



MINISTÈRE  
DE L'ÉDUCATION NATIONALE,  
DE L'ENSEIGNEMENT SUPÉRIEUR  
ET DE LA RECHERCHE

*Ministère de l'Enseignement Supérieur  
et de la Recherche Scientifique*



*INSTITUT SUPERIEUR M2I Formation*

**Projet de fin d'étude pour l'obtention  
de la licence en  
Concepteur Développeur Informatique**

---

# **Réalisation d'une application mobile de géolocalisation de TAXI et site internet sous Android et PHP5.6**

---

**Réalisé par :**

**M Lagrini Rachid 13 Rue Pierre Cot 92160 Antony**

**Encadré par :**

**Encadreur professionnel :  
M Sébastien Morlan**

**Encadreur pédagogique :  
M Pascal Buguet**



MINISTÈRE  
DE L'ÉDUCATION NATIONALE,  
DE L'ENSEIGNEMENT SUPÉRIEUR  
ET DE LA RECHERCHE

**Ministère de l'Enseignement Supérieur  
et de la Recherche Scientifique**



**INSTITUT SUPERIEUR M2I Formation**

# *Remerciements*

*Mes remerciements les plus sincères à tous ceux et celles qui  
m'ont apporté leur aide, leurs encouragements et leur soutien pour me  
permettre de mener à bien ce projet.*

*Mes remerciements s'adressent à mon encadreur,*

*M. Pascal Buguet, pour son assistance, son soutien, sa disponibilité et ses précieux  
conseils durant la période de ce stage.*

*Mes remerciements à mon encadreur à M. Sébastien Morlan,  
pour son entière disponibilité à me fournir son assistance et sa confiance la plus attentionnée.*

*Mes remerciements à la société TAXI et en particulier à son  
directeur M. Karim Belhadri et à tous ceux et celles qui m'ont apporté leur contribution.*

## Table des matières

Introduction Générale.....	11
<i>CHAPITRE I : ETUDE PREALABLE</i> .....	12
Introduction : .....	12
1. Présentation du cadre de stage : .....	12
1.1. Présentation de l'organisme d'accueil : .....	13
1.2. Présentation de la société Kb_Driver : .....	13
1.3. Cadre du sujet.....	13
1.4. Impact du sujet.....	14
2. Etude de l'existant : .....	14
2.1. Analyse de l'application « 'X' » : .....	14
2.2.1. Présentation de l'application « 'X' » : .....	14
2.2.2. Les fonctionnalités: .....	15
2.2.3. Critique de l'application « 'X' » : .....	16
2.3. Analyse de l'application « GoCatch » : .....	17
2.3.1. Présentation de l'application «GoCatch» : .....	17
2.3.2. Les fonctionnalités: .....	17
2.3.3. Critique de l'application « GoCatch » : .....	19
2.4. Analyse de l'application « Y » : .....	20
2.4.1. Présentation de l'application «Y» : .....	20
2.4.2. Les fonctionnelles : .....	21
2.4.3. Critique de l'application «Y» : .....	22
2.5. Synthèse : .....	23
2.6. Les Solution proposées : .....	24
3. Langage et méthodologie de conception .....	24
4. Pourquoi Scrum .....	25
5. Pilotage du projet avec Scrum : .....	27
5.1. Planification d'un projet par Scrum : .....	27
5.2. Equipe et rôles : .....	27
6. Généralités sur la localisation .....	28
6.1. Historique et évolution des besoins de localisation .....	28
6.2. Le principe de géolocalisation.....	29
6.3. Technique de la géolocalisation.....	30
6.4. Comparaison entre les techniques de géo-localisation .....	31

6.5.	Les outils et techniques de géolocalisation .....	32
6.5.1.	Google Maps.....	32
6.5.2.	Le système GPS.....	32
	Conclusion .....	32
<i>Chapitre II : Planification et architecture.....</i>		<i>33</i>
	Introduction : .....	33
1.	Analyse des besoins : .....	33
1.1.	Identification des acteurs :.....	33
1.2.	Les besoins fonctionnels: .....	34
1.3.	Les besoins non fonctionnels: .....	36
2.	Planning du traitement des cas d'utilisation :.....	36
2.1.	Priorités : .....	37
2.2.	Risque : .....	37
3.	Prototypage des interfaces : .....	37
4.	Le Backlog du produit : .....	39
	Conclusion .....	43
Release 1 : Introduction : .....		43
I.	Premier Sprint : .....	44
1.	Spécification fonctionnelle : .....	44
1.1.	Scénario et cas d'utilisation.....	44
1.1.1.	Description du cas d'utilisation « Gérer la demande » :.....	45
1.1.2.	Description du cas d'utilisation « Gérer la rencontre » .....	48
1.1.3.	Description du cas d'utilisation « Gérer statut » : .....	49
2.	Conception.....	50
2.1.	Diagramme de séquence système .....	50
a.	Diagramme de séquence système du cas d'utilisation « Gérer la demande » .....	52
b.	Diagramme de séquence système du cas d'utilisation « Gérer statut » .....	53
c.	Diagramme de séquence système du cas d'utilisation « Gérer rencontre ».....	54
2.2.	Diagramme des classes des cas d'utilisation .....	55
a.	Diagramme de classe du cas d'utilisation « Gérer la demande » .....	56
b.	Diagramme de classe du cas d'utilisation « Gérer statut » .....	57
c.	Diagramme de classe du cas d'utilisation « Gérer rencontre » .....	57
2.3.	Diagramme de séquence détaillé : .....	58
a.	Diagramme de séquence détaillé du cas d'utilisation « Gérer Demande » .....	58
b.	Diagramme de séquence détaillé du cas d'utilisation « Gérer statut » .....	61
c.	Diagramme de séquence détaillé du cas d'utilisation « Gérer la rencontre ».....	62

2.4.	Diagramme des classes de conception :.....	63
3.	Codage :.....	64
4.	Test :.....	66
4.1.	Test unitaire :.....	66
4.2.	Les interfaces:.....	66
II.	Deuxième Sprint :.....	68
1.	Spécification Fonctionnelle .....	68
1.1.	Scénario et cas d'utilisatin.....	68
1.2.	Description textuelle des cas d'utilisation .....	69
1.2.1.	Description du cas d'utilisation « Gérer la réservation » .....	69
1.2.2.	Description du cas d'utilisation « Gérer l'évaluation » .....	70
1.2.3.	Description du cas d'utilisation « Gérer le contact » .....	70
2.	Conception .....	71
2.1.	Diagramme de séquence système .....	71
a.	Diagramme de séquence système du cas d'utilisation « Gérer la réservation » .....	72
b.	Diagramme de séquence système du cas d'utilisation « Gérer le Contact » .....	73
c.	Diagramme de séquence système du cas d'utilisation « Gérer l'évaluation ».....	74
2.2.	Diagramme des classes de conception.....	74
a.	Diagramme de classe pour le cas « Gérer réservation ».....	75
b.	Diagramme de classe du cas d'utilisation « Gérer le Contact » .....	75
c.	Diagramme de classe du cas d'utilisation « Gérer l'évaluation ».....	76
2.3.	Diagramme de séquence détaillé.....	77
a.	Diagramme de séquence détaillé du cas d'utilisation « Gérer la réservation » .....	77
b.	Diagramme de séquence détaillé du cas d'utilisation « Gérer le contact ».....	78
c.	Diagramme de séquence détaillé du cas d'utilisation « Gérer l'évaluation ».....	79
2.4.	Diagramme des classes de conception :.....	80
3.	Codage :.....	81
4.	Test :.....	83
4.1.	Tests unitaires :.....	83
4.2.	Les interfaces :.....	83
	<i>Conclusion</i> :.....	84
	Release 2 :Gestion de la supervision .....	84
	Introduction :.....	84
I.	Le premier sprint : .....	85
1.	Spécification fonctionnelle :.....	85

1.1.	Scénario et cas d'utilisation :	85
1.2.	Description textuelle des cas d'utilisation :	86
1.2.1.	Description du cas d'utilisation « Consulter Conducteur	86
1.2.2.	.....	86
1.2.3.	Description du cas d'utilisation « Bannir conducteur » .....	86
1.2.4.	Description du cas d'utilisation « Valider Conducteur » .....	87
2.	Conception :	88
2.1.	Diagramme de séquence système :	88
a.	Diagramme de séquence système du cas d'utilisation « Consulter conducteur » .....	88
b.	Diagramme de séquence système du cas d'utilisation « Supprimer Conducteur » .....	89
c.	Diagramme de séquence système du cas d'utilisation « Bannir conducteur » .....	89
d.	Diagramme de séquence système du cas d'utilisation « Valider Conducteur » .....	90
2.2.	Diagramme des classes d'utilisation .....	91
a.	Diagramme de classe du cas d'utilisation « Consulter Conducteur » .....	91
b.	Diagramme de classe du cas d'utilisation « Supprimer Conducteur » .....	92
c.	Diagramme de classe du cas d'utilisation « Bannir conducteur » .....	93
d.	Diagramme de classe du cas d'utilisation « Valider Conducteur » .....	93
2.3.	Diagramme de séquences détaillées .....	94
a.	Diagramme de séquence détaillé du cas d'utilisation « Consulter conducteur » .....	94
b.	Diagramme de séquence détaillé du cas d'utilisation « Supprimer Conducteur » .....	95
c.	Diagramme de séquence détaillé du cas d'utilisation « Bannir Conducteur » .....	96
d.	Diagramme de séquence détaillé du cas d'utilisation « Valider Conducteur » .....	97
2.4.	Diagramme des classes de conception :	98
3.	Codage :	98
II.	Deuxième Sprint .....	100
5.	Spécification fonctionnelle :	100
5.1.	Scénario et cas d'utilisation .....	100
5.2.	Description textuelle des cas d'utilisation .....	101
5.2.1.	Description du cas d'utilisation « Gérer compte Administrateurs » .....	101
5.2.2.	Description du cas d'utilisation « Gérer Statistique .....	104
6.	Conception .....	105
6.1.	Diagramme de séquence système .....	105
a.	Diagramme de séquence système du sous cas « Consulter administrateur .....	105
b.	Diagramme de séquence système du sous cas « Supprimer Administrateur » .....	106
c.	Diagramme du sous cas système du sous cas d'utilisation « Ajouter Administrateur » 107	
6.2.	Diagramme des classes de conception .....	108

a.	Diagramme de classe du cas d'utilisation « Consulter compte administrateur ».....	108
b.	Diagramme de classe du cas d'utilisation « Gérer statistique » .....	109
6.3.	Diagramme de séquence détaillé.....	109
a.	Diagramme de séquence détaillé du sous cas d'utilisation « Ajouter Administrateur ».....	109
b.	Diagramme de séquence détaillé du sous cas d'utilisation « Supprimer Administrateur ».....	111
c.	Diagramme de séquence détaillé du sous cas d'utilisation « Consulter Administrateur ».....	112
6.4.	Diagramme de classes de conception : .....	113
III.	Interface : .....	113
<i>Chapitre V : La phase de clôture</i> .....		114
Introduction.....		114
1.	Environnement de travail : .....	114
1.1.	Choix de l'architecture de l'application.....	114
1.2.	Elaboration du diagramme de déploiement : .....	116
1.3.	Environnement matériel .....	116
1.4.	Technologies utilisées.....	117
1.4.1.	Java : .....	117
1.4.2.	Android : .....	117
1.4.3.	PHP : .....	117
1.4.4.	HTML5 : .....	117
1.5.	Environnements logiciels .....	118
1.5.1.	PHP Storm : .....	118
1.5.2.	Wamp server.....	118
1.5.3.	SGBD MySQL.....	118
2.	Protocole et formats de données.....	119
2.1.	Protocole de communication .....	119
2.2.	Format de données communiquées.....	119
Conclusion .....		119
Conclusion et perspectives .....		120
<i>Annexes</i> .....		121

## Liste des figures

Figure 1 : Carte d'identité de la société « 'X' » .....	<b>Error! Bookmark not defined.</b>
Figure 2: Application 'X' .....	15
Figure 3: – Application 'X' .....	16
Figure 4: Application GoCatch .....	18
Figure 6: – Application GoCatch. ....	18
Figure 7: Application GoCatch. ....	19
Figure 8: L'application 'Y' .....	21
Figure 9: L'application 'Y'. ....	<b>Error! Bookmark not defined.</b>
Figure 10 : Le processus Scrum .....	26
Figure 11: Diagramme de contexte statique.....	34
Figure 12:Interface authentification passager. ....	37
Figure14:Interface demander Taxi. ....	38
Figure13:Interface authentification conducteur. ....	38
Figure 15:Interface l'acceptation d'une demande. ....	39
Figure 16: Planification des sprints. ....	42
Figure 17: Diagramme de cas d'utilisation global .....	42
Figure 18: Description des cas d'utilisations du Sprint 1. ....	50
Figure 19:Diagramme de séquence système du cas d'utilisation "Gérer la demande".....	52
Figure 20:Diagramme de séquence système du cas d'utilisation "Gérer statut". ....	53
Figure 21: Diagramme de séquence système du cas d'utilisation "Gérer rencontre". ....	54
Figure 22:Diagramme de classe de conception du cas d'utilisation "Gérer la demande". ....	56
Figure 23:Diagramme de classe de conception du cas d'utilisation "Gérer Statut". ....	57
Figure 24:Diagramme de classe de conception du cas d'utilisation "Gérer rencontre".....	57
Figure 25: Diagramme de séquence détaillé du sous cas "Demande rapide". ....	58
Figure 26: Diagramme de séquence détaillé du sous cas "Demande par sélection". ....	59
Figure 27: Diagramme de séquence détaillé du sous cas "Consulter profil". ....	60
Figure 28:Diagramme de séquence détaillé du cas d'utilisation "Gérer statut". ....	61
Figure 29:Diagramme de séquence détaillé du cas d'utilisation "Gérer la rencontre" .....	62
Figure 30: Diagramme de classe de conception du sprint 1.....	63
Figure 31: Interface authentification Conducteur. ....	67
Figure 32: Interface de sélection de TAXI(Passager). ....	67
Figure 33: Interface effectuer une demande.....	67
Figure 34: Description des cas d'utilisation du Sprint 2.....	71
Figure 35: Diagramme de séquence système du cas d'utilisation "Gérer la réservation". ....	72
Figure 36: Diagramme de séquence système "Gérer le contact". ....	73
Figure 37: Diagramme de séquence système du cas d'utilisation "Gérer l'évaluation". ....	74
Figure 38: Diagramme de classe du cas d'utilisation "Gérer réservation". ....	75
Figure 39: Diagramme de classe du cas d'utilisation "Gérer le Contact".....	75
Figure 40: Diagramme de classe du cas d'utilisation "Gérer l'évaluation". ....	76
Figure 41: Diagramme de séquence détaillé du cas d'utilisation "Gérer réservation". ....	77



Figure 42: Diagramme de séquence du cas d'utilisation "Gérer le contact passager" .....	78
Figure 43: Diagramme de séquence détaillé du cas d'utilisation "Gérer l'évaluation" .....	79
Figure 44: Diagramme de classe de conception du sprint 2.....	80
Figure 45: Cas de récupération de demandes.....	<b>Error! Bookmark not defined.</b>
Figure 46: Interface de demandes reçues. ....	84
Figure 47: Interface de rencontre. ....	84
Figure 48: Description des cas d'utilisation du Sprint 1.....	87
Figure 49: Diagramme de séquence système du cas d'utilisation "Consulter conducteur" .....	89
Figure 50: Diagramme de séquence système du cas d'utilisation "Supprimer conducteur" ....	89
Figure 51: Diagramme de séquence système du cas d'utilisation "Bannir conducteur". ....	90
Figure 52: Diagramme de séquence système du cas d'utilisation "Valider Conducteur". ....	91
Figure 53: Diagramme de classe de conception "Consulter conducteur". ....	92
Figure 54: Diagramme de classe de conception du cas d'utilisation "Supprimer conducteur".	92
Figure 55: Diagramme de classe de conception "Bannir conducteur". ....	93
Figure 56: Diagramme de classe de conception du cas d'utilisation "Valider Conducteur". ...	93
Figure 57: Diagramme de séquence détaillé du cas d'utilisation "Consulter conducteur" .....	94
Figure 58: Diagramme de séquence détaillé du cas d'utilisation "Supprimer Conducteur" .....	95
Figure 59: Diagramme de séquence détaillé du cas d'utilisation "Bannir Conducteur" .....	96
Figure 60: Diagramme de séquence détaillé du cas d'utilisation "Valider Conducteur" .....	97
Figure 61: Diagramme de classe de conception du sprint 1.....	98
Figure 62: Description du sprint 2 (release 2). ....	104
Figure 63: Diagramme de séquence système du sous cas d'utilisation "Ajouter Administrateur". .....	105
Figure 64: Diagramme de séquence détaillé du sous cas d'utilisation "Supprimer Administrateur". ....	106
Figure 65: Diagramme de séquence système du sous cas d'utilisation "Ajouter Administrateur". .....	107
Figure 66: Diagramme de classe de conception du cas d'utilisation "Gérer comptes administrateur". ....	108
Figure 67: Diagramme de classe de conception du cas d'utilisation "Gérer statistique". ....	109
Figure 68: Diagramme de séquence détaillé du sous cas d'utilisation "Ajouter Administrateur". .....	110
Figure 69: Diagramme de classe de conception du sous cas d'utilisation "Supprimer Administrateur". ....	111
Figure 70: Diagramme de séquence de conception du sous cas d'utilisation "Consulter Administrateur". ....	112
Figure 71: Diagramme de classe de conception du sprint 2.....	113
Figure 72: Interface administrateur. ....	113
Figure 73:Architecteure générale de l'application. ....	115
Figure 74:Diagramme de déploiement .....	116

## Liste des Tableaux

Tableau 1: Avantages et inconvénients d'application GoCatch Android.....	20
Tableau 2: Synthèse des fonctionnalités.....	24
Tableau 3: Comparaison entre les techniques de géolocalisation. ....	32
Tableau 4: Backlog produit. ....	41
Tableau 5 : Backlog du premier sprint (release1).....	44
Tableau 6: Description du cas d'utilisation "S'authentifier passager" .....	45
Tableau 7: Description du cas d'utilisation "Envoyer une demande rapide". ....	46
Tableau 8: Description du cas d'utilisation "Envoyer une demande à un Taxi sélectionné". ....	46
Tableau 9: Description du cas d'utilisation "Consulter le profil du conducteur". ....	47
Tableau 10 : Description du cas d'utilisation "Consulter le profil du conducteur" .....	47
Tableau 11: Description du cas d'utilisation "Consulter les informations d'un Taxi sélectionné" .....	47
Tableau 12: Description du cas d'utilisation "Annuler la demande". ....	48
Tableau 13: Tableau 13: Description du cas d'utilisation "Gérer la rencontre" .....	48
Tableau 14: Description du cas d'utilisation "Gérer Statut" .....	49
Tableau 15: Structure de la table " Taxi " .....	64
Tableau 16: Structure de la table "Passager".....	65
Tableau 17: Structure de la table "Demande". ....	65
Tableau 18: Cas d'envoi de demande au conducteur.....	<b>Error! Bookmark not defined.</b>
Tableau 19: Backlog du second sprint (Release 1).....	68
Tableau 20: Description du cas d'utilisation "Gérer la réservation". ....	69
Tableau 21: Description du cas d'utilisation "Gérer l'évaluation". ....	70
Tableau 22: Description du cas d'utilisation "Gérer le contact". ....	70
Tableau 23: Structure de la table " Taxi " .....	81
Tableau 24: Structure de la table "Passager".....	82
Tableau 25: Structure de la table "Demande". ....	82
Tableau 26: Backlog du premier sprint (release2).....	85
Tableau 27: Description du cas d'utilisation "Consulter conducteur". ....	86
Tableau 28: Description du cas d'utilisation "Bannir conducteur". ....	86
Tableau 29: Description du cas d'utilisation "Valider Conducteur". ....	88
Tableau 30: Structure de la table " Taxi " .....	99
Tableau 31: Structure de la table "Passager".....	99
Tableau 32: Structure de la table "Administrateur".....	99
Tableau 33: Backlog du quatrième sprint (release 2).....	100
Tableau 34: Description du sous cas d'utilisation "Consulter Administrateur".....	101
Tableau 35: Description du sous cas d'utilisation "Ajouter Administrateur". ....	102
Tableau 36: Description du sous cas d'utilisation "Supprimer Administrateur". ....	102
Tableau 37: Description du sous cas d'utilisation "Modifier Administrateur". ....	103
Tableau 38: Description du cas d'utilisation "Gérer statistique". ....	104

# Introduction Générale

De nos jours, le transport public devient de plus en plus inconfortable, surtout dans une ville peuplée. En effet, les rues sont devenues de plus en plus encombrées notamment, aux heures de pointe.

Les Smartphones et les applications mobiles rencontrent un succès fulgurant. Après seulement quelques années, ce marché est devenu un secteur économique important. Le cabinet Gartner a estimé qu'en 2013 environ 102 milliards d'applications mobiles avaient été téléchargées, soit une hausse importante par rapport à 2012 qui s'était conclue sur un total de 64 milliards. Des résultats qui concernent, principalement, les boutiques d'IOS et Android, qui réunissent environ 1 million d'applications, pour chacune d'elles.

Les applications mobiles ont été initialement proposées pour la productivité et la récupération d'information, comprenant courrier électronique, calendrier électronique, contacts, marché boursier et informations météorologiques. Toutefois, la demande du public et la disponibilité d'outils de développement ont conduit une expansion rapide dans d'autres domaines, comme les jeux mobiles, les automatismes industriels, les GPS et services basés sur la localisation.

Dans ce contexte, on se propose de présenter une application qui facilite le transport en cherchant un taxi par le biais d'un téléphone et dont le principe est de trouver et choisir le taxi libre le plus proche du client.

C'est dans ce cadre que se situe ce projet de fin d'études intitulé : « Conception et développement d'une application de géolocalisation des Taxis » dont l'objectif est de concevoir une application appelée « ***Kb\_Driver*** » dédiée aux téléphones mobiles tablette et phablette.

Cette application permet à l'utilisateur de profiter de ce service grâce aux outils et techniques de géolocalisation et d'orientation offerts par cette plateforme.

Notre application est destinée aux périphériques mobiles ce qui donnera la possibilité aux clients de pouvoir demander un taxi quel que soit l'endroit où ils se trouvent.

Ainsi, un superviseur disposant d'un terminal web pourrait consulter en temps réel un rapport d'activités où figureront les informations utiles sur le trafic de Taxis telles que : le nombre des demandes effectuées, nombre d'utilisateurs connectés, leurs positions, les réservations en cours d'exécution, etc.

En conséquence, ce rapport s'articule autour de cinq chapitres comme suit :

***Le premier chapitre*** « étude préalable » permet de placer le projet dans son contexte général. Dans ce premier chapitre introductif nous présentons l'organisme d'accueil ainsi qu'une brève description du projet.

***Le second chapitre***, « planification et architecture » illustre la première phase dans le cycle Scrum. Dans ce chapitre, nous dévoilons les principales exigences de notre application, nous présenterons quelques interfaces graphiques pour mettre notre application dans son contexte et nous le clôturerons par un planning de travail.

***Le troisième chapitre*** « gestion des demandes et réservations »

***Le quatrième chapitre*** « gestion de la supervision » constituent le corps de notre rapport. Ces deux chapitres seront consacrés pour le développement des deux releases de notre système en respectant les principes fondamentaux de Scrum.

***Le cinquième chapitre*** « la phase de clôture » détaille tous les outils utilisés pour la conception et le développement de notre application ainsi que quelques captures d'écran de la version finale de notre système.

---

# ***CHAPITRE I : ETUDE PREALABLE***

---

## **Introduction :**

Dans ce chapitre nous nous intéressons à la présentation de la société **Kb\_Driver** au sein de laquelle j'ai effectué mon stage.

Ce chapitre contient aussi la présentation de mon projet en incluant l'étude de l'existant.

### **1. Présentation du cadre de stage :**

## 1.1. Présentation de l'organisme d'accueil :

Dans le cadre du projet de fin d'études, nous allons procéder à une présentation générale de l'organisme d'accueil afin de mieux comprendre le contexte. Mon stage s'est déroulé au sein de la société ***Kb\_Driver***.

## 1.2. Présentation de la société KB DRIVER :

***Kb\_Driver*** est une Société de Service de Taxi de référence dans le domaine des transports luxueux, opérant en France avec une clientèle internationale.

En effet ***Kb\_Driver*** opère ses services en France depuis les quatre coins du monde.

Grâce à son expertise et sa certification aux normes internationales, ***Kb\_Driver*** exporte ainsi sa notoriété au-delà des frontières.

Kb\_Driver est un partenaire de choix pour la mise en place de solutions de services personnalisées. Elle étudie les attentes, conseille, crée, assiste et développe solution pour ses clients très délicats. Elle met à la disposition de sa clientèle les meilleurs services d'une stratégie de webmarketing interactive réussie.

Les services développés par ***Kb\_Driver*** sont :

- \*prise en charge total depuis l'aéroport vers l'hôtel partenaire
- \*conduite vers les restaurants, musées, attractions diverses et culturels
- \*sécurité et discrétion assuré par les chauffeurs
- \*disposition totale avec véhicule
- \*accompagnement de garde du corps

***Kb\_Driver*** propose à ses clients des solutions modulaires et performantes.

## 1.3. Cadre du sujet

Le domaine de la mobilité dans la géolocalisation sociale est un secteur très concurrentiel. L'essentiel des programmes qui existent ou apparaissent sur le marché est basé sur un simple check-in (action qui permet d'actualiser la position).

La plupart des sites sociaux essaient aujourd'hui, de fusionner les services de géolocalisation afin d'enrichir leurs domaines d'activités et d'améliorer leurs fonctionnalités offertes aux utilisateurs.

Les systèmes de géolocalisation sociale nous permettent de connaître la position de nos amis et de savoir leurs opinions sur les lieux qu'ils visitent.

L'intégration de ces services dans leurs Smartphones perfectionne la qualité du service de ces systèmes et leurs apportent des fonctions supplémentaires.

#### **1.4. Impact du sujet**

En adoptant cette solution, « **Kb\_Driver** » obtiendra fabuleusement des bénéfices que l'on peut résumer comme suit :

- Affermissement de sa position dans le développement des prestations de service avec cette application pour Smartphones.
- Investissement de ses expériences et capacités dans les applications se basant sur les technologies de la géolocalisation sociale que l'on trouve fréquemment répandues sur le marché des applications mobiles et très appréciées de la part des utilisateurs.

## **2. Etude de l'existant :**

L'étude de l'existant permet de déterminer les points faibles et les points forts d'un produit actuel pour pouvoir déterminer les besoins du client, en vue d'en prendre en considération lors de la conception et la réalisation de mon application. Dans cette section, je présente une analyse de quelques exemples d'applications marchands. Ensuite, je formulerai une solution de la problématique.

### **2.1. Analyse de l'application « 'X' » :**

Dans cette section, je présenterai l'application 'X'. Ensuite, nous procéderons à une analyse fonctionnelle et une critique de l'application étudiée.

#### **2.2.1. Présentation de l'application » 'X' » :**

‘X’ permet à tous ceux qui possèdent un Smartphone de réserver un taxi immédiatement. Quelques secondes plus tard, l'utilisateur de ‘X’ pourra visualiser sur son écran le chauffeur de taxi qui a accepté la demande et suivre l'approche directement sur l'écran. Les frais de service de « ‘X’ » sont à payer directement au chauffeur.

### 2.2.2. Les fonctionnalités :

Suivant la figure 2, l'application permet aux utilisateurs de :

- Choisir entre un taxi rapide et un chauffeur privé.
- Saisir l'adresse de départ et d'arrivée avec le nombre de passagers.
- Saisir un commentaire (optionnel).
- Valider la création de course.
- Connaître sa position actuelle dans la carte géographique.
- Signaler la fin de la course.

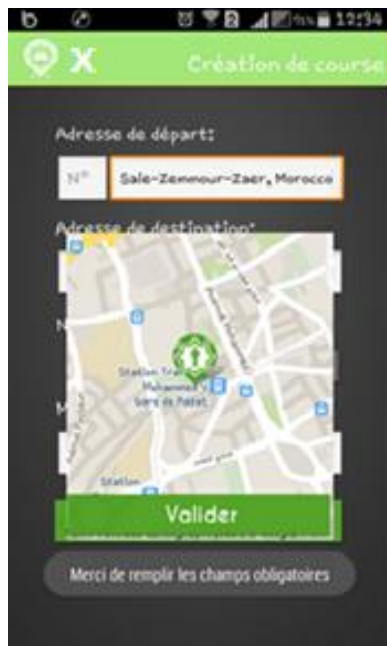


(a)



(b)

Figure 1: Application ‘X’.



(a)



(b)

Figure 2: – Application ‘X’

- La figure 2 (a) illustre la page d'accueil de l'application pour l'utilisateur.
- La figure 2 (b) illustre la course en cours en précisant l'emplacement du passager et le taxi.
- La figure 3 (a) illustre la vérification de l'adresse saisie sur la carte géographique.
- La figure 3 (b) illustre la page de création de course détaillée.

### 2.2.3. Critique de l'application « 'X' » :

L'application présente un certain nombre de complexité. Ces difficultés sont présentées comme suit :

L'application peut être non compréhensible pour l'utilisateur.

Du temps perdu lors de la validation de la création.

Le client ne peut pas suivre son taxi sur une carte géographique.

Il y a une absence d'informations sur l'emplacement actuel du taxi avant d'effectuer une réservation.



## 2.3. Analyse de l'application « GoCatch » :

Dans cette section, nous présenterons l'application GoCatch. Ensuite, nous procéderons à une analyse fonctionnelle et une critique de l'application étudiée.

### 2.3.1. Présentation de l'application « GoCatch » :

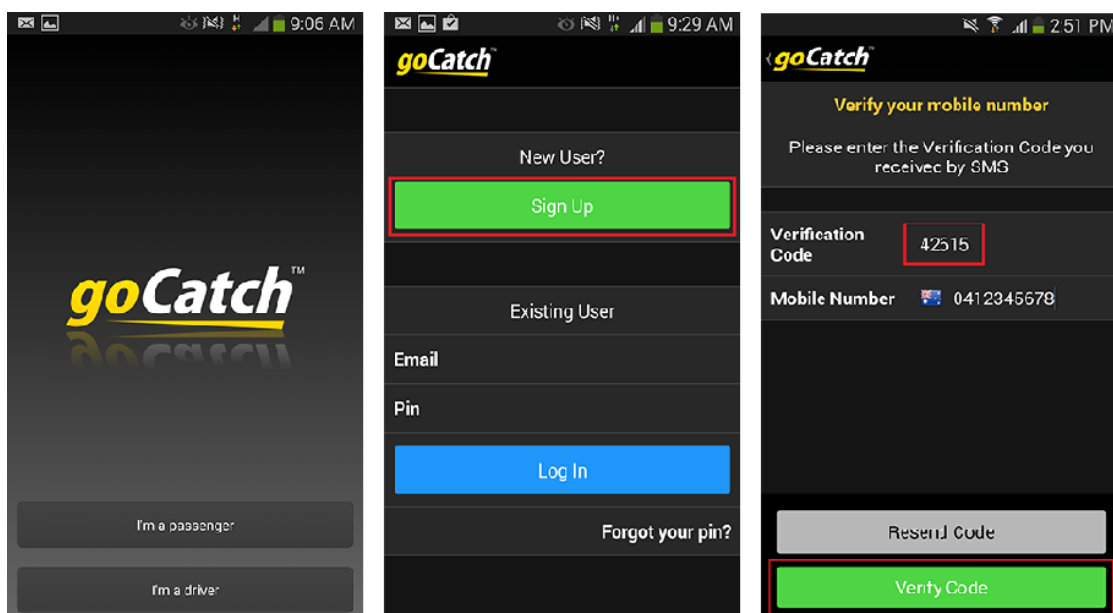
L'application GoCatch permet aux conducteurs et passagers de se connecter directement et suivre les emplacements des taxis en temps réel. Elle permet aussi aux passagers de savoir quand le conducteur arrivera et le voir arriver sur la carte.

Le chauffeur doit garder GoCatch ouvert lorsqu'il conduit pour trouver une course rapidement.

### 2.3.2. Les fonctionnalités :

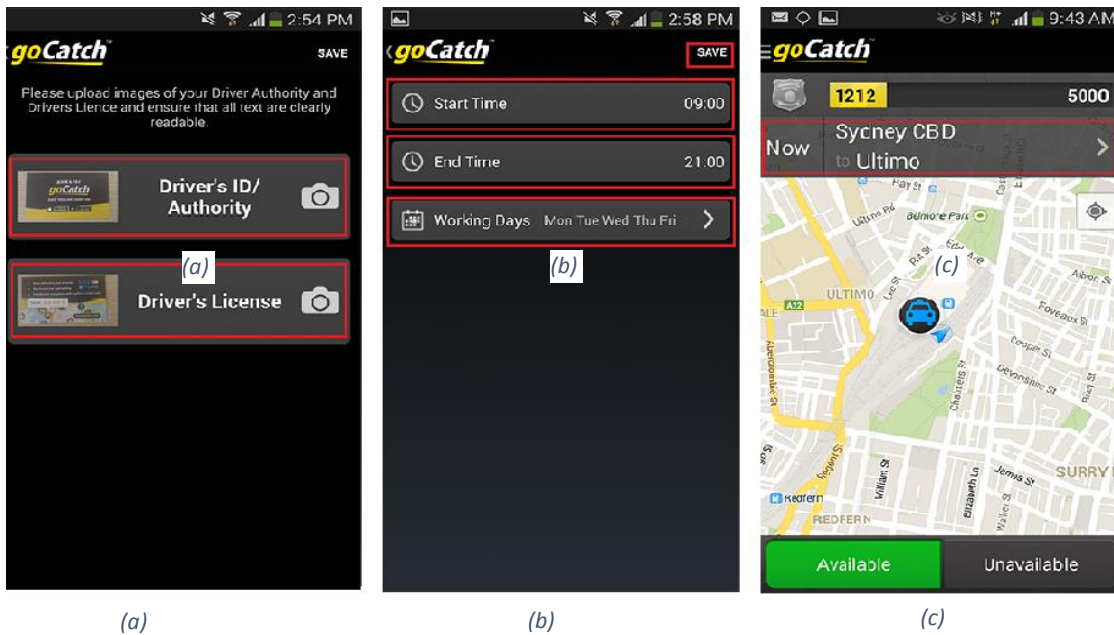
Cette application permet au conducteur de taxi de :

- Créer un profil vérifié.
- Choisir son horaire de travail.
- Préciser s'il est disponible ou occupé.
- Préciser les coordonnées de son compte bancaire.
- Voir son emplacement actuel sur la carte géographique
- Accepter ou refuser une course.
- Voir le repère de son passager sur la carte.
- Appeler le passager qui a accepté.
- Confirmer lorsqu'il arrive à son client.
- Confirmer l'arrivée du passager à sa destination souhaitée.



**Figure 3: Application GoCatch**

- La figure 4 (a) illustre la première ouverture de l'application.
- La figure 4 (b) illustre la page de connexion et la possibilité de créer un nouveau compte.
- La figure 4 (c) montre la vérification d'un nouveau compte créé avec un sms sur le téléphone portable.



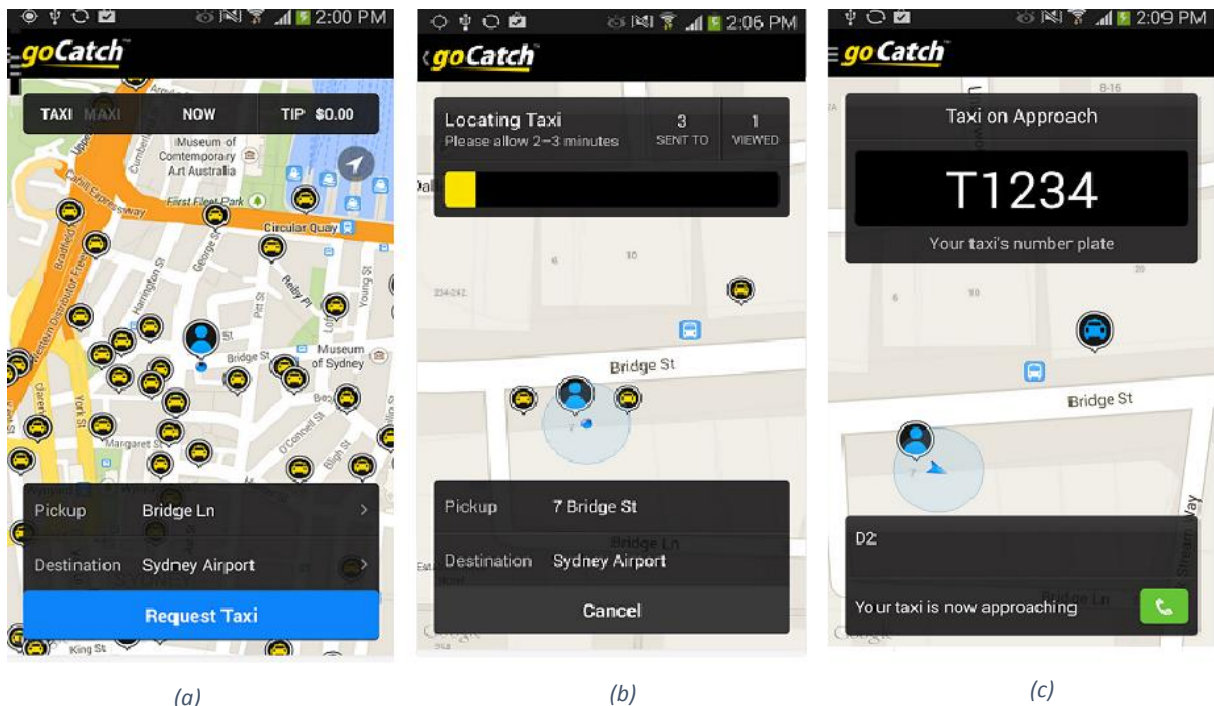
**Figure 4: – Application GoCatch.**

- Figure 5 (a) illustre le téléchargement des pièces justificatives pour vérifier le compte du conducteur.
- Figure 5 (b) illustre la page de choix d'horaire du travail.
- Figure 5 (c) montre l'emplacement de taxi sur le plan avec la possibilité de choisir son mode actuel (disponible ou occupé).

Elle permet aussi au passager de :

- Créer un profil vérifié.
- Voir le trafic des taxis dans sa région.
- Demander un taxi en précisant le lieu de rencontre avec le conducteur et la destination.

- Avoir un aperçu sur le nombre de conducteurs qui ont reçu la demande et ceux qui l'ont vu.
- Connaitre le numéro de taxi réservé d'avance.
- Voir le taxi en mouvement sur le plan.
- Confirmer l'arrivée du taxi.
- Appeler le chauffeur.
- Payer par espèce ou par le biais d'une carte bancaire.
- Noter le conducteur.



**Figure 5: Application GoCatch.**

- Figure 6 (a) illustre la page d'accueil de l'application dans laquelle l'utilisateur saisit le lieu de rencontre et la destination pour demander un taxi.
- Figure 6 (b) montre l'attente d'une réponse d'un conducteur avec un aperçu sur le plan des taxis à proximité.
- Figure 6 (c) illustre l'emplacement du passager et du taxi réservé sur le plan.

### 2.3.3. Critique de l'application « GoCatch » :

Dans ce tableau nous présenterons quelques avantages et inconvénients pour cette application.

<i>Avantages</i>	<i>Inconvénients</i>
<ul style="list-style-type: none"> <li>• Les comptes d'utilisateurs (passagers et conducteurs) sont vérifiés à travers le numéro du téléphone.</li> <li>• Choisir l'horaire du travail.</li> <li>• Avoir la possibilité de mettre occupé ou disponible au cours de son travail.</li> <li>• Affichage de la position de l'utilisateur sur la carte, ainsi que le nom de la rue en gras.</li> <li>• Le passager peut informer le conducteur de sa destination au cours de la demande.</li> <li>• Les deux utilisateurs peuvent effectuer un appel téléphonique entre eux.</li> <li>• L'utilisation de carte est plus compréhensible et simple pour l'utilisateur.</li> <li>• L'utilisation des TIC, réseau 3G, Google API, GPS, etc.</li> <li>• Gain de temps.</li> <li>• Paiement sécurisé.</li> </ul>	<ul style="list-style-type: none"> <li>• Le passager n'a pas une idée s'il y a des taxis « disponible » à proximité.</li> <li>• Tous les taxis sont affichés avec la même couleur sur le plan, il n'y a pas de différence entre disponibles et occupés.</li> <li>• Si le passager quitte le point de rencontre, le conducteur ne sera pas informé. (Aucune alerte)</li> <li>• Absence d'itinéraire entre le passager et le taxi.</li> <li>• La demande est envoyée automatiquement à plusieurs conducteurs à la fois qui sont à proximité du passager.</li> <li>• Le temps nécessaire pour arriver au client n'est pas affiché chez ce dernier.</li> <li>• La distance entre ces deux acteurs est inconnue.</li> </ul>

**Tableau 1: Avantages et inconvénients d'application GoCatch.**

## **2.4. Analyse de l'application « 'Y' » :**

Dans cette section, nous présenterons l'application 'Y'. Ensuite nous procéderons à une analyse fonctionnelle et une critique de l'application étudiée.

### **2.4.1. Présentation de l'application « 'Y' » :**

Cette application consiste à réserver le taxi le plus proche. En l'attendant, le client peut consulter les informations du chauffeur tel que son nom, son numéro de téléphone et la marque de sa voiture. Il peut aussi le suivre en temps réel sur la carte.

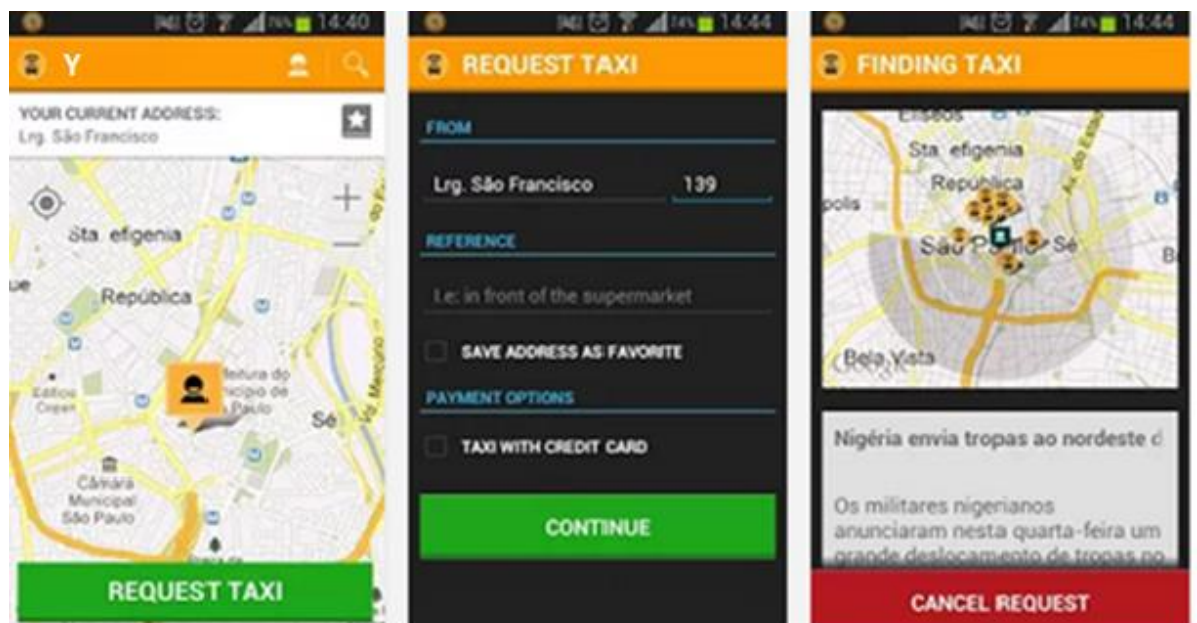
### 2.4.2. Les fonctionnelles :

*Cette application permet au passager de :*

1. Demander un Taxi.
2. Saisir l'adresse de départ (Ou encore le lieu de rencontre).
3. Préciser une référence (optionnelle).
4. Saisir le mode de paiement.
5. Appeler le chauffeur de Taxi.
6. Connaître sa position actuelle sur la carte géographique.
7. Annuler la demande de Taxi (avec la Saisie de cause d'arrêt).
8. Enregistrer le Taxi en tant que favoris.
9. Voir l'historique de ses mouvements.

*Cette application permet au conducteur de :*

1. Consulter les autres chauffeurs de taxi se trouvant dans sa région.
2. Appeler le passager.
3. Accepter ou refuser la demande du passager.
4. Connaître sa position actuelle sur la carte géographique.
5. Voir l'historique de ses mouvements.



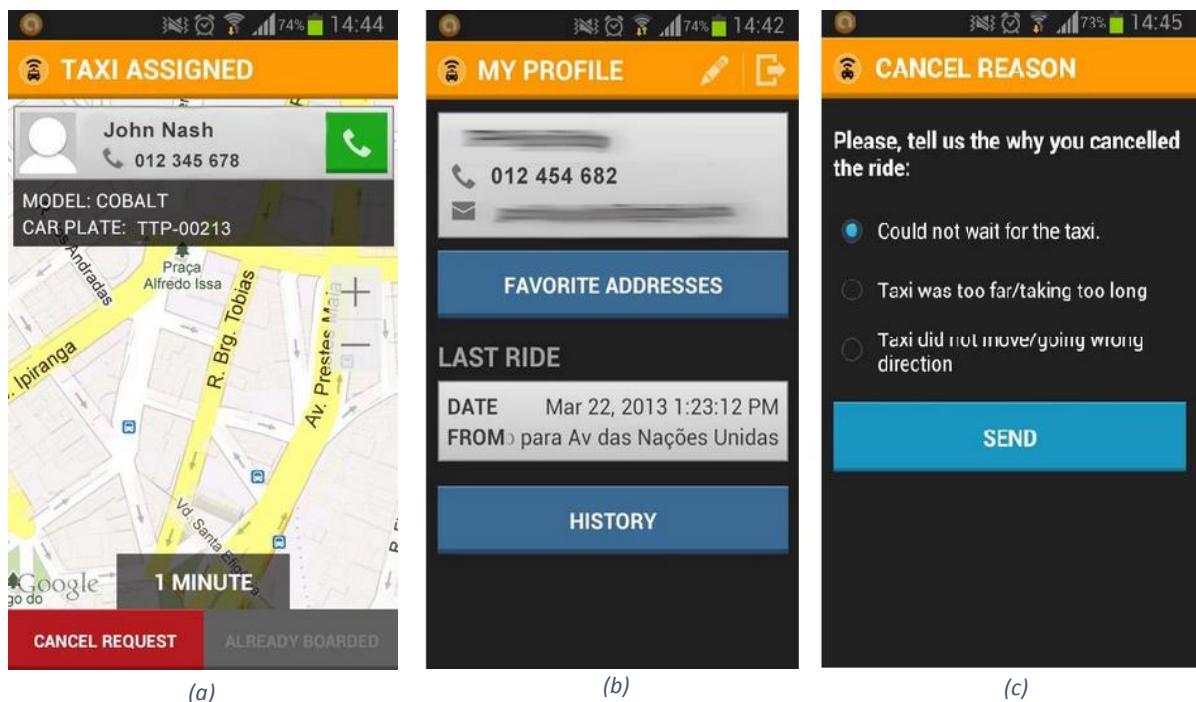
(a)

(b)

(c)

Figure 6: L'application 'Y'

- La figure 7 (a) illustre la page d'accueil de l'application pour le passager.
- La figure 7 (b) illustre la page de demande de Taxi détaillée.
- La figure 7 (c) illustre la position du client ainsi que les Taxis autour de lui.
- 



- La figure 8 (a) illustre l'appel d'un taxi au client.
- La figure 8 (b) illustre la page de profil du passager.
- La figure 8 (c) illustre un questionnaire au cas où le passager veut annuler sa demande.

### 2.4.3. Critique de l'application « 'Y' » pour Android :

L'application présente un certain nombre de complexités. Ces difficultés sont présentées comme suit :

- L'application présente beaucoup d'interfaces pour l'utilisateur.



- L'application ne permet pas aux passagers de suivre leurs taxis.
- Manque d'informations sur la disponibilité des taxis.
- Cette application est liée à la centrale.

## 2.5. Synthèse :

Le tableau 1 synthétise les fonctionnalités les plus importantes des solutions étudiées

« 'X' », « GoCatch » et « 'Y' ».

Les lignes qui ne sont pas marquées par des croix et signalées avec une couleur différente sont des fonctions que nous allons étudier dans notre solution proposée.

Les applications	'X'	GoCatch	'Y'
Les fonctionnalités			
<b>Module de Sécurité :</b>			
Profil classique			
Profil Facebook		X	
Vérification par SMS		X	
Vérification par appel téléphonique			
Vérification des pièces justificatives.		X	
<b>Module de géolocalisation</b>			
Carte géographique	X	X	X
Itinéraire			
Calcul de distance			
Calcul du temps			X
Point de rencontre (placer un point sur la carte)			
Différentes couleurs pour les Taxis sur le plan			

<b>Numéro de Taxi</b>		<b>X</b>	<b>X</b>
<b>Compteur (cout estimé)</b>			
<b>Module de contact</b>			
<b>Appel téléphonique</b>		<b>X</b>	<b>X</b>
<b>Référence (message personnalisé du client)</b>			<b>X</b>

Tableau 2: Synthèse des fonctionnalités

## 2.6. Les Solution proposées :

Après une étude approfondie de l'existant, plusieurs limites des applications de géolocalisation des taxis pour téléphones présents sur le marché ont été identifiées. Toutes les solutions sont incomplètes car elles ne tiennent pas compte de tous les besoins de gestion inhérents à ce type d'application. De ce fait, je propose la conception et le développement d'une application de géolocalisation des taxis qui offrent aux propriétaires de téléphone une solution permettant la demande d'un taxi à tout moment tout en profitant d'un maximum d'outils disponibles pour le développement, qui seront certes utiles et bénéfiques pour répondre à la problématique citée précédemment. D'autre part, ma solution est bénéfique au point de vue organisation de la route grâce à l'outil de choix du taxi le plus proche de la position du client.

## 3. Langage et méthodologie de conception

La méthodologie est une démarche organisée rationnellement pour aboutir à un résultat. Parmi les différentes méthodologies existantes, nous pouvons citer le modèle en cascade utilisée souvent dans les simples projets dont les besoins sont clairs et bien définis dès le début, le modèle en Y utiliser pour le développement des applications mobiles, ainsi que le processus unifié et les méthodologies agiles (Scrum & extrême programming) caractérisées par leurs souplesses et utilisées dans des grands projets.

Pour bien conduire mon projet et m'assurer du bon déroulement des différentes phases, j'ai opté pour Scrum comme une méthodologie de conception et de développement.

Après le choix de la méthodologie, j'ai eu besoin d'un langage de modélisation unifiée pour la modélisation de mon projet. Pour concevoir mon système, j'ai choisi UML<sup>1</sup> comme langage de modélisation.

---

<sup>1</sup> Unified Modeling Language



Mon choix s'est basé sur les points forts de ce langage notamment sa standardisation et les divers diagrammes qu'il propose. Aussi UML présente le meilleur outil pour schématiser des systèmes complexes sous un format graphique et textuel simplifié et normalisé.

En effet, UML n'est ni un processus ni une démarche, d'où il fallait choisir une méthodologie de conception et de développement que je dois adopter.

## 4. Pourquoi Scrum

*« Scrum signifie mêlée au rugby. Scrum utilise les valeurs et l'esprit du rugby et les adaptes aux projets de développement. Comme le pack lors d'un ballon porté au rugby, l'équipe chargée du développement travaille de façon collective, soudée vers un objectif précis. Comme un demi de mêlée, le Scrum Master aiguillonne les membres de l'équipe, les repositionne dans la bonne direction et donne le tempo pour assurer la réussite du projet. » [2]*

Scrum est issu des travaux de deux des signataires du Manifeste Agile<sup>2</sup>, Ken Schwaber et Jeff Sutherland, au début des années 1990. Il appartient à la famille des méthodologies itératives et incrémentales et repose sur les principes et les valeurs agiles.

Le plus souvent, les experts de Scrum, même ses fondateurs, le décrivent comme un cadre ou un patron de processus orienté gestion de projet et qui peut incorporer différentes méthodes ou pratiques d'ingénierie. S'il est difficile de définir la nature de Scrum, sa mise en place est beaucoup plus simple et peut être résumée par la Figure 7.

Le principe de base de Scrum est le suivant :

- Dégager dans un premier lieu le maximum des fonctionnalités à réaliser pour former le back log du produit,
- En second lieu définir les priorités des fonctionnalités et choisir lesquelles seront réalisées dans chaque itération,
- Par la suite focaliser l'équipe de façon itérative sur l'ensemble de fonctionnalités à réaliser, dans des itérations appelées Sprints,
- Un Sprint aboutit toujours sur la livraison d'un produit partiel fonctionnel appelé incrément.

---

<sup>2</sup> Le manifeste agile est un texte rédigé et signé en 2001 par 17 experts dans le domaine de développement d'applications informatique.

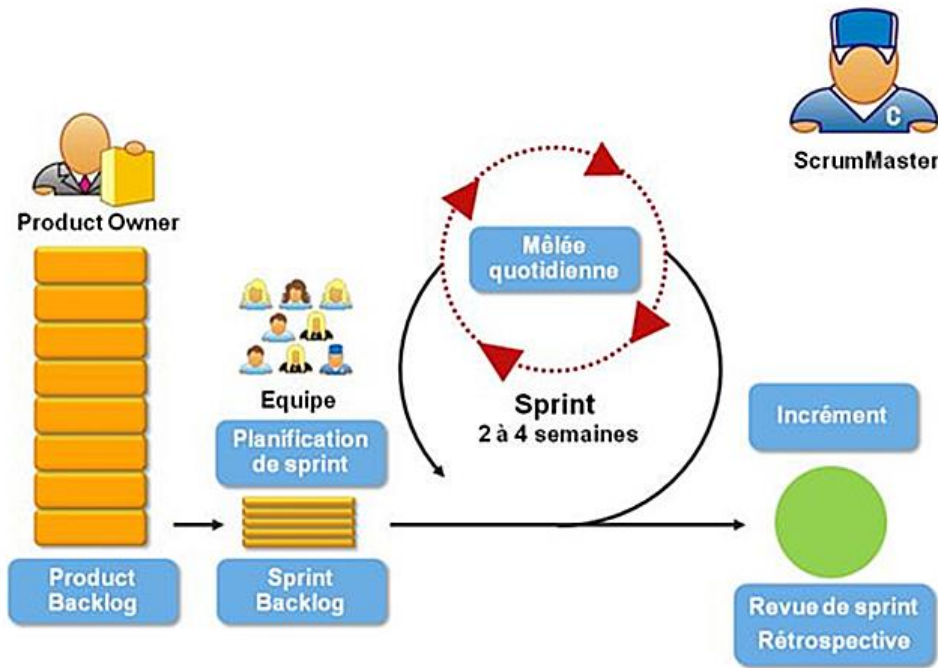


Figure 7 : Le processus Scrum

Le choix de Scrum comme une méthodologie de pilotage pour mon projet s'est basé sur les atouts de ce dernier. Il se résume comme suit :

- Plus de souplesse et de réactivité,
- La grande capacité d'adaptation au changement grâce à des itérations courtes,
- La chose la plus importante, c'est que Scrum rassemble les deux cotés théorique et pratique et se rapproche beaucoup de la réalité.

Vu que Scrum ne couvre que les aspects de gestion de projet, et pour compléter le vide laissé en matière de pratiques de développement, j'ai pris la décision de coupler Scrum avec une autre méthodologie agile qui est l'extrême programming et qui couvre les bonnes pratiques d'ingénierie logicielle notamment le développement dirigé par le test, qui sera détaillé dans les chapitres qui suivent, et la programmation en binôme, etc.

## **5. Pilotage du projet avec Scrum :**

### **5.1. Planification d'un projet par Scrum :**

Planification du sprint : Elle s'appuie sur la planification de la « release » réalisée en pente. La première réunion du sprint ne se limite pas à planifier, on y trouve les activités suivantes :

- 1) Valider les « stories » du Back log pris en compte dans le sprint, concevoir les solutions.
- 2) Identifier et estimer les tâches.
- 3) Prise des tâches par chacun des membres de l'équipe ...

Revue du sprint : Elle permet de montrer les résultats du développement effectués au cours du sprint, seule une version opérationnelle est montrée.

Rétrospective : Elle est faite en interne en équipe (avec la présence du Scrum Master), l'objectif est de comprendre ce qui n'a pas bien fonctionné dans le sprint, les erreurs commises et de prendre des décisions pour procéder aux améliorations.

Scrum quotidien : il s'agit d'une réunion de synchronisation de l'équipe de développement qui se fait debout en 15 minutes maximum au cours de laquelle chacun répond principalement à 3 questions :

- 1) Qu'est-ce que j'ai fait hier ?
- 2) Qu'est-ce que je ferai aujourd'hui ?
- 3) Quels obstacles me retardent ?

### **5.2. Equipe et rôles :**

« L'équipe a un rôle capital dans Scrum : elle est constituée dans le but d'optimiser la flexibilité et la productivité ; pour cela, elle s'organise elle-même et doit avoir toutes les compétences nécessaires au développement du produit. Elle est investie avec le pouvoir et l'autorité pour faire ce qu'elle a à faire ».<sup>3</sup>

Bref, Scrum définit trois rôles qui sont :

---

<sup>3</sup> C. Aubry, SCRUM le guide pratique de la méthode agile la plus populaire, Dunod, 2010.

**Le Product Owner (le propriétaire du produit) :** c'est une personne qui porte la vision du produit à réaliser, généralement c'est un expert dans le domaine.

**Le Scrum Master (le directeur de produit) :** c'est la personne qui doit assurer le bon déroulement des différents sprints du release, et qui doit impérativement maîtriser Scrum.

**Le Scrum Team (l'équipe de Scrum) :** constitué des personnes qui seront chargées d'implémenter les différents besoins du client. Bien évidemment, cette équipe sera constituée des développeurs, des testeurs, etc.

Dans le contexte de notre projet, Karim Belhadri sera à la fois le propriétaire et le directeur de produit puisqu'il satisfait les différents pré-requis des deux rôles cités précédemment et moi je formerai le membre de l'équipe Scrum.

## **6. Généralités sur la localisation**

### **6.1. Historique et évolution des besoins de localisation**

L'être humain a toujours eu besoin de localiser les objets et de se situer dans l'environnement. Pour répondre à cette nécessité, plusieurs techniques ont été utilisées.

Au début de l'Humanité, l'homme utilisait les pierres (ou montagnes) pour se repérer. Ensuite les techniques de localisation à l'aide des "corps célestes" ont été utilisées à savoir les éléments naturels utilisés comme points de référence sont bien sûr le soleil, la lune et les étoiles.

Ces éléments naturels ont défini le cap à suivre pour rejoindre leur point de destination. Plus tard, Dans les années 1950, les premières techniques de localisation par les ondes radio sont apparues. Ainsi, afin d'offrir une couverture totale de la planète par ce type de service, l'idée d'un système de localisation par satellite s'est imposée, ceci a été satisfait avec le système GPS (Global Positioning System) qui sera décrit ultérieurement.

Cependant les technologies de localisation ont connu un essor important avec le développement de la radio émission. Plusieurs systèmes de localisation basés sur les radars et les balises électromagnétiques ont été développés pour les besoins militaires et la navigation maritime et aérienne.

Le grand public n'est pas en reste. Avec l'accroissement du nombre de téléphones portables, des applications créant des tribus et pour lesquelles il est nécessaire de connaître la position des autres membres de la tribu ont été mises en place.

Nous voyons par-là que la localisation peut être nécessaire dans de nombreuses situations de la vie quotidienne, tant sur le plan personnel que professionnel. Aujourd'hui certains services de localisation sont disponibles. Le GPS a été mis en place depuis 1978, des applications ont alors trouvé une réponse adéquate à leurs besoins de localisation. Certaines entreprises proposent à leurs clients des services de localisation à partir des techniques et technologies existantes.

## **6.2. Le principe de géolocalisation**

La géolocalisation est un procédé permettant de positionner un objet ou une personne sur un plan ou une carte à l'aide de ses coordonnées géographiques. Cette opération est réalisée à l'aide d'un terminal capable d'être localisé (grâce à un récepteur GPS ou à d'autres techniques) et de publier (en temps réel ou de façon différée) ses coordonnées géographiques (latitude/longitude).

Les positions enregistrées peuvent être stockées au sein d'un terminal et être extraites postérieurement, ou être transmises en temps réel vers une plateforme logicielle de géolocalisation. Ceci permet de visualiser la position du terminal sur une carte à travers une plateforme de géolocalisation le plus souvent accessible depuis Internet.

Le réseau satellitaire de positionnement le plus connu est le GPS. Dans le cas du GPS, pour que le repérage spatial fonctionne, un immense réseau constitué de 27 satellites (dont 3 de secours) qui tournent autour de la Terre (2 tours en 24 heures) à une altitude de 20 200 km et répartis sur 6 orbites (4 par orbite) différentes est nécessaire. Ces satellites constituent un maillage du ciel et servent de repères aux navigateurs GPS dans leur processus de calcul de 10 positions.

Ce système de satellites est conçu de façon à ce qu'il y en ait toujours au moins quatre visibles par les navigateurs GPS, sans quoi la position ne peut pas être déterminée.

Pour qu'un terminal soit capable de se géo localiser grâce au réseau GPS, il doit absolument être équipé d'une puce électronique GPS.

Les composants essentiels d'une plateforme de géolocalisation sont les suivants :

- Terminal communicant : C'est le terminal qui reçoit les coordonnées géographiques (via GPS ou tout autre moyen) et qui les envoie via un réseau de télécommunications à la plateforme.
- Système informatique capable de recevoir, stocker et traiter les informations : il s'agit des serveurs informatiques qui hébergent l'infrastructure et qui reçoivent et traitent les données envoyées par les terminaux.
- Module cartographique : c'est le module intégré au système informatique qui va permettre d'afficher la position des terminaux sur un fond cartographique adapté. Ce module prend en charge les calculs de distances, d'itinéraires, détecte l'interaction avec les zones et permet d'avoir accès à des informations terrain.

En effet, la donnée (position) générée par un terminal, qui se trouve sur le terrain, doit être transmise à une plateforme logicielle qui va la traiter, la présenter graphiquement à l'utilisateur et l'associer à d'autres données afin d'enrichir les informations relatives à l'état du terminal ou de la flotte de terminaux.

### **6.3. Technique de la géolocalisation**

Cette technologie est réalisée grâce à un récepteur qui reçoit les ondes émises par un appareil et déduit les coordonnées géographiques.

On cite ci-après les quatre types de géolocalisation.

- **Par satellite**

La technique de la géolocalisation par satellite est la technologie la plus répandue actuellement, elle permet à un équipement de calculer sa position en utilisant les signaux émis aux différents satellites. Pour calculer sa position, une puce GPS doit mesurer sa distance à l'aide de trois satellites. La position du récepteur est alors traduite en termes de latitude, longitude et peut être représentée sur une carte. Ce système est assez précis (une dizaine de mètres) mais les puces présentes sur les Smartphones sont très petites et ne reçoivent donc qu'un signal provenant des satellites très atténué, ce qui fait que la géolocalisation par satellite n'est possible qu'en extérieur et elle est assez lente.

- **Par wifi**

La géolocalisation utilisant les points d'accès Wifi est une technologie beaucoup plus récente, particulièrement adaptée à la ville. Pour se localiser, un Smartphone commence par dresser la liste de tous les BSSID présentées autour de l'appareil (les BSSID sont les adresses MAC des cartes réseaux wifi des points d'accès). Pour permettre cette géolocalisation à l'aide de points d'accès wifi, il faut auparavant établir la liste la plus complète possible des bornes wifi avec leurs coordonnées géographiques. Afin de créer ces derniers ou bien les enrichir, il est possible d'utiliser des données envoyées par les téléphones mobiles soit lorsqu'ils essayent de se localiser, soit en envoyant régulièrement les points d'accès qu'ils ont rencontrés.

- **Par GSM**

La géolocalisation par GSM a été utilisée avant la démocratisation des puces GPS dans les Smartphones. Cette technique permet le positionnement d'un terminal par rapport aux informations relatives aux antennes GSM aux quelles le terminal est connecté. La précision du positionnement par GSM peut aller de deux cents mètres à plusieurs kilomètres, selon la densité d'antennes, supérieure en ville qu'en milieu rural.

- **Par adresse IP (sur internet)**

En se basant sur l'adresse IP d'un terminal connecté à internet, la position géographique de ce dernier peut être déterminée. La géolocalisation par adresse IP permet de trouver la position géographique d'un ordinateur ou tout autre terminal connecté à internet grâce à son adresse IP. La précision des informations est en fonction du pays, voire de la ville. Leur géolocalisation est plutôt du point de vue sécurité.

## **6.4. Comparaison entre les techniques de géolocalisation**

Le choix d'une technique de géo localisation dépend certainement de la situation, les conditions et l'intérêt d'usage pour le client cible. Donc, chaque service exige un choix bien déterminé. Afin de classifier les catégories, le tableau ci-dessous représente une comparaison entre les différentes techniques de géolocalisation en se basant sur les termes vitesse, précision et le mode de fonctionnement :

Technologie	GSM	Wifi	GPS
Vitesse	Très rapide	Rapide	Lent
Précision	100m-10km	10m-50m	10m-100m
Fonction	Mieux en ville	Ville	Mieux hors ville

**Tableau 3: Comparaison entre les techniques de géolocalisation.**

## **6.5.Les outils et techniques de géolocalisation**

### **6.5.1. Google Maps**

Google Maps est un service gratuit de cartographie en ligne qui a été créé par Google. Ce service permet, à partir de l'échelle d'un pays, de pouvoir zoomer jusqu'à l'échelle d'une rue. Deux types de vue sont disponibles : une vue en plan classique, avec nom des rues, quartier, villes et une vue en image satellite, qui couvre aujourd'hui le monde entier. Pour intégrer ces cartes interactives Google Maps à sa propre application et bénéficier des données associées, l'utilisateur doit disposer d'une clé (Google Maps API Key) propre à son domaine d'utilisation.

### **6.5.2. Le système GPS**

C'est un système de positionnement par satellite, il permet à l'utilisateur de connaître sa position, planifier son itinéraire, estimer son temps du trajet, connaître la localisation des radars, etc. Profitant de cette technologie, Google a créé Google Maps Navigation. Il s'agit d'un service de guidage par GPS ou encore un système de navigation GPS, gratuit disponible sur les téléphones.

## **Conclusion**

Dans ce chapitre j'ai présenté le cadre général de mon projet en déterminant la problématique et en proposant la solution envisagée pour faire face à la situation courante. J'ai dévoilé le langage et la méthodologie de conception qui seront utilisés dans les prochains chapitres de ce rapport que j'ai argumenté.



---

## *Chapitre II : Planification et architecture*

---

### **Introduction :**

Les travaux réalisés dans cette période de temps conduit à construire une bonne vision du produit, identifier les rôles des utilisateurs et dégager les fonctionnalités principales afin de produire le back log initial ainsi qu'une première planification des sprints.

### **1. Analyse des besoins :**

Dans cette partie, je présenterai les besoins fonctionnels et non-fonctionnels identifiés après la sélection des besoins.

#### **1.1. Identification des acteurs :**

Un acteur est l'idéalisation d'un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système. Dans le cas de mon projet je présenterai les acteurs suivant :

**Le passager** : C'est l'utilisateur qui aura la possibilité de demander un taxi.

**Le conducteur** : Cet utilisateur gère les demandes provenant du client en acceptant une demande reçue.

**L'administrateur** : Il a comme mission principale de contrôler les données du client et de vérifier les pièces justificatives des conducteurs pour une meilleure sécurité de l'application.

### ▪ Diagramme de contexte statique

Ce diagramme d'UML permet simplement de montrer la relation des différents acteurs avec le système. Il spécifie le nombre d'instances de chaque acteur relié au système à un moment donné.

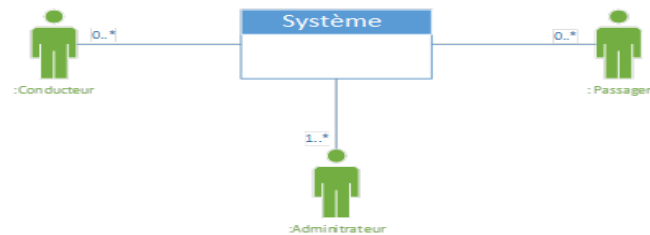


Figure 8: Diagramme de contexte statique

## 1.2. Les besoins fonctionnels :

Au cours de cette étape, je vais extraire les différentes fonctionnalités qu'offrira mon projet :

- L'application permet au *conducteur* de :
  - S'authentifier :
    - Se connecter à travers un login et un mot de passe.
  - Gérer la réservation :
    - Répondre aux demandes des passagers.
    - Consulter le profil du client.
  - Gérer l'évaluation :
    - Signaler le passager.
    - Noter le passager.
  - Gérer le statut :
    - Choisir le mode disponible.
    - Choisir le mode occupé.
    - Choisir le mode hors service.
  - Gérer le contact :
    - Effectuer un appel téléphonique au client.

- **L'application permet au **client** de :**

- S'authentifier :
  - Se connecter via Facebook.
- Gérer la demande :
  - Sélectionner un Taxi sur la carte géographique.
  - Envoyer une demande à un Taxi sélectionné.
  - Exécuter une demande rapide.
  - Connaitre les différents états des Taxis.
  - Connaitre le temps, distance et cout estimé.
  - Consulter le profil du conducteur.
  - Annuler la demande
- Gérer le contact :
  - Effectuer un appel téléphonique au conducteur.
- Gérer la rencontre :
  - Préciser le repère de rencontre sur la carte géographique.
  - Préciser une référence lors d'une demande de Taxi.
  - Préciser la cause d'annulation
- Gérer l'évaluation :
  - Signaler le conducteur.
  - Noter le conducteur.

- **L'application permet à l'**administrateur** de :**

- S'authentifier.
- Gérer le compte utilisateur :
  - Consulter, supprimer, bannir.
  - Valider : vérifier les pièces justificatives des conducteurs (accepter ou décliner).
- Gérer compte administrateur.
  - Ajouter, supprimer, consulter, modifier les comptes administrateurs.

### **1.3. Les besoins non fonctionnels :**

Il s'agit d'un ensemble de règles à respecter, lors du développement pour s'assurer de la bonne utilisation de l'application.

- **L'ergonomie :**

L'application doit être simple et facile à manipuler par l'utilisateur. Le passage entre les interfaces de l'application doit se faire dans des délais prompts. Une alerte prévient l'utilisateur, chaque fois qu'il commet une erreur d'utilisation.

- **La Fiabilité :**

Touche à l'aspect qualité des données et persistance des informations dans l'application ainsi que la vitesse de chargement des interfaces. D'une part, les interfaces de l'application doivent s'adapter à la taille des différents écrans des appareils mobiles. Elle devra aussi fonctionner correctement pour les téléphones Android et OS avec les versions les plus récentes. D'autre part, elle doit fonctionner d'une façon cohérente sans erreur.

- **L'Efficacité :**

L'efficacité de mon application doit permettre l'accomplissement de la tâche avec le minimum de manipulation. Ceci doit être garanti pour que mon application puisse s'intégrer facilement dans le marché des applications mobiles.

- **La Sécurité :**

Les différents comptes utilisés par le passager et le conducteur doivent être sécurisés et vérifiés pour éviter les faux comptes et les fausses informations.

## **2. Planning du traitement des cas d'utilisation :**

Après l'identification des cas d'utilisation, je dois maintenant les classifier. La classification des cas d'utilisation doit tenir compte d'un facteur principal qui est la priorité. Cette technique est utilisée généralement lors de la conception des applications se basant sur le processus unifié, mais elle reste valable et intéressante pour mon cas.

## 2.1. Priorités :

Généralement, on dit qu'un cas d'utilisation A est plus prioritaire que B, si sa réalisation accélère la stabilisation du système. Le choix des priorités dans cette section s'est basé sur la dépendance entre les fonctionnalités de l'application. Par exemple, nous ne pouvons pas effectuer une réservation sans avoir demandé un conducteur, tant que nous n'avons pas encore terminé la pré-demande. Par conséquent, nous pouvons dégager trois niveaux de priorité qui sont : priorité haute, moyenne et faible.

## 2.2. Risque :

Lors du pilotage d'un projet, l'identification des risques critiques présente une étape indispensable pour la réussite de ce dernier. Pour mon cas, le seul risque qui peut me ralentir est lié à la complexité de l'application et aux différentes contraintes à respecter.

## 3. Prototypage des interfaces :

Ci-dessous je présente le prototype de l'interface avec les étapes nécessaires pour accéder à l'application et effectuer une réservation.



Figure 9: Interface authentication passager.



Figure11: Interface authentication Chauffeur.



Figure10: Interface demander taxi.



Figure 12:Interface l'acceptation d'une demande.

#### 4. Le back log du produit :

Le back log du produit est l'artefact le plus important de Scrum, c'est l'ensemble des caractéristiques fonctionnelles ou techniques qui constituent le produit souhaité. Les caractéristiques fonctionnelles sont appelées des « histoires utilisateur » (user story) et les caractéristiques techniques sont appelées des « histoires techniques » (technical story).

Le tableau ci-dessous résume le back log produit de notre application. Il est à noter que nous n'avons pas cité les histoires techniques comme la préparation de la maquette graphique, les travaux de conception et les jeux de tests, etc.

Dans le tableau suivant, chaque histoire utilisateur est caractérisée par sa priorité expliquée dans la section 2 de ce même chapitre. Pour le traitement de nos « histoires utilisateur » nous faisons le choix de commencer avec les cas d'utilisation les plus prioritaires et plus importante.

<b>Id</b>	<b>Nom</b>	<b>En tant que</b>	<b>Je veux qu'</b>	<b>Pour</b>	<b>Priorité</b>
<b>1</b>	Authentification	Passager	Il soit possible de se connecter à l'application grâce à un compte Facebook.	Avoir accès à l'application en mode passager.	Elevée
		Conducteur	Il soit possible de se connecter à l'application grâce à un login et mot de passe.	Avoir accès à l'application en mode conducteur.	Elevée
<b>2</b>	Gérer la demande	Passager	Il soit possible de Sélectionner un taxi sur la carte géographique	Consulter les informations du conducteur. (Profil)	Elevée
			Il soit possible de Sélectionner un taxi sur la carte géographique.	Connaitre le temps, distance et coût estimé.	Elevée
			Il soit possible d'envoyer une demande à un taxi sélectionné.	Effectuer une réservation.	Elevée
<b>3</b>	Gérer la rencontre	Passager	Il soit possible de préciser le repère de rencontre sur la carte géographique.	Faciliter au conducteur la localisation de son passager.	Moyenne
			Il soit possible de préciser une référence sous forme d'un message personnalisé.	Pour reconnaître son passager lors de la rencontre.	Moyenne
<b>4</b>	Gérer la demande	Passager	Il soit possible d'exécuter une demande rapide.	Minimiser le temps de la recherche d'un taxi.	
<b>5</b>	Gérer le contact	Passager/ Conducteur	Il soit possible d'effectuer un appel téléphonique.	-----	Faible
<b>6</b>	Gérer la rencontre	Passager	Il soit possible d'annuler la demande.	-----	Faible



<b>Id</b>	<b>Nom</b>	<b>En tant que</b>	<b>Je veux qu'</b>	<b>Pour</b>	<b>Priorité</b>
<b>7</b>	Gérer la demande	Passager	Il soit possible de saisir la cause d'annulation de la demande.	Laisser une trace aux administrateurs qui supervisent l'application.	Faible
<b>8</b>	Gérer la réservation	Conducteur	Il soit possible de répondre aux demandes des passagers par acceptation ou refus.	Choisir le passager qui lui convient.	Elevée
			Il soit possible de consulter les informations du passager et de la demande.	Trouver et connaître son passager facilement.	Elevée
<b>9</b>	Gérer le statut	Conducteur	Il soit possible de choisir son mode convenable (disponible, occupé ou hors service)	Eviter les demandes inutiles, et consulter les différents états des autres collègues.	Elevée
<b>10</b>	Gérer l'évaluation	Passager/ Conducteur	Il soit possible de signaler et noter le passager/conducteur.	Améliorer le contrôle des utilisateurs et laisser des traces.	Faible
<b>11</b>	Authentification	Administrateur	Il soit possible de se connecter à l'application grâce à un login et mot de passe.	Avoir l'accès à l'application et contrôler les utilisateurs.	Elevée
<b>12</b>	Gérer compte utilisateur	Administrateur	Il soit possible de gérer les comptes des utilisateurs (valider les comptes des nouveaux conducteurs)	Pouvoir les contrôler (valider, consulter, supprimer, bannir)	Elevée
<b>13</b>	Gérer compte administrateur	Super Administrateur	Il soit possible de gérer les comptes (supprimer, ajouter, modifier)	-----	Faible
			Il soit possible de consulter tous les autres administrateurs.	-----	Faible
<b>14</b>	Gérer Statistiques	Administrateur	Il soit possible à l'administrateur de consulter les statistiques.	-----	Faible

Tableau 4:Backlog produit.

#### 4.1. Planification des sprints :

La planification des sprints est ainsi schématisée dans le graphique suivant :

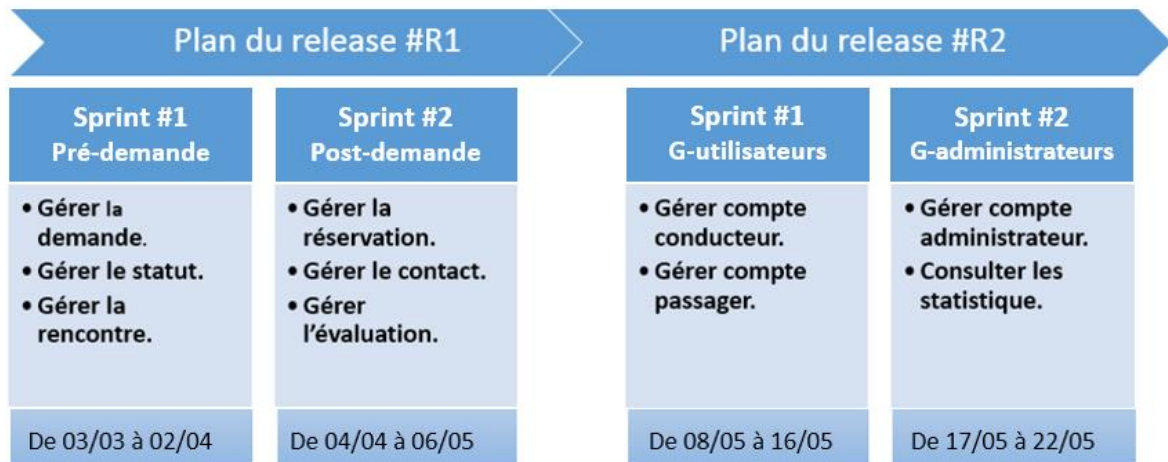


Figure 13: Planification des sprints.

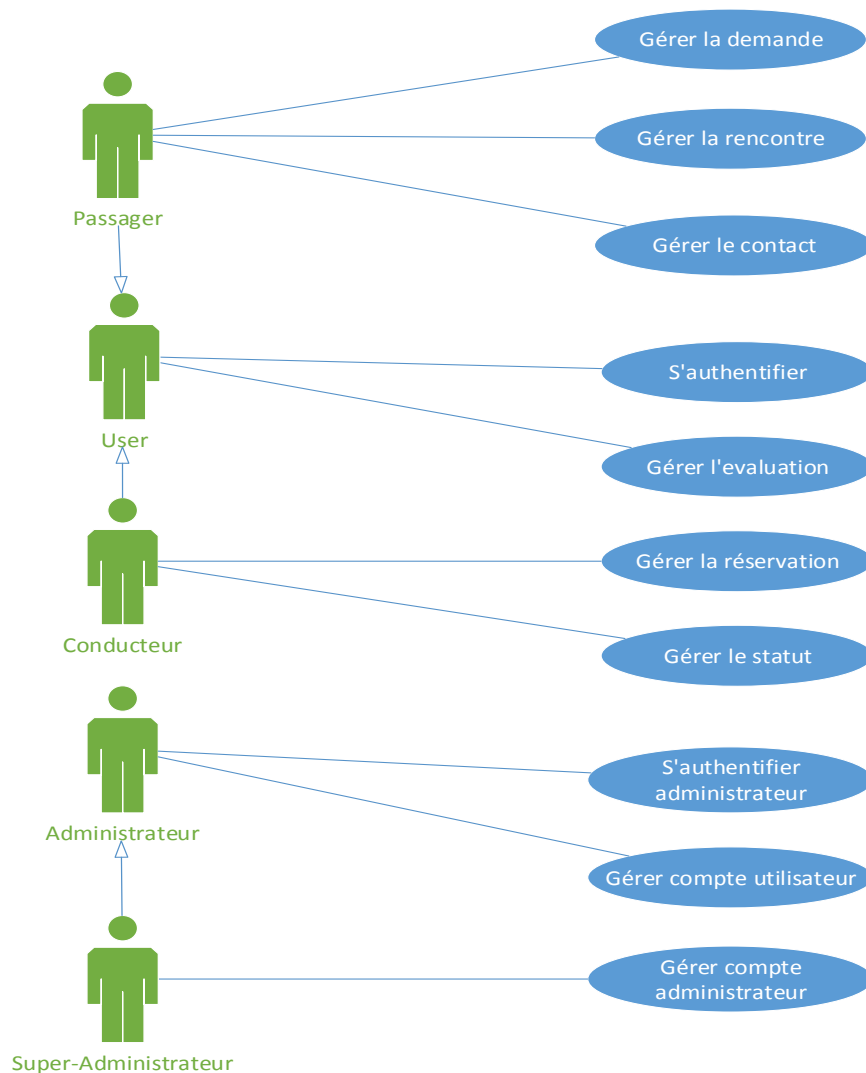


Figure 14: Diagramme de cas d'utilisation global

## Conclusion

Dans ce chapitre j'ai préparé mon plan de travail. J'ai capturé les besoins fonctionnels de mon application, les rôles des utilisateurs, par la suite j'ai préparé l'architecture logique ainsi que le plan de release de mon projet.

---

---

# Release 1 : Gestion des demandes et réservations

---

---

## Introduction :

Le terme release peut-être défini comme une version distribuée d'une application ou une période de temps qui permet de la produire. Peu importe quelle définition j'utilise, une release est constituée d'une suite d'itérations (sprint) qui se terminent quand les incréments de ces derniers construisent un produit présentant suffisamment de valeur aux utilisateurs finaux.

Tout au long de ce chapitre, je vais traiter les histoires utilisateurs de mes sprints pour produire un incrément potentiellement livrable.

## **I. Premier Sprint :**

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

Avant de se lancer dans un sprint, l'équipe Scrum doit obligatoirement définir le but de ce dernier. Il s'agit de répondre à une question fondamentale « pourquoi faisons-nous ce sprint ? ». Suite à une étude sur la division du projet, j'ai décidé le but suivant : « Préparer les fonctionnalités qui sont liées à la pré-réservation. ».

Une fois, nous avons défini le but de mon sprint, il est temps de décider quelles sont les fonctionnalités et leurs degrés d'importance incluent dans ce dernier. Plus précisément, quels noms de mon back log du produit seront inclus dans le back log du sprint. Ce qui résume donc le Tableau 1 ci-dessous, back log de mon premier sprint :

<b>Histoire utilisateur</b>	<b>Priorité</b>
Préciser le repère de rencontre sur la carte géographique.	Faible
Préciser une référence.	Faible
Exécuter une demande rapide.	Elevée
Consulter les informations d'un taxi sélectionné (Distance, Coût, Durée).	Moyenne
Consulter le profil du conducteur.	Moyenne
Envoyer une demande à un taxi sélectionné.	Elevée
Annuler la demande.	Faible
Gérer Statut.	Moyenne

**Tableau 5 : Backlog du premier sprint (release1).**

### **1. Spécification fonctionnelle :**

La spécification fonctionnelle dans mon cas se traduit par le diagramme des cas d'utilisation d'UML et la description textuelle de ces derniers.

#### **1.1. Scénario et cas d'utilisation**

Après avoir effectué mon Backlog, je peux maintenant l'expliquer sous forme de tableaux :

Chaque tableau présente la description détaillée du fonctionnement de tous les cas d'utilisation. Dans ce Sprint, je vais m'intéresser aux 3 cas d'utilisations suivants :



Scrum nous permet de ne pas figurer quelques cas d'utilisation dans le Backlog du sprint, comme l'inscription du chauffeur de taxi ainsi que son authentification pour la simple raison de simplification. En effet, je m'intéresse aux diagrammes les plus importants du projet. Dans la Figure 18. J'illustre le diagramme des cas d'utilisation raffinés pour ce premier sprint.

### 1.1.1. Description du cas d'utilisation « Gérer la demande » :

#### ➤ Description textuelle du cas d'utilisation « S'authentifier passager »

Cas d'utilisation :	<b>S'authentifier passager</b>
Acteur :	Passager
Pré condition :	- Le passager n'est pas encore authentifié.
Post condition :	- Le passager est connecté.
Description du scénario principal :	<ul style="list-style-type: none"> <li>- Le passager appuie sur le bouton (S'authentifier avec Facebook).</li> <li>- Le système affiche l'interface « Facebook » au client pour saisir ses coordonnées.</li> <li>- Le passager confirme la saisie.</li> <li>- Le système enregistre les données du Facebook de passager dans la table appropriée s'il s'est connecté pour la première fois.</li> <li>- Le passager saisit son numéro de téléphone.</li> <li>- Le système effectue un appel téléphonique au passager pour être sûr du numéro saisi.</li> <li>- Le système affiche « Vous êtes maintenant authentifiées ».</li> <li>- Le système affiche le menu principal de l'application.</li> </ul>
Exception :	- Si le passager ne reçoit aucun appel téléphonique dans la minute qui suit, le système affiche un message d'alerte « Authentification refusée ».

Tableau 6: Description du cas d'utilisation "S'authentifier passager"

➤ **Description textuelle du cas d'utilisation « Exécuter une demande Rapide »**

Cas d'utilisation :	<b>Exécuter une demande rapide</b>
Acteur :	Passager
Pré condition :	<ul style="list-style-type: none"> <li>- Le passager est authentifié.</li> <li>- Le passager n'a pas encore demandé un taxi.</li> </ul>
Post condition :	<ul style="list-style-type: none"> <li>- Le passager a demandé un taxi.</li> </ul>
Description du scénario principal :	<ul style="list-style-type: none"> <li>- Le passager effectue une recherche rapide en cliquant sur le bouton « Demande rapide »</li> <li>- Le système consulte les Taxis qui se trouvent dans un rayon bien déterminé et leur envoie la demande.</li> <li>- Le système informe le passager de l'acceptation de la demande.</li> <li>- Dans le cas d'acceptation : <ul style="list-style-type: none"> <li>• Le système informe le passager en affichant l'itinéraire sur la carte, la distance en kilométrage et le temps nécessaire pour l'arrivée du Taxi.</li> <li>• Le système informe le passager du numéro de Taxi demandé.</li> </ul> </li> <li>- Le système présente les informations du conducteur au passager.</li> </ul>
Exception :	<ul style="list-style-type: none"> <li>- Si la demande dépasse un temps précis, un message d'alerte sera déclenché au client.</li> </ul>

**Tableau 7: Description du cas d'utilisation "Envoyer une demande rapide".**

➤ **Description textuelle du cas d'utilisation « Envoyer une demande à un Taxi sélectionné »**

Cas d'utilisation :	<b>Envoyer une demande à un Taxi sélectionné</b>
Acteur :	Passager
Pré condition :	<ul style="list-style-type: none"> <li>- Le passager est authentifié.</li> <li>- Le passager n'a pas encore demandé un Taxi.</li> </ul>
Post condition :	<ul style="list-style-type: none"> <li>- Le passager a demandé un Taxi.</li> </ul>
Description du scénario principal :	<ul style="list-style-type: none"> <li>- Le passager sélectionne un Taxi sur la carte géographique et clique sur le bouton « Demander ».</li> <li>- Le système envoie la demande au conducteur de Taxi.</li> <li>- Le système retourne la réponse de la demande.</li> <li>- Dans le cas d'acceptation : <ul style="list-style-type: none"> <li>• Le système informe le passager en affichant l'itinéraire sur la carte, la distance en kilométrage et le temps nécessaire pour l'arrivée du Taxi.</li> <li>• Le système informe le passager du numéro de Taxi demandé.</li> </ul> </li> <li>- Le système présente les informations du conducteur au passager.</li> </ul>
Exception :	<ul style="list-style-type: none"> <li>- Si la demande dépasse un temps précis, un message d'alerte sera déclenché au client.</li> </ul>

**Tableau 8: Description du cas d'utilisation "Envoyer une demande à un Taxi sélectionné".**

➤ **Description textuelle du cas d'utilisation « Consulter le profil du conducteur »**

Cas d'utilisation :	<b>Consulter le profil du conducteur</b>
Acteur :	Passager
Pré condition :	- Le passager est authentifié.
Post condition :	- Le passager a consulté le profil.
Description du scénario principal :	<ul style="list-style-type: none"> <li>- Le passager sélectionne un Taxi sur la carte géographique et clique sur le bouton « Profil ».</li> <li>- Le système charge les informations du conducteur de Taxi.</li> <li>- Le système affiche les informations.</li> </ul>
Exception :	<ul style="list-style-type: none"> <li>- Le passager clique sur le bouton « back » pour retourner vers le menu principal.</li> <li>- Le système annule le chargement des informations.</li> <li>- Le système affiche l'interface approprié.</li> </ul>

**Tableau 9: Description du cas d'utilisation "Consulter le profil du conducteur".**

➤ **Description textuelle du cas d'utilisation « Consulter les informations d'un Taxi sélectionné »**

Cas d'utilisation :	<b>Consulter informations d'un Taxi sélectionné.</b>
Acteur :	Passager
Pré condition :	- Le passager est authentifié.
Post condition :	- Le passager a consulté les informations d'un Taxi sur la carte géographique.
Description du scénario principal :	<ul style="list-style-type: none"> <li>- Le passager sélectionne un Taxi sur la carte géographique.</li> <li>- Le système charge les informations (distance, cout estimé, temps, note).</li> <li>- Le système affiche les informations.</li> </ul>
Exception :	NEANT

**Tableau 11: Description du cas d'utilisation "Consulter les informations d'un**

➤ **Description textuelle du cas d'utilisation « Annuler la demande »**

Cas d'utilisation :	<b>Annuler la demande</b>
Acteur :	Passager
Pré condition :	<ul style="list-style-type: none"> <li>- Le passager est authentifié.</li> <li>- Le passager n'a pas encore annulé la demande.</li> </ul>
Post condition :	<ul style="list-style-type: none"> <li>- La demande est annulée</li> </ul>
Description du scénario principal :	<ul style="list-style-type: none"> <li>- Le passager clique sur le bouton « Annuler ».</li> <li>- Le système affiche un message de confirmation avec des multichoix de la cause d'annulation.</li> <li>- Le passager saisit son choix et le confirme.</li> <li>- Le système enregistre le choix dans la base de données.</li> <li>- Le système supprime la demande.</li> </ul>
Exception :	<ul style="list-style-type: none"> <li>- Le passager annule son choix.</li> <li>- Le système annule la suppression et affiche l'interface précédente.</li> </ul>

Tableau 12:Description du cas d'utilisation "Annuler la demande".

**1.1.2. Description du cas d'utilisation « Gérer la rencontre »**

Cas d'utilisation :	<b>Gérer la rencontre</b>
Acteur :	Passager
Pré condition :	<ul style="list-style-type: none"> <li>- Le passager doit être authentifié.</li> </ul>
Post condition :	<ul style="list-style-type: none"> <li>- Le passager a saisi le lieu de rencontre.</li> </ul>
Description du scénario principal :	<ul style="list-style-type: none"> <li>- Le passager fait un clic sur la carte géographique.</li> <li>- Le système affiche le message de confirmation « Êtes-vous sûr de vouloir choisir cette position comme point de rencontre ? ».</li> <li>- Le passager appuie sur « Non » pour annuler l'action.</li> <li>- Le passager appuie sur « Oui » pour valider la saisie.</li> <li>- Le système affiche la position de la rencontre sur la carte géographique sous forme de marker.</li> </ul>
Exception :	Néant

Tableau 13: Tableau 13 : Description du cas d'utilisation "Gérer la rencontre"



### 1.1.3. Description du cas d'utilisation « Gérer statut » :

Cas d'utilisation :	<b>Gérer statut</b>
Acteur :	Conducteur
Pré condition :	- Le conducteur doit être authentifié.
Post condition :	- Le conducteur a changé son mode.
Description du scénario principal :	<ul style="list-style-type: none"> <li>- Le conducteur saisit son login pour accéder à l'application.</li> <li>- Le conducteur a le choix entre 3 modes différents : <ul style="list-style-type: none"> <li>• Mode disponible : Le chauffeur de Taxi est prêt pour recevoir des demandes.</li> <li>• Mode occupé : Le chauffeur de Taxi est en train d'effectuer une course.</li> <li>• Mode Hors-service : Le chauffeur de Taxi ne reçoit aucune demande des passagers. Toutefois il peut avoir une vision sur le trafic de ses collègues.</li> </ul> </li> <li>- Le système enregistre le mode saisi dans la table du conducteur.</li> </ul>
Exception :	NEANT

**Tableau 14: Description du cas d'utilisation "Gérer Statut"**

Le schéma suivant synthétise le cas d'utilisation du sprint 1 :

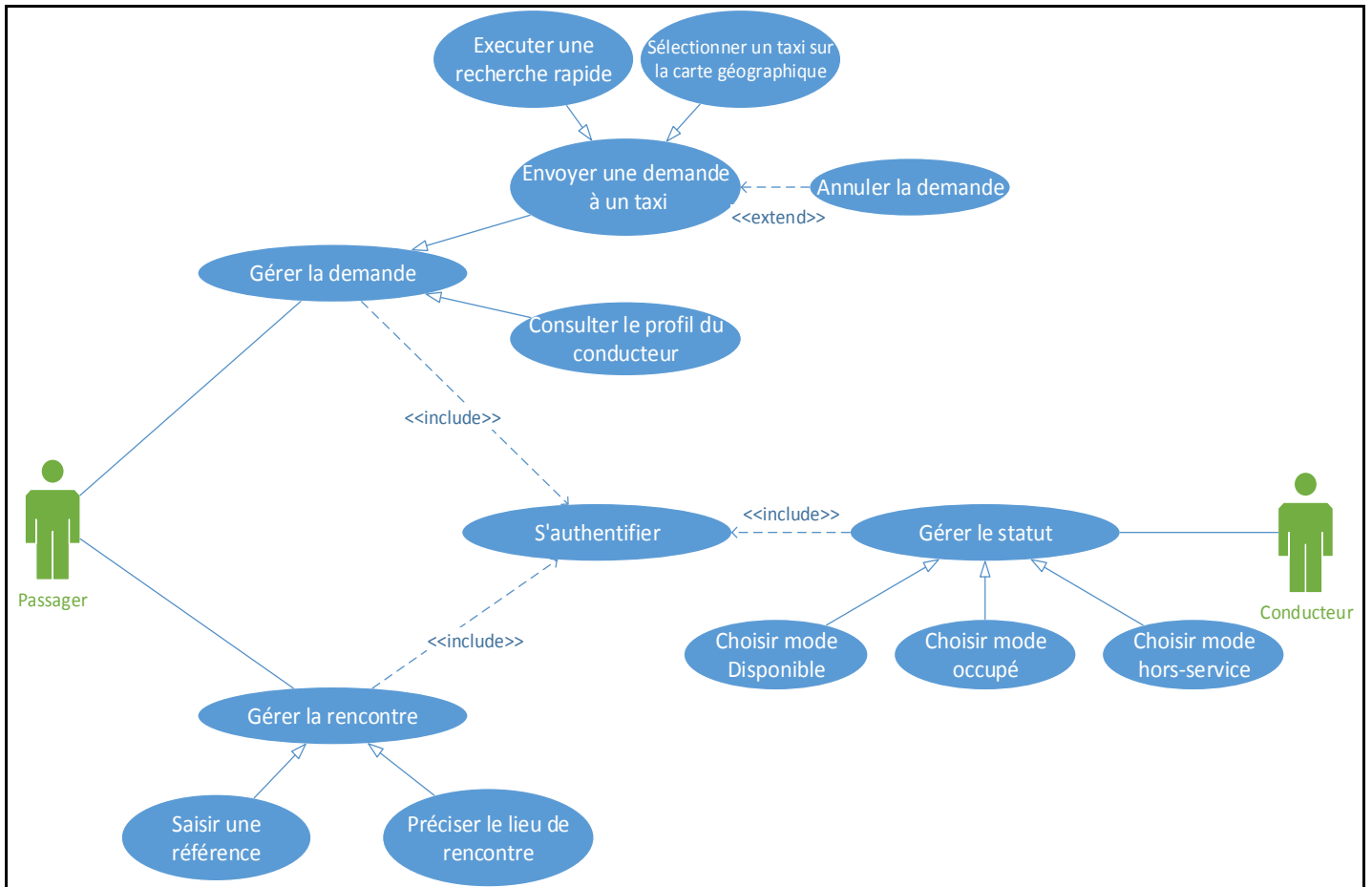


Figure 15: Description des cas d'utilisations du Sprint 1.

## 2. Conception

La conception est la deuxième activité dans un sprint. Elle se traduit par le diagramme de séquence, le diagramme des classes participantes et le diagramme de classe d'UML.

### 2.1. Diagramme de séquence système

Le diagramme de séquence est un diagramme d'objet qui collabore les messages qui s'envoient les uns aux autres présentés de façon chronologique et montrant comment les cas d'utilisation sont réalisés.

Le diagramme de séquence est de plus en plus utilisé, il permet de décrire la dynamique d'un système et faire le lien entre les diagrammes de cas d'utilisation et le diagramme de classe.

Dans ce cas, en nous référant aux descriptions textuelles de la section précédente, nous présentons les diagrammes de séquences système pour connaître les étapes basiques de notre projet.

a. Diagramme de séquence système du cas d'utilisation « Gérer la demande »

Ci-dessous, le schéma représentatif :

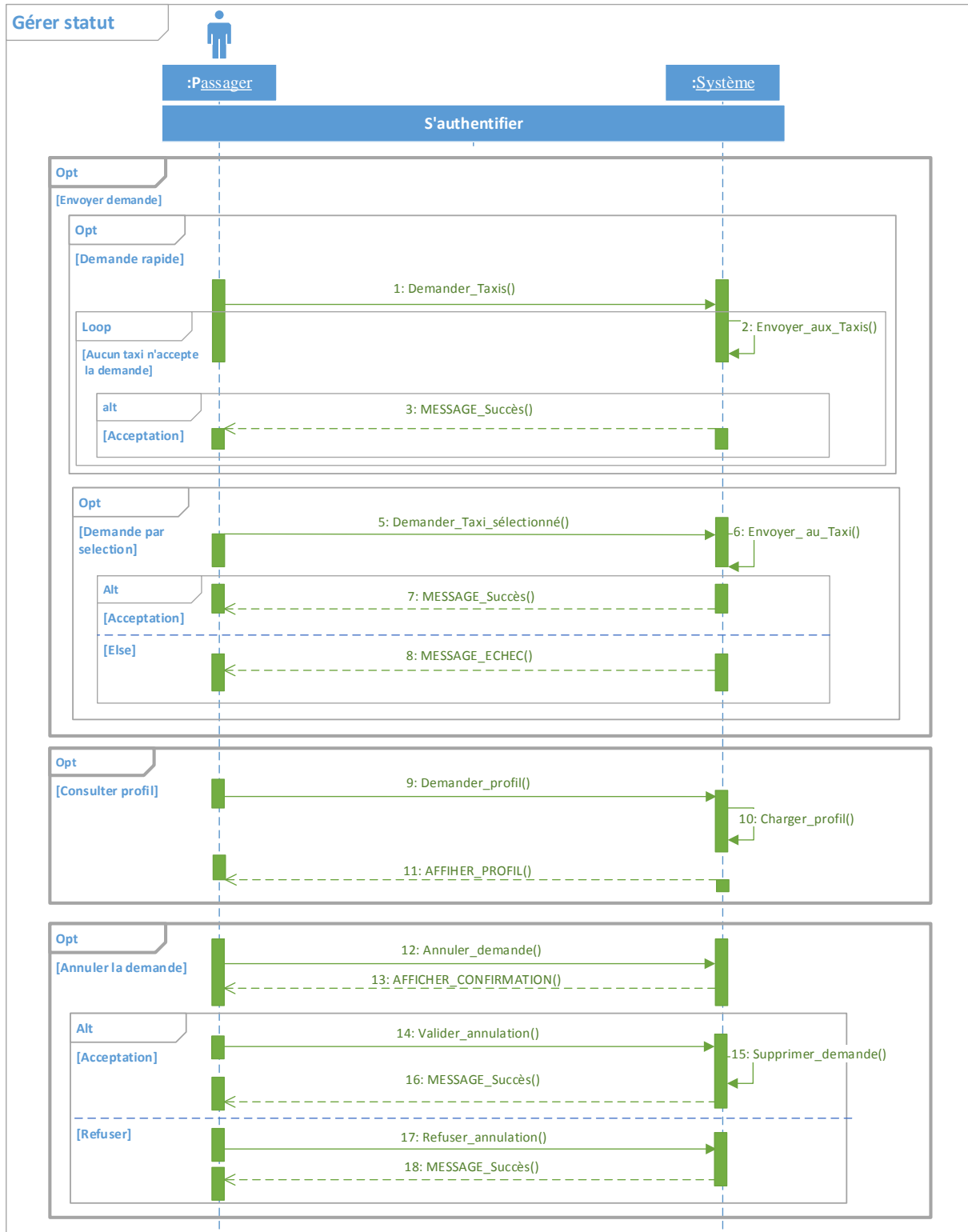
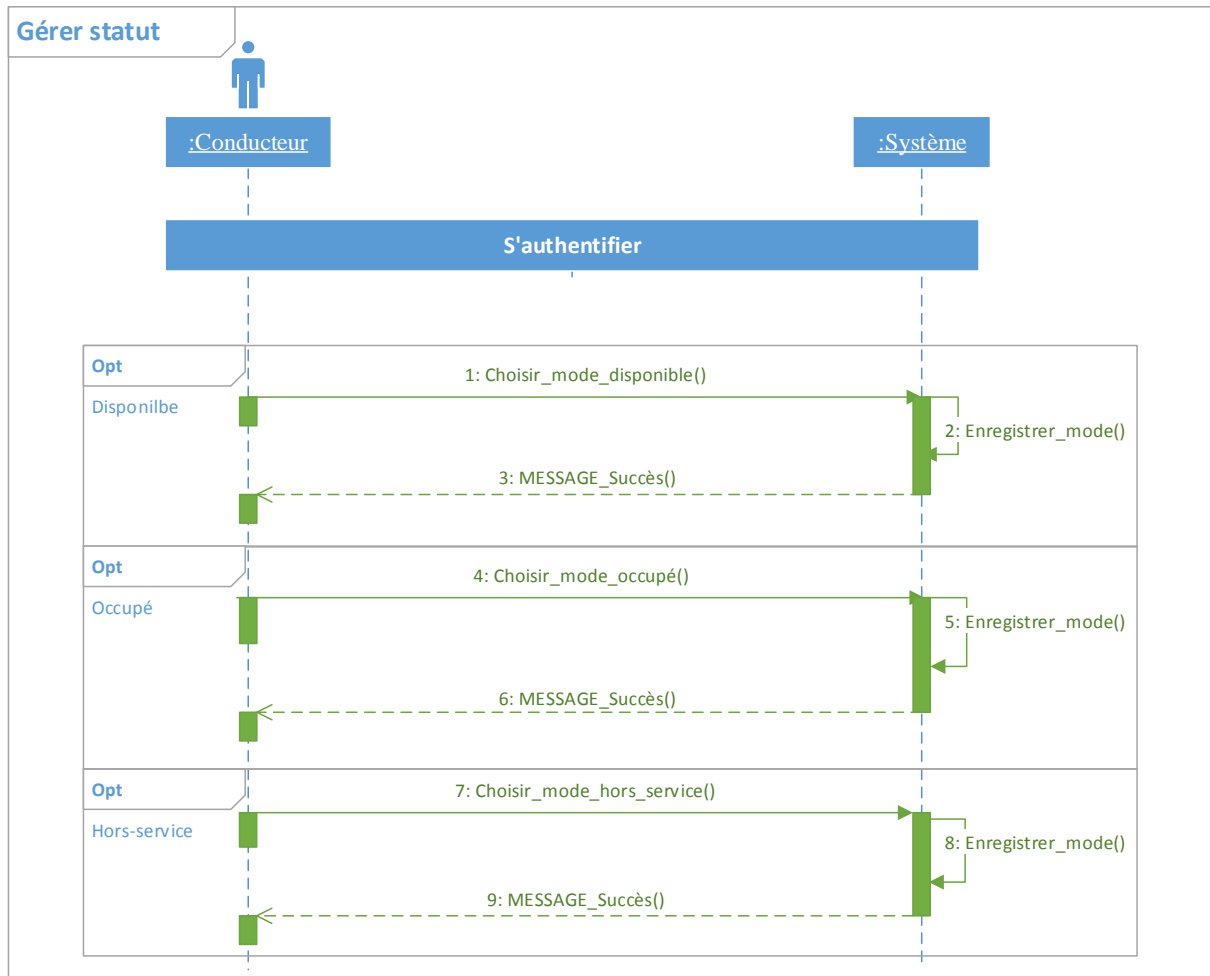


Figure 16:Diagramme de séquence système du cas d'utilisation "Gérer la demande".

**b. Diagramme de séquence système du cas d'utilisation « Gérer statut »**

Il est illustré comme suit :



**Figure 17:Diagramme de séquence système du cas d'utilisation "Gérer statut".**

c. Diagramme de séquence système du cas d'utilisation « Gérer rencontre »

Ci-dessous, le schéma représentatif :

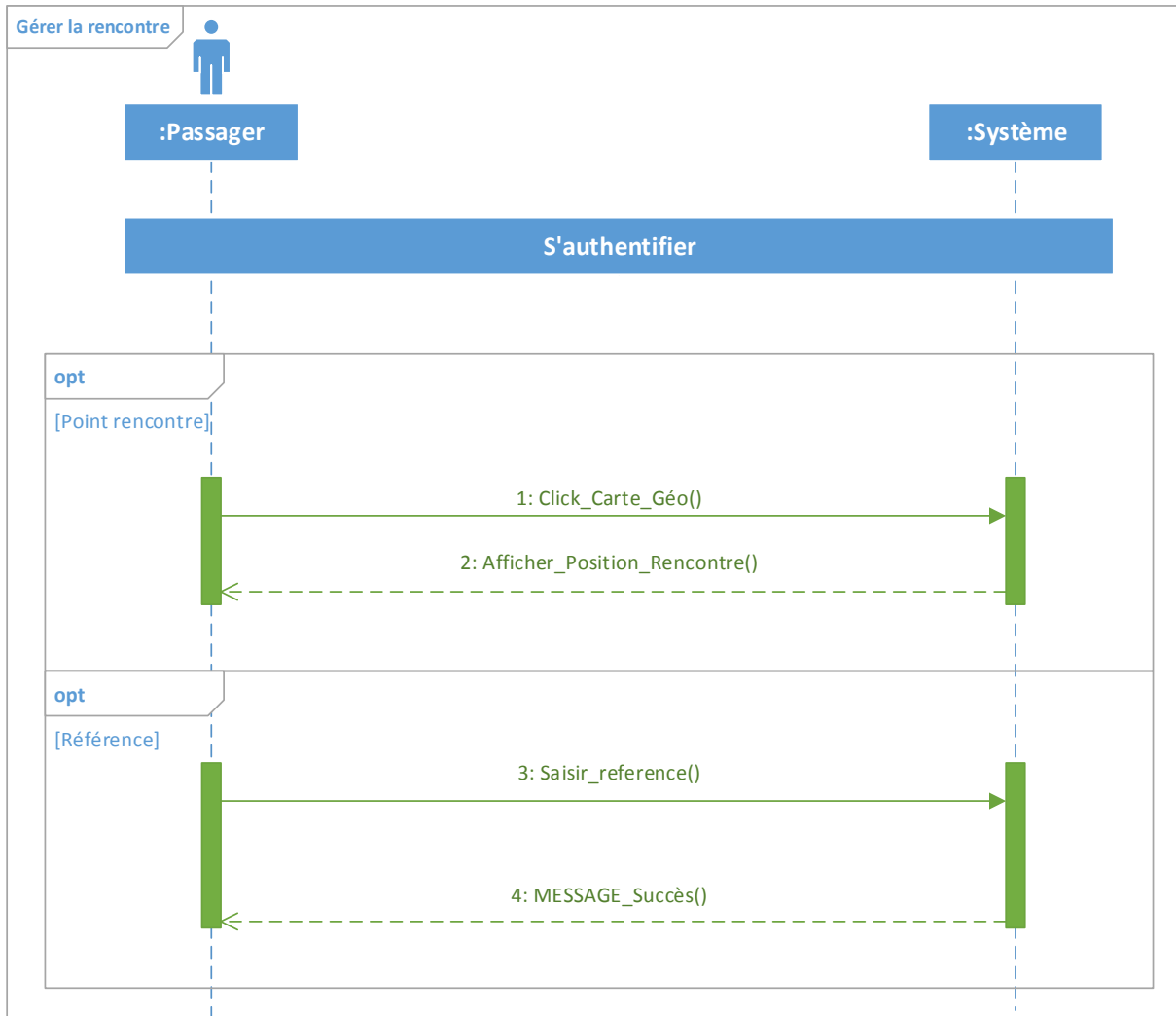


Figure 18: Diagramme de séquence système du cas d'utilisation "Gérer rencontre".

## 2.2. Diagramme des classes des cas d'utilisation

A présent j'effectue le diagramme des classes des cas d'utilisation présentés ci-dessus. Dans un diagramme de classe, je peux distinguer trois stéréotypes<sup>4</sup> de classe qui sont :

Classe interface : représente les interfaces IHM<sup>5</sup> de l'application. Permet aux utilisateurs de communiquer avec le système.

Classe contrôle : représente les méthodes et les fonctions utilisées pour chaque interface.

Pour la construction de mes diagrammes de classes, j'ai choisi d'adopter une certaine règle pour le nommage des éléments :

**Règle 1** : les noms des classes Boundary commence par le préfixe « IU ».

**Règle 2** : les diagrammes seront organisés par les tables cibles et non par fonctionnalités.

Classe entité : ce sont les classes métiers de mon application qui me servent pour la construction du diagramme de classe d'analyse.

---

<sup>4</sup> Stéréotype : famille de modèle en UML

<sup>5</sup> IHM : interface homme-machine

### a. Diagramme de classe du cas d'utilisation « Gérer la demande »

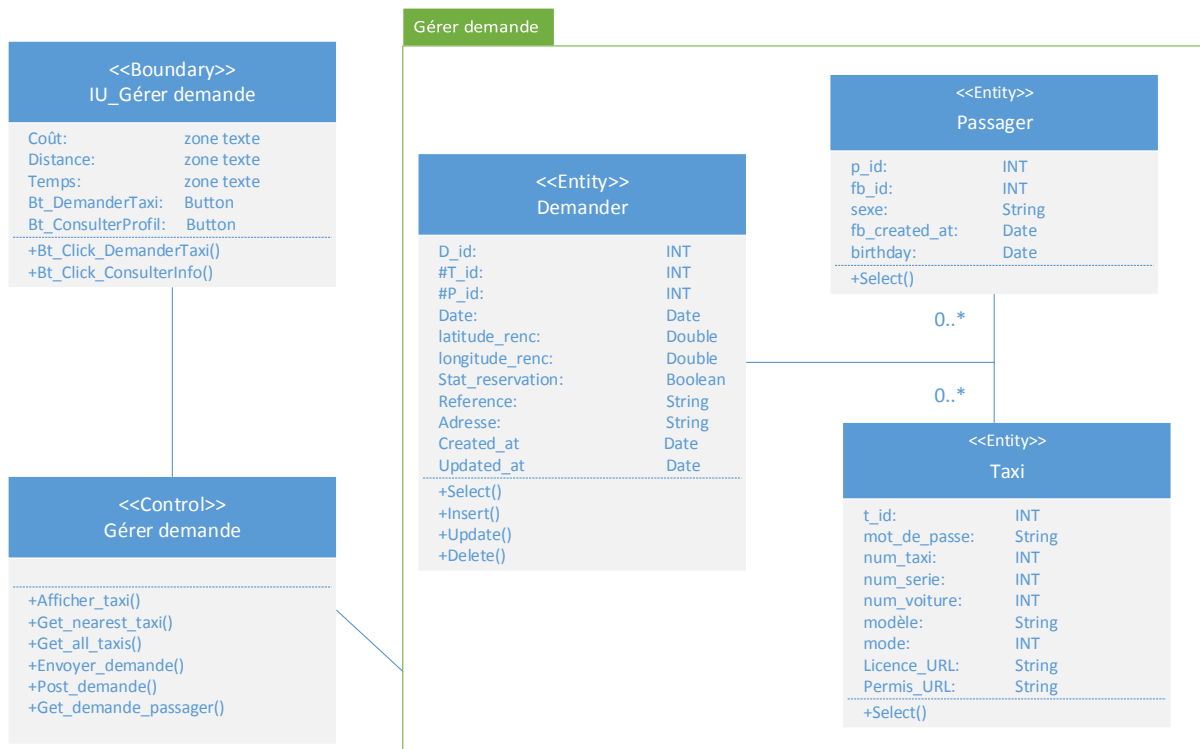


Figure 19: Diagramme de classe de conception du cas d'utilisation "Gérer la demande".



## b. Diagramme de classe du cas d'utilisation « Gérer statut »

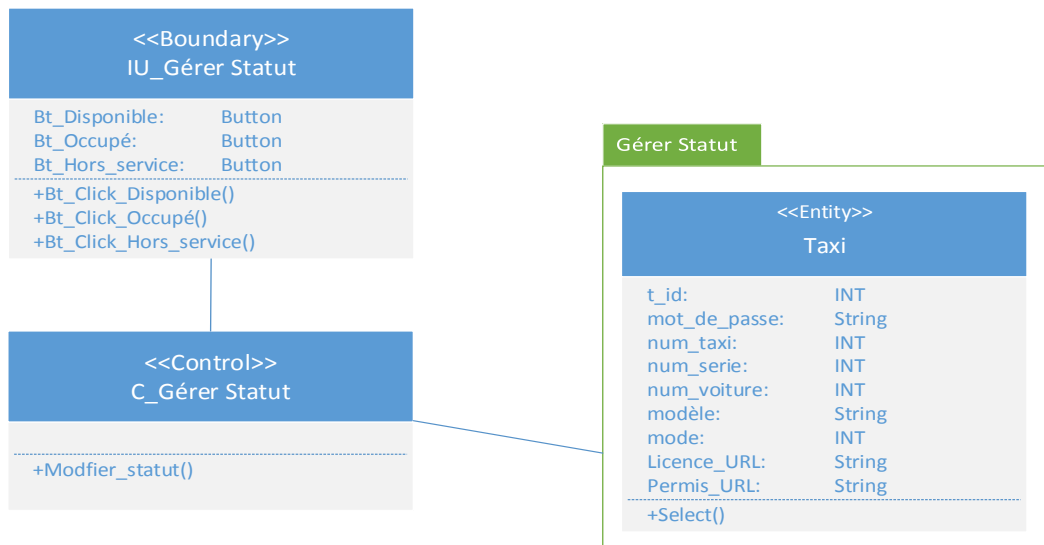


Figure 20:Diagramme de classe de conception du cas d'utilisation "Gérer Statut".

## c. Diagramme de classe du cas d'utilisation « Gérer rencontre »

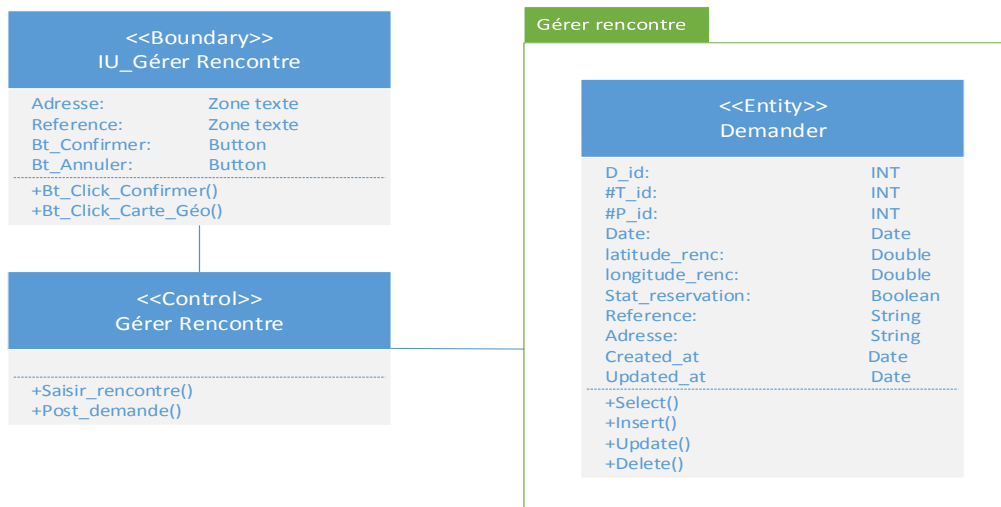


Figure 21:Diagramme de classe de conception du cas d'utilisation "Gérer rencontre".

### 2.3. Diagramme de séquence détaillé :

En plus de l'interaction avec les acteurs, le diagramme de séquence détaillé permet de schématiser la communication entre les différents composants du système. Pour ce qui suit, nous continuons avec les mêmes cas d'utilisation présentés précédemment.

#### a. Diagramme de séquence détaillé du cas d'utilisation « Gérer Demande »

##### ➤ Diagramme de séquence détaillé du sous cas d'utilisation « Demande Rapide »

Lors de l'appuie sur le bouton « Demande rapide », le système consulte la latitude et longitude actuelle du passager et des conducteurs disponibles et les compares pour obtenir les chauffeurs de taxi les plus proches de la position actuelle du client. Si aucun taxi ne répond à la demande, le système recherche de nouveau d'autres taxi plus loin.

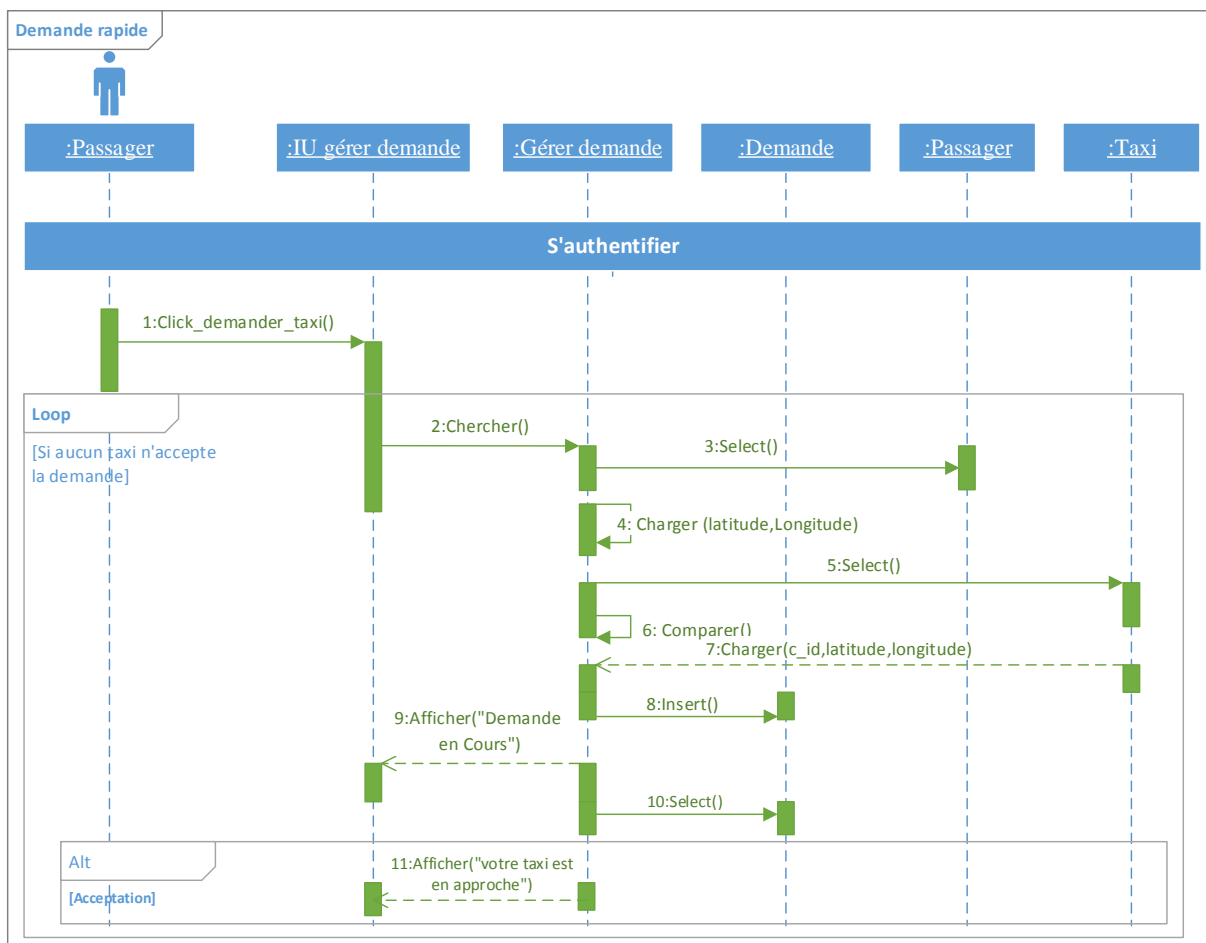
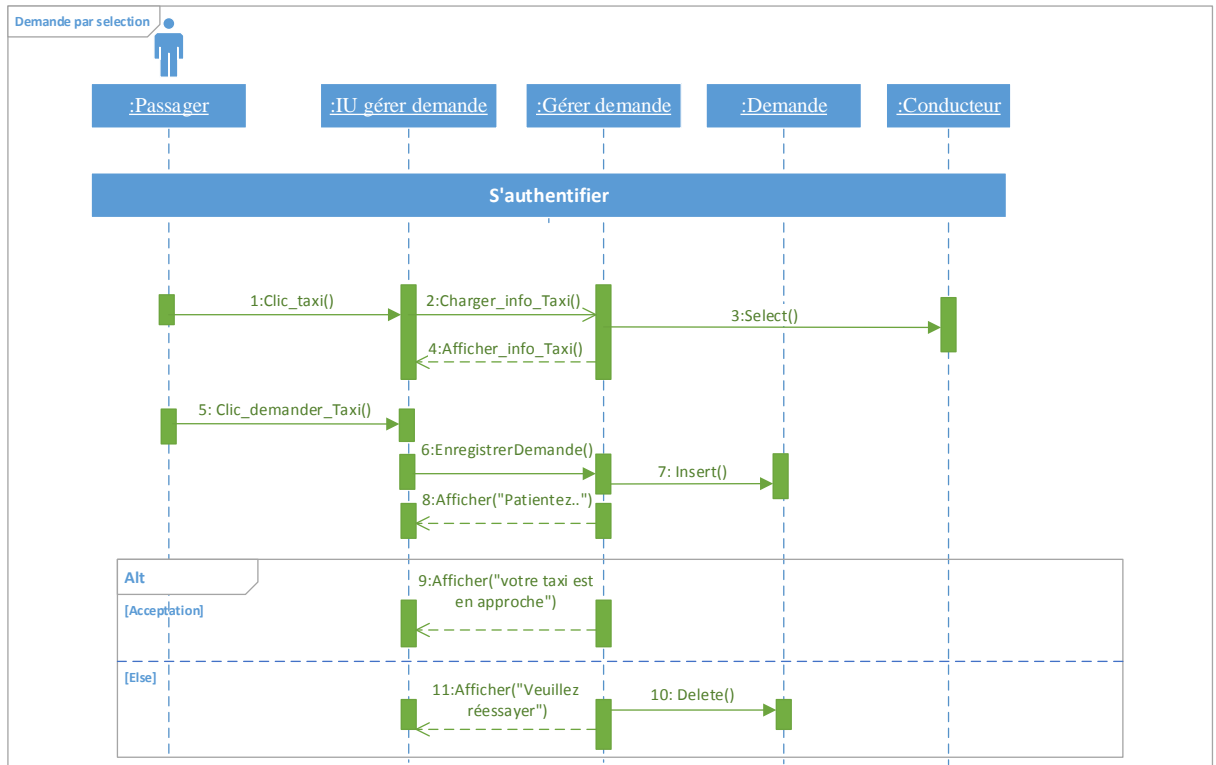


Figure 22: Diagramme de séquence détaillé du sous cas "Demande rapide".

➤ **Diagramme de séquence détaillé du sous cas d'utilisation « Demande par sélection »**



Le passager sélectionne un taxi sur la carte géographique et lui envoie une demande. Si le taxi accepte un autre client ou ne répond pas à la demande, le système avertit le passager du refus de la demande.



**Figure 23: Diagramme de séquence détaillé du sous cas "Demande par sélection".**

### Diagramme de séquence détaillé du sous cas d'utilisation « Consulter profil »

Le passager a le droit de consulter les informations du conducteur après avoir appuyé sur « Consulter le profil ». Le système récupère les informations du chauffeur de taxi et l'affiche au passager.

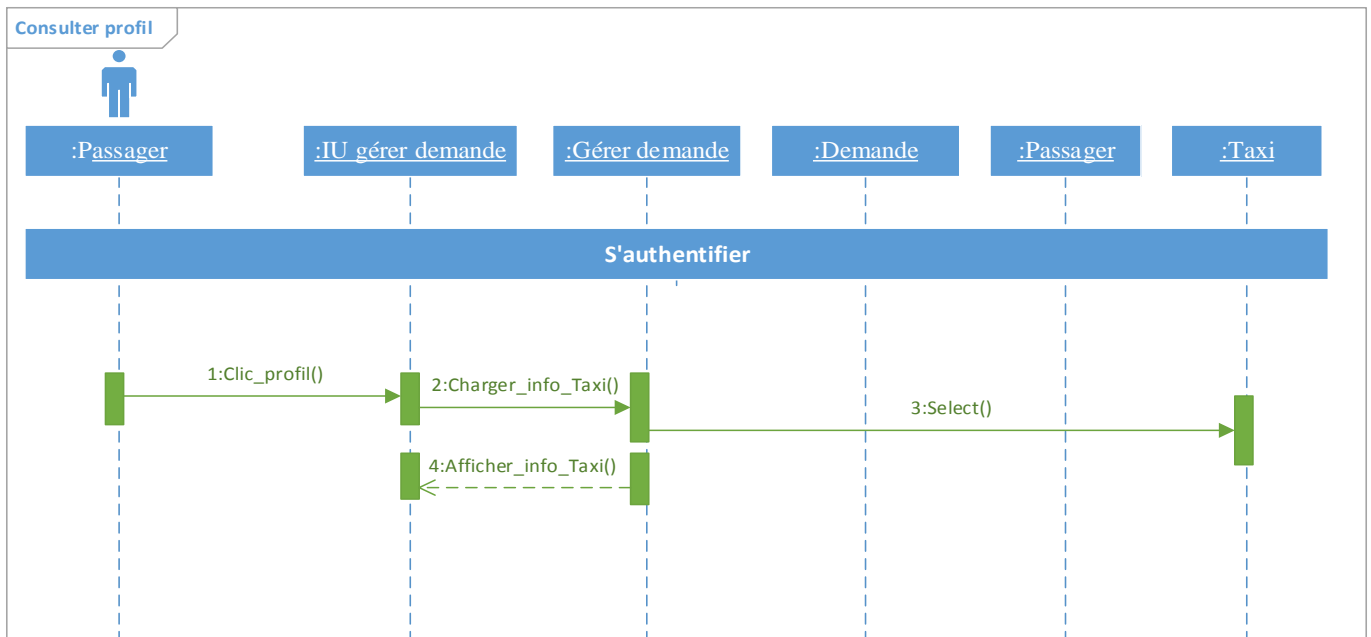


Figure 24: Diagramme de séquence détaillé du sous cas "Consulter profil".

### b. Diagramme de séquence détaillé du cas d'utilisation « Gérer statut »

Lorsque le chauffeur de taxi accède à son application, il sera obligé de saisir son mode pour que les passagers connaissent sa situation actuelle (Disponible, Occupé, Hors-service).

Au moment où le conducteur saisit son mode, il sera enregistré dans la table « taxi » sous le champ « Stat mode ».

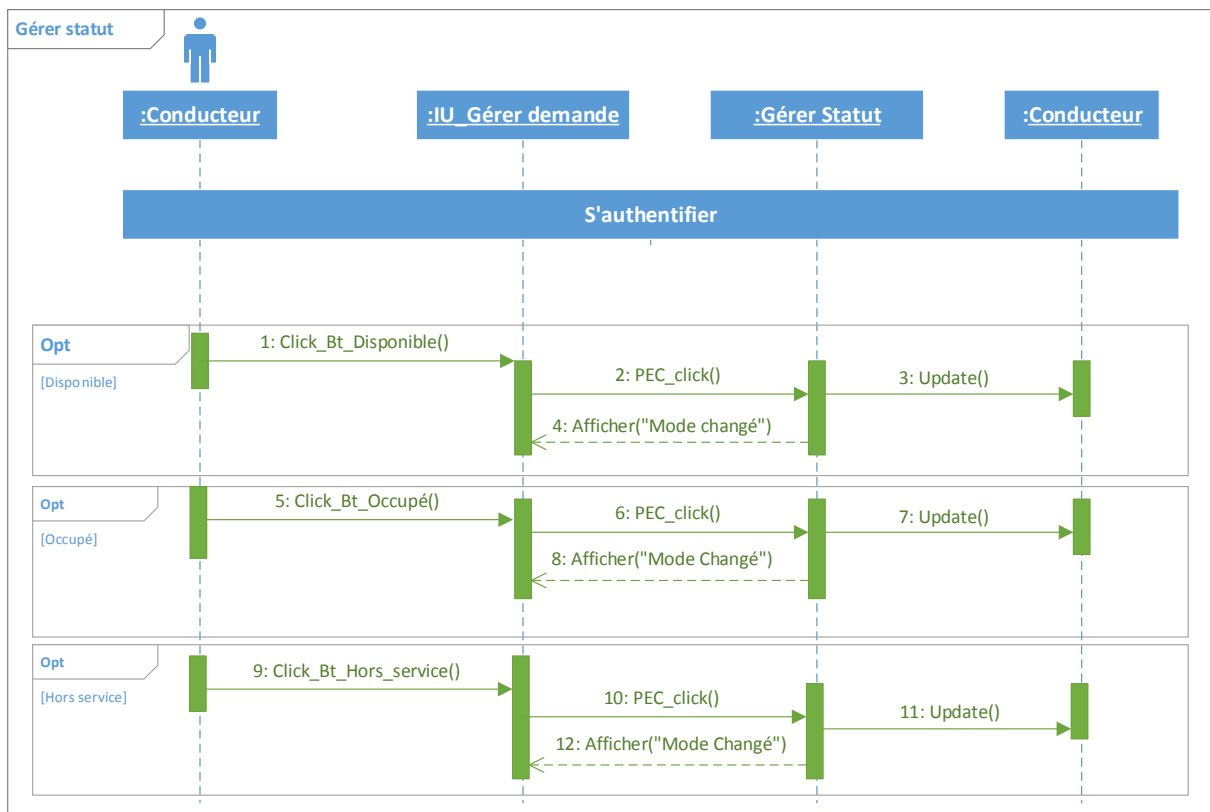


Figure 25:Diagramme de séquence détaillé du cas d'utilisation "Gérer statut".

### c. Diagramme de séquence détaillé du cas d'utilisation « Gérer la rencontre »

Pour assurer la rencontre avec son conducteur, le passager a le choix de saisir une référence qui consiste à lui envoyer un message personnalisé en lui précisant un indice convenu entre eux ou son emplacement exact. Il peut aussi présenter un point de rencontre en faisant un clic sur la carte géographique. A ce moment, le chauffeur de taxi le rejoint à cette position.

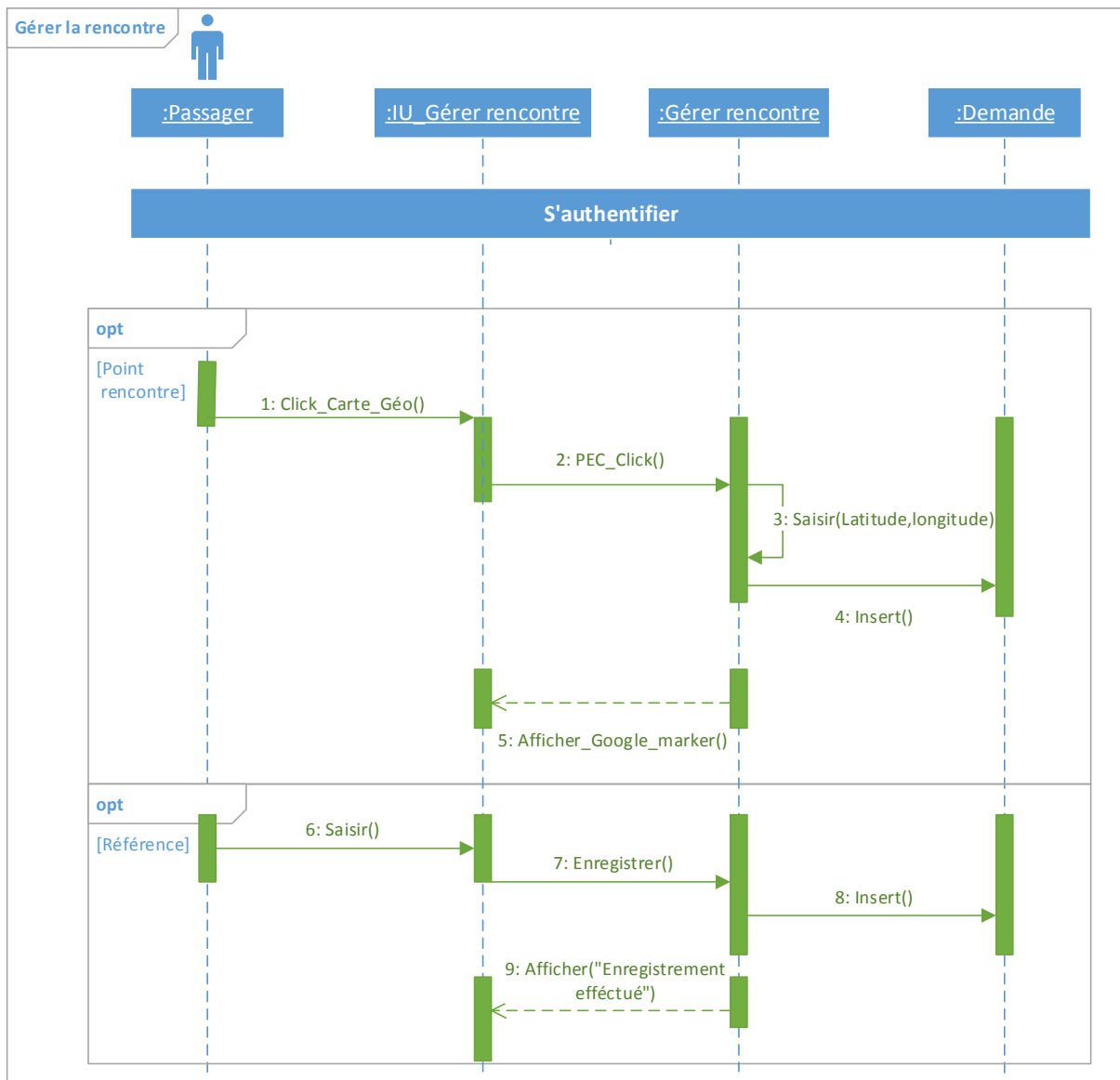


Figure 26:Diagramme de séquence détaillé du cas d'utilisation "Gérer la rencontre"

## 2.4. Diagramme des classes de conception :

Le diagramme de classe est l'un des diagrammes statiques d'UML. Il permet de décrire la structure d'un système informatique tout en montrant les différentes classes, leurs attributs, leurs méthodes ainsi que les relations entre eux. Tout au long de mon sprints, j'essaierai de construire ce diagramme au fur et mesure en ajoutant les différentes classes déduites.

La figure ci-dessous illustre le diagramme de classe de conception de ce premier sprint :

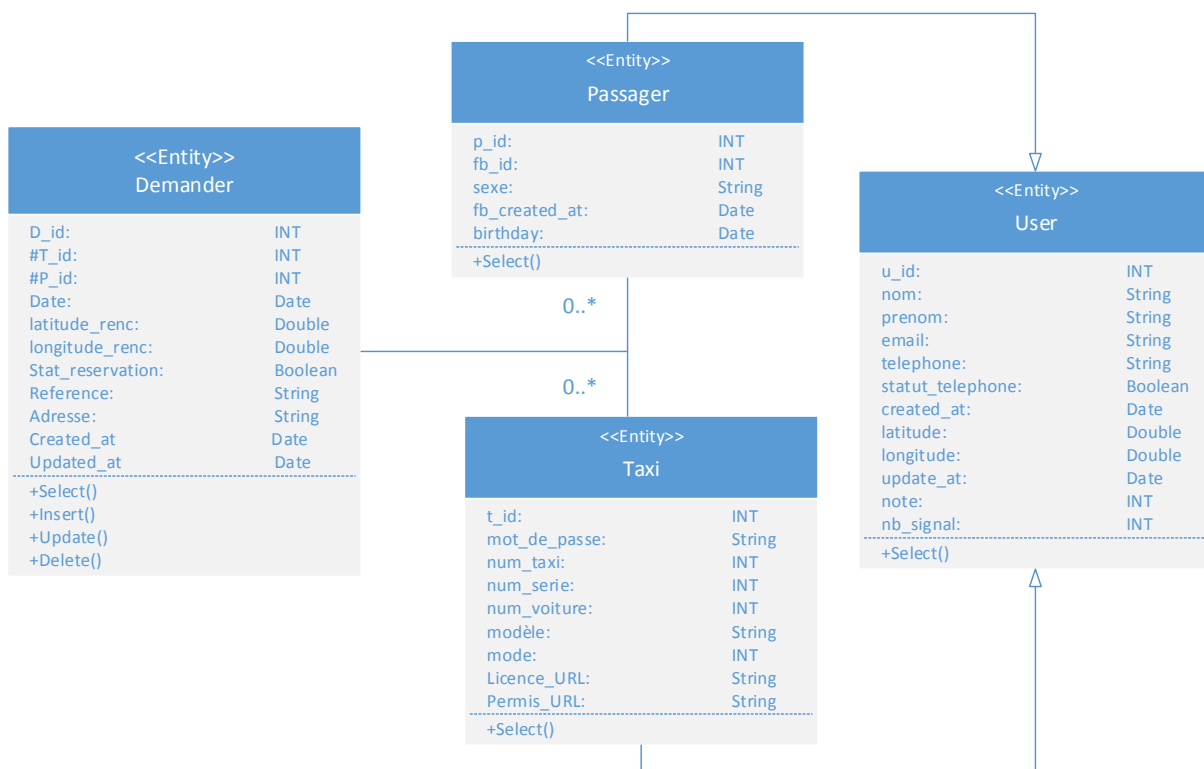


Figure 27: Diagramme de classe de conception du sprint 1.

### 3. Codage :

Les travaux menés dans cette activité se résument tout simplement dans l'implémentation et la réalisation des histoires utilisateurs analysés lors des étapes précédentes. Pour notre cas, nous nous intéresserons seulement au schéma de la base de données.

Dans ce qui suit, je présente les tables de ma base de données, tout en tenant compte du type et des contraintes de leurs champs.

Champs	Types	Contraintes
<b>Id</b>	INT	PRIMARY KEY
<b>Nom</b>	VARCHAR (20)	---
<b>Prénom</b>	VARCHAR (20)	---
<b>Email</b>	VARCHAR (20)	UNIQUE
<b>mot_de_Passe</b>	VARCHAR (20)	---
<b>num_taxi</b>	INT	---
<b>num_série</b>	INT	---
<b>num_voiture</b>	INT	---
<b>Modèle</b>	VARCHAR (20)	---
<b>stat_mode</b>	INT	---
<b>telephone</b>	VARCHAR (8)	---
<b>stat_Tel</b>	BOOLEAN	---
<b>created_at</b>	DATE	---
<b>updated_at</b>	DATE	---
<b>licence_URL</b>	VARCHAR (20)	---
<b>permis_URL</b>	VARCHAR (20)	---
<b>Latitude</b>	FLOAT (10,6)	---
<b>longitude</b>	FLOAT (10,6)	---
<b>Note</b>	VARCHAR (20)	---

**Tableau 15 : Structure de la table "Taxi".**



Champs	Types	Contraintes
<b>Id</b>	INT	PRIMARY KEY
<b>Nom</b>	VARCHAR (20)	---
<b>Prénom</b>	VARCHAR (20)	---
<b>Email</b>	VARCHAR (20)	UNIQUE
<b>Téléphone</b>	VARCHAR (8)	---
<b>Stat Téléphone</b>	BOOLEAN	---
<b>Created_at</b>	DATE	---
<b>Fb_Created_at</b>	Date	---
<b>Latitude</b>	FLOAT (10,6)	---
<b>Longitude</b>	FLOAT (10,6)	---
<b>Note</b>	INT	---

**Tableau 15: Structure de la table "Passager".**

Champs	Types	Contraintes
<b>D_id</b>	INT	PRIMARY KEY
<b>P_id</b>	INT	FOREIGN KEY
<b>T_id</b>	INT	FOREIGN KEY
<b>created_at</b>	DATE	---
<b>latitude_rencontre</b>	FLOAT (10,6)	---
<b>Longitude_rencontre</b>	FLOAT (10,6)	---
<b>Stat réservation</b>	BOOLEAN	---
<b>Adresse</b>	VARCHAR (50)	---

**Tableau 16: Structure de la table "Demande".**

## **4. Test :**

Le test est un processus manuel ou automatique, qui vise à établir qu'un système vérifie les propriétés exigées par sa spécification, ou à détecter des différences entre les résultats engendrés par le système et ceux qui sont attendus par la spécification. (Définition issue de la norme IEEE-STD729, 1983).

Les activités de test constituent un axe très important dans le cycle de développement d'un logiciel. Ils permettent de détecter les erreurs afin de les corriger et d'assurer la qualité du logiciel fourni.

Contrairement aux cycles de développement séquentiel<sup>6</sup>, avec la méthodologie agile, le test n'est pas une phase qui se déroule après la fin de développement. En effet, les tests seront intégrés dès le début du premier sprint jusqu'à la livraison du produit final. En outre, la qualité du logiciel n'est pas négligeable, c'est dans ce cadre que Scrum doit être complété par les bonnes pratiques d'ingénierie techniques du logiciel. Parmi ces pratiques<sup>7</sup>, seulement deux m'intéressent et qui est le pilotage par les tests centrés sur les tests unitaires et les interfaces de l'application.

### **4.1. Test unitaire :**

Le principe de cette pratique est d'écrire les tests avant même d'écrire le code et de profiter par la suite de l'existence des tests automatiques pour l'amélioration et le remaniement du code. Cette technique permet aux programmeurs de rester simples au niveau du code et de s'assurer de son bon fonctionnement après des changements.

La figure suivante illustre le cas de succès pour l'envoi d'une demande. Cette figure nous montre que ma méthode est bien fonctionnelle et que la demande a bien été envoyée.

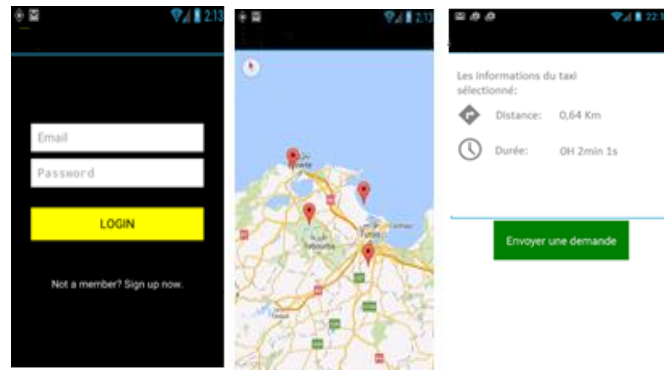
### **4.2. Les tests de l'application mobile :**

---

<sup>6</sup> Ce sont les méthodologies dont les activités de développement (spécification, conception, codage et test) se déroulent séquentiellement notamment le modèle en cascade ou en V. (Nommage de Claude Aubry)

<sup>7</sup> Les pratiques les plus connues sont : l'intégration continue, la programmation en binôme, etc.

À ce stade, nous exposons quelques interfaces de notre application à travers différents imprimes écrans réalisés. En essayant à chaque fois de décrire les différents objets interactifs mis à la disposition de l'utilisateur.



**Figure28:**  
**Interface**  
**authentification**  
**Conducteur.**

**Figure 30:**  
**Interface de**  
**sélection de taxi**  
**(Passager).**

**Figure 29:**  
**Interface effectuer**  
**une demande.**

La figure 30 présente l'interface où le conducteur doit saisir ses coordonnées pour accéder à l'application.

Les figures 31, 32 présentent les interfaces nécessaires au passager pour effectuer une demande.

## II. Deuxième Sprint :

En partant sur le même principe que le sprint précédent, je commence par définir le but de mon second sprint. Suite à une conversation entre le Product Owner et l'équipe Scrum, nous avons décidé le but suivant : « terminer la partie qui concerne la réservation et le contact ».

Une fois que le but de mon sprint est défini, il est temps de décider quelles histoires inclure dans ce dernier.

Le tableau ci-dessous résume le Backlog de mon second sprint.

Histoire utilisateur	Priorité
Consulter les demandes	Elevée
Accepter la demande	Elevée
Refuser la demande	Elevée
Consulter le profil du passager	Moyenne
Signaler le passager	Moyenne
Signaler le passager	Moyenne
Noter le conducteur	Moyenne
Noter le passager	Moyenne
Envoyer des messages instantanés	Faible
Effectuer un appel téléphonique.	Faible

Tableau 17: Backlog du second sprint (Release 1).

### 1. Spécification Fonctionnelle

Pour la spécification fonctionnelle de ce sprint, j'ai commencé par le diagramme des cas d'utilisation.

#### 1.1. Scénario et cas d'utilisation

Dans la figure 34 j'illustre le diagramme des cas d'utilisation global du second sprint.

En respectant toujours le même principe que le sprint précédent, j'ai découpé certaines histoires en un ensemble de tâches.

## 1.2. Description textuelle des cas d'utilisation

Je vais maintenant décrire chacun des cas d'utilisation énumérés dans le paragraphe précédent en identifiant les acteurs et les différents scénarios possibles.

### 1.2.1. Description du cas d'utilisation « Gérer la réservation »

Cas d'utilisation :	<b>Gérer la réservation</b>
Acteur :	Conducteur
Pré condition :	<ul style="list-style-type: none"><li>- Le passager est Authentifié.</li><li>- Le conducteur est Authentifié.</li><li>- Le conducteur est en attente de demandes.</li></ul>
Post condition :	<ul style="list-style-type: none"><li>- Le conducteur accepte ou refuse les demandes reçues.</li></ul>
Description du scénario principal :	<ul style="list-style-type: none"><li>- Le système alerte le conducteur des demandes reçues.</li><li>- Le chauffeur de taxi sélectionne le bouton de notification pour voir toutes les demandes reçues sous la rubrique « Demande rapide ».</li><li>- Le système affiche la liste des demandes :<ul style="list-style-type: none"><li>• Le système affiche la demande reçue par sélection des passagers sur son interface sous forme de message d'alerte et ainsi que dans la liste des demandes.</li><li>• Le système affiche la demande reçue par demande rapide directement dans la liste des demandes.</li></ul></li></ul> <p>Le chauffeur de taxi appuie sur « Accepter » pour valider la demande du client.</p> <ul style="list-style-type: none"><li>- Le système affiche l'emplacement et l'itinéraire vers le client.</li><li>- Le conducteur appuie sur « Refuser » pour décliner la demande du passager.</li></ul>
Exception :	<ul style="list-style-type: none"><li>- Si le conducteur ne répond pas aux demandes, le système avertit le passager du refus de la demande.</li><li>- Si le conducteur essaye de répondre à plus qu'une demande, le système affiche un message d'avertissement « Vous ne pouvez pas accepter plus qu'une demande à la fois ».</li></ul>

Tableau 18: Description du cas d'utilisation "Gérer la réservation".

### 1.2.2. Description du cas d'utilisation « Gérer l'évaluation »

Cas d'utilisation :	<b>Gérer l'évaluation</b>
Acteur :	Conducteur Passager
Pré condition :	<ul style="list-style-type: none"> <li>- Le passager est Authentifié.</li> <li>- Le chauffeur de taxi est Authentifié.</li> <li>- Le chauffeur de taxi a accepté la demande.</li> </ul>
Post condition :	<ul style="list-style-type: none"> <li>- Le conducteur a évalué le passager.</li> <li>- Le passager a évalué le conducteur.</li> </ul>
Description du scénario principal :	<ul style="list-style-type: none"> <li>- Le passager appuie sur le bouton « Arriver » pour signaler l'arrivée du Taxi.</li> <li>- Le système affiche le « Rating Bar » au passager pour qu'il note le Conducteur.</li> <li>- Lorsque le chauffeur de taxi signale la fin de la course, le système lui affiche le « Rating Bar » pour noter le passager.</li> <li>- Le passager appuie sur « Annuler la demande » et saisit les causes d'annulation pour signaler le conducteur.</li> <li>- Le conducteur appuie sur « Annuler la réservation » et saisit les causes d'annulation pour signaler le passager.</li> </ul>
Exception :	<ul style="list-style-type: none"> <li>- Si le passager ne note pas le conducteur, le système annule automatiquement cet acte.</li> <li>- Si le conducteur ne note pas le passager, le système annule automatiquement cet acte.</li> </ul>

Tableau 19: Description du cas d'utilisation "Gérer l'évaluation".

### 1.2.3. Description du cas d'utilisation « Gérer le contact »

Cas d'utilisation :	<b>Gérer le contact</b>
Acteur :	Passager
Pré condition :	<ul style="list-style-type: none"> <li>- Le passager est Authentifié.</li> <li>- Le chauffeur de taxi est Authentifié.</li> </ul>
Post condition :	<ul style="list-style-type: none"> <li>- Le passager a contacté le conducteur.</li> </ul>
Description du scénario principal :	<ul style="list-style-type: none"> <li>- Le conducteur peut autoriser les appels téléphoniques.</li> <li>- Le passager appuie sur le bouton « Appel téléphonique » pour appeler le conducteur.</li> <li>- Le système transfère l'appel au conducteur.</li> </ul>
Exception :	<ul style="list-style-type: none"> <li>- Si le conducteur n'autorise pas les appels téléphoniques, le système affiche au passager le message « Vous ne pouvez pas effectuer d'appel téléphonique ».</li> </ul>

Tableau 20: Description du cas d'utilisation "Gérer le contact".

Le schéma suivant synthétise le cas d'utilisation du sprint 2 :

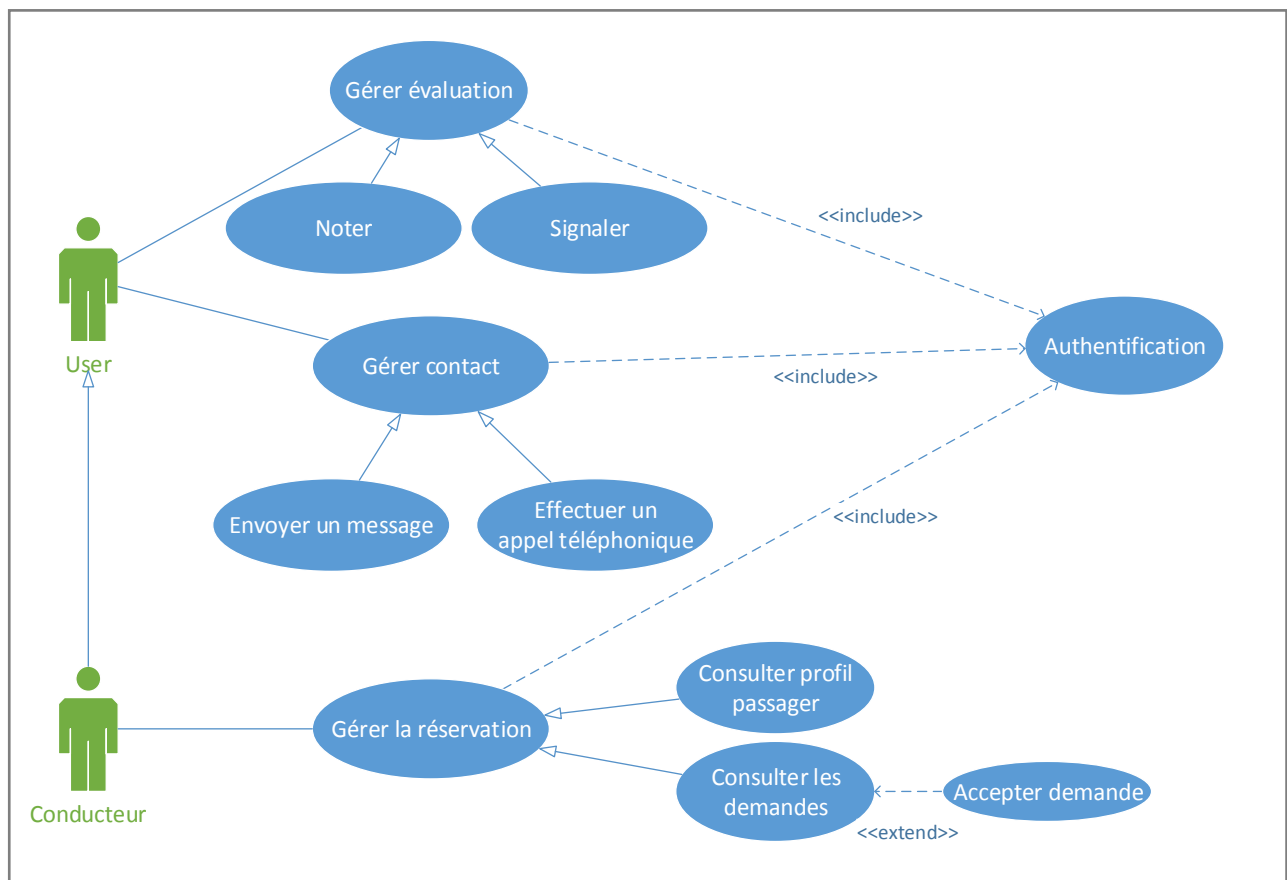


Figure 31: Description des cas d'utilisation du Sprint 2.

## 2. Conception

Nous effectuons dans ce qui suit les diagrammes de séquence système des différents cas d'utilisation déjà expliqués dans la section précédente.

### 2.1. Diagramme de séquence système

En me référant aux descriptions textuelles dans la section précédente, je présente les diagrammes de séquences systèmes adéquats. Sur cette base j'expose les cas d'utilisation vus précédemment.

### a. Diagramme de séquence système du cas d'utilisation « Gérer la réservation »

Il est illustré comme suit :

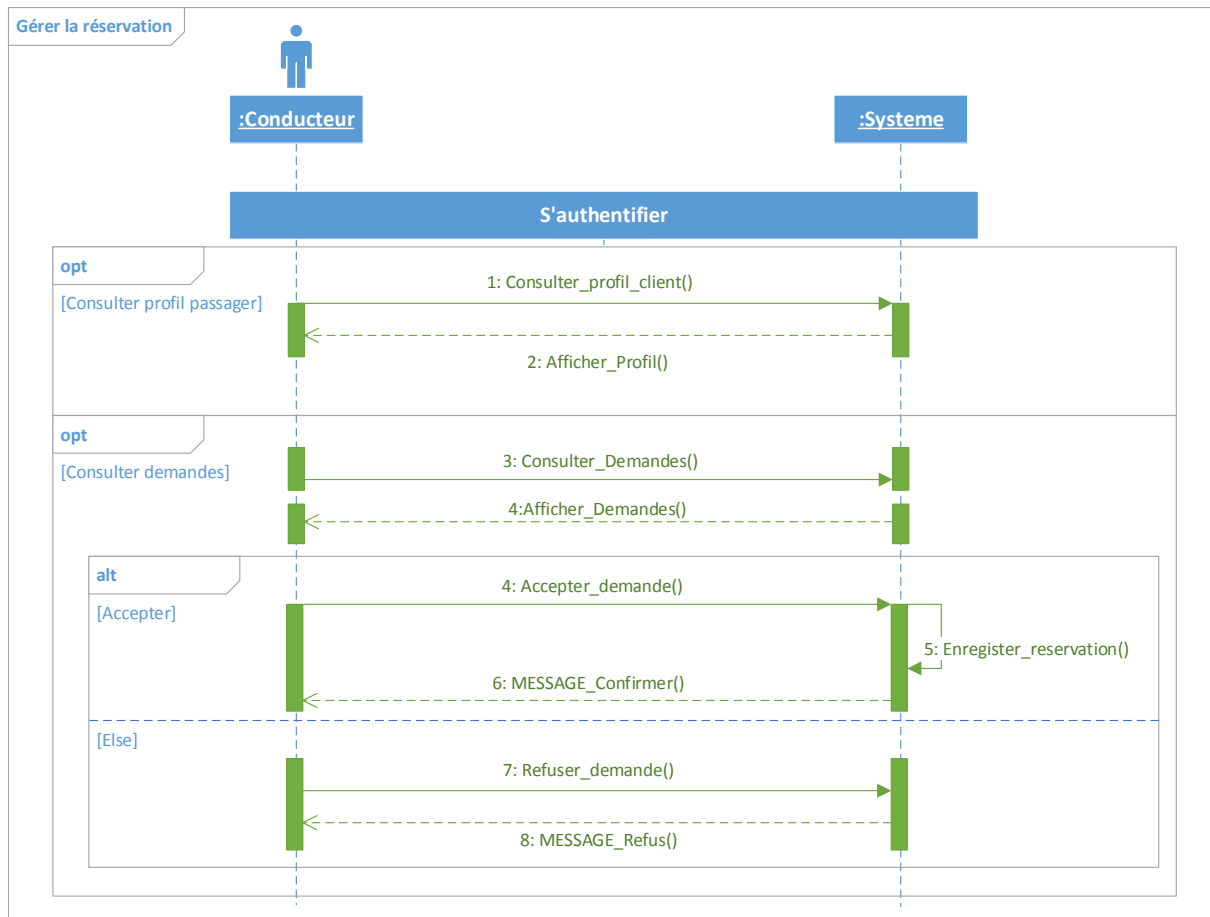


Figure 32: Diagramme de séquence système du cas d'utilisation "Gérer la réservation".



## b. Diagramme de séquence système du cas d'utilisation « Gérer le Contact »

Ci-dessous, le schéma représentatif :

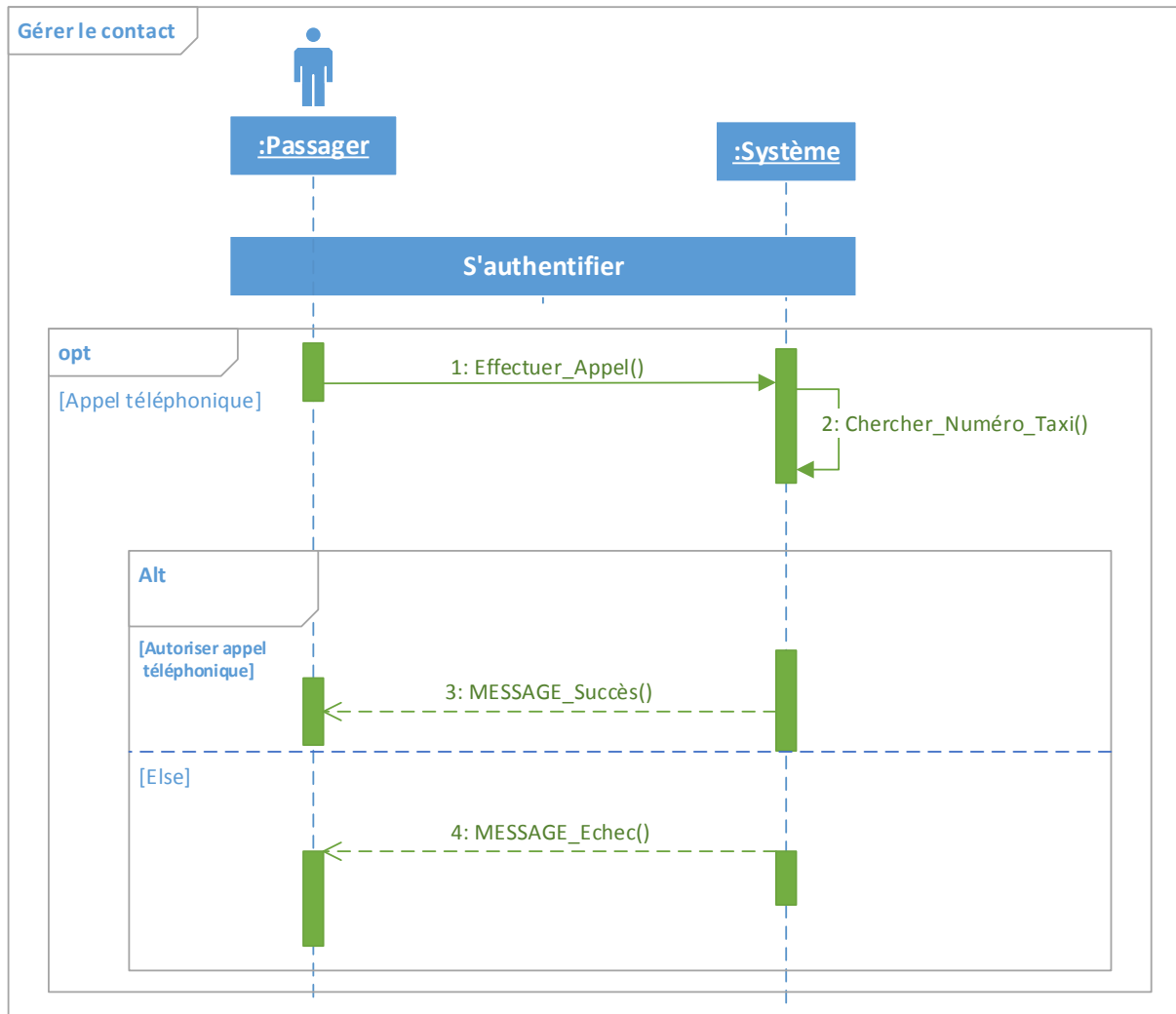


Figure 33: Diagramme de séquence système "Gérer le contact".

### c. Diagramme de séquence système du cas d'utilisation « Gérer l'évaluation »

Il est illustré comme suit :

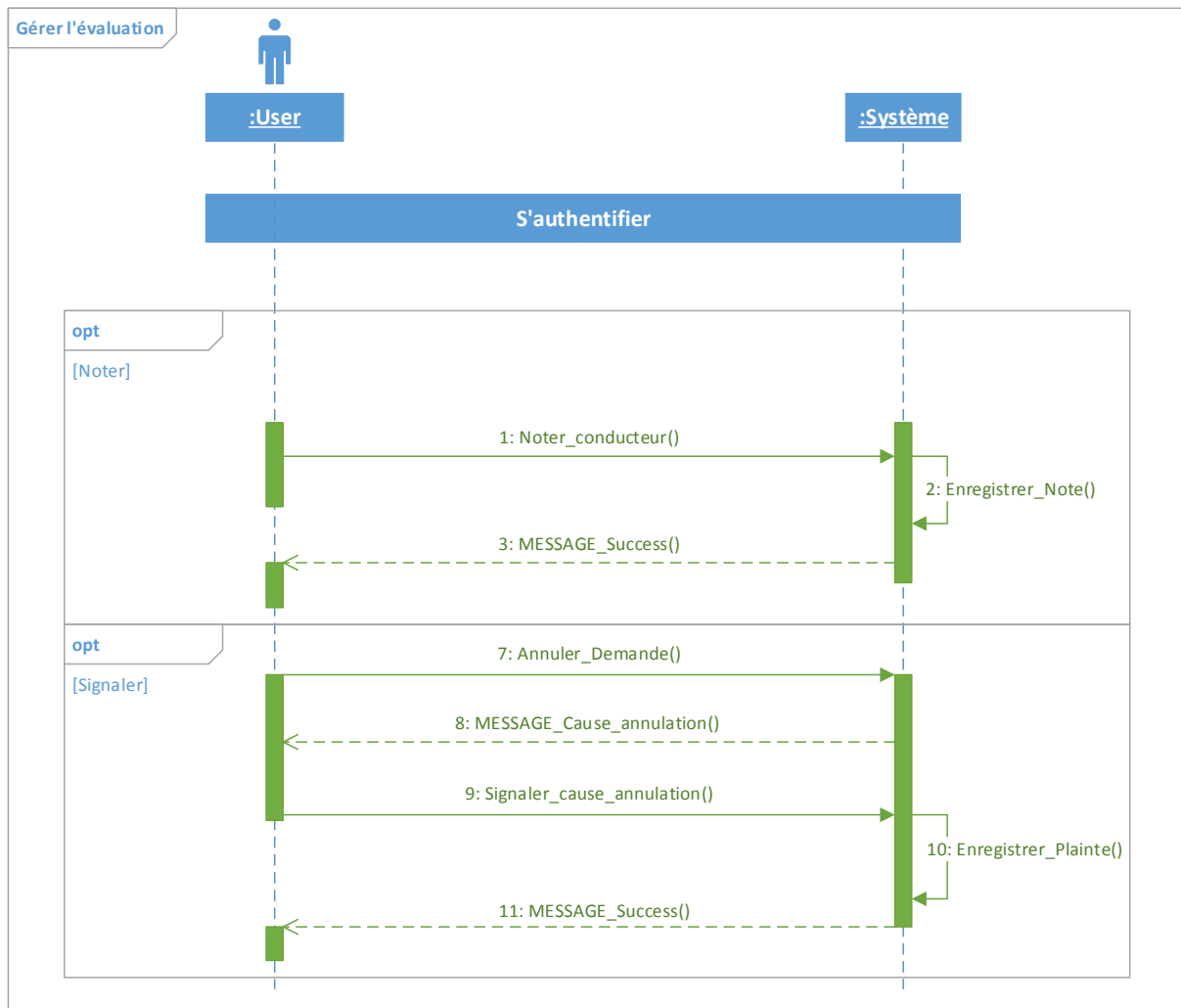


Figure 34: Diagramme de séquence système du cas d'utilisation "Gérer l'évaluation".

## 2.2. Diagramme des classes de conception

A ce stade, je présente les interfaces « IU\_Interface », et les tables « Entity » cible pour expliquer le déroulement des cas vus précédemment dans le diagramme de séquence système. Dans la section suivante, j'illustre les différentes tables concernées.

### a. Diagramme de classe pour le cas « Gérer réservation »

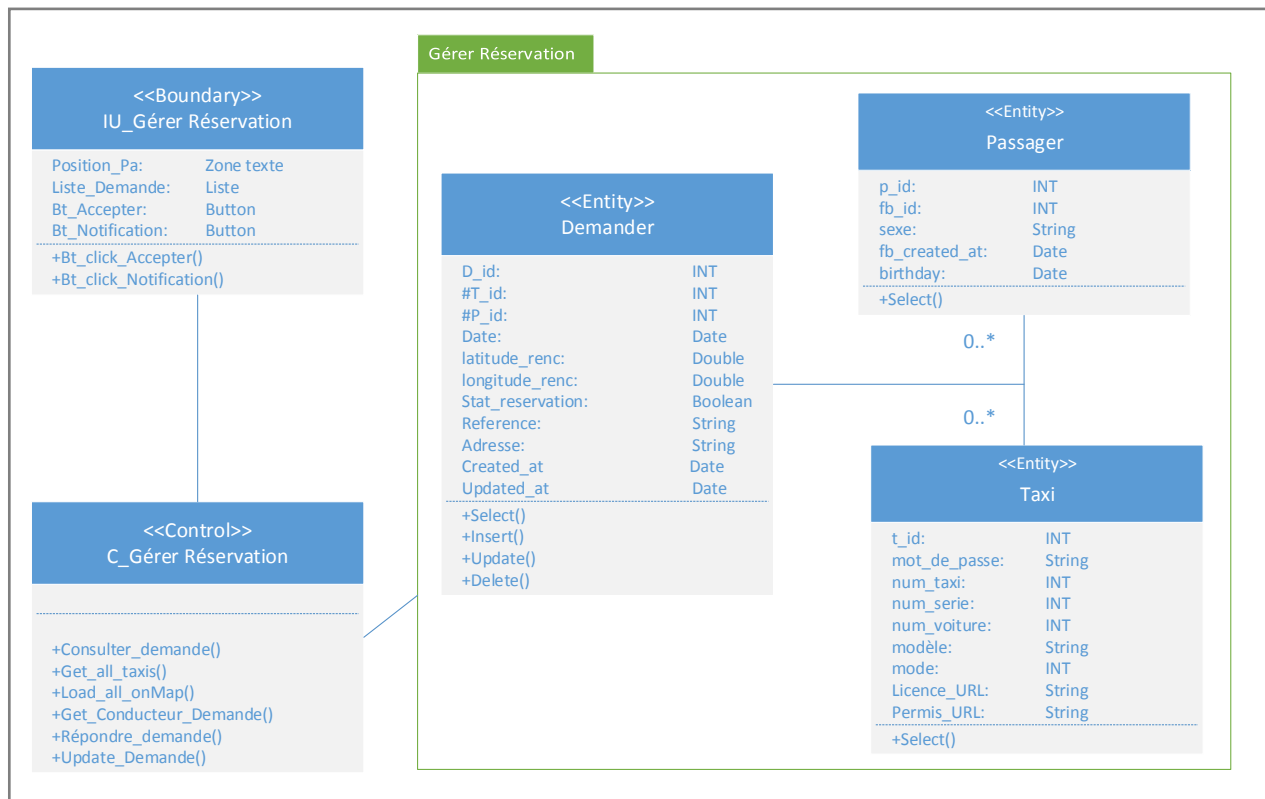


Figure 35: Diagramme de classe du cas d'utilisation "Gérer réservation".

### b. Diagramme de classe du cas d'utilisation « Gérer le Contact »

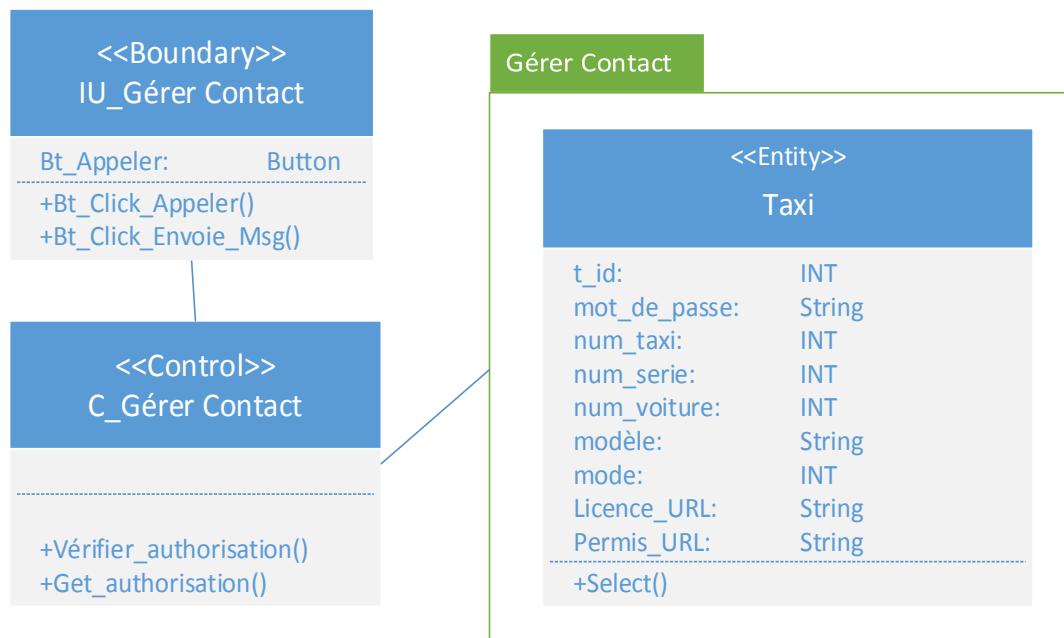


Figure 36: Diagramme de classe du cas d'utilisation "Gérer le Contact".

### c. Diagramme de classe du cas d'utilisation « Gérer l'évaluation »

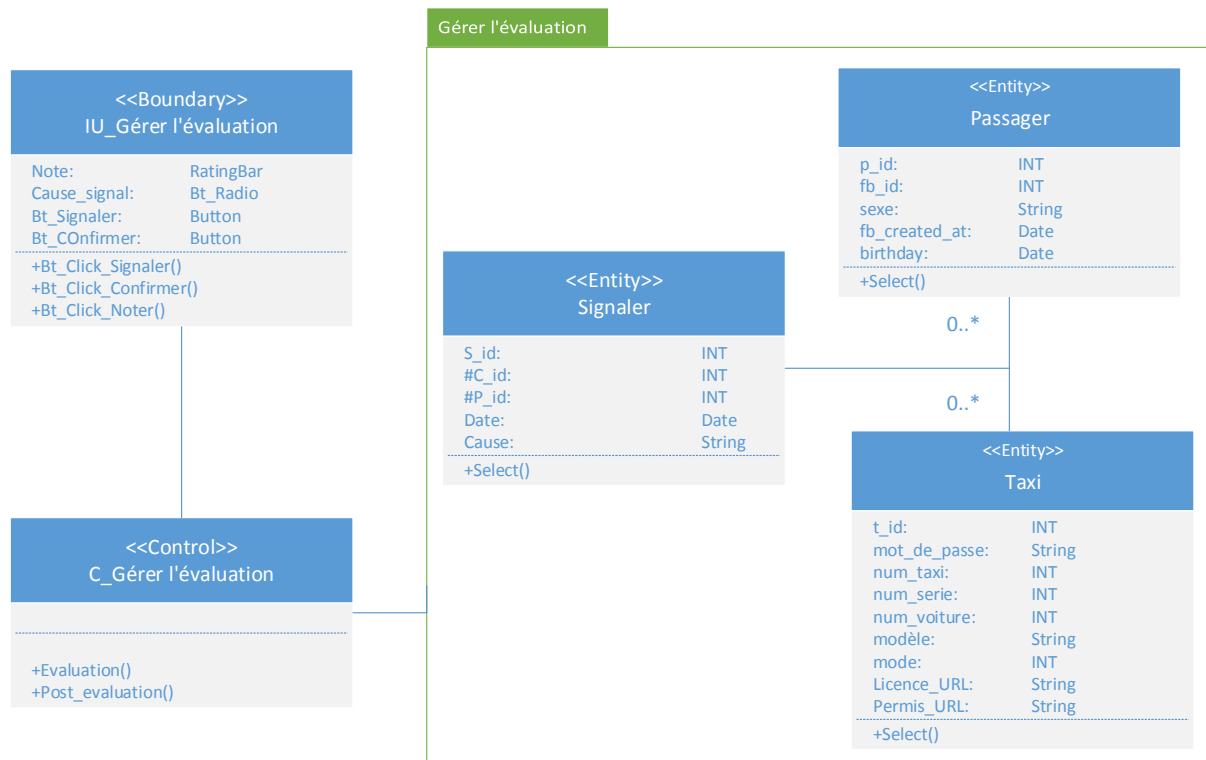


Figure 37: Diagramme de classe du cas d'utilisation "Gérer l'évaluation".

### 2.3. Diagramme de séquence détaillé

#### a. Diagramme de séquence détaillé du cas d'utilisation « Gérer la réservation »

Dans ce cas d'utilisation, le chauffeur de Taxi peut consulter le profil du passager et aussi consulter les demandes envoyées par les passagers.

Si le conducteur accepte une demande, le champ « Statut réservation » aura la valeur « 1 » pour signaler la réservation et toutes les autres demandes seront automatiquement et immédiatement refusées et effacées de la table « Demandes ».

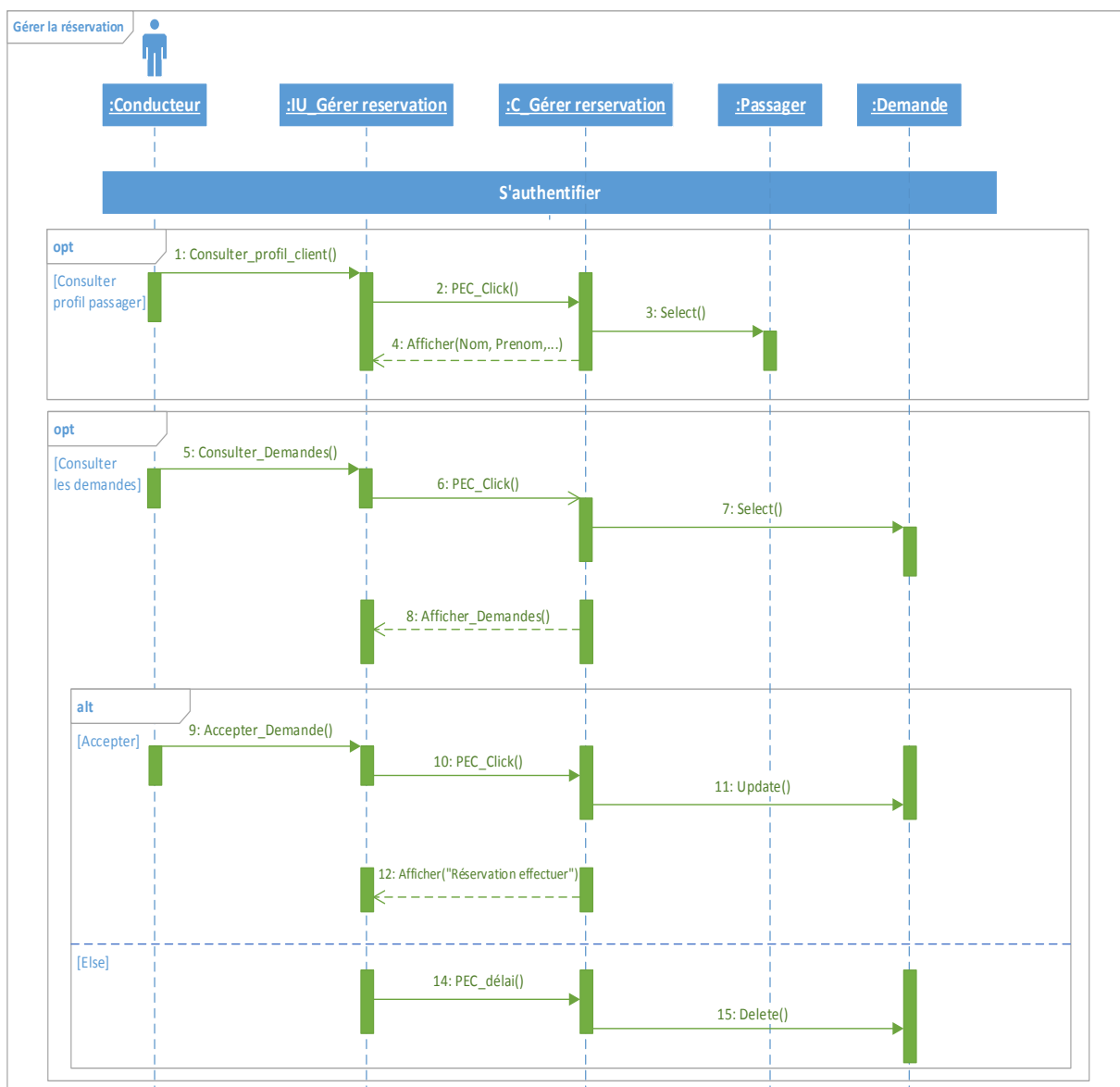
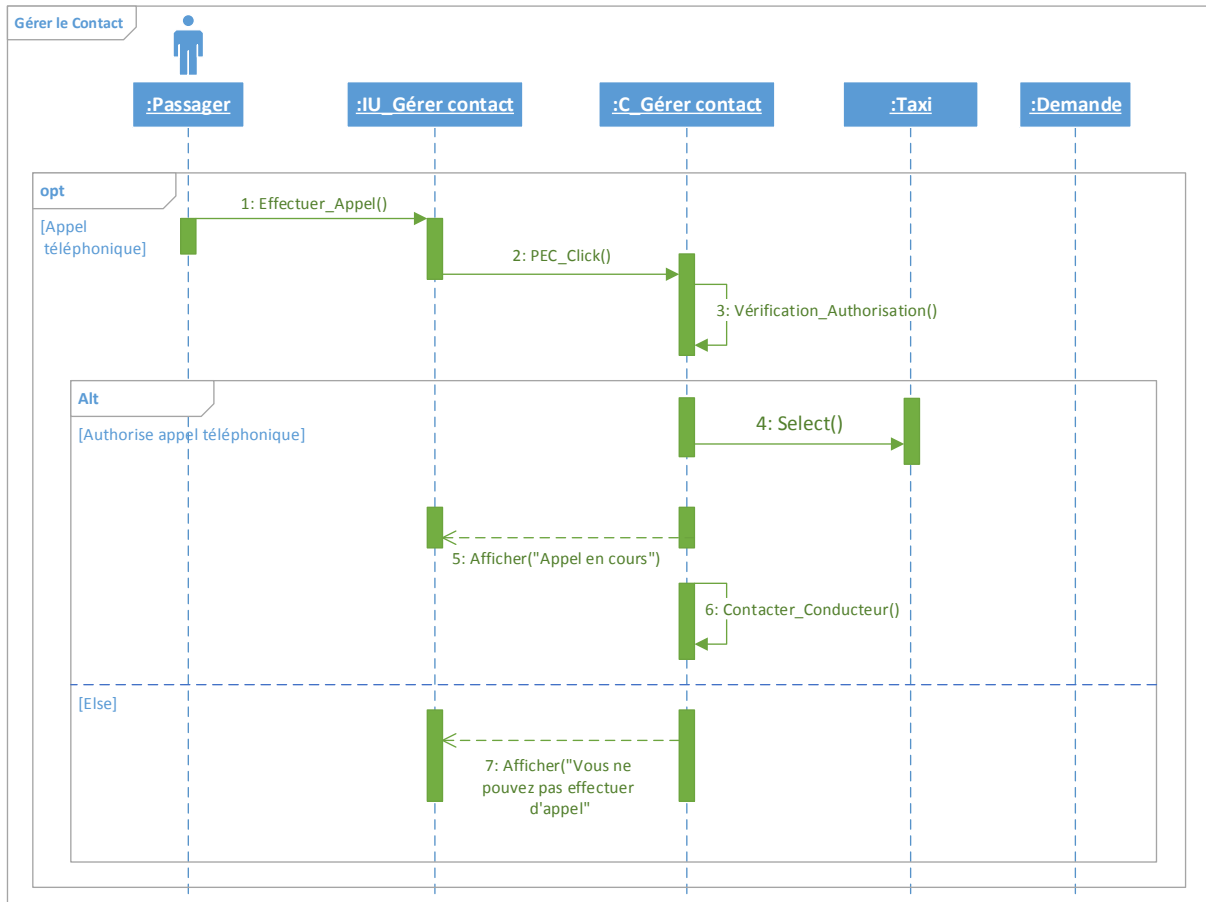


Figure 38: Diagramme de séquence détaillé du cas d'utilisation "Gérer réservation".

**b. Diagramme de séquence détaillé du cas d'utilisation « Gérer le contact »**

Le passager a la possibilité d'effectuer un appel téléphonique, après avoir vérifié l'autorisation du conducteur avant que le système transfère l'appel à ce dernier.



**Figure 39: Diagramme de séquence du cas d'utilisation "Gérer le contact passager".**

### c. Diagramme de séquence détaillé du cas d'utilisation « Gérer l'évaluation »

Dans ce cas, je cite, comme exemple, l'évaluation du passager :

En notant le conducteur, le système calcule le pourcentage de note acquise par les passagers et l'enregistre dans le champ « Note » dans la table « Taxi ». Par suite le champ « Signal » dans la table « Taxi » sera incrémenté d'un point pour chaque plainte reçue par un passager.

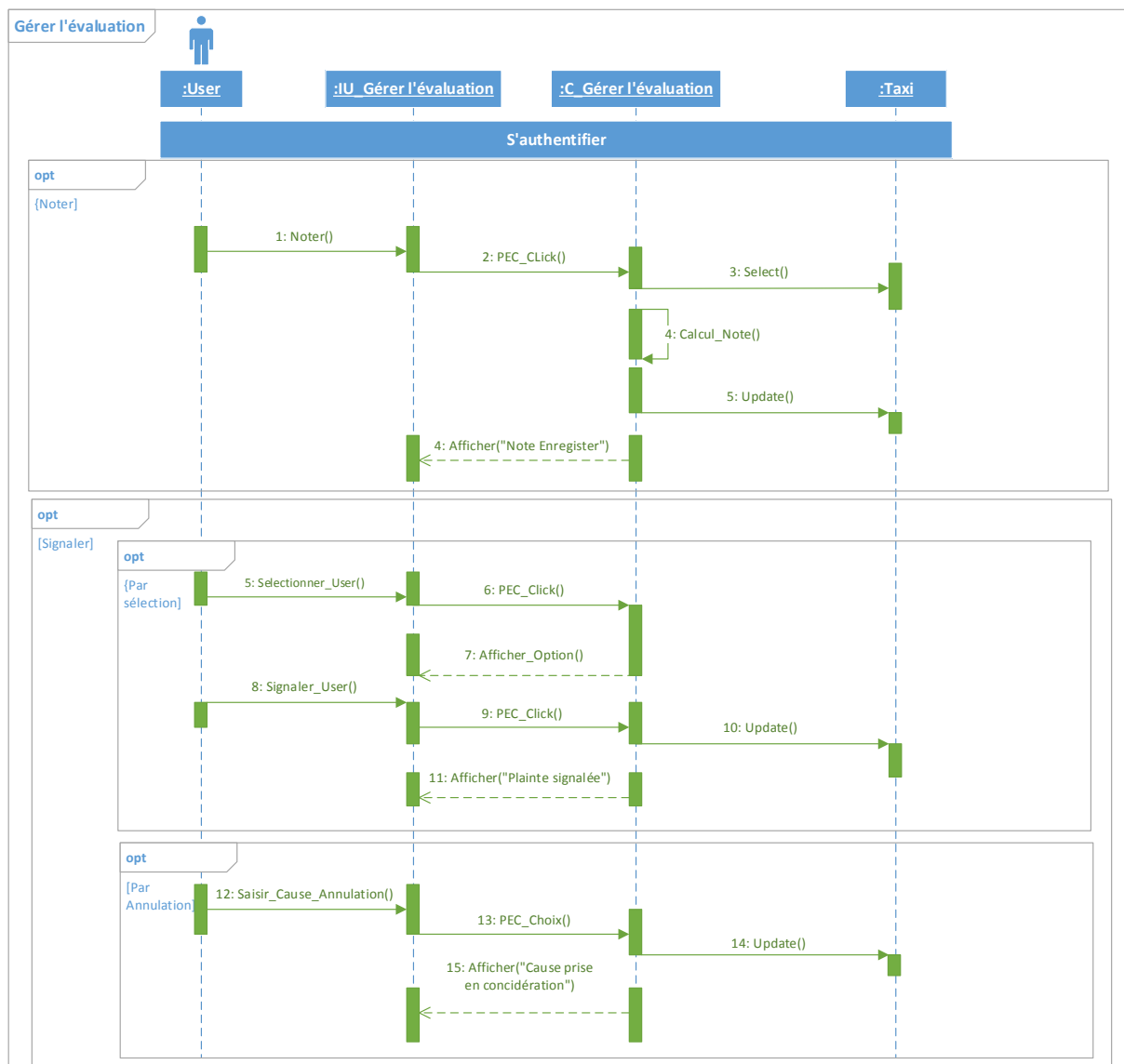


Figure 40: Diagramme de séquence détaillé du cas d'utilisation "Gérer l'évaluation".

## 2.4. Diagramme des classes de conception :

Après tous le travail de spécification et de conception, je peux maintenant construire le nouvel incrément de mon diagramme de classes en ajoutant les différents éléments (classes, associations, attributs, etc.) déduits à partir des activités précédentes.

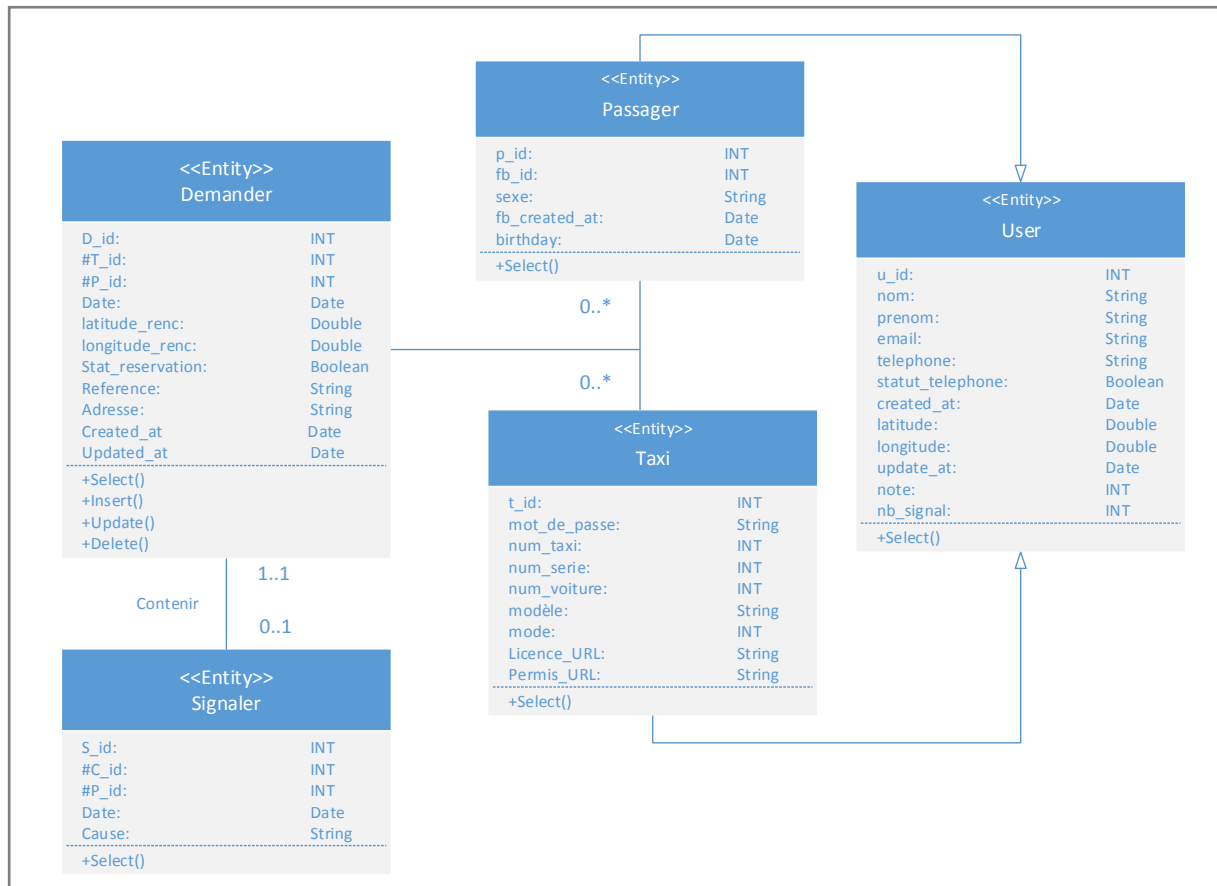


Figure 41: Diagramme de classe de conception du sprint 2.



### 3. Codage :

Champs	Types	Contraintes
id	INT	PRIMARY KEY
nom	VARCHAR (20)	---
prenom	VARCHAR (20)	---
email	VARCHAR (20)	UNIQUE
mot_de_Passe	VARCHAR (20)	---
num_taxi	INT	UNIQUE
num_série	INT	---
num_voiture	INT	---
modèle	VARCHAR (20)	---
stat_mode	INT	---
telephone	VARCHAR(8)	---
stat_Tel	BOOLEAN	---
created_at	DATE	---
updated_at	DATE	---
licence_URL	VARCHAR (20)	---
permis_URL	VARCHAR (20)	---
latitude	FLOAT (10,6)	---
longitude	FLOAT (10,6)	---
note	VARCHAR (20)	---

Tableau 21:Structure de la table "taxi".

Champs	Types	Contraintes
<b>id</b>	INT	PRIMARY KEY
<b>nom</b>	VARCHAR (20)	---
<b>prenom</b>	VARCHAR (20)	---
<b>email</b>	VARCHAR (20)	UNIQUE
<b>telephone</b>	VARCHAR(8)	---
<b>Stat_Telephone</b>	BOOLEAN	---
<b>created_at</b>	DATE	---
<b>fb_created_at</b>	Date	---
<b>latitude</b>	FLOAT (10,6)	---
<b>Longitude</b>	FLOAT (10,6)	---
<b>Note</b>	INT	---

**Tableau 22: Structure de la table "Passager".**

Champs	Types	Contraintes
<b>D_id</b>	INT	PRIMARY KEY
<b>P_id</b>	INT	FOREIGN KEY
<b>T_id</b>	INT	FOREIGN KEY
<b>created_at</b>	DATE	---
<b>latitude_rencontre</b>	FLOAT (10,6)	---
<b>longitude_rencontre</b>	FLOAT (10,6)	---
<b>Stat_reservation</b>	BOOLEAN	---
<b>adresse</b>	VARCHAR(50)	---

**Tableau 23: Structure de la table "Demande".**

## **4. Test :**

En partant toujours du même principe que le sprint précédent, et en appliquant les bonnes pratiques d'ingénierie logicielle inspirée de la méthodologie extrême programming (XP), je commence par les tests unitaires.

### **4.1. Tests unitaires :**

Les figures suivantes illustrent le cas de succès pour. Cette figure nous montre que ma méthode est bien fonctionnelle.

### **4.2. Les tests de l'application mobile :**

À ce stade, j'expose quelques interfaces de mon application à travers différents imprimés écrans réalisés. En essayant à chaque fois de décrire les différents objets interactifs mis à la disposition de l'utilisateur.

Les figures 40 et 41 montrent les différentes étapes du déroulement de la réservation.

**Figure 40 : Interface de demandes reçues.**

**Figure 41 : Interface de rencontre.**

## *Conclusion :*

Le résultat d'un release est un produit livrable au client contrairement au résultat d'un sprint qui est un produit potentiellement livrable. A la fin de ce chapitre, j'ai réussi à produire un incrément ayant suffisamment de valeur pour le client.

Dans le chapitre qui suit, mon effort sera consacré à produire un nouveau release couvrant les fonctionnalités de gestion de la supervision.

---

## **Release 2 :**

# **Gestion de la supervision**

---

### **Introduction :**

Après avoir terminé le premier release de mon système informatique, je peux maintenant me lancer dans les travaux nécessaires pour produire le second release qui concerne les différentes tâches d'un administrateur. En effet les méthodologies agiles, et Scrum en particulier, sont

caractérisées par un rythme régulier. Tout au long du chapitre suivant, j’aurai deux sprints ayant une durée moins importante que les sprints précédents, et je vais traiter les histoires utilisateurs de ces derniers pour avoir à la fin de ce release le logiciel complet, livrable et fonctionnel.

## **I. Le premier sprint :**

En partant sur le même principe que les sprints précédents, je commence par définir le but de mon premier sprint pour ce release. Suite à une conversation entre le Product Owner et l’équipe Scrum, nous avons décidé le but suivant : « terminer la partie qui concerne le contrôle d’administrateur ».

Une fois, nous avons défini le but de notre sprint, il est temps de décider quelles histoires inclure dans ce dernier.

Le tableau 26 résume donc le Backlog de mon sprint :

Histoire utilisateur	Priorité
Consulter conducteur.	Moyenne
Valider conducteur.	Elevé
Supprimer conducteur.	Faible
Bannir conducteur.	Elevée
Consulter passager.	Elevée
Supprimer passager.	Moyenne
Bannir passager.	Faible

**Tableau 24: Backlog du premier sprint (release2).**

### **1. Spécification fonctionnelle :**

La spécification fonctionnelle dans mon cas se traduit par le diagramme des cas d’utilisation d’UML et la description textuelle de ces derniers.

#### **1.1. Scénario et cas d’utilisation :**

Dans la figure 48 j’illustre le diagramme des cas d’utilisation raffiné pour ce premier sprint.

En respectant toujours le même principe que les sprints précédents, je vais découper certaines histoires utilisateurs en un ensemble des tâches.

## 1.2. Description textuelle des cas d'utilisation :

Je vais maintenant décrire chacun des cas d'utilisation énuméré dans le paragraphe précédent en identifiant les acteurs et les différents scénarios possibles.

Chaque tableau présente la description détaillée du fonctionnement du cas d'utilisation. Dans ce Sprint nous allons nous intéresser aux deux cas d'utilisations suivantes :

Gérer compte conducteur



Gérer compte passager

Dans ce sprint, il existe des cas d'utilisations qui ont la même fonctionnalité tels que « Supprimer conducteur », « Supprimer Passager », ...

Dans ce qui suit, nous présentons la description de quelques cas d'utilisation.

### 1.2.1. Description du cas d'utilisation « Consulter Conducteur »

Cas d'utilisation :	<b>Consulter conducteur</b>
Acteur :	Administrateur
Pré condition :	- L'administrateur s'est authentifié.
Post condition :	- L'administrateur a consulté le profil.
Description du scénario principal :	<ul style="list-style-type: none"><li>- L'administrateur choisit l'option « Consulter profil » après avoir cliqué sur le bouton « Conducteur » dans le menu précédent.</li><li>- Le system affiche un tableau qui contient tous les conducteurs avec la possibilité de trier le résultat.</li><li>- L'administrateur sélectionne un profil du conducteur et clique sur le bouton « Voir ».</li><li>- Le système charge les informations du conducteur.</li><li>- Le système affiche les informations.</li></ul>
Exception :	<ul style="list-style-type: none"><li>- L'administrateur annule la demande.</li><li>- Le système annule le chargement des informations.</li><li>- Le système affiche l'interface précédente.</li></ul>

Tableau 25: Description du cas d'utilisation "Consulter conducteur".

### 1.2.2. Description du cas d'utilisation « Bannir conducteur »

Le schéma suivant synthétise le cas d'utilisation du sprint 1 :

Cas d'utilisation :	<b>Bannir conducteur</b>
---------------------	--------------------------

Tableau 26: Description du cas d'utilisation "Bannir conducteur".

Acteur :	Administrateur
Pré condition :	<ul style="list-style-type: none"> <li>- L'administrateur s'est authentifié.</li> <li>- Le conducteur déjà inscrit dans la base « Conducteur »</li> </ul>
Post condition :	<ul style="list-style-type: none"> <li>- L'administrateur a supprimé le profil.</li> </ul>
Description du scénario principal :	<ul style="list-style-type: none"> <li>- L'administrateur choisit l'option « Bannir profil » après avoir cliqué sur le bouton « Conducteur » dans le menu précédent.</li> <li>- Le système affiche les profils qui sont signalés en haut du tableau avec une couleur différente et les profils.</li> <li>- L'administrateur sélectionne ou saisit le profil à bannir.</li> <li>- Le système affiche toutes les informations du conducteur.</li> <li>- L'administrateur clique sur le bouton « bannir » et confirme son choix.</li> <li>- Le système bannit le compte.</li> </ul>
Exception :	<ul style="list-style-type: none"> <li>- L'administrateur annule son choix.</li> <li>- Le système annule l'expulsion du conducteur.</li> </ul>

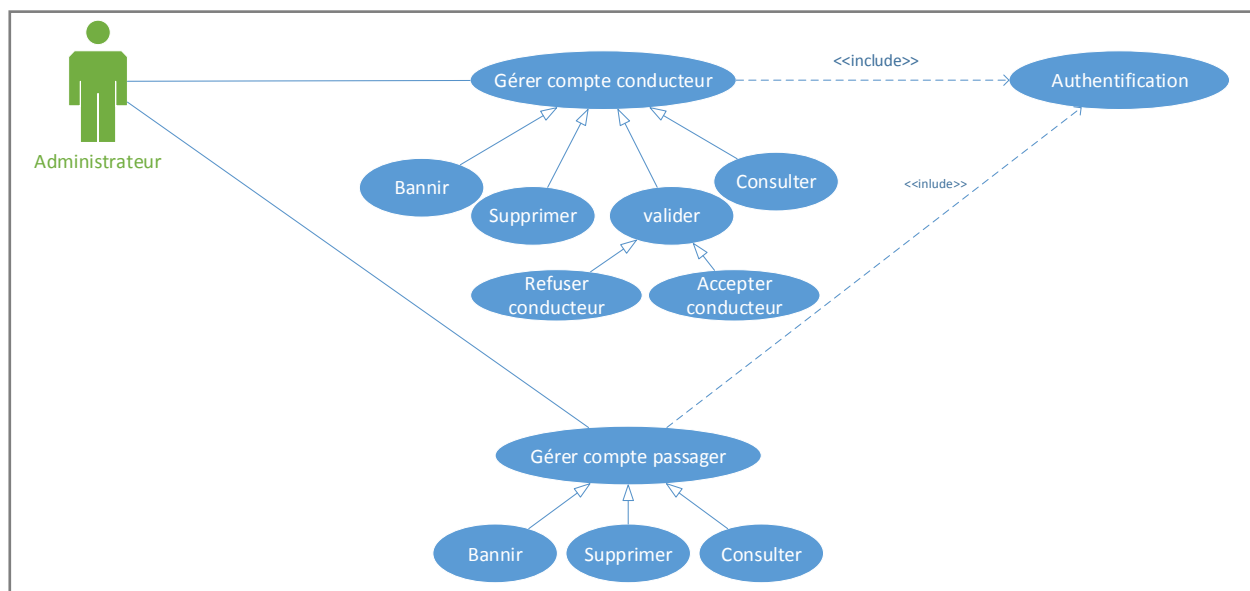


Figure 42: Description des cas d'utilisation du Sprint 1.

### 1.2.3. Description du cas d'utilisation « Valider Conducteur »

Cas d'utilisation :	<b>Valider conducteur</b>
---------------------	---------------------------

Acteur :	Administrateur
Pré condition :	- L'administrateur s'est authentifié.
Post condition :	- L'administrateur a validé le nouveau conducteur.
Description du scénario principal :	<ul style="list-style-type: none"> <li>- L'administrateur choisit l'option « valider un conducteur » après avoir cliqué sur le bouton « Conducteur » dans le menu précédent.</li> <li>- Le système affiche les nouveaux profils qui sont en attente de confirmation.</li> <li>- L'administrateur sélectionne un profil, il vérifie les pièces justificatives, par la suite il confirme sa décision, par le bouton « valider », ou le bouton « refuser » en précisant la cause du refus.</li> <li>- Le système enregistre la confirmation.</li> <li>- Le système informe le conducteur dans le cas de refus.</li> </ul>
Exception :	<ul style="list-style-type: none"> <li>- L'administrateur annule son choix.</li> <li>- Le système annule la validation.</li> </ul>

## 2. Conception :

Dans cette **Tableau 27: Description du cas d'utilisation "Valider Conducteur"**. section je commence par le diagramme de séquence système des différents cas d'utilisation déjà détaillés dans la section précédente.

### 2.1. Diagramme de séquence système :

En me référant aux descriptions textuelles dans la section précédente, je présente les diagrammes de séquences systèmes adéquats. Sur la base de ces descriptions, je peux constater que certains cas d'utilisations sont similaires comme : « Consulter passager », « Supprimer passager », « Bannir passager » etc. Dans ce contexte, j'ai choisi de présenter ces cas ci-dessous :

#### a. Diagramme de séquence système du cas d'utilisation « Consulter conducteur »

Il est illustré comme suit :



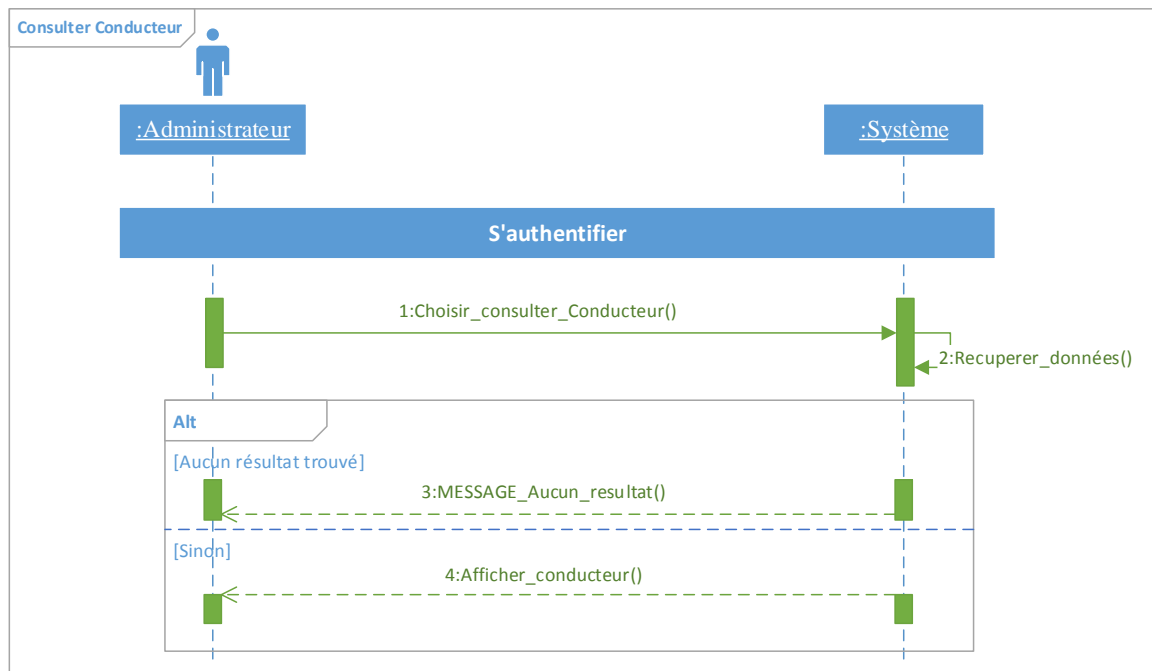


Figure 43: Diagramme de séquence système du cas d'utilisation "Consulter conducteur"

## b. Diagramme de séquence système du cas d'utilisation « Supprimer Conducteur »

Ci-dessous, le schéma représentatif :

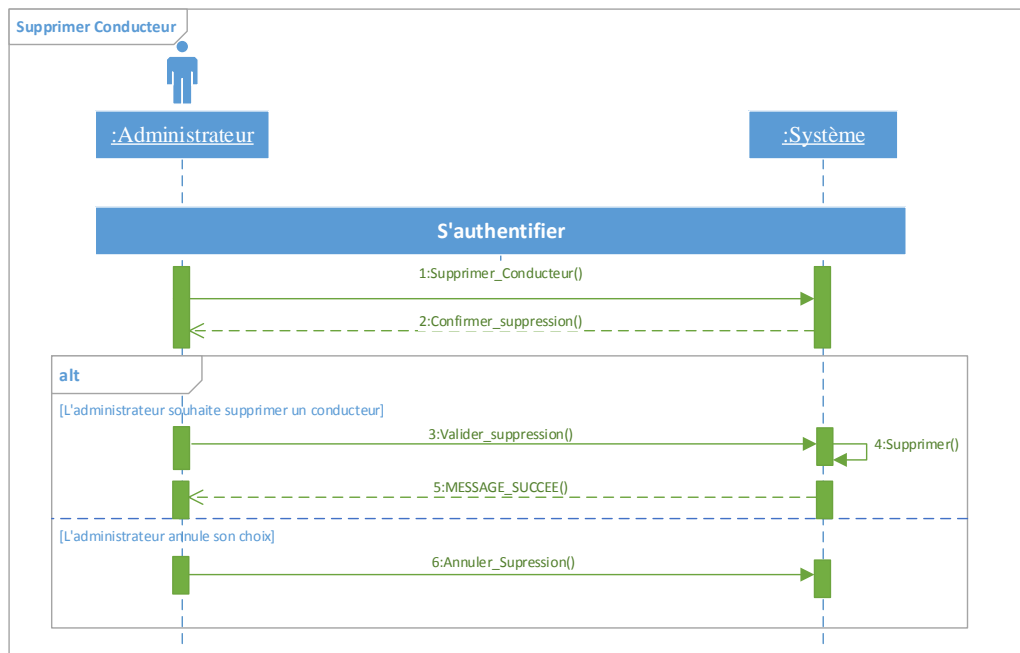
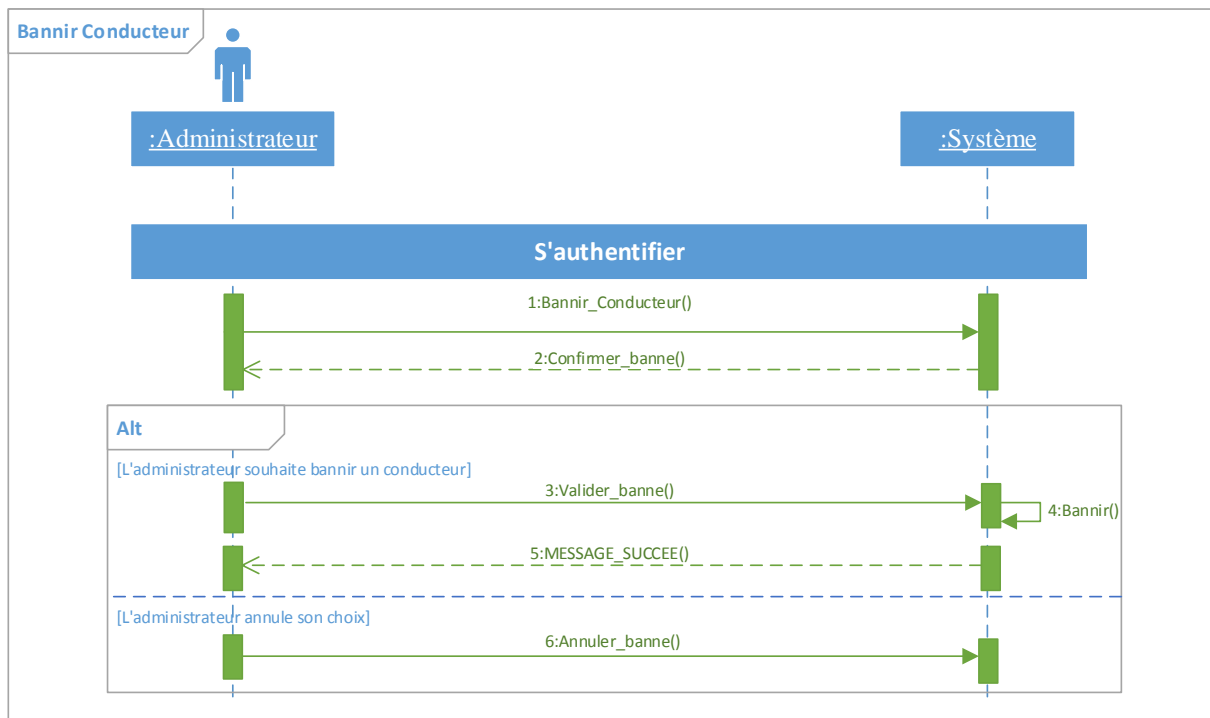


Figure 44: Diagramme de séquence système du cas d'utilisation "Supprimer conducteur"

## c. Diagramme de séquence système du cas d'utilisation « Bannir conducteur »

Il est illustré comme suit :



**Figure 45: Diagramme de séquence système du cas d'utilisation "Bannir conducteur".**

#### **d. Diagramme de séquence système du cas d'utilisation « Valider Conducteur »**

Ci-dessous, le schéma représentatif :

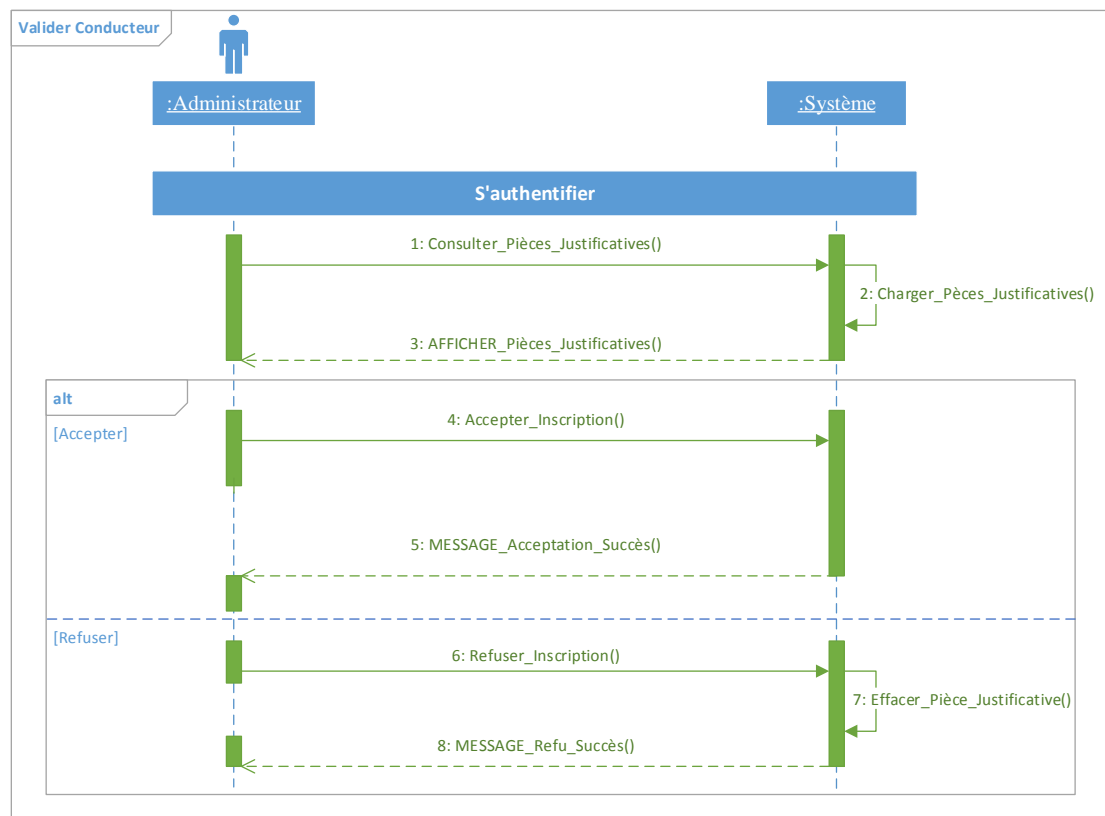


Figure 46: Diagramme de séquence système du cas d'utilisation "Valider Conducteur".

## 2.2. Diagramme des classes d'utilisation

### a. Diagramme de classe du cas d'utilisation « Consulter Conducteur »

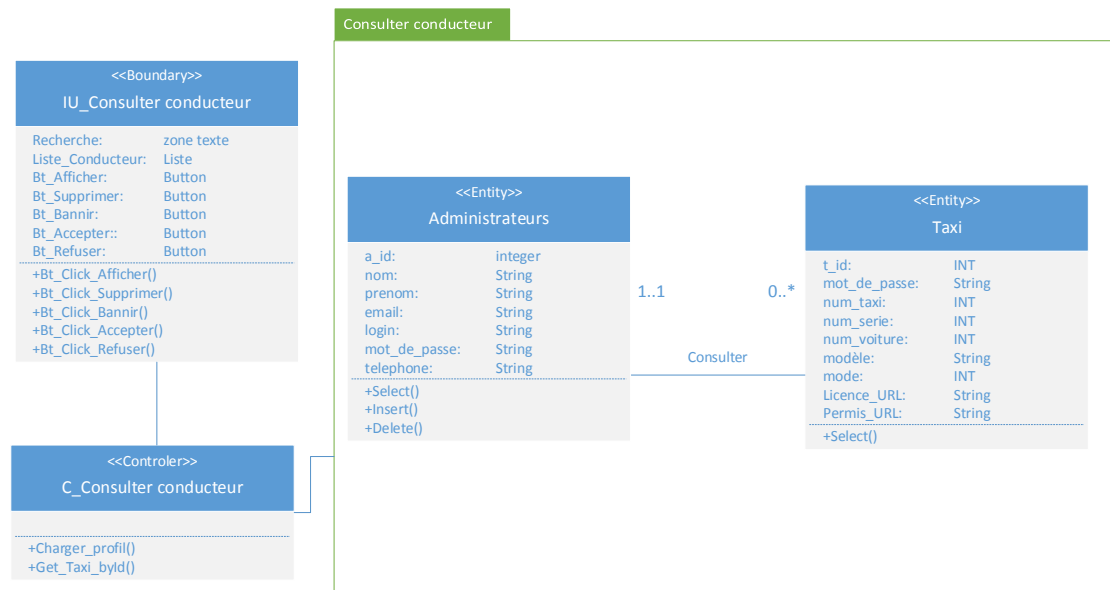


Figure 47: Diagramme de classe de conception "Consulter conducteur".

## b. Diagramme de classe du cas d'utilisation « Supprimer Conducteur »

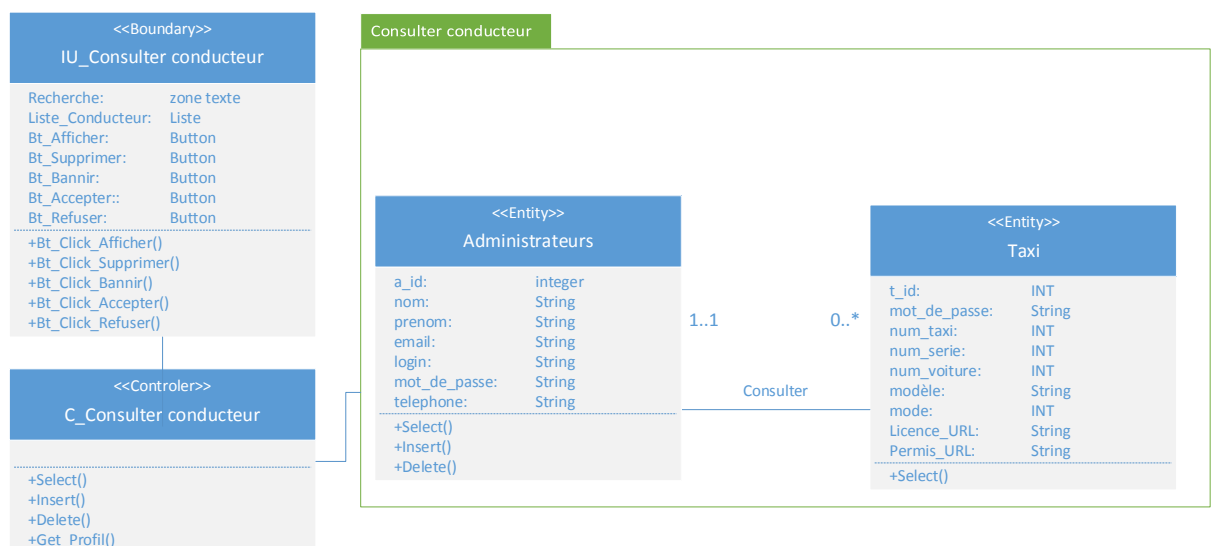


Figure 48: Diagramme de classe de conception du cas d'utilisation "Supprimer conducteur".

### c. Diagramme de classe du cas d'utilisation « Bannir conducteur »

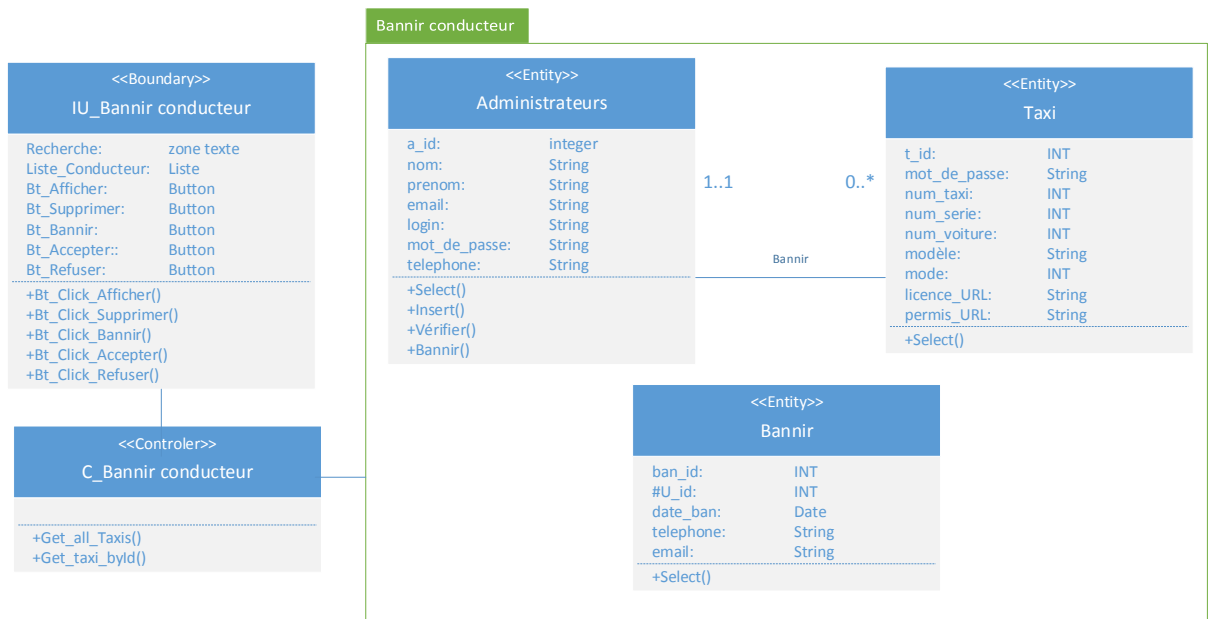


Figure 49: Diagramme de classe de conception "Bannir conducteur".

### d. Diagramme de classe du cas d'utilisation « Valider Conducteur »

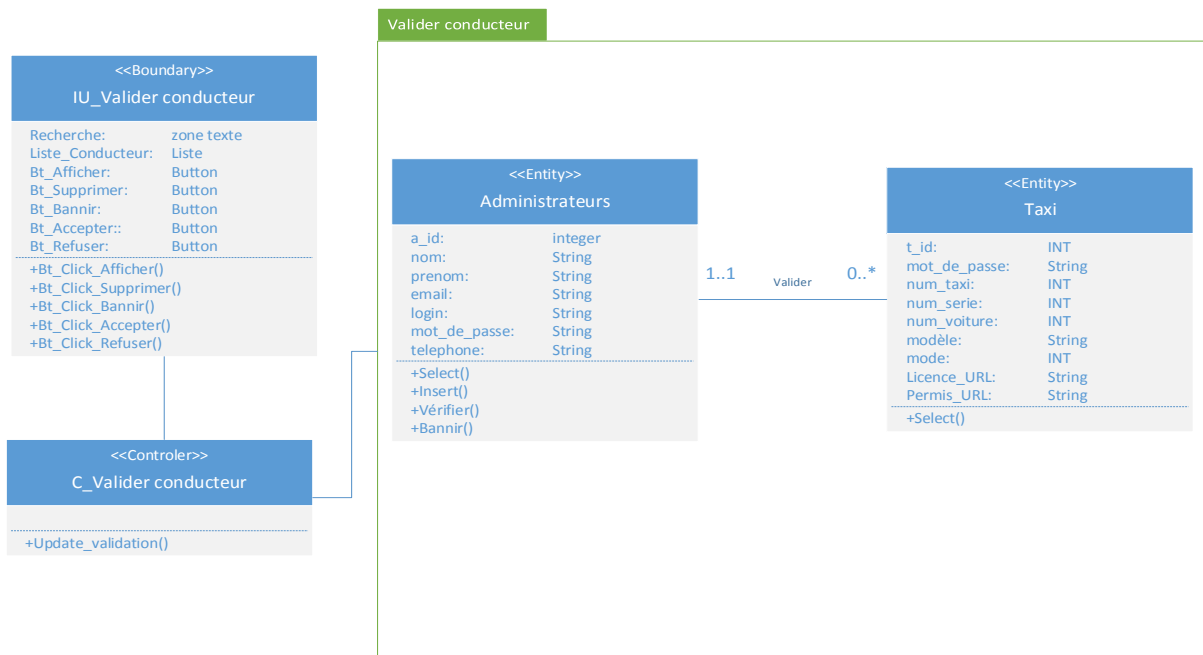


Figure 50: Diagramme de classe de conception du cas d'utilisation "Valider Conducteur".

### 2.3. Diagramme de séquences détaillées

#### a. Diagramme de séquence détaillé du cas d'utilisation « Consulter conducteur »

Lorsque l'administrateur appuie sur le bouton « Consulter conducteur », le système vérifie si le compte conducteur existe dans la table « TAXI ». Si la vérification est valide, le système affiche les coordonnées du conducteur.

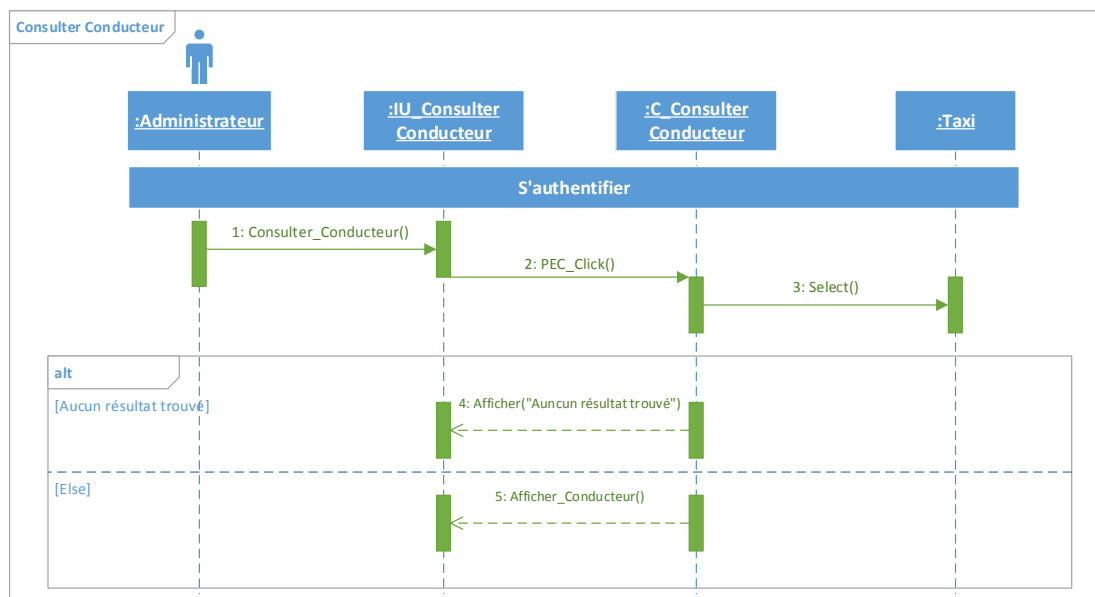


Figure 51: Diagramme de séquence détaillé du cas d'utilisation "Consulter conducteur".

## b. Diagramme de séquence détaillé du cas d'utilisation « Supprimer Conducteur »

Lorsque l'administrateur souhaite supprimer un conducteur, il sélectionne celui-ci parmi la liste des conducteurs qui lui sera affichée sous la rubrique « Lst\_Conducteur » dans l'interface « IU\_Supprimer Conducteur » ; le système demande alors la confirmation de la saisie. Si l'administrateur confirme son choix, le contrôleur « C\_Supprimer Conducteur » efface ce dernier de la table « TAXI ».

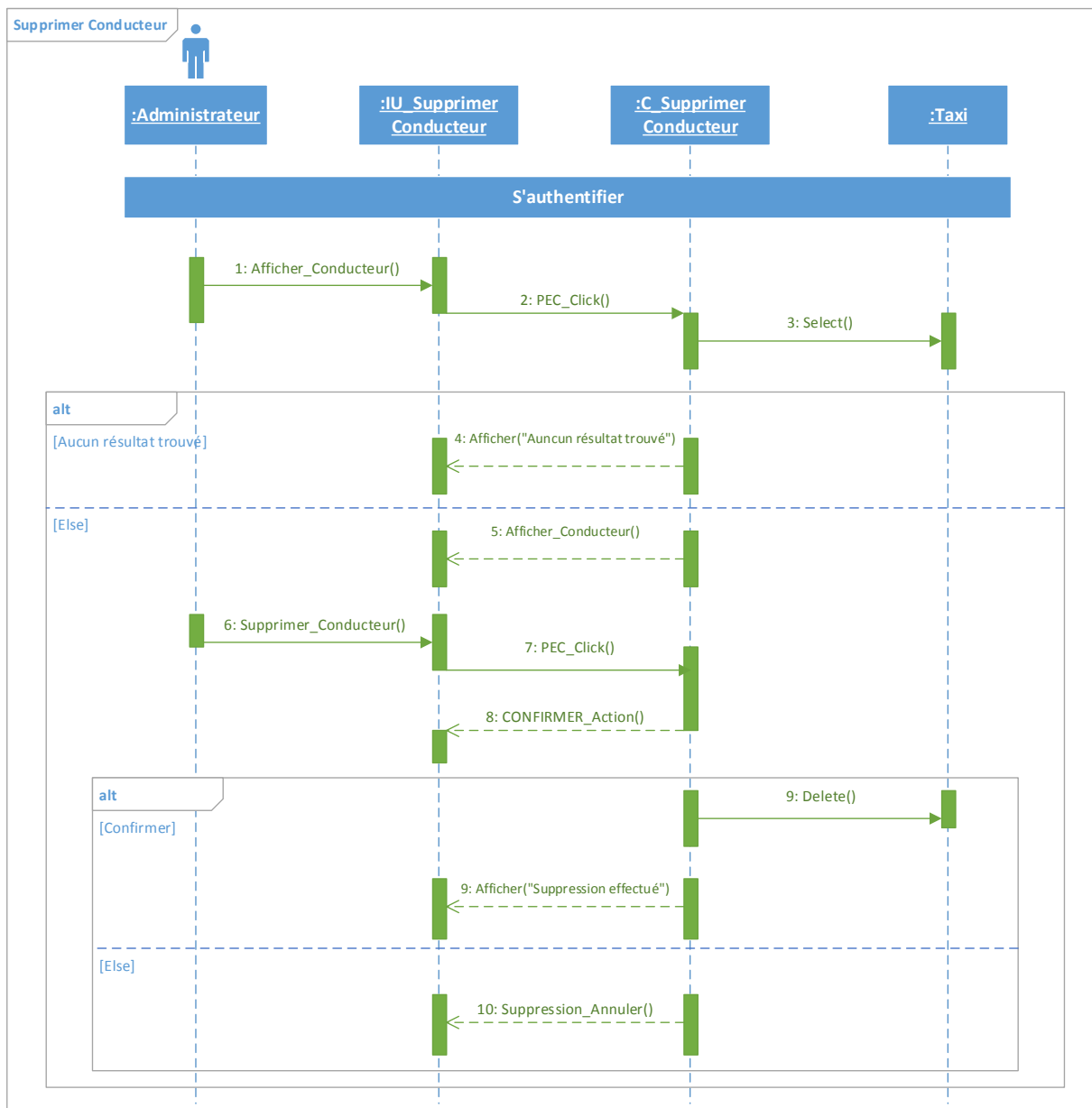


Figure 52: Diagramme de séquence détaillé du cas d'utilisation "Supprimer Conducteur".

### c. Diagramme de séquence détaillé du cas d'utilisation « Bannir Conducteur »

Lorsque l'administrateur souhaite bannir un conducteur, il en choisit un de la liste des conducteurs qui lui sera affichée sur la liste « Lst\_Conducteur » dans l'interface « IU\_Bannir Conducteur » et le système demande une confirmation de la saisie. Si l'administrateur confirme son choix, le contrôleur « C\_Bannir Conducteur » efface ce dernier de la table « Conducteur » et l'enregistre dans la table « Bannir » avec la prise en compte de son numéro de téléphone et son Email pour qu'il ne puisse plus accéder à l'application avec ce compte.

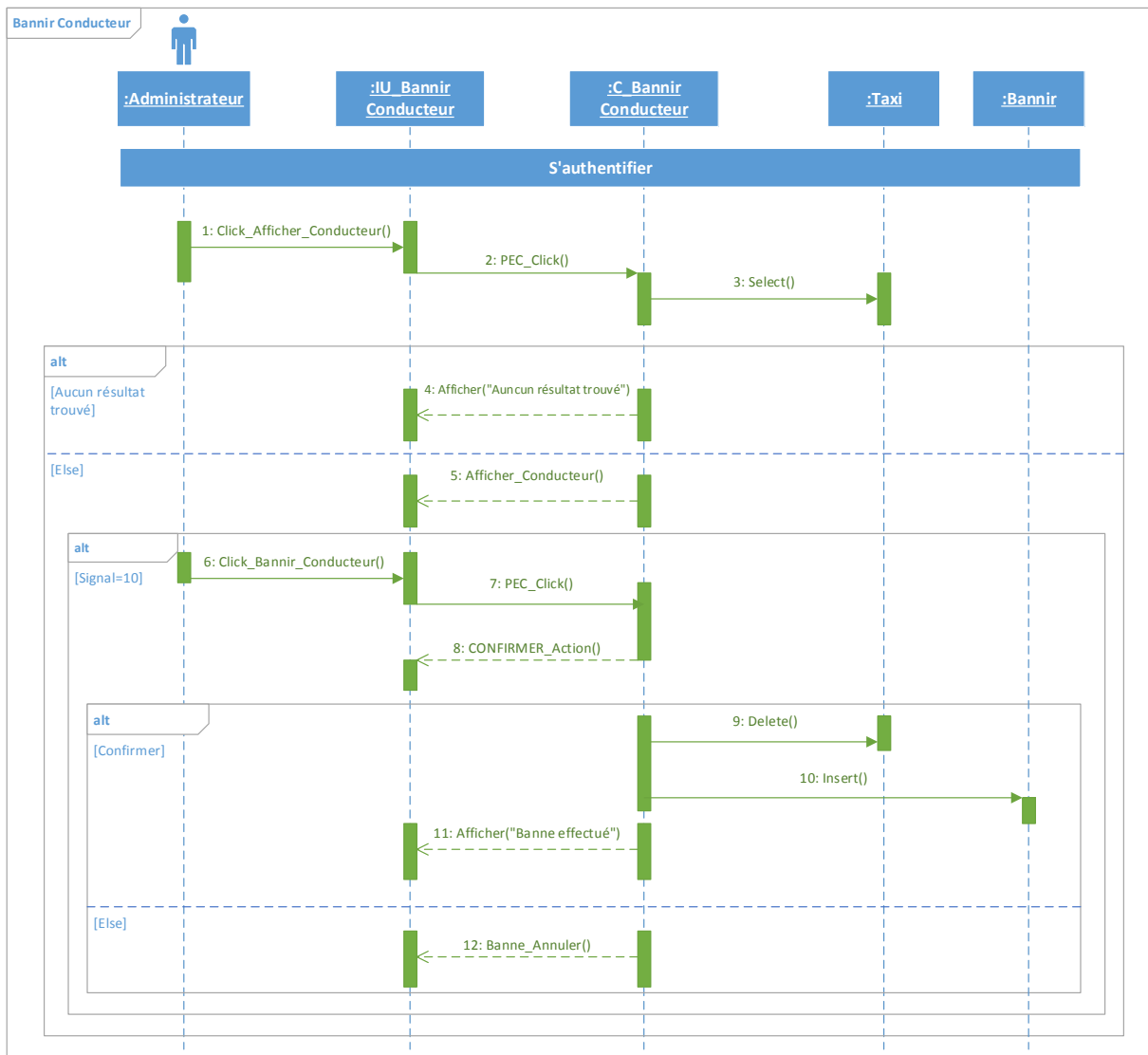


Figure 53: Diagramme de séquence détaillé du cas d'utilisation "Bannir Conducteur".



#### d. Diagramme de séquence détaillé du cas d'utilisation « Valider Conducteur »

L'administrateur vérifie les pièces justificatives et garantit la validité du compte avec une confirmation de la saisie et le champ « Stat\_verif » dans la table « Conducteur » aura alors la valeur « True ». Si l'administrateur refuse le compte, il sera automatiquement supprimé de la table « TAXI ».

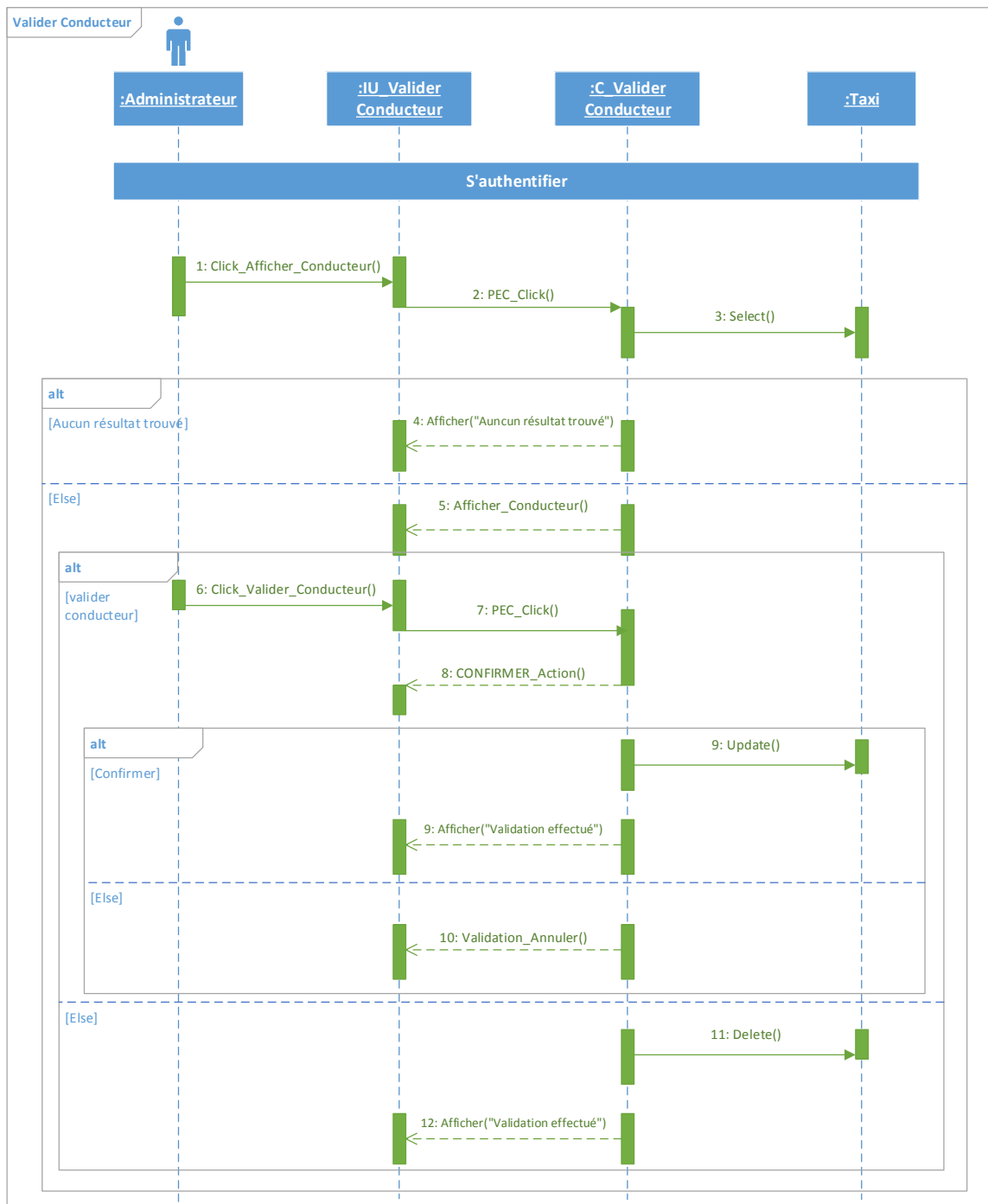


Figure 54: Diagramme de séquence détaillé du cas d'utilisation "Valider Conducteur".

## 2.4. Diagramme des classes de conception :

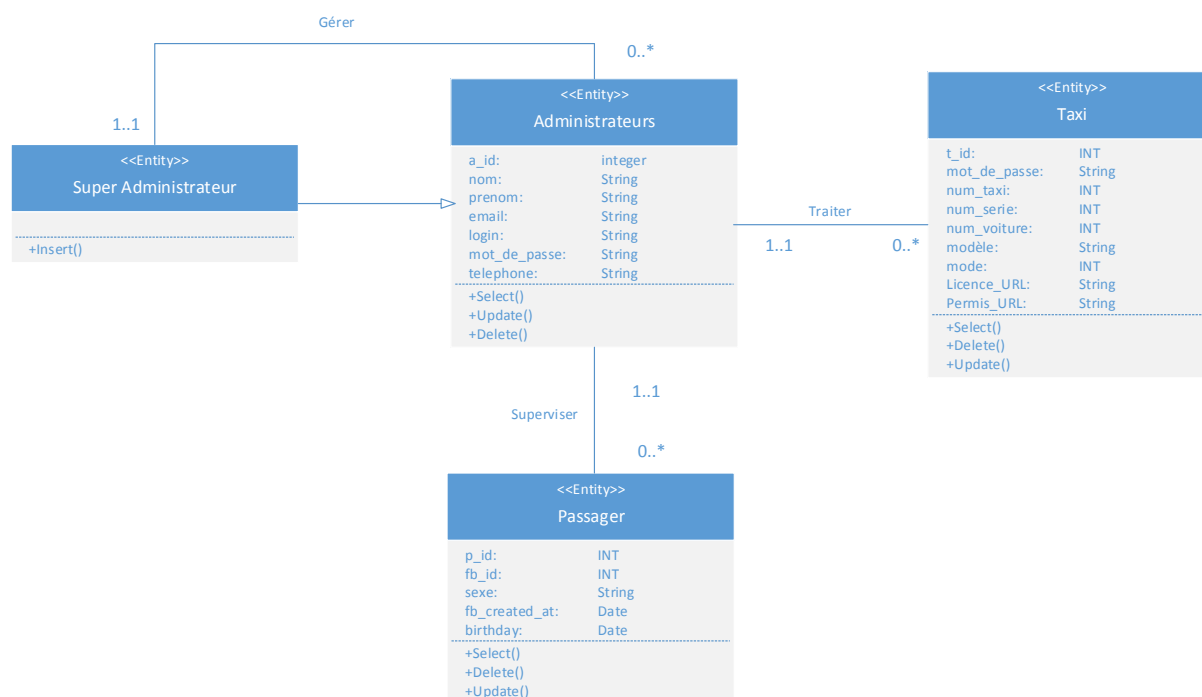


Figure 55: Diagramme de classe de conception du sprint 1.

## 3. Codage :

Après avoir construit le nouveau diagramme de classe pour ce sprint et en appliquant les règles de passage vers le schéma logique de l'application, nous obtenons le schéma de la base de données suivant :

Champs	Types	Contraintes
id	INT	PRIMARY KEY
nom	VARCHAR (20)	---
prenom	VARCHAR (20)	---
email	VARCHAR (20)	UNIQUE
mot_de_Passe	VARCHAR (20)	---
num_taxi	INT	---
num_série	INT	---
num_voiture	INT	---
modèle	VARCHAR (20)	---
stat_mode	INT	---
telephone	VARCHAR(8)	---

<b>stat_Tel</b>	BOOLEAN	---
<b>created_at</b>	DATE	---
<b>updated_at</b>	DATE	---
<b>licence_URL</b>	VARCHAR (20)	---
<b>permis_URL</b>	VARCHAR (20)	---
<b>latitude</b>	FLOAT (10,6)	---
<b>longitude</b>	FLOAT (10,6)	---
<b>note</b>	VARCHAR (20)	---

**Tableau 28: Structure de la table "taxi".**

Champs	Types	Contraintes
<b>id</b>	INT	PRIMARY KEY
<b>nom</b>	VARCHAR (20)	---
<b>prenom</b>	VARCHAR (20)	---
<b>email</b>	VARCHAR (20)	UNIQUE
<b>telephone</b>	VARCHAR(8)	---
<b>Stat_Telephone</b>	BOOLEAN	---
<b>created_at</b>	DATE	---
<b>fb_created_at</b>	Date	---
<b>latitude</b>	FLOAT (10,6)	---
<b>Longitude</b>	FLOAT (10,6)	---
<b>Note</b>	INT	---

**Tableau 29: Structure de la table "Passager".**

Champs	Types	Contraintes
<b>a_id</b>	INT	PRIMARY KEY
<b>nom</b>	VARCHAR (20)	---
<b>prenom</b>	VARCHAR (20)	---
<b>Login</b>	VARCHAR (20)	---
<b>Mot_de_passe</b>	VARCHAR (20)	---
<b>email</b>	VARCHAR (20)	UNIQUE
<b>telephone</b>	VARCHAR(8)	---

**Tableau 30: Structure de la table "Administrateur".**

## II. Deuxième Sprint

En partant du même principe que le sprint précédent, je commence par définir le but de mon second sprint. Suite à une conversation entre le Product Owner et l'équipe Scrum, nous avons décidé le but suivant : « terminer la partie qui concerne la gestion du compte administrateur ».

Une fois que le but de notre sprint est défini, il est temps de décider quelles histoires inclure dans ce dernier. Le tableau ci-dessous résume le Backlog de mon quatrième sprint.

Histoire utilisateur	Priorité
Ajouter un administrateur	Elevée
Modifier un administrateur	Moyenne
Consulter un administrateur	Moyenne
Supprimer un administrateur	Moyenne
Consulter nombre conducteur	Moyenne
Consulter nombre passager	Moyenne
Consulter nombre de réservations effectué	Moyenne
Consulter nombre de demandes refusé	Moyenne
Consulter nombre de plaintes	Faible

Tableau 31: Backlog du quatrième sprint (release 2)

### 5. Spécification fonctionnelle :

La spécification fonctionnelle dans notre cas se traduit par le diagramme des cas d'utilisation d'UML et la description textuelle de ces derniers.

#### 5.1. Scénario et cas d'utilisation

Après avoir effectué mon Backlog, je peux maintenant l'explicitier sous forme de tableaux :

Chaque tableau présente la description détaillée du fonctionnement de tous les cas d'utilisation.

Dans ce Sprint nous allons nous intéresser aux deux cas d'utilisations suivants :

Gérer compte Administrateur



Consulter les statistiques

## 5.2. Description textuelle des cas d'utilisation

Je vais maintenant décrire chacun des cas d'utilisation énumérés dans le Backlog en identifiant les acteurs et les différents scénarios possible.

### 5.2.1. Description du cas d'utilisation « Gérer compte Administrateurs »

#### ➤ Description du sous cas d'utilisation « Consulter Administrateur »

Cas d'utilisation :	<b>Consulter Administrateurs</b>
Acteur :	Super-Administrateur
Pré condition :	- Le Super-administrateur s'est authentifié.
Post condition :	- Le Super-administrateur a consulté le profil.
Description du scénario principal :	<ul style="list-style-type: none"><li>- L'administrateur saisit son login et son mot de passe pour accéder à l'interface.</li><li>- L'administrateur choisit l'option « Consulter profil » pour consulter les profils des autres éventuels administrateurs</li><li>- Le system affiche un tableau qui contient tous les administrateurs avec la possibilité de trier le résultat.</li><li>- L'administrateur sélectionne un profil et clique sur le bouton « Voir ».</li><li>- Le système charge les informations de l'administrateur.</li><li>- Le système affiche les informations.</li></ul>
Exception :	<ul style="list-style-type: none"><li>- L'administrateur annule la demande.</li><li>- Le système annule le chargement des informations.</li><li>- Le système affiche l'interface précédente.</li></ul>

Tableau 32: Description du sous cas d'utilisation "Consulter Administrateur".

➤ **Description du sous cas d'utilisation « Supprimer Administrateur »**

Cas d'utilisation :	<b>Supprimer Administrateur</b>
Acteur :	Super-administrateur
Pré condition :	<ul style="list-style-type: none"> <li>- Le Super-administrateur s'est authentifié.</li> <li>- L'administrateur est existant</li> </ul>
Post condition :	<ul style="list-style-type: none"> <li>- L'administrateur a supprimé le profil.</li> </ul>
Description du scénario principal :	<ul style="list-style-type: none"> <li>- Le Super-administrateur saisit son login et son mot de passe pour accéder à l'interface.</li> <li>- Le Super-administrateur choisit l'option « Supprimer profil » après avoir cliqué sur le bouton « Administrateur » dans le menu précédent.</li> <li>- Le Super-administrateur sélectionne le profil à supprimer.</li> <li>- Le système affiche toutes les informations de l'administrateur.</li> <li>- Le Super-administrateur clique sur le bouton « supprimer » et confirme son choix.</li> <li>- Le système supprime le compte.</li> </ul>
Exception :	<ul style="list-style-type: none"> <li>- Le Super-administrateur annule son choix.</li> <li>- Le système annule la suppression de l'administrateur</li> </ul>

**Tableau 35 : Description du sous cas d'utilisation "Supprimer Administrateur".**

➤ **Description du sous cas d'utilisation « Ajouter Administrateur »**

Cas d'utilisation :	<b>Ajouter Administrateur</b>
Acteur :	Super-administrateur
Pré condition :	<ul style="list-style-type: none"> <li>- Le Super-administrateur s'est authentifié.</li> </ul>
Post condition :	<ul style="list-style-type: none"> <li>- Le Super-administrateur a ajouté un administrateur.</li> </ul>
Description du scénario principal :	<ul style="list-style-type: none"> <li>- Le Super-administrateur choisit l'option « Ajouter administrateur » pour ajouter un compte administrateur.</li> <li>- Le system affiche le formulaire d'ajout.</li> <li>- Le Super-administrateur remplit le formulaire par les informations adéquates et confirme sa saisie.</li> <li>- Le système enregistre les informations saisies.</li> <li>- Le système affiche « Ajout effectué ».</li> </ul>
Exception :	<ul style="list-style-type: none"> <li>- Le Super-administrateur annule la demande.</li> <li>- Le système annule le chargement des informations.</li> <li>- Le système affiche l'interface précédente.</li> </ul>

**Tableau 36 : Description du sous cas d'utilisation "Ajouter Administrateur".**

➤ **Description du sous cas d'utilisation « Modifier Administrateur »**

Cas d'utilisation :	<b>Modifier Administrateur</b>
Acteur :	Super-administrateur
Pré condition :	- Le Super-administrateur s'est authentifié.
Post condition :	- Le Super-administrateur a modifié un administrateur.
Description du scénario principal :	<ul style="list-style-type: none"> <li>- Le Super-administrateur choisit l'option « Modifier profil » pour modifier les profils des autres administrateurs ainsi que le sien.</li> <li>- Le system affiche un tableau qui contient tous les administrateurs avec la possibilité de trier le résultat.</li> <li>- Le Super-administrateur sélectionne un profil et clique sur le bouton « Modifier ».</li> <li>- Le système charge les informations de l'administrateur.</li> <li>- Le système affiche les informations.</li> <li>- Le Super-administrateur saisit les changements qu'il souhaite faire.</li> <li>- Le système enregistre les changements.</li> </ul>
Exception :	<ul style="list-style-type: none"> <li>- Le Super-administrateur annule la demande.</li> <li>- Le système annule le chargement des informations.</li> <li>- Le système affiche l'interface précédente.</li> </ul>

**Tableau 33:Description du sous cas d'utilisation "Modifier Administrateur".**

### 5.2.2. Description du cas d'utilisation « Gérer Statistique »

Cas d'utilisation :	<b>Gérer statistique</b>
Acteur :	Administrateur
Pré condition :	L'administrateur s'est authentifié.
Post condition :	L'administrateur a consulté les statistiques.
Description du scénario principal :	<ul style="list-style-type: none"> <li>- L'administrateur saisit son login et son mot de passe pour accéder à son interface.</li> <li>- L'administrateur appuie sur « Consulter nombre conducteur » pour voir le pourcentage de conducteur qui se connecte à l'application.</li> <li>- Le système affiche le nombre de conducteur connectés et ainsi que le pourcentage depuis leur première utilisation de l'application.</li> <li>- L'administrateur appuie sur « Consulter nombre passager » pour voir le pourcentage de passager qui se connecte à l'application.</li> <li>- Le système affiche le nombre de passager connectés ainsi que le pourcentage depuis leur première utilisation de l'application.</li> <li>- L'administrateur appuie sur « Consulter nombre de réservations effectuées » pour voir le pourcentage de réservations effectuées.</li> <li>- Le système affiche le pourcentage de réservation effectuée.</li> <li>- L'administrateur appuie sur « Consulter nombre de plaintes » pour voir le pourcentage de plaintes existantes.</li> <li>- Le système affiche le pourcentage des nombres de plaintes.</li> </ul>
Exception :	Si l'administrateur ajoute un compte déjà existant, le système affiche le message « Compte déjà existant ».

Tableau 34: Description du cas d'utilisation "Gérer statistique".

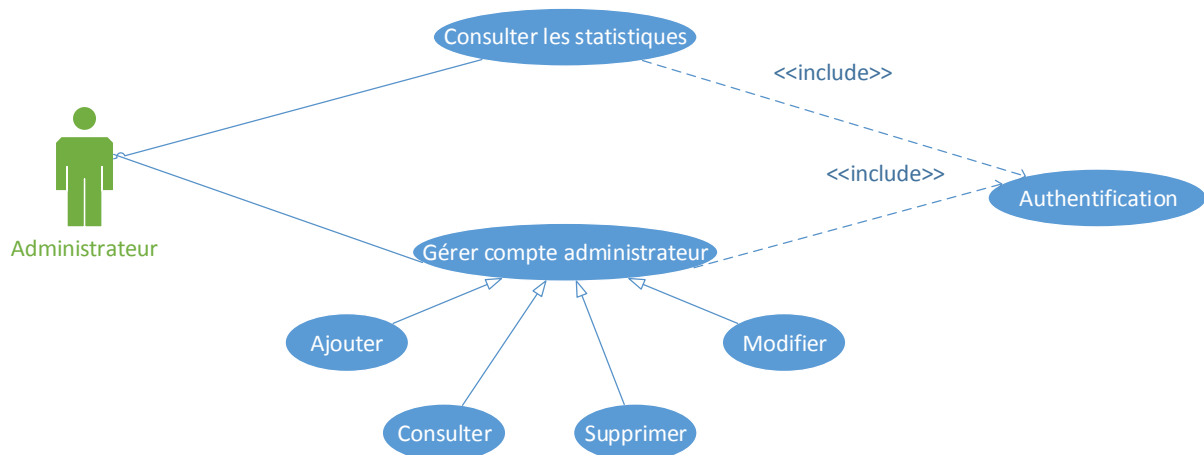


Figure 56: Description du sprint 2 (release 2).



## 6. Conception

J'effectue à présent les diagrammes de séquence système des différents cas d'utilisation déjà expliqués dans la section précédente.

### 6.1. Diagramme de séquence système

En me référant aux descriptions textuelles dans la section précédente, je présente les diagrammes de séquences systèmes adéquats. Sur cette base j'expose les cas d'utilisation vu précédemment.

#### a. Diagramme de séquence système du sous cas « Consulter administrateur »

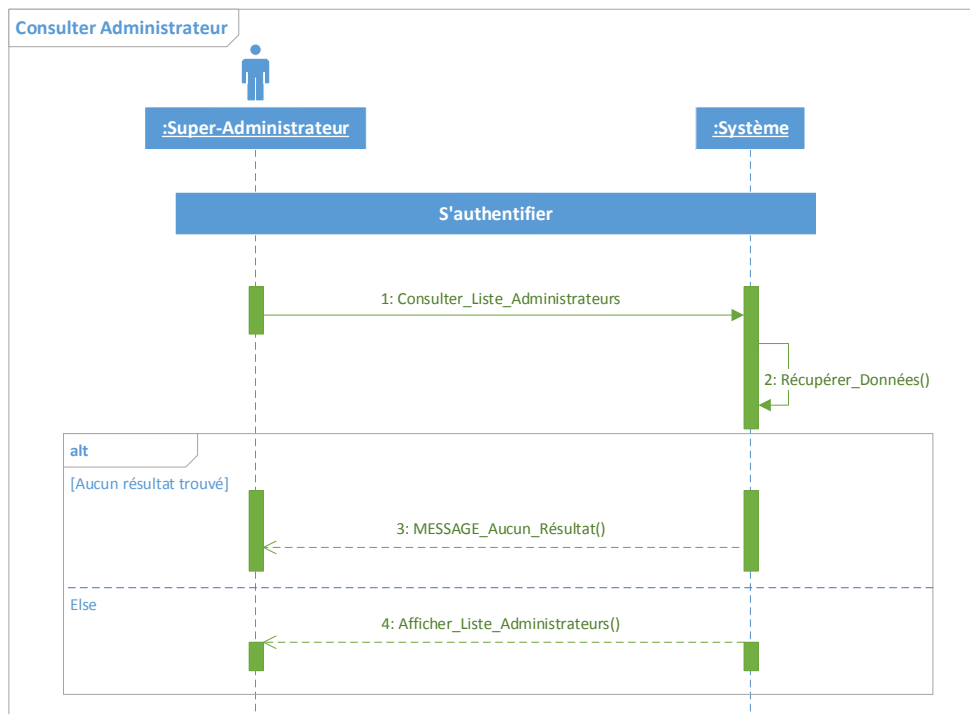
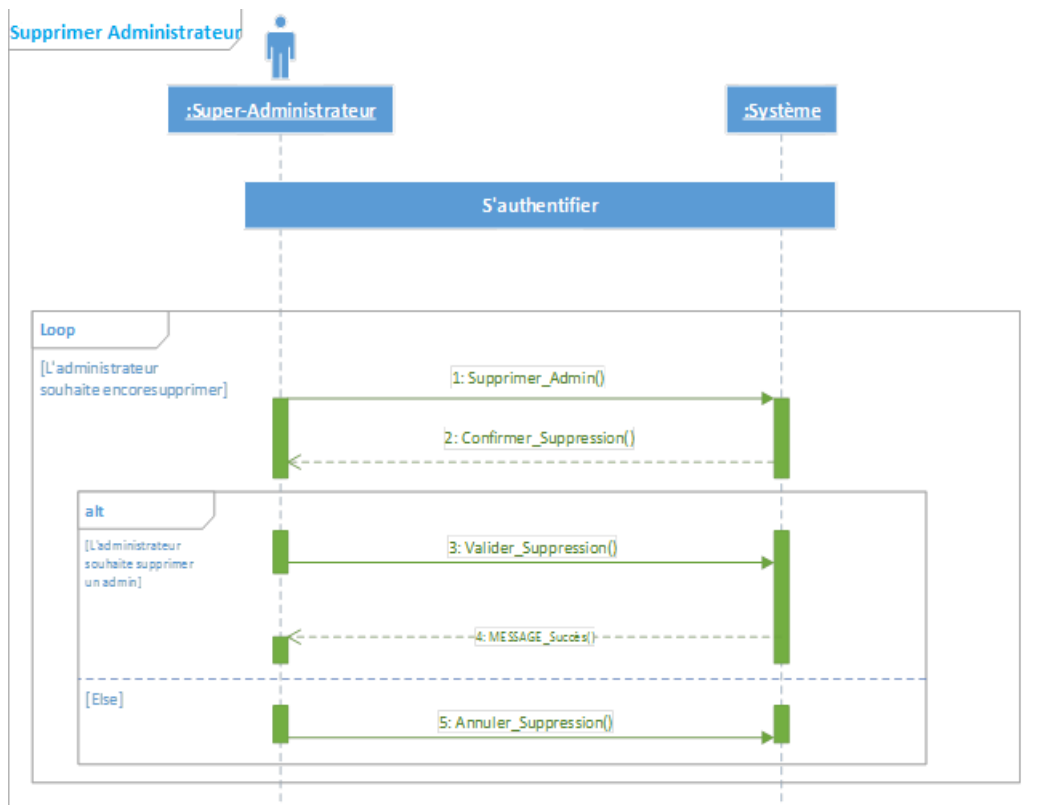


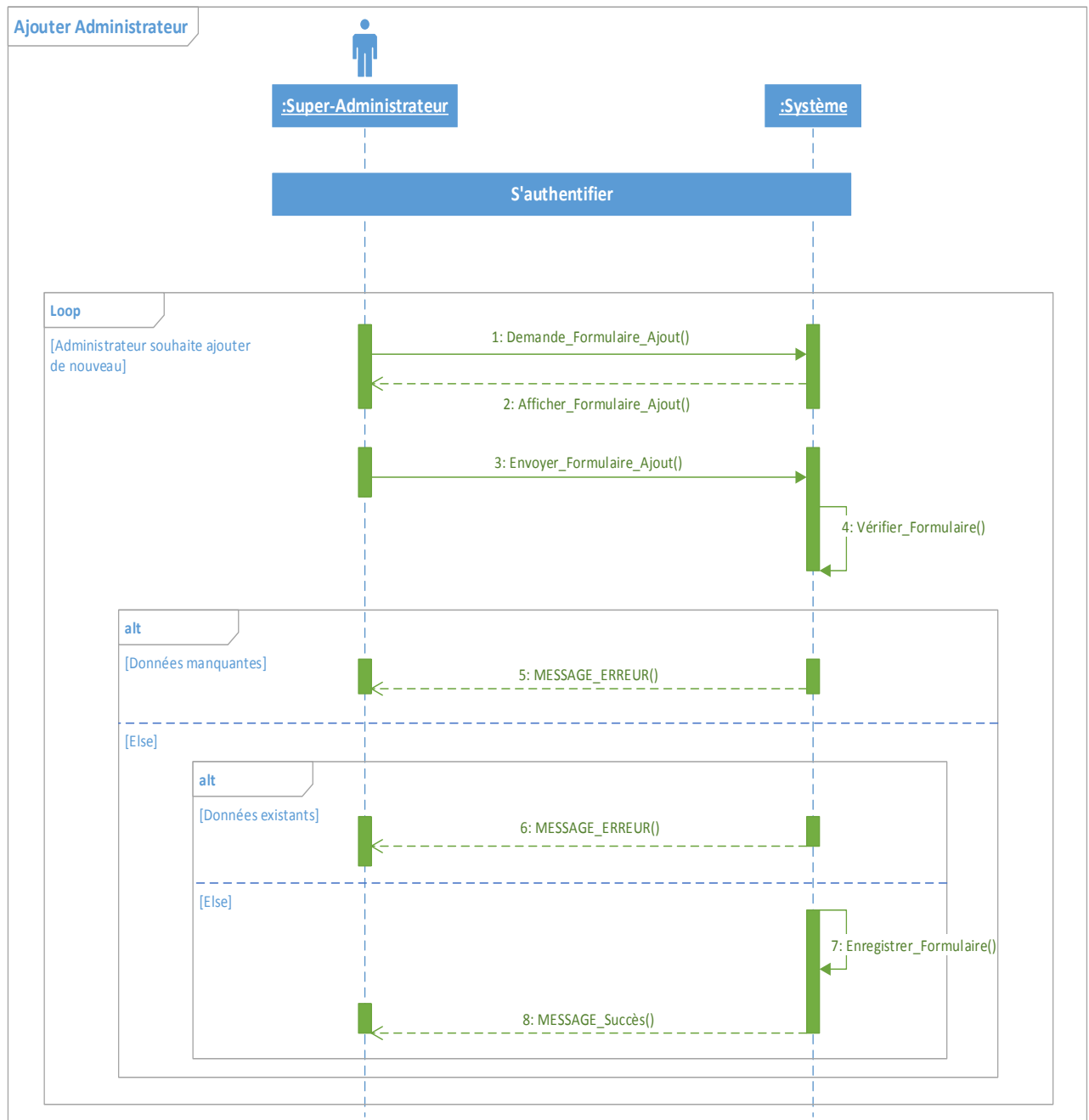
Figure 57: Diagramme de séquence système du sous cas d'utilisation "Ajouter Administrateur".

**b. Diagramme de séquence système du sous cas « Supprimer Administrateur »**



**Figure 58: Diagramme de séquence détaillé du sous cas d'utilisation "Supprimer Administrateur".**

**c. Diagramme du sous cas système du sous cas d'utilisation « Ajouter Administrateur »**



**Figure 59: Diagramme de séquence système du sous cas d'utilisation "Ajouter Administrateur".**

## 6.2. Diagramme des classes de conception

A ce stade, je présente les interfaces « IU\_Interface », et les tables « Entity » cible pour expliquer le déroulement des cas vus précédemment dans le diagramme de séquence système. Par suite, dans la section suivante, j'illustre les différentes tables concernées.

### a. Diagramme de classe du cas d'utilisation « Consulter compte administrateur »

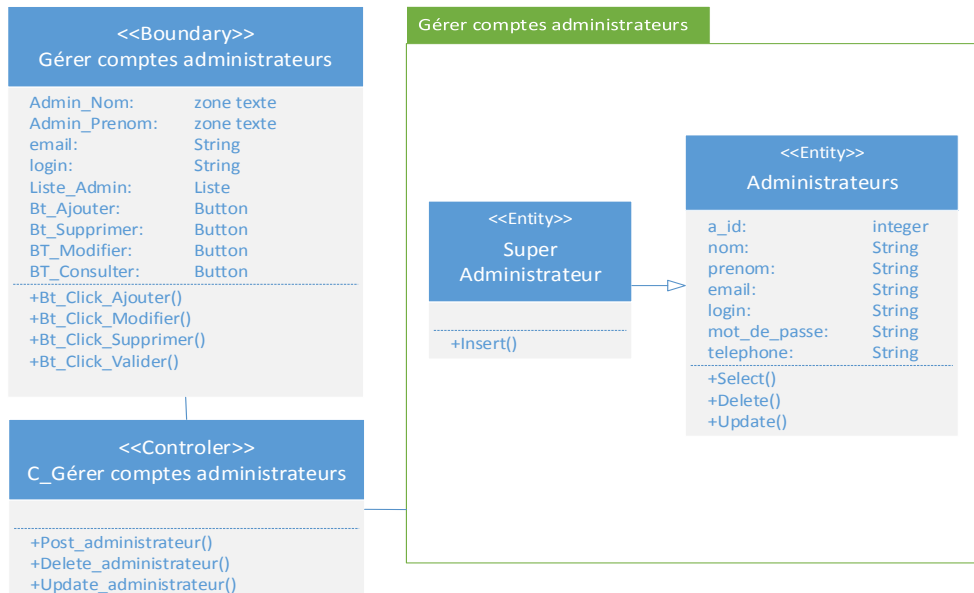


Figure 60: Diagramme de classe de conception du cas d'utilisation "Gérer comptes administrateur".

## b. Diagramme de classe du cas d'utilisation « Gérer statistique »

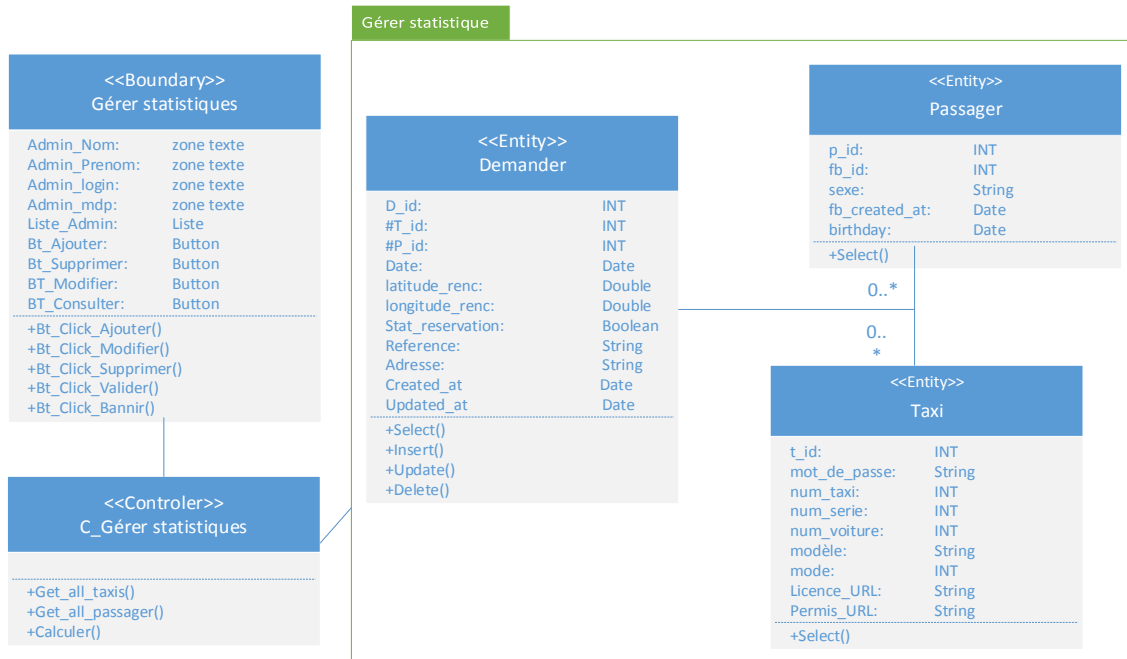


Figure 61: Diagramme de classe de conception du cas d'utilisation "Gérer statistique".

### 6.3. Diagramme de séquence détaillé

#### a. Diagramme de séquence détaillé du sous cas d'utilisation « Ajouter Administrateur »

Lorsque le Super-administrateur souhaite ajouter un administrateur, il remplit le formulaire d'ajout affiché sur l'interface « IU\_Ajouter Administrateur ». Une fois qu'il valide son choix, le contrôleur « C\_Ajouter Administrateur » consulte la base « Administrateur » et vérifie si le login est existant.

En cas d'absence dans la table, l'ajout du compte administrateur est alors effectué, sinon un message d'erreur sera affiché au Super-administrateur.

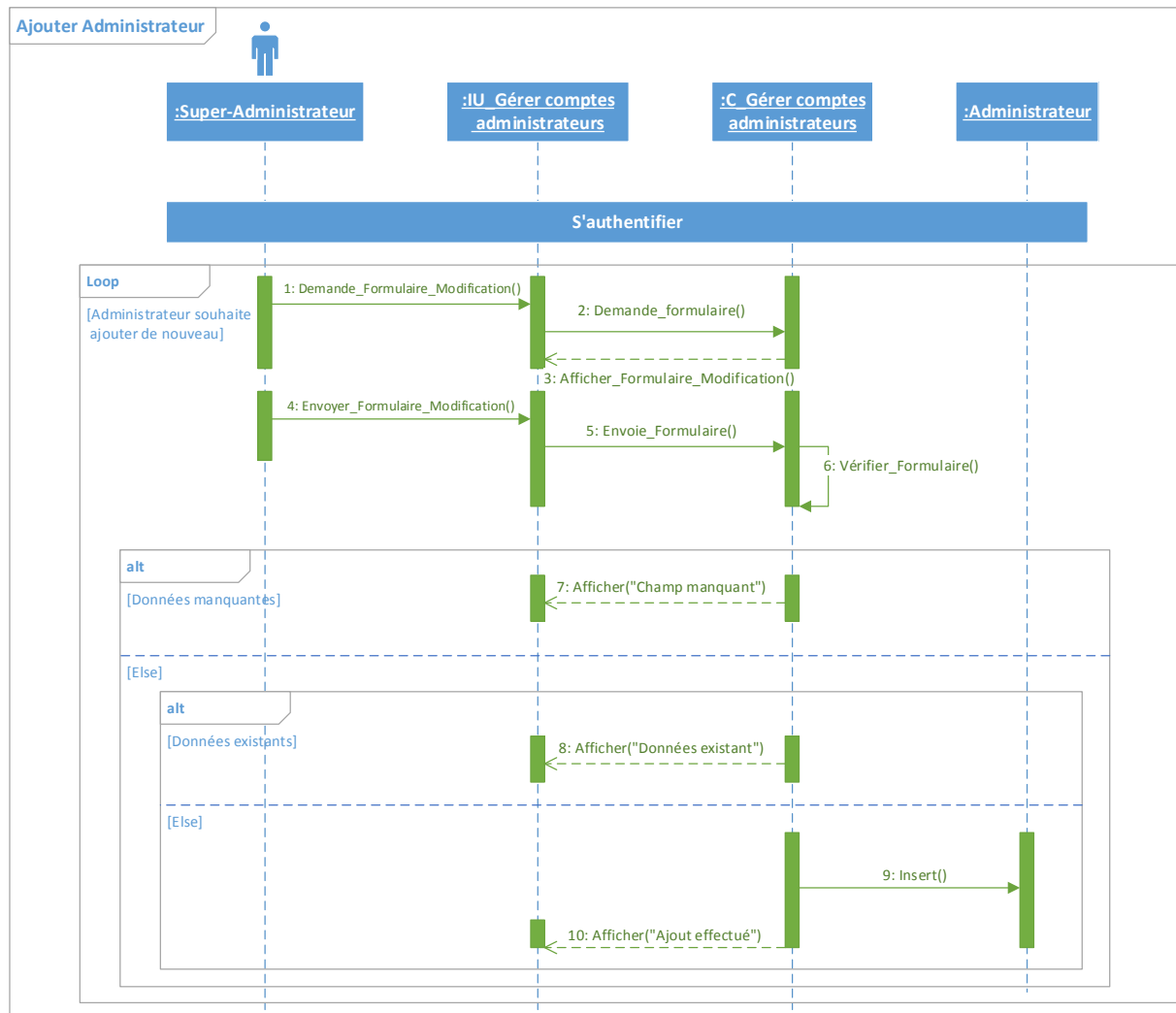


Figure 62: Diagramme de séquence détaillé du sous cas d'utilisation "Ajouter Administrateur".

## b. Diagramme de séquence détaillé du sous cas d'utilisation « Supprimer Administrateur »

Lorsqu'un Super-administrateur souhaite supprimer un compte administrateur, il en sélectionne un dans la liste des administrateurs affichée sous la rubrique « Lst\_Admin » dans l'interface « IU\_Supprimer Administrateur ». Une fois que le Super-administrateur confirme son choix, le contrôleur « C\_Supprimer Administrateur » efface ce dernier de la table « Administrateur ».

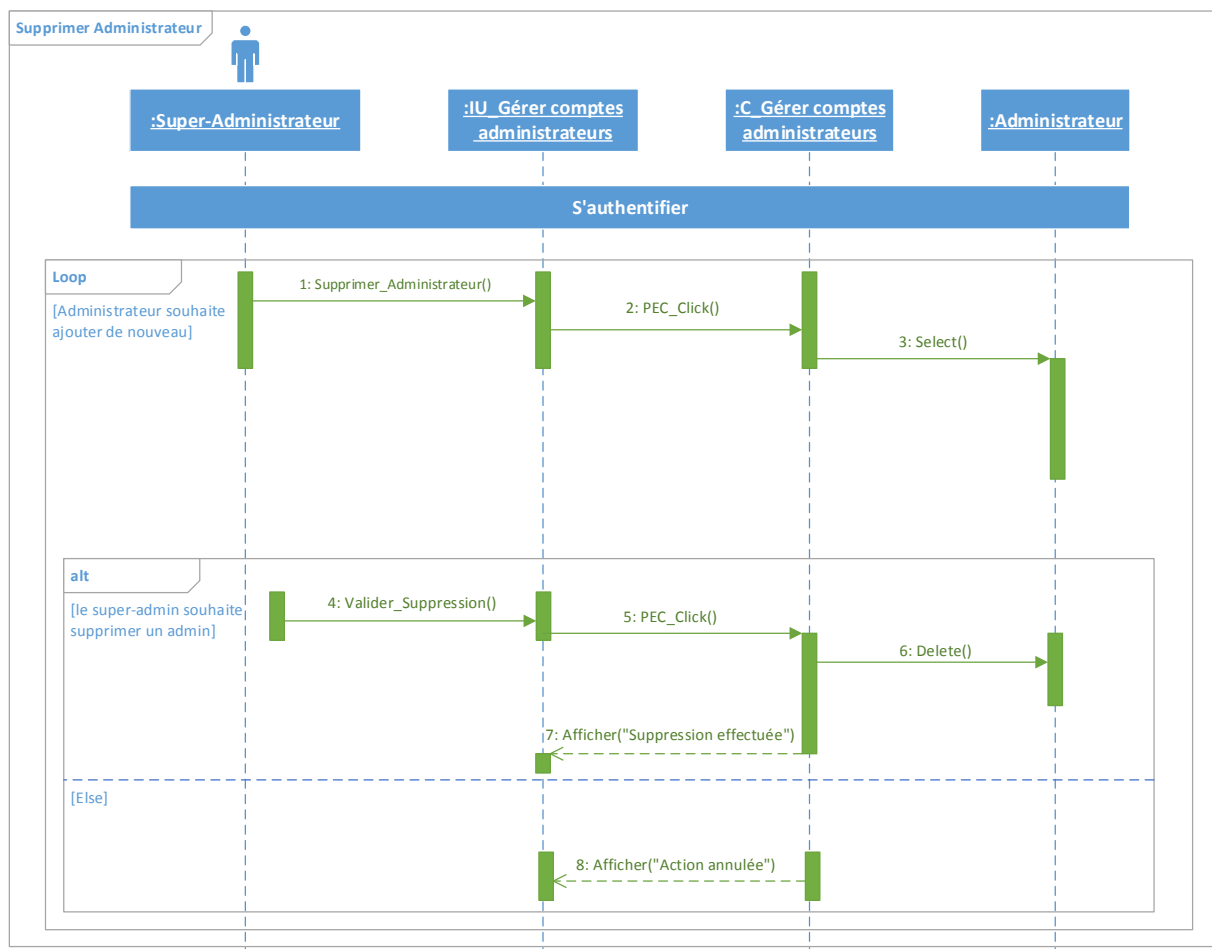
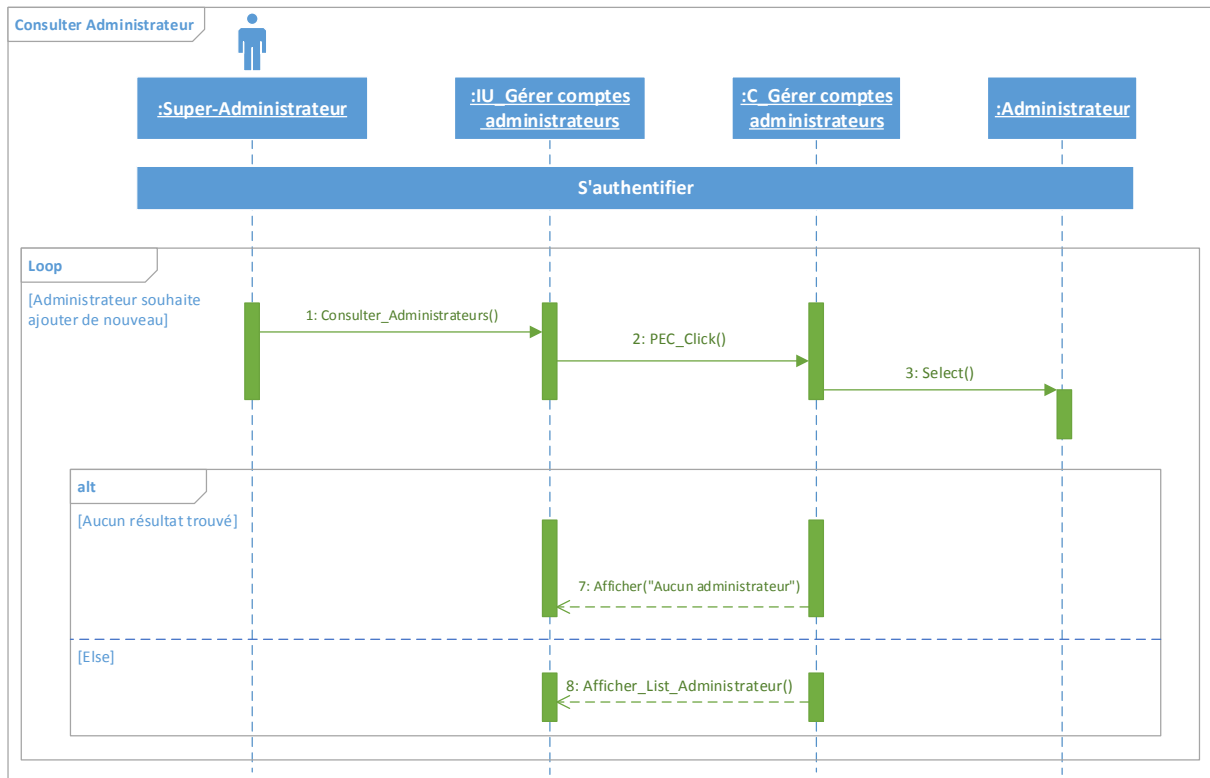


Figure 63: Diagramme de classe de conception du sous cas d'utilisation "Supprimer Administrateur".

### c. Diagramme de séquence détaillé du sous cas d'utilisation « Consulter Administrateur »

Lorsque le Super-administrateur appuie sur le bouton « Consulter administrateur », le système vérifie si le compte administrateur existe dans la base « Administrateur ». Si la vérification est valide, le système affiche les coordonnées de ce dernier.



**Figure 64: Diagramme de séquence de conception du sous cas d'utilisation "Consulter Administrateur".**



## 6.4. Diagramme de classes de conception :

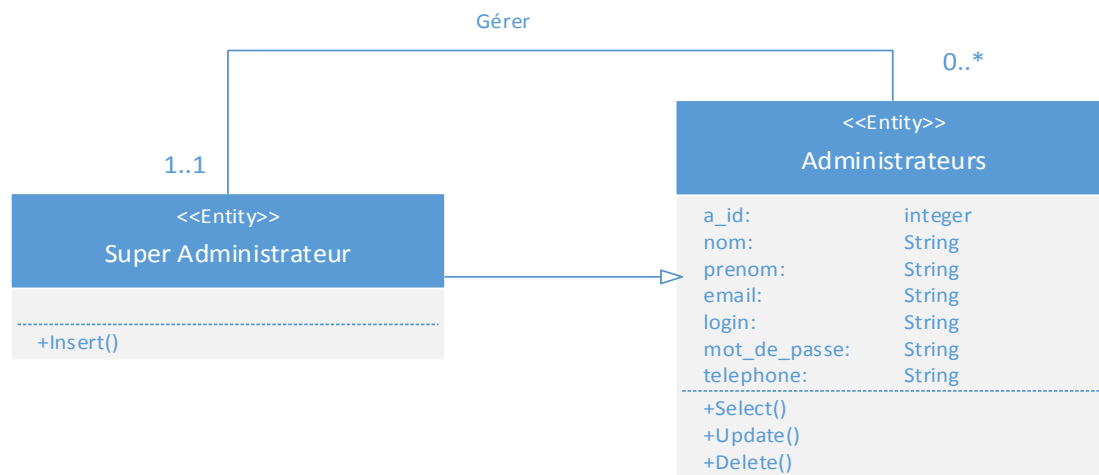


Figure 65: Diagramme de classe de conception du sprint 2.

## III. Les tests de l'application mobile :

A ce stade je présente l'interface administrateur complète où il pourra effectuer toutes ses tâches.

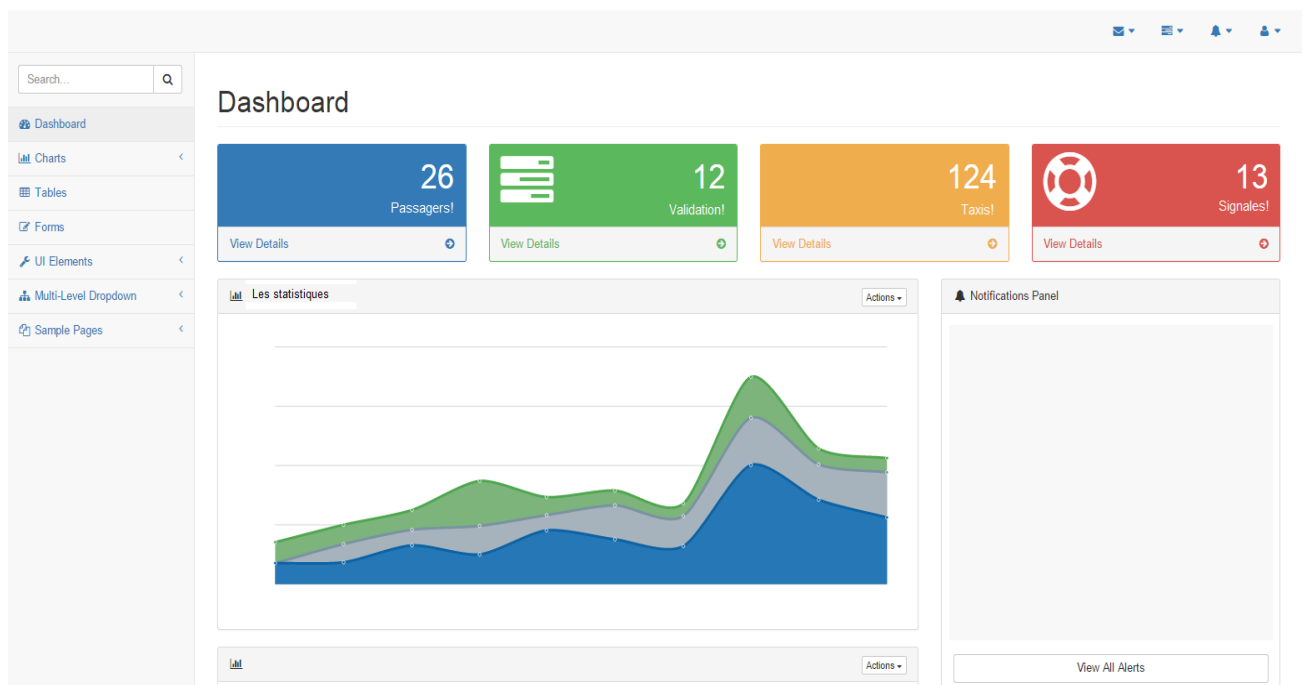


Figure 66: Interface administrateur.

## Conclusion

Une fois le deuxième release est ficelé, l'administrateur peut désormais effectuer la supervision des comptes.

A ce stade j'ai réussi à concevoir et développer un produit complet et fonctionnel.

---

# *Chapitre V : La phase de clôture*

---

## Introduction

La phase clôture est la dernière phase dans le cycle de développement d'un logiciel avec Scrum. Cette phase est souvent appelée sprint de stabilisation [6]. Les tâches effectuées pendant cette phase ne sont pas claires, et elles dépendent fortement du type de déploiement du logiciel (mise en production à chaud, packaging du produit, mise à disposition par téléchargement en ligne...).

Dans cette partie, je vais décrire l'environnement de travail qui m'a permis de mettre en œuvre la conception abordée dans le chapitre précédent.

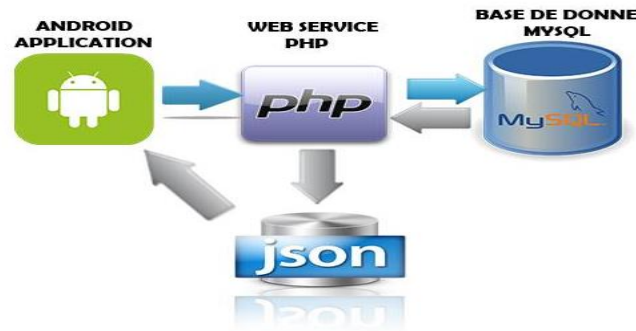
### 1. Environnement de travail :

#### 1.1. Choix de l'architecture de l'application

La méthode la plus répandue pour se connecter à une base de données MySQL à distance à partir d'un appareil mobile, est de mettre une sorte de service dans le milieu. MySQL est habituellement utilisé avec PHP, donc le plus évident est d'écrire des scripts PHP pour gérer la base de données et exécuter ces scripts en utilisant le protocole HTTP. J' ai codé les données

dans le format JSON, afin de communiquer les données entre PHP et Android, en exploitant les options faciles à utiliser construit dans les fonctions JSON dans les deux langages.

En fait, si nous parlons de l'architecture 3-tiers de point de vue technologie, le client est la plateforme Android, le serveur web est le PHP et le serveur de bases de données est le MySQL.



**Figure 67: Architecture générale de l'application.**

En fait mon projet est divisé en deux parties à savoir l'écriture des scripts PHP, et le développement de l'application Android.

La partie serveur est composée de deux serveurs distants : le serveur web et le serveur de base données. Le serveur Web utilisé est le serveur Apache, il est le serveur le plus répandu sur internet pour déployer les scripts PHP. Le serveur de bases de données utilisé est le serveur MySQL.

## 1.2. Elaboration du diagramme de déploiement :

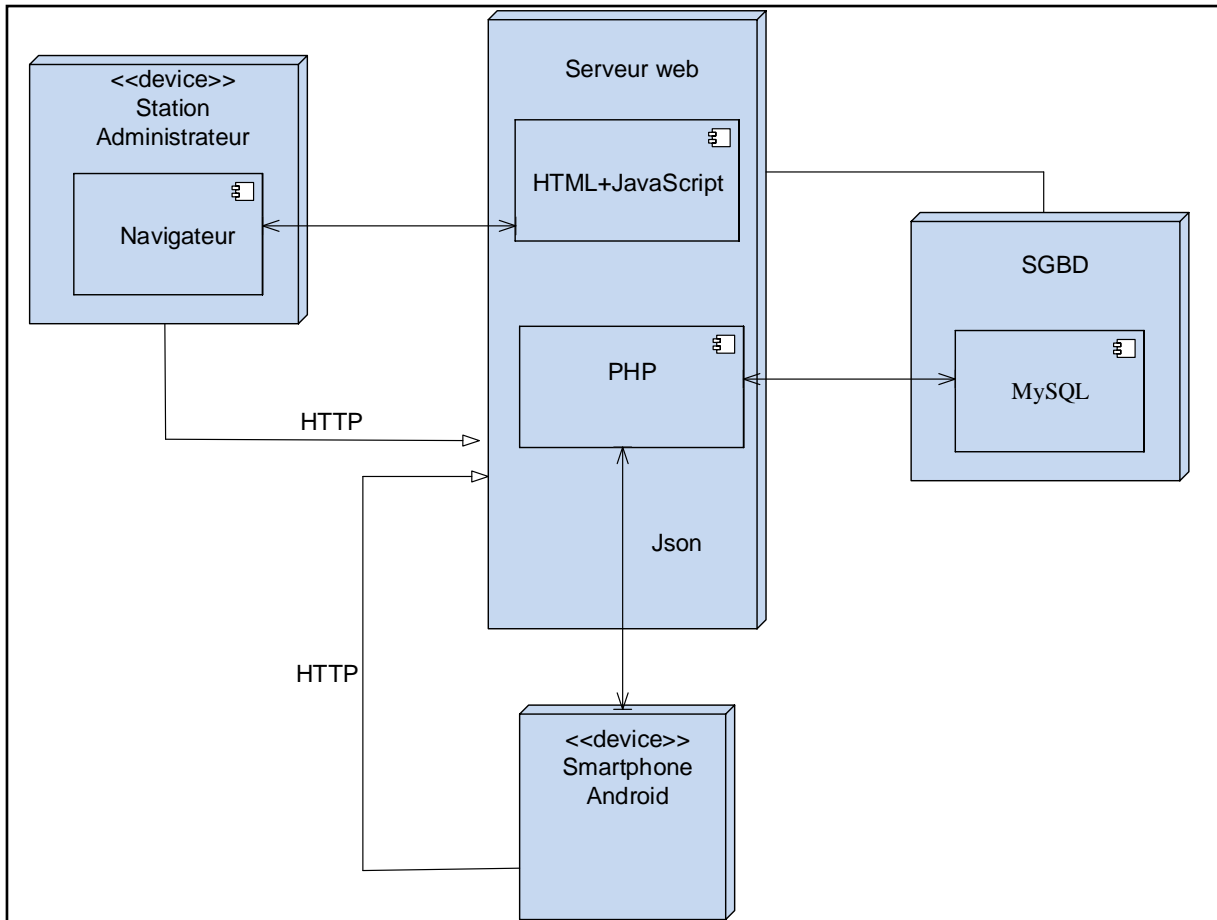


Figure 68:Diagramme de déploiement

## 1.3. Environnement matériel

J'ai utilisé comme environnement matériel un ordinateur, HP qui possèdent ces caractéristiques suivantes :

- **HP :**
  - ✓ Un processeur Intel® Core i7-4700 CPU @ 2.80GHz 2.40GHz
  - ✓ Une mémoire vive de 12 GO
  - ✓ Une carte graphique Nvidia GeForce GT 750M 2048 MO
  - ✓ Un disque dur de 1 Terra
  - ✓ System exploitation: Windows 10 (x64)
  - ✓ Une mémoire vive de 8,0 GO
  - ✓ Une carte graphique ATI Mobility Radeon HD 5000 Séries

## **1.4. Technologies utilisées**

### **1.4.1. Java :**

Android est un système d'exploitation conçu pour téléphone mobile développé par Google, qui a mis à disposition un kit de développement logiciel (SDK) basé sur le langage Java. Nous rappelons que Java, étant un langage de programmation orienté objet utilisable sur divers systèmes d'exploitation, est un langage assez robuste, portable et à hautes performances.

### **1.4.2. Android :**

Nous avons parlé de l'Android dans le premier chapitre. Je cite ici les principales qualités d'Android qui m'ont encouragées à choisir ce système :

- L'interaction avec les services Google (Maps, Navigation) est plus aisée et offre plus de fonctionnalités.
- L'interface utilisateur (bureaux virtuels, etc.) sont plus développés.
- Android est gratuit et open source.

Les différentes étapes de la préparation de l'environnement de développement sous Android sont illustrées dans l'annexe 2. Ainsi pour gérer la carte Google Maps, j'ai eu recours à des API Google (annexe 3).

### **1.4.3. PHP :**

PHP ou Hyper Text Preprocessor est un langage de script extrêmement puissant et destiné pour le développement des applications web. PHP est l'un des langages de programmation les plus populaire. Le point fort de ce langage c'est qu'il est portable et simple à utiliser.

### **1.4.4. HTML5 :**

HTML5 (HyperText Mark up Langage 5) est la nouvelle révision majeure d'HTML (format de données conçu pour représenter les pages web). Cette version est devenue stable depuis 2012. Le langage comprend une couche application avec de nombreuses API.

## 1.5. Environnements logiciels

### 1.5.1. PhpStorm :

**PhpStorm** est un éditeur pour [PHP](#), [HTML](#) et [JavaScript](#), édité par [JetBrains](#).

Il permet d'éditer du code PHP 5.3, 5.4, 5.5, 5.6 et 7.0.

Il possède :

- Une coloration syntaxique ;
- Affichage des erreurs à la volée ;
- Auto-complétion intelligente du code ;
- Réusinage du code.

Il intègre :

- L'envoi des fichiers via FTP ;
- Un gestionnaire de version.

Il permet aussi de visualiser l'architecture de bases de données de différentes sources (MySQL, SQLite, ...).

### 1.5.2. Wamp server

Wamp server est une plateforme de développement des applications web dynamiques. Ce logiciel est très intéressant puisqu'il englobe tous les outils nécessaires pour le fonctionnement d'une application web notamment un serveur de base de données MySQL, un serveur web apache et une interface de gestion des bases de données facile à utiliser PHP My admin.

### 1.5.3. SGBD MySQL

MySQL est le serveur de base de données le plus utilisé dans le monde. Son architecture logicielle le rend extrêmement rapide et facile à personnaliser.

Les principaux avantages sont :

- Solution très courante en hébergement public.
- Open Source
- Facilité de déploiement et de prise en main.
- Plusieurs moteurs de stockage adaptés aux différentes problématiques

## **2. Protocole et formats de données**

### **2.1. Protocole de communication**

Dans mon projet, j'ai utilisé le protocole HTTP, afin de communiquer les données entre l'application Android et le serveur web. En effet, Le HTTP est un protocole qui définit la communication entre un serveur et un client. En général, j'ai utilisé la méthode Post pour envoyer des données au programme situé à une URL spécifiée.

### **2.2. Format de données communiquées**

JSON (JavaScript Object Notation) est un format léger d'échange de données. Il peut être aisément analysé et généré par des machines.

Lorsque l'application Android s'exécute, elle se connectera au script PHP. Le script PHP va récupérer les données depuis la base de données MySQL. Ensuite les données seront encodées au format JSON et envoyées au système Android. Ensuite, l'application Android va obtenir ces données codées. Elle les analysera et les affichera sur l'appareil Android.

## **Conclusion**

Tout au long de ce chapitre j'ai essayé de présenter les différents travaux qui se déroulent à la fin du cycle de développement Scrum. J'ai présenté le diagramme de déploiement de mon application et j'ai préparé la documentation nécessaire pour les futurs développeurs.

---

## Conclusion et perspectives

---

La géolocalisation de KB\_DRIVER présente de nos jours un atout considérable pour les gens ne voulant plus gaspiller du temps en attendant la disponibilité du transport public, surtout aux moments d'encombrement et notamment dans une ville peuplée et pauvre en organisation de transport. En fait, il m'a été confié de concevoir et réaliser une application de géolocalisation de Taxi destinée aux périphériques mobiles.

J'ai, à cet effet, essayé d'adopter les meilleures solutions techniques et méthodes de développement. Dans une première étape, j'ai commencé par introduire le cadre de l'élaboration du projet ainsi qu'une étude théorique sur les notions de base et les technologies employées, suivie d'une étude de l'existant.

J'ai préparé par la suite mon planning de travail en respectant les priorités de mes besoins suite à une discussion entre l'équipe du développement et le directeur du produit. Tout au long de notre cycle de développement j'ai couplé la méthodologie Scrum par une autre méthodologie agile ; l'extrême programming afin de profiter des bonnes pratiques d'ingénierie logicielle proposées par cette dernière.

La richesse de ce sujet c'était une occasion de profiter tant dans l'acquisition des connaissances que dans l'initiation au travail de groupe. Cependant ce projet lié au stage, était aussi une



opportunité de découvrir le monde professionnel, ses réalités, du point de vue adaptation et difficultés.

Comme d'autres applications, mon application peut être aisément améliorée. En effet, grâce à son aspect ouvert, Android offre l'opportunité de créer des logiciels mobiles innovants et révolutionnaires en encourageant les développeurs à avancer dans leur imagination et à mobiliser toutes leurs compétences pour le meilleur de cette plateforme.

## *Bibliographie*

- ✓ <http://stackoverflow.com/>
- ✓ <https://github.com/>
- ✓ <http://www.androidhive.info/>
- ✓ <http://www.tutos-android.com/>
- ✓ <https://www.cognalys.com/#>
- ✓ <https://developers.facebook.com/>
- ✓ <https://developers.google.com/>
- ✓ <https://console.developers.google.com/>
- ✓ <http://jsonformatter.curiousconcept.com/>
- ✓ <http://json.parser.online.fr/>
- ✓ <https://play.google.com/store>
- ✓ <http://www.pluralsight.com/>
- ✓ <http://excerpts.numilog.com/books/9782100540181.pdf>
- ✓ [http://www.numilog.com/extrait/extrait.asp?id\\_livre=91910](http://www.numilog.com/extrait/extrait.asp?id_livre=91910)

# *Annexes*

## **Annexe 1. Cycle de vie d'une application Android**

L'activité Android surveille et réagit à ces événements par l'instanciation des méthodes qui remplacent les méthodes de la classe d'activité pour chaque événement :

- **onCreate** : C'est la méthode appelée lorsque l'activité est d'abord créée.
- **onStart** : Cette méthode est appelée juste avant que l'activité devienne visible à l'écran. Une fois onStart est terminé, l'activité peut devenir l'activité du premier plan, contrôle le transfert en méthode onResume. Si l'activité ne peut pas devenir l'activité du premier plan pour une raison quelconque, elle sera transférée en méthode onStop.
- **onResume** : Appelée juste après onStart si l'activité est l'activité du premier plan sur l'écran. A ce stade, l'activité est en cours d'exécution et d'interaction avec l'utilisateur. Ministère de l'Enseignement Supérieur et de la Recherche scientifique et de technologies de l'information et de la communication onResume est également appelé, si l'activité cède l'avant-plan à une autre activité et que l'activité existe encore.
- **onPause** : Appelée lorsqu'Android est sur le point de reprendre une activité différente. A ce stade, l'activité n'aura plus accès à l'écran.
- **onStop** : Appelée lorsque l'activité n'est plus visible, soit parce qu'une autre activité a pris le premier plan, soit parce que l'activité est détruite.
- **onDestroy** : C'est la dernière chance pour l'activité à faire tout traitement avant qu'elle ne soit détruite. Mais comme mentionné précédemment, la méthode peut être appelée dans le cas où le système décide d'allouer les ressources système à une autre activité.

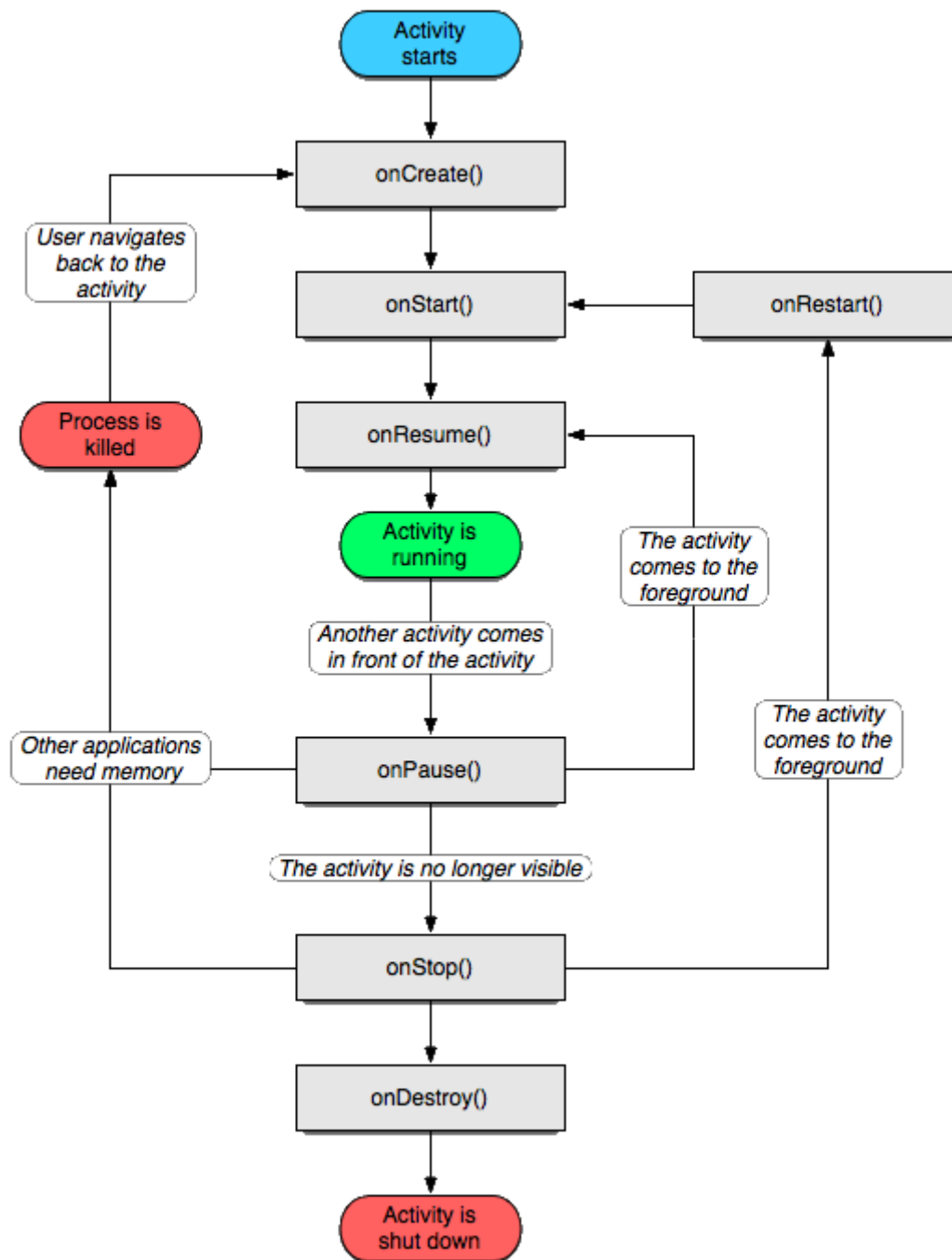


Figure : Cycle de vie d'une application Android

## Annexe 2. Prise en main de l'environnement Android

La première étape de mon travail avec l'environnement Android a été d'appréhender le SDK

- SDK Android

L'outil le plus important est le SDK Android. Facile à installer, il permet de télécharger tous les outils indispensables au développement d'applications. Un petit logiciel permet d'abord de télécharger les différentes versions du SDK (une version du SDK par version d'Android : 1.4, 1.5, 1.6, 2.0 ,4.0 etc.). Il permet également de télécharger les différentes versions des Google APIs (APIs pour intégrer des fonctionnalités liées aux services Google).

- Emulateur

Le SDK propose un émulateur Android. Il permet de lancer sur la machine du développeur un terminal virtuel représentant à l'écran un téléphone Android. C'est bien évidemment un outil indispensable pour le développement mobile. A chaque version d'Android est associée une version de l'émulateur, permettant au développeur de voir exactement à quoi ressemblera son application sur un matériel réel. Rappelons cependant que l'émulateur ne propose pas toutes les fonctionnalités d'un vrai téléphone. Il ne permet par exemple pas d'émuler la gestion du Bluetooth.

- Création d'interfaces utilisateur sous Android

Sous Android, nous pouvons décrire nos interfaces utilisateur de deux façons différentes :

- Avec une description déclarative XML ou directement dans le code d'une activité en utilisant les classes adéquates. La façon la plus simple de réaliser une interface est d'utiliser la méthode déclarative XML via la création d'un fichier XML que nous placerons dans le dossier /res/layout de notre projet.

En adoptant cette dernière solution, les interfaces de mon application sont décrites dans des fichiers XML, afin de faciliter leur modification en cas de besoin.

- Procédure d'accès à la Carte Google Maps

Avant de pouvoir insérer une carte Google Maps dans l'interface graphique, il faut que nous obtenions une clé API. Une fois la clé API obtenue, il faut l'ajouter dans le fichier String.xml. J'ai suivi les différentes étapes pour la génération de clé API afin de l'obtenir :

« 0iqj7lgFnMqvC1ek49u5PQdIdFN5rMAy-z\_dMjg ».

### Annexe 3. Quelques APIs utilisé

**LocationListener** : Cette classe est utilisée quand la localisation du téléphone est mise à jour. Ainsi la première étape est d'appeler la méthode `requestLocationUpdates(String, long, float, LocationListener)` qui permet de s'abonner aux changements de localisation. Implémenter l'interface `LocationListener` signifie qu'on surcharge les 4 méthodes suivantes :

- `onLocationChanged(Location location)` : Cette méthode est appelée quand la géolocalisation de l'utilisateur change. L'argument représente la nouvelle localisation du téléphone.
- `onProviderDisabled(String provider)` : Cette méthode est appelée quand l'utilisateur désactive la source de localisation (GPS, 3G..etc).
- `onProviderEnabled(String provider)` : Cette méthode est appelée quand l'utilisateur active une source qu'on peut utiliser pour géo localiser (GPS, 3G..Etc).
- `onStatusChanged(String provider, int status, Bundle extras)` : Appeler quand le statut d'une source change.

**Location Manager** : cette classe fournit un accès au service de localisation du système. Elle permet de choisir le meilleur fournisseur de localisation pour enregistrer des écouteurs de mise à jour de localisation et d'alertes de proximité et plus encore.

**Location** : cette classe représente une localisation géographique à un instant donnée. Elle est composée d'une latitude, d'une longitude...