



N° Réf :.....

Centre Universitaire
Abd Elhafid Boussouf Mila

Institut des Sciences et Technologie

Département de Mathématiques et Informatique

Mémoire préparé en vue de l'obtention du diplôme de Master

En : Informatique

**Spécialité : Sciences et Technologies de l'Information et de la
Communication (STIC)**

Conception et Implémentation d'une Plateforme de Gestion de livraison par la foule (Crowd local delivery)

**Préparé par : Aziza khadidja
Tebboub ikram**

Soutenue devant le jury

Encadré par bessouf hakim

Président Bouchemal Nardjes

Examineur Selmen Samir

Année Universitaire : 2020/2021

Table des matières

Table des matières

Liste des tableaux

Table des figures

Remerciement

Dédicace

Résumé

Introduction générale

- CHAPITRE 01: Généralités sur les applications mobiles

■ Introduction :	2
I. Les applications mobiles :	3
1. Définition :	3
2. Les types d'application :	4
3. Caractéristiques des applications mobiles :	4
II. Les différentes plateformes mobiles :	5
1. Android :	6
2. IOS :	7
3. BlackBerry OS :	7
4. Windows Phone 7 :	8
5. Comparaison entre les systèmes d'exploitation :	8
III. Le système d'exploitation Android :	9
1. La naissance d'Android :	10
2. l'évolution des versions d'Android:	10
3. Architecture logicielle :	13
4. Kit de développement :	15
IV. Les bases de données :	15
1. Définition :	15
2. les différents types de bases de données :	15
V. la géolocalisation :	15
1. Gps :	15

2.google maps :	15
-----------------------	----

CHAPITRE 02: Étude préliminaire

I. Etude préliminaire.....	16
Introduction.....	16
2.Élaboration du cahier des charges	16
2.1 La présentation du projet	17
2.2 Les grands choix techniques	18
2.3. Identifications des acteurs	18
2.4. les besoins fonctionnels	19
2.5.les besoins opérationnels	20
3. Description de contexte de système à réaliser	20
3.1. Identifications des messages	21
3.2. Le diagramme de contexte de système	21
Conclusion	22

Chapitre 03 : Capture des besoins

Introduction.....	23
I. Situation de l'étape dans le processus 2TUP	23
1.Capture des besoins fonctionnels	23
1.1 L'analyse	24
1.1.1. Identification des cas d'utilisation	24
1.1.2. Diagramme de cas d'utilisation	24
1.1.3. Description des cas d'utilisation	25
1.1.4. Identification des classes candidates.....	40
1.1.5. Diagramme de classe participant.....	42
2.Capture des besoins techniques.....	47
2.1. La spécification technique du point de vue matériel	47
2.2. Capture des spécifications logicielles.....	48
2.2.1. Exploitant.....	48

2.2.2. Cas d'utilisation technique.....	48
Conclusion.....	52

Chapitre 4 : Analyse et Conception

1.dictionnaire des données	53
2.le diagramme de classe générale.....	54
3.Le model relationnel	54
4.Diagramme de séquence	55
4.1. Les diagrammes de séquence	55
Conclusion.....	63

Chapitre 5: implémentation

Introduction.....	64
I. Présentation des outils de développement.....	64
1. Le Langage de programmation java.....	64
2. Le langage xml.....	64
3. Le SDK (Software Development Kit).....	64
4. Android studio	65
5. WampServer.....	65
6. MySQL.....	66
7. Php.....	66
II.Présentation de quelques interfaces de l'application.....	67
Conclusion.....	69

Conclusion générale

Tableau 1	Une comparaison entre les systèmes d'exploitation mobile
Tableau 2	Les différentes versions Android
Tableau 3	Légende des messages de diagrammes de contexte
Tableau 4	Liste des cas d'utilisations fonctionnels
Tableau 5	Description textuelle « Passer demande livraison
Tableau 6	Description textuelle « consulter les offres de livraison
Tableau 7	Description textuelle « choisir une offre »
Tableau 8	Description textuelle « consulter l'état de livraison d'un colis »
Tableau 9	Description textuelle « Valider la livraison du colis »
Tableau 10	Description textuelle « passer un Feedback»
Tableau 11	Description textuelle « Consulter les demandes »
Tableau 12	Description textuelle « valider réponse client »
Tableau 13	Description textuelle « proposer une offre »
Tableau 14	Description textuelle « Mettre à jour l'état de livraison »
Tableau 15	Description textuelle « «consulter Feedback »
Tableau 16	Description textuelle « Avertir compte »
Tableau 17	Description textuelle « bloquer compte »
Tableau 18	dictionnaire des données

Figure01	des applications mobiles
Figure02	les systèmes d'exploitation
Figure03	Logo du système Android
Figure04	Logo du système iOS
Figure05	Logo du système BlackBerry OS
Figure06	Logo du système Windows phone 7
Figure07	Architecture logicielle de l'Android
Figure08	la méthode 2TUP
Figure09	exemple de Cahier de charge
Figure10	Diagramme de contexte
Figure11	les étapes de processus 2TUP
Figure12	Diagramme de cas d'utilisation générale
Figure13	Description graphique « Passer demande livraison »
Figure14	Description graphique « consulter les offres de livraison »
Figure15	Description graphique « choisir une offre »
Figure16	Description graphique « consulter l'état de livraison d'un colis »
Figure17	Description graphique « Valider la livraison du colis »
Figure18	graphique « passer un Feedback »
Figure19	Description graphique « Consulter les demandes »
Figure 20	Description graphique « valider réponse client »
Figure 21	Description graphique « proposer une offre de livraison »
Figure 22	Description graphique « Mettre à jour l'état de livraison »
Figure 23	Description graphique « Consulter Feedback»
Figure24	Description graphique « Avertir compte »
Figure 25	Description graphique « Bloquer compte »
Figure26	Description graphique « Consulter activité livreur »
Figure 27	Les responsabilités de la classe candidate<<utilisateur >>
Figure 28	Les responsabilités de la classe candidate<<client >>
Figure29	Les responsabilités de la classe candidate<< livreur >>
Figure30	Les responsabilités de la classe candidate<< offre>>
Figure31	Les responsabilités de la classe candidate<< demande >>
Figure 32	Les diagrammes de classes participantes dans la démarche
Figure33	diagramme de classe participantes «Modifier profile »
Figure34	diagramme de classe participantes « S'authentifier »
Figure35	diagramme de classe participantes « Passer demande livraison »
Figure36	diagramme de classe participantes « Consulter les offres de livraison »
Figure37	diagramme de classe participantes« Consulter l'état de livraison d'un colis »
Figure38	diagramme de classe participantes « Valider la livraison du colis »
Figure39	diagramme de classe participantes « Proposer une offre de livraison »
Figure40	diagramme de classe participantes « Valider réponse

	client »
Figure41	diagramme de classe participante « Mettre à jour l'état de livraison »
Figure42	diagramme de classe participantes « Avertir compte »
Figure43	Architecture 2 niveaux
Figure44	Modèle de spécification logicielle du système
Figure45	Description textuelle « modifier profile »
Figure46	Description textuelle « s'authentifier »
Figure47	Diagramme de séquence de cas d'utilisation « Gérer l'intégrité des données »
Figure 48	diagramme de classe générale
Figure 49	diagrammes de séquences << modifier profile>>
Figure50	diagrammes de séquences << S'authentifier>>
Figure51	diagrammes de séquences << passer demande de livraison>>
Figure 52	diagrammes de séquences << Consulter les offres de livraison>>
Figure53	diagrammes de séquences << Consulter l'état de livraison d'un colis>>
Figure54	diagrammes de séquences << valider la livraison du colis>>
Figure55	diagrammes de séquences <<valider réponse client>>
Figure56	diagrammes de séquences << proposer une offre de livraison>>
Figure57	diagrammes de séquences << Mettre à jour l'état de livraison>>
Figure 58	diagrammes de séquences << Avertir compte >>
Figure 59	logo java.
Figure 60	logo Android studio
Figure 61	logo WampServer
Figure 62	logo MySQL
Figure 63	interface pour crée un compte client <<crée compte livreur>>
Figure 64	interface pour crée un compte livreur
Figure 65	interface pour passer une demande de livraison
Figure 66	<<page accueil client>>

Dédicaces

A Ma mère, mon évidence, ma raison de vivre, à travers ce travail je tiens à te dire merci, merci pour l'amour, les principes et l'éducation que tu m'as donnée sache que c'est grâce à toi qu je suis devenue la Femme que je suis Aujourd'hui, ta confiance me donne confiance en moi pour réussir et réaliser mes rêves.

A Mon père, Merci pour les ailes que tu m'as données, pour m'avoir appris à me lever et à élargir mes horizons vers les cieux, d'avoir toujours cru en moi j'ai pris de ta force, Merci papa

A Ma petite sœur Amani, Merci pour le soutien moral, pour les fois où nous avons partagé nos rêves ensemble. Je te souhaite beaucoup de réussite.

A mes frères Mohamed et Raouf, vous êtes des frères unique, attentionnés et Gentils, que dieu vous garde pour moi

A Ma tante BENZEMMAMOUCHE Warda, sa petite famille mais surtout, Ma cousine, Mon amie, BOUHENNACHE Hind, tu m'as donné du courage merci d'avoir été présente pour moi durant la rédaction de ce travail.

A ma binôme, IKRAM Je souhaite personnellement te remercier. On a pris beaucoup de plaisir à travailler ensemble. Nous avons formé une belle équipe, je te remercie donc pour tout ce que tu m'as apporté au cours de ces années qu'on a partagées.

Enfin, à toutes les personnes chères à mon cœur qui m'ont soutenue et m'ont encouragé et à toutes les personnes qui ne sont pas là aujourd'hui pour partager mon soulagement et ma satisfaction.

Mlle AZIZA khadidja

Dédicaces

*A toi ma très chère maman **SALIMA**, ma confidentes et ma source de courage qui a souffert sans nous laisser souffrir, rien ne pourrait exprimer la profondeur des sentiments que j'éprouve pour toi sans toi je ne serais jamais arrivé là où je suis.*

*A toi mon papa **ABD EL MADJID** mon idole mon précieux offre de dieu qui doit ma vie, ma réussite et tout mon respect*

*A mes sœurs **ISRAA, AMANI** pour tout leur soutien morale et précieux conseils tout au long de mes études.*

*A tous les moments d'enfance passés avec vous mes frères **OUSSAMA, AMDJED**, en gage de ma profonde estime pour l'aide que tu m'as apporté..*

A tous mes tantes oncles que dieu leurs apporte longue et heureuse vie.

*A mes chères couines **ASSALA ,AMIRA ET MERIEM** en souvenir des moments heureux passés ensemble, avec mes vœux sincères de réussite, bonheur, santé et de prospérité.*

*Sans oublier mon binôme et ma chère amie **KHADIDJA** pour ton soutien morale, ta compréhension ton amitié que je n'échangerais pour rien au monde, ainsi qu'à toute la famille **AZIZA**.*

Mlle TEBBOUB ikram

REMERCIEMENTS

*Tout d'abord, On tient à remercier chaleureusement notre directeur de thèse monsieur **BESSOUF HAKIM** pour la confiance qu'il nous a accordé en acceptant de diriger ce travail. On tient à lui exprimer toute notre reconnaissance et notre gratitude pour son encadrement et ses précieux conseils, pour sa patience et sa compréhension et surtout pour son soutien dans toutes nos ambitions.*

Résumé :

La livraison à domicile devient de plus en plus populaire. Accélérée par la crise du Covid-19, ce modèle de vente devient un incontournable pour les entreprises, quel que soit leurs secteurs d'activités.

L'objectif de ce mémoire concerne la conception et la réalisation d'une application mobile pour la gestion de livraison par la foule. C'est une application qui permet de fournir un moyen de communication entre le client et le livreur pour partager des informations (information des demandes, informations des offres) et échanger des messages.

Pour atteindre ces objectifs, nous avons mené une étude conceptuelle en suivant le processus de développement logiciel 2TUP, utilisé UML comme langage de modélisation, le langage JAVA sous la plateforme ANDROID pour le développement et la programmation, et le SGBDR MySQL pour l'implémentation de la base de données.

Mot clé : 2TUP, UML, JAVA, ANDROID, MySQL, Application mobile.

Abstract :

Home delivery is becoming more and more popular. Accelerated by the Covid-19 crisis, this sales model is becoming a must for companies, whatever their sectors of activity.

The objective of this dissertation concerns the design and the realization of a mobile application for the management of delivery by the crowd. It is an application that provides a means of communication between the customer and the delivery person to share information (information on requests, information on offers) and exchange messages.

To achieve these objectives, we conducted a conceptual study following the 2TUP software development process, used UML as the modeling language, the JAVA language under the ANDROID platform for development and programming, and the MySQL RDBMS for the implementation of the database.

Keyword: 2TUP, UML, JAVA, ANDROID, MySQL, Mobile application.

تلخيص:

أصبح توصيل الطلبات للمنازل أكثر شيوعاً، حيث تسارعت أثناء أزمة فيروس كورونا Covid-19 ، وأصبح نموذج المبيعات هذا أمراً ضرورياً للشركات ، بغض النظر عن قطاعات نشاطها .

الهدف من هذه الرسالة هو تصميم وتنفيذ تطبيقات الهاتف المتحرك لإدارة التسليم من قبل الجمهور. هو تطبيق يوفر وسيلة اتصال بين العميل والشخص المسؤول عن التوصيل لمشاركة المعلومات (معلومات عن الطلبات ، معلومات عن العروض) وتبادل الرسائل.

لتحقيق هذه الأهداف ، أجرينا دراسة مفاهيمية بعد عملية تطوير برامج TUP2 ، واستخدمنا UML كلغة نمذجة ، ولغة JAVA ضمن منصة ANDROID للتطوير والبرمجة ، و MySQL RDBMS لتنفيذ قاعدة البيانات.

الكلمة الرئيسية: TUP 2 ، UML ، JAVA ، ANDROID ، MySQL ، تطبيق الهاتف المحمول.

Introduction Générale

Avec l'indispensabilité de la communication et de l'information dans notre vie quotidienne et grâce à la généralisation des téléphones portables tactiles à écrans large ainsi qu'au développement des logiciels et des réseaux, la recherche d'information est accessible du bout des doigts par l'intermédiaire de téléphone ou pads à tout moment et en tous lieux.

Il est important pour tout individu de se tenir au courant de ce qui se passe autour de lui et d'avoir accès aux informations dont il a besoin, quand il en a besoin plus facilement et plus rapidement.

Ce mémoire est destiné à toute personne curieuse, désireuse de connaître et comprendre le concept de la livraison par foule au plus communément appelée la crowd sourcing.

Il a expliqué que le crowdsourcing est le fait pour une entreprise de déléguer à la « foule » des activités qui étaient autre fois gérées par des ses agents désignés tels que les employés. En d'autres termes, le crowdsourcing peut être considéré comme l'externalisation par une entreprise de certaines activités à la foule. Notre manuscrit est organisé comme suit :

Le premier chapitre consacré à la présentation des applications mobiles, les system d'exploitation.

Dans le deuxième chapitre effectuer un premier repérage des besoins fonctionnels et opérationnels, en donnant une version textuelle préliminaire du cahier des charges. Après avoir identifié les acteurs qui interagiront avec le système, nous développerons un premier modèle UML de niveau contexte, pour pouvoir établir précisément les frontières fonctionnelles du système.

Dans le troisième chapitre, nous allons décrire les besoins et les acteurs qui vont interagir avec notre système et présenter un recueil des besoins fonctionnels et techniques du système.

Dans le quatrième chapitre, nous allons présenter les phases d'analyse et de conception. Partie analyse qui permet d'étudier précisément les spécifications fonctionnelles de manière à obtenir une idée de ce que va réaliser le système en termes de métier. la phase de conception qui comprend la conception préliminaire et détaillée.

Dans le cinquième chapitre, nous allons présenter les environnements et les outils de développement de l'application mobile et de l'application web, ainsi que quelques interfaces de l'application.

CHAPITRE 1 :

Généralités sur les applications mobiles.

Introduction :

Dans le monde actuel, les smart phones, tablettes et télévision aussi font partis de notre vie quotidienne, ils sont équipés d'applications qui sont conçues pour des plateformes mobiles et utilisés pour des services d'information, de médias sociaux, Activités commerciales etc....

Un smartphone a un système d'exploitation qui varie d'une marque à l'autre.

Dans ce chapitre nous présentons les différentes notions de l'environnement Android qui est le plus répandu sur le marché et ses fonctionnalités ainsi que les bases de données.

I. Les applications mobiles :

1. Définition :

Une application mobile, ou « appli », est un logiciel développé spécialement pour être utilisé sur de petits appareils informatiques sans fil, tels que smartphones et tablettes, plutôt que sur des ordinateurs de bureau ou portables.

Ainsi, les applications mobiles sont conçues pour tenir compte des exigences et contraintes des différents appareils tout en exploitant leurs fonctionnalités propres. Une application de jeu, par exemple, pourra tirer parti de l'accéléromètre de l'iPhone.

On classe parfois les applications mobiles selon qu'elles sont de type Web ou natives, c'est-à-dire créées spécialement pour une plateforme donnée. Une troisième catégorie, les applications hybrides, combine des éléments des deux premières. Avec l'évolution des technologies, les efforts de développement d'applications mobiles devraient porter sur la création d'applications Web basées sur le navigateur et donc indépendantes du matériel. (1)



Figure 01 : des applications mobiles

2.les types d'application :

Il existe trois familles d'applications mobiles : les **applications natives** ou « Natives Apps », les **applications Web** ou « Web Apps » et les **applications hybrides** ou « Hybrid Apps ».

a) Natives Apps ou application natives :

Dans un premier temps, nous retrouvons donc les applications natives, Il s'agit de logiciels qui sont exclusivement créés pour une plateforme mobile spécifique. Leur développement se fait via le software développent kit, ou SDK, de la plateforme concernée. Le nom **Natives Apps** vient donc du fait que ces applications sont développées grâce à l'unique utilisation du langage « natif », soit JAVA pour Android ou Objective C pour Apple.

À noter : ce type d'applications se télécharge directement depuis la plateforme de téléchargement associée au système d'exploitation. On parle alors d'un store applicatif : Google Play, pour les dispositifs Android ou encore Apple Store pour les dispositifs Apple.

b) Web Apps ou applications web :

Nous retrouvons ensuite les applications web. Ce type d'application correspondent à des sites web. Ces derniers sont mis en place pour proposer un affichage entièrement optimisé pour les téléphones mobiles. Contrairement aux applications natives, qui nécessitent un téléchargement préalable, les Web Apps sont accessibles depuis un navigateur pour mobile.

De manière générale, ces dernières sont développées grâce à des technologies web comme le HTML5 ou le CSS3.

C) Hybride Apps ou applications hybrides :

Enfin, nous retrouvons les applications de type hybride. Comme évoqué précédemment, ces applications sont accessibles depuis toutes les plateformes mobiles. Par ailleurs, elles se présente comme un mix entre les Natives Apps et les Web Apps.

Principalement développées à partir du HTML5, elles utilisent aussi des langages Web tels que le CSS ou encore le JavaScript.

De cette manière, contrairement aux Natives Apps, elles ne sont pas dépendantes d'une plateforme particulière. D'autre part, contrairement aux Web Apps, elles ont la possibilité d'accéder à toutes les fonctions qui sont présentes sur un téléphone mobile. (2)

3.les caractéristique des applications mobiles :

a) Une application mobile est plus rapide :

En effet, sauf si elle est très mal programmée (ce qui peut arriver), une application native va exploiter au mieux les capacités du téléphone. En plus, en utilisant les composants graphiques du téléphone (boutons, barre de titre, onglets...), l'interface est plus facile à appréhender pour l'utilisateur qui retrouve rapidement ses marques.

b) Une application offre plus de possibilités :

Accéder aux photos, à l'agenda, aux contacts, proposer de la réalité augmentée, réagir aux mouvements du téléphone, lire un code barre : autant de fonctionnalités qui ne sont possibles qu'avec une application mobile.

c) Il faut développer pour chaque plateforme :

Là encore, c'est tout à fait vrai. Pour s'adapter à l'ergonomie de chaque plateforme, exploiter au mieux les fonctionnalités, il faut développer pour chaque type de téléphone (iPhone, Android...). Cependant, décliner une application vers une nouvelle plateforme coûte moins cher, car une bonne partie de l'application peut être réutilisée.

d) On gagne plus avec une application mobile :

Si vous avez comme projet de vendre votre application, c'est effectivement plus aisé qu'avec une Web App, vente de l'application sur l'AppStore ou l'Android Market, vente d'articles ou de fonctionnalités à l'intérieur de l'application, tout ceci est possible très simplement.

e) Le déploiement est plus compliqué, plus long :

Apple impose une validation des applications avant publication. C'est vrai. Il n'y a cependant rien de tel sur AndroidMarket⁶, et les mises à jour sont dans tous les cas signalés aux utilisateurs, ce qui remet un coup de projecteur dessus. Enfin, pour les entreprises, il est possible de déployer ses applications en interne, même sur iPhone/iPad, sans passer par l'AppStore. (11)

II. Les différentes plateformes mobiles :

Les systèmes d'exploitation mobiles (OS) peuvent être définis comme les logiciels permettant à un smartphone ou un téléphone mobile basique de fonctionner.

Ils permettent de ce fait aux utilisateurs de pouvoir passer un appel téléphonique, naviguer sur leurs téléphones parmi toutes les rubriques, télécharger des applications ou encore paramétrer et personnaliser leurs smartphones.

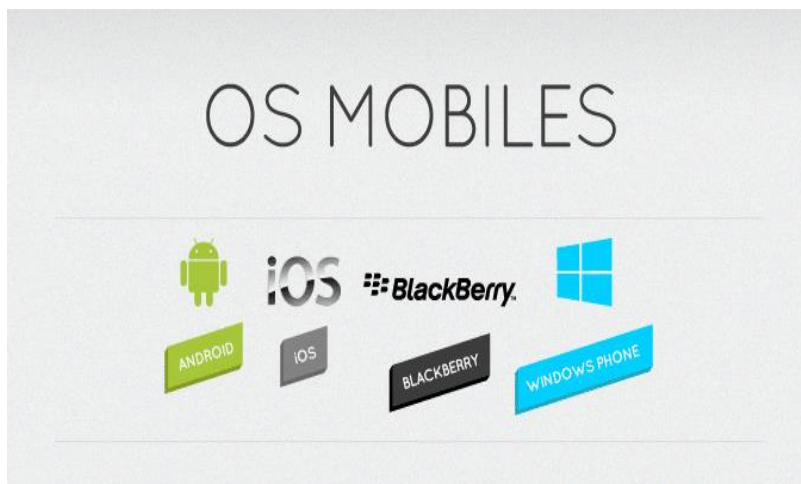


Figure 02 : les systèmes d'exploitation

1. Android :

Android est le système d'exploitation mobile créé par Google. Il équipe la majorité des téléphones portables du moment (smartphones). Son principal concurrent est Apple avec l'iPhone. Android est un système vous permettant de personnaliser votre téléphone, télécharger des applications (navigateur Internet, GPS, Facebook...). Android équipe également les tablettes tactiles.

Android est réputé pour être plus libre et ouvert que iOS, le système qui équipe l'iPad d'Apple.

Google a souhaité, en développant Android, créer un système facile à utiliser, qui équipera la majorité des **smartphones** et tablettes dans le monde. C'est un pari réussi. Le système vous permet de télécharger de nouvelles **applications** répondant à vos besoins. Votre téléphone, tablette seront donc unique et vous ressemble. Des milliers de développeurs **programment** chaque jour des nouvelles applications pour votre téléphone.

De nouvelles versions paraissent chaque année, et vous pourrez mettre à jour facilement votre téléphone ou tablette pour tirer le meilleur parti d'Android. (3)



Figure 03: Logo du système Android.

2. iOS :

iOS est le système d'exploitation d'Apple qui équipe actuellement les smartphones iPhone et les tablettes iPad. C'est le système qui apparaît à l'écran et qui permet d'utiliser votre appareil : téléphoner, écrire un message, lancer une app...

Le système a été conçu dès l'origine pour fonctionner entièrement sur un écran tactile.

Le système d'exploitation iOS existe depuis la création du tout premier iPhone en 2007 et une évolution majeure du système arrive chaque année. La mise à jour est gratuite.

Les appareils trop vieux ne pourront pas forcément profiter de la mise à jour.

Sur la toute première version d'iOS il n'était pas possible d'installer de nouvelles **apps** ! Mais cela a très vite évolué et aujourd'hui des milliers de développeurs proposent leurs apps. Cela a créé de nombreux nouveaux emplois. (4)



Figure 04 : Logo du système iOS

3.BlackBerry OS :

La plate-forme de communication sans fil BlackBerry est développée par Research In Motion (RIM). Les solutions proposées par RIM intègrent à la fois les matériels, les logiciels et les services qui permettent un accès simple et transparent aux informations importantes, telles que messages électronique, téléphoniques, SMS ainsi qu'aux applications Internet et Intranet.

Les produits, services et technologies de RIM ont reçu de nombreuses récompenses et sont utilisées par des milliers d'entreprises dans le monde. Longtemps réservé aux professionnels, BlackBerry a dû s'adapter à sa nouvelle clientèle, plus jeune et tombée sur le charme des claviers AZERTY, très pratiques pour l'envoi de messages écrits. D'une interface austère et dédiée à l'email (que l'on peut envoyer et recevoir en instantané), BlackBerry OS est devenu au fil du temps agréable à utiliser et de plus en plus complet.

Le système regorge de bonnes idées à commencer par la possibilité de séparer son espace de travail de sa vie privée sur un seul et même téléphone. Le magasin d'application maison, l'App World, continue à se remplir, mais la force de l'OS est sa compatibilité avec les programmes Android, qui ne nécessitent Qu'une petite optimisation de la part des développeurs. (5)



Figure 05: Logo du système BlackBerry OS

4. Windows phone 7 :

Windows Phone est le nom du système d'exploitation pour smartphones que Microsoft a publié en octobre 2010. Il est venu remplacer Windows Mobile en introduisant une interface utilisateur totalement redessinée et pensée pour les terminaux à écran tactile. Celle-ci se compose de tuiles dynamiques dont l'affichage évolue en temps réel selon l'activité de l'application concernée. (6)



Figure 06 : Logo du système Windows phone 7

5. Comparaison entre les systèmes d'exploitation :

	IOS	BlackBerry	Windows phone	Android
Langage de programmation	Objective-C	Java	C, C++	Java
	Intégré à Xcode	Gratuit	Gratuit	Gratuit
Disponibilité de l'environnement de développement	Xcode	JDE	Visual Studio, eMbeddev VC++	Eclipse, NetBeans, Android studio
Multiplateforme De déploiement	iPhone, iPod Touch, iPad	BlackBerry seulement	Windows Mobile, Windows CE	Android seulement
Cout d'outils de développement	Gratuit	Gratuit	Gratuit	Gratuit
Magasin en ligne	App Store	App World	Windows Market Place	AndroidMarket
Open source	Non	Oui	Non	Oui
Constructeur	Apple	RIM	Microsoft	Google

Tableau 01: Une comparaison entre les systèmes d'exploitation mobile.

III. Le système d'exploitation Android :

Nous avons décrit en grosso modo les différents systèmes d'exploitation utilisés par les applications mobiles. Dans cette section, nous allons mettre le point sur la plateforme qu'on a choisi pour développer notre application. Nous avons choisi le système d'exploitation Android pour les raisons suivantes :

- Une application est développée à l'aide du SDK android et de l'IDE eclipse avec le plugin crée par google. À ce niveau-là, le développeur n'a pas besoin de déboursier d'argent.
- Android est basé sur le noyau Linux. Alors, y a plusieurs avantages comme une grande mémoire, la gestion de processus, le modèle de sécurité, le soutien de bibliothèque partage, etc.
- Une appli android est codé à l'aide de deux langages, l'interface se développe en XML et le reste de l'appli en java. Deux langages déjà très populaires et dont les notions sont utiles pas seulement pour android.
- La possibilité de tester l'appli que l'on développe gratuitement, Alors que sous IOS, la validation de l'appli pour être utilisé sur son propre iphone est obligatoire (et payante) sous android on peut tester son appli sans aucune validation.
- Les services Google comme Maps, Navigation, etc. peuvent très facilement être utilisé dans chaque appli. (7)

1. La naissance d'Android :

La création d'Android remonte en 2003 avec une société américaine dénommée ANDROID. Deux ans plus tard, donc en 2005, Google rachète Android tout en gardant sa renommée. A cette époque son objectif était de développer un système d'exploitation facile et compréhensif pour tous. Avec BugDroid, sa mascotte, Android a conquis presque le monde entier.

Android est le système d'exploitation mobile le plus utilisé au monde et c'est parce qu'elle a été rachetée par Google. (8)

2. L'évolution des versions d'Android:

Version	Caractéristique	
Android 1.0 G1 (2008)	Lors de son lancement, Android 1.0 proposait 35 applications via Android Market. Son Google Maps utilisait le GPS et le Wi-Fi du téléphone et un navigateur Android y était intégré	
Android 1.5 Cupcake (2009)	Avec Android Eclair, les utilisateurs Microsoft ont pu bénéficier de la prise en charge d'Exchange, de plusieurs comptes Google, d'un moteur de recherche dans les SMS, des gestes multitouch et d'améliorations au niveau de l'appareil photo avec un flash et un zoom numérique.	
Android 2.0 Eclair (2009)	Avec Android Eclair, les utilisateurs Microsoft ont pu bénéficier de la prise en charge d'Exchange, de plusieurs comptes Google, d'un moteur de recherche dans les SMS, des gestes multitouch et d'améliorations au niveau de l'appareil photo avec un flash et un zoom numérique.	
Android 2.2 Froyo (2010)	Froyo a apporté Flash Player 10.1 qui a permis aux smartphones de pouvoir lire des contenus vidéos et audio en streaming. Le flash de l'appareil photo fonctionne avec la vidéo, la compatibilité Bluetooth est élargie et l'on peut transformer son mobile en point d'accès Wi-Fi.	

Android 2.3 Gingerbread (2011)	Arrivée de la technologie Near Field Communication (NFC) qui permet aux terminaux de se connecter avec d'autres appareils compatibles à proximité immédiate. Les appels vidéo sont désormais disponibles via la caméra frontale et l'OS hérite d'un gestionnaire de téléchargement.	
Android 3.0 Honeycomb (2011)	Cette mise à jour fut la première dédiée uniquement aux tablettes Android. Elle introduit la prise en charge des graphismes en 3D, le chat vidéo avec Google Talk, le partage de connexion via Bluetooth et un mode plein écran dans la galerie photo.	
Android 4.0 Ice Cream Sandwich (2011)	Ice Cream Sandwich a réconcilié les versions smartphones et tablette de l'OS en ajoutant au passage, la reconnaissance faciale pour déverrouiller le mobile, des SMS préprogrammés pour décliner les appels téléphoniques et des effets en live pour l'enregistrement vidéo.	
Android 4.1 Jelly Bean (2012)	Grâce au « Projet Butter », Jelly Bean gagne en performance et en fluidité. Les notifications s'enrichissent, le navigateur Chrome est adopté par défaut, les widgets sont redimensionnables et Google Now est préinstallé.	
Android 4.4 KitKat (2013)	KitKat apporte des émoticônes au clavier Google, l'impression à distance avec Google Cloud Print et une utilisation plus parcimonieuse de la mémoire vive pour pouvoir tourner sur des smartphones d'entrée de gamme.	
Android 5.0 Lollipop (2014)	Grosse évolution esthétique avec l'introduction du « Material Design » et son interface aplanie. Les notifications s'affichent désormais sous forme de bannières sur l'écran de verrouillage et d'alertes pop-up.	

Android 6.0 Marshmallow (2015)	Arrivée du mode Doze qui préserve l'autonomie de la batterie, prise en charge native des lecteurs d'empreintes digitales, de l'USB-C et de l'Ultra HD pour les applications.	
Android 7.0 Nougat (2016)	Avec Nougat, il devient enfin possible de fermer toutes les applications depuis l'aperçu d'un seul geste. La mise à jour permet de modifier le teint des émoticônes, le menu des réglages rapides d'enrichit et l'OS peut fonctionner avec la plateforme de réalité virtuelle Google Daydream.	
Android 8.0 Oreo (2017)	Android Oreo renforce le multitâche, améliore le copier-coller, la sécurité et la gestion de la batterie	
Android 9.0 Pie (2018)	La mise à jour d'Android 9.0 a rendu les smartphones Android plus rapides et économes en énergie. Pour y arriver, Android a eu recours à l'intelligence artificielle pour suggérer des applications et des raccourcis qui anticipent les besoins de l'utilisateur, charger des aperçus ciblés sur l'information recherchée sans avoir à ouvrir une application ou une page web.	
Android 10 Q (2019)	Avec Android 10, Google abandonne les dénominations de desserts pour ses mises à jour système. Il faudra se contenter dorénavant d'un "simple" chiffre à partir de cette dixième version. Cette mise à jour, sortie en septembre 2019, a notamment mis l'accent sur les fonctionnalités de confidentialité et de sécurité.	
Android 11 (2020)	Cette version 2020 se concentre toujours autant sur les fonctionnalités de confidentialité et de sécurité. Le contrôle des autorisations est plus poussé, un nouveau système de capture d'écran a fait son apparition tout comme le système de bulles de notification. Nous avons compilé toutes les choses importantes à savoir sur Android 11 dans cet article.	
Android 12 (2021)	Android 12 apporte des changements esthétiques et pratiques par rapport à Android 11. Le panneau de notifications profite d'un design plat (flat design), le <i>double-tap</i> comme sur iOS fait son apparition tout comme une petite icône indiquant si le micro ou la caméra fonctionne. Si votre smartphone est	

	compatible, vous pouvez d'ores et déjà installer la bêta. La version stable est attendue pour plus tard dans l'année.	
--	---	--

Tableau 02 : Les différentes versions Android

3. l'architecture logicielle :

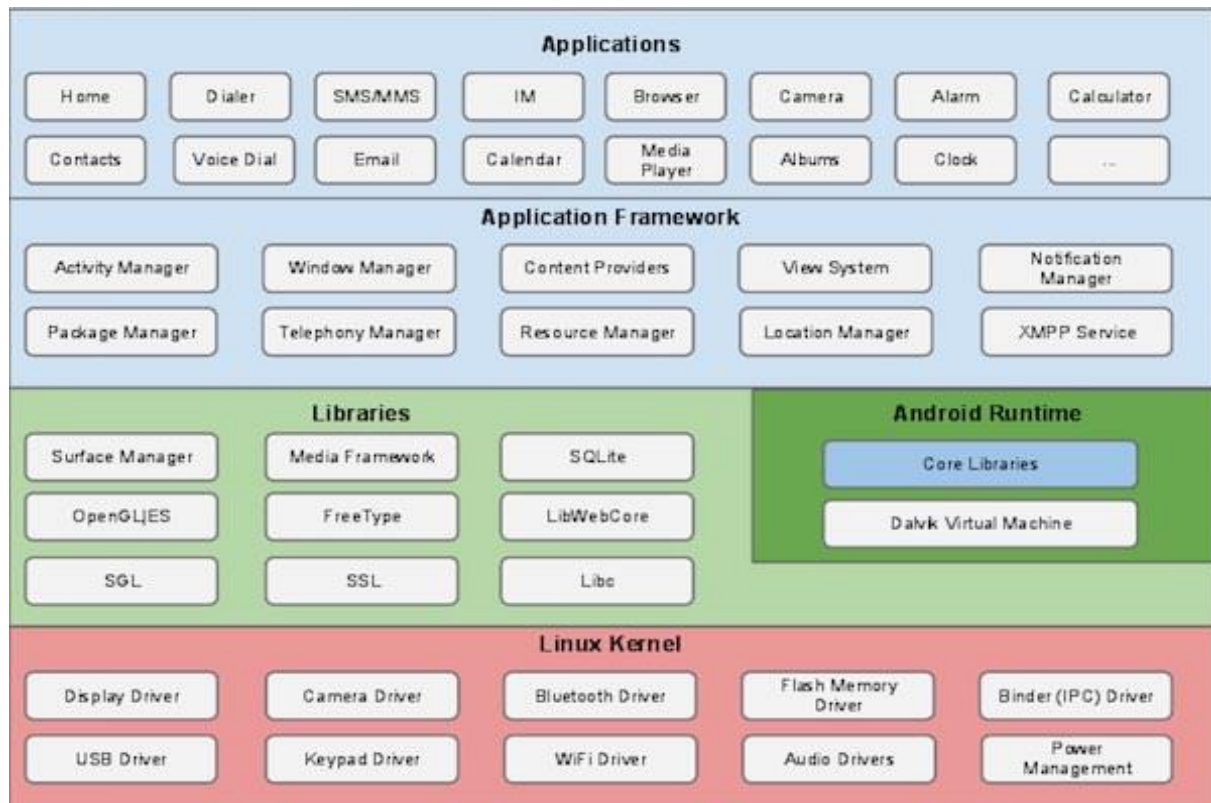


Figure 07 : Architecture logicielle de l'Android.

a) Noyau Linux :

Sous la plupart de toutes les couches est Linux - y compris environ 115 Patch Linux 3.6. Il fournit des fonctions de base du système, telles que la gestion des processus, gestion de la mémoire, la gestion des périphériques (tels que caméra, clavier, écran). Pendant ce temps, le noyau Linux gère tout le bon travail, et beaucoup d'équipements tels que les pilotes de réseau, afin d'éviter un grand nombre d'interface matériel périphérique compatible inconvenients.

b) Bibliothèque :

Dans la couche supérieure du noyau Linux est un ensemble de bibliothèques, y compris Webkit moteur de navigateur Web open-source, connue bibliothèque libc pour l'entreposage de données et base de données SQLite de partage d'application pour la lecture, l'enregistrement audio et les bibliothèques vidéo, bibliothèques SSL pour la sécurité du réseau.

c) Android Runtime :

Ceci est la troisième partie de l'architecture, le second fond de la couche. Cette section fournit la composante clé appelée machine virtuelle Dalvik, semblable à la machine virtuelle Java, mais conçu et optimisé spécifiquement pour Android.

Machine virtuelle Dalvik permet d'utiliser les fonctionnalités du noyau Linux en Java, tels que la gestion de la mémoire et multi-threading. Machine virtuelle Dalvik telle que toutes les applications Android fonctionnent dans son propre processus de machine virtuelle séparée.

Android offre une gamme de bibliothèques de base en cours d'exécution dans le même temps d'utiliser le langage des développeurs d'applications Java standard Android pour écrire des applications Android.

d) Application Framework :

Couche application-cadre sous la forme de classes Java offrent de nombreux services avancés pour l'application. Les développeurs d'applications sont autorisés à utiliser ces services dans l'application.

- Events Manager - Contrôle des activités applications du cycle de vie et de tous les aspects de la pile.
- Les fournisseurs de contenu - permet l'édition et le partage des données entre les applications.
- Resource Manager - permet d'accéder à des ressources non code embarqués, tels que des chaînes, des paramètres de couleur et interface utilisateur mise en page.
- Gestionnaire de notification - Permet à l'application affiche une boîte de dialogue ou d'une notification à l'utilisateur.
- Vues système - un ensemble évolutif de vues pour créer des interfaces utilisateur de l'application.

e) Applications :

La couche supérieure à toutes les applications Android. Vous écrivez des applications seront installées dans cette couche. Ces applications comprennent des contacts, navigateur, jeux. (9)

4. Kit de développement :

Un kit de développement logiciel (**Software Développement Toolkit ou SDK**) est un ensemble d'outils fourni avec une plateforme matérielle, un système d'exploitation ou un langage de programmation. Il permet aux développeurs de logiciels de créer des applications propres à cette plateforme, ce système ou ce langage de programmation. C'est un peu comme une boîte à outils, ou comme le sachet d'outils fourni avec les éléments d'un meuble à assembler soi-même, mais pour développer une application. Il renferme tous les composants, ou outils de développement, nécessaires pour effectuer la tâche, et son contenu varie selon le fabricant

Un SDK de base comprend un compilateur, un débogueur et des interfaces de programmation d'application (API). Il peut également contenir tout ou partie des éléments suivants :

- Documentation
- Bibliothèques
- Éditeurs
- Environnements de développement et/ou d'exécution
- Outils de test et d'analyse
- Pilotes
- Protocoles réseau (10)

IV. Les bases de données :

Les bases de données informatiques sont utilisées dans un grand nombre d'entreprises pour stocker, organiser et analyser les données. Découvrez tout ce que vous devez savoir à ce sujet : qu'est-ce qu'une base de données, à quoi sert-elle, comment fonctionne-t-elle, quelles sont les différentes catégories, et quelles sont les meilleures.

Une base de données (que nous nommerons BDD par commodité) est une collection d'informations organisées afin d'être facilement consultables, gérables et mises à jour. Au sein d'une database, les données sont organisées en lignes, colonnes et tableaux. Elles sont indexées afin de pouvoir facilement trouver les informations recherchées à l'aide d'un logiciel informatique. Chaque fois que de nouvelles informations sont ajoutées, les données sont mises à jour, et éventuellement supprimées.

1. les différents types de bases de données :

a) Base de données hiérarchique :

Les bases de données hiérarchiques comptent parmi les plus anciennes bases de données. Au sein de cette catégorie, les enregistrements sont organisés dans une structure d'arborescence. Chaque niveau d'enregistrements découle sur un ensemble de catégories plus petites.

b) Base de données réseau :

Les bases de données réseau sont également parmi les plus anciennes. Plutôt que de proposer des liens uniques entre différents ensembles de données à divers niveaux, les bases de données réseaux créent des liens multiples entre les ensembles en plaçant des liens, ou des pointeurs, sur un ensemble d'enregistrements ou un autre. La vitesse et la polyvalence des bases de données réseau ont conduit à une adoption massive de ce type de databases au sein des entreprises ou dans le domaine du e-commerce.

c) Base de données SQL (relationnelle) :

Les bases de données relationnelles ont été inventées en 1970 par E.F. Codd de IBM. Il s'agit de documents tabulaires dans laquelle les données sont définies afin d'être accessibles et de pouvoir être réorganisées de différentes manières.

Les bases de données relationnelles sont constituées d'un ensemble de tableaux. Au sein de ces tableaux, les données sont classées par catégorie. Chaque tableau comporte au moins une colonne correspondant à une catégorie. Chaque colonne comporte un certain nombre de données correspondant à cette catégorie.

L'API standard pour les bases de données relationnelles est le **Structured Query Language (SQL)**. Les bases de données relationnelles sont facilement extensibles, et de nouvelles catégories de données peuvent être ajoutées après la création de la database originale sans avoir besoin de modifier toutes les applications existantes.

d) Base de données distribuée :

Une BDD distribuée est une database dont certaines portions sont stockées à plusieurs endroits physiques. Le traitement est réparti ou répliqué entre différents points d'un réseau. Les bases de données distribuées peuvent être homogènes ou hétérogènes. Dans le cas d'un système de base de données distribuée homogène, tous les emplacements physiques fonctionnent avec le même hardware et tournent sous le même système d'exploitation et les mêmes applications de bases de données. Au contraire, dans le cas d'une database distribuée

hétérogène, le hardware, les systèmes d'exploitation et les applications de bases de données peuvent varier entre les différents endroits physiques. (12)

V. la Géolocalisation :

Technologie permettant de déterminer la localisation d'un objet ou d'une personne avec une certaine précision. La technologie s'appuie généralement sur le système GPS ou sur les interfaces de communication d'un téléphone mobile. Les applications et finalités de la géolocalisation sont multiples : de l'assistance à la navigation, à la mise en relation des personnes, mais aussi à la gestion en temps réel des moyens en personnel et en véhicules des entreprises, etc.(13)

1-GPS :

Le Système de positionnement mondial (GPS, Global Positioning System) est un utilitaire qui appartient aux Etats-Unis et qui assure des services de positionnement, de navigation et de référence temporelle, dits « services PNT » (positioning, navigation, and timing). Il se compose de trois segments : le segment spatial, le segment de contrôle et le segment utilisateur. L'Armée de l'Air des Etats-Unis assure le développement, l'entretien et le fonctionnement du segment spatial et du segment de contrôle.(14)

2-Google Maps :

Est un service gratuit de carte géographique et de plan en ligne. Le service a été créé par Google. Il s'agit d'une forme de géoportail. Lancé en 2004 aux États-Unis et au Canada et en 2005 en Grande Bretagne (sous le nom de Google Local), Google Maps a été lancé jeudi 27 avril 2006, simultanément en France, Allemagne, Espagne et Italie.

Ce service a ceci de particulier qu'il permet, à partir de l'échelle d'un pays, de pouvoir zoomer jusqu'à l'échelle d'une rue. Deux types de plan sont disponibles : un plan classique, avec nom des rues, quartier, villes et un plan en image satellite, qui couvre aujourd'hui le monde entier. Ce service est actuellement en bêta test .(15)

Conclusion :

Dans ce chapitre, nous avons présenté les applications mobiles, les technologies associées et plus spécifiquement le système d'exploitation Android en étalant ces fonctionnalités, son architecture ainsi que les principaux composants du système. Ensuite nous avons présenté une idée générale sur les bases de données.

Chapitre 02 :

Étude préliminaire

I. L'Étude préliminaire :

1. Introduction :

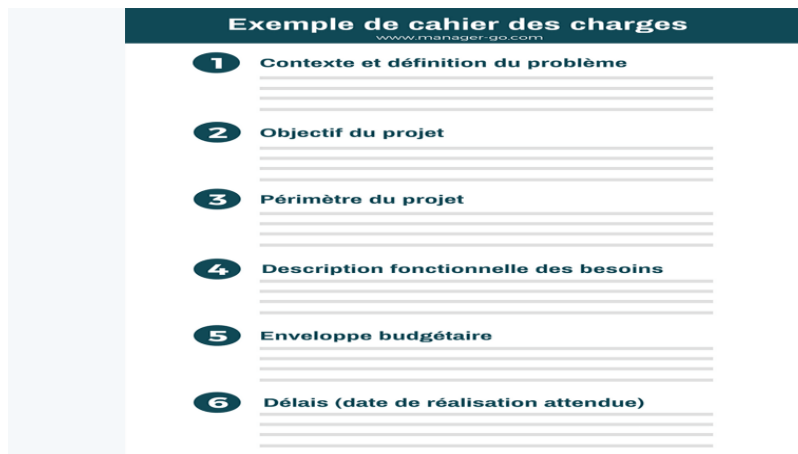
L'étude préliminaire est la toute première étape du processus zTUP. Elle consiste à effectuer un premier repérage des besoins fonctionnels et opérationnels, en utilisant principalement le texte, ou diagrammes très simples. Elle prépare les activités plus formelles de capture des besoins fonctionnels et de capture technique. (1)



Figure 08:la méthode zTUP

2. Élaboration du cahier des charges :

Un cahier des charges constitue sur le plan légal un document contractuel. Pour une application mobile, il permet d'intégrer des jalons de validation dans le déroulement du projet, et de formuler des objectifs. Il engage toutes les parties prenantes du projet dans leurs missions et obligations respectives, du commanditaire aux équipes techniques. (2)



Exemple de cahier des charges

www.manager-go.com

- 1 Contexte et définition du problème
- 2 Objectif du projet
- 3 Périmètre du projet
- 4 Description fonctionnelle des besoins
- 5 Enveloppe budgétaire
- 6 Délais (date de réalisation attendue)

Figure 09 : exemple de Cahier de charge

L'élaboration du cahier de charges s'effectue sur plusieurs étapes :

- La présentation du projet.
- Les grands choix techniques.
- Identifications des acteurs.
- Les besoins fonctionnels.
- Les besoins opérationnels.

2.1. La présentation du projet :

○ **Équipe du projet :**

Ce projet est à réaliser par les deux étudiantes TEBBOUB IKRAM et AZIZA KHADIDJA qui sont inscrits dans la spécialité 'STIC' en informatique master2 à l'Institut de science et technologie du centre universitaire de Mila.

○ **Encadrement**

Ce travail est dirigé par Mr. Hakim Bessouf. Maître-Assistant classe A au même établissement. C'est lui qui a proposé l'idée du projet

○ **Cadre de réalisation :**

Ce projet est réalisé dans le cadre de la préparation d'un diplôme master2 spécialité 'STIC' en informatique à l'institut des sciences et technologies du centre universitaire de Mila.

○ **Durée du projet :**

Ce projet s'est étalé sur la durée de 5 mois. À partir du AVRIL 2021 au SEPTEMBRE 2021.

○ **Problématique :**

Parfois, les gens et les entreprises ont besoin de se déplacer ou d'apporter leurs Marchandises, mais ils trouvent un problème, qui est de trouver le livreur en fonction de sa localisation (elle peut être éloignée) et du moyen de transport utilisé. Cela entraîne des retards dans leur travail, En plus de gaspiller leur argent.

Dans ce travail, nous proposons une solution pour faciliter la recherche et le choix de livreurs dans la foule.

○ **Objectif :**

Nous développons une application mobile pour les Smartphones et les tablettes tactiles sous ANDROID qui aura vocation à être un portail d'information pour les clients afin de chercher et situer le livreur le plus proche.

L'application utilise GPS pour pouvoir voir la localisation de ces livreurs, ainsi d'avoir la possibilité de les contacter Avec eux. Ce travail simplifiera la façon avec laquelle est cherchée géographiquement et très rapidement les livreurs existant dans la région.

2.2. Les grands choix techniques :

Nous allons utiliser d'un certain nombre de techniques-clés qui sont principalement :

- Le langage JAVA pour la programmation de la logique applicative de ce projet dans une plateforme Android.
- L'adoption d'une architecture client/serveur.
- Pour la modélisation on va utiliser la méthode de développement 2TUP à partir de langage standard UML
- SGBD mysql pour le stockage et la gestion des données.

2.3. Identifications des acteurs :

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié.

Un acteur peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données. (3)

On peut distinguer deux types des acteurs :

✓ **Les acteurs principaux :**

Ceux qui vont réaliser le cas d'utilisation (la relation avec le cas d'utilisation est illustrée par le trait liant le cas d'utilisation et l'acteur dans un diagramme de cas d'utilisation)

✓ **Les acteurs secondaires :** ceux qui ne font que recevoir des informations à l'issue de la réalisation du cas d'utilisation.

Les différents acteurs qui doivent interagir dans notre système sont :

- **Acteurs principaux :**

Le client : c'est la personne qui prend la décision pour livrer ou bien de recevoir un colis à partir de choisir un livreur sur l'application.

Le livreur : C'est la personne de livraison qui choisit par le client à partir de l'application.

L'admin : C'est la personne qui gère les comptes et les activités des livreurs.

- **Acteur secondaire : Baridi MOB** : interface de paiement.

2.4. Les besoins fonctionnels :

- **Passer une demande**

Le client ajoute les informations d'une demande de livraison, et envoie à tous les Livreurs près de lui.

- **Consulter offre (choisir une offre)**

Le client consulte la liste des offres et choisit le meilleur offre pour sa demande

- **Consulter état de livraison**

après de choisi une offre le client peut consulte l'état de livraison de sa demande

- **Valider livraison du colis (payer) :**

Après la réception du colis le client doit être payé et confirmer sa réception.

- **Passer Feedback :**

Le client qui veut évalue le livreur pour sa livraison fait cette démarche.

- **Consulter les demandes de livraison (proposer une offre) :**

Le livreur peut voir les demandes de livraison des clients et peut faire une offre.

- **Valider réponse client :**

Après le choix du client, le livreur doit accepter ou refuser la demande.

- **Mettre à jour l'état de livraison :**

Le livreur pendant l'opération de livraison doit être modifier l'état de demande.

- **Consulter Feedback (avertir compte) (bloquer compte) :**

L'administrateur de système consulte les feedbacks de livreur et peut d'avertir ou bloquer le en cas de négligence des demandes.

- **Consulter activités :**

A la fin de journée l'administrateur consulte les activités effectuées par chaque livreur et autorise son paiement.

- **Payer livreur**

après l'autorisation de l'administrateur la plateforme envoie le montant total de chaque livreur.

2.5. Les besoins opérationnels :

Après avoir déterminé les besoins fonctionnels, nous présentons ci-dessous l'ensemble des contraintes à respecter les besoins de l'utilisateur et qui peut faire face à des risques de panne ou de non fonctionnement :

- **Performance** : Afin que notre travail puisse être accepté par le client, notre application doit respecter le critère de performance tout en assurant un temps de réponse minimum et des fonctionnalités répondant aux besoins de l'utilisateur.
- **La simplicité** : Un visiteur assez modeste pour utiliser un tel service de façon intuitive.
- **L'ergonomie de l'interface** : Les interfaces doivent être simples et conviviales : on doit essayer le maximum d'éliminer l'encombrement.
- **La modularité de l'application** : Avoir un code simple facile à maintenir et comprendre en cas de besoin.

3. Description de contexte de système à réaliser :

Une fois ce premier recueil de besoins effectué, la description du contexte du système peut commencer. Elle consiste en deux activités successives :

- l'identification des messages,
- la réalisation des diagrammes de contexte.

3.1. Identifications des messages :

Un message est un élément de diagramme Unified Modeling Language (UML) qui définit un type particulier de communication entre les instances au cours d'une interaction. Un message fait circuler des informations d'une instance, représentée par une ligne de vie, à une autre instance au cours d'une interaction. (4)

✓ Les messages émis par le système :

- Informations des livreurs
- Informations des demandes
- Information des offres
- Affichage des formulaires
- Les notifications

✓ Les messages reçus par le système :

- Demande d'authentification
- Demande de consultation, de modification
- Demande d'inscription
- Passer les demandes
- Proposer les offres
- payer

3.2. Le diagramme de contexte de système :

Le diagramme de contexte est l'un des premiers diagrammes pouvant être réalisé afin de décrire à haut niveau le futur système en mode boîte noire. Il permet de se concentrer sur l'identification des différents acteurs interagissant avec lui. Il est aussi possible de représenter les interactions entre acteurs si cela semble pertinent pour la compréhension globale du système. Lors de sa réalisation on ne parle ni d'implémentation, ni de solution, mais seulement de son contexte d'utilisation. (5)

Le schéma suivant représente le diagramme de contexte de système :

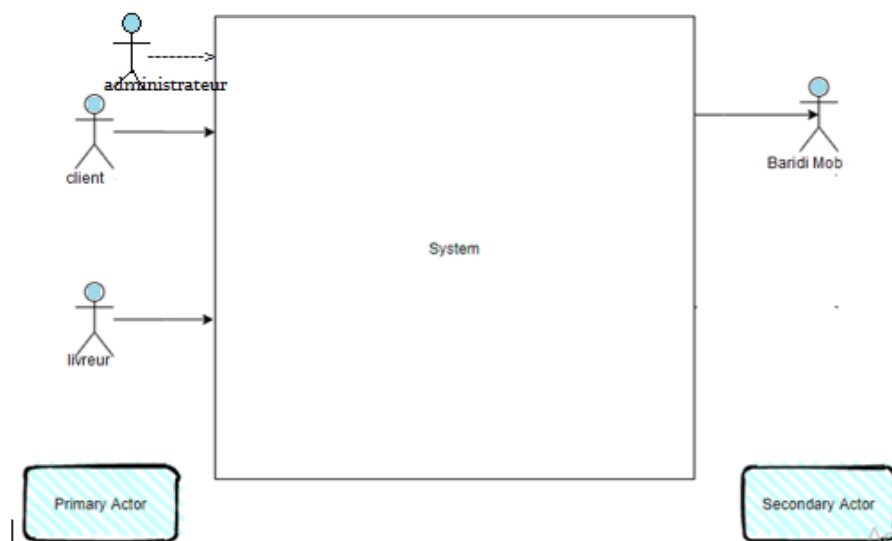


Figure 10: Diagramme de contexte

- Légende des messages du diagramme de contexte dynamique :

Messages reçues	Messages émises
1. Client vers system : - Demande l'inscription - Demande d'authentification - Passe des demandes - Demande de consultation des offres - Demande de consultation d'état de livraison - Passe Feedback	2. System vers client : - Affichage des informations d'offres - Affichage des informations d'état - Affichage des formulaires

3. Livreur vers system : - Demande l'inscription - Demande d'authentification - Demande consultation des demandes - Valider réponse - Mettre a jour l'état de livraison	4. System vers livreur - Affichage des formulaires - Affichage des informations des demandes
---	--

Tableau 03 : Légende des messages de diagrammes de contexte

Conclusion :

Nous avons représenté dans ce chapitre l'étude préliminaire du développement d'une application mobile. Aussi Nous avons défini les grands choix techniques, et déterminer les besoins fonctionnels et opérationnels de notre projet en utilisant principalement du texte ou des diagrammes très simples.

CHAPITRE₀₃ :

Capture des besoins

Introduction :

Le but de ce chapitre est, la capture des besoins fonctionnels et la capture des besoins techniques du processus de développement zTUP, c'est l'une des phases les plus importante car elle doit permettre aux utilisateurs finaux et/ou au maître d'œuvre, qui ne sont pas toujours des informaticiens, de bien exprimer leurs besoins et de bien comprendre les fonctionnalités que le système va fournir.

I. Situation de l'étape dans le processus zTUP :

zTUP est un processus de développement logiciel qui implémente le processus unifié). Chacune des étapes du cycle découle des précédentes. Il préconise un cycle de vie en y, et s'apparente à un cycle de développement en cascade, par ailleurs elle est incrémentale : à partir de la capture des besoins fonctionnels, on définit plusieurs cas d'utilisation représentant chacun un incrément du cycle de développement. Elle favorise des formes de recherche de qualité et de performance intéressante telle que le service réutilisation.(1)

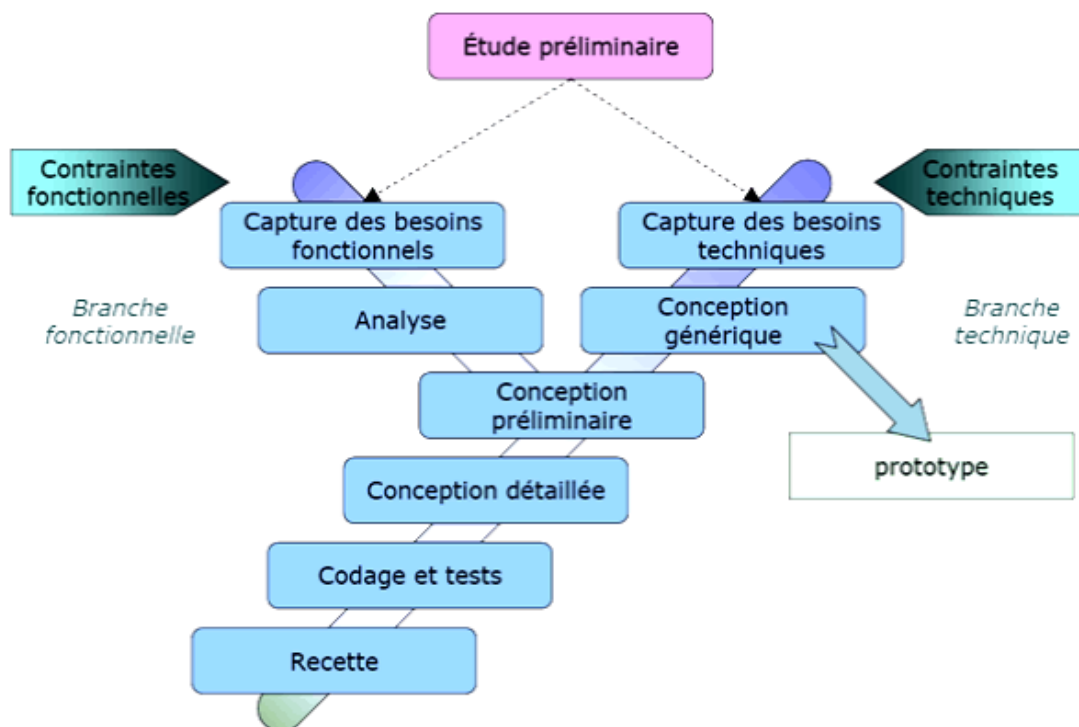


Figure 11 : les étapes de processus zTUP

1.Capture des besoins fonctionnels :

La capture des besoins fonctionnels est la première étape de la branche gauche du cycle en Y. Elle formalise et détaille ce qui a été ébauché au cours de l'étude préliminaire. Elle est

complétée au niveau de la branche droite du Y par la capture des besoins techniques et prépare l'étape suivante de la branche gauche :analyse.

1.1 L'analyse :

1.1.1. Identification des cas d'utilisation :

num	Cas d'utilisation	Acteur
1	- S'authentifier - S'inscrire - Consulter profile - Modifier profile	Utilisateur
2	- Consulter l'état de livraison - Passer demande de livraison - Valider la livraison d'un colis - Consulter les offres de livraison - Choisir une offre - Payer	Client
3	- Consulter les demandes de livraison - Proposer une offre - Valider réponse client - Mettre à jrs l'état de livraison	Livreur
4	- Consulter feedback - Avertir compte - Bloque compte - Consulter activités - Payer livreur	Admin

Tableau 04 : Liste des cas d'utilisations fonctionnels.

1.1.2. Diagramme de cas d'utilisation :

Les diagrammes de cas d'utilisation décrivent les fonctions générales et la portée d'un système. Ces diagrammes identifient également les interactions entre le système et ses acteurs. Les cas d'utilisation et les acteurs dans les diagrammes de cas d'utilisation décrivent ce que le système fait et comment les acteurs l'utilisent, mais ne montrent pas comment le système fonctionne en interne. (2)

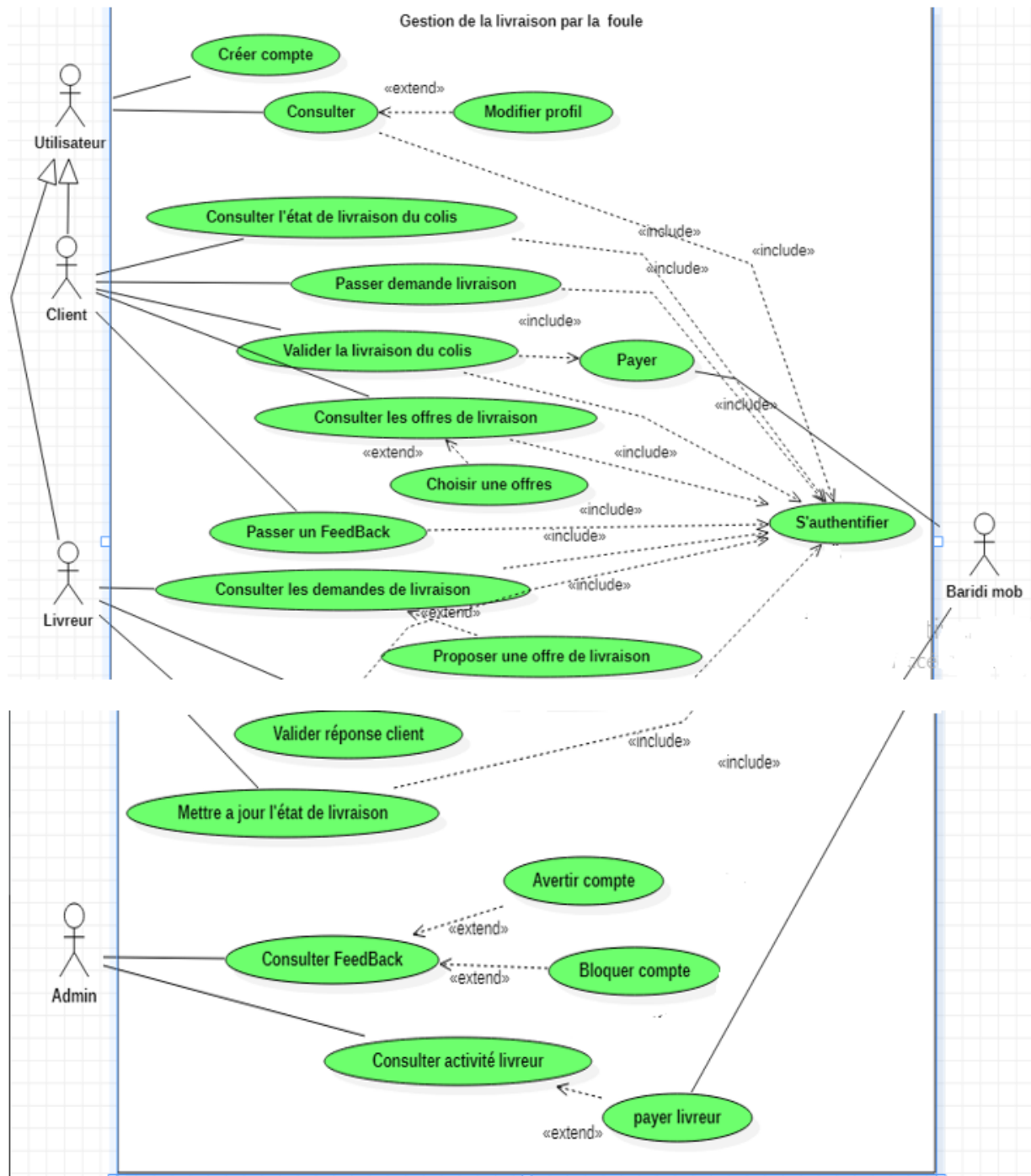


Figure 12 : Diagramme de cas d'utilisation générale

1.1.3. Description des cas d'utilisation :

Dans cette phase, on décrit les différentes descriptions graphiques et descriptions Textuelles relatifs aux cas d'utilisations.

- **Description textuelle « Passer demande livraison »**

Cas d'utilisation	Passer demande livraison
Acteur principale	Client
Acteur secondaire	/
Objectifs	Permet au client de passer une demande
Pré condition	Le client doit être authentifié
Post condition	Une demande sera enregistrée et envoyée aux livreurs
Scénario nominal	1-Le client demande de passer une demande de livraison 2-le système affiche l'interface pour passer la demande 3- Le client saisit les informations nécessaires 4-Le client valide les informations saisies. 5- Le système vérifie la validité des informations ,et envoie un message de confirmation.
Scenario alternative	4.a-des champs obligatoires non remplis. Le système affiche un Message d'erreur. 4.b- Reprend le scenario du point 2.

Tableau 05 : Description textuelle « Passer demande livraison »

- **Description graphique « Passer demande livraison »**

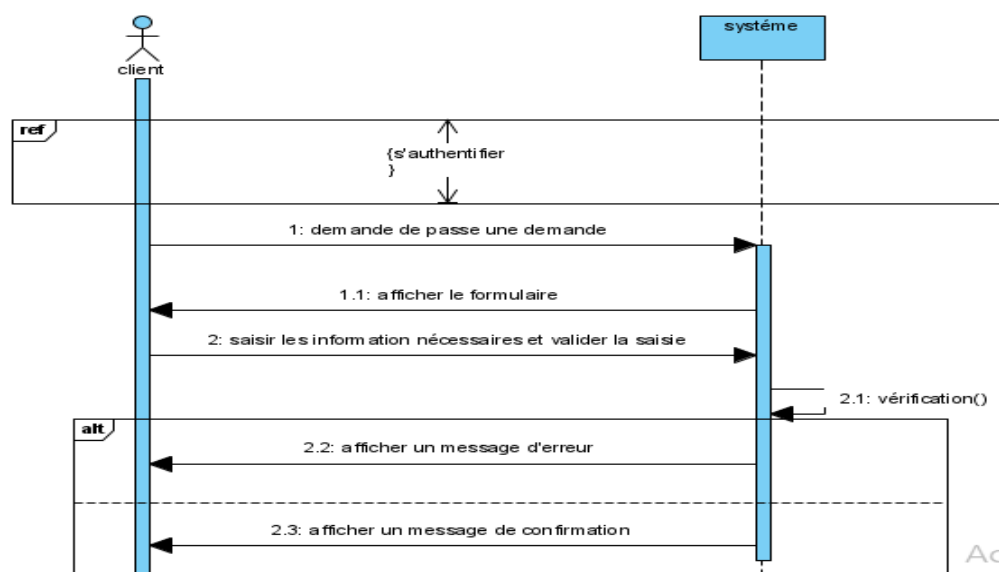


Figure 13 : Description graphique « Passer demande livraison »

• **Description textuelle « consulter les offres de livraison »**

Cas d'utilisation	Consulter les offres de livraison
Acteur principale	Client
Acteur secondaire	/
Objectifs	Affiche tous les offres de livraison
Pré condition	Le client doit être authentifié Au moins une offre est déjà passée
Post condition	Permet de consulter les offres
Scénario nominal	1-le client demande de consulter les offres 2-le system affiche la liste des demandes 3-le client choisit une demande 4-le système affiche la liste des offres de la demande choisi
Scenario alternative	

Tableau o6: Description textuelle « consulter les offres de livraison »

• **Description graphique « consulter les offres de livraison »**

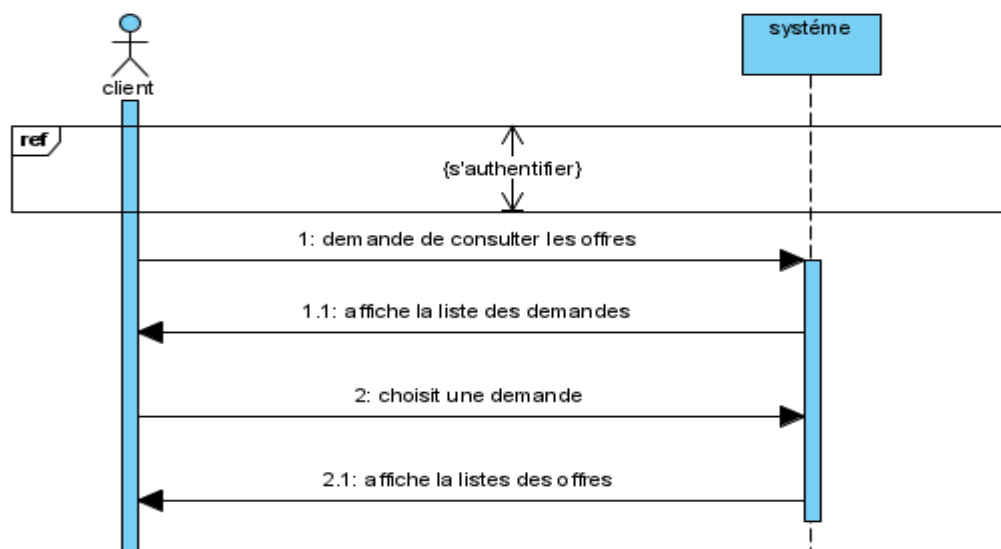


Figure 14 : Description graphique « consulter les offres de livraison »

• Description textuelle « choisir une offre »

Cas d'utilisation	Choisir une offre
Acteur principale	Client
Acteur secondaire	/
Objectifs	Permet au client de choisir une offre
Pré condition	Le client doit être authentifié et consulter les offres de livraison
Post condition	L'envoi d'acceptation d'offre au livreur
Scénario nominal	1-le client sélectionné une offre dans la liste des offres 2-le system affiche les détails d'offre 3-le client clique sur bouton « choisir » 4-le système demande de confirmer le choix 5-le client confirmer
Scenario alternative	2. le client clique sur bouton « précédente » . Reprend le scenario du point 1.

Tableau 07: Description textuelle « choisir une offre »

• Description graphique « choisir une offre »

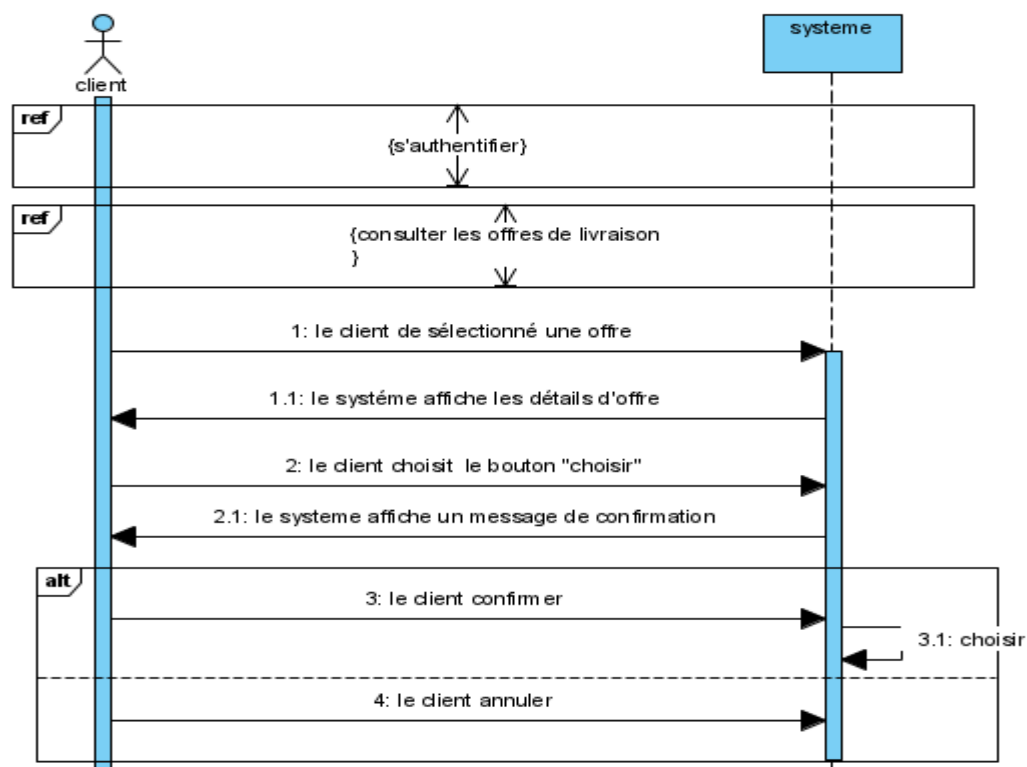


Figure 15 : Description graphique « choisir une offre »

• **Description textuelle « consulter l'état de livraison d'un colis »**

Cas d'utilisation	Consulter l'état de livraison d'un colis
Acteur principale	Client
Acteur secondaire	/
Objectifs	Suivi l'état de livraison d'un colis
Pré condition	Le client doit être authentifié Demande a été accepter par le livreur
Post condition	Permet au client de suivi l'état de livraison
Scénario nominal	1-le client demande de consulter l'état de livraison 2- le système affiche la liste des demandes du client 3-le client sélectionné une demande 4-le system affiche l'état
Scenario alternative	/

Tableau o8: Description textuelle « consulter l'état de livraison d'un colis »

• **Description graphique « consulter l'état de livraison d'un colis »**

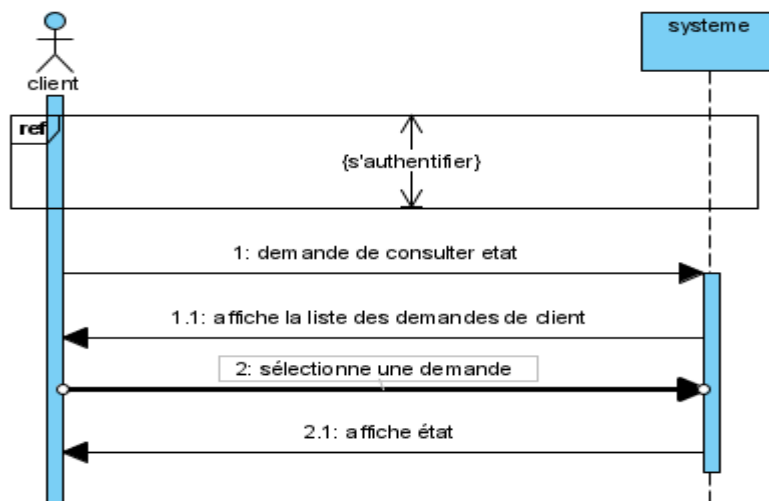


Figure 16: Description graphique « consulter l'état de livraison d'un colis »

• **Description textuelle « Valider la livraison du colis »**

Cas d'utilisation	Valider la livraison du colis
Acteur principale	Client et le système de paiement
Acteur secondaire	/
Objectifs	Ce cas permet au client de valider la livraison du colis et envoyer le paiement
Pré condition	Le client doit être authentifié
Post condition	Le paiement effectué et la livraison du colis validée
Scénario nominal	1-Le client demande de valider la livraison 2-le système affiche la liste des demandes 3-le client choisit la demande qui veut valider 2-le système renvoie l'interface de paiement « Baridi mob » 3-le client remplit le formulaire de paiement et valide la livraison 4- le système affiche un message d'information « livraison validée »
Scenario alternative	3-des champs obligatoires non remplis. Le système affiche un Message d'erreur. Reprend le scenario du point 2.

Tableau 09 : Description textuelle « Valider la livraison du colis »

• **Description graphique « Valider la livraison du colis »**

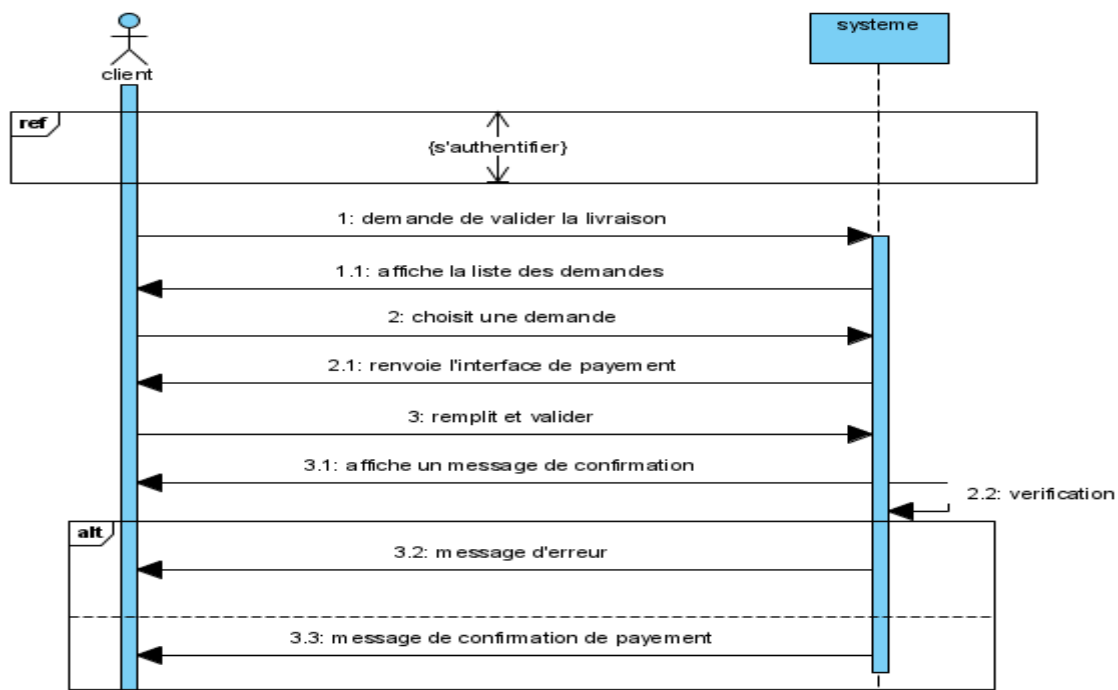


Figure 17 : Description graphique « Valider la livraison du colis »

• **Description textuelle « passer un Feedback »**

Cas d'utilisation	Passer un feedback
Acteur principale	Client
Acteur secondaire	/
Objectifs	Permet au client d'évaluer le livreur
Pré condition	Le client doit être authentifié
Post condition	L'évaluation est enregistrée dans la base de données.
Scénario nominal	1-Le client demande de Passer un FeedBack 2- Le système affiche la liste des livreurs qui ont terminé leur livraison 3- le client sélectionne le livreur et remplit le champ et valide 4- le système affiche un message de confirmation 5-le client confirmer
Scenario alternative	5.a. Le client annule le processus d'évaluation 5.b. Le système retourne à l'étape numéro 2.

Tableau 10 : Description textuelle « passer un Feedback »

• **Description graphique « passer un Feedback »**

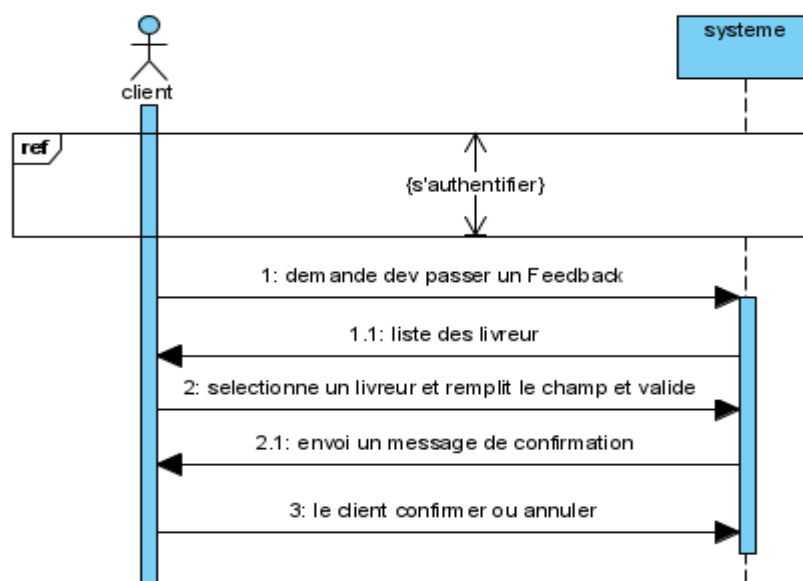


Figure 18 :graphique « passer un Feedback »

- **Description textuelle « Consulter les demandes »**

Cas d'utilisation	Consulter les demandes
Acteur principale	Livreur
Acteur secondaire	/
Objectifs	Permet au livreur voir toutes les demandes de livraison
Pré condition	Le livreur doit être authentifié Au moins une demande est déjà passée
Post condition	Affiche tous les demandes de livraison
Scénario nominal	1-le livreur clique sur le bouton « consulter les demandes » 2-le system affiche la liste des demandes
Scenario alternative	/

Tableau 11 : Description textuelle « Consulter les demandes »

- **Description graphique « Consulter les demandes »**

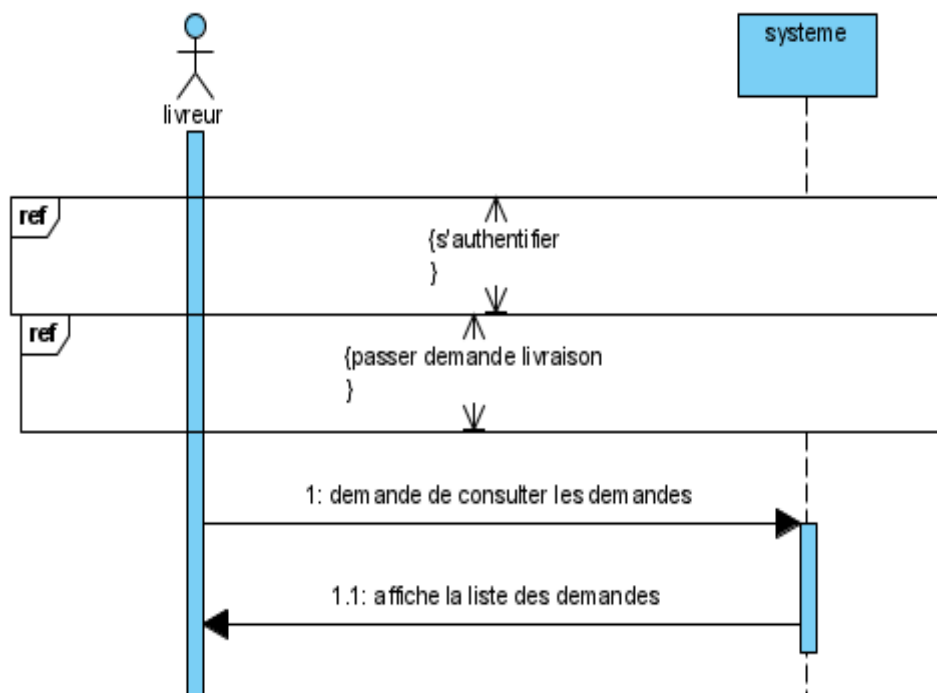


Figure 19: Description graphique « Consulter les demandes »

• Description textuelle « valider réponse client »

Cas d'utilisation	Valider réponse client
Acteur principale	Livreur
Acteur secondaire	/
Objectifs	Permet au livreur d'accepter ou refuser la livraison
Pré condition	Le livreur doit être authentifié L'offre a déjà accepté par le client
Post condition	Réponse envoi au client
Scénario nominal	1-le livreur demande de consulter demande 2-le système affiche la liste des demandes 2-le livreur sélectionne une demande 3-le système affiche les détails de l'offre et la demande 4.le livreur envoi la réponse 5.l système affiche un message de confirmation 6.le livreur confirmer
Scenario alternative	6.a. le client annuler 6.b. reprend à l'étape 3 de Scénario nominal

Tableau 12 : Description textuelle « valider réponse client »

• Description graphique « valider réponse client »

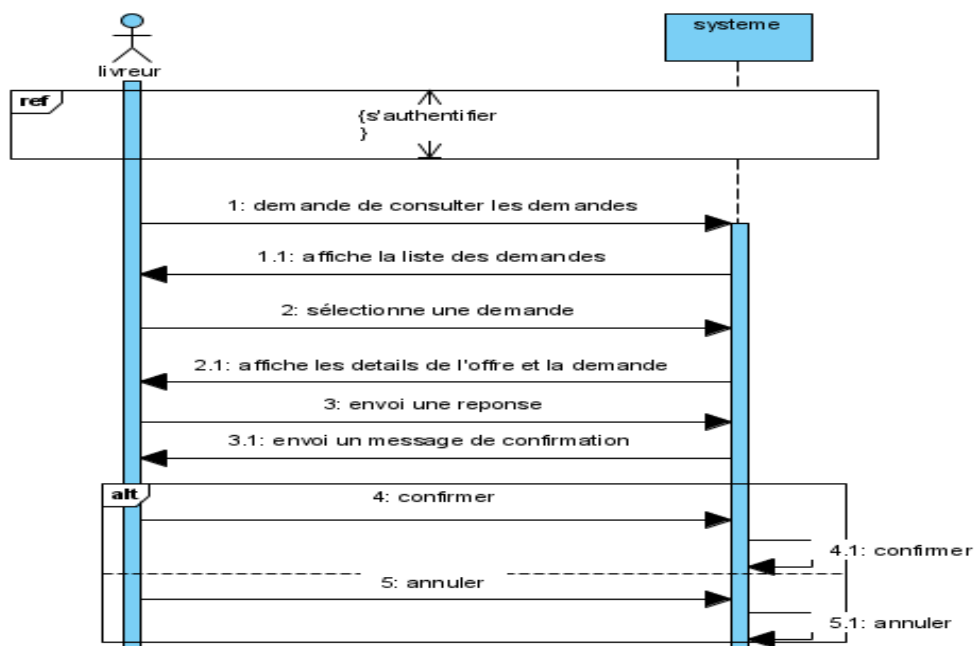


Figure 20: Description graphique « valider réponse client »

• **Description textuelle « proposer une offre de livraison »**

Cas d'utilisation	Proposer une offre livraison
Acteur principale	Livreur
Acteur secondaire	/
Objectifs	Permet au livreur d'envoyer une offre au client
Pré condition	Le livreur doit être s'authentifier Une demande a été déjà envoyer et consulter
Post condition	Une offre sera enregistrée et envoyée aux client
Scénario nominal	1-Le livreur choisit une demande 2-le système affiche l'interface d'offre 3- Le livreur saisit les informations nécessaires 4-Le livreur valide les informations saisies. 4- - Le système vérifie la validité des informations
Scenario alternative	4.a-des champs obligatoires non remplis. Le système affiche un Message d'erreur. 4.b- Reprend le scenario du point 2.

Tableau 13 : Description textuelle « proposer une offre »

• **Description graphique « proposer une offre »**

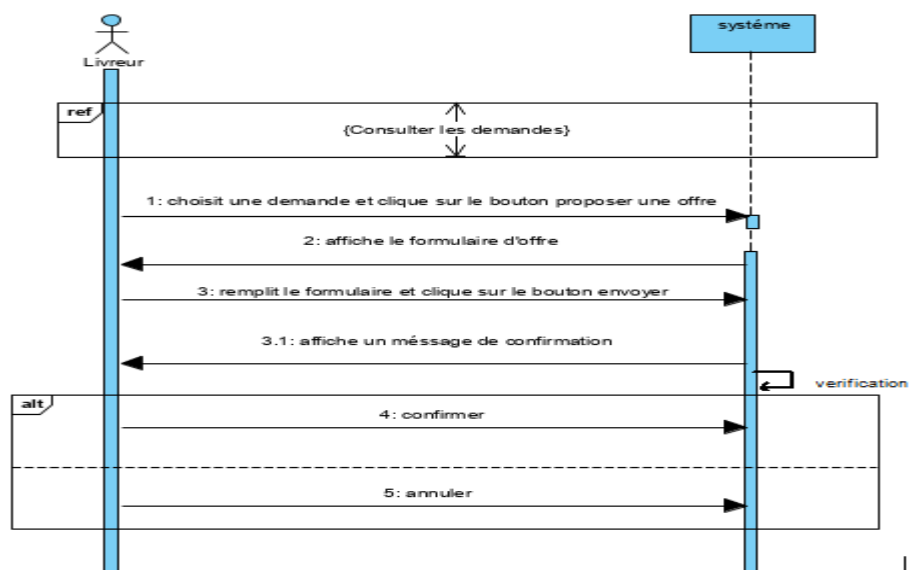


Figure 21 : Description graphique « proposer une offre de livraison »

• Description textuelle « Mettre à jour l'état de livraison »

Cas d'utilisation	Mettre a jour l'état de livraison
Acteur principale	Livreur
Acteur secondaire	/
Objectifs	Permet au livreur de changer l'état de livraison
Pré condition	Le livreur doit être authentifié La demande a été déjà accepter
Post condition	L'état de livraison a été changé
Scénario nominal	1-Le livreur demande de mètre a jour l'état de livraison 2-le système affiche la liste des demandes accepter 3- le client sélectionne une demande 4-Le système affiche les détails de la demande 4- le client change l'état 5-le système envoie un message de confirmation 6-le livreur confirmer 7-le système enregistrer la modification
Scenario alternative	6.a- le livreur annuler la mise a jour 6.b- Reprend le scenario du point 2 .

Tableau 14: Description textuelle « Mettre à jour l'état de livraison »

• Description graphique « Mettre à jour l'état de livraison »

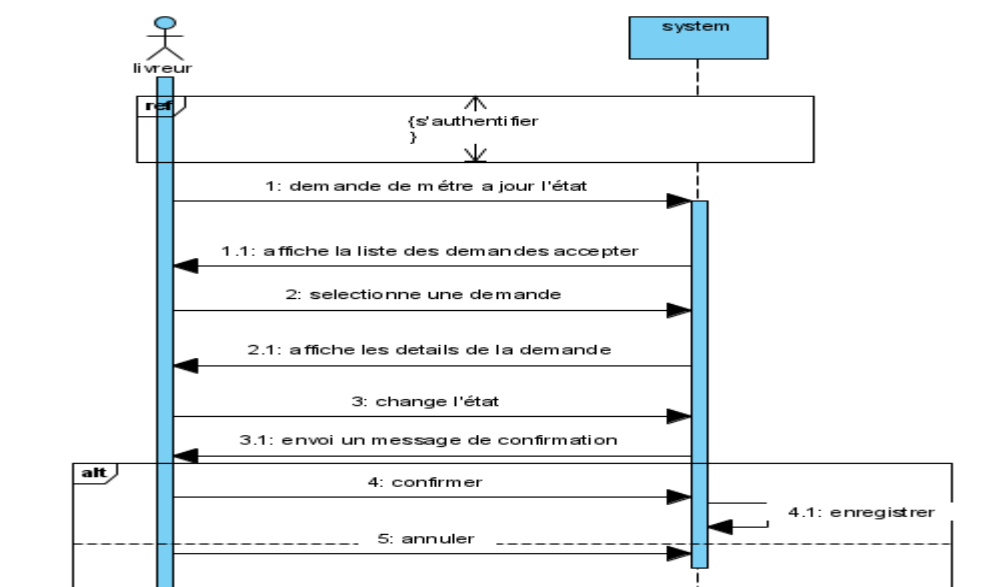


Figure 22 : Description graphique « Mettre à jour l'état de livraison »

• **Description graphique « Consulter Feedback»**

Cas d'utilisation	Consulter Feedback
Acteur principale	Admin
Acteur secondaire	/
Objectifs	Permet au admin de voir toutes les Feedback
Pré condition	L'admin doit être authentifié
Post condition	Consulter tous les Feedback
Scénario nominal	1-l'admin demande de consulter Feedback 2-le system affiche la liste des comptes des livreurs 3-l'admin sélectionne un compte 4-le système affiche le FeedBack
Scenario alternative	/

Tableau 15 : Description textuelle « « consulter Feedback » »

• **Description graphique « Consulter Feedback »**

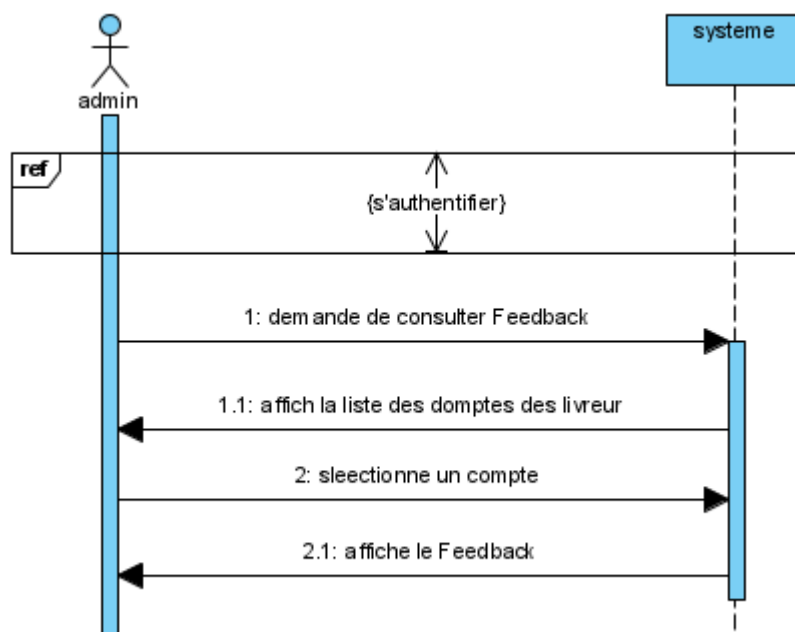


Figure 23 : Description graphique « Consulter Feedback»

• Description textuelle « Avertir compte »

Cas d'utilisation	Avertir compte
Acteur principale	Admin
Acteur secondaire	/
Objectifs	Permet au admin d'envoyer un avertissement au livreur
Pré condition	L'admin doit être authentifié Les Feedback a été consulter
Post condition	Avertissement envoyé au livreur
Scénario nominal	1-L'admin sélectionné un compte et clique sur « avertir compte » 2-le système envoie un message d'alerte a l'admin pour valider son choix 3-l'admin valide son choix 4-le système envoie l'avertissement
Scenario alternative	2.a.l'admin annuler la tâche . 2.b.Reprend le scenario du point 1.

Tableau 16 : Description textuelle « Avertir compte »

• Description textuelle « Avertir compte »

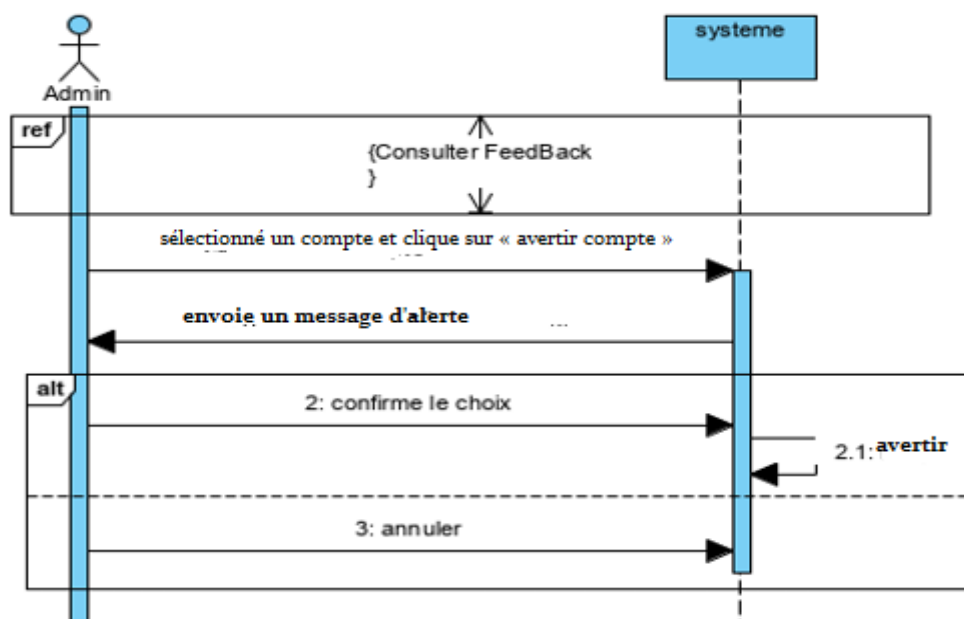


Figure 24 : Description graphique « Avertir compte »

• Description textuelle « Bloquer compte »

Cas d'utilisation	Bloquer compte
Acteur principale	Admin
Acteur secondaire	/
Objectifs	Permet à un admin de retirer un compte de la base de données
Pré condition	L'admin doit être authentifié Compte a été avertir
Post condition	Compte a été bloquer
Scénario nominal	1-L'admin sélectionné un compte et clique sur « bloque compte » 2-le système envoie un message d'alerte a l'admin pour valider son choix 3-l'admin valide son choix 4-le système bloqué et retirer le compte de la base de donnes
Scenario alternative	2.a.l'admin annuler la tâche. 2.b.Reprend le scenario du point 1.

Tableau 17 : Description textuelle « bloquer compte »

• Description graphique « Bloquer compte » :

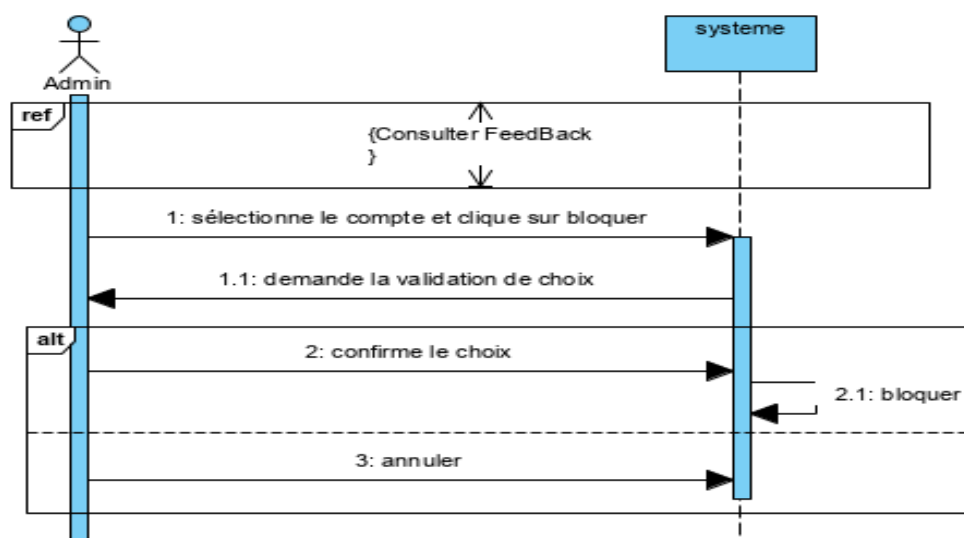


Figure 25 : Description graphique « Bloquer compte »

• **Description textuelle « Consulter activité livreur /payer»**

Cas d'utilisation	Consulter activité livreur/payer
Acteur principale	Admin
Acteur secondaire	/
Objectifs	Permet au admin de voir toutes les activités et payer le livreur
Pré condition	L'admin doit être authentifié
Post condition	Consulter tous les activité
Scénario nominal	1-l'admin demande de consulter activités 2-le system affiche la liste des comptes des livreurs 3-l'admin sélectionne un compte 4-le système affiche l'activité 5-l'admin envoi le paiement 6-le système envoie un message de confirmation 7-l'admin confirmer
Scenario alternative	7.a. l'admin annuler 7.b. Reprend le scenario du point 2

Tableau 18 : Description textuelle « Consulter activité livreur »

• **Description graphique « Consulter activité livreur »**

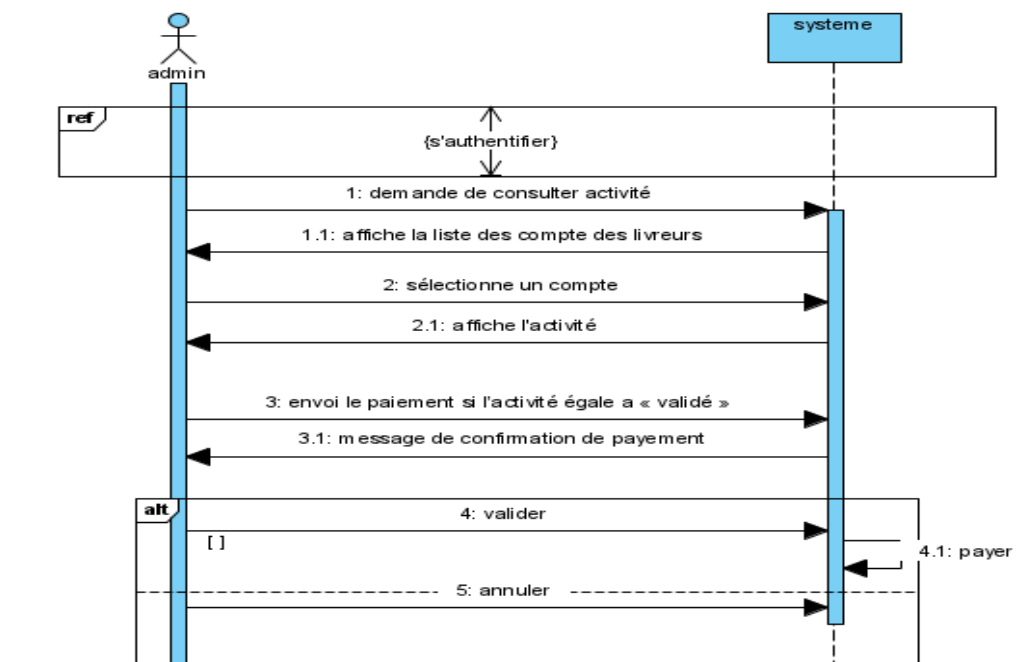


Figure 26: Description graphique « Consulter activité livreur »

1.1.4. Identification des classes candidates :

- Liste des classes candidates :

Le tableau ci-dessous résume la liste des classes candidates de notre système :

La liste des classes candidates
- Utilisateur - Client - Livreur - Admin - Demande - offre

Tableau 19: liste des classes candidates.

- Responsabilités des classes :

Les figures suivantes représentent les responsabilités de chaque classe :

- Classe Utilisateur :

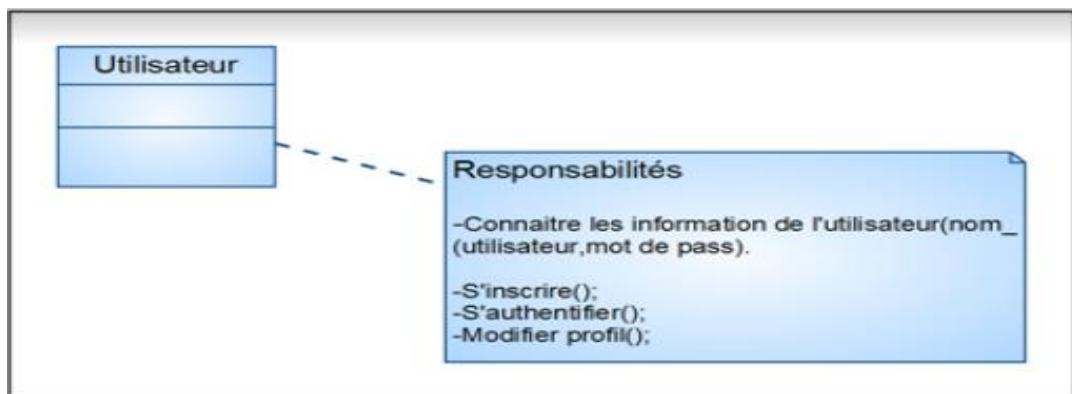


Figure 27: Les responsabilités de la classe candidate<<utilisateur >>

- Classe client :

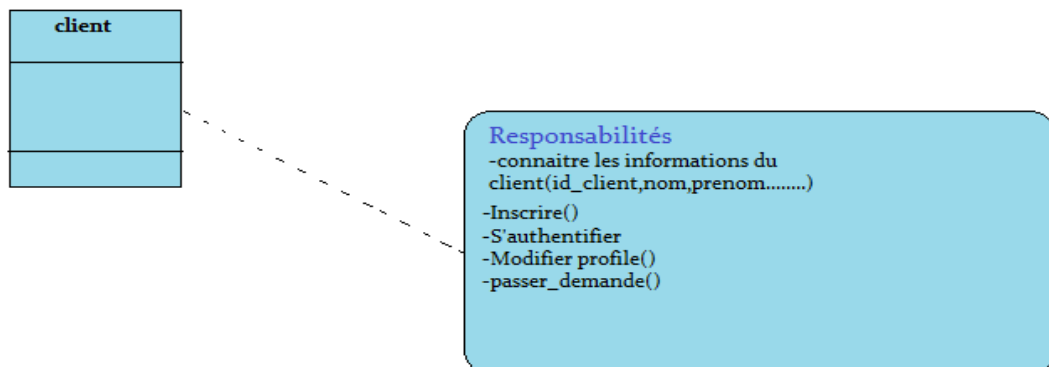


Figure 28 : Les responsabilités de la classe candidate<<client >>

- Classe livreur :

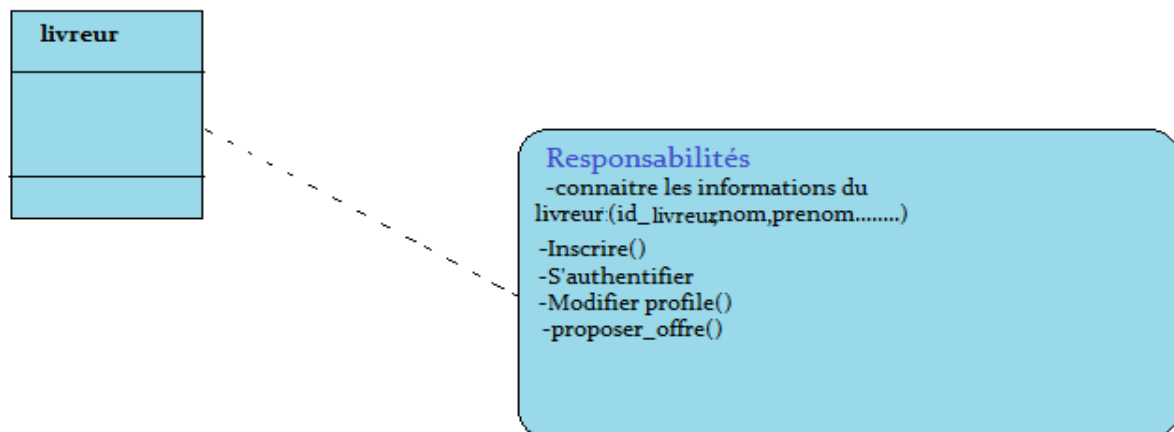


Figure 29: Les responsabilités de la classe candidate<< livreur >>

- Classe offre :

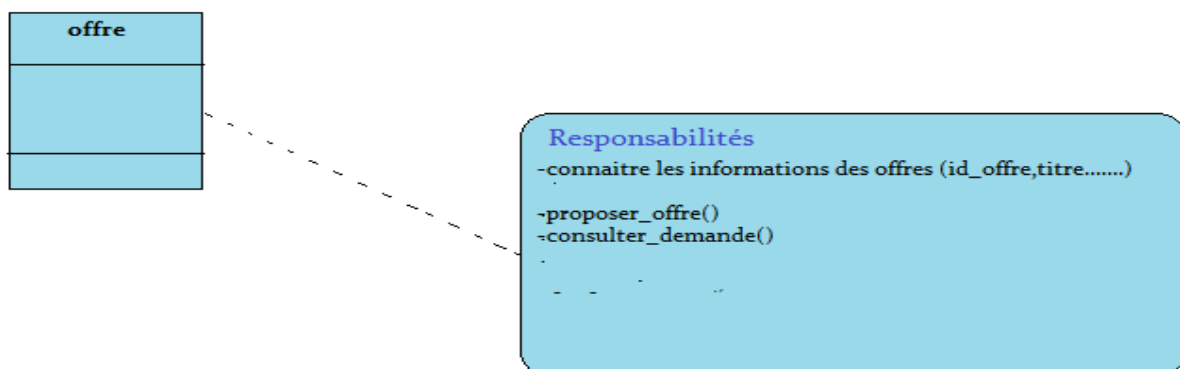


Figure 30 : Les responsabilités de la classe candidate<< offre>>

- Classe demande :

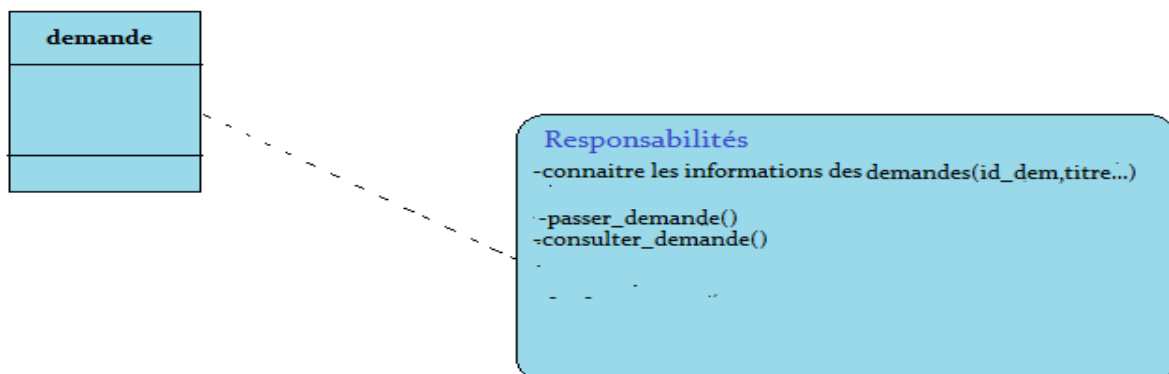


Figure 31 : Les responsabilités de la classe candidate<< demande >>

1.1.5. Diagramme de classe participant :

Le point névralgique de notre démarche s'appelle le diagramme de classes participantes. Il s'agit de diagrammes de classes UML qui décrivent, cas d'utilisation par cas d'utilisation, les trois principales classes d'analyse et leurs relations

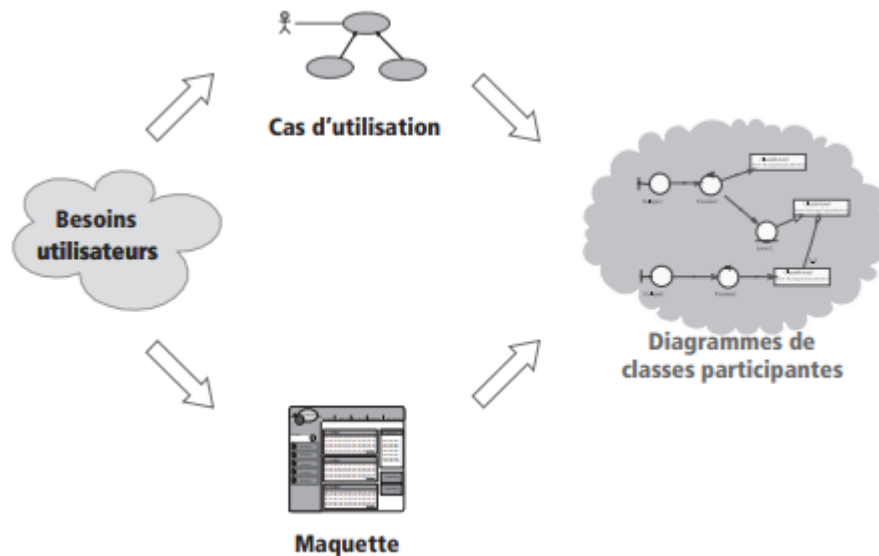


Figure 32 : Les diagrammes de classes participantes dans la démarche

Les diagrammes de classes participantes sont donc particulièrement importants car ils font la jonction entre les cas d'utilisation, la maquette et les diagrammes de conception logicielle. pour compléter ce travail d'identification, nous allons ajouter des attributs et des opérations dans les classes d'analyse, ainsi que des associations entre elles. (3)

- **Modifier profile :**

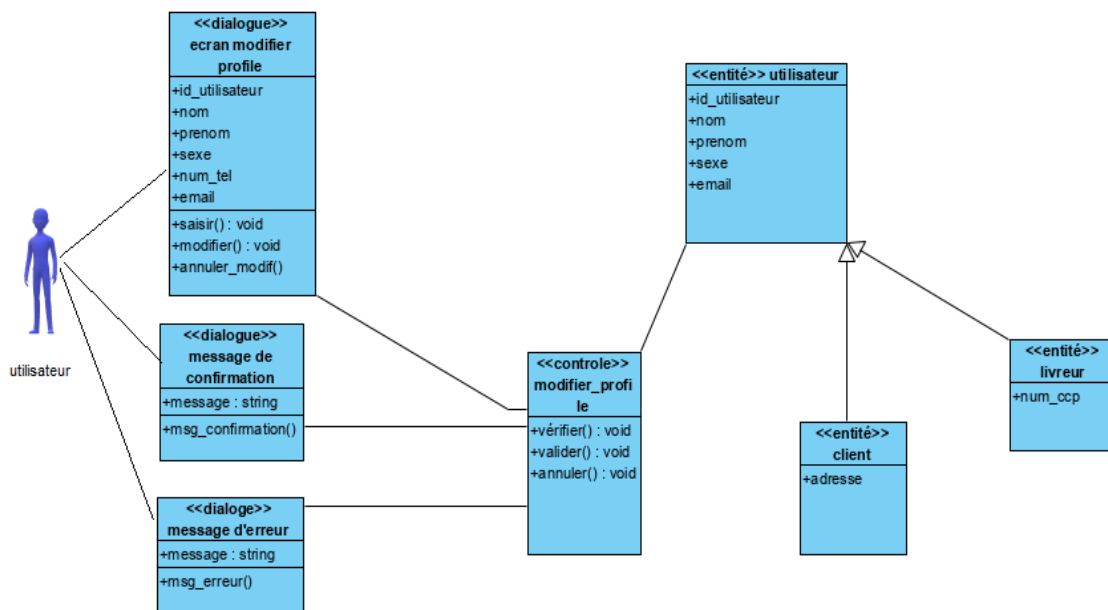


Figure 33: diagramme de classe participant «Modifier profile »

- **S'authentifier :**

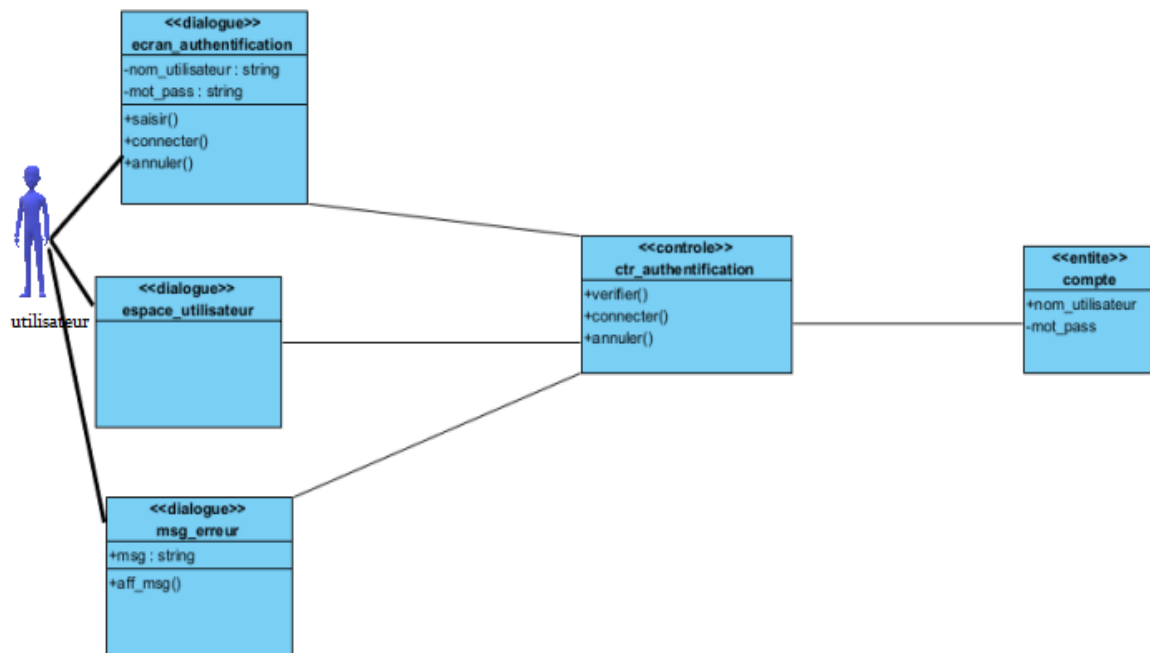


Figure 34 : diagramme de classe participantes « S'authentifier »

- **Passer demande livraison :**

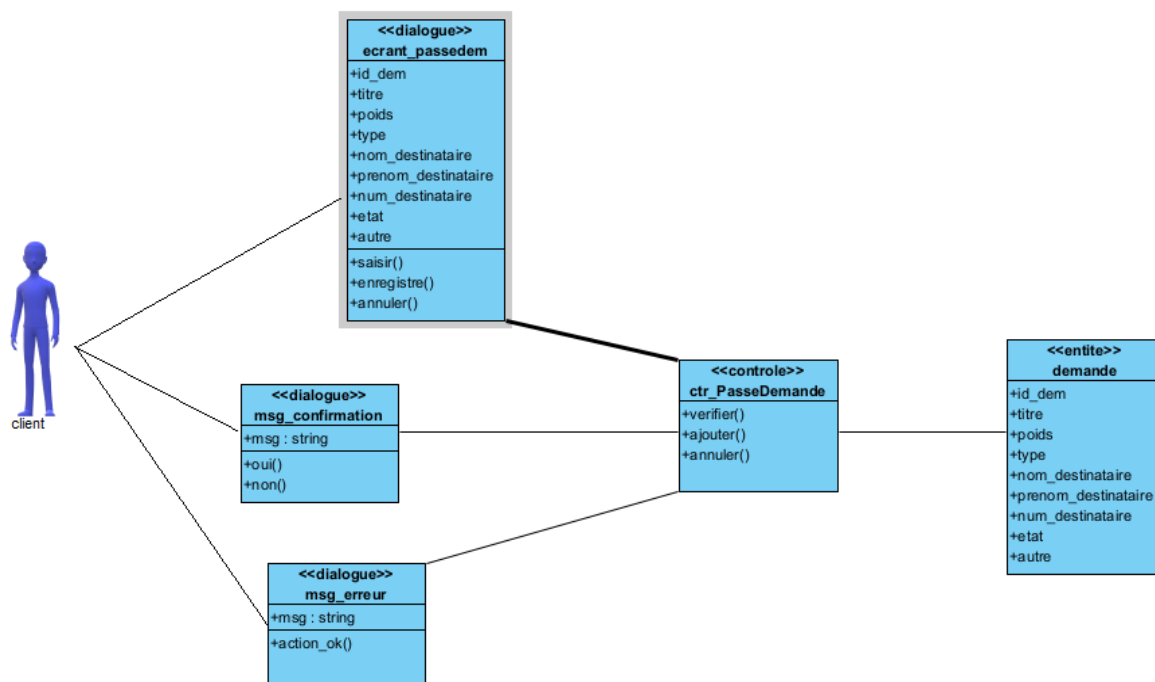


Figure 35 : diagramme de classe participantes « Passer demande livraison »

- Consulter les offres de livraison :

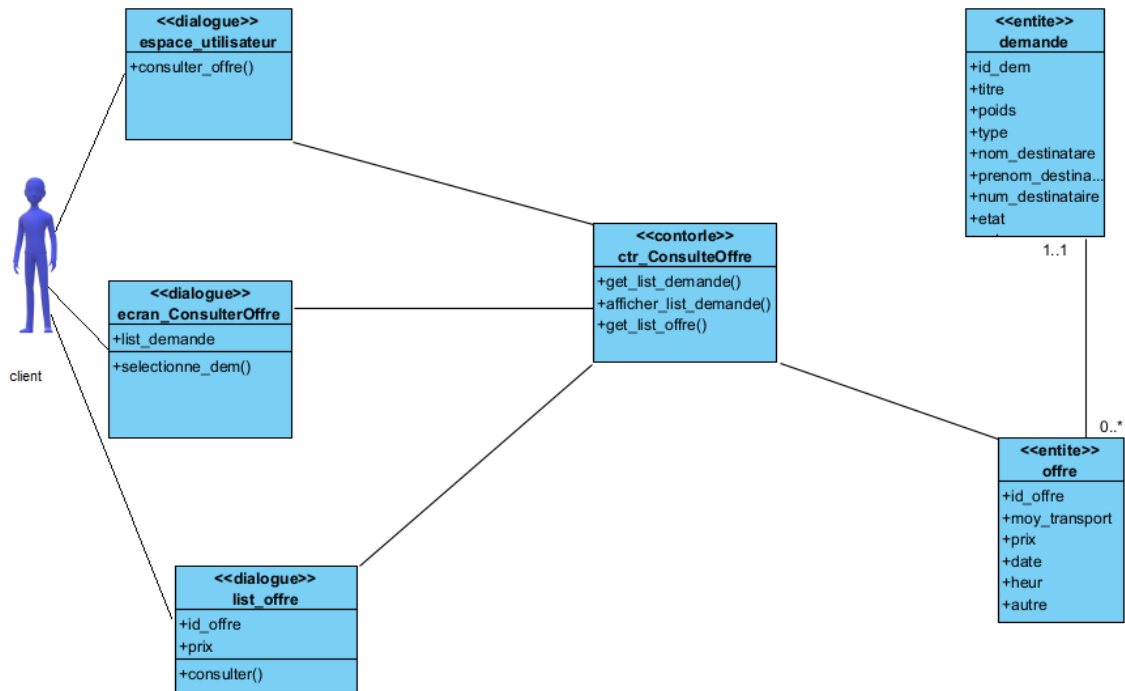


Figure 36: diagramme de classe participantes « Consulter les offres de livraison »

- Consulter l'état de livraison d'un colis :



Figure 37 : diagramme de classe participantes« Consulter l'état de livraison d'un colis »

- Valider la livraison du colis :

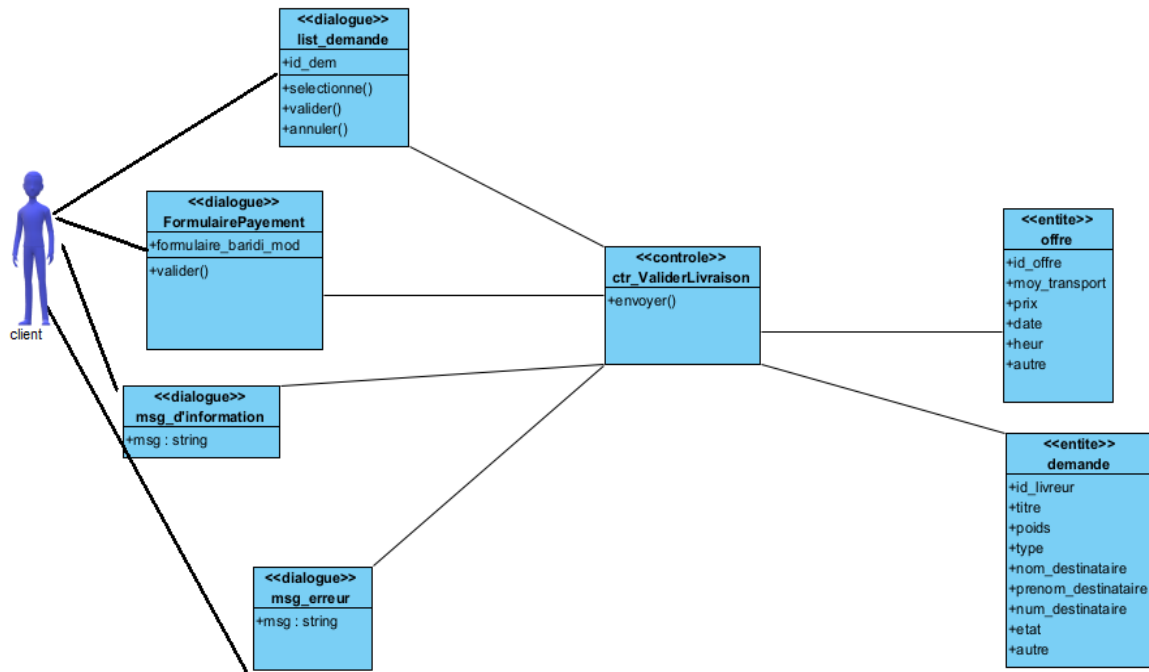


Figure 38: diagramme de classe participantes « Valider la livraison du colis »

- Proposer une offre de livraison :

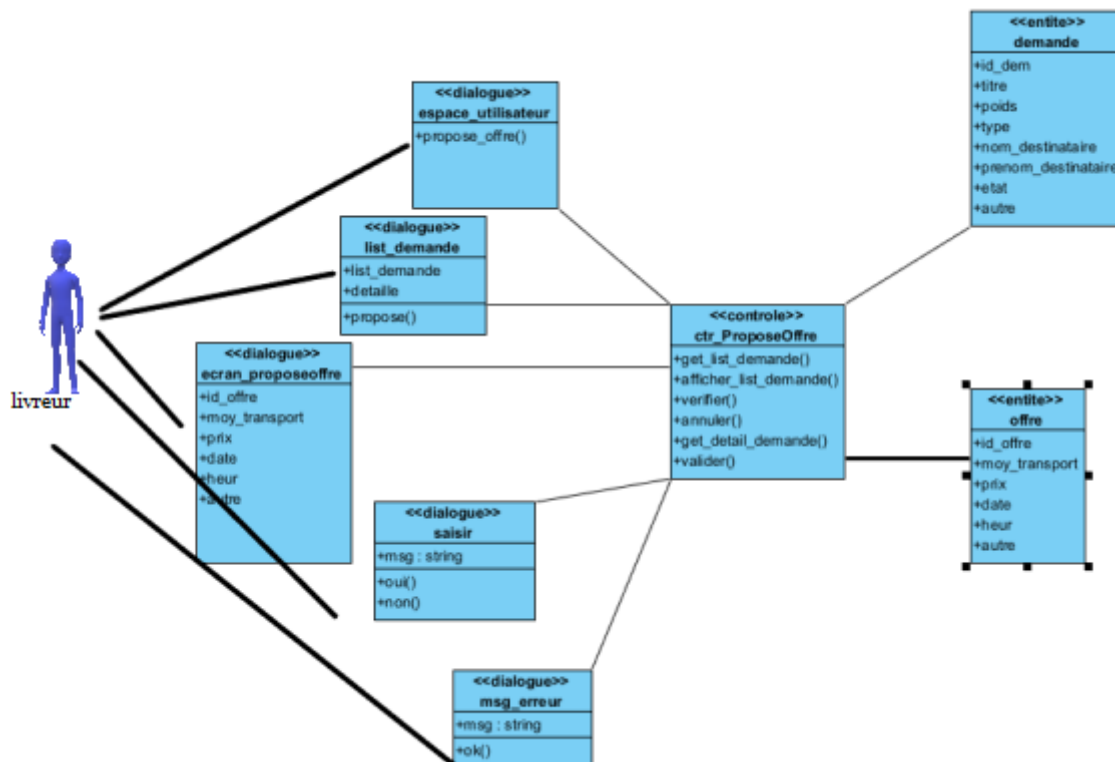


Figure 39: diagramme de classe participantes « Proposer une offre de livraison »

• Valider réponse client :

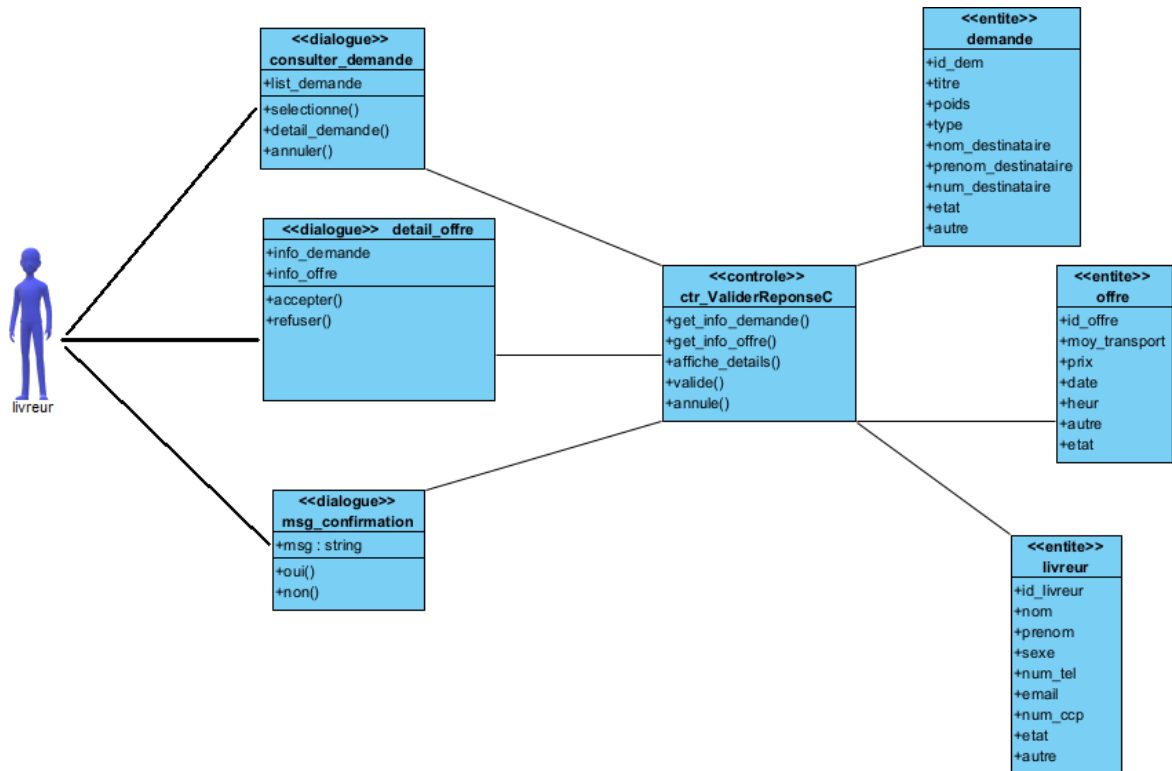


Figure 40 : diagramme de classe participantes « Valider réponse client »

• Mettre à jour l'état de livraison :

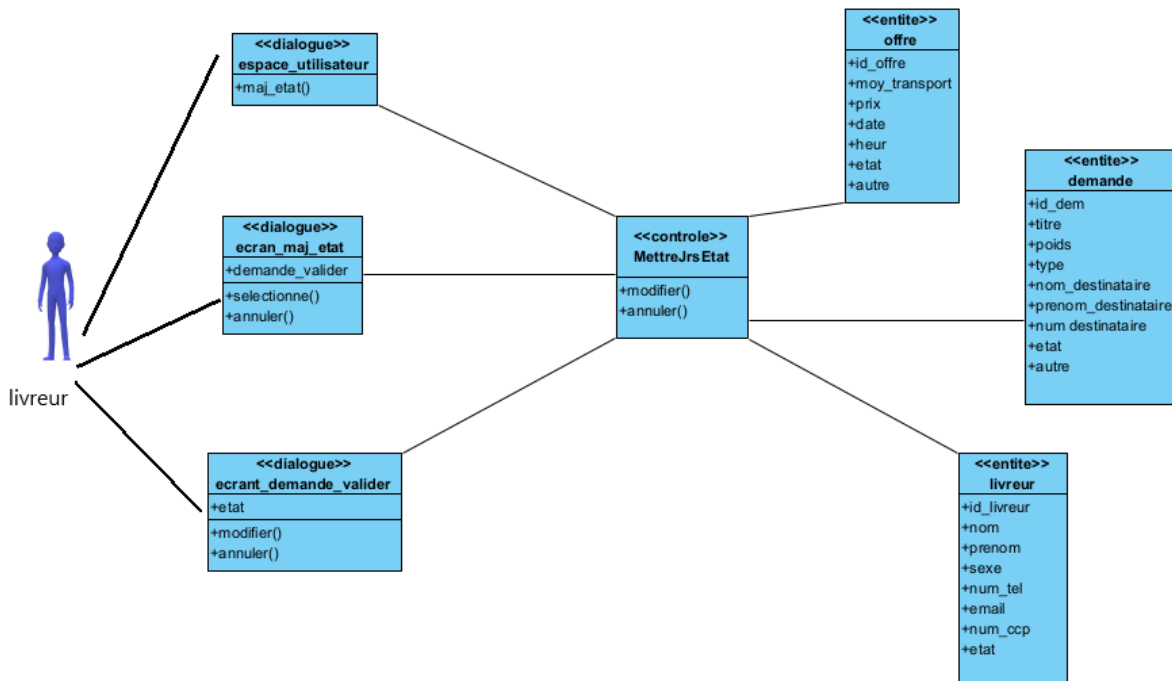


Figure 41: diagramme de classe participante « Mettre à jour l'état de livraison »

• Cas d'utilisation « Avertir compte »

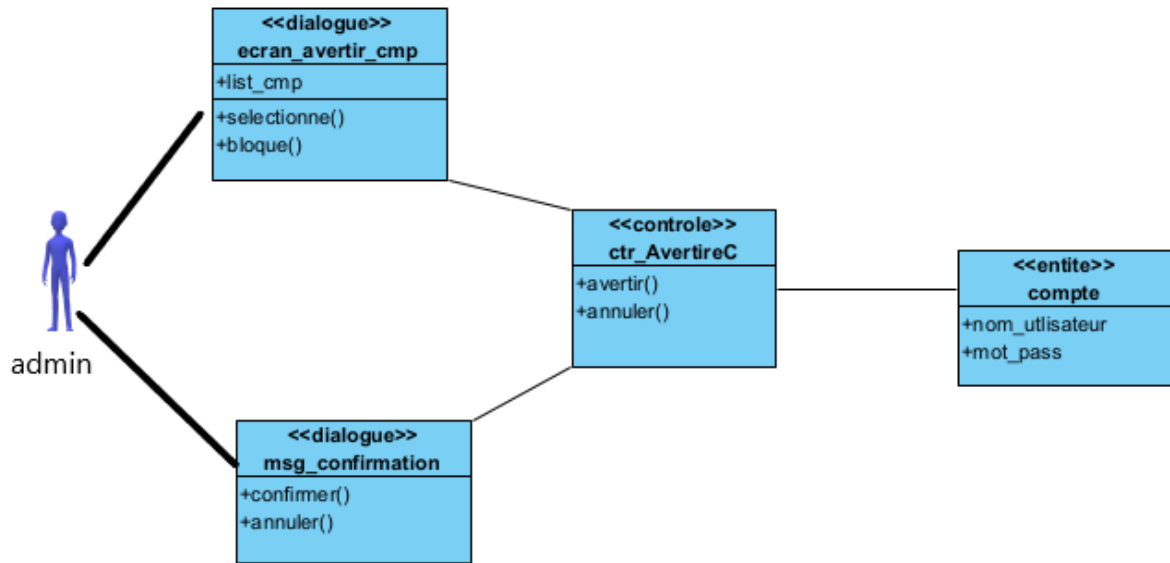


Figure 42 : diagramme de classe participantes « Avertir compte »

2. Capture des besoins technique :

La capture des besoins techniques recense toutes les contraintes et les choix dimensionnant la conception du système.

Les outils et les matériels sélectionnés ainsi que la prise en charge des contraintes d'intégration conditionnant généralement des prés requis d'architecture générale.

La capture des besoins techniques se présente comme suit :

- Capture des spécifications logicielles.
- Capture des spécifications liées à la configuration matérielle. (4)

2.1. La spécification technique du point de vue matériel :

Les prérequis techniques ont été exprimés dans l'étude préliminaire, lors de l'expression des besoins opérationnels et de celle des choix stratégiques de développement. Ces choix impliquent des contraintes relatives à la configuration du réseau matériel. Elles sont de nature géographique, organisationnelle, et technique. Elles concernent les performances d'accès aux données, la sécurité du système, la volumétrie.

- Style d'architecture en niveaux :

Le style d'architecture en niveaux spécifie le nombre de niveaux géographiques et organisationnels où vont se situer les environnements d'exécution du système.

Dans notre système La configuration géographique impose le développement d'une solution client/serveur à deux niveaux.

- Spécification d'architecture 2 niveaux :

Une architecture à deux niveaux aussi appelée client-serveur, est une architecture où un Poste client exécute une application se trouvant sur un serveur. Cela effectué comme un dialogue entre le client et le serveur dans lequel :

- Client : demande une ressource.
- Serveur : la lui fournit directement.

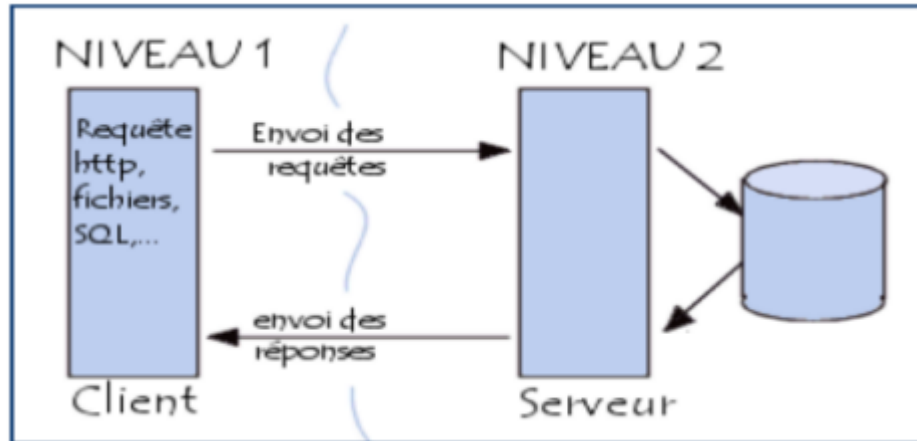


Figure 43 : Architecture 2 niveaux

2.2. Capture des spécifications logicielles :

Une fois les spécifications matérielles déterminées, on passe aux spécifications non fonctionnelles autrement dit techniques ou logicielles. C'est des fonctionnalités techniques que le système va assurer à l'utilisateur indépendamment du terme métier ou fonctionnel. On va donc énumérer une liste de fonctionnalités techniques que notre système va pouvoir assurer et offrir aux utilisateurs, c'est pourquoi nous avons introduit les concepts d'exploitant et de cas d'utilisation techniques :

2.2.1. Exploitant :

Appelés aussi « acteurs techniques » du système, ce sont les acteurs qui bénéficient des fonctionnalités techniques du système.

- **Identification des exploitants du système :**

Les exploitants de notre système sont la majorité des acteurs de la branche fonctionnelle sont donc utilisateurs dans la dimension technique. Quel que soit : client, livreur ou admin.

2.2.2. Cas d'utilisation technique :

Un cas d'utilisation technique est une séquence d'actions qui concernent des fonctionnalités purement techniques.

- **Identification des cas d'utilisation technique :**

Les principaux cas d'utilisation techniques de notre système sont illustrés par la figure suivante :

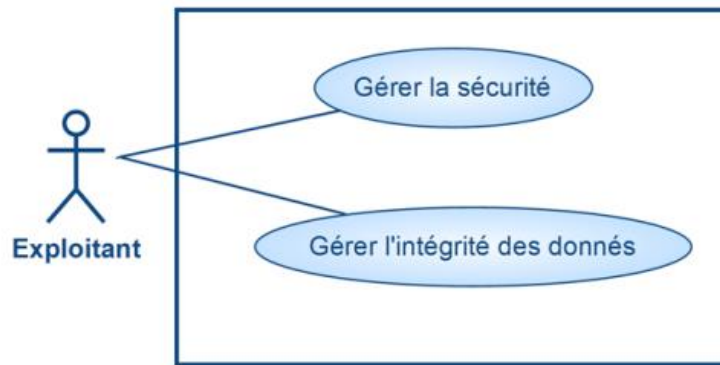


Figure 44 : Modèle de spécification logicielle du système.

- Description des cas d'utilisation techniques :
1- « Gérer la sécurité »

Créer compte :

Cas d'utilisation	Créer compte
Acteur principale	Utilisateur
Acteur secondaire	/
Objectifs	Ce cas permet à l'utilisateur de créer un compte
Pré condition	/
Post condition	Un compte sera créé
Scénario nominal	1. l'utilisateur demande les formulaires d'inscription 2. Le système affiche les formulaires. 3. l'utilisateur remplit les formulaires. 4. l'utilisateur enregistre et valide ses informations. 5. Le système affiche un message de confirmation.
Scénario d'erreur	3.a l'utilisateur annuler le processus 3.b Reprend le scenario du point 2 5.a. ce compte est déjà existant. Le système affiche le message d'erreur. 5.b. Reprend le scenario du point 3

Tableau 20: Description textuelle « Créer compte »

Consulter profile :

Cas d'utilisation	Consulter profile
Acteur principale	Utilisateur
Acteur secondaire	/
Objectifs	Affiche le profile
Pré condition	L'utilisateur doit être s'authentifier
Post condition	Le profile est affichée
Scénario nominal	1- l'utilisateur clique sur le bouton « Mon profile » 2-le system affiche la page du profile
Scenario d'erreur	/

Tableau 21: Description textuelle « consulter »

Modifier profile :

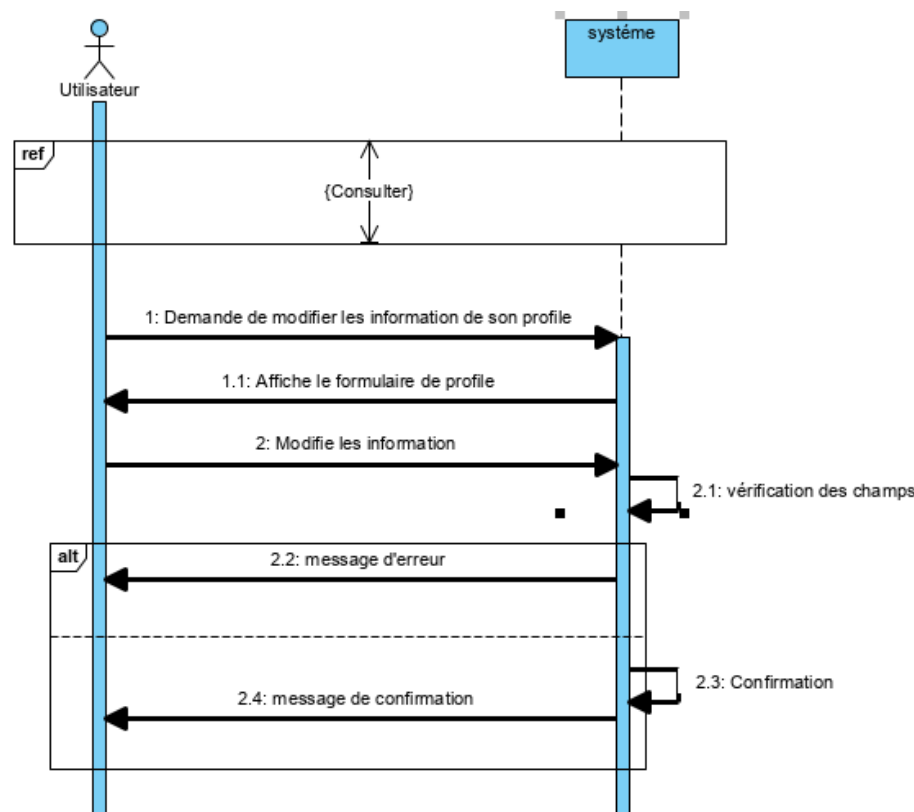


Figure 45 : Description textuelle « modifier profile »

S'authentifier :

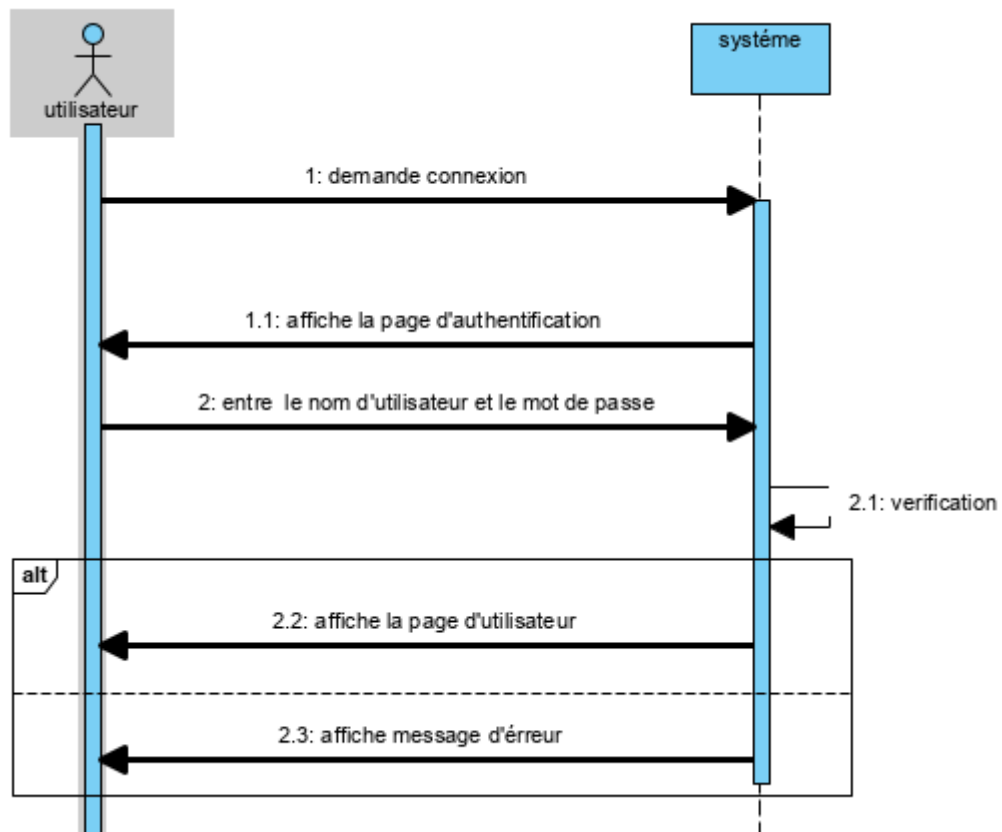


Figure 46: Description graphique « S'authentifier »

2- « Gérer l'intégrité des données »

Cas d'utilisation	Gérer l'intégrité des données.
Acteur	L'exploitant.
But	Vérifier l'intégrité des données.
Pré condition	L'exploitant s'authentifie.
Scenario nominale	1-L'exploitant saisit les données. 2-Le système contrôle la validité de chaque champ.
Scénario alternatif	• Le système affiche une notification « champs non valide ». Le scénario reprend au point 1.

Tableau 22: Description textuelle du cas d'utilisation «Gérer l'intégrité des données»

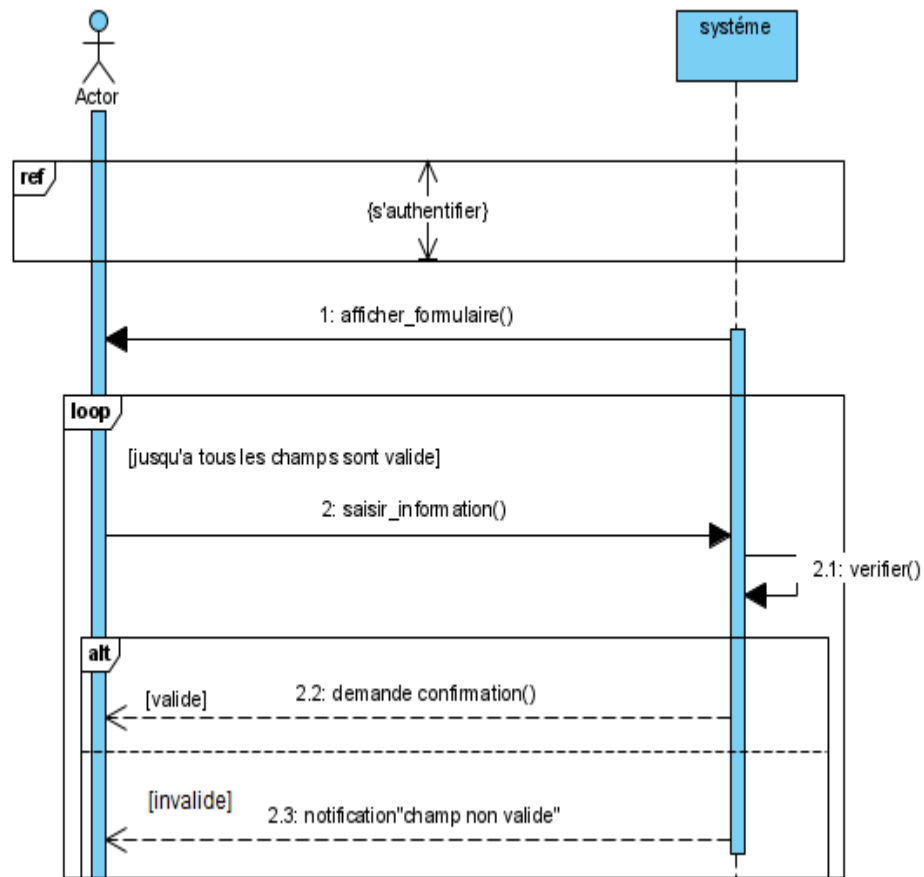


Figure 47 : Diagramme de séquence de cas d'utilisation « Gérer l'intégrité des données »

Conclusion :

La phase de capture des besoins est une étape qui œuvre d'un côté pour compléter les recueils des besoins opérés pendant la phase d'étude préliminaire, et d'un autre côté, elle donne une première vue pour la prochaine étape. Cette dernière qu'elle sera le noyau du prochain chapitre.

Chapitre 04 :

Analyse et Conception

1.dictionnaire des données :

Classe	Code	Désignation	Type	Taille
Compte	-email -motPass	-email -mot de passe	- Varchar - Varchar	40 10
Admin	-num_ccp	-Numéro CCP	- INT	20
Client	-id_client -nom -prenom -sexe -N_tel -adress	-Id Client -Nom -Prénom -sexe -numéro Téléphone -adresse	- INT - Varchar - Varchar - Varchar - INT - Varchar	40 40 40 10 40
livreur	-id_livreur -nom -prenom -sexe -N_tel -etat -num_ccp	-Id Livreur -Nom -Prénom -sexe -numéro Téléphone -etat -Numéro ccp	- INT - Varchar - Varchar - Varchar - INT - Varchar - Varchar	-20 -20 -20 -20 -10 -20 -20
demande	-id_dem -poids -Type -nom_destinataire -prenom_destinataire -num_destinataire -etat -date	- Id demande -Poids -Type -Nom destinataire -Prénom destinataire -Numérodestinataire -état -date	- INT -float - Varchar - Varchar - Varchar - INT - VARCHAR - DATE	-10 -10 -20 -20 -20 -20 -10 -20
offre	-Id_offre -moy_transport -prix -date -etat	-Id offre -moyen transport -prix -date -état	- INT - Varchar - INT - DATE - Varchar	-10 -20 -10 -20 -20
feedback	-rating	-Rating	- Varchar	-20

Tableau 18 : dictionnaire des donnes

2. diagramme de classe générale :

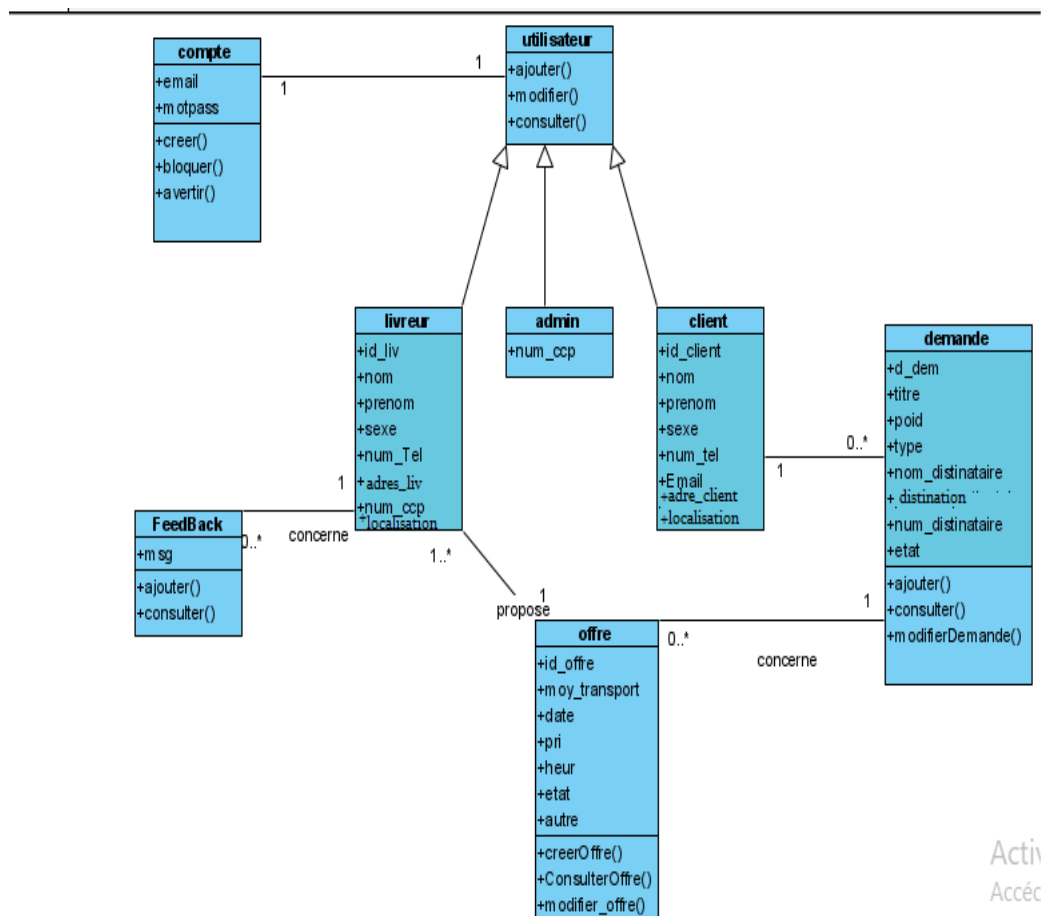


Figure 48: diagramme de classe générale

3. Le model relationnel :

- Compte (Email,mot_pass) ;
- Client (id_client, nom, prénom, sexe, num_tel, email, adress, #Email, #mot_pass);
- Livreur (id_livreur, nom, prénom, sexe, num_tel, email,etat, num_ccp, #Email, #mot_pass) ;
- Demande (id_dem, poids,type, nom_destinataire, prenom_destinataire, num_destinataire, etat, date, #id_client)
- Offre (id_offre, moy_transport, prix, date, etat,#id_livreur)
- Admin(num_ccp,#email,#mot_pass)
- Feedback (rating,#id_livreur)

4. Diagramme de séquence :

Un diagramme de séquence montre les différentes parties d'un système qui travaillent dans une "séquence" pour obtenir quelque chose.

4.1. Les diagrammes de séquences :

- Modifier profile

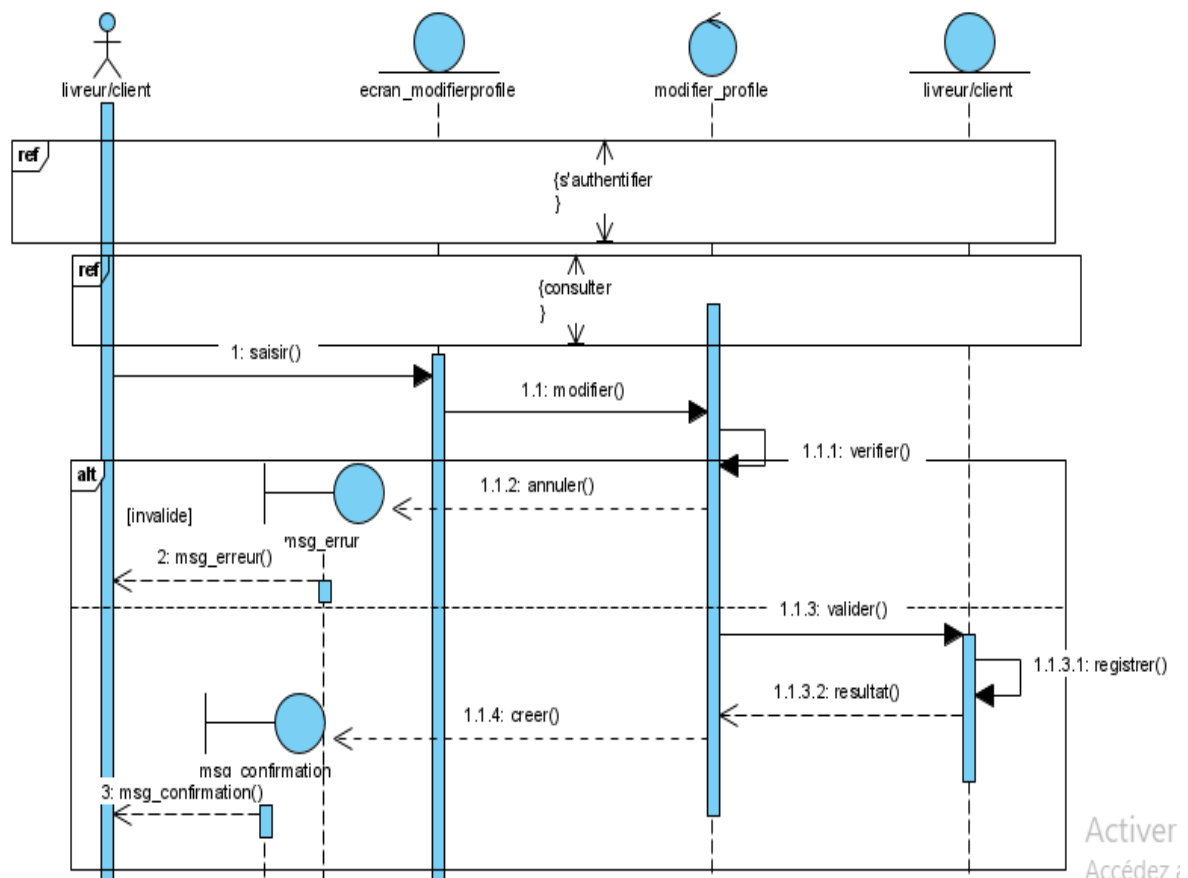


Figure 49 : diagrammes de séquences << modifier profile>>

• **S'authentifier**

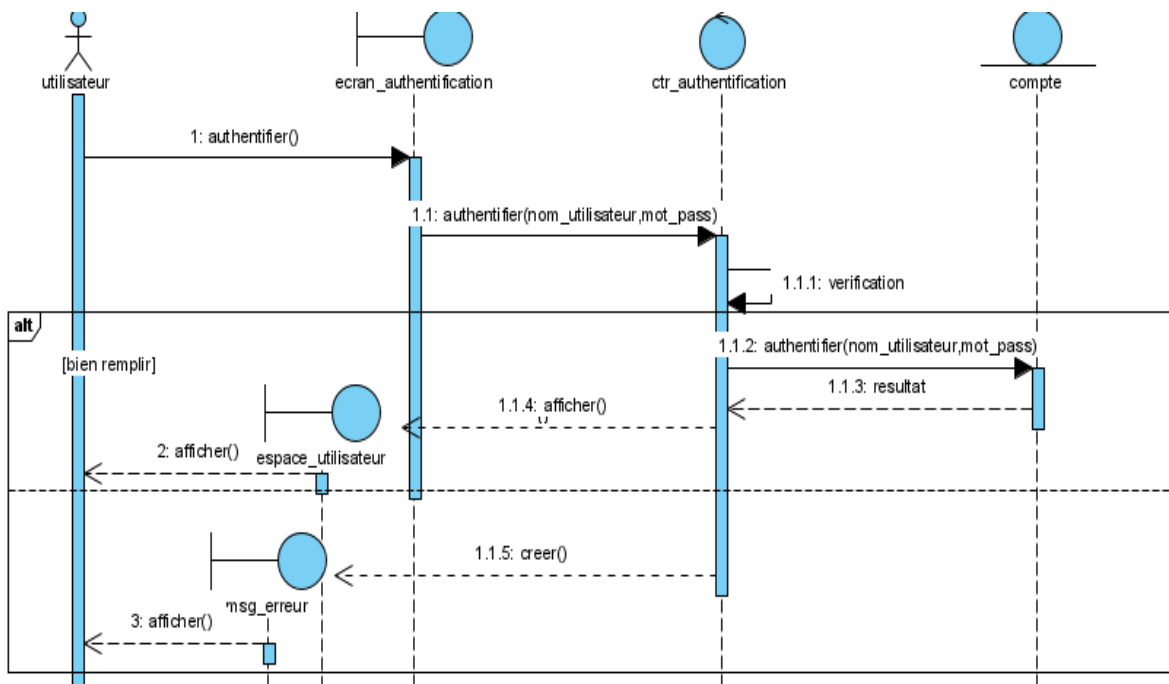


Figure 50 : diagrammes de séquences << S'authentifier>>

• Passer demande de livraison

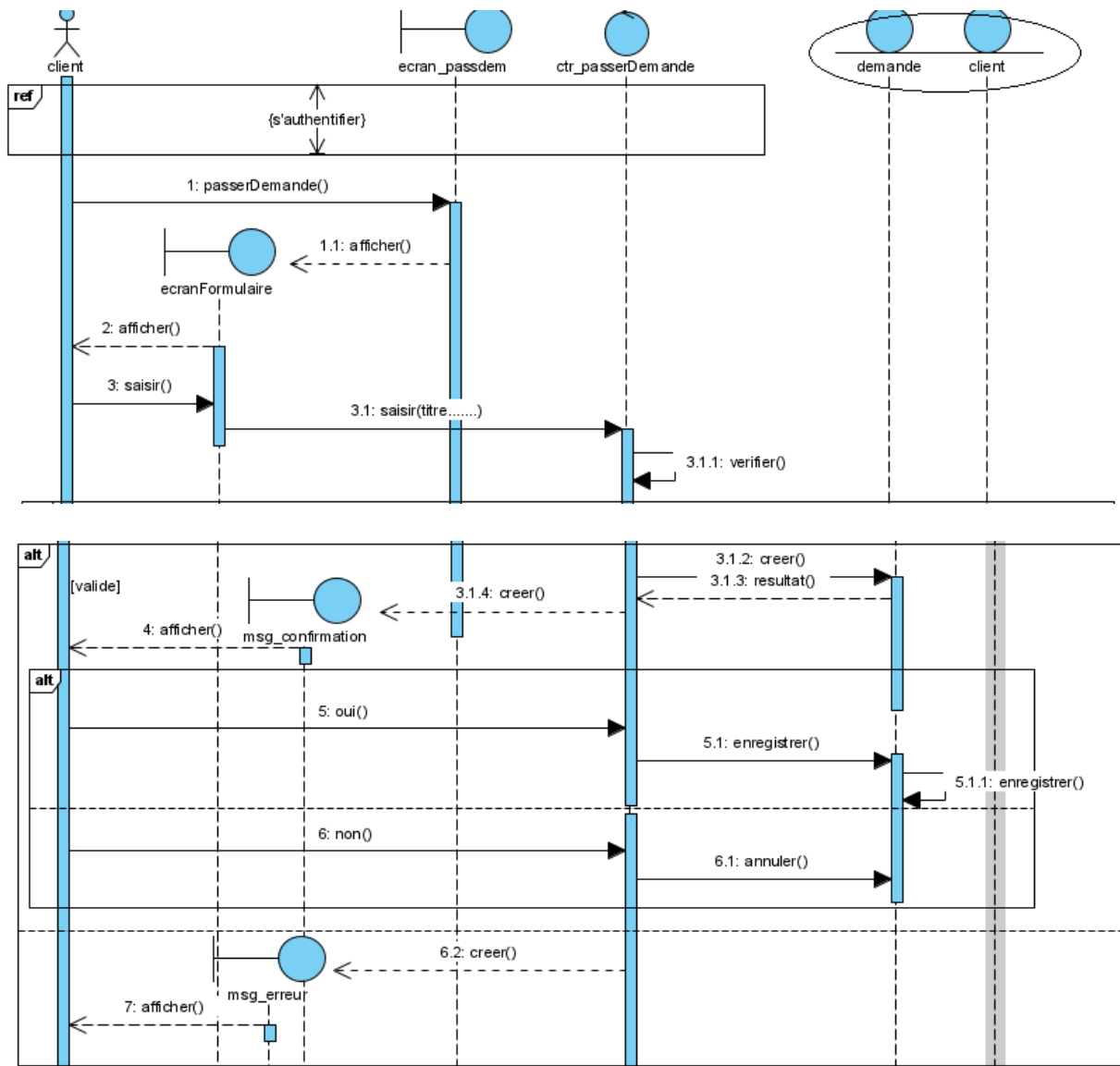


Figure 51 : diagrammes de séquences << passer demande de livraison>>

- Consulter les offres de livraison

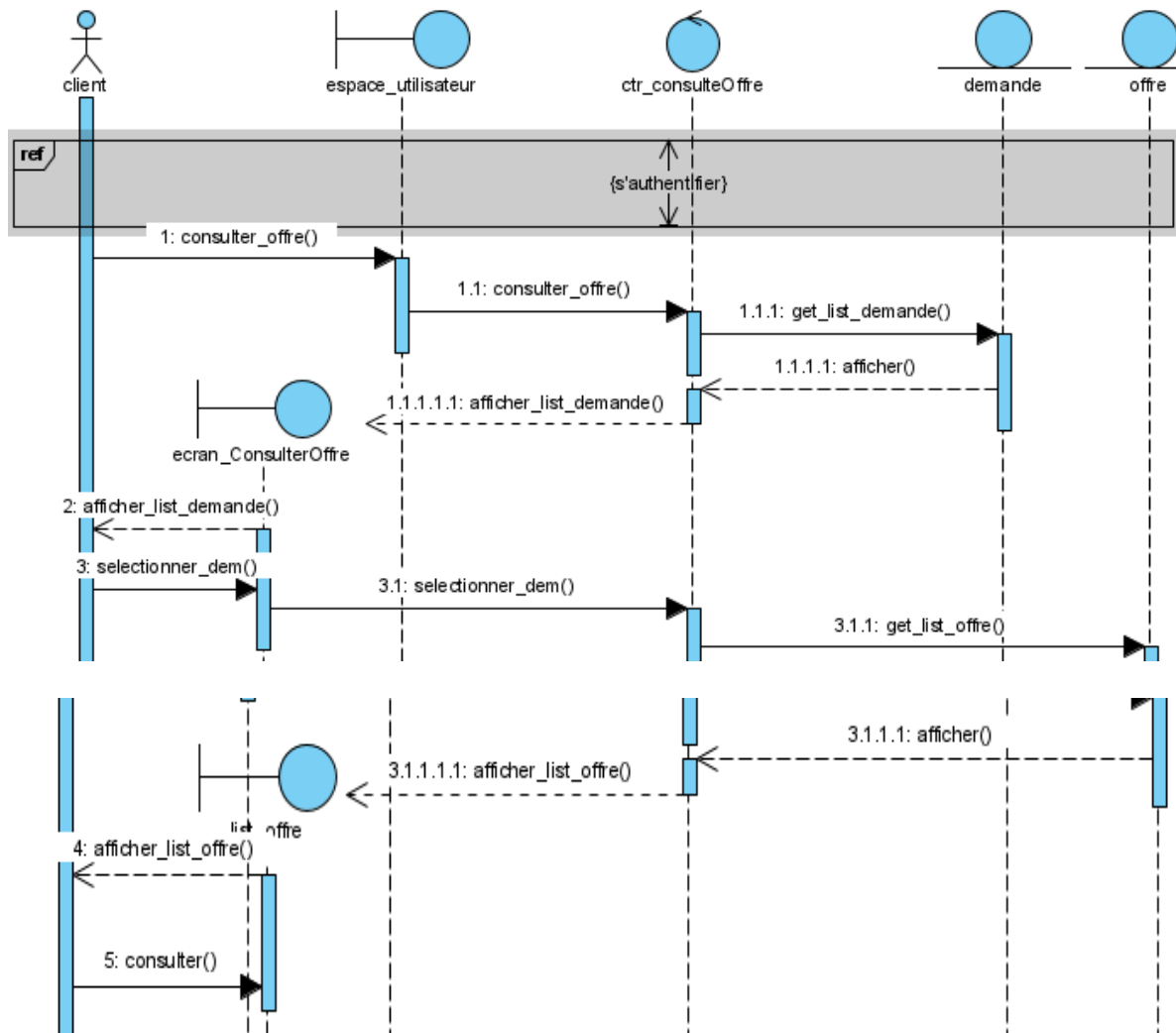


Figure 52: diagrammes de séquences << Consulter les offres de livraison>>

- Consulter l'état de livraison d'un colis

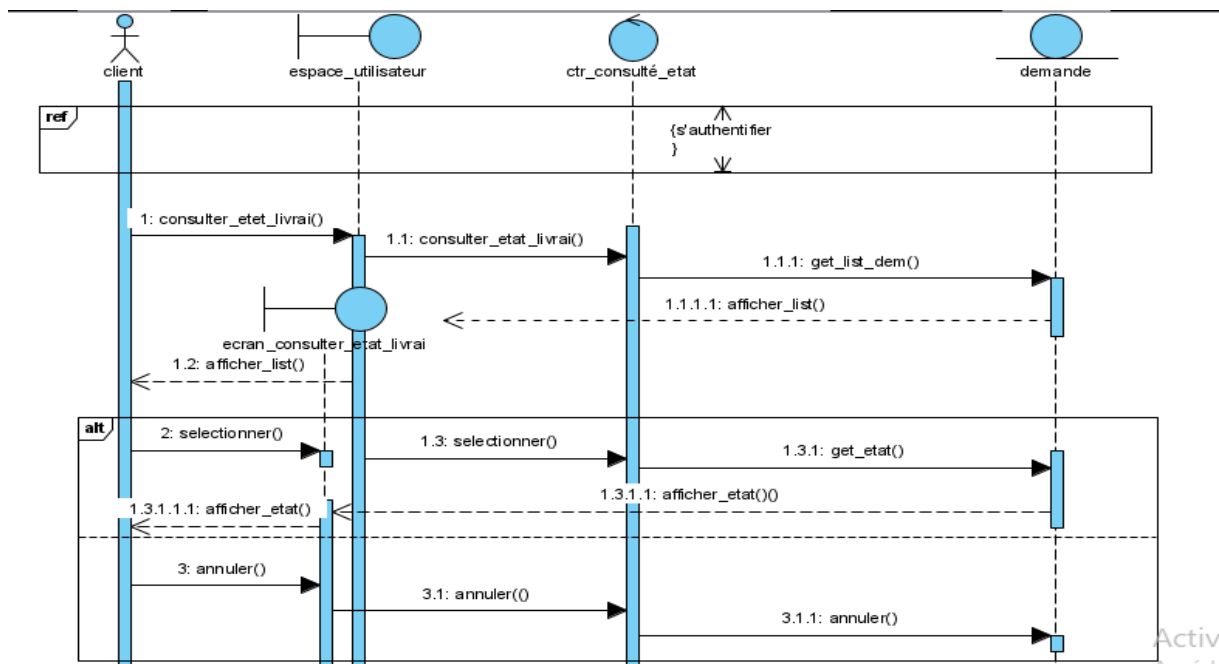


Figure 53 : diagrammes de séquences << Consulter l'état de livraison d'un colis>>

- Valider la livraison du colis

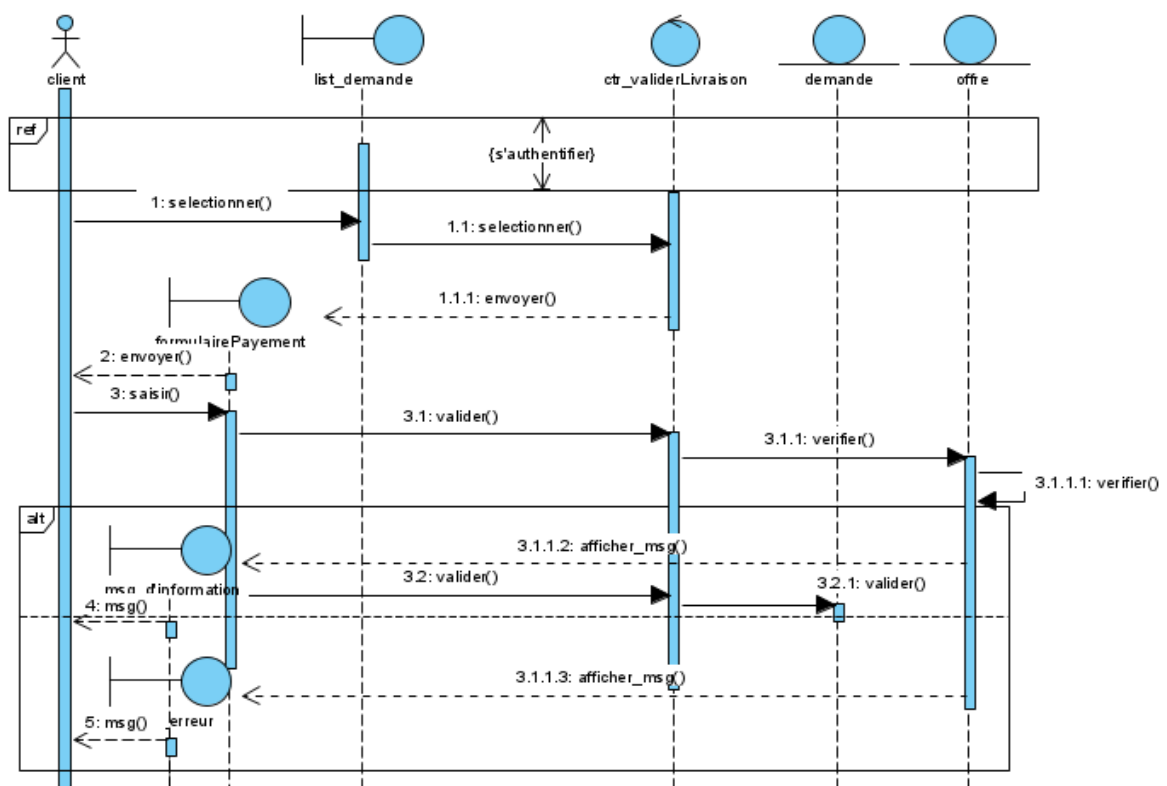


Figure 54: diagrammes de séquences << valider la livraison du colis>>

• Valider réponse client

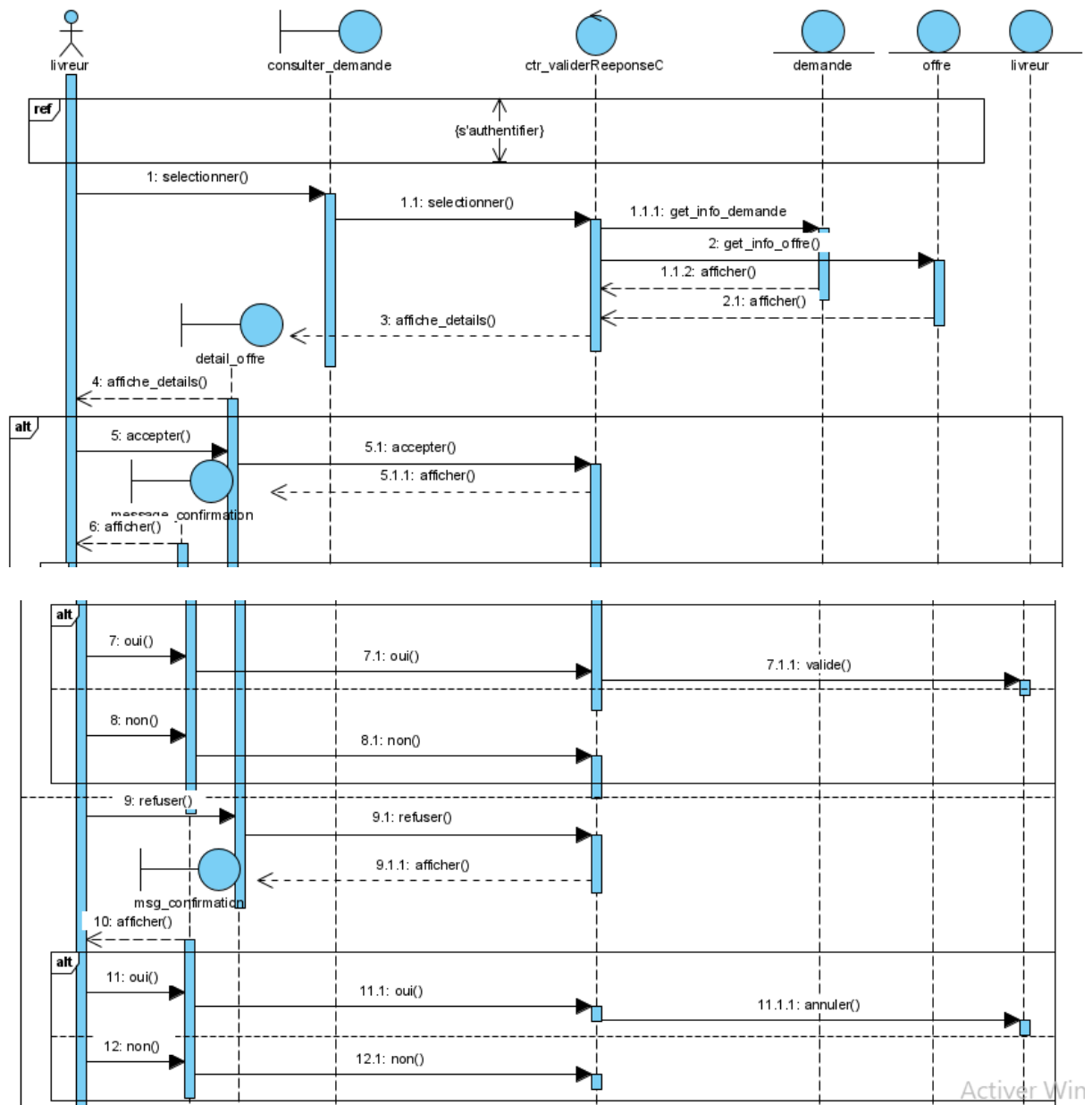


Figure55: diagrammes de séquences <<valider réponse client>>

• Proposer une offre de livraison

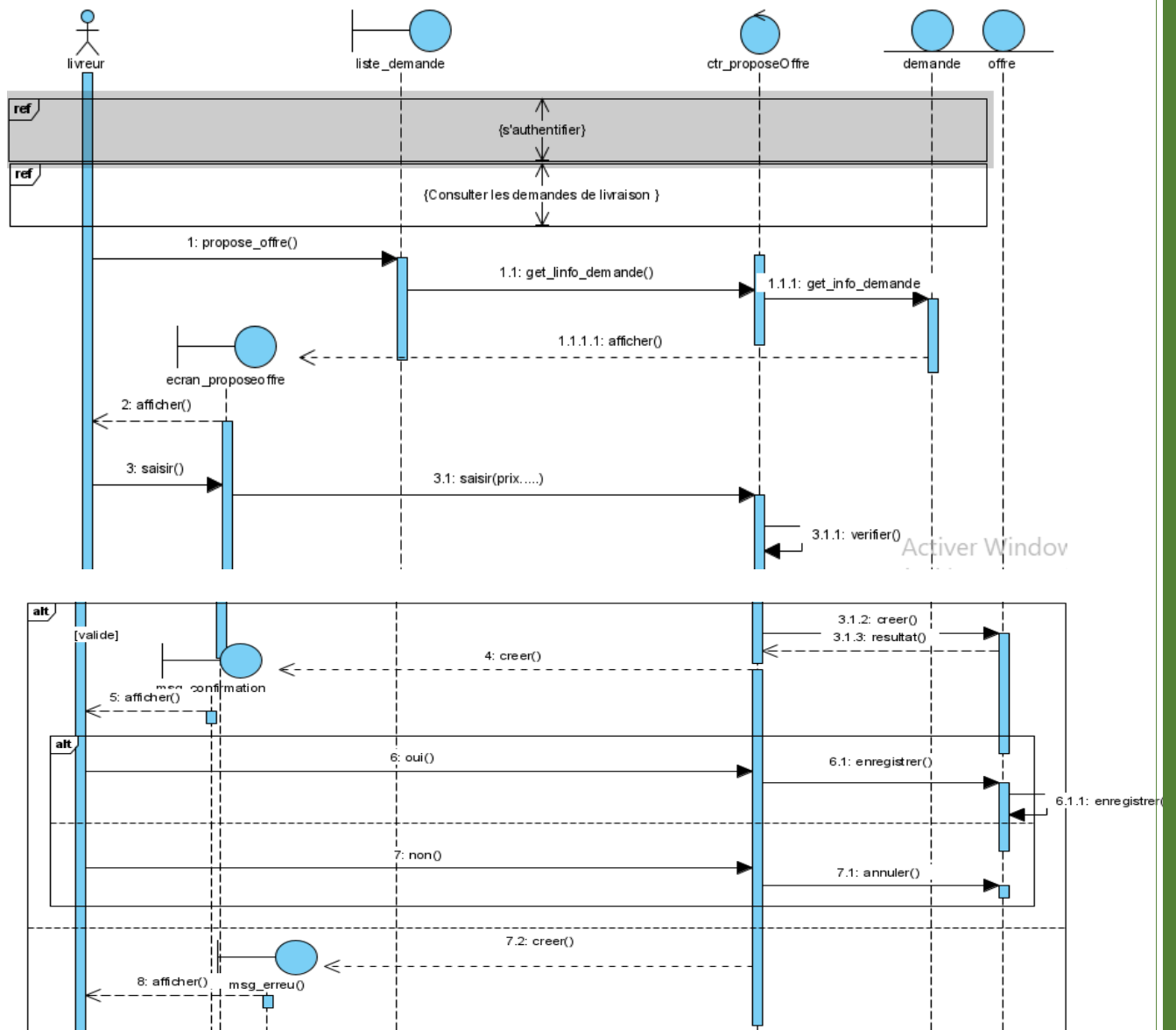


Figure 56: diagrammes de séquences << proposer une offre de livraison>>

- Mettre à jour l'état de livraison

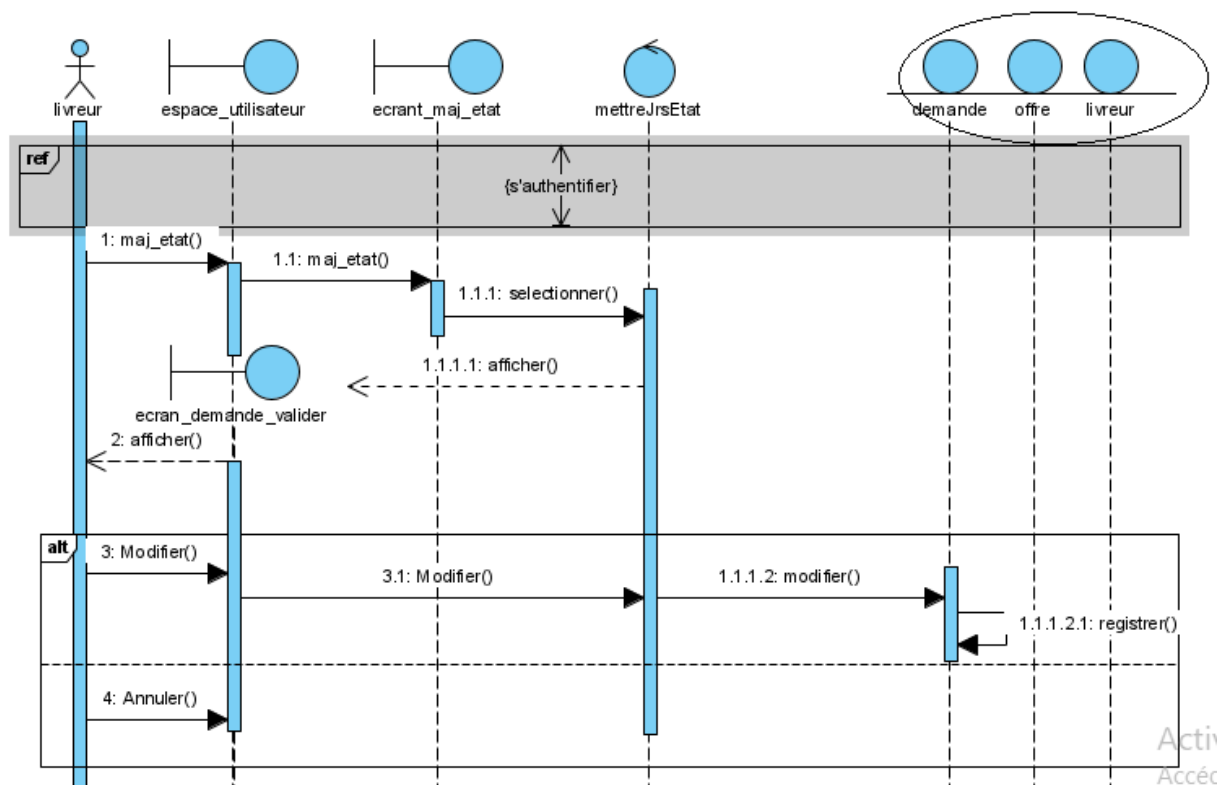


Figure 57: diagrammes de séquences << Mettre à jour l'état de livraison>>

- Avertir compte

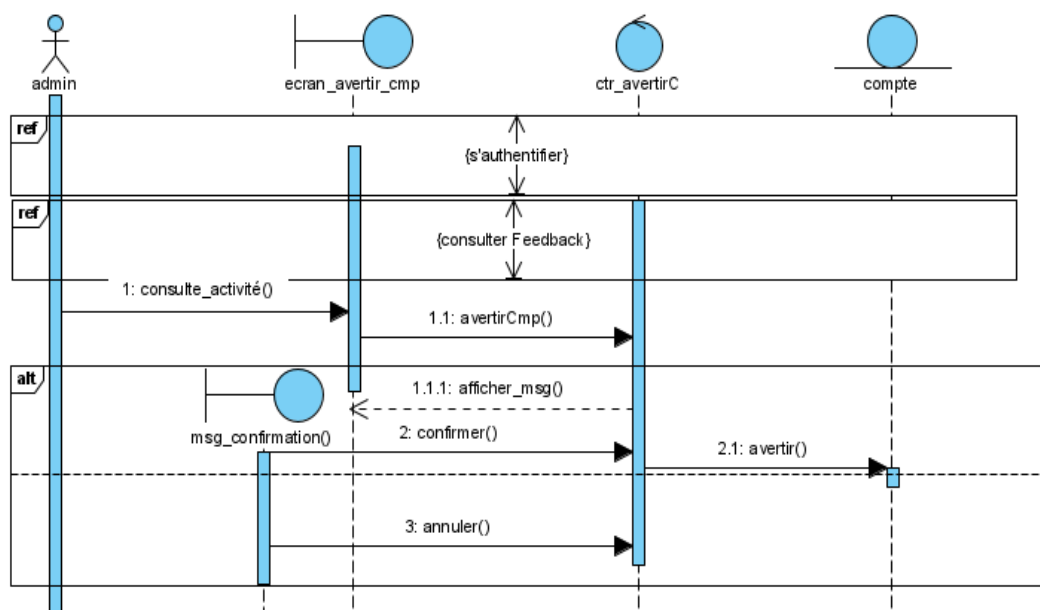


Figure 58: diagrammes de séquences << Avertir compte >>

Conclusion :

La phase de conception prend en compte les choix d'architecture technique retenue pour le développement et l'exploitation du système. Elle vise à trouver des solutions informatiques et techniques pour mettre en œuvre et construire le système analysé au cours des phases précédentes.

Chapitre 05 : implémentation

Introduction :

Cette partie concerne l'implémentation du projet où on convertit les diagrammes UML à un code source exécutable. Nous allons présenter l'environnements de développement de l'application mobile Android Studio, les langages de programmation Java, SQL et PHP. Ensuite, nous allons présenter quelques fenêtres de notre application mobile.

I. Présentation des outils de développement :

1. Le Langage de programmation java :

Java est un langage de programmation et une plate-forme informatique qui ont été créés par Sun Microsystems en 1995. Beaucoup d'applications et de sites Web ne fonctionnent pas si Java n'est pas installé et leur nombre ne cesse de croître chaque jour. Java est rapide, sécurisé et fiable. Des ordinateurs portables aux centres de données, des consoles de jeux aux superordinateurs scientifiques, des téléphones portables à Internet, la technologie Java est présente sur tous les fronts.(1)



Figure 59: logo java.

2. Le langage xml :

Le XML, pour Extensible Markup Language, désigne un langage informatique (ou métalangage pour être plus précis) utilisé, entre autres, dans la conception des sites Web et pour faciliter les échanges d'informations sur Internet. Ce langage de description a pour mission de formaliser des données textuelles. Il s'agit, en quelque sorte, d'une version améliorée du langage HTML avec la création illimitée de nouvelles balises.

Comme le langage HTML, le XML permet la mise en forme de documents via l'utilisation de balises. Développé et standardisé par le World Wide Web Consortium à la fin des années 1990, il répondait à l'objectif de définition d'un langage simple, facile à mettre en application.

Le XML se classe dans la catégorie des langages de description (il n'est ni un langage de programmation, ni un langage de requêtes). Il est donc naturellement utilisé pour décrire des données en s'appuyant sur des balises et des règles personnalisables. (2)

3.Le SDK (Software Development Kit):

Un SDK, pour Software Development Kit, désigne un ensemble d'outils utilisés par les développeurs pour le développement d'un logiciel destiné à une plateforme déterminée (Linux, Windows, Android, etc.).

Un SDK est composé, a minima, d'un traducteur capable de traduire le langage de programmation en langage machine, d'un éditeur de liens en mesure de relier, en un fichier exécutable, différents éléments et de bibliothèques de routines.

Généralement, un SDK est imaginé et mis au point pour un seul ou plusieurs langages de programmation. Il peut avoir une seule ou plusieurs cibles comme un système d'exploitation, une application web, un serveur web, un jeu vidéo, etc. Ces dernières années, on le retrouve fréquemment dans le développement des applications mobiles.(3)

4.Android studio :

Android Studio propose un environnement de travail complet pour développer des applications sous Android. Vous pouvez y trouver un assistant permettant de créer rapidement un projet pour tous les périphériques Android, smartphones et tablettes, mais aussi Android Wear, Android auto, Google Glass et Android TV. L'assistant va se charger de récupérer le bon SDK Android, de paramétrer l'environnement de développement et de créer un émulateur pour tester votre appli pour la plateforme choisie.

L'éditeur de code d'Android Studio intègre des fonctions intelligentes comme l'autocomplétions ou l'analyse du code et met à disposition un outil permettant de gérer les différentes traductions de votre application. L'IDE de Google permet aussi de visualiser le résultat de votre travail sur différentes tailles d'écran et différents appareils émulés. Un moniteur permet aussi de vérifier l'optimisation de votre application en affichant l'utilisation de la mémoire durant son fonctionnement.(4)



Figure 60:logo Android studio

5. WampServer :

WampServer est une collection d'outil de développement web que tu peux utiliser pour installer un serveur Apache avec une base de données PHP et MySQL. Un module d'exploitation libre dont tu as besoin pour tester en local la fonction de ton site internet.(5)

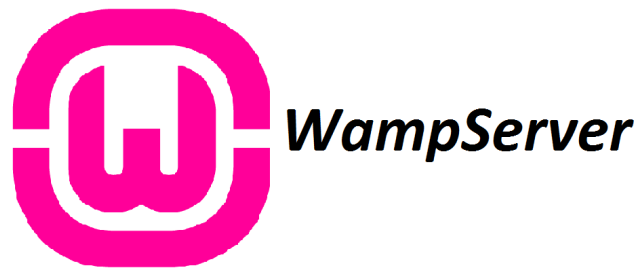


Figure 61 : logo WampServer

6.MySQL :

MySQL est un serveur de bases de données libre. Il intègre le langage SQL. Il est multi-thread, et multi-utilisateurs. Le logiciel a été développé sous double licence, en fonction de l'utilisation qui en est faite : dans un produit libre (open-source) ou dans un produit propriétaire. Pour un produit propriétaire, la licence est payante, sinon elle est libre. MySQL fonctionne sur plusieurs plates-formes dont: AIX, BSDi, FreeBSD, HP-UX, Linux, Mac OS X, NetWare, NetBSD, OpenBSD, OS/2 Warp, SGI Irix, Solaris, SunOS, SCO OpenServer, SCO UnixWare, Tru64 Unix, Windows 95, 98, NT, 2000, XP et Vista. Les bases de données sont accessibles en utilisant les langages de programmation C, C++, C#, Delphi / Kylix, Eiffel, Java, Perl, PHP, Python, Ruby et Tcl.(6)



Figure 62 :logo MySQL

7.Php :

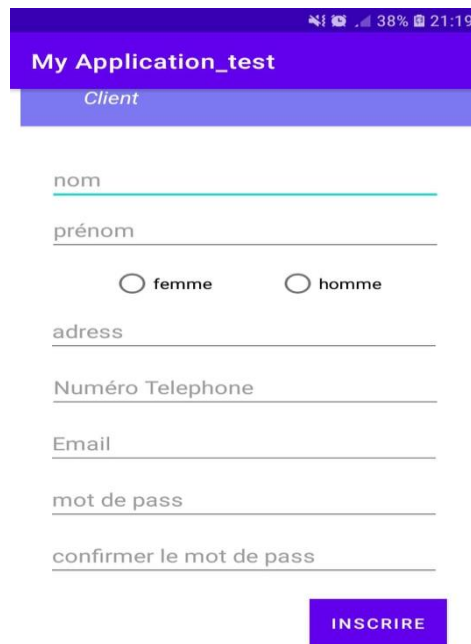
Le PHP, pour Hypertext Preprocessor, désigne un langage informatique, ou un langage de script, utilisé principalement pour la conception de sites web dynamiques. Il s'agit d'un langage de programmation sous licence libre qui peut donc être utilisé par n'importe qui de façon totalement gratuite.

Créé au début des années 1990 par le Canadien et Groenlandais Rasmus Lerdorf, le langage PHP est souvent associé au serveur de base de données MySQL et au serveur Apache. Avec le système d'exploitation Linux, il fait partie intégrante de la suite de logiciels libres LAMP.

Sur un plan technique, le PHP s'utilise la plupart du temps côté serveur. Il génère du code HTML, CSS ou encore XHTML, des données (en PNG, JPG, etc.) ou encore des fichiers PDF. Il fait, depuis de nombreuses années, l'objet d'un développement spécifique et jouit aujourd'hui une bonne réputation en matière de fiabilité et de performances.(7)

II. Présentation de calques interfaces de l'application :

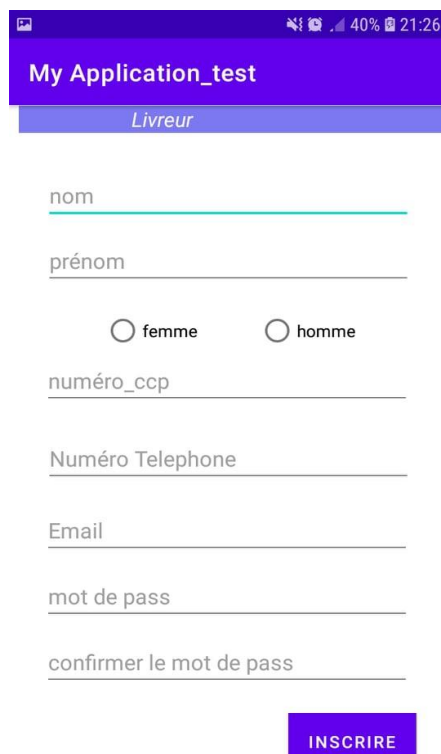
<<crée compte client >> :



The screenshot shows a mobile application interface titled "My Application_test" with a subtitle "Client". The interface contains several text input fields: "nom", "prénom", "adress", "Numéro Telephone", "Email", "mot de pass", and "confirmer le mot de pass". Between the "prénom" and "adress" fields, there are two radio buttons labeled "femme" and "homme". At the bottom right, there is a blue button labeled "INSCRIRE". The status bar at the top indicates a battery level of 38% and the time 21:19.

figure 63 :interface << crée un compte client>>

<<crée compte livreur>> :



The screenshot shows a mobile application interface titled "My Application_test" with a subtitle "Livreur". The interface contains several text input fields: "nom", "prénom", "numéro_ccp", "Numéro Telephone", "Email", "mot de pass", and "confirmer le mot de pass". Between the "prénom" and "numéro_ccp" fields, there are two radio buttons labeled "femme" and "homme". At the bottom right, there is a blue button labeled "INSCRIRE". The status bar at the top indicates a battery level of 40% and the time 21:26.

Figure 64 :interface <<crée un compte livreur>>

<<passer demande de livraison >> :



The screenshot shows a mobile application interface with a purple header bar labeled 'My Application_test'. Below the header, the text 'caractéristiques de la demande:' is displayed. The form consists of several input fields: 'Id_demande', 'titre', a dropdown menu for weight (showing '1-5kg', '5-20kg', '20-50kg', and '50kg>...'), 'Nom_destinataire', 'prénom_destinataire', 'Numéro_destinataire', 'état', and 'date_demande'. A purple button labeled 'PASSER' is located at the bottom right of the form.

Figure 65 : interface << passer une demande de livraison>>

<<page accueil client>>:



The screenshot shows a mobile application interface with a purple header bar labeled 'My Application_test'. Below the header, the text 'accueil' is displayed next to a house icon. A vertical list of six purple buttons is shown, each with an icon and text: 'PASSER UNE DEMANDE' (with a plus icon), 'CONSULTER LES OFFRES DE LIVRAISON' (with a list icon), 'VALIDER LA LIVRAISON DU COLIS' (with a checkmark icon), 'PASSER FEEDBACK' (with a speech bubble icon), 'CONSULTER PROFILE' (with a person icon), and 'DECONNEXION' (with a logout icon).

Figure 66 : interface <<page accueil client>>

Conclusion :

Dans ce chapitre, nous avons présenté les outils et les logiciels utilisés pour le développement de notre application. Ensuite, nous avons présenté quelques interfaces de l'application. La phase d'implémentation, nous a permis de matérialiser les résultats des phases précédentes. Elle nous a permis également d'apprendre des langages et des outils nécessaires pour le développement des applications mobile.

Les Références :

Chapitre01 :

- (1) <https://whatis.techtarget.com/fr/definition/application-mobile>
- (2) <https://infonet.fr/lexique/definitions/application-mobile/>
- (3) <https://cours-informatique-gratuit.fr/dictionnaire/android/>
- (4) <https://cours-informatique-gratuit.fr/dictionnaire/ios/>
- (5) <http://www.dicodunet.com/definitions/multimedia/blackberry.htm>
- (6) <https://www.futura-sciences.com/tech/definitions/smartphone-windows-phone-15584/>
- (7) <http://www.android-dev.fr/pourquoi-developper-sous-android>
- (8) <https://www.243tech.com/naissance-et-evolution-android/>
- (9) <http://www.w3big.com/fr/android/android-architecture.html>
- (10) <https://www.redhat.com/fr/topics/cloud-native-apps/what-is-SDK>
- (11) <https://www.nartex.fr/application-ou-site-mobile-la-suite/>
- (12) <https://www.lebigdata.fr/base-de-donnees>
- (13) <https://www.cnil.fr/>
- (14) [https://www.gps.gov/systems/gps/french.php#:~:text=Le%20GPS%2C%20qu'est%2D,%2C%20navigation%2C%20and%20timing\).](https://www.gps.gov/systems/gps/french.php#:~:text=Le%20GPS%2C%20qu'est%2D,%2C%20navigation%2C%20and%20timing).)
- (15) <https://www.techno-science.net/definition/375.html#:~:text=Google%20Maps%20est%20un%20service,sous%20le%20nom%20de%20Google>

Chapitre02 :

- (1) <https://m.21-bal.com/law/6091/index.html?page=5>
- (2) <https://blog.hubspot.fr/marketing/cahier-des-charges-application-mobile>
- (3) <http://prive.iutenligne.net/iuRxMoCThIXDIjzG/informatique/langages/kettaf/UML/o4modeleconceptuel/o3o2acteurs.html>
- (4) <https://www.ibm.com/docs/fr/rational-soft-arch/9.6.1?topic=diagrams-messages-in-uml>
- (5) <https://blog.ippon.fr/2020/10/16/diagramme-de-contexte-vs-hexagone/>

Chapitre03 :

- (1) https://www.memoireonline.com/10/18/10437/m_Developpement-d-une-application-d-apprentissage-de-langue3.html
- (2) <https://www.ibm.com/docs/fr/rational-soft-arch/9.5?topic=diagrams-use-case>
- (3) Pascal Roques : « Les cahiers du programmeur UML2 modélisé une application Web » édition Groupe Eyrolles, 2007.
- (4) <https://wikimemoires.net/2011/04/capture-des-besoins-techniques/>

Chapitre05 :

- (1) [https://www.java.com/fr/download/help/whatis_java.html]
- (2) [<https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203615-xml-extensible-markup-language-definition-traduction/>]
- (3) [<https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203359-sdk-software-development-kit-definition-traduction/#:~:text=Un%20SDK%2C%20pour%20Software%20Development,%2C%2>]

[oAndroid%2C%20etc.\).&text=G%C3%A9n%C3%A9ralement%2C%20un%20SDK%20est%20imagin%C3%A9,ou%20plusieurs%20langages%20de%20programmation. \]](#)

(4)

[<https://www.oinet.com/telecharger/windows/Programmation/creation/fiches/129241.html>]

(5)

[<https://wampserver.fr.uptodown.com/windows>]

(6)[<https://www.clubic.com/telecharger-fiche204276-mysql.html>]

(7) [<https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203597-php-hypertext-preprocessor-definition/>]