compte »**

Cas d'utilisation	Consulter compte
Acteur	Administrateur ou moniteur
Pré condition	Acteur authentifié
Post condition	Ensembles des informations relatives au compte sont affichées.
Description du scénario principale	<ol style="list-style-type: none"> 1. L'acteur clique sur le lien « Paramètres ». 2. Le système affiche la liste des informations.
Extension	<ul style="list-style-type: none"> - L'acteur peut modifier les informations correspondantes à son compte. - L'acteur peut réinitialiser son mot de passe.

Tableau 40 Description textuelle du cas d'utilisation “Consulter compte”

- **Description textuelle du cas d'utilisation « Modifier compte »**

Cas d'utilisation	Modifier compte
Acteur	Administrateur ou moniteur
Pré condition	<ul style="list-style-type: none"> • Acteur authentifié • L'acteur est dans l'espace « paramètres »
Post condition	Ensembles des données de l'acteur sont modifiées.
Description du scénario principale	<ol style="list-style-type: none"> 1. L'utilisateur clique sur « Modifier ». 2. Le système affiche le formulaire de modification des informations. 3. L'utilisateur modifie le formulaire. 4. L'utilisateur valide le formulaire. 5. Le système vérifie les informations saisies par l'utilisateur. 6. Le système affiche un message indiquant que les informations sont mises à jour.
Scénario alternatif	Un message d'erreur est affiché si l'un des champs obligatoires est vide et/ou invalide.

Tableau 41 Description textuelle du cas d'utilisation “Modifier compte”

- **Description textuelle du cas d'utilisation « Réinitialiser mot de passe »**

Cas d'utilisation	Réinitialiser mot de passe
Acteur	Administrateur ou moniteur
Pré condition	<ul style="list-style-type: none"> • Acteur authentifié. • L'acteur est déjà dans « paramètres ».
Post condition	Mot de passe de l'acteur est modifié.
Description du scénario principale	<ol style="list-style-type: none"> 1. L'utilisateur clique sur « Réinitialiser mot de passe ». 2. Le système affiche le formulaire de réinitialisation de mot de passe. 3. L'utilisateur remplit le formulaire de modification de mot de passe. 4. L'utilisateur valide le formulaire. 5. Le système vérifie les informations saisies par l'utilisateur. 6. Le système affiche un message indiquant que les informations sont mises à jour.
Scénario alternatif	Un message d'erreur est affiché si l'un des champs obligatoires est vide et/ou invalide.

Tableau 42 Description textuelle du cas d'utilisation “Réinitialiser mot de passe”

2.4.3 Analyse du cas d'utilisation « Consulter le planning »

- **Raffinement du cas d'utilisation « Consulter le planning »**

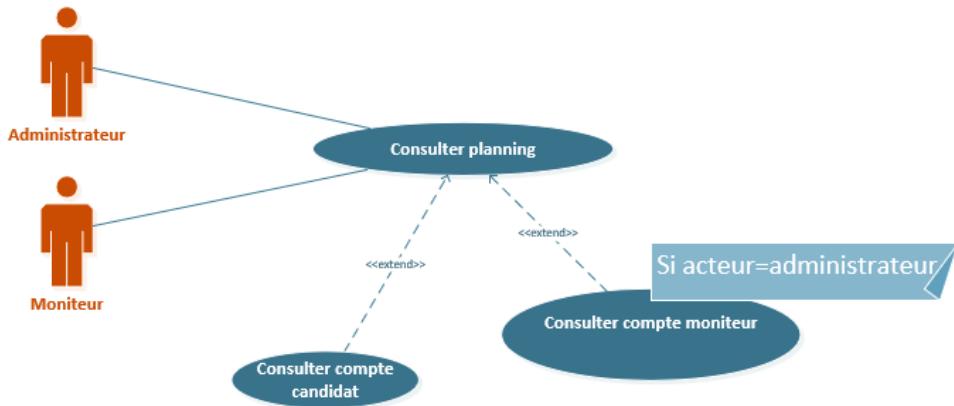


Figure 71 Diagramme de cas d'utilisation “Consulter le planning”

- **Description textuelle du cas d'utilisation « Consulter le planning »**

Cas d'utilisation	Consulter le planning
-------------------	-----------------------

Acteur	Moniteur ou administrateur.
Pré condition	Acteur authentifié.
Post condition	Ensembles des informations relatives au planning sont affichées.
Description du scénario principale	<ol style="list-style-type: none"> 1. L'acteur clique sur le lien « Planning ». 2. Le système affiche le planning.
Extension	<ul style="list-style-type: none"> - Le moniteur peut consulter les informations relatives aux candidats. - L'administrateur peut consulter les informations relatives aux candidats et aux moniteurs.

Tableau 43 Description textuelle du cas d'utilisation “Consulter le planning”

- **Description textuelle du cas d'utilisation « Consulter compte candidat »**

Cas d'utilisation	Consulter compte candidat
Acteur	Moniteur ou administrateur.
Pré condition	Acteur authentifié.
Post condition	Ensembles des informations affichées.
Description du scénario principale	<ol style="list-style-type: none"> 1. L'acteur clique sur le lien « Plus de détails » relatif à un candidat prévu dans le planning. 2. Le système affiche les informations du candidat choisi.

Tableau 44 Description textuelle du cas d'utilisation “Consulter compte candidat”

- **Description textuelle du cas d'utilisation « Consulter compte moniteur »**

Cas d'utilisation	Consulter compte moniteur
Acteur	Administrateur
Pré condition	Administrateur authentifié.
Post condition	Ensembles des informations affichées.
Description du scénario principale	<ol style="list-style-type: none"> 1. L'acteur clique sur le lien « Plus de détails » relatif à un moniteur prévu dans le planning. 2. Le système affiche les informations du moniteur choisi.

Tableau 45 Description textuelle du cas d'utilisation “Consulter compte moniteur”

2.4.4 Analyse du cas d'utilisation « Consulter liste des candidats »

- Raffinement du cas d'utilisation « Consulter liste des candidats »

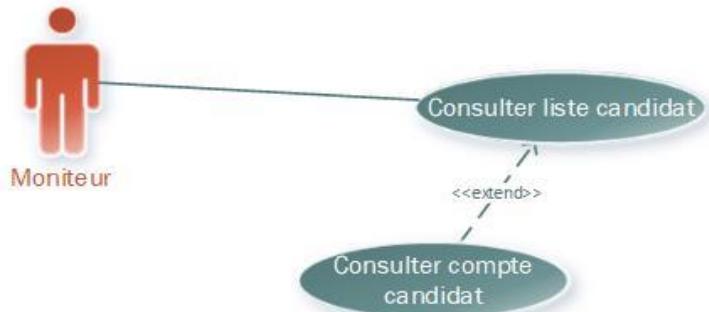


Figure 72 Diagramme de cas d'utilisation “Consulter liste des candidats”

- Description textuelle du cas d'utilisation « Consulter la liste des candidats »

Cas d'utilisation	Consulter la liste des candidats
Acteur	Moniteur
Pré condition	<ul style="list-style-type: none"> • Moniteur authentifié • Moniteur a accédé à son profile
Post condition	Liste des candidats du moniteur affichée.
Description du scénario principale	<ol style="list-style-type: none"> 1. Le moniteur clique sur le bouton « Consulter la liste des candidats ». 2. Le système affiche la liste des candidats suivis par le moniteur.
Extension	- Le moniteur peut voir les détails d'un candidat.

Tableau 46 Description textuelle du cas d'utilisation "Consulter la liste des candidats"

- Description textuelle du cas d'utilisation « Consulter compte candidat »

Cas d'utilisation	Consulter compte candidat
Acteur	Moniteur
Pré condition	<ul style="list-style-type: none"> • Moniteur authentifié • Moniteur est entrain de consulter la liste des candidats
Post condition	Ensembles des informations relatives au candidat sélectionné sont affichées.
	1. Le moniteur clique sur le candidat.

Description du scénario principale

2. Le système affiche les détails du candidat sélectionné par le moniteur.

Tableau 47 Description textuelle du cas d'utilisation "Consulter compte candidat"

III. Conception

3.1 Conception de l'application web

3.1.1 Conception du cas « S'authentifier »

Pour l'administrateur et le candidat la conception de cas s'authentifier se fait de la même façon d'où on a opté pour la réalisation de la conception de cas s'authentifier pour le candidat.

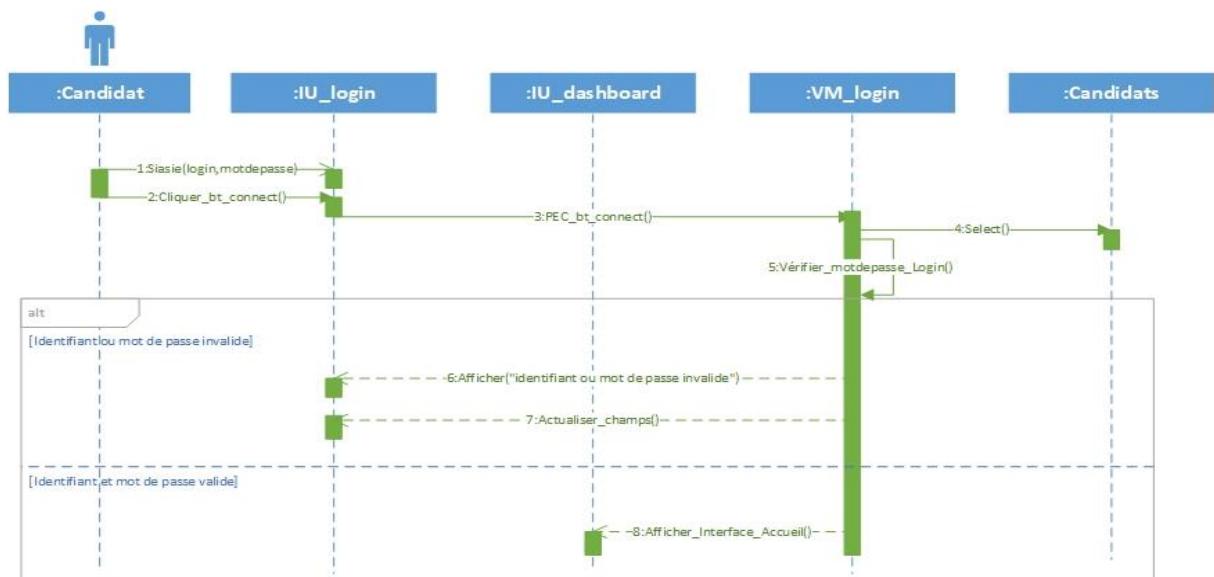


Figure 73 Diagramme de séquence du cas d'utilisation "S'authentifier"

3.1.2 Conception du cas « Consulter la liste des dépenses »

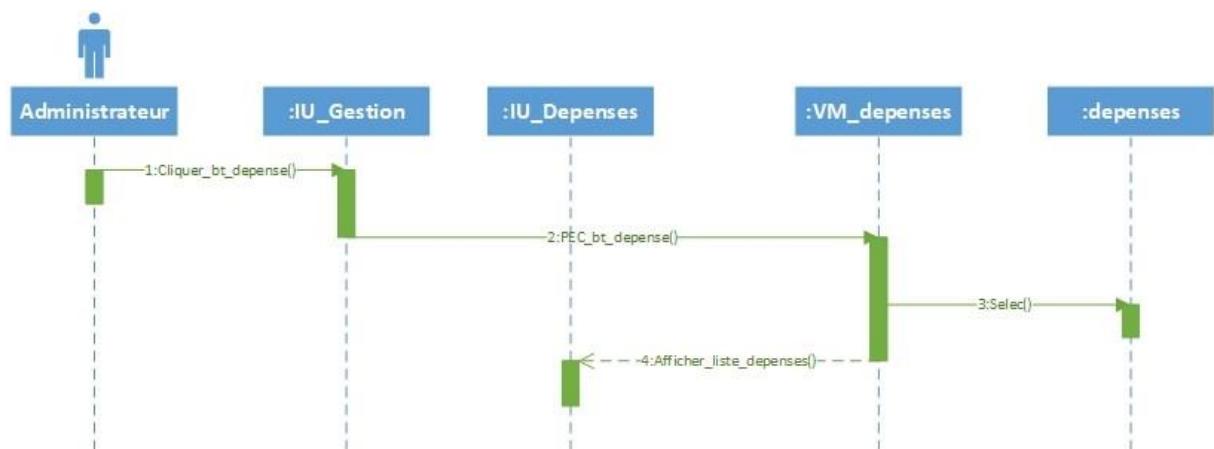


Figure 74 Diagramme de séquence de cas d'utilisation "Consulter la liste des dépenses"

3.1.3 Conception du cas « Ajouter une dépense »

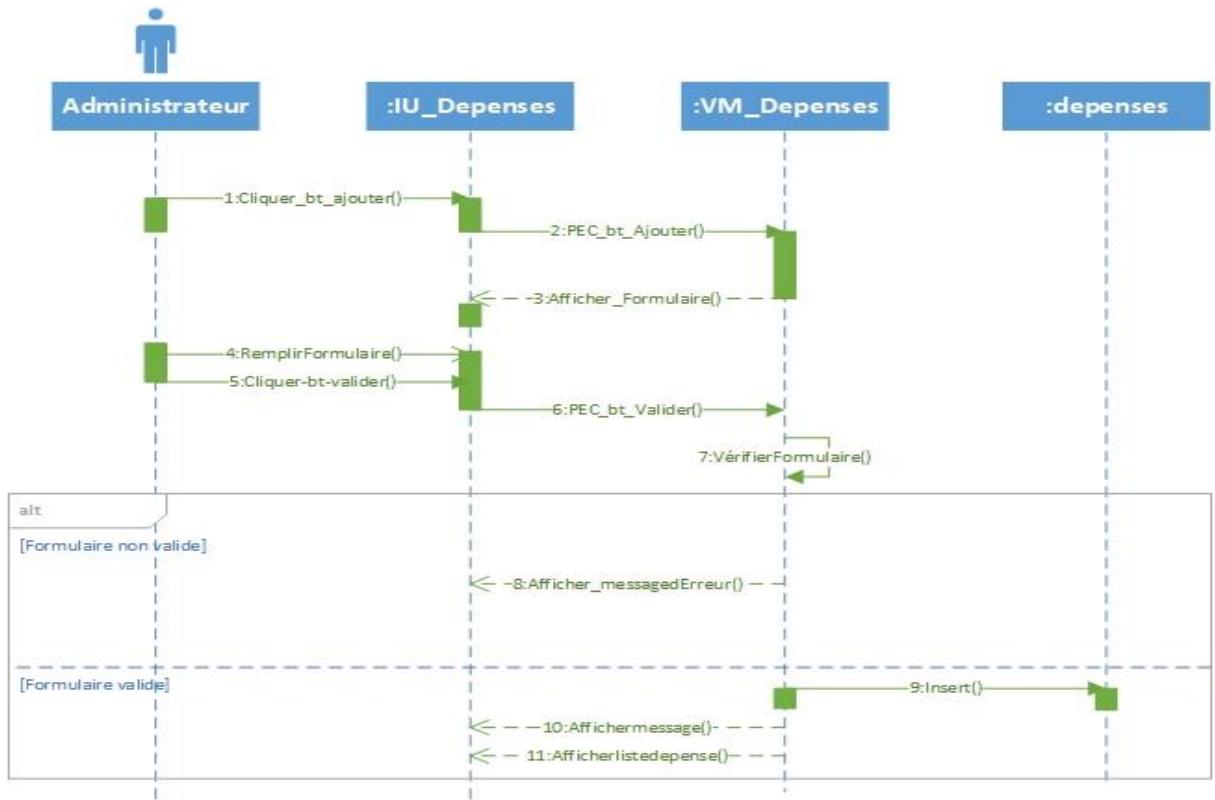


Figure 75 Diagramme de séquence du cas d'utilisation "Ajouter une dépense"

3.1.4 Conception du cas « Modifier une dépense »

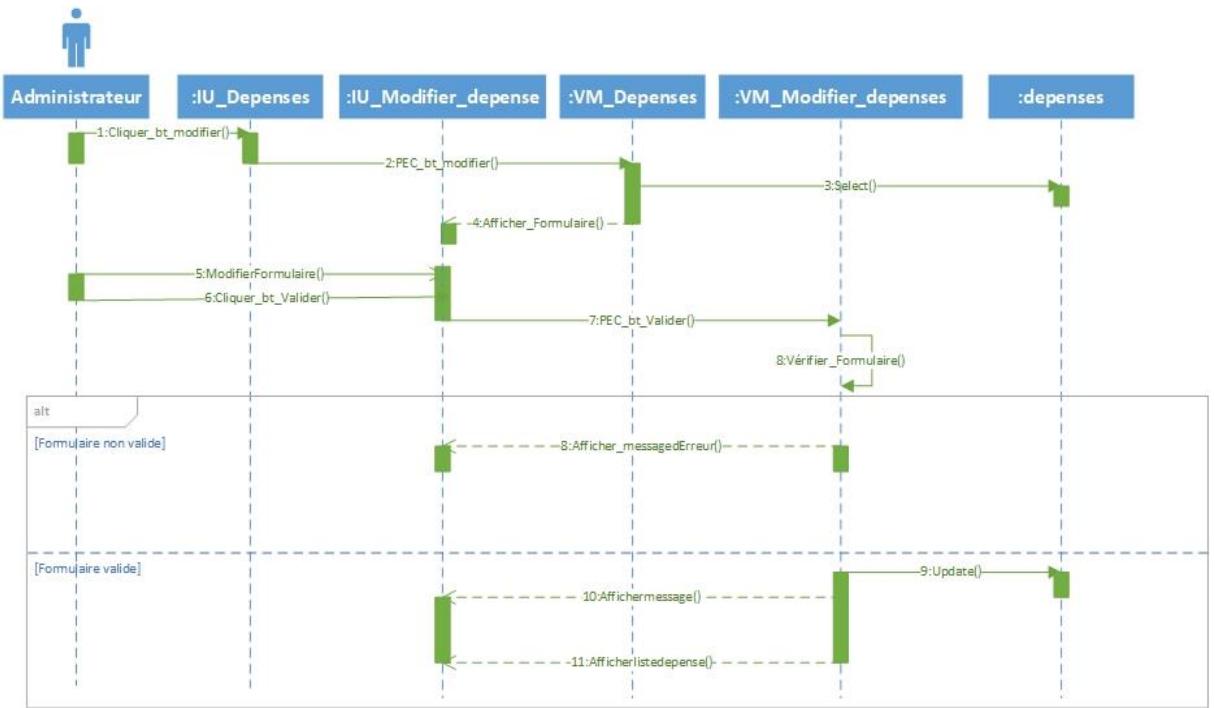


Figure 76 Diagramme de séquence du cas d'utilisation "Modifier une dépense"
3.1.5 Conception du cas « Supprimer une dépense »

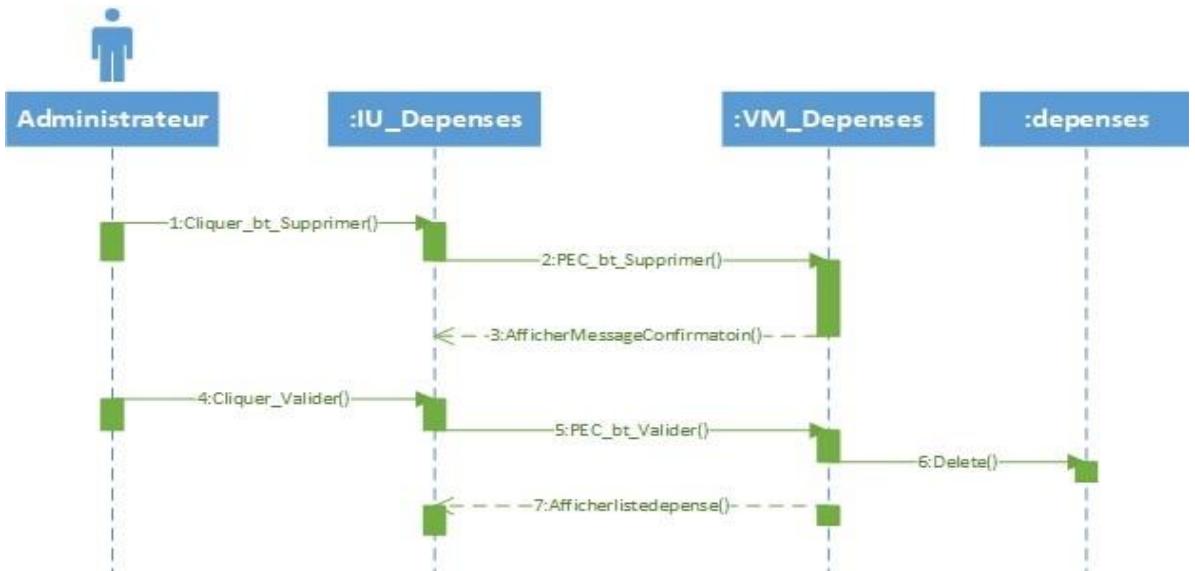


Figure 77 Diagramme de séquence du cas d'utilisation "Supprimer une dépense"

3.1.6 Conception du cas « Consulter statistiques »

La conception de cas consulter statistiques se fait de la même façon que consulter la liste des dépenses. (Figure 74, page 88).

3.1.7 Conception du module « Consulter la liste des gains »

La conception de cas consulter la liste de gains se fait de la même façon que le module consulter les la liste des dépenses. (Figure 74-75-76-77, page 88 -> 89).

3.1.8 Conception du module « Consulter la liste des factures »

La conception de cas consulter la liste des factures se fait de la même façon que le module consulter les la liste des dépenses. (Figure 74-75-76-77, page 88-> 89).

3.1.9 Conception du module « Gérer candidat »

La conception du module gérer candidat est similaire au module gérer les dépenses. (Figure 74-75-76-77, page 88 -> 89).

3.1.10 Conception du cas « Consulter cours »

La conception du module consulter cours par le candidat se fait de la même façon que le module consulter la liste des dépenses. (Figure 74, page 88).

3.1.11 Conception du cas « Consulter le planning »

La conception de cas consulter le planning par l'application web se fait de la même façon que consulter la liste des dépenses. (Figure 74, page 88).

3.1.12 Conception du cas « Consulter compte »

La conception de cas consulter compte par l'application web se fait de la même façon que consulter la liste des dépenses. (Figure 74, page 88).

3.1.13 Conception du cas « Modifier compte »

La conception de modifier compte par l'application web se fait de la même façon que modifier une dépense. (Figure 76, page 89).

3.1.14 Conception du module « Consulter la liste des cours »

La conception du module consulter la liste des cours par l'administrateur se fait de la même façon que le module consulter la liste des dépenses. (Figure 74-75-76-77, page 88 -> 89).

3.1.15 Conception de cas « Passer les tests en ligne »

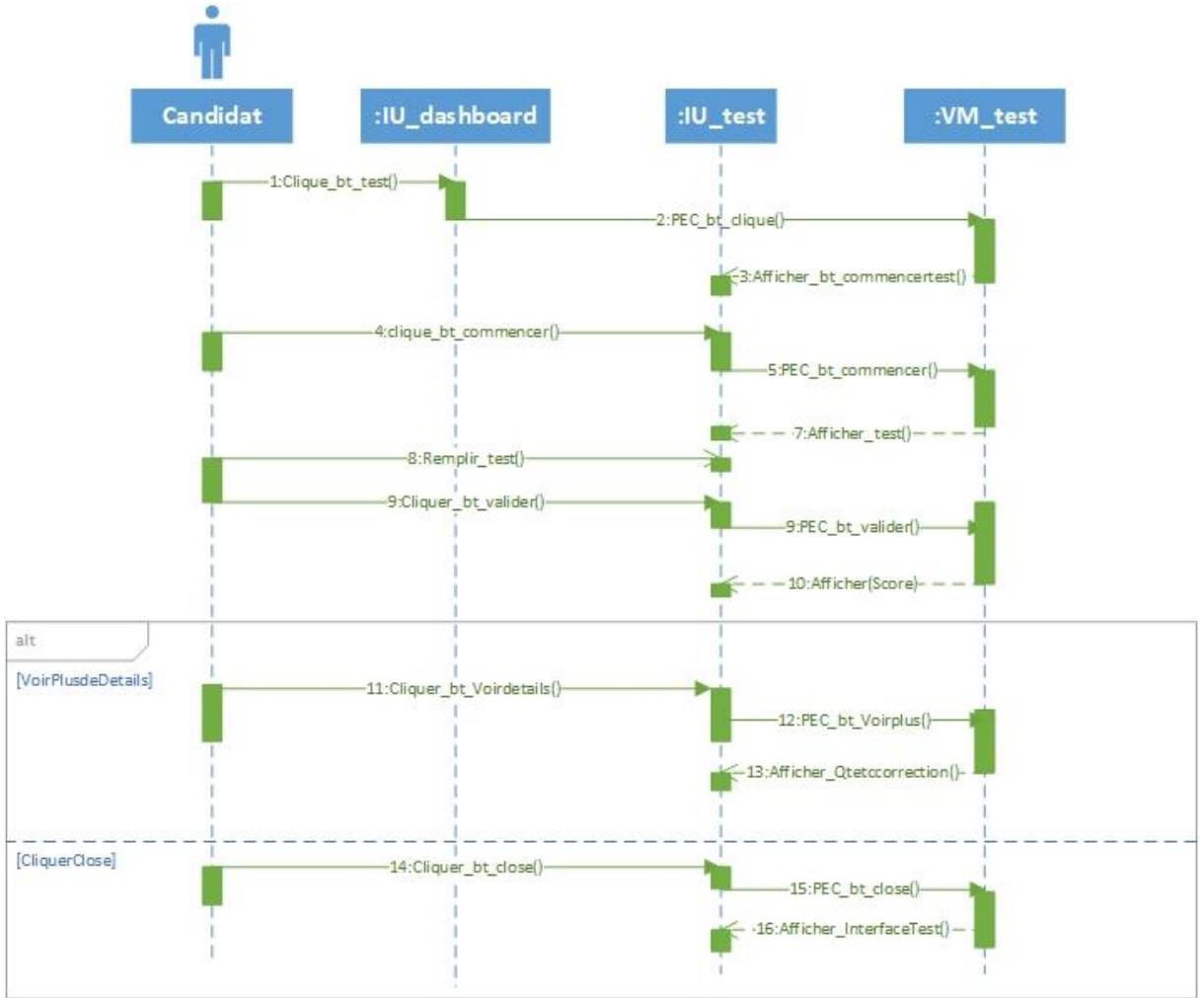


Figure 78 Diagramme de séquence de "Passer les tests en ligne"

3.2 Conception de l'application mobile

3.2.1 Conception du cas « Consulter le planning »

La conception de consulter le planning se fait de la même manière pour l'acteur administrateur et l'acteur moniteur d'où on a opté la réalisation de conception consulter le planning par l'administrateur.

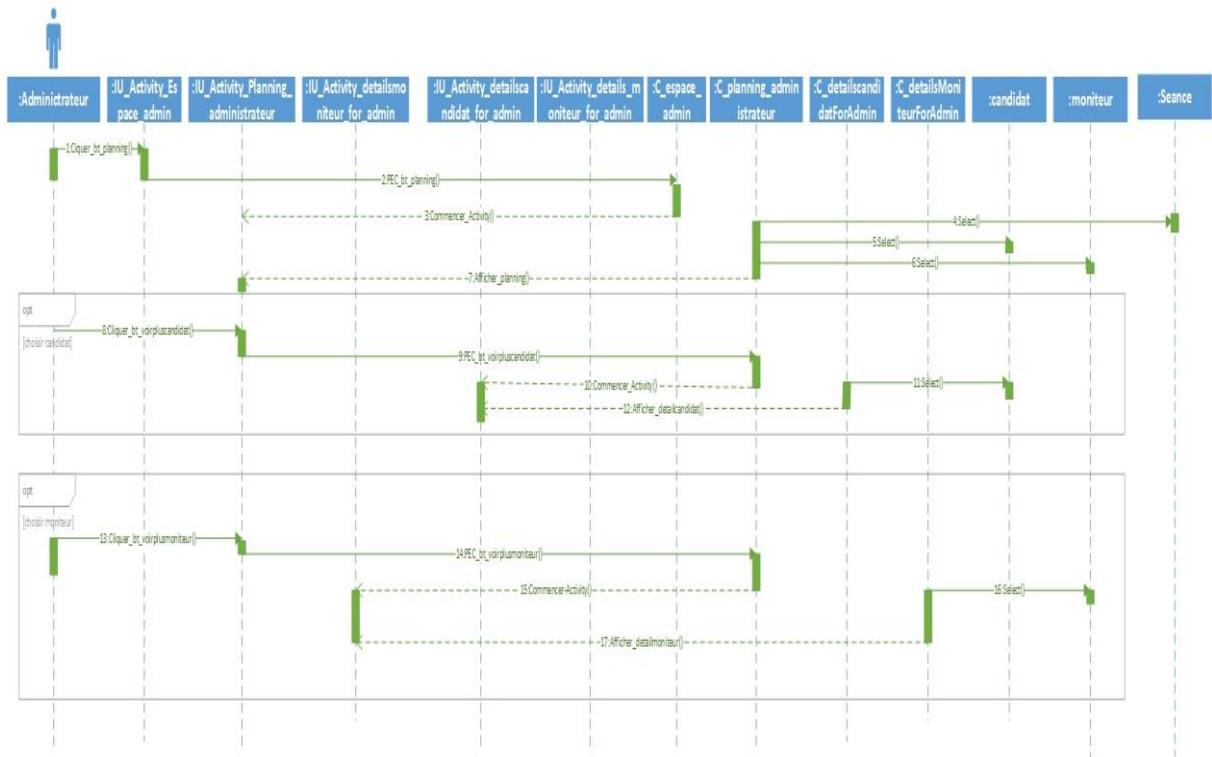


Figure 79 Diagramme de séquence de "Consulter le planning"

3.2.2 Conception du cas « Consulter compte »

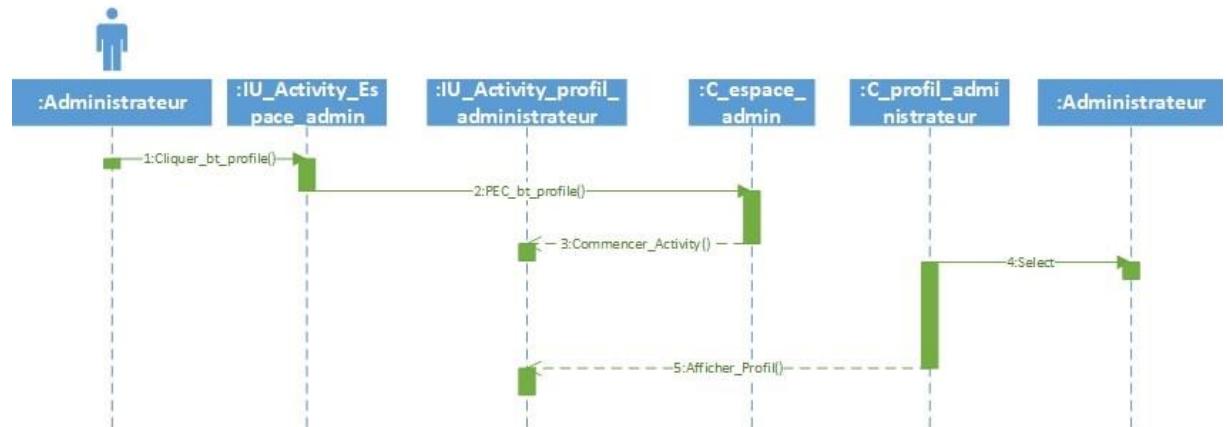


Figure 80 Diagramme de séquence du cas d'utilisation "Consulter compte"

3.2.3 Conception du cas « Modifier compte »

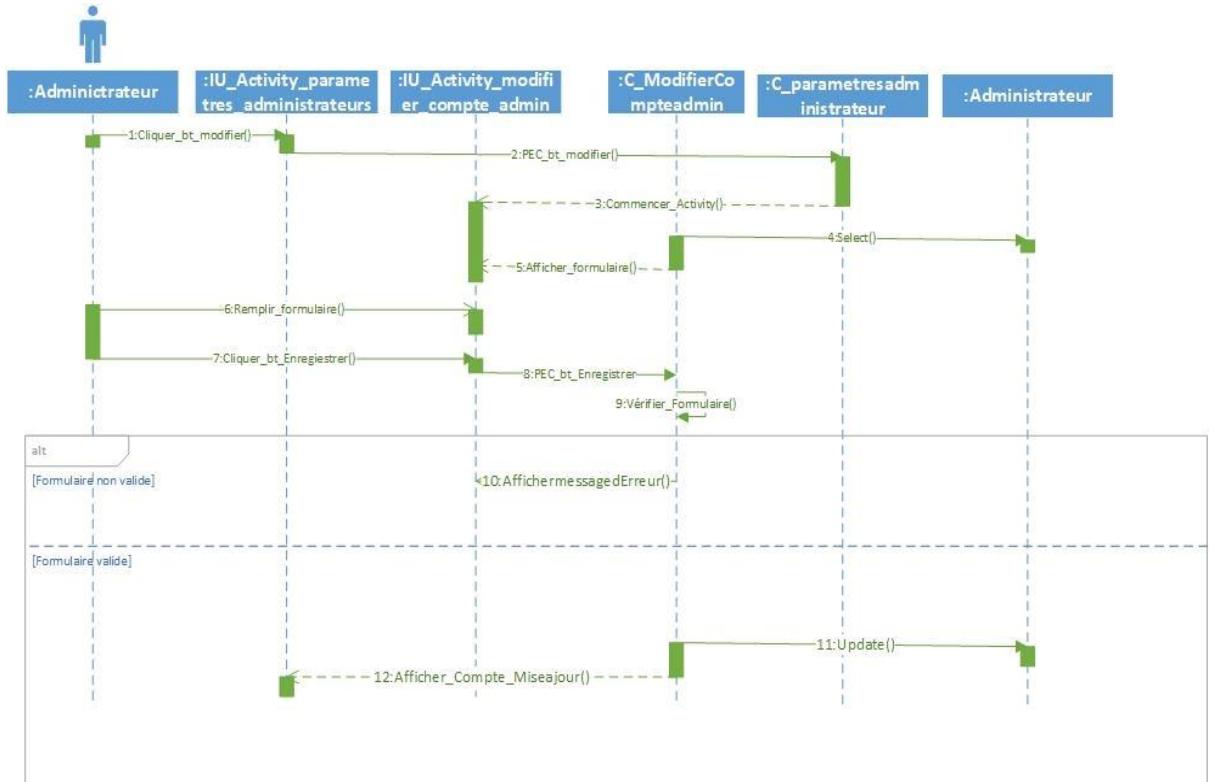


Figure 80 Diagramme de séquence du cas d'utilisation "Modifier compte"

3.2.4 Conception du cas « Réinitialiser mot de passe »

La conception de cas réinitialiser mot de passe par l'application mobile se fait de la même façon que le cas modifier compte (figure 81, page 93).

3.2.5 Conception de cas « S'authentifier »

La conception de cas s'authentifier par l'application mobile se fait de la même façon que s'authentifier de l'application web (figure 73, page 87).

IV. Diagramme de classe globale du deuxième sprint

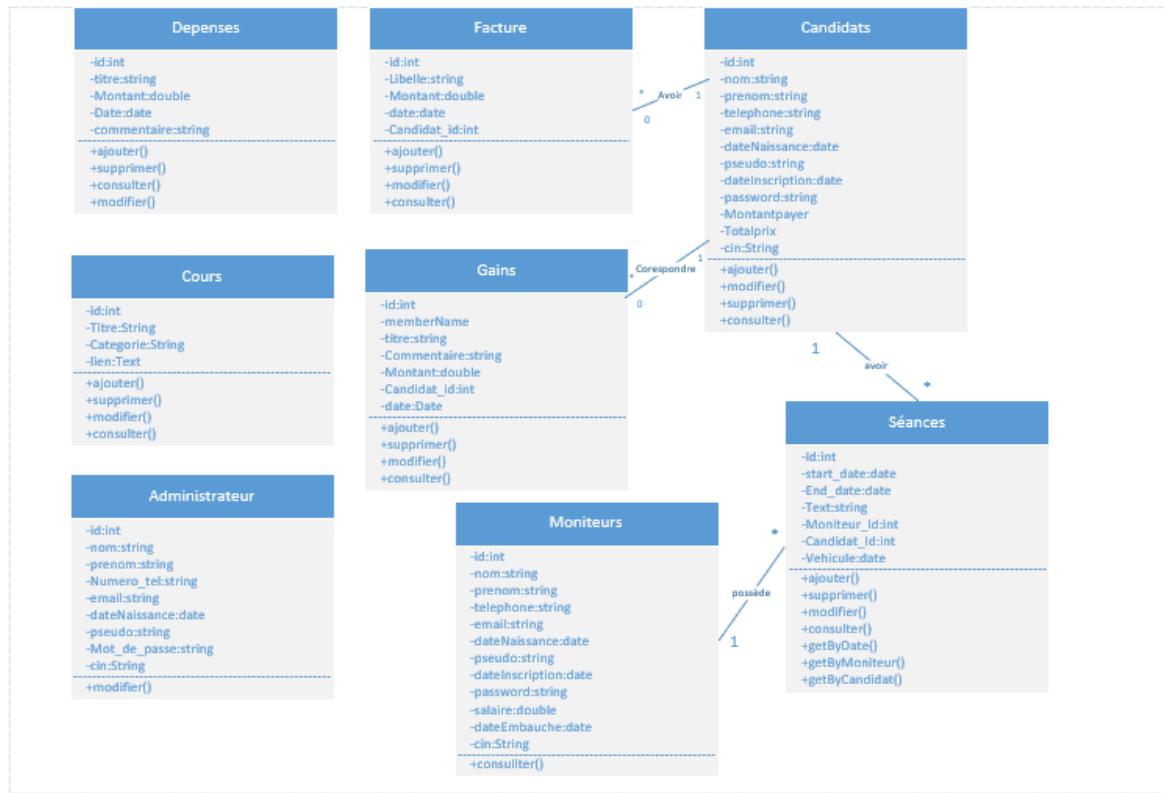


Figure 81 Diagramme de classe globale du deuxième sprint

V. Structure de la base de données

Comme le dernier sprint, dans cette partie nous allons présenter la structure de notre base de données du deuxième sprint.

- **Table « Depenses »**

Champs	Types	Contraintes
Id	Bigint	Primary key
Titre	Character varying	Not null
Montant	Double	Not null
Date	Date	Not null
Commentaire	Character varying	Not null

Tableau 48 Table des dépenses

- **Table « Gains »**

Champs	Types	Contraintes
Id	Bigint	Primary key
Titre	Character varying	Not null
Montant	Double	Not null

Date	Date	Not null
Commentaire	Character varying	Not null
Candidat_id	Bigint	

Tableau 49 Table des gains

- **Table « Cours »**

Champs	Types	Contraintes
Id	Bigint	Primary key
Titre	Character varying	Not null
Categorie	Character varying	Not null
Lien	Text	Not null

Tableau 50 Table des cours

- **Table « Factures »**

Champs	Types	Contraintes
Id	Bigint	Primary key
Libelle	Character varying	Not null
Montant	Double	Not null
Date	Date	Not null
Candidat_id	Bigint	Not null

Tableau 51 Table des factures

- **Table « Candidats »**

Champs	Types	Contraintes
Id	Bigint	Primary key
Nom	Character varying	Not null
Telephone	Character varying	Not null
Prenom	Character varying	Not null
Email	Character varying	Not null
DateNaissance	Date	Not null
Pseudo	Character varying	Not null
Password	Character varying	Not null
Montantpayer	Double	Not null

Dateinscription	Date	Not null
Totalprix	Double	Not null
Cin	Character varying	Not null

Tableau 52 Table des candidats

- **Table « Administrateur »**

Champs	Types	Contraintes
Id	Bigint	Primary key
Numero_tel	Character varying	Not null
Cin	Character varying	Not null
Nom	Character varying	Not null
Prenom	Character varying	Not null
Email	Character varying	Not null
Date_naissance	Date	Not null
Pseudo	Character varying	Not null
Mot_de_passe	Character varying	Not null
Photo	Character varying	---

Tableau 53 Table des administrateurs

VI. Implémentation

6.1 Diagramme de composants

- **Diagramme de composant de la fonctionnalité « S'authentification »**

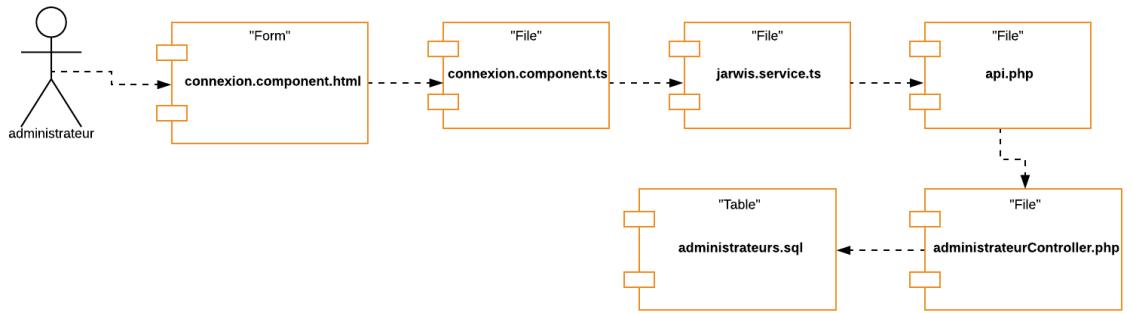


Figure 82 Diagramme de composants de la fonctionnalité :
« S'authentification » version web

➤ **Interface
S'authentifier - Web »**

du cas «

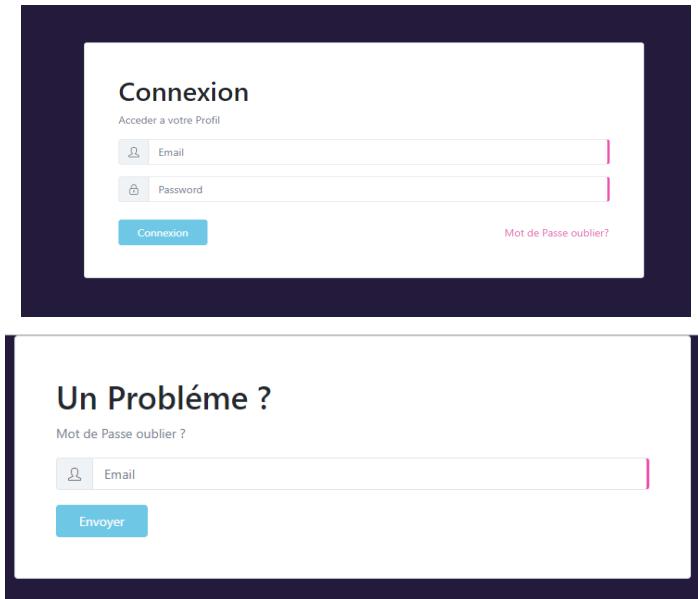


Figure 83 Interfaces web du cas "S'authentifier"

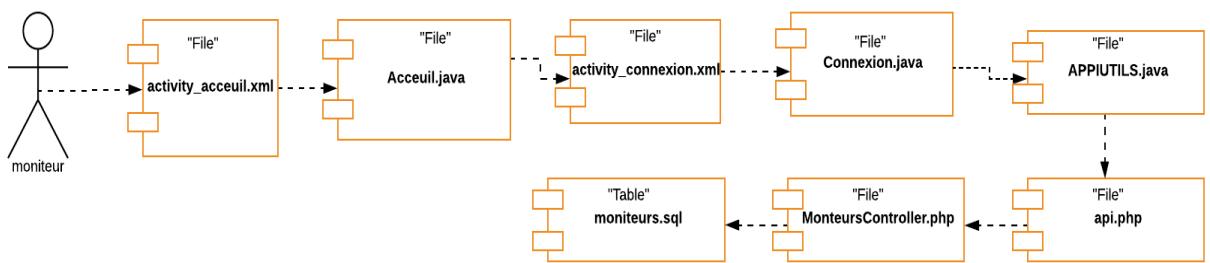


Figure 84 Diagramme de composants de la fonctionnalité : « S'authentification » version mobile

➤ Interface du cas « S'authentifier - Mobile »

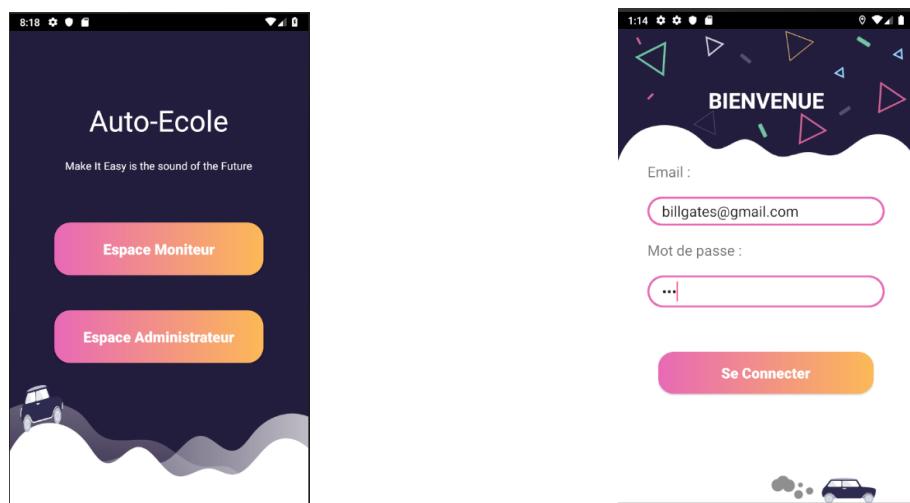


Figure 85 Interfaces mobiles du cas "S'authentifier"

- Diagramme de composant de la fonctionnalité « Gestion Comptabilité »

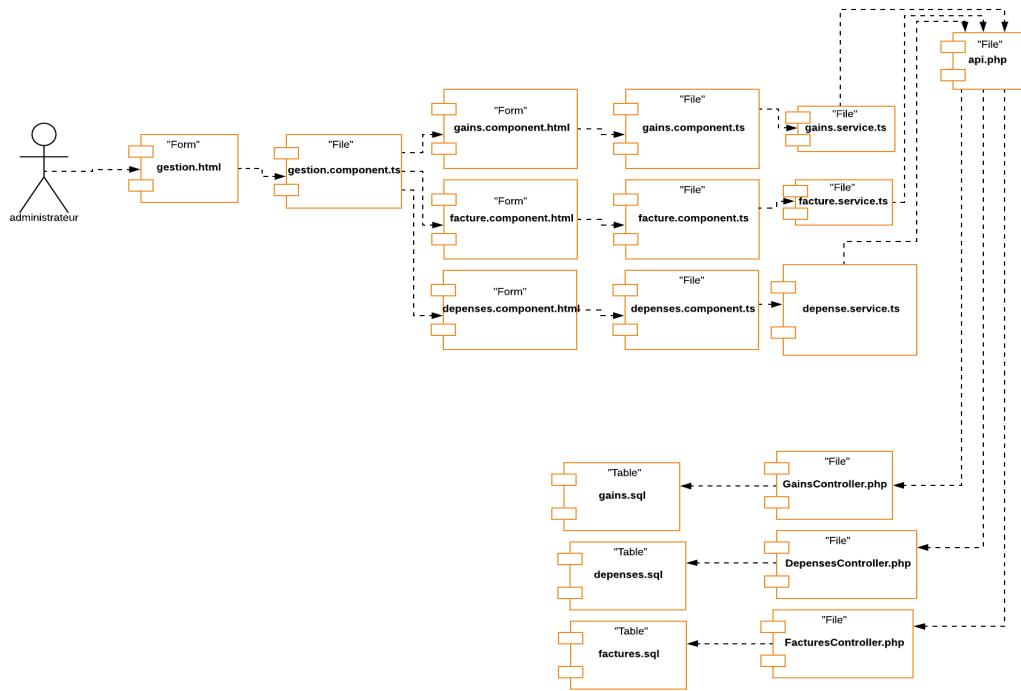


Figure 87 Diagramme de composants de la fonctionnalité : Gestion comptabilité Partie 1

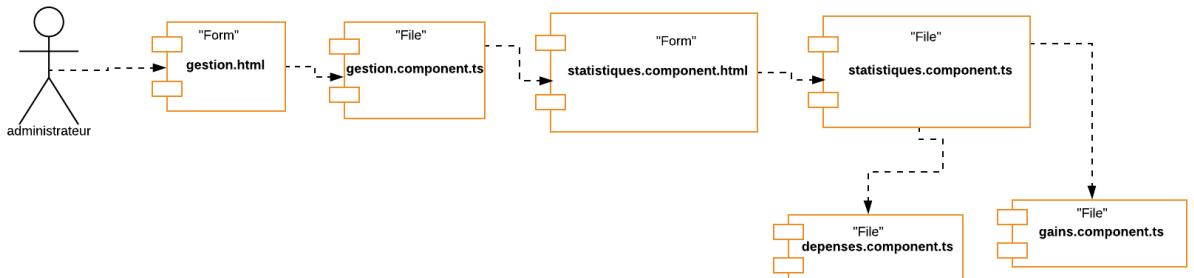


Figure 86 Diagramme de composants de la fonctionnalité : « Gestion comptabilité » Partie 2

➤ Interface du cas « Gérer comptabilité »

Interface utilisateur de l'application « Auto-Ecole » pour la gestion des gains.

Le menu principal inclut les sections ACCUEIL, GESTION (Calendrier, Véhicule, Moniteur, Candidat, Cours), et COMPTABILITÉ (Factures, Dépenses). L'onglet COMPTABILITÉ est actif, montrant la liste des gains.

La liste des gains affiche les colonnes : Titre, Montant, Date, Commentaire, Candidat, Modifier, Supprimer. Un gain est listé :

Titre	Montant	Date	Commentaire	Candidat	Modifier	Supprimer
libellé	250	2019-05-08	rien	1	<button>Modifier</button>	<button>Supprimer</button>

Un message de confirmation apparaît dans le bas de l'écran : « Gain ajouté avec succès ». Des boutons de navigation sont également visibles au bas de l'écran.

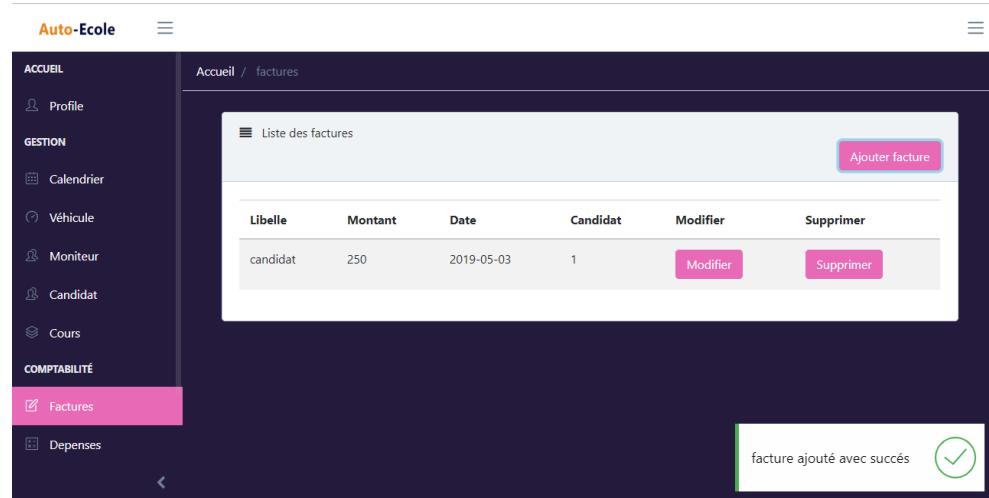


Figure 88 Interfaces de la fonctionnalité « Gestion comptabilité »

- **Diagramme de composant de la fonctionnalité « Gestion cours »**

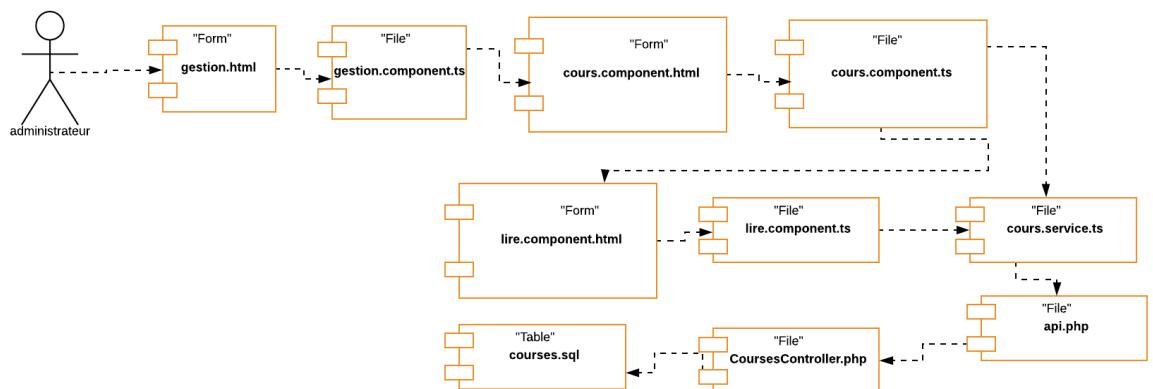


Figure 89 Diagramme de composants de la fonctionnalité : « Gestion cours »

➤ **Interface de la fonctionnalité « Gestion cours »**

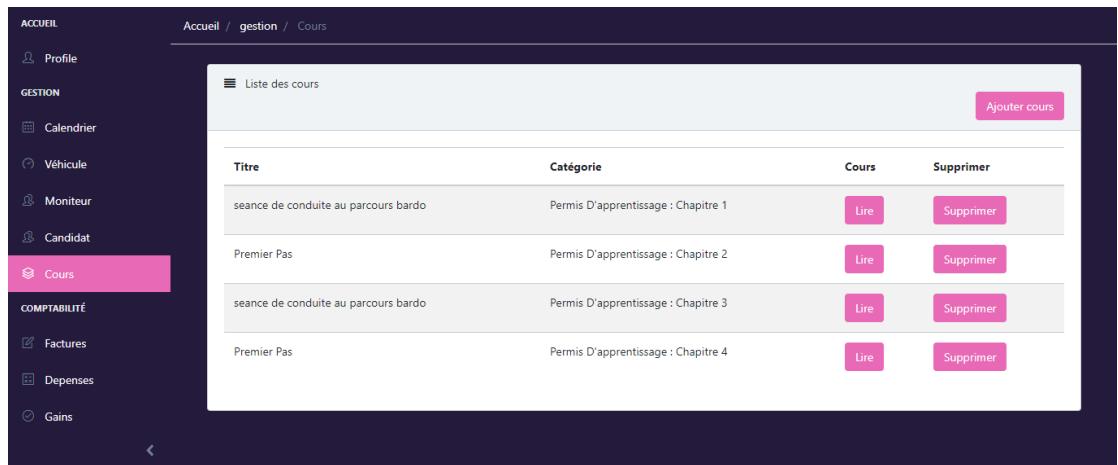


Figure 90 Interface de la fonctionnalité "Gestion cours"

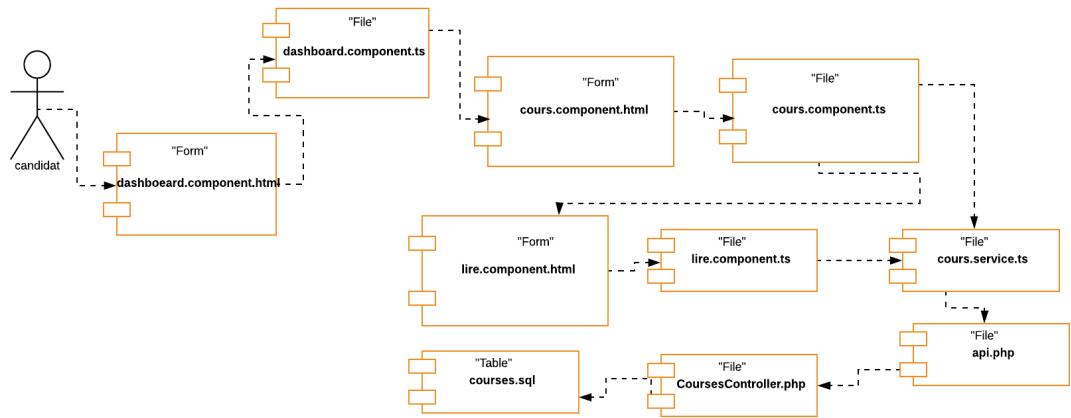


Figure 91 Diagramme de composants de la fonctionnalité : « Consulter cours »

- Diagramme de composant de la fonctionnalité « Consulter planning »**

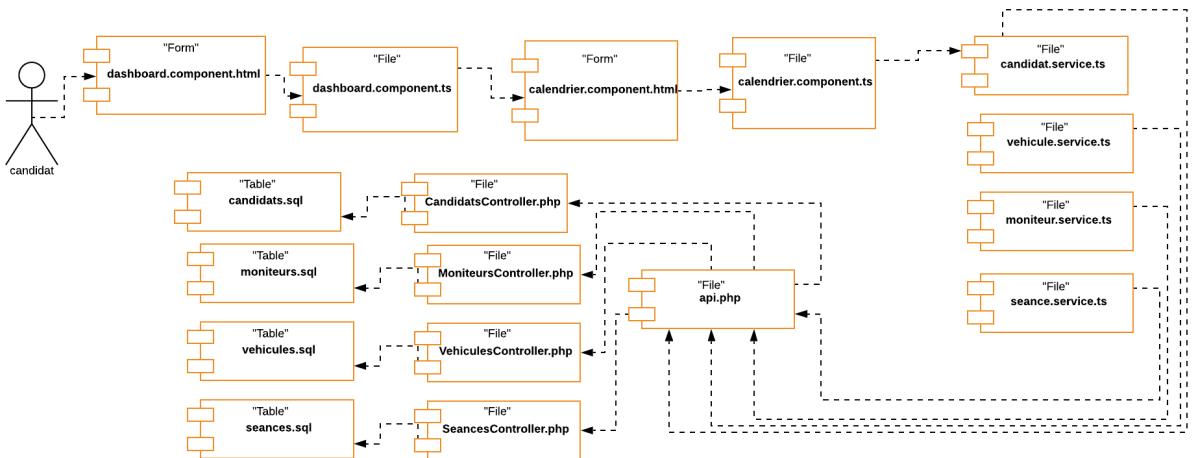


Figure 92 Diagramme de composants de la fonctionnalité : « Consulter planning » Partie web

➤ Interface du cas « Consulter planning – Web »

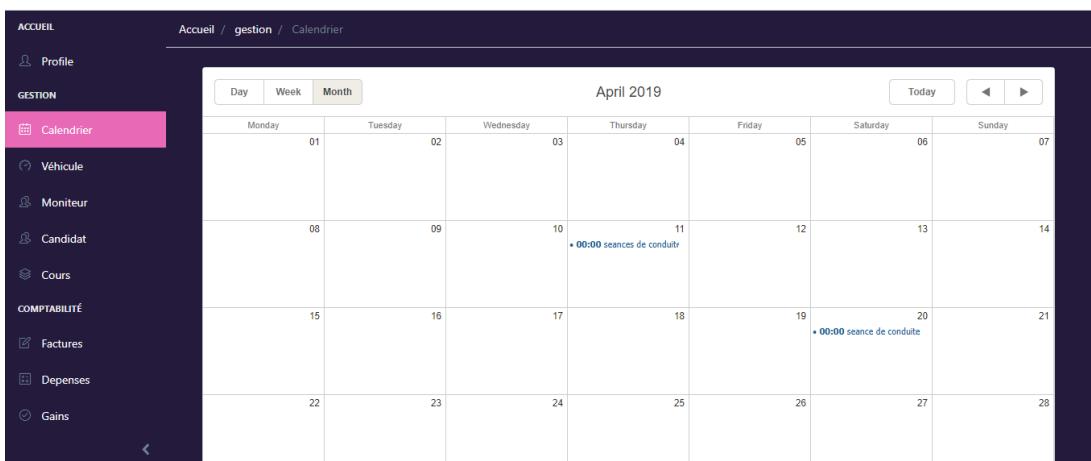


Figure 93 Interface du cas "Consulter planning"

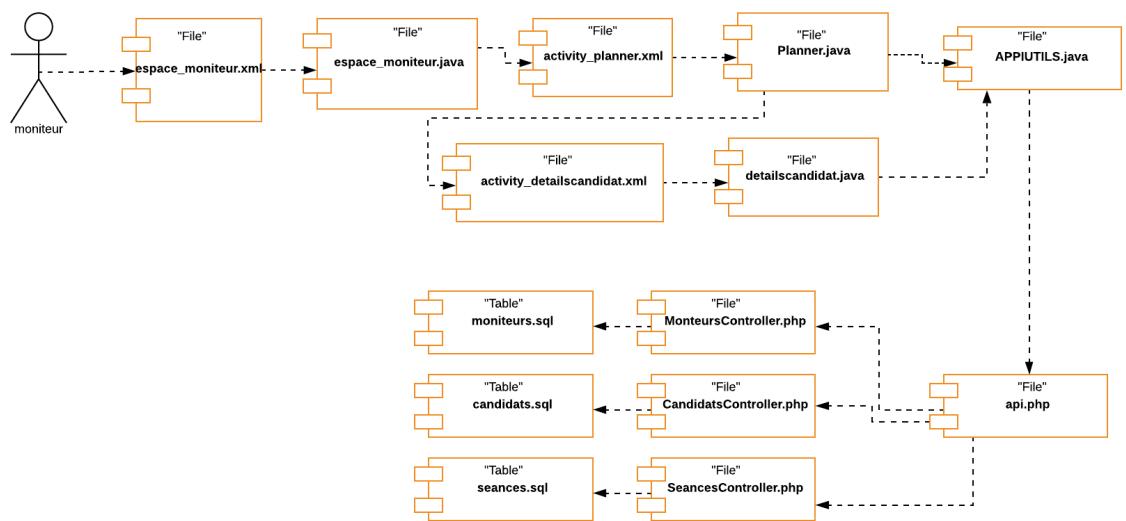


Figure 94 Diagramme de composants de la fonctionnalité : « Consulter planning »
Mobile Partie 1

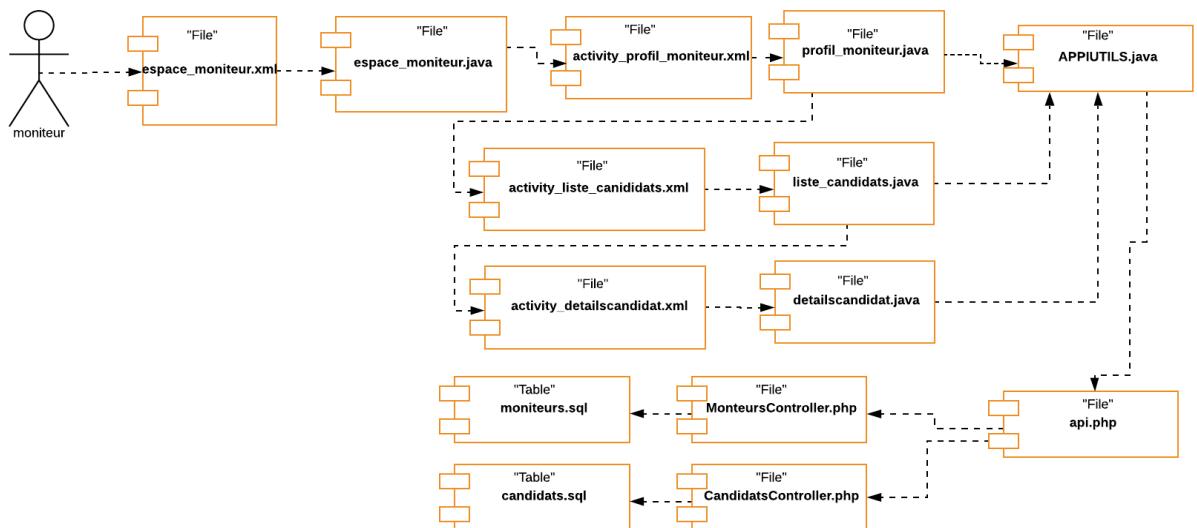


Figure 95 Diagramme de composants de la fonctionnalité : « Consulter planning » mobile partie 2

➤ Interface du cas « Consulter planning – mobile »



Figure 96 Interface consulter planning -mobile

- **Diagramme de composant de la fonctionnalité « Gérer compte »**

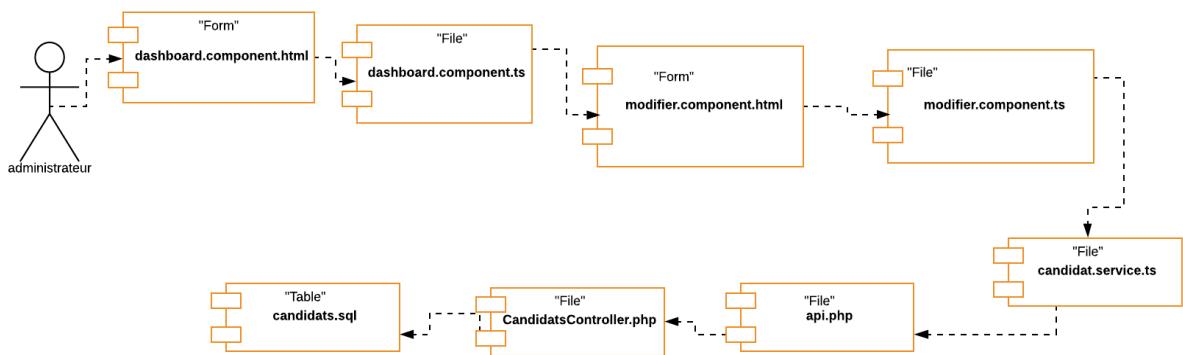


Figure 97 Diagramme de composants de la fonctionnalité : « Gérer compte »

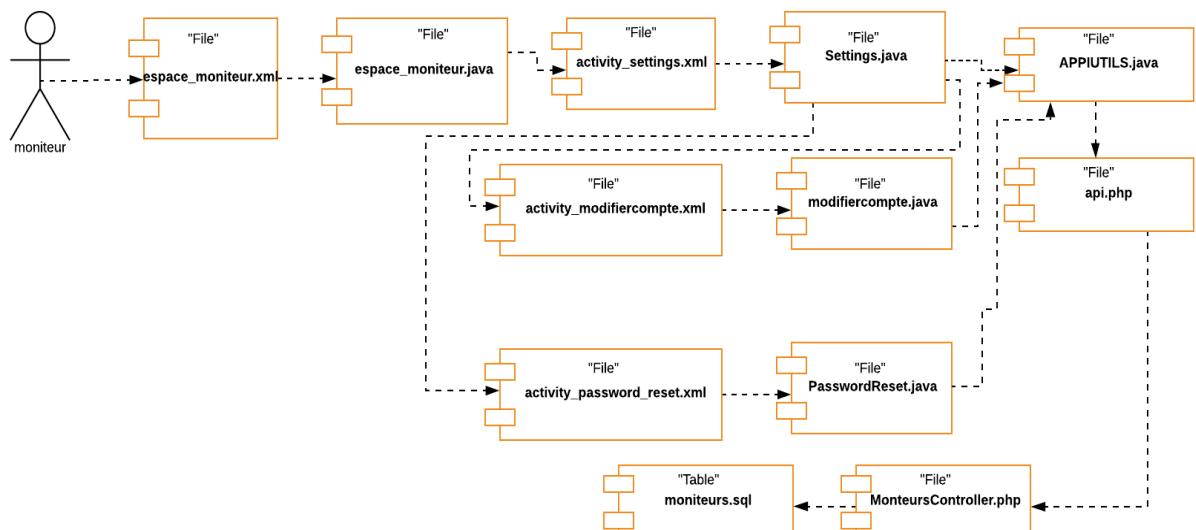


Figure 98 Diagramme de composants de la fonctionnalité : « Gérer compte » web

➤ Interface du cas « Gérer compte - Web »

The screenshot shows a web application interface for managing a user profile. On the left, there is a sidebar with navigation links: ACCUEIL, Profile (highlighted in pink), GESTION (Calendrier, Véhicule, Moniteur, Candidat, Cours), COMPTABILITÉ (Factures, Dépenses, Gains). The main content area is titled 'Profile' and contains fields for Nom (Ghozzi), Prénom (Ayoub), Email (ayoubghozzi@gmail.com), Téléphone (26920953), Date de naissance (jj/mm/aaaa), and Mot de passe (*****). A back arrow is visible at the bottom left.

Figure 99 Interface du cas « Gérer compte » web

➤ Interface du cas « Gérer compte - Mobile »

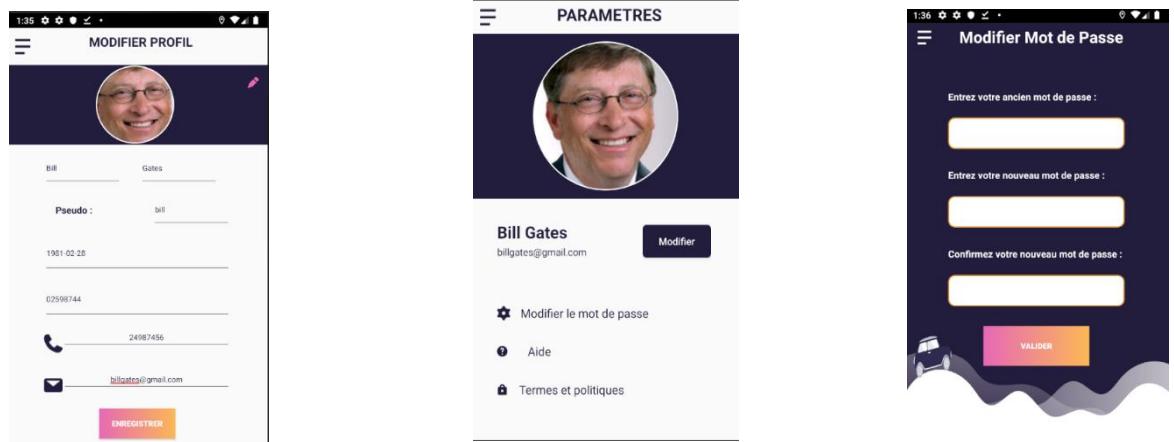


Figure 100 Interface du cas « Gérer compte » -mobile

• Diagramme de composant de la fonctionnalité « Passer les tests en ligne »

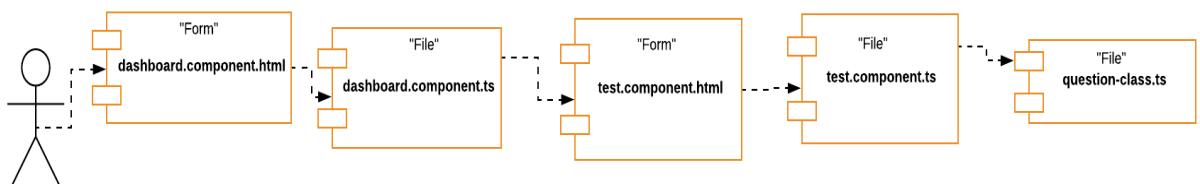


Figure 101 Diagramme de composants de la fonctionnalité : « Passer les tests en ligne »

➤ Interface du cas « Passer test en ligne »

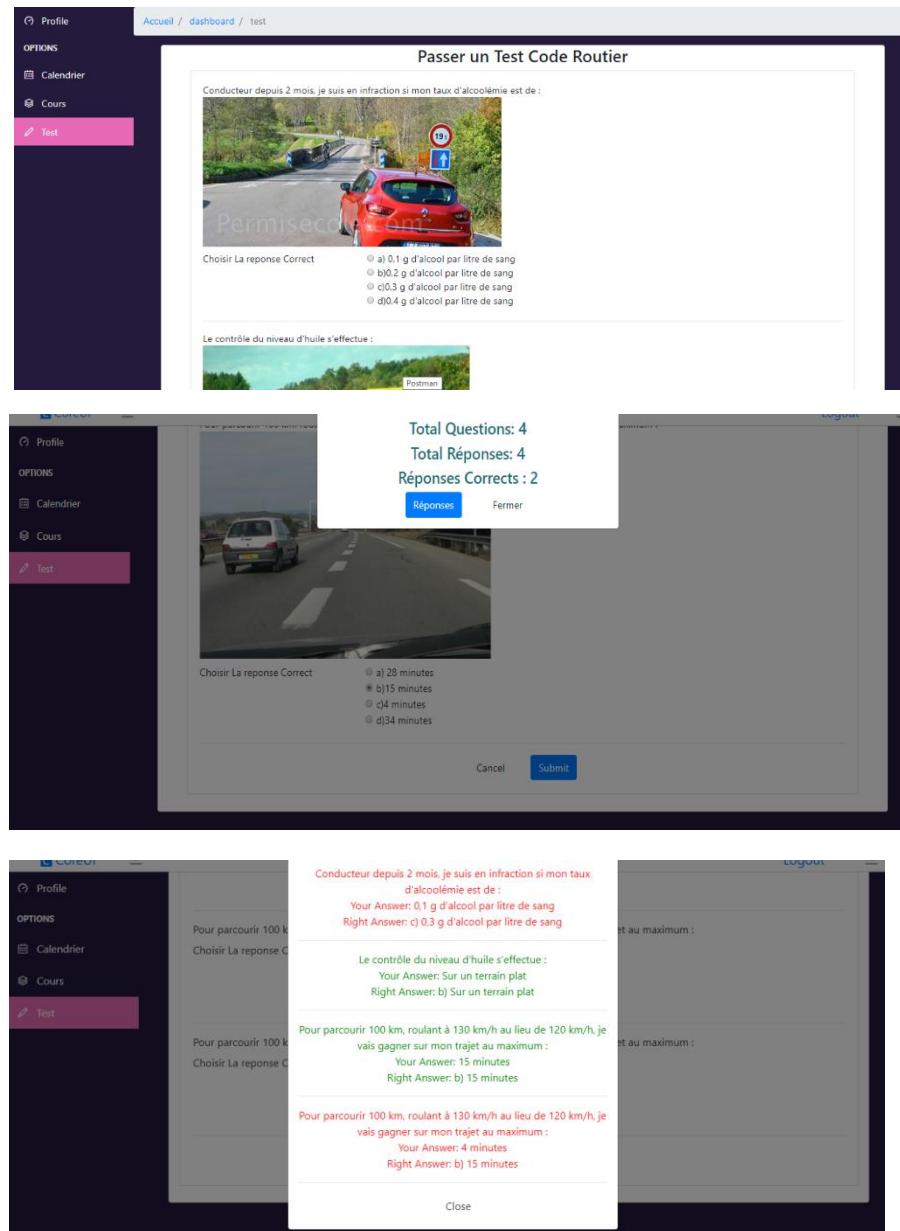


Figure 102 Interface du cas " Passer test en ligne "

VII. Les web services

Comme on l'a déjà mentionné on a utilisé une architecture orientée service. Notre tableau des web service se trouve dans l'annexe 2 (page 111.).

VIII. Test

8.1 Les Tests des web service

Voici quelques captures de nos tests.

➤ Test du web service consulter les dépenses

```

1 [
2   {
3     "id": 1,
4     "titre": "Essance",
5     "montant": "25000",
6     "date": "2019-04-09",
7     "commentaire": "visite de voiture",
8     "created_at": "2019-04-29 15:14:39",
9     "updated_at": "2019-04-29 15:14:39"
10    }
11 ]

```

Figure 103 Collection "dépense"

➤ Test du web service modifier compte

The screenshot shows the Postman interface with a POST request to `http://127.0.0.1:8000/api/moniteur/1/updateprofile?cin=...`. The request body is a JSON object:

```

1 {
2   "result": true,
3   "message": "profile update !"
4 }

```

Figure 104 Modifier compte -Postman

8.2 Les tests unitaires

8.2.1 Le test unitaire du cas « Ajouter une dépense »

➤ Raisonnement

Afin de pouvoir tester l'ajout d'une dépense on a suivi la démarche suivante :

- Création un objet JSON contenant comme paramètre la méthode appropriée, la route du contrôleur ainsi que l'objet JSON de la dépense à ajouter.
- Le résultat attendu par la méthode d'ajout est donné par la méthode « `assertJson()` ».

La réussite du test est représentée par l'équivalence du résultat reçu au résultat attendu. Dans ce cas la méthode « `assertStatus()` » prend la valeur ‘true’

➤ **Code source de la méthode de test du cas « Ajouter une dépense »**

```
/*
public function testAjout_Depense_is_true()
{
    $response = $this->json('POST', 'api/admin/comptabilite/depenses/store',
    [
        'id'=> '1' ,
        'titre' => 'depense du mois de Juin',
        'montant'=> '23000',
        'date'=> '30/06/2019',
        'commentaire'=> 'les factures ne sont pas incluses dans le montant'
    ]);

    $response
        ->assertStatus(200)
        ->assertJson([
            "result"=>true,
            "message"=> "votre ajout a été effectuée avec succès"
        ]);
}
```

Figure 105 méthode de test d'ajouter une dépense

➤ **Capture d'écran représentant le résultat du test du cas d'utilisation « Ajouter dépense »**

```
PS C:\Users\ASUS\Desktop\Auto_backend> vendor\bin\phpunit --filter testAjout_Depense_is_true
PHPUnit 7.5.10 by Sebastian Bergmann and contributors.

.

1 / 1 (100%)

Time: 20.61 seconds, Memory: 18.00 MB
OK (1 test, 2 assertions)
```

Figure 106 Résultat du test ajout dépense

➤ **Capture d'écran représentant le résultat du test du cas d'utilisation « Ajouter dépense » en violant certaines contraintes.**

```
PS C:\Users\ASUS\Desktop\Auto_backend> vendor\bin\phpunit --filter testAjout_Depense_is_true
PHPUnit 7.5.10 by Sebastian Bergmann and contributors.

F
1 / 1 (100%)

Time: 8.4 seconds, Memory: 18.00 MB
There was 1 failure:
```

Figure 107 Résultat d'échec lors du test ajout dépense

IX. Conclusion

Dans ce chapitre, nous avons fait la réalisation du dernier sprint, durant lequel nous avons effectué l'analyse, la conception, l'implémentation et les tests des fonctionnalités suivantes : la s'authentification, la gestion des cours, la gestion des candidats et la gestion des statistiques, la consultation des plannings, passer les tests en ligne et enfin la gestion des comptes.

Conclusion générale et perspectives

Ce rapport présente notre travail, réalisé au sein de la société FORMALAB, dans le cadre de notre projet de fin d'études pour l'obtention d'une licence fondamentale en informatique de gestion à l'institut Supérieur de Gestion de Tunis.

L'objet de ce projet de fin d'études était la conception et le développement d'une application web et mobile de gestion d'une auto-école qui sert à informatiser le système de gestion des auto-écoles et à automatiser leurs différentes activités. Notre projet consiste aussi à la création d'un site vitrine qui présentera notre produit. Pour ce faire, nous avons suivi la méthodologie Scrum. Dans le cas de ce projet, on a effectué le travail demandé en 2 sprints.

Tout au long de ce stage nous sommes parvenues à développer la plupart des fonctionnalités du système. Cependant, notre produit englobe les modules de gestion des moniteurs, des candidats, de la comptabilité et des statistiques, du planning et des rapports des candidats. Il offre aussi la possibilité de gérer les cours et les tests des candidats en ligne ainsi que la géolocalisation instantanée des véhicules.

L'application web a été développée en Angular et la partie mobile a été réalisée sous Android. On a adopté une architecture orientée service pour nos deux applications en utilisant Laravel.

Malgré les différents obstacles tels que les nouveaux langages non manipulés durant notre cursus universitaire, on a pu apprendre à gérer les nouveaux outils en bâtiissant des nouvelles connaissances en matière de méthodologie de travail, de programmation web, de programmation mobile, d'utilisation des services WEB. Ce stage a contribué à améliorer notre sens d'organisation et de planification et nous a offert une expérience de collaboration suivant la méthodologie de travail SCRUM.

A l'issue de ce projet, nous pouvons proposer les perspectives d'évolution et d'amélioration suivantes :

- Un système de notification, qui servira à notifier les différents acteurs impliqués par leurs séances.
- Un chatbot pour faciliter l'obtention de l'information.
- Un système d'apprentissage automatisé pour permettre de rendre les tests des candidats plus stratégiques.
- Une amélioration au niveau de l'expérience utilisateur (UX), ainsi qu'au niveau du design des interfaces de la plateformes web.

Bibliographie

- [1] C. Daudet, «L'atelier,» 1 décembre 2015. [En ligne]. Available: <https://atelier.bnpparibas/life-work/article/transformation-digitale-entreprises-ont-veritablement-saute>. [Accès le 13 mai 2019].
- [2] D. Galiana, «Planzone,» 20 Juillet 2017. [En ligne]. Available: https://www.planzone.fr/blog/quest-ce-que-la-methodologie-scrum?fbclid=IwAR0qototasEnpfjGl6A1kfeC6gnkQBcMqiVgXPT3f_YdXW5ZqV9oDUKMIg8. [Accès le 13 février 2019].
- [3] «Wikipedia,» 21 mars 2019. [En ligne]. Available: <https://fr.wikipedia.org/wiki/Laravel>. [Accès le 13 mai 2019].
- [4] «Wikipedia,» 9 mai 2019. [En ligne]. Available: <https://fr.wikipedia.org/wiki/Angular>. [Accès le 13 mai 2019].
- [5] «Wikipedia,» 27 avril 2019. [En ligne]. Available: https://fr.wikipedia.org/wiki/Android_Studio. [Accès le 13 mai 2019].
- [6] «Wikipédia,» 10 mai 2019. [En ligne]. Available: https://en.wikipedia.org/wiki/Adobe_XD. [Accès le 11 mai 2019].
- [7] «Wikipedia,» 21 avril 2019. [En ligne]. Available: <https://fr.wikipedia.org/wiki/PostgreSQL>. [Accès le 12 mai 2019].
- [8] «Wikipedia,» 2 novembre 2017. [En ligne]. Available: <https://fr.wikipedia.org/wiki/Firebase>. [Accès le 13 mai 2019].
- [9] «Wikipedia,» 30 avril 2019. [En ligne]. Available: https://fr.wikipedia.org/wiki/Microsoft_Word. [Accès le 13 mai 2019].
- [10] «Wikipedia,» 12 février 2018. [En ligne]. Available: <https://fr.wikipedia.org/wiki/Trello>. [Accès le 13 mai 2019].
- [11] «Wikipedia,» 17 avril 2019. [En ligne]. Available: [https://fr.wikipedia.org/wiki/Slack_\(plateforme\)](https://fr.wikipedia.org/wiki/Slack_(plateforme)). [Accès le 13 mai 2019].
- [12] «Wikipedia,» 12 mai 2019. [En ligne]. Available: https://fr.wikipedia.org/wiki/Microsoft_Visio. [Accès le 13 mai 2019].
- [13] D. Martin, «Ippon,» 30 janvier 2015. [En ligne]. Available: <https://blog.ippon.fr/2015/01/30/postman-le-client-pour-api-web-qui-vous-fera-oublier-les-autres/>. [Accès le 13 mai 2019].
- [14] C. Regnault, «Axopen,» 24 février 2017. [En ligne]. Available: <https://blog.axopen.com/2017/02/gitlab-cest-quoi/>. [Accès le 13 mai 2019].

- [15] «Wikipedia,» 8 mai 2019. [En ligne]. Available: [https://fr.wikipedia.org/wiki/Bootstrap_\(framework\)](https://fr.wikipedia.org/wiki/Bootstrap_(framework)). [Accès le 13 mai 2019].
- [16] [En ligne]. Available: <http://www.gantt.com/>.

Annexe 1

Diagramme de Gantt

« Le diagramme de Gantt, couramment utilisé en gestion de projet, est l'un des outils les plus efficaces pour représenter visuellement l'état d'avancement des différentes activités (tâches) qui constituent un projet » [16]

Le diagramme de GANTT nous permet de consulter l'avancement des différentes tâches correspondantes à notre projet.

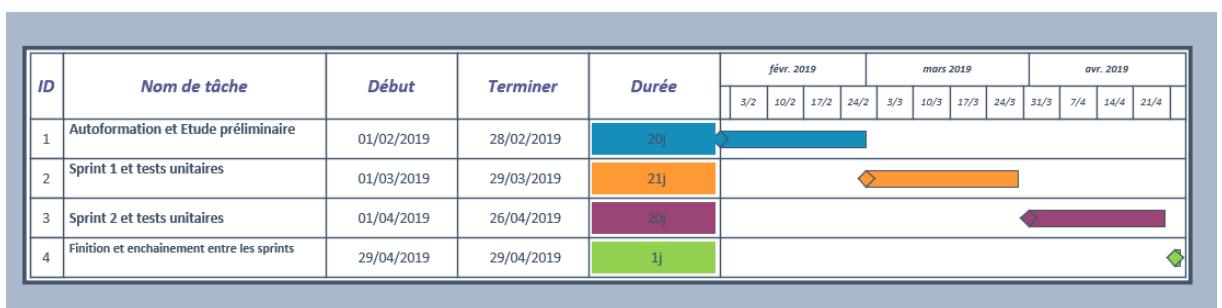


Figure 109 Diagramme de Gantt

Annexe 2

Voici ci-dessus notre tableau de web service utilisé tout au long de notre projet qui présente l'acteur concerné, l'URL, la méthode utilisée ainsi qu'une description du web service.

Acteur	URL	Méthode supporté	Request	Response	Description
Candidat, Admin, Moniteur	/login	POST	Json object de Candidat	Access_token : Token Expires_in :3600 Token_type : « bearer » User : user logged in data	Vérifie si les données sont compatibles avec celles sauvegardées dans la base de données si les données sont valides il renvoie un Token avec les données de l'utilisateur connecté.
Candidat, Admin, Moniteur	/logout	POST	---	Message : ‘Successful ly logged out’	Déconnecte l'utilisateur en supprimant le access_token du local storage
Candidat, Admin, Moniteur	/sendPas swordRe setLink	POST	Json object contenant email	Email valide :’email de réinitialisation password envoyer’ Email non valide :’email erroné’	Si l'email est valide envoie un email de réinitialisation de passeword. Sinon renvoie un message d'erreur.
Candidat, Admin,	/resetPas word	POST	Json object contenant	Data :’password successfully changed’	Modifie les données de la base par les

Moniteur			t email, password, confirmer mot de passe		données du formulaire et renvoie un message de confirmation
Candidat, Admin, Moniteur	/user	GET	---	User, CIN, dateInscription, dateNaissance, email, id, montantPayer, nom, prenom, pseudo, telephone, totalPrix	Renvoie les données de l'utilisateur connecté
Candidat	/candidat/{idcandidate}/Agenda	GET	Idcandidat : int	Json object séance	Retourne les séances du candidat ayant l'identifiant passé en paramètre
Administrateur	/admin/moniteurs	GET	---	JsonArray des moniteurs	Retourne tous les moniteurs
Administrateur	/admin/moniteurs/store	POST	Json Object de Moniteur	Json Objet : {"result" : true;} => si la sauvegarde est effectuée {"result" : false} Message : message d'erreur} => sinon	Sauvegarder les informations d'un moniteur dans la base de données
Administrateur	/admin/moniteurs/update	GET	Json Object Moniteur	Json Objet : {"result" : true;} => si la modification est effectuée	Modification des données du moniteur

				{“result” : false Message : message d’erreur} => sinon	
Administrateur	/admin/moniteurs /delete/{id}	DELETE	Id :int	Json Objet : {“result” : true ;} => si la suppression est effectuée {“result” : false Message : message d’erreur} => sinon	Supprimer un moniteur de la base de données
Administrateur	/moniteur/select/{id}	GET	Id :int	Json Array de moniteur avec l’id correspondant	Affiche les informations du moniteur précis
Moniteur	/moniteur/Agenda/{idmoniteur}	GET	Idmoniteur :int	Json Array de séance	Retourne les informations des séances concernant le moniteur précis
Moniteur	/moniteur/Agenda/{idseance}/candidats/{idcandidat}	GET	Idmoniteur :int Idcandidat :int	Json Object candidat	Retourne les informations du candidat précis concernant le moniteur ayant l’identifiant précis
Moniteur	/moniteur/{idmoniteur}/rapports	GET	Idmoniteur :int	JsonArray de rapport	Retourne les informations des rapports qui correspondent au moniteur ayant l’identifiant

					passé en paramètre
Moniteur	/moniteur/{idmoniteur}/rapports/candidats/{idcandidat}/	GET	Idmoniteur :int Idcandidat :int	JsonArray rapport	Retourne les informations des rapports du candidats et du moniteurs précis
Moniteur	/moniteur/{idmoniteur}/candidats/{idcandidat}/resultat/{idresultat}/update	UPDATE	Idmoniteur :int Idcandidat :int	{"'result'" : true ;} => si la modification est effectuée {"'result'" : false} Message : message d'erreur} => sinon	Modifie les données du rapport dans la base de données
Moniteur	/moniteur/{idmoniteur}/candidats/{idcandidat}/seance/{idseance}/resultat/store	POST	Idmoniteur :int Idcandidat :int Idrapport : int	{"'result'" : true ;} => si la sauvegarde est effectuée {"'result'" : false} Message : message d'erreur} => sinon	Enregistre les données du rapport précis dans la base de données
Moniteur	/moniteur/{idmoniteur}/candidats/{idcandidat}/rapport/{id}/delete	DELETE	Idmoniteur :int Idcandidat :int Idrapport : int	Json Objet : {"'result'" : true ;} => si la suppression est effectuée {"'result'" : false} Message : message d'erreur} => sinon	Supprimer le rapport de la base de données

Administrateur	/admin/candidats	GET	---	JsonArray de candidat	Retourner les informations des candidats
Administrateur	/admin/candidats/store	POST	Json Object candidat	Json Object : {'result' : true;} => si la sauvegarde est effectuée {'result' : false} Message : message d'erreur} => sinon	Sauvegarde les données d'un candidat dans la base de données
Administrateur	/admin/candidats/select/{id}	GET	Id :int	JsonArray de candidat avec id précis	Affiche les informations d'un candidat
Administrateur	/admin/candidats/update	PATCH	Json Object candidat	Json Object : {'result' : true;} => si la modification est effectuée {'result' : false} Message : message d'erreur} => sinon	Mettre à jour les données d'un candidat
Administrateur	/admin/candidats/delete/{id}	DELETE	Id :int	Json Object : {'result' : true;} => si la suppression est effectuée {'result' : false}	Supprimer un candidat de la base de données

				Message : message d'erreur} => sinon	
Administrateur	/admin/update/{id}	UPDATE	Json Object administrateur	Json Object : {“result” : true ;} => si la modification est effectuée {“result” : false Message : message d'erreur} => sinon	Mettre à jour les données de l'administrateur
Administrateur	/admin/vehicules	GET	---	JSONArray de véhicule	Retourne la liste des véhicules
Administrateur	/admin/vehicules/update/{id}	UPDATE	Json Object véhicule	Json Object : {“result” : true ;} => si la modification est effectuée {“result” : false Message : message d'erreur} => sinon	Mettre à jour les données d'un véhicule
Administrateur		DELETE	Matricule : String	Json Object : {“result” : true ;} => si la suppression est effectuée {“result” : false	Supprimer un véhicule de la base de données

				Message : message d'erreur} => sinon	
Candidat	/candidat/cours	GET	---	JsonArray de cours	Retourne tous les cours
Administrateur	/admin/Agenda	GET	---	JsonArray de séance	Retourne toutes les séances de la base de données
Administrateur	/admin/Agenda/delete/{id}	DELETE	Id :int	Json Object : {"result" : true;} => si la suppression est effectuée {"result" : false} Message : message d'erreur} => sinon	Supprime une séance de la base de données
Administrateur	/admin/Agenda/update/{id}	PATCH	Json Object séance	Json Object : {"result" : true;} => si la modification est effectuée {"result" : false} Message : message d'erreur} => sinon	Modifie une séance de la base de données
Administrateur	/admin/Agenda/store	POST	Json Object séance	Json Object : {"result" : true;} => si la sauvegarde est effectuée	Sauvegarder une séance dans la base de données

				{“result” : false Message : message d’erreur} => sinon	
Administrateur	/admin/cours	GET	---	JsonArray de cours	Retourne tous les cours de la base de données
Administrateur	/admin/cours/selection/{id}	GET	Id :int	JsonArray de cours	Retourne un cours précis
Administrateur	/admin/cours/update/{id}	GET	Json Object cours	Json Object : {“result” : true ;} => si la modification est effectuée {“result” : false Message : message d’erreur} => sinon	Modifier un cours précis de la de données
Administrateur	/admin/cours/delete/{id}	DELETE	Id :int	Json Object : {“result” : true ;} => si la suppression est effectuée {“result” : false Message : message d’erreur} => sinon	Supprimer un cours de la base de données
Administrateur	/admin/comptabilite/gains	GET	---	JsonArray de gains	Retourne tous les gains enregistrés dans

					la base de données
Administrateur	/admin/comptabilité/dépenses	GET	---	JsonArray de dépenses	Retourne toutes les dépenses enregistrées dans la base de données
Administrateur	/admin/comptabilité/gains/store	POST	Json Object de gain	Json Object : {“result” : true;} => si la sauvegarde est effectuée {“result” : false} Message : message d’erreur} => sinon	Enregistre un gain dans la base de données
Administrateur	/admin/comptabilité/dépenses/store	POST	Json Object de dépense	Json Object : {“result” : true;} => si la sauvegarde est effectuée {“result” : false} Message : message d’erreur} => sinon	Enregistre une dépense dans la base de données
Administrateur	/admin/comptabilité/factures	GET	---	JsonArray de facture	Retourne toutes les factures enregistrées dans la base de données
Administrateur	/admin/comptabilité/factures/{id}	GET	Id :int	JsonArray de facture	Retourne une facture précise

Administrateur	/admin/comptabilité/factures/store	POST	Json Object facture	Json Object : {'result' : true;} => si la sauvegarde est effectuée {'result' : false} Message : message d'erreur} => sinon	Enregistre une facture dans la base de données
Visiteur	/contact	POST	Json Object Type contact	Json Object : {'result' : true;} => si la sauvegarde est effectuée {'result' : false} Message : message d'erreur} => sinon	Enregistre les données fournies par le visiteur dans la base de données

Tableau 54 tableau des WEB SERVICE