

Sommaire

Chapitre I : Introduction Générale

I-Contexte	1
II- Problématique	2
III- Objectifs.....	2
IV- Plan du mémoire.....	3

Chapitre II : Analyse et Conception

I-Introduction	4
II- Les outils de géolocalisation et les techniques de tracking	4
II-1 Google Map API.....	4
II-2 GPS (Global Positioning System).....	4
II-3 Traçabilité	5
II-4 Tracking	5
II-5 Géolocalisation	5
III- Phase d'analyse.....	6
III -1 le langage UML	6
III -2 Entreprise Architect	6
III -3 Analyse globale du système.....	7
IV-Diagramme de cas d'utilisation.....	8
V- Diagramme de séquence	8
VI- Diagramme de class	23
VII- Conclusion	25

Chapitre III : Réalisation

I- Gestion du projet	26
<u>I-1</u> La méthode Scrum	26
<u>I-2</u> Fichier de suivi du projet.....	28
<u>I-3</u> Diagramme de Gantt	29
II-L'environnement de développement.....	30
II-1 Le Framework Laravel.....	30
II-2 Installation et mise en marche du Framework	31
III- Le système de gestion de base de donnée « My SQL ».....	33
IV- Tests et Intégration	34
V- Démonstration.....	35
V-1 Authentification	35
V-2 Menu Administrateur.....	35
V-3 Gestion des employés	36
V-4Gestion des localités	36
V-5 Gestion des secteurs d'activités	39
V-6 Géolocalisation	40
V-7 Réclamation client	44
V-8 Preuve de livraison	44
V-9 Suivie de livraison	45
VI- Conclusion	45

I- Contexte

La société Eurequat Algérie a démarré son activité en mars 2006, avec le soutien et les moyens techniques de la société Eurequat Technologie Europe, spécialisé dans le domaine de la distribution des produits de traçabilité et d'identification qui englobe le domaine du badge et du contrôle d'accès, l'entreprise dispose des équipes commerciales, métiers et techniques au niveau de trois site Alger, Oran et Tlemcen.

L'entreprise travaille actuellement sur quatre projets différents, dont le projet intitulé « Tracking colis », c'est une solution informatique destinée aux entreprises postales telles que UPS et DHL, *etc.* afin de leur assurer une meilleur gestion, un bon suivi des colis, des livraisons plus optimales et leur permettre d'offrir un service après livraison. La figure ci-dessous détail l'architecture de ce projet.

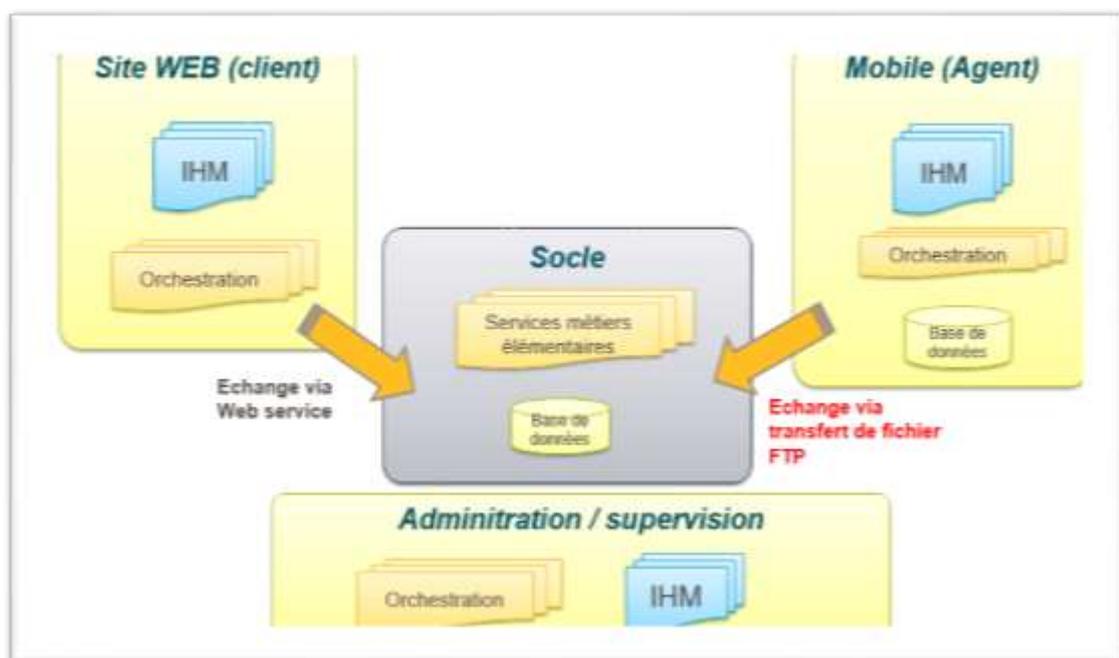


Figure I.1 Architecture globale du projet Tracking Colis

- Le module **Front office web** destiné aux clients finaux et/ou aux concessionnaires qui offre la possibilité de créer un compte à fin d'accéder aux services de *tracking* (suivi) et de suivre l'évolution des différentes livraisons les concernant.
- Le module **Front office mobile** qui permet aux agents d'enregistrer les différents évènements sur le terrain, sauvegarder les problèmes ainsi que le motif de non

livraison ou de réception et d'enregistrer la preuve de livraison en l'occurrence la signature du client.

- Le module **Back office web** qui permet de faire la liaison entre les deux autres modules. C'est sur ce module que nous allons nous intéresser dans ce travail.

Après avoir pris connaissance de ce projet proposé par l'entreprise, nous avons été fortement motivés de travailler dessus non seulement pour son intérêt mais également pour pouvoir vivre une expérience professionnelle au sein de l'entreprise.

II- Problématique

A l'issue de notre entretien avec le chef de projet on a conclu que seulement deux modules du système ont été réalisés, c'est la solution mobile qui est dédiée aux agents et le front office web destiné aux clients, mais à ce niveau-là on ne peut pas mettre en relation ces deux solutions car lorsque le client passe une commande on ne peut ni affecter cette dernière à un agent ni effectuer le suivi des différents colis ou d'assurer le service après livraison, ni gérer les différents secteurs d'activité et de gérer les employés.

III- Objectifs

Le meilleur moyen c'est de compléter les deux solutions avec un autre module qui sera dédié à deux types d'utilisateurs :

Un **administrateur** qui a la possibilité de :

- ✓ enregistrer les clients et de leur fournir des droits d'accès
- ✓ gérer les employés
- ✓ gérer les agences
- ✓ gérer les secteurs d'activité

Un **superviseur** ayant le droit de :

- ✓ suivre les tournées des agents
- ✓ planifier l'activité des agents
- ✓ suivre les demandes des clients et leurs réclamations
- ✓ fournir des preuves de livraison
- ✓ générer et visualiser des rapports.

IV- Plan du mémoire

Notre mémoire est subdivisé en trois chapitres, Le premier est une introduction générale sur le contexte du projet et la présentation de l'entreprise. Le second s'attèle sur l'analyse et la conception du projet. Le troisième est consacré à la gestion de projet et à la réalisation du Back office. Enfin nous terminerons par une conclusion générale qui résume notre travail avec des perspectives pour les futurs travaux sur ce projet.

I-Introduction

Il est absurde de penser que nous pourrions comprendre un besoin exprimé totalement dès le début du projet. Comprendre ce que veut un client ou un utilisateur n'est pas toujours évident, au début du projet, nous étions un peu perturbés, nous nous sommes posés plusieurs questions : que souhaitent-ils faire avec le Back office ? Quelles fonctionnalités veulent-ils ? Pour quel usage ? Comment l'action devrait-elle fonctionner?

II- Les outils de géolocalisation et les techniques de tracking**II-1 Google Map API**

L'API Google Map est l'une des applications de cartographie les plus utilisées au monde. C'est une application de service de géolocalisation gratuite en ligne. Il s'agit d'un géo portail lancé il y a quelques années aux Etats-Unis puis à l'Europe.

Elle offre une vue de carte sur quatre plans à savoir un plan classique, un plan en image satellite, un plan mixte et un plan relief de la région.

Pour intégrer ces cartes interactives Google Map à son propre application et bénéficier des données associées, l'utilisateur doit disposer d'une clé (Google Map API Key) propre à son domaine d'utilisation. [16]

II-2 GPS (Global Positioning System)

C'est un système de positionnement par satellite connu sous le nom de « Global Positioning System », conçu à l'origine par le département de la Défense des Etats-Unis. Il repose sur 24 satellites qui transmettent les positions en orbite ainsi que des signaux horaires.



Couplé avec un logiciel de navigation, le GPS permet à l'utilisateur de connaître sa position, planifier son itinéraire, estimer son temps du trajet, connaître la localisation des radars, etc. Il renvoie même la vitesse de son déplacement à chaque instant. En outre, le positionnement par satellite est entièrement gratuit, mais pour s'en servir il faut disposer d'un dispositif GPS qui est intégrés dans la plupart des smartphone.[16]

II-3 Traçabilité

La traçabilité est, selon la norme ISO 8402, "l'aptitude à retrouver l'historique, l'utilisation ou la localisation d'un article ou d'une activité au moyen d'une identification enregistrée. Elle se réfère au recueil d'informations à chaque étape d'un processus afin d'en tracer les événements tels que le suivi d'activité, suivi de colis... [14]

II-4 Tracking

Dans le domaine de la logistique, tracing & tracking est le concept de traçabilité et de localisation d'un bien, d'une marchandise ou d'un équipement à chaque moment d'un processus de fabrication ou de transport.

Cette association des deux termes tracing & tracking est apparue au moment de l'utilisation des moyens informatiques utilisant Internet dans le domaine du transport de colis (FedEx, DHL...). Le tracing est l'action d'obtenir l'historique horodaté d'un colis. Le tracking est l'action de recherche en temps réel de la localisation de ce colis. [14]



II-5 Géolocalisation

A l'origine, La géolocalisation est la localisation d'un objet sur une carte à l'aide de positions géographiques. On utilise la géolocalisation par GPS (Global Positioning System) pour localiser des mobiles en déplacement.

Depuis quelques années, ce concept est de plus en plus utilisé pour des applications de gestion de flottes et de sécurisation de véhicules.

Les dispositifs de géolocalisation permettent aux employeurs de prendre connaissance de la position géographique de leurs salariés, à un instant donné ou en continu, par la localisation des véhicules mis à leur disposition pour l'accomplissement de leur mission.

Les systèmes de géolocalisation des véhicules sont basés sur le traitement d'informations issues de satellites GPS couplés à l'utilisation d'un réseau De communication électroniques.

Aujourd'hui la définition est celle du CNIL : « Les dispositifs dits de géolocalisation permettant aux employeurs privés ou publics de prendre connaissance de la position géographique, à un instant donné ou en continu, des employés par la localisation d'objets

dont ils ont l'usage (badge, téléphone mobile) ou des véhicules qui leur sont confiés. » [15]

La géolocalisation utilise un support logiciel qui gère la traçabilité des biens, marchandises, équipements ou véhicules, mais aussi d'individus, en utilisant les méthodes d'identification courantes (codes-barres, RFID, GPS,...), et qui transmet ces informations en temps réel. En effet c'est le point fort de notre application.[14]

III- Phase d'analyse

Pour la clarification, l'affinage et la validation des besoins des utilisateurs nous avons entamé **la phase d'analyse** pour bien comprendre et décrire de façon précise les besoins des utilisateurs ou des clients.

La phase de modélisation nécessite des méthodes permettant de mettre en place un modèle d'analyse. Dans ce cas, nous avons adopté le langage UML.

III -1 le langage UML

UML « Unified Modeling Language »[1], est un langage de modélisation graphique des données et des traitements. À travers le langage UML, nous pouvons donner plusieurs vues d'un système.

Pour l'analyse et la conception du système nous avons choisi d'étudier le diagramme de cas d'utilisation et le diagramme de séquence pour la partie analyse, et diagramme de classe pour la partie conception, l'ensemble des diagrammes sera validé par la conception des maquettes du Back office.



III -2 Entreprise Architect

Nous avons utilisé « Enterprise Architect » un outil de conception UML (Unified Modeling Langage) complet,



Couvre le développement de logiciels de la collecte des besoins jusqu'à la phase d'analyse, des modèles de conception, les tests et la maintenance. C'est un multi-utilisateur, basé sur Windows, outil graphique qui nous a aidé dans les différentes tâches liées à l'analyse et la conception. [2]

III -3 Analyse globale du système

Chaque usage que les acteurs font du système est représenté par un cas d'utilisation [3]. Ce dernier décrit le comportement du système du point de vue de son utilisateur. Le projet Tracking comporte quatre principaux acteurs qui interagissent avec le système :

a- Client

Il peut être une personne ou un organisme, ayant la possibilité d'effectuer une commande, saisir les réclamations et traquer son colis via le **front office web**.

**b- Agent**

C'est un employé de l'agence ayant la possibilité de lister ses activités, enregistrer les différents événements liés à la livraison du colis (prélèvement client, réception agence, prélèvement agence, livraison client), et éventuellement signaler des problèmes en cas de non-livraison, en utilisant l'application mobile

**c- Administrateur**

Il a la possibilité de contrôler le back Office, gérer les employés, les clients, les agences et les secteurs d'activités.

d- Superviseur

Il a le privilège de suivre les tournés des agents, planifier l'activité d'un agent, suivre les réclamations clients, suivre la livraison et fournir la preuve de livraison, via le back office. Il est intéressant de noter que notre module Back office est dédié uniquement aux administrateurs et superviseurs.



IV-Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation ci-dessous est un diagramme global, il définit quatre principaux acteurs qui interagissent avec le système, en déterminant les besoins de l'utilisateur et tout ce que doit faire le système pour ces acteurs.

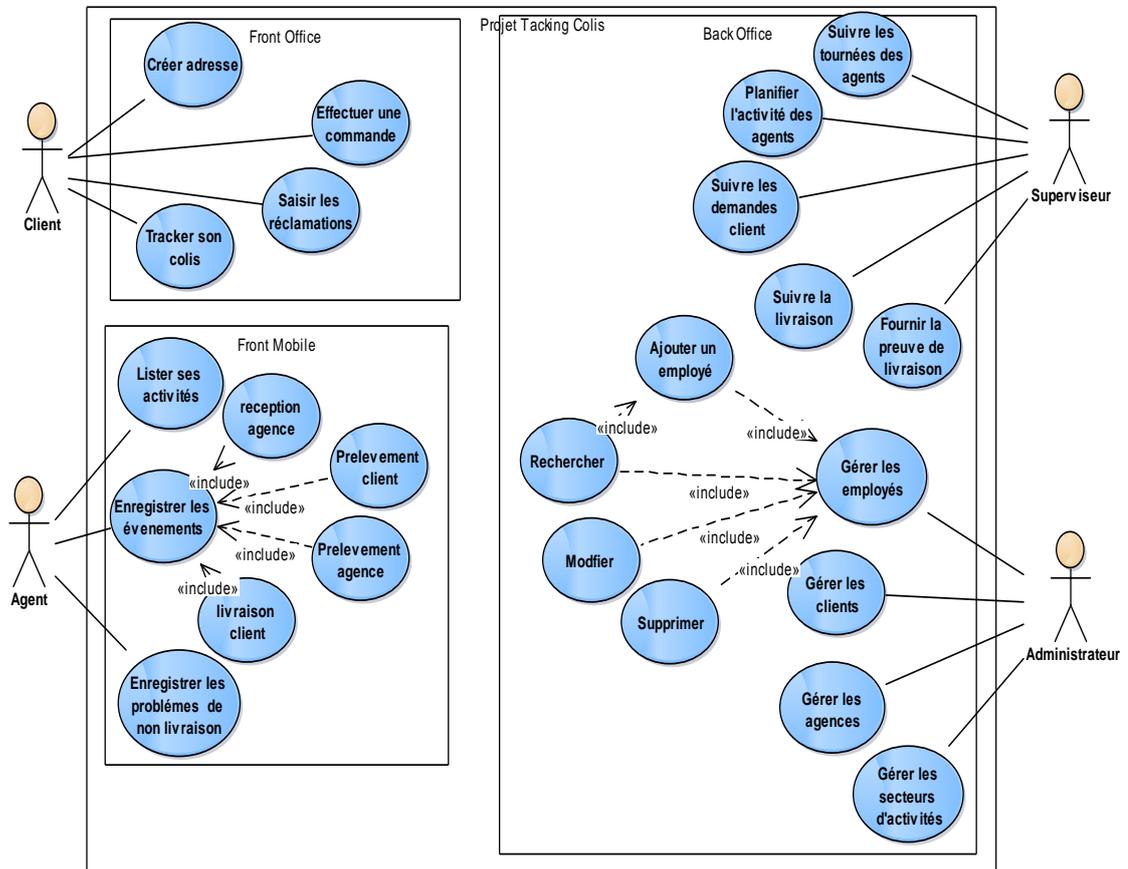


Figure II.1 Diagramme de cas d'utilisation

V- Diagramme de séquence

Après avoir identifié les différents acteurs et les cas d'utilisation du système, Nous allons maintenant les décrire de façon détaillée afin d'obtenir une expression précise des besoins, chaque **diagramme de séquence** [4] présenté ci-dessous sera validé par quelque exemple de **maquette de conception** que nous jugeons les plus importantes.

a- Diagramme de séquence « Créer adresse »

Pour effectuer une commande, le client doit créer son carnet d'adresse, afin de pouvoir spécifier l'adresse d'envoi et de destination du colis, donc pour créer une adresse le client doit être connecté et remplir un formulaire dédié au adresses.

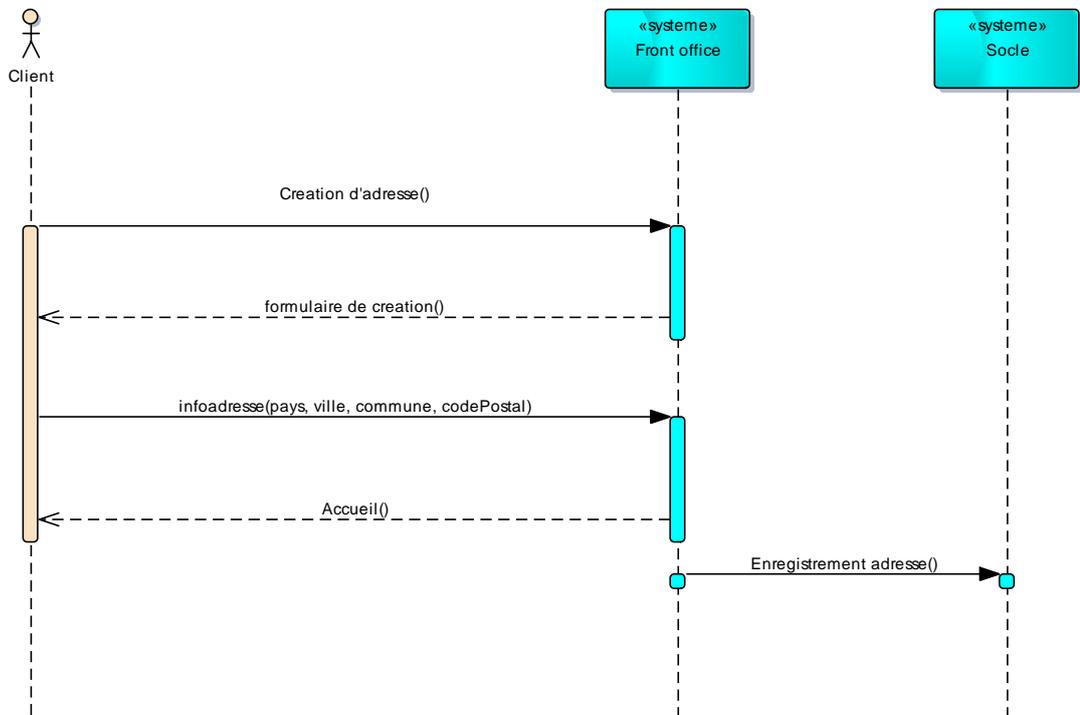


Figure II.2 Diagramme de séquence « créer adresse »

b- Diagramme de séquence « Effectuer une commande »

Après avoir créé son carnet d'adresse le client a la possibilité de passer des commandes, pour ce faire il doit être connecté et renseigner les différentes informations concernant le colis (adresse, poids, quantité, longueur, largeur...).

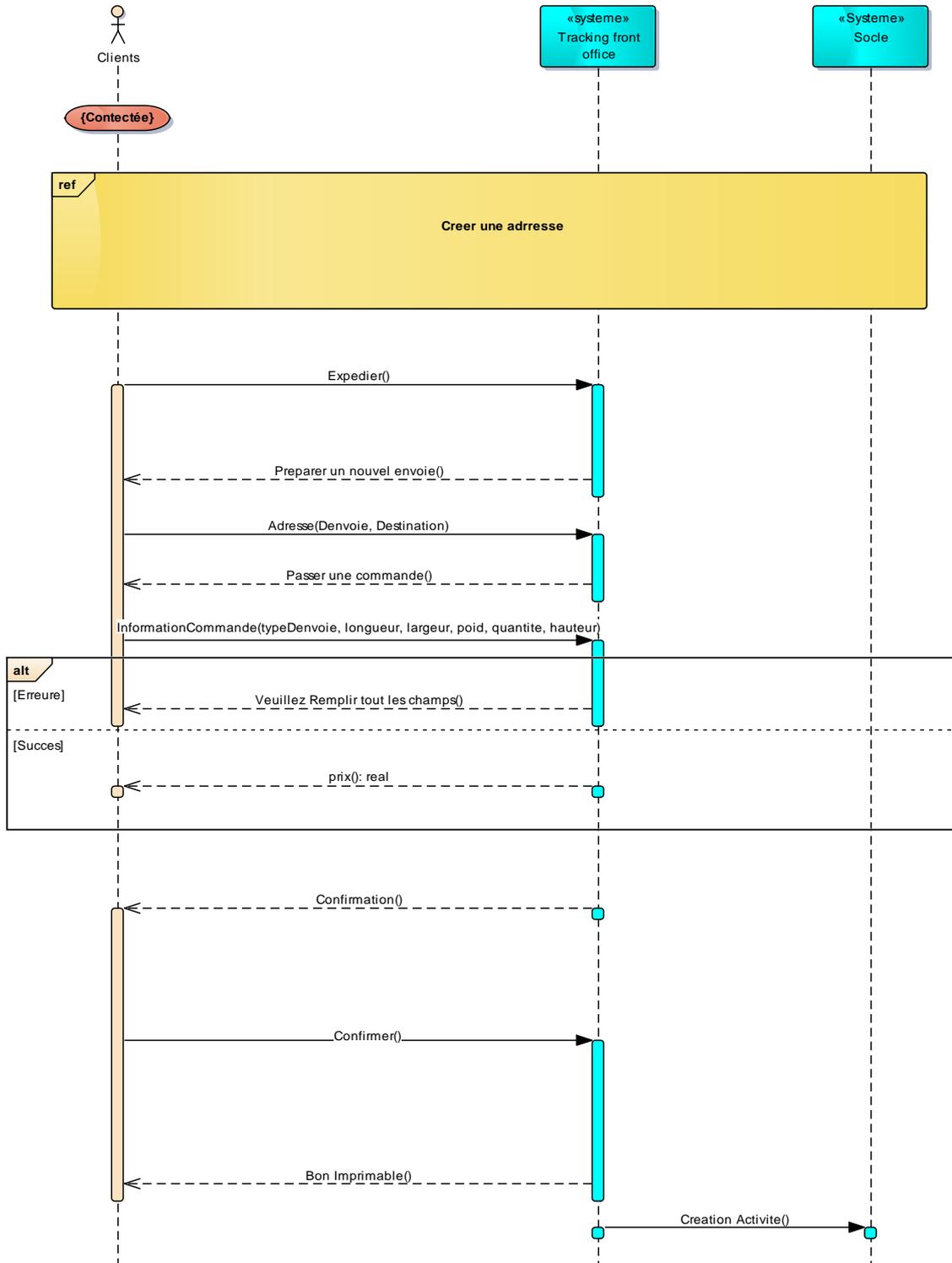


Figure II.3 Diagramme de séquence « passer commande »

c- Diagramme de séquence « Affecter les activités »

Une fois que le client valide la commande une activité sera automatiquement générer, cette dernière va être utilisée par le superviseur qui lui va l’affecter à un agent de terrain via le back office.

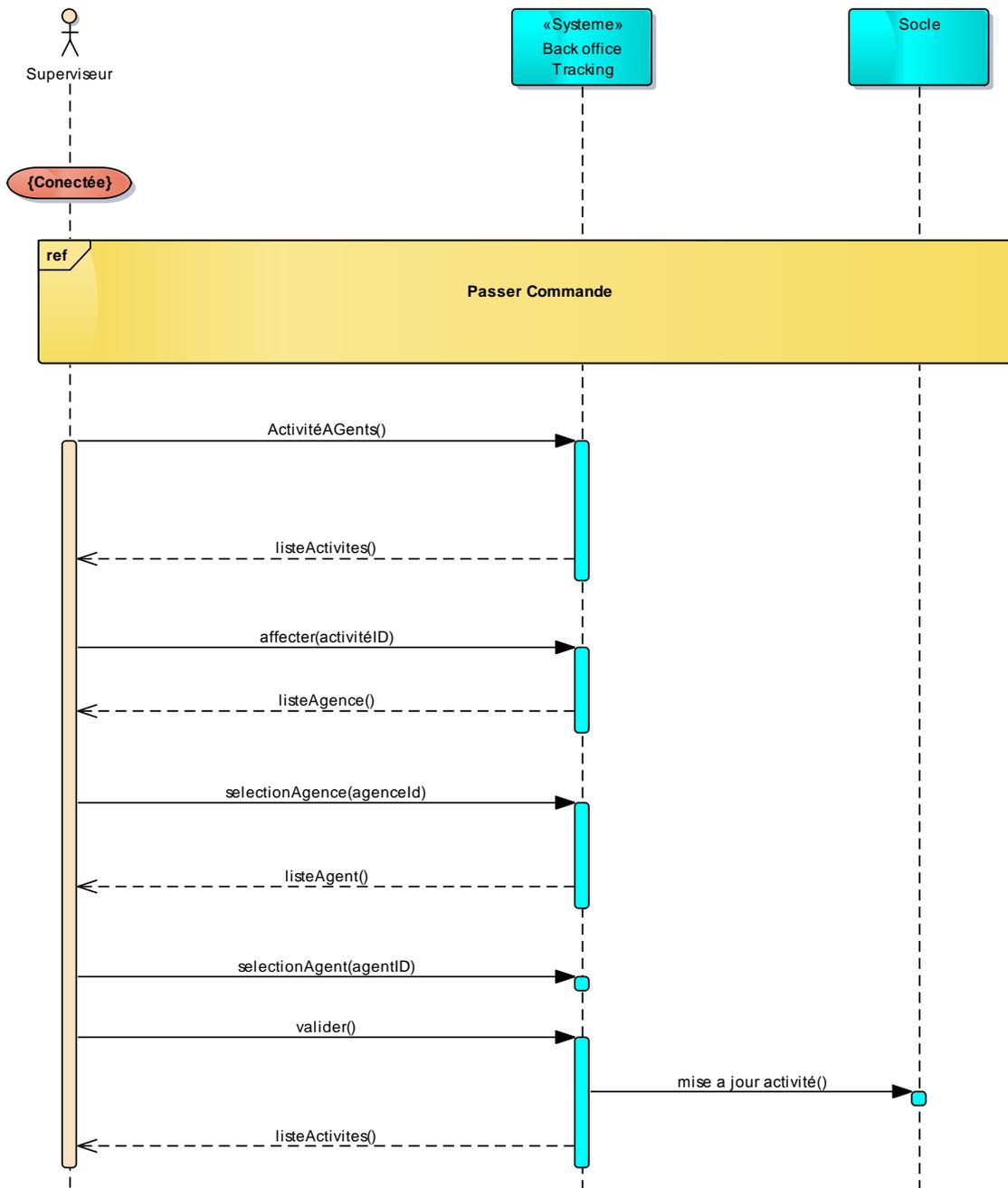


Figure II.4 Diagramme de Séquence « Affecter une activité »

Nous avons validé le scénario d'affectation d'activité par cette maquette :

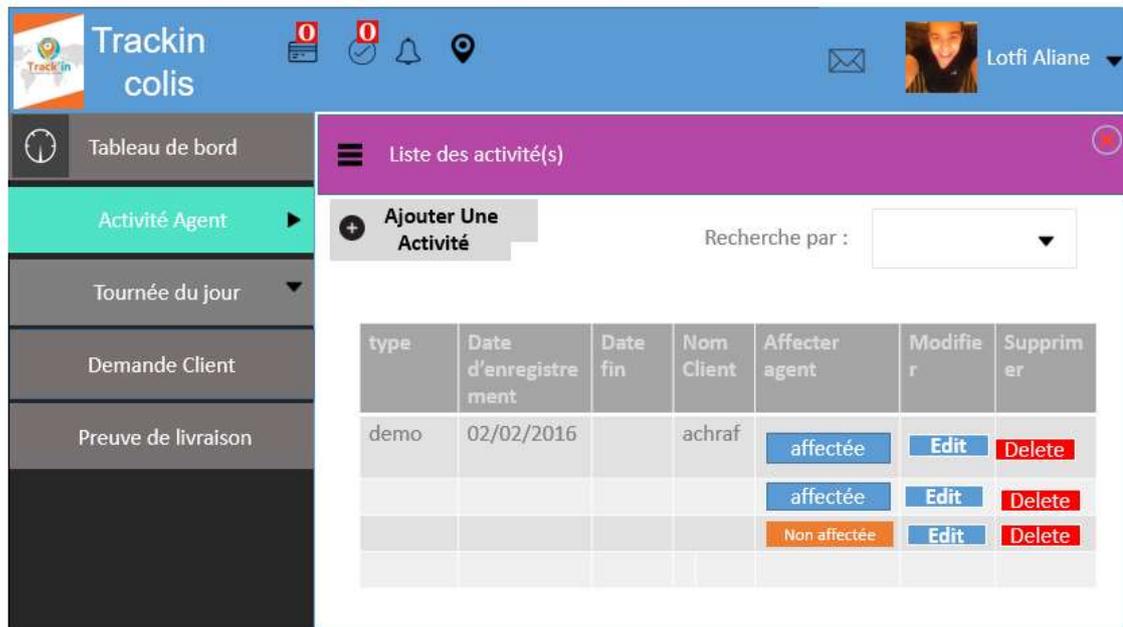


Figure II.5 Maquette d'activité Agent

Pour la réalisation des maquettes du BackOffice nous sommes passés par deux étapes : la définition de la mise en page de la maquette fonctionnelle ensuite on l'a développée en maquette graphique avec l'outil dédié à la réalisation des maquettes « PowerMockup »¹, qui fournit une grande collection d'éléments de l'interface utilisateur et des icônes entièrement réalisés à partir de formes PowerPoint permettant de créer des prototypes d'écran de mobile, web et applications desktop.²



¹<https://www.powermockup.com/>

²Nous n'avons pas présenté la totalité des maquettes de conception dans cette phase car elles seront détaillées dans la phase de réalisation en l'occurrence le chapitre III

d- Diagramme de séquence « Prélèvement Client »

L'agent a la possibilité lister les différentes activités qui lui ont été affectées par le superviseur, en sélectionnant une activité le système renvoie une vue où il aura la possibilité de scanner le code de la commande pour enregistrer le prélèvement de la commande du client.

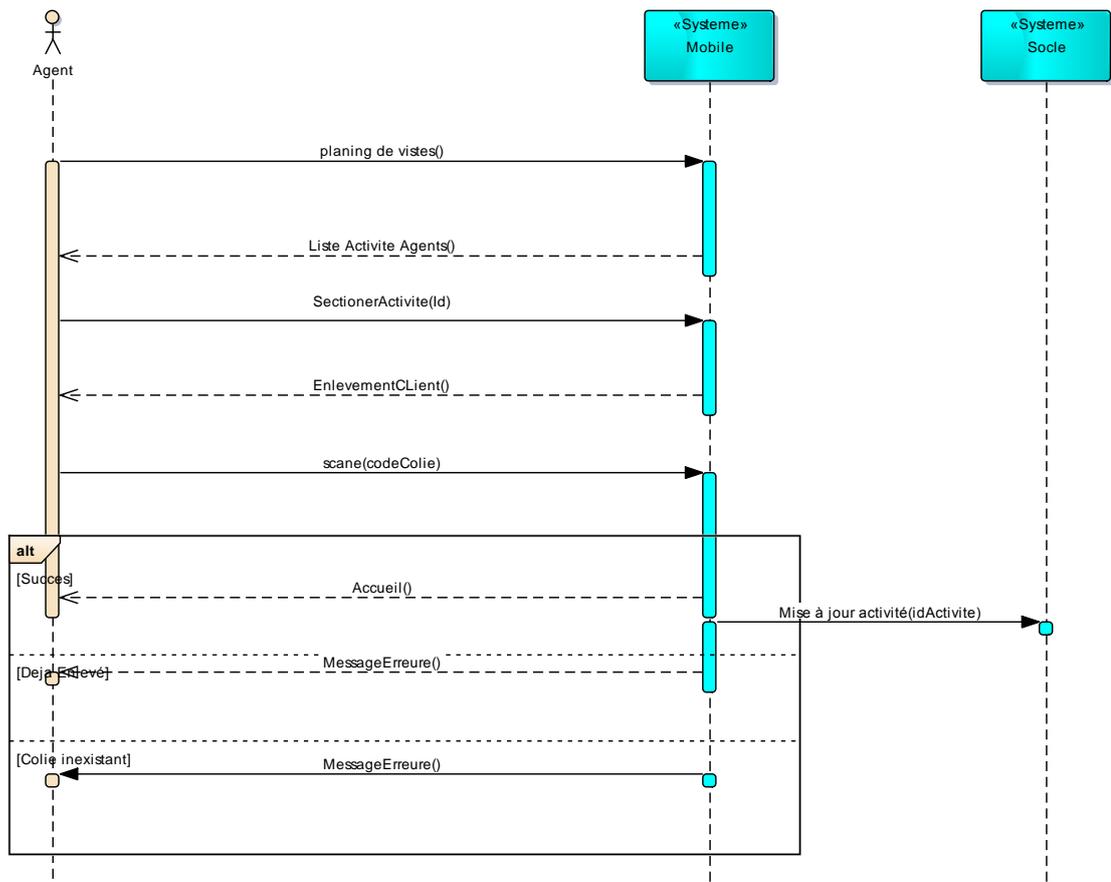


Figure II.6 Diagramme de Séquence « Prélèvement client »

e- Diagramme de séquence « Réception Agence »

Au moment du dépôt du colis à l'agence, l'agent scanne le code du colis qui mettra à jour l'activité au niveau du serveur.

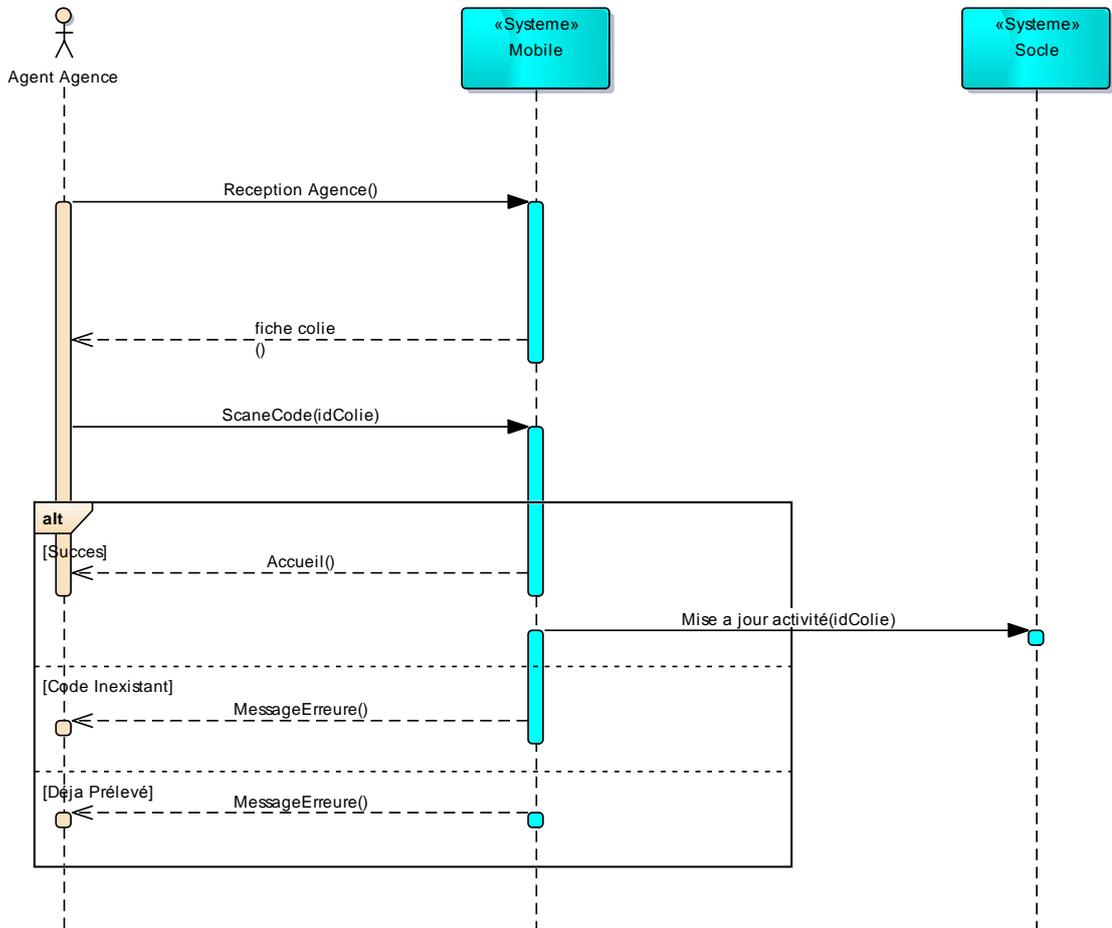


Figure II.7 Diagramme de Séquence « Réception Agence »

f- Diagramme de séquence « Prélèvement Agence »

Au moment du prélèvement du colis de l'agence, l'agent accède à la vue dédiée au prélèvement puis scanne le code du colis en validant le système mets automatiquement l'activité à jour.

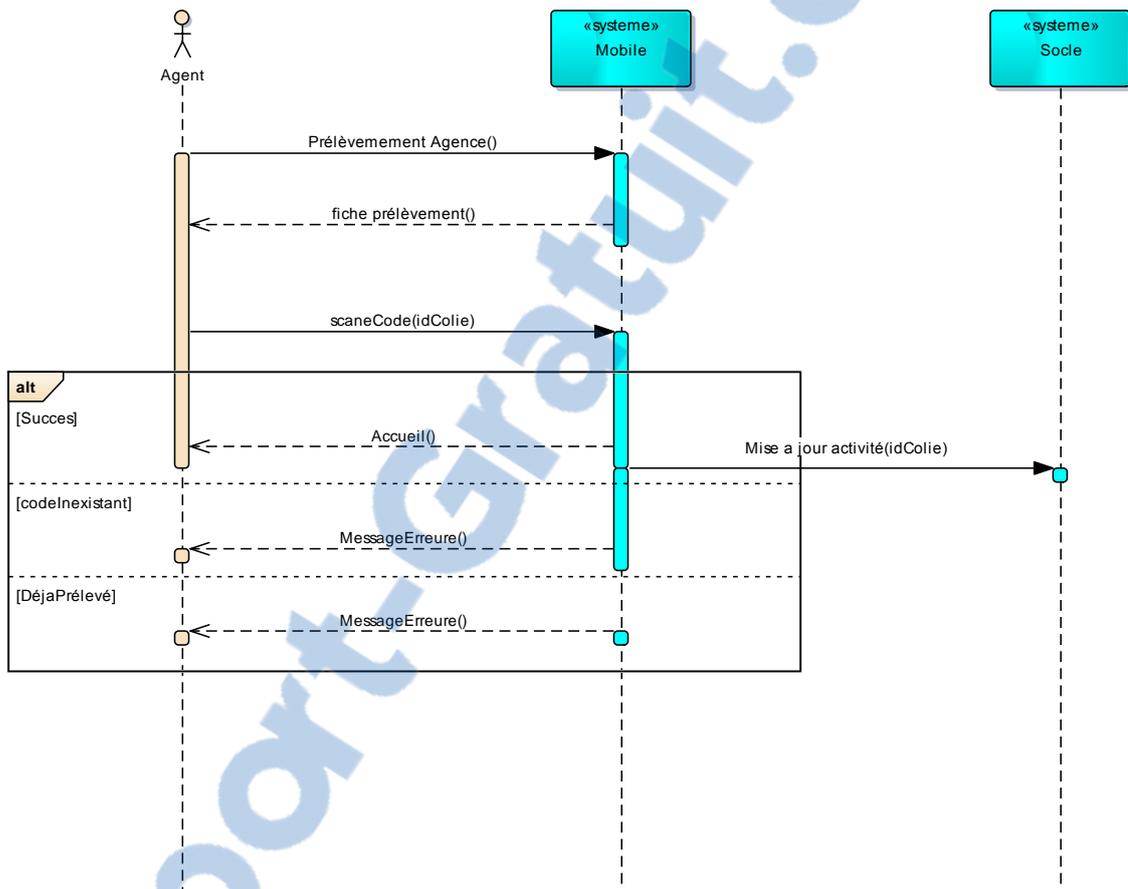


Figure II.8 Diagramme de Séquence « Prélèvement Agence »

g- Diagramme de séquence « Livraison Client »

L'agent accède à la vue de livraison puis scanne le code et donne la main au client de signer et valide ce qui mettra à jour l'activité.

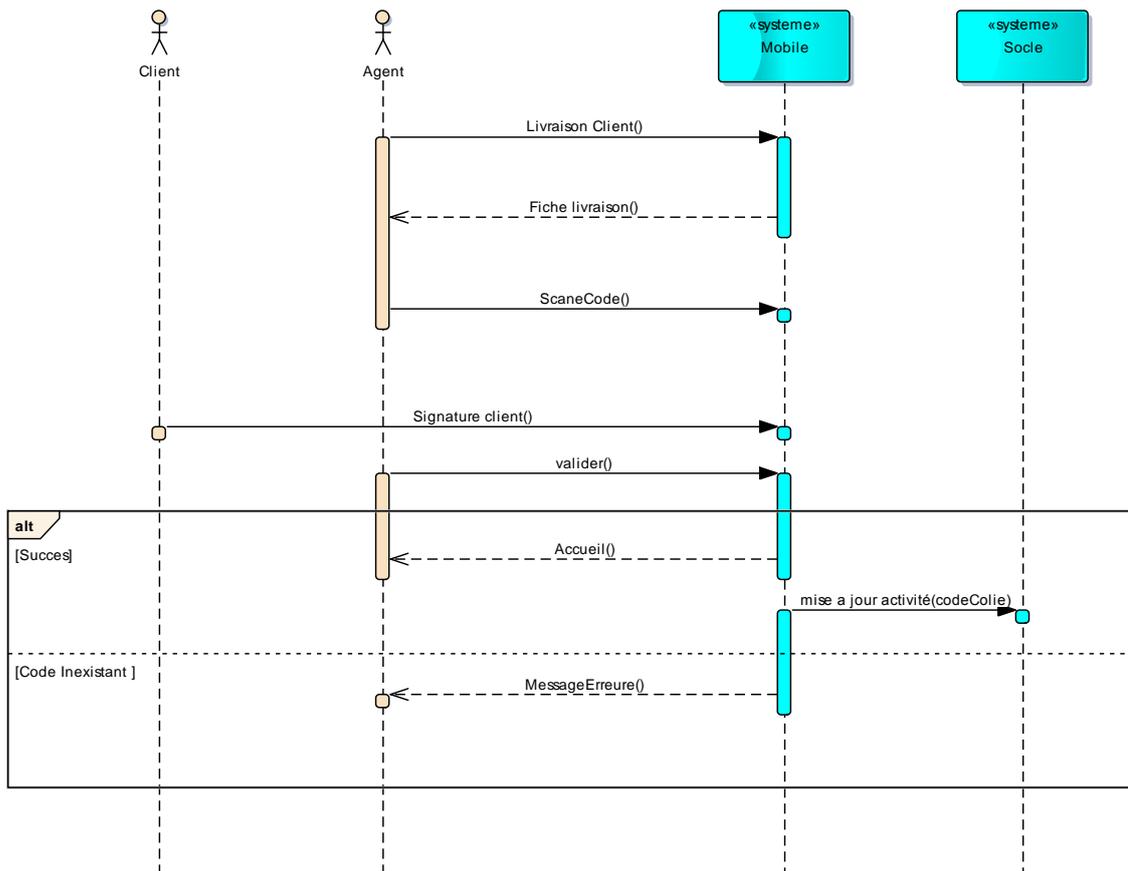


Figure II.9 Diagramme de Séquence « Livraison Client »

h- Diagramme de séquence « Traquer colis client »

Une fois que la commande est effectuée le client a la possibilité de voir le suivi de son colis une fois qu'il aura saisi le code du colis et valider via le front office.

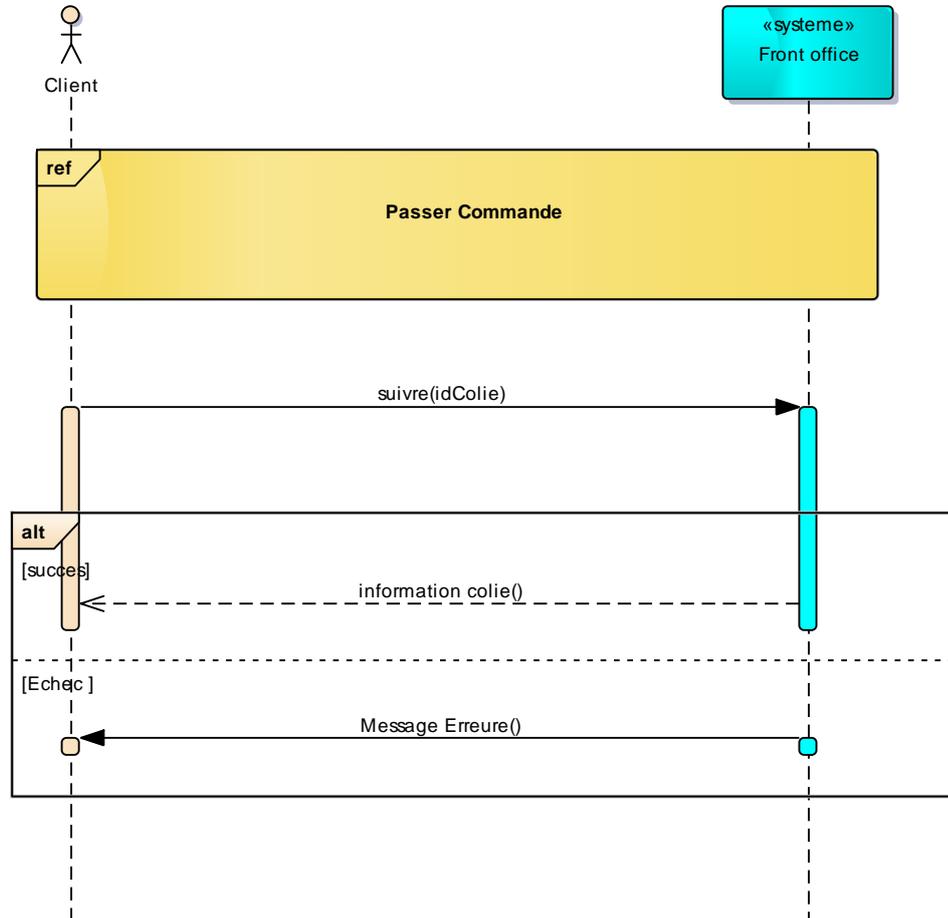


Figure II.10 Diagramme de Séquence « Traquer colis client »

i- Diagramme de séquence « Traquer colis Superviseur »

En accédant à la liste des activités le superviseur voir l'état et accéder au suivi du colis via le back office.

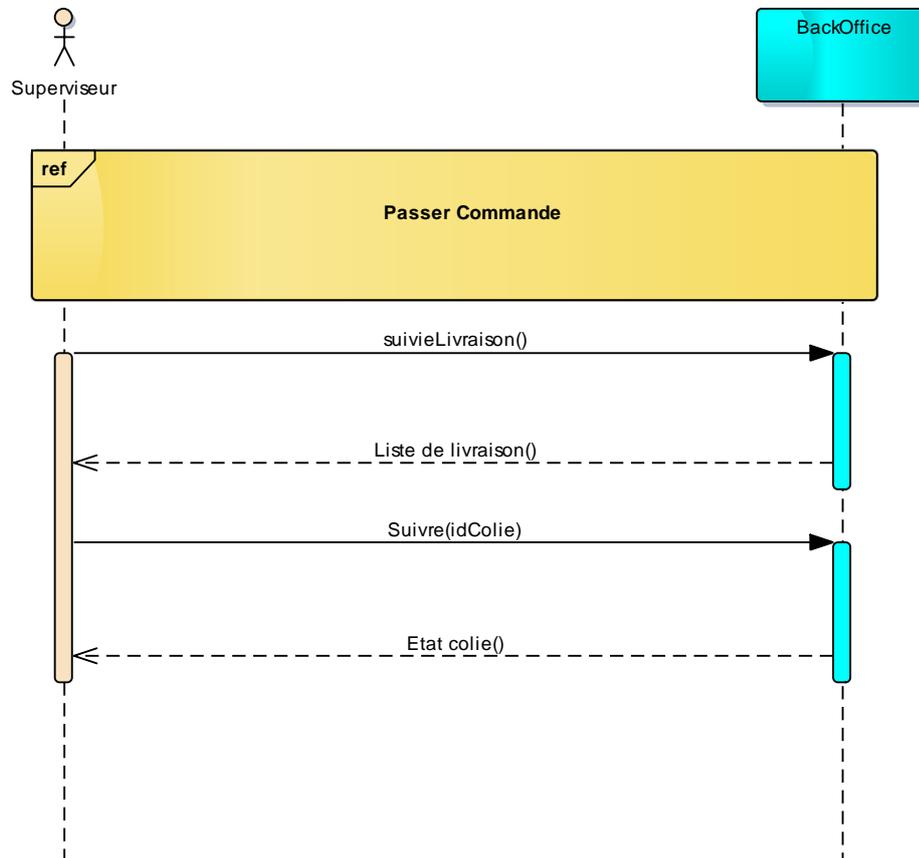


Figure II.11 Diagramme de Séquence « Traquer colis Superviseur »

j- Diagramme de séquence « géo localiser un employé »

En accédant en menu tourné agent le superviseur pourra l’localiser en temps réel ses employés, et éventuellement localiser chaque employé séparément.

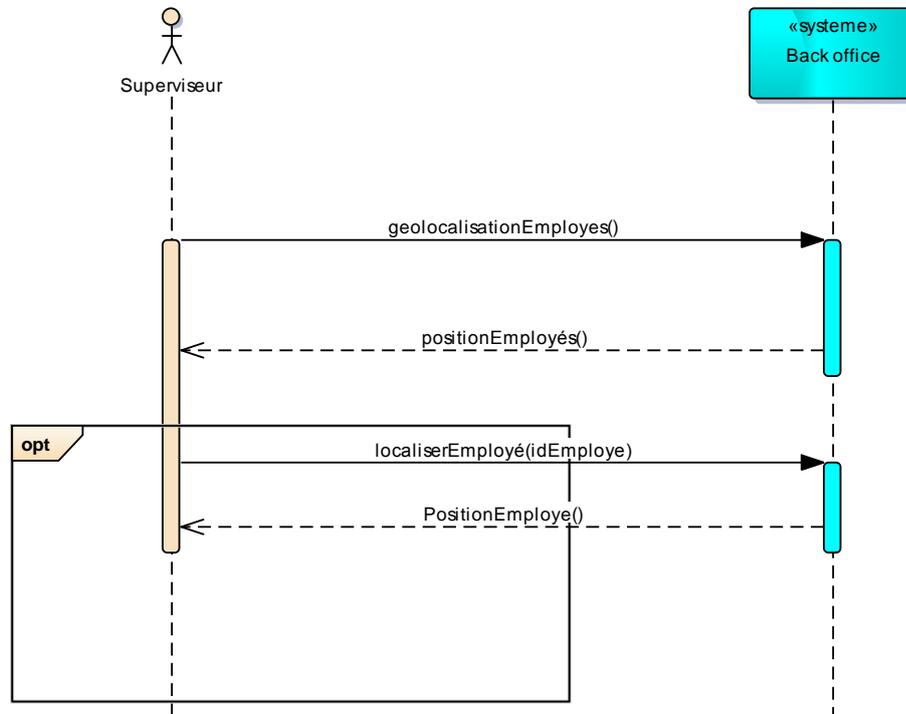


Figure II.12 Diagramme de Séquence « géo localiser un employé »

Nous avons validé le scénario de géolocalisation des employés par la maquette suivante :



Figure II.13 Maquette de « géolocalisation d'employé »

k- Diagramme de séquence « gérer les localités »

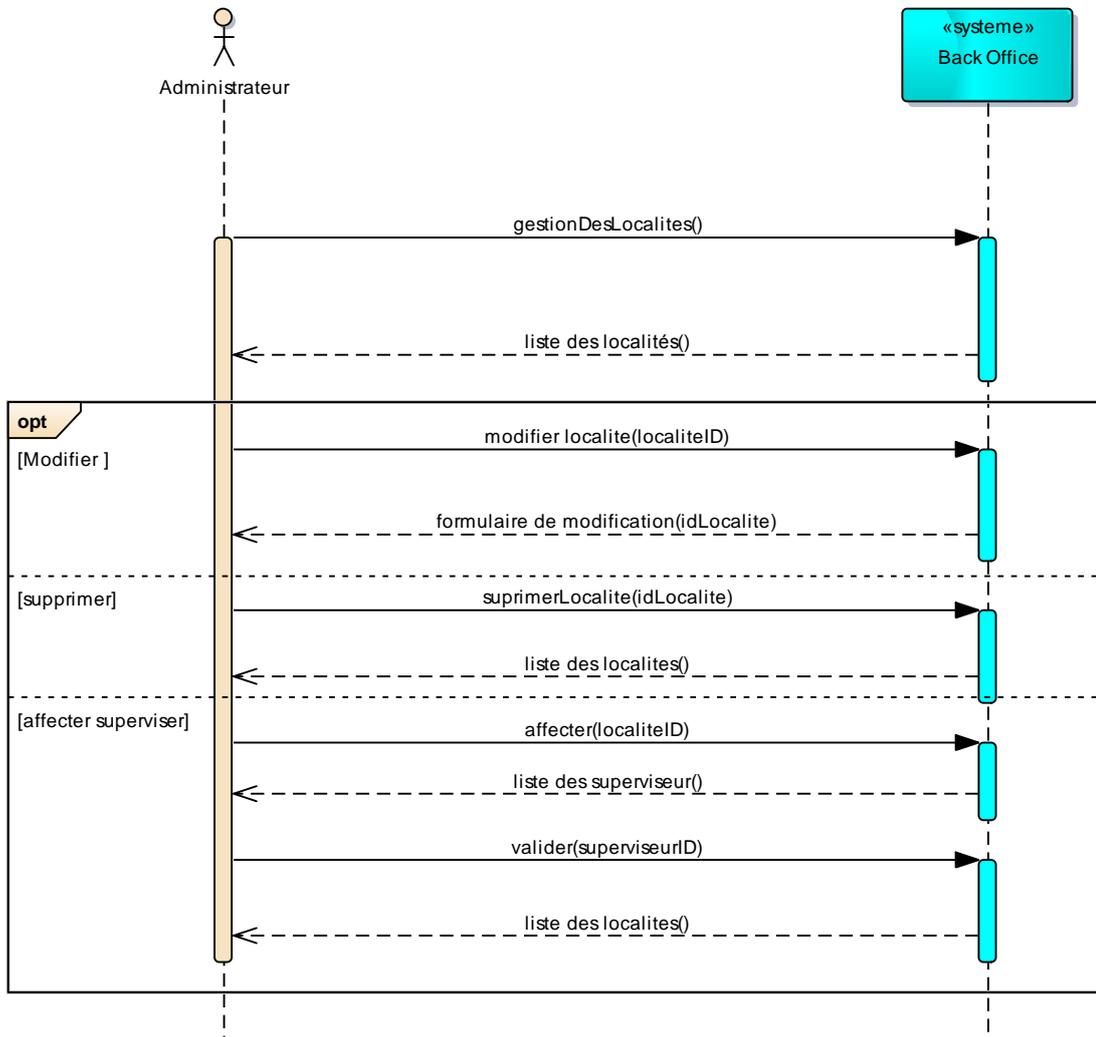


Figure II.14 Diagramme de Séquence « gérer les localités »

I- Diagramme de séquence « gérer Les secteurs d'activités »

En accédant en menu gestion des activités, l'administrateur aura la liste des différents secteurs d'activités avec les employés associés à ce secteur, il peut éventuellement dissocier, associer d'autres employés a un secteur d'activités.

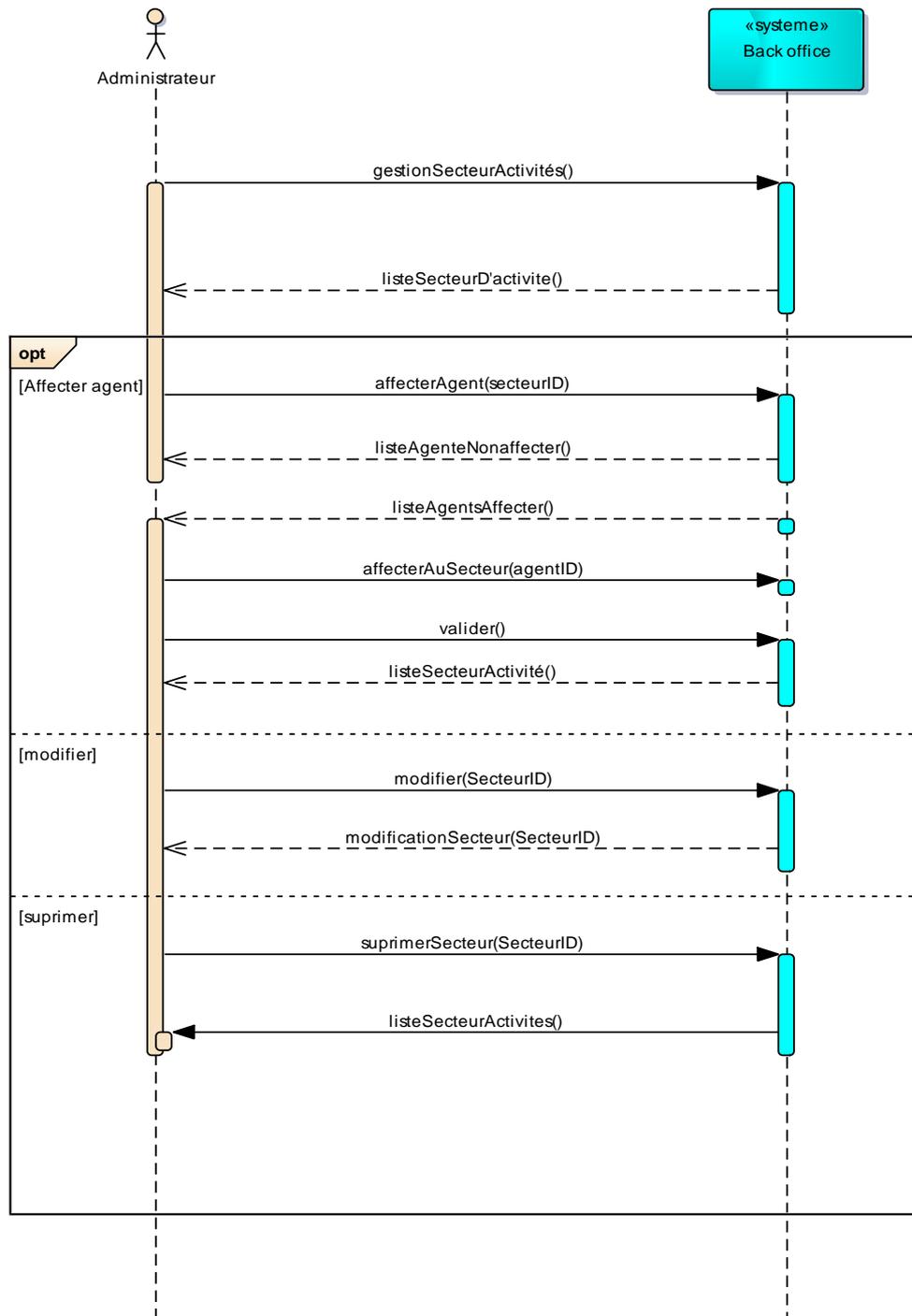


Figure II.15 Diagramme de Séquence « gérer les secteurs d'activité »

m- Diagramme de séquence « fournir la preuve de livraison »

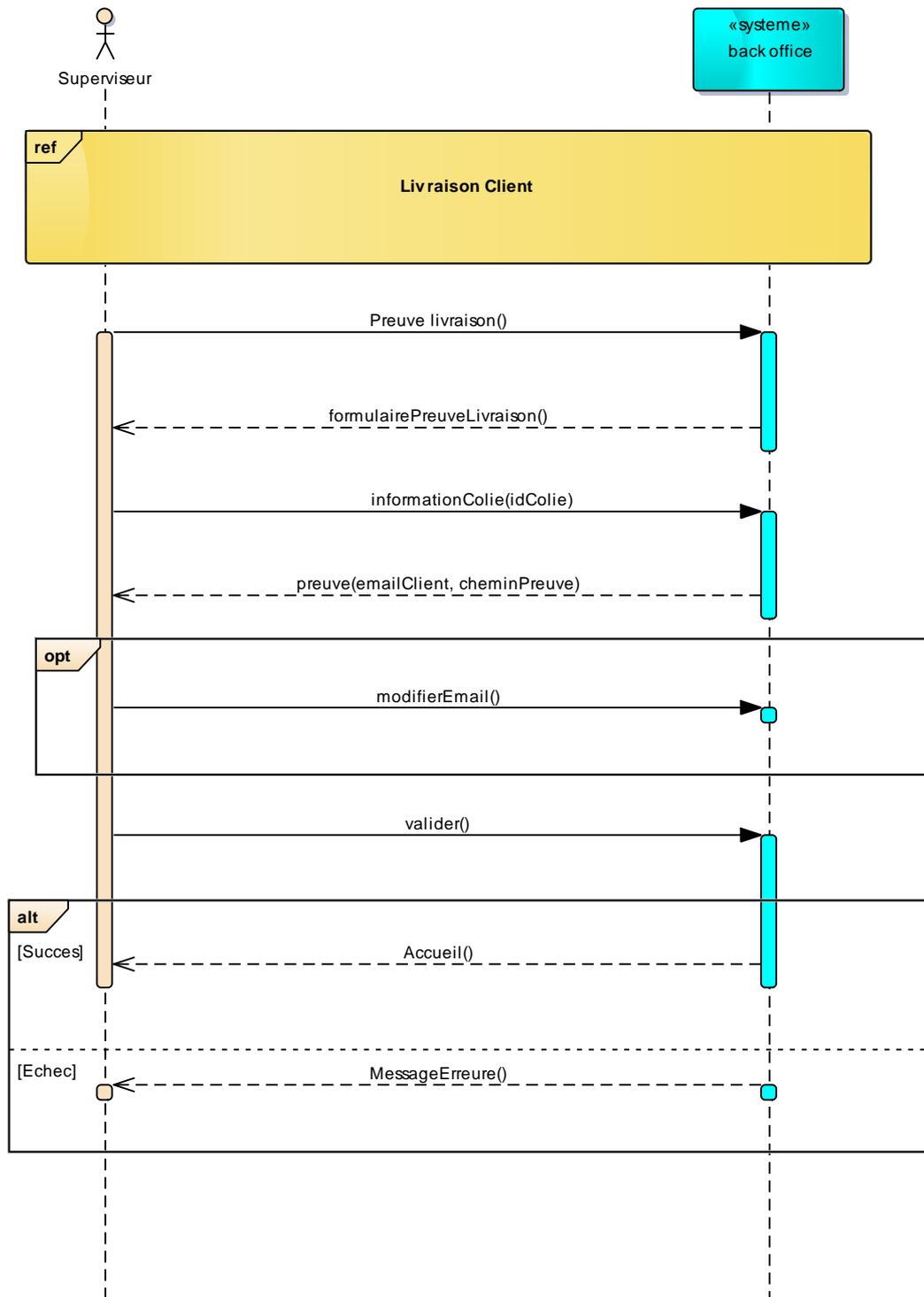


Figure II.16 Diagramme de Séquence « Fournir la preuve de livraison »

Nous avons validé le scénario de preuve de livraison par la maquette suivante :

La maquette présente l'interface utilisateur pour la preuve de livraison. Elle est divisée en deux sections principales :

- Barre de navigation latérale (gauche) :** Contient des menus tels que 'Tableau de bord', 'Tournée du jour', 'Activité Agent', 'Réclamation Client', et 'Preuve de livraison' (surligné en vert).
- Section principale (droite) :**
 - Titre : 'Preuve de livraison'.
 - Champ de saisie 'N° Coli' avec un bouton 'Charger les information' à droite.
 - Champs de saisie pour 'Nom Client', 'Telephone', et 'Email'.
 - Champ de saisie 'Date de livraison'.
 - Champ de saisie 'Image de la Signature'.
 - Deux boutons d'action : 'Annuler' (rouge) et 'Envoyer' (bleu).

Figure II.17Maquette de preuve de livraison

VI- Diagramme de class

La figure II.18 montre le diagramme de class [5] du projet Tracking colis qui représente les données de l'application.

Nous avons adopté une démarche par itérations, sur l'ensemble des diagrammes, l'itération signifie que l'on a fait plusieurs allers retours sur les diagrammes avant d'avoir un dossier d'analyse entièrement satisfaisant.

Après avoir fait une étude sur le diagramme de class déjà réalisé par les ingénieurs de l'entreprise, et en fonction du besoin, on a jugé nécessaire d'effectué les modifications suivantes :

- L'ajout de trois class : « Réclamation », « localisation » et « Preuve »
 - 1- La class Localisation : pour la géolocalisation d'un employé, il s'agit du point fort de notre travail.
 - 2- La class réclamation : quand le client ne reçoit pas l'accusé de réception du colis, il peut faire une réclamation par téléphone.
 - 3- La class Preuve : Quand l'administrateur reçoit une réclamation, il répond à cette dernière par la preuve de livraison qui est une image de la signature du destinataire.

- Modification de certaines classes telles que « suivi actif » en lui rajoutant l'attribut codeBordereau, « secteur_Activite », « localite » l'ajout de l'attribut employeID et la class « Activite »

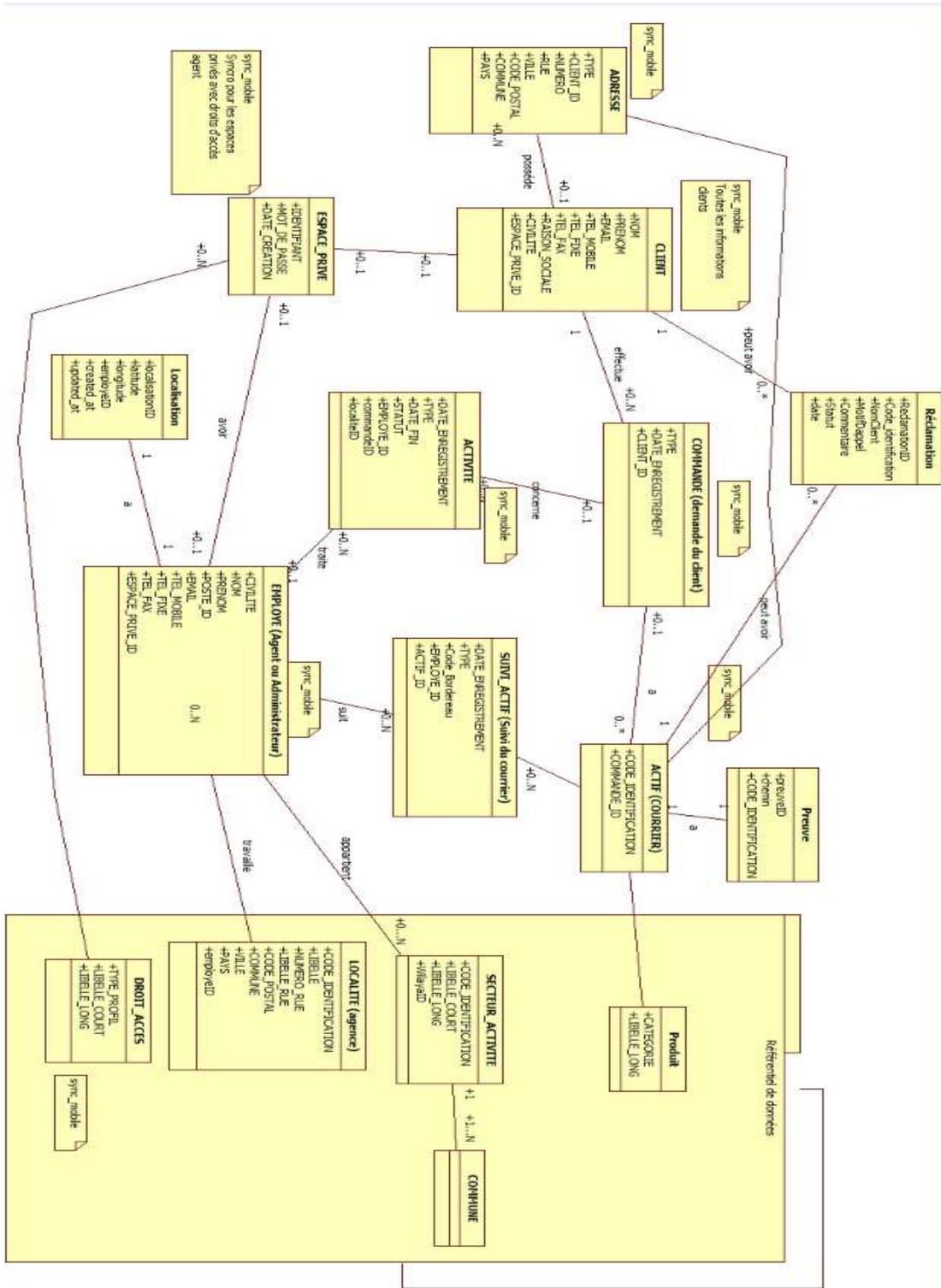


Figure II.18 Le diagramme de class du projet tracking

VII- Conclusion

Dans ce chapitre, nous avons modélisé et conçu le Back-office de tracking de colis tout en appliquant le formalisme du langage Objet UML.

Dans le chapitre suivant, nous présenterons l'environnement de développement, les différents éléments de structures utilisés, pour la réalisation de notre projet, ainsi que la description de l'application conçue d'une manière générale, tout en évoquant ses différentes interfaces.

Rapport-Gratuit.com

I- Gestion du projet

I-1 La méthode Scrum

La méthode Scrum se range sous la bannière d'un mouvement, l'agilité. Elle possède des valeurs et des principes et se met en œuvre avec des pratiques. De ce mouvement novateur émergent les méthodes agiles dont Scrum est actuellement la plus populaire. [6]

Scrum est une méthode pour gérer les projets de façon agile. L'entreprise l'adopte depuis sa création comme méthode de gestion et nous l'avons utilisée dans notre projet pour le développement du « back office traking », parce qu'elle permet d'offrir une meilleure visibilité, une forte inspection et une meilleure adaptation.

Scrum utilise un principe de développement itératif pour cela nous avons découpez le projet en plusieurs étapes que nous appelons « *itérations* » ou « *sprints* ». Ces itérations sont constituées d'un ensemble de sous besoins appelé « **User story** », en détaillant les différentes fonctionnalités qui seront développées. Un planning correspondant aux tâches nécessaires pour le développement de ces fonctionnalités est établi.

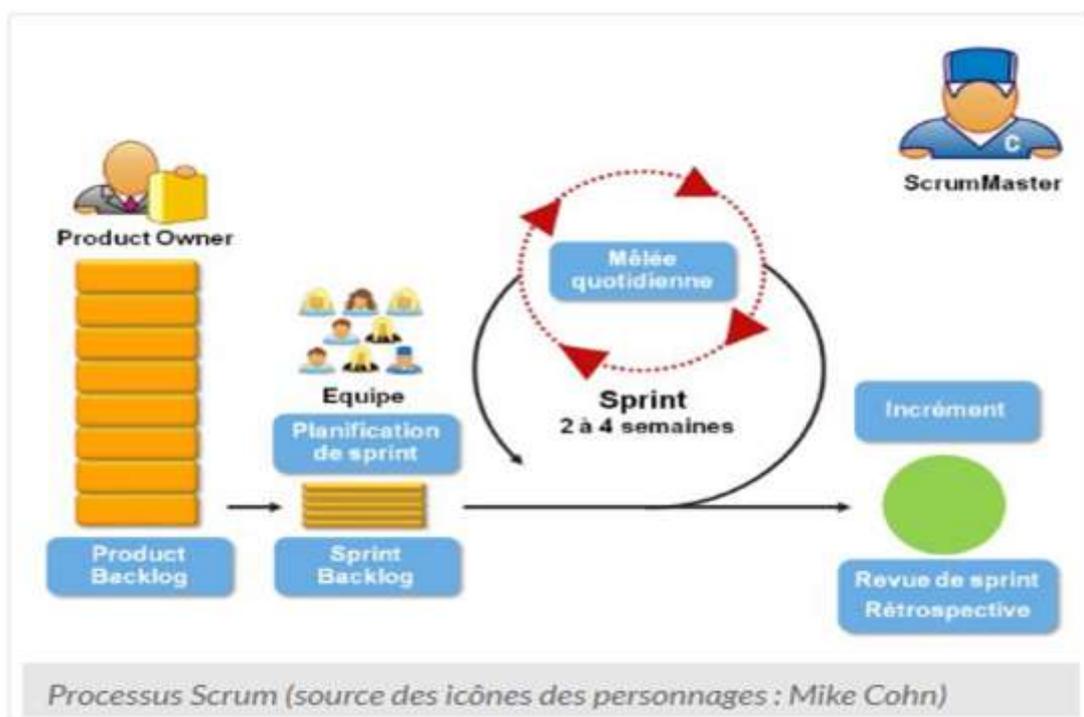


Figure III.1 Cycle de vie Scrum

Scrum se base sur une équipe avec différents rôles :

Product Owner : Le Product Owner définit le produit et priorise les fonctionnalités voulues. Dans notre projet le Product Owner c'est le chef de projet *Downia*

ScrumMaster : C'est un des ingénieurs de l'entreprise, il est chargé d'assister et guider l'équipe de développement pendant le sprint (itération).

Team (équipe de développement) : nous-même et les deux ingénieurs de l'entreprise.

Le cycle de vie Scrum (figure III.1) est rythmé par des itérations (sprints) de deux à quatre semaines. Dans notre cas, nous avons fait 7 itérations de deux semaine chacune.

Ce cycle commence à partir d'un **Product Backlog** qui représente l'ensemble des fonctionnalités ou besoins techniques qui constituent le produit. À partir de cet ensemble nous sélectionnons les exigences les plus prioritaires qui seront développées « **User Story** ».

La figure montre l'ensemble des tâches associées à l'équipe de développement avec le nombre de jours restant pour la réalisation de la fonctionnalité. Pendant un sprint, des points de contrôle sur le déroulement des travaux sont effectués, pour cela chaque jour une réunion est organisée avec tous les membres de l'équipe. Ce que nous appelons par la suite « **Daily Scrum Meeting** » ou « **Daily Standup Meeting** ». [7]

Dans notre cas, nous avons fait pour chaque jour une réunion à 9h pour dresser le bilan de la journée précédente. Nous planifions celles qui commencent et nous repérons les éventuels obstacles rencontrés par chaque membre de l'équipe. Cela permet au *ScrumMaster* de déterminer l'avancement en modifiant le fichier de suivi (la figure III.2) et d'appliquer avec l'équipe des ajustements pour assurer le succès du sprint. À la fin du sprint, une réunion est faite, « **Sprint Review Meeting** » pour montrer le travail accompli et donc l'équipe obtient un livrable qui fonctionne et sera potentiellement utilisable et qui s'enrichit d'un nouveau incrément à chaque sprint. Son évaluation et le feedback récolté permettent d'ajuster le **Backlog** pour le sprint précédent. [8]

b- Fichier de suivi du projet

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
Tâche	Description	Priority	Assigned	Assignee	Charge totale	CONVERSION	REALISATION	VALIDATION	SUPPORT	RAF-CONCEP	RAF-REALIS	RAF-VALER	RAF-SUPPORT							
1	Accueil_Gestion_Agence	EC-RC-Gestion-Agence	LOPP	Lotfi	6,00	3,75	6,75	3,00	1,50	3,8	0,0	3,0	1,5							
2	Gestion_Agence	EC-Gestion-Agence	LOPP		8,00	2,00	3,60	1,80	0,80	2,8	0,0	1,8	0,8							
3	Modification_Agence	EC-Modif-Agence	LOPP		5,00	1,25	2,25	1,00	0,50	1,3	0,0	1,0	0,5							
4	Confirmation_Suppression_Agence	POSRP-Confirmation-Supp	LOPP		5,00	1,25	2,25	1,00	0,50	1,3	0,0	1,0	0,5							
5	Abolition_Suppression_Agence	POSRP-Ab-Suppr-Agence	LOPP		8,00	2,00	3,60	1,80	0,80	2,8	0,0	1,8	0,8							
6	Accueil_Gestion_Employe	EC-RC-Gestion-Employe	NOIR		6,00	3,75	6,75	3,00	1,50	3,8	0,0	3,0	1,5							
7	Gestion_Employe	EC-Gestion-Employe	NOIR		8,00	2,00	3,60	1,80	0,80	2,8	0,0	1,8	0,8							
8	Modification_Employe	EC-Modif-Employe	NOIR		5,00	1,25	2,25	1,00	0,50	1,3	0,0	1,0	0,5							
9	Confirmation_Suppression_Employe	POSRP-Confirmation-Supp	NOIR		3,00	0,75	1,35	0,60	0,30	0,8	0,0	0,6	0,3							
10	Abolition_Section_Employe	POSRP-Ab-Secteur-Employe	NOIR		8,00	2,00	3,60	1,80	0,80	2,8	0,0	1,8	0,8							
11	Accueil_Gestion_Client	EC-RC-Gestion-Client	VOIRSERP		6,00	3,75	6,75	3,00	1,50	3,8	0,0	3,0	1,5							
12	Gestion_Client	EC-Gestion-Client	VOIRSERP		8,00	2,00	3,60	1,80	0,80	2,8	0,0	1,8	0,8							
13	Modification_Client	EC-Modif-Client	VOIRSERP		5,00	1,25	2,25	1,00	0,50	1,3	0,0	1,0	0,5							
14	Suppression_Client	POSRP-Confirmation-Supp	VOIRSERP		3,00	0,75	1,35	0,60	0,30	0,8	0,0	0,6	0,3							
15	Accueil_Gestion_Secteur	EC-RC-Gestion-Secteur	AMHE		6,00	3,75	6,75	3,00	1,50	3,8	0,0	3,0	1,5							
16	Gestion_Secteur	EC-Gestion-Secteur	AMHE		8,00	2,00	3,60	1,80	0,80	2,8	0,0	1,8	0,8							
17	Modification_Secteur	EC-Modif-Secteur	AMHE		5,00	1,25	2,25	1,00	0,50	1,3	0,0	1,0	0,5							
18	Abolition_Secteur_Agent	POSRP-Ab-Secteur-Agent	AMHE		10,00	2,50	4,50	2,00	1,00	2,5	0,0	2,0	1,0							
19	Suppression_Secteur	POSRP-Confirmation-Supp	AMHE		3,00	0,75	1,35	0,60	0,30	0,8	0,0	0,6	0,3							
20	Accueil_Activite_Agent	EC-RC-Activite-Agent	AMHE		6,00	3,75	6,75	3,00	1,50	3,8	0,5	3,0	1,5							
21	Abolition_Activite_Agent	POSRP-Ab-Activite-Agent	AMHE		8,00	2,00	3,60	1,80	0,80	2,8	0,0	1,8	0,8							
22	Confirmation_Activite_Agent	POSRP-Confirmation-Act	AMHE		4,00	1,00	1,60	0,80	0,40	1,6	0,0	0,8	0,4							
23	Modification_Activite_Agent	EC-Modif-Activite-Agent	AMHE		5,00	1,25	2,25	1,00	0,50	1,3	0,0	1,0	0,5							
24	Suppression_Activite_Agent	POSRP-Confirmation	AMHE		5,00	1,25	2,25	1,00	0,50	1,3	0,0	1,0	0,5							
25	Globalisation	EC-Globalisation	LOPP		20,00	5,00	9,00	4,00	2,00	9,0	0,0	4,0	2,0							
26	Accueil_Saisie_Livraison	EC-RC-Saisie-Livraison	NOIR		6,00	3,75	6,75	3,00	1,50	3,8	0,0	3,0	1,5							
27	Ligne_Activite_Adresse	POSRP-Ligne-Adresse	NOIR		8,00	2,00	3,60	1,80	0,80	2,8	0,0	1,8	0,8							
28	Accueil_Reclamation_Client	EC-RC-Reclamation	Reu		6,00	3,75	6,75	3,00	1,50	3,8	0,0	3,0	1,5							
29	Appareil_Reclamation_Client	EC-Appareil-Reclamation	Reu		8,00	2,00	3,60	1,80	0,80	2,8	0,0	1,8	0,8							
30	Preuve_Livraison	EC-Preuve-Livraison	LOPP		8,00	2,00	3,60	1,80	0,80	2,8	0,0	1,8	0,8							
31	Accueil_Rapport	EC-RC-Rapport	VOIRSERP		8,00	2,00	3,60	1,80	0,80	2,8	3,8	1,8	0,8							

Figure III.2 Fichier de suivi du Back office (projet tracking colis)

Prenons par exemple la première ligne de ce tableau qui indique que la fonctionnalité « accueil Gestion Agence » doit être faite par Lotfi et qui lui reste 6.75 jour pour la réaliser.

Pour une meilleur vision du suivi on génère un graphe (Burn Down chart) à partir du tableau de la figure III.20 qui représente les taches en fonction des nombre de jours, ce graph se constitue de deux courbe celle qui est représenté avec des petite ligne discontinue indique comment l'avancement doit être et l'autre courbe indique l'avancement réel de l'équipe du développement, la figure III.2 montre l'avancement du 14 avril 2016 et à partir de ce graphe on conclue que l'équipe avance bien plus vite que prévu.

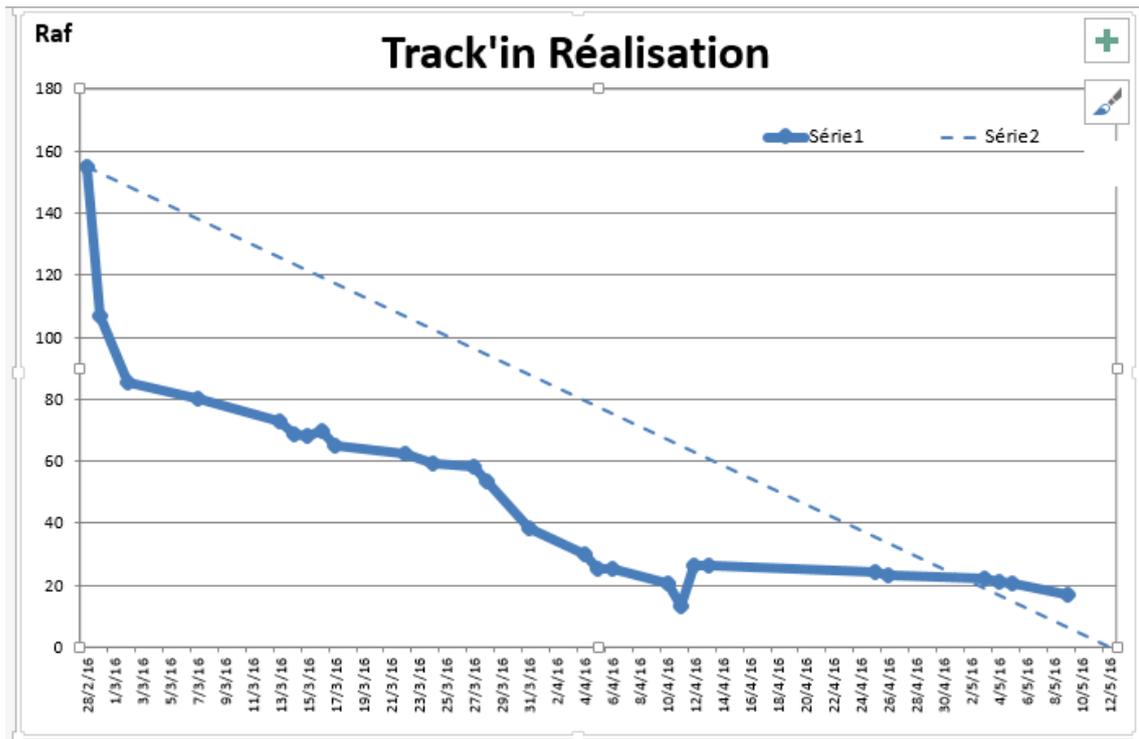


Figure III.3 Graphe généré par le fichier de suivi du projet tracking colis

I-3 Diagramme de Gantt

Le graphe exporté par le fichier de suivi nous a permis de gérer la phase de développement du Back office qui a commencé le 28/02/2016 jusqu'au 28/04/2016.

Au début du stage nous avons fixé une date de début et une date de fin, pour une meilleure gestion de notre projet, nous avons exploité « le diagramme de Gantt » [9], un outil très populaire dans le monde de la gestion de projet permettant de visualiser plus facilement la durée totale du projet, ainsi que de suivre l'état d'avancement des tâches du projet.

Pour la création du diagramme de Gantt, nous avons utilisé « Gantt Project »³[10], un logiciel libre, permettant de modéliser sous forme d'un diagramme de Gantt la planification des différentes tâches qui sont nécessaires à un projet.

Le diagramme de Gantt relatif à notre projet est représenté dans la figure III.4 :

³<https://www.ganttproject.biz/download>



Figure III.4 Diagramme de Gantt du Back office

II- L'environnement de développement

Pour la réalisation de notre application, nous avons utilisé les langages de programmation suivants : Le Framework PHP Laravel, JavaScript (jQuery), HTML 5 & CSS 3 et comme environnement de travail nous avons opté pour l'utilisation de Sublime Text 3, éditeur de texte spécialisé dans les langages web essentiellement en PHP, qui permet de développer de manière plus efficace et plus propre.



II-1 Le Framework Laravel

Vu que nous avons dû utiliser le Framework [11] Laravel, il est intéressant de noter que nous avons commencé notre stage par une formation de plus de trois semaines sur ce Framework. C'est ce qui nous a également permis de perfectionner nos connaissances sur le PHP objet. Par ailleurs ce Framework nous a aidé à avoir un code plus structuré, plus clair et mieux organisé, grâce aux diverses fonctionnalités qu'il propose telles que :

- Un système de routage perfectionné (RESTful et ressources),
- Un créateur de requêtes SQL et un ORM performants,
- Un moteur de Template efficace,
- Un système de validation,
- Un système de pagination,

- Un système de migration pour les bases de données,
- Une gestion des sessions...

Nous avons exploité la version actuelle de Laravel « 5.2 » qui nécessite au minimum la version « 5.5.9 » de PHP.

II-2 Installation et mise en marche du Framework

Laravel utilise Composer comme gestionnaire de dépendances. Nous commençons par le téléchargement de ce dernier de ce dernier, puis nous lançons l'invité de commande et procédons à l'installation avec la commande suivante :

```
1 composer create-project laravel/laravel laravel5 --prefer-dist
```

a- Organisation de Laravel :

Avant de débiter nous avons commencé par comprendre l'utilité de chacun de ces dossiers proposés par laravel donc commençons par le dossier « app » :

Ce répertoire contient les éléments essentiels de l'application :



Figure III.5 Architecture de base de laravel

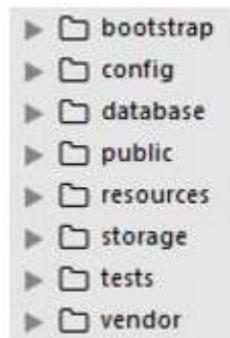
- **Http** : contient tout ce qui concerne la communication : contrôleurs, routes, middlewares (il y a 4 middlewares de base) et requêtes,
- **Providers** : contient tous les providers, il y en a déjà 4 au départ. Les providers servent à initialiser les composants.

On trouve également le fichier `activif.php`, `activité.php` ou `category.php` et tt les autres classes de notre projet.

On y trouve aussi d'autres dossiers tel que :

Figure III.6 Dossier laravel

- **bootstrap** : scripts d'initialisation de Laravel pour le chargement automatique des classes, la fixation de l'environnement et des chemins, et pour le démarrage de l'application,



- **public** : tout ce qui doit apparaître dans le dossier public du site : images, CSS, scripts...
- **vendor** : tous les composants de Laravel et de ses dépendances,
- **config** : toutes les configurations : application, authentification, cache, base de données, espaces de noms, emails, systèmes de fichier, session...
- **database** : pour les migrations et les populations,
- **resources** : pour les vues, les fichiers de langage et les assets (par exemple les fichiers LESS ou Sass),
- **storage** : pour stocker les données temporaires de l'application : vues compilées, caches, clés de session...

Il y a aussi des fichiers dans la racine dont voici les principaux :

- **artisan** : outil en ligne de Laravel pour des tâches de gestion,
- **composer.json** : fichier de référence de Composer,
- **phpunit.xml** : fichier de configuration de phpunit (pour les tests unitaires),
- **.env** : fichier pour spécifier l'environnement d'exécution la figure ci-dessous indique le contenu de notre fichier `.env`. [12]

```

1 APP_ENV=local
2 APP_DEBUG=true
3 APP_KEY=A8UhME3dbOyFPYWgSazh5I7Gg4TMbnhI
4
5 DB_HOST=localhost
6 DB_DATABASE=trakin-v2
7 DB_USERNAME=root
8 DB_PASSWORD=
9
10 CACHE_DRIVER=file
11 SESSION_DRIVER=file
12 QUEUE_DRIVER=sync
13
14 REDIS_HOST=localhost
15 REDIS_PASSWORD=null
16 REDIS_PORT=6379
17
18 MAIL_DRIVER=smtp
19 MAIL_HOST=smtp-mail.outlook.com
20 MAIL_PORT=587
21 MAIL_USERNAME=alanelotfi@outlook.com

```

Figure III.7 Connexion à la base de données

III- Le système de gestion de base de données « My SQL »

L'accès à une base de données par des utilisateurs ou des applications passe indirectement par un système connu par le système de gestion de base de données (SGBD).

Pour le stockage et la gestion des flux d'informations nous avons choisi le système de gestion de base de données MySQL.

```

--
-- Structure de la table `localisation`
--
CREATE TABLE IF NOT EXISTS `localisation` (
  `localisationID` int(11) NOT NULL AUTO_INCREMENT,
  `latitude` double NOT NULL,
  `longitude` double NOT NULL,
  `nom` varchar(255) NOT NULL,
  `employeID` int(11) NOT NULL,
  `created_at` timestamp NOT NULL,
  `updated_at` timestamp NOT NULL,
  PRIMARY KEY (`localisationID`),
  KEY `clientID` (`employeID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;

--
-- Contenu de la table `localisation`
--

INSERT INTO `localisation` (`localisationID`, `latitude`, `longitude`, `nom`, `employeID`, `created_at`, `updated_at`) VALUES
(1, 53.4008, -2.2374, 'Lotfi', 5, '2016-02-26 11:45:00', '2016-04-12 09:25:00'),
(2, 53.38112809999999, -1.4700650000000004, 'Oussama', 6, '2016-03-28 10:45:00', '2016-03-26 11:45:00'),
(3, 53.4167, -3, 'Nour', 6, '2016-03-28 10:45:00', '2016-03-30 08:01:00');

```

Figure III.8la table localisation

Nous avons utilisé un serveur équipé de PHP avec au minimum la version 5.5.9 et aussi de MySQL. **Wampserver**, qui répond à toutes nos attentes et qui permet de basculer entre les versions de PHP et de MySQL en un simple clic.

La figure ci-dessus montre le script de la table « localisation » , avec quelques enregistrements.

IV- Tests et Intégration

Les tests sont nécessaires pour détecter le plus rapidement possible les problèmes logiciels ou les bugs de manière à pouvoir les corriger avant la publication et le déploiement de l'application. Ils sont également réalisés pour développer la confiance et la connaissance du produit livré.

L'un des objectifs principale de notre stage de fin d'étude c'est le travail en équipe, donc dans un court premier temps on a formé une équipe à quatre composée de deux ingénieurs et nous même, ou chacun avait des tâches à accomplir et à l'issue de chaque tâche réalisée, nous effectuons des tests en locale par rapport à ce qui a été définit dans la conception puis nous mettons à jours le projet déjà partagé sur le *Git hub*[13] afin que tous les membres de l'équipe puisse avoir accès à la dernière version, par la suite on a continué à deux en gardant la même méthodologie de travail.

V- Démonstration

V-1 Authentification

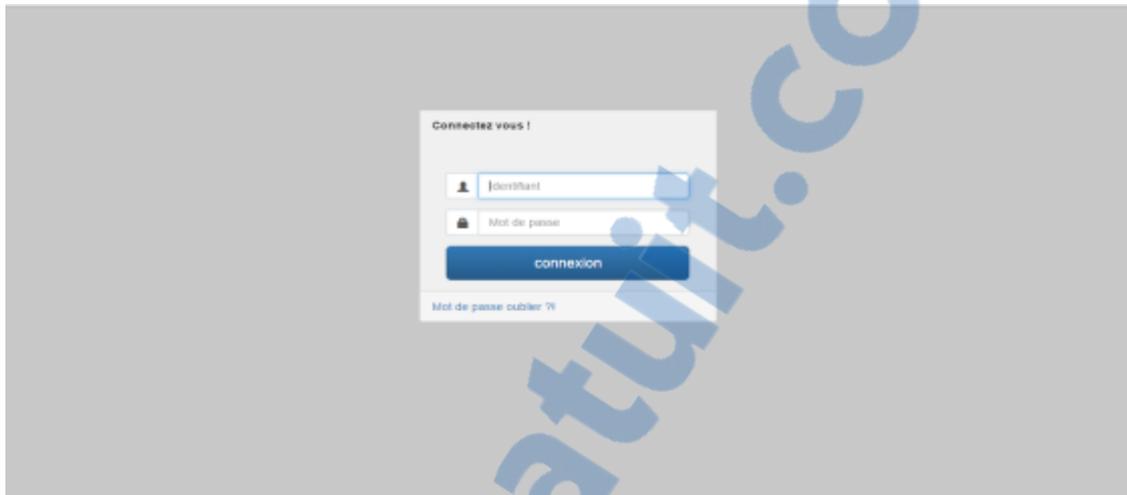


Figure III.9 Authentification

L'accès au back office est protégé par un système d'authentification avec deux types d'utilisateur un administrateur et un superviseur.

V-2 Menu Administrateur



Figure III.10 Tableau de bord Administrateur

Ce tableau de bord permet à l'administrateur d'avoir accès à toute les fonctionnalité du back office (gestion des employes, clients, localités ..) et d'avoir un résumé des fonctionnalités importante sous forme de vignette comme le montre la figure III.10avec un accès rapide a ces derniers.

V-3 Gestion des employés

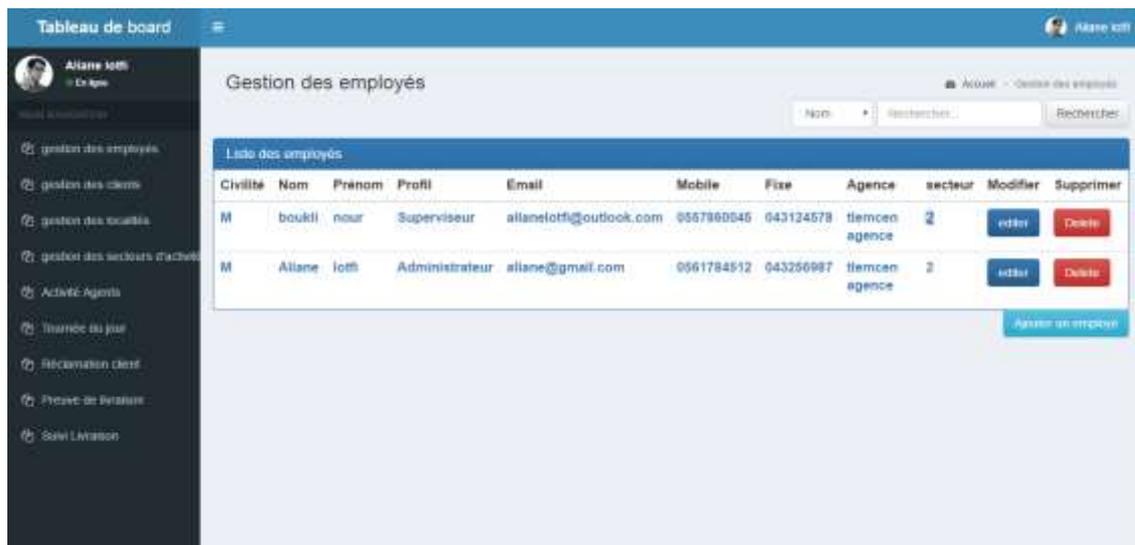


Figure III.11 Gestion des employés

Cette partie permet à l’administrateur de voir l’ensemble des employés avec la possibilité de modifier, supprimer ou de lister les différents secteurs d’activité de chaque employé avec aussi la possibilité de filtrer cette liste par nom, prénom ou poste.

V-4 Gestion des localités

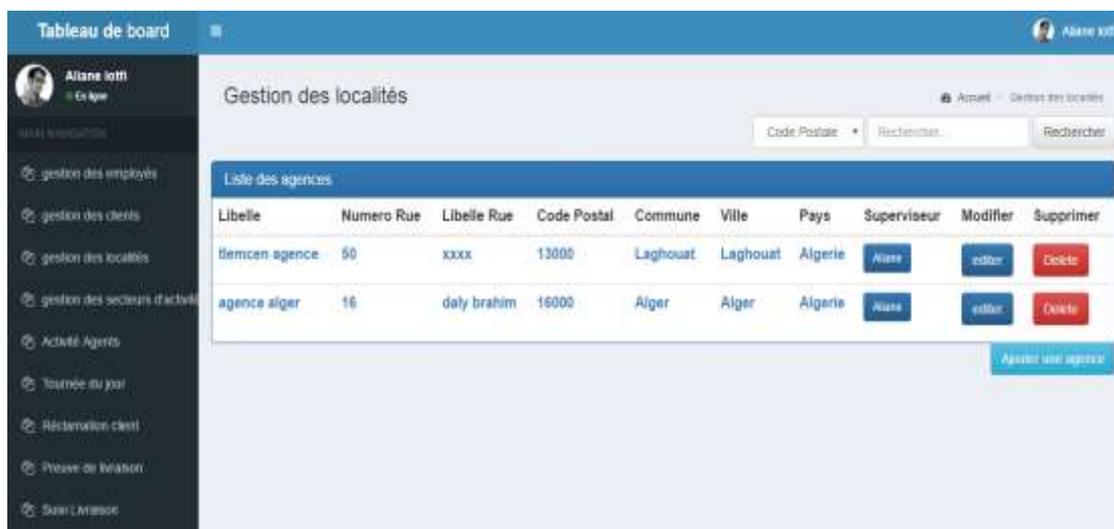


Figure III.12 Gestion des localités

Ce menu donne la possibilité à l’administrateur de lister les différentes localités et de pouvoir ajouter, modifier, supprimer et ajouter une localité et aussi la possibilité d’associer un superviseur à une agence comme l’indique la figure ci-dessous.

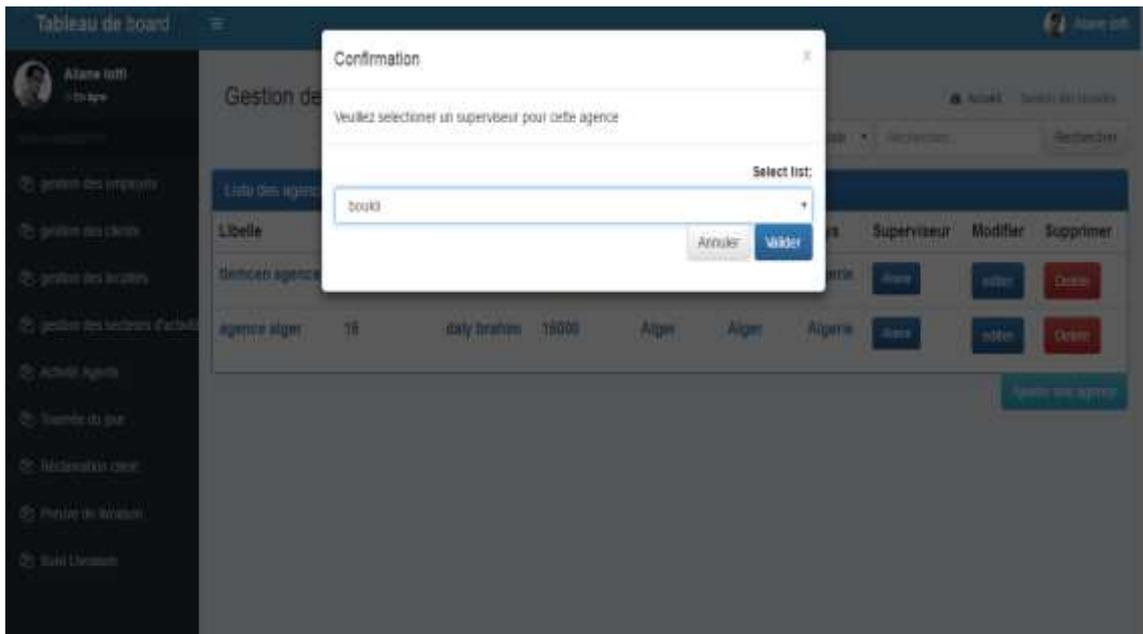


Figure III.13Affectation d'un superviseur

Le code ci-dessous détaille les différentes étapes du développement de la gestion des localités.

```

1 <?php
2
3
4 namespace App;
5
6 use Illuminate\Database\Eloquent\Model;
7
8 class localite extends Model
9 {
10     //
11     public $timestamps = false;
12     public $table = 'localite';
13     protected $primaryKey = 'localiteID';
14     public $fillable = ['libelleLocalite', 'numrue', 'libellerue', 'codepostale', 'commune', 'ville', 'pays'];
15     public static $rules = array(
16         'libelleLocalite' => 'required|min:4',
17         'codepostale' => 'required|numeric',
18         'numrue' => 'required|numeric',
19         'pays' => 'required|min:2');
20     public function getKeyname(){
21
22         return "localiteID";
23     }
24
25     public function employe()
26     {
27         return $this->belongsTo('App\employe', 'employeID');
28     }
29 }
30

```

Figure III.14 Model localité

Comme nous l'avons indiqué au deuxième chapitre nous avons utilisé l'architecture Model-view-Controller donc les model associer à ce menu sont localité et employé

```

1 <?php
2
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class employe extends Model
8 {
9     //
10    public $timestamps = false;
11
12    // le nom de la table associée à la base de données
13    protected $table = 'employe';
14    // le nom de la clé primaire
15    protected $primaryKey = 'employeeID';
16    // les attributs
17    protected $fillable = ['civilite', 'nom', 'prenom', 'poste', 'email', 'tel', 'telex', 'tefax', 'localiteID', 'espacePriveID'];
18
19
20    public function getKeyname()
21    {
22        return "employeeID";
23    }
24
25    // la mise en relation avec les différentes table
26    public function espace_prive()
27    {
28        return $this->belongsTo('App\espace_prive');
29    }
30
31    public function secteur()
32    {
33        return $this->belongsToMany('App\secteur', 'employe_secteur', 'employeeID', 'secteurID');
34    }
35
36    // relation à une seule localité
37    public function localite()
38    {
39        return $this->hasOne('App\localite', 'employeeID');
40    }
41
42    // pour indiquer qu'un employé a une seule localisation
43    public function localisation()
44    {
45        return $this->hasOne('App\localisation', 'employeeID', 'employeeID');
46    }
47
48 }

```

Figure III.15 Model employé

Dans un model on y trouve l'ensemble des attributs ainsi que les différentes relation avec les autres model et éventuellement d'autres fonctions.

```

1 <?php
2
3 namespace App\Http\Controllers;
4 use Illuminate\Http\Request;
5 use App\localite;
6 use App\wilaya;
7 use App\commune;
8 use App\employe;
9 use Illuminate\Support\Facades\Validator;
10 use Illuminate\Support\Facades\Redirect;
11 use Illuminate\Support\Facades\Session;
12 use App\Http\Requests;
13 use Input;
14 use App\Http\Controllers\Controller;
15 use App\Http\Requests\localiteRequest;
16
17 //l'appelle des différent package, model, bibliothèque utilisé dans ce controller
18
19 class localiteController extends Controller
20 {
21     //
22     //au moment du chargement du menu gestion des localité
23     public function index()
24     {
25         $localites = Localite::with('employe')->paginate(5);
26         //on récupère les localité avec superviseur associé
27         $employees_employe = where('poste', '=', 'Superviseur')->get();
28
29         return view('Localite.index', compact('localites', 'employees'));
30         // on envoie le tout à la vue pour pouvoir les afficher
31     }
32
33     //cette fonction se déclenche au moment où on click sur ajouter une nouvelle localité
34     public function create()
35     {
36         $localite = new Localite();
37
38         $wilayas = wilaya::all();
39         //on récupère tte les wilaya
40         $communes = new commune();
41         // toutes les commune
42         return view('Localite.create', compact('localite', 'wilayas', 'communes'));
43         //on envoie le tout à la vue qui un formulaire d'ajout
44     }
45 }

```

Figure III.16 Controller localité

Le Controller met en relation le model et la vue la première fonction index démontre comment on a récupérer l'ensemble des localités et employés ayant un statut de superviseur (rectangle n°1) et l'envoyé au niveau de la vue afin de les affichez (flèche n°1).

V-5 Gestion des secteurs d'activités

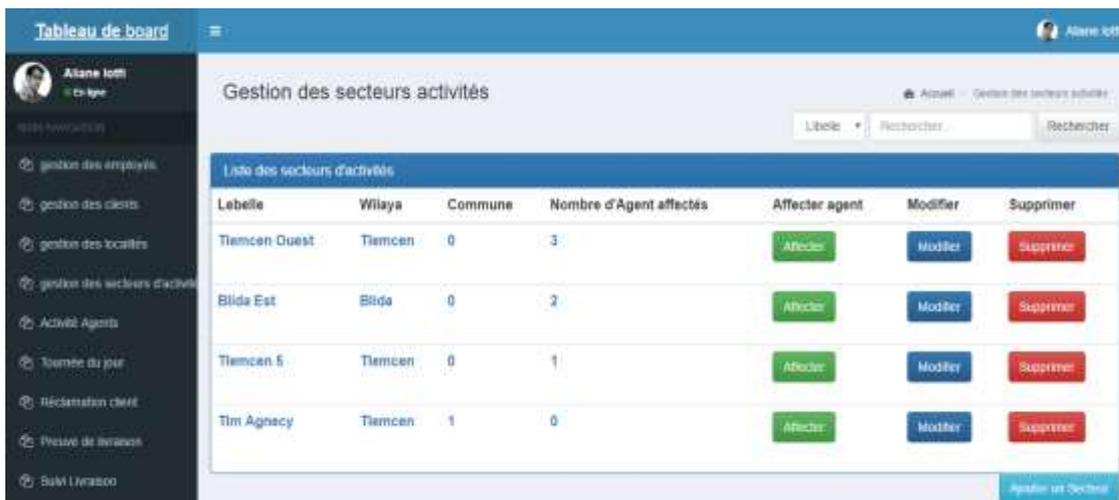


Figure III.17 Liste des secteurs d'activités

Cet écran lui offre la possibilité de gérer les différents secteurs d'activités avec la possibilité d'affecter de nouveaux agents à un secteur d'activité comme le montre la figure suivante.

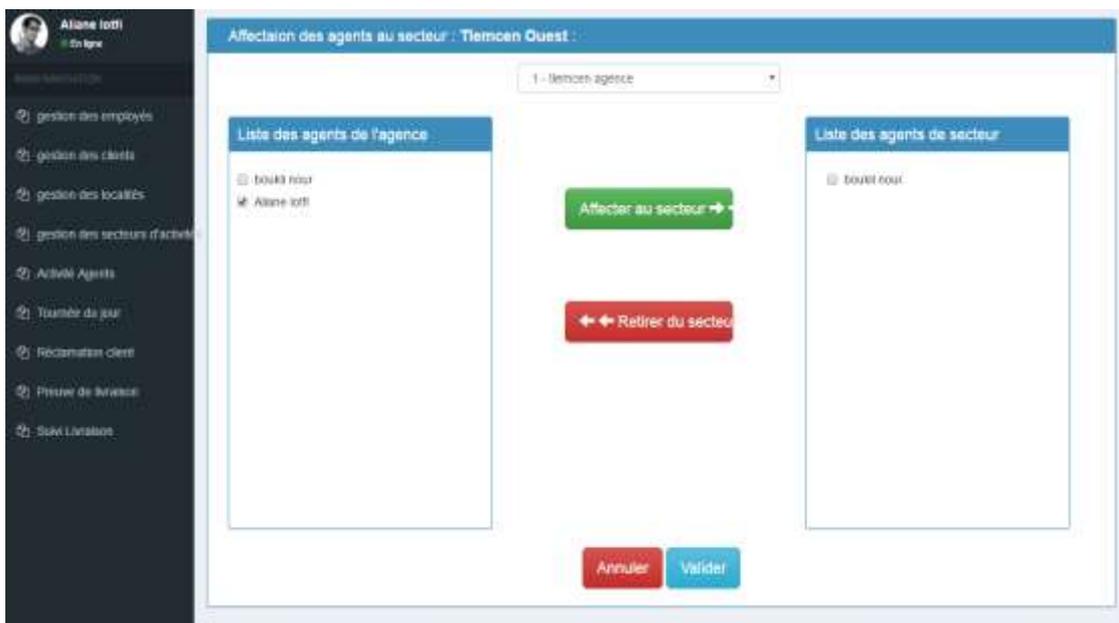


Figure III.18 Affectation des agents au secteur

A travers cette fenêtre nous avons mis en place un système de filtre d'agents par agence puis par secteur, ce qui offre à l'utilisateur une facilité de recherche.

V-6 Géolocalisation

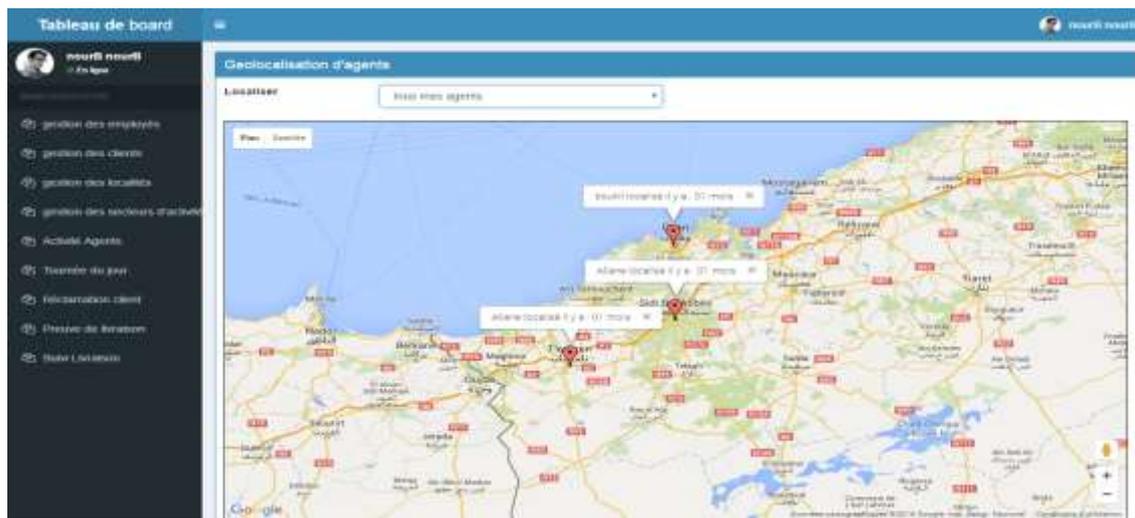


Figure III.19 Géolocalisation d’agents

La figure III.16 montre un exemple de géolocalisation des employés, le superviseur a la possibilité de localiser les différents agents de terrain en temps réel qu’il supervise dans une Mapp avec le nom et la date de la dernière localisation de l’agent comme il peut aussi localiser chaque agent séparément, on peut aussi basculer sur un autre mode d’affichage satellite ou en relief.

- **Système de géolocalisation mis en place**

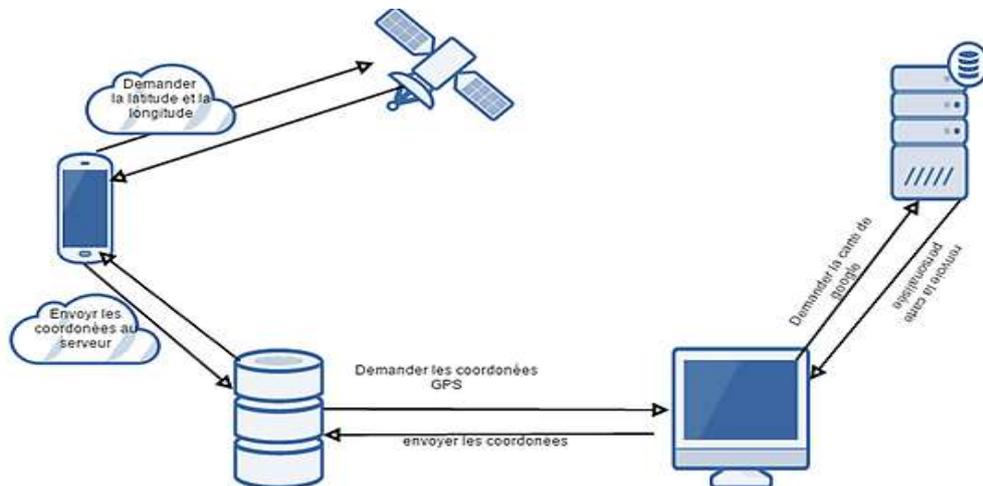


Figure III.20 Schéma du système de géolocalisation

La solution mobile envoie périodiquement et au déclenchement de chaque activité les coordonnées GPS au serveur en récupérant la latitude et longitude en fonction de sa position par rapport au satellite, ensuite au niveau du back office on récupère les coordonnées stocker au niveau du serveur, ce bout de code du Controller détails la récupération de la latitude et longitude du serveurs avec l'employé associé (flèche n°2) et l'envoyé au niveau de la vue (flèche n°3) pour pouvoir les exploiter pour l'affichage.

```

$coordone Localisation :all()->last();

$employes employe: where('localiteID','=', $session: get('SlocaliteID'))->get(); //recuperation des employés qui font partie de la meme agence du superviseur
$localiteID $session: get('SlocaliteID'); //recupération de ID du secteur du superviseur qui est stocké dans une session
// Add information window for each address
$collection = Localisation: with('employe')->whereHas('employe',function($q){ //recupération de la liste des coordonnées GPS du serveur chacune associée à son employé
    return $q->where('localiteID','=', $session: get('SlocaliteID')); // qui fait partie de la meme agence du superviseur
})->get();

return view('geolocalisation.index',compact('collection','coordone','employes')); // envoi de l'ensemble des données afin de pouvoir les afficher en exploitant le

```

Figure III.21 géolocalisation controller

Afin de pouvoir indiquer la position de chaque employé nous avons utilisé l'API de Google pour l'affichage de la Mapp avec les différents marqueurs indiquant la position exacte des différents agents de terrain, le code ci-dessous montre comment nous avons pu exploiter le Google map API et avoir un affichage qui correspond aux besoins établis au début du stage.

```

41 <script src="http://maps.googleapis.com/maps/api/js?key=AIzaSyCABw7e2hVW-Tk4KRbdwZrj9GEB19umqBH"></script>
42 <!-- l'appel du googleMap API avec la cle d'activation -->
43 <script>
44
45
46 // initialisation
47 fonction initialize() {
48     var proprieti_peta = {
49
50         center: new google.maps.LatLng([! $coordone['latitude'] !]), [! $coordone['longitude'] !]),
51         //on initialise le positionnement de la map sur les dernier agents localiser
52         zoom:8,
53         //on y applique un zoom de 8 afin d'avoir une vision précise et assez large
54         mapTypeId: google.maps.MapTypeId.ROADMAP
55     };
56
57
58     var peta = new google.maps.Map(document.getElementById("tempat_peta"), proprieti_peta);
59     //on indique qu'on veut afficher la map au niveau du div ayant comme id "tempat_peta"
60     @foreach ($collection as $coordinate)
61         //on parcourt tout les coordoné récupérer
62         var marker= new google.maps.Marker({
63             position: new google.maps.LatLng([! $coordinate['latitude'] !]), [! $coordinate['longitude'] !]),
64             // pour chaque coordoné on utilise un marker
65         });
66
67
68
69 <?php
70     $tz = new DateTimeZone('Africa/Algiers');
71

```

Figure III.22 géolocalisation scripte

La figure ci-dessus montre les scriptes de l'appel de l'API Google AP (flèche n°1) et l'initialisation de la map sur les derniers agents localisés avec un zoom de 8 pour avoir une vision précise et assez large sur la map.

```

70     $tz = new DateTimeZone('Africa/Algiers');
71
72     $date = new DateTime('now', $tz);
73     //on recupere l'horaire actuel
74     $lancement = new DateTime($coordinate->updated_at);
75     //on recupere la date de la dernière localisation de cet employé
76     $d = $date->diff($lancement)->format('localisé il y a : %M mois');
77     $color="red";
78
79     if ($d == "localisé il y a : 00 mois"){
80
81         $d = $date->diff($lancement)->format('localisé il y a : %D jours');
82         if ($d == "localisé il y a : 00 jours"){
83
84             $d = $date->diff($lancement)->format('localisé il y a : %H Heurs');
85             if ($d == "localisé il y a : 00 Heurs"){
86
87                 $d = $date->diff($lancement)->format('localisé il y a : %i minute');
88                 $color="green";
89             }
90         }
91     }
92
93 }
94 //on compare entre la date de la dernière localisation la date actuel et on affiche
95 //à quel moment il a été localiser pour la dernière fois
96
97
98
99 <?

```

Figure III.23 géolocalisation scripte

La figure montre comment nous avons calculé l'horaire exact de la localisation de l'agent de terrain.

```

100 marker.setIcon('http://maps.google.com/mapfiles/ms/icons/!!$color!!-dot.png');
101 //on choisie le type et la couleur de l'icone du marker
102 marker.setMap(peta);
103 //on designe sur quel map on veut associe ce marker dans ce cas c'est la sup ou on e appele peta
104 @if (count($coordinate->employe) != 0){
105
106     var informaci = new google.maps.InfoWindow({
107         content: "!!$coordinate->employe->nom!! {!!$d!!} "
108         //on affiche le nom de l'employe dans un petit rectangle dedié au information
109     });
110     informaci.open(peta, marker);
111 }
112 @endif
113 @endforeach
114 }
115
116
117
118
119

```

Figure III.24 géolocalisation scripte

Ensuite nous avons marqué les différentes positions des agents de terrain sur la map avec un marker de couleur verte s'il a été localisé il y a moins d'une heure sinon en rouge (rectangle1) et indique le nom de chaque employés dans un petit rectangle qui sera positionné au-dessus du marker

```

<div class="box box-primary box-solid">
  <div class="box-header with-border">
    <h3 class="box-title">Geolocalisation d'agents </h3>
  </div><!-- /.box-header -->
  <div class="box-body">
    <div class="row">
      {!! Form::label('Employe', 'Localiser', ['class' => 'col-md-2 control-label block ']) !!}

      <div class="col-lg-4">
        <select class="form-control block" name="employe" id="employe" >
          <option value="all" >tous mes agents</option>
          @foreach($employes as $employe)
          <option value="{{ $employe->employeID }}" >{!! $employe->nom !!</option>
          @endforeach
        </select>
        <!-- on ajoute tt les employe dans une liste afin de pouvoir localisé l'ensemble des
        employe ou bien un employe précise | -->
      </div>
    </div>
    <br />
    <div class="embed-responsive embed-responsive-16by9">
      <!-- on affiche la map a l'interieur d'une div responsive -->
      <div id="tempat_peta" class="embed-responsive-item" style="border: 1px solid black"></div>
    </div>
  </div>
</div>
</div>

```

Figure III.25 View d'utilisation de Google API

Cette dernière représente la vue dans la quel nous affichons la map avec la liste des différents agents de terrain avec la possibilité de soit localisé l'ensemble des agents ou bien de sélectionné un seul.

V-7 Réclamation client

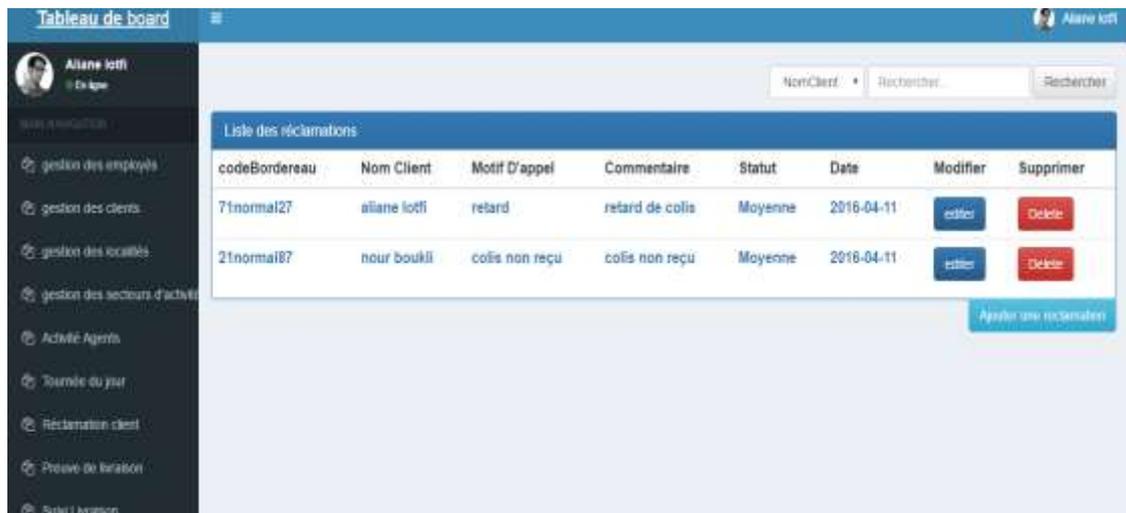


Figure III.26 Réclamation client

Le superviseur liste les différentes réclamations des clients avec la possibilité de les modifier, de les supprimer ou d'ajouter des nouvelles réclamations et de pouvoir les trier selon le client, code bordereau ou bien par ordre de priorité.

V-8 Preuve de livraison

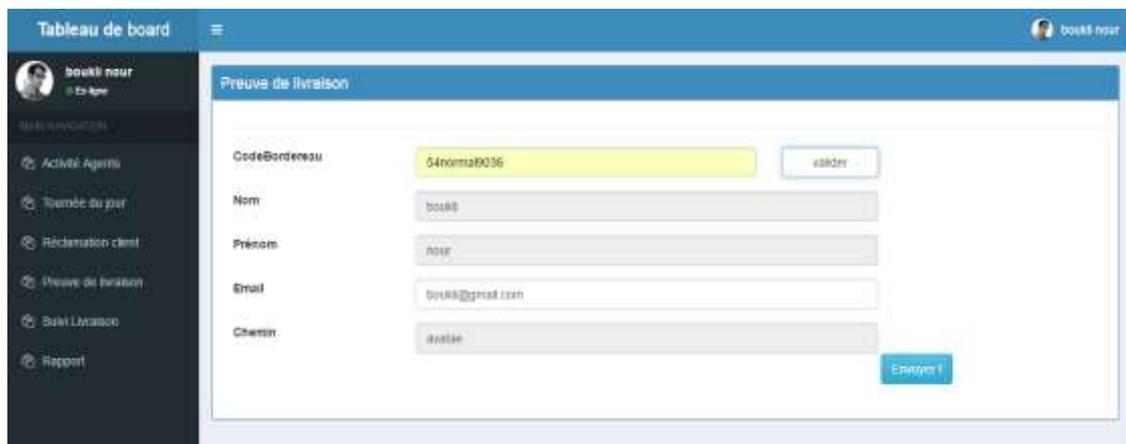


Figure III.27 Preuve de livraison

Dans le cas d'une réclamation client de non réception du colis le superviseur a la possibilité de fournir une preuve de livraison en saisissons le code du bordereau, qui va automatiquement charger les coordonnées relatif à la commande nom, prénom, email (modifiable) et le chemin exact de la signature du client en format PNG et d'envoyé la photo de la signature du client par mail.

V-9 Suivi de livraison



Figure III.28 Suivi livraison

Cet écran permet au superviseur d'avoir accès à la différente commande passée par le biais de son agence avec la possibilité de traquer chaque colis comme le montre le pop-up de la figure ci-dessous

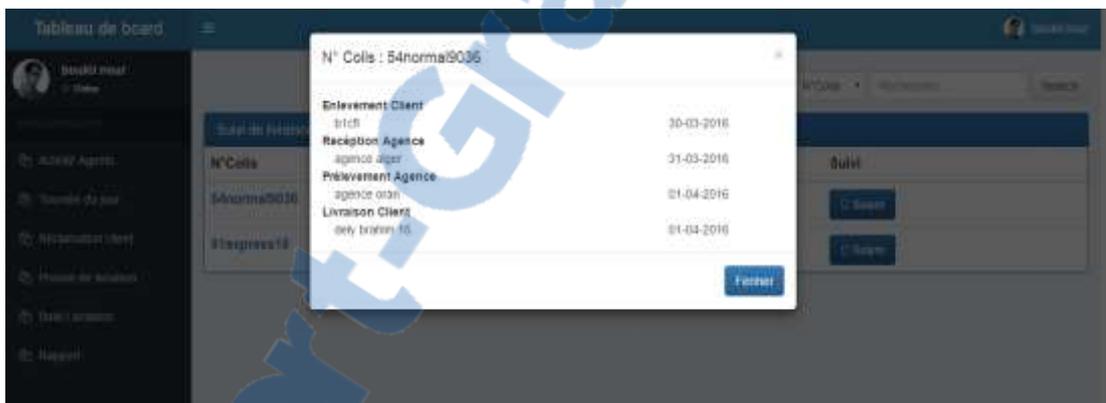


Figure III.29 Liste des adresses

On voit bien par exemple pour le numéro de colis 54normal9036 il a été enlevé du client, reçu par l'agence, prélevé de l'agence et livrés au client le premier d'avril 2016

VI- Conclusion

Dans ce chapitre, nous avons présenté les différentes étapes du processus de réalisation de notre système en spécifiant l'environnement de développement et notre démarche pour l'achèvement de notre travail. Après avoir mené une étude sur les modules déjà réalisé (front office et front mobile), nous avons rajouté un autre module (Back office).

Conclusion Générale

L'objectif principale de ce stage est le développement d'un module web afin de compléter les deux modules déjà existant offrant la possibilité aux superviseurs et administrateurs de mettre en relations les différentes commandes clients effectuer depuis le front office avec les agents de terrain en fonction de leurs situation géographique aussi de pouvoir gérer les différents secteurs d'activités ainsi que les comptes clients et employés avec la possibilité de fournir un service après livraison en cas de réclamation client.

Dans ce mémoire, nous avons détaillé les différentes étapes de conception et réalisation du back office que nous avons développé.

Ce stage de fin d'étude a fait l'objet d'une expérience intéressante, qui nous a permis de découvrir le monde professionnel et de mettre en pratique ce que nous avons acquis durant notre cursus et d'améliorer nos connaissances et nos compétences sur le plan technique.

Le travail que nous avons accompli répond aux exigences et objectifs fixés au début du stage. Cependant il peut encore évoluer et se voir améliorer. Comme perspective nous envisageons la mise en œuvre de l'affectation des activités automatiquement en fonction de l'adresse d'envoi d'un colis et la position géographique d'un agent. Nous souhaitons également générer et visualiser les rapports journaliers de l'ensemble des activités.

Références Bibliographiques

- [1] Hugues Bersini, La programmation orientée objet, Eyrolles, 6e édition, 7 mars 2013, 650p, Noire programmation et langages, N° 198, ISBN 2212135785
- [2] Sparx System Ltd and SparxSystems Software Gmbh, Entreprise Architect [en ligne].publier en 2016, Disponible sur <<http://www.sparxsystems.eu/start/home/>> (traduit).Consulté le 26/04/2016
- [3] Pierre Gérard, IUT de Villetaneuse. DUT informatique .Diagramme de cas d'utilisation [Cours en ligne], S2 2013.Disponible sur <<http://lipn.univ-paris13.fr/~gerard/uml-s2/uml-cours04.html>> (traduit), Consulté le 9/05/2016
- [4] Cian .Developpez.com, Diagramme de séquence [Cours en ligne], publier le 03/12/03, Disponible sur <<http://cian.developpez.com/uml2/tutoriel/sequence/>> (traduit).Consulté le 10/05/2016
- [5] Books LLC, diagramme UML, Livres groupe, 01 aout 2010, 42p ,ISBN 978-1159648398
- [6] Claude Aubry, SCRUM, le guide pratique de la méthode agile la plus populaire, Préface de François Beauregard, Dunod, 2011.
- [7] Sondra Ashmore Ph.D., Kristin Runyan, Introduction to Agile Methods, Livre, Addison-Wesley Professional, 2014
- [8] <https://www.mountaingoatsoftware.com/agile/scrum> consulté le 13/05/2016.
- [9] Gregory Mcbride. The Gantt Chart Handbook – Everything you need to knew about Gantt chart . Emereo Publishing.29 avril 2016.46p.ISBN 1489120416
- [11] TRACKBUSTERS, Définition Framework [en ligne], publier le 01/10/2014.
Disponible sur <<http://www.trackbusters.fr/definition-framework.html>>, Consulté le 27/04/2016
- [12] Maurice Chavelli, Découvrez le framework PHP Larevel,Eyrolles,16 juin 2016,développement web,N°132, ISBN 2212143982
- [13] Margaret Rouse,TechTarget, GitHub [Article], publier en mai 2016, Disponible sur « <http://searchitoperations.techtarget.com/definition/GitHub> », consulté le 10 mai 2016

Références Bibliographiques

[14] Konate DAHABA, L'audit logistique de la distribution de masse de masse des moustiquaires impregnées de l'Ong PSI CI, Cote d'Ivoire : Ecoles Supérieur HETEC, 2010-2011, 139p.

[15] cliCours , Etude de la géolocalisation dans les réseaux mobiles GSM et CDMA à MOBILINF, 2015, Disponible sur<<http://www.clicours.com/memoire-online-etude-de-la-geolocalisation-dans-les-reseaux-mobiles-gsm-et-cdma-a-mobilinfo/>>

[16] Nouha KHYARI, rapport de stage sur le projet « Locate my car », Ecole nationale des sciences de l'informatique Tunisie, 2010, Disponible sur <http://www.memoireonline.com/03/12/5548/m_Rapport-de-stage-sur-le-projet-Locate-my-car-google-map-android.html>

Liste des figures

Figure I.1 Architecture globale du projet Tracking Colis	1
Figure II.1 Diagramme de cas d'utilisation	8
Figure II.2 Diagramme de séquence « créer adresse »	9
Figure II.4 Diagramme de Séquence « Affecter une activité ».....	11
Figure II.5 Maquette d'activité Agent.....	12
Figure II.6 Diagramme de Séquence « Prélèvement client ».....	13
Figure II.7 Diagramme de Séquence « Réception Agence »	14
Figure II.8 Diagramme de Séquence « Prélèvement Agence »	15
Figure II.9 Diagramme de Séquence « Livraison Client »	16
Figure II.10 Diagramme de Séquence « Traquer colis client »	17
Figure II.11 Diagramme de Séquence « Traquer colis Superviseur »	18
Figure II.12 Diagramme de Séquence « géo localiser un employé»	19
Figure II.13 Maquette de « géolocalisation d'employé ».....	19
Figure II.14 Diagramme de Séquence « gérer les localités ».....	20
Figure II.15 Diagramme de Séquence « gérer les secteurs d'activité ».....	21
Figure II.16 Diagramme de Séquence « Fournir la preuve de livraison ».....	22
Figure II.17 Maquette de preuve de livraison	23
Figure II.18 Le diagramme de class du projet tracking	24
Figure III.1 Cycle de vie Scrum.....	26
Figure III.2 Fichier de suivi du Back office (projet tracking colis)	28
Figure III.3 Graphe générer par le fichier de suivi du projet tracking colis	29
Figure III.4 Diagramme de Gantt du Back office	30
Figure III.5 Architecture de base de laravel.....	31
Figure III.6 Dossier laravel	32
Figure III.7 Connexion à la base de donnée.....	33
Figure III.8 La table localisation.....	33

Liste des figures

Figure III.9 Authentification	35
Figure III.10 Tableau de bord Administrateur	35
Figure III.11 Gestion des employés	36
Figure III.12 Gestion des localités	36
Figure III.13 Affectation d'un superviseur	37
Figure III.14 Model localité	37
Figure III.15 Model employé	38
Figure III.16 Controller localité	38
Figure III.17 Liste des secteurs d'activités	39
Figure III.18 Affectation des agents au secteur	39
Figure III.19 Géolocalisation d'agents.....	40
Figure III.20 Schéma du système de géolocalisation.....	40
Figure III.21 Géolocalisation controller.....	41
Figure III.22 Géolocalisation scripte	42
Figure III.23 Géolocalisation scripte	42
Figure III.24 Géolocalisation scripte	43
Figure III.25 View d'utilisation de Google API	43
Figure III.26 Réclamation client	44
Figure III.27 Preuve de livraison	44
Figure III.28 Suivi livraison.....	45
Figure III.29 Liste des adresses	45

Résumé

Les sociétés de logistique sont de plus en plus contraintes par leurs clients d'avoir des solutions fiables de suivi de colis ou autre type de produits. Dans le cadre de ce marché et suite à la sollicitation de sociétés de logistique, la société Eurequat a lancé le projet « Trak'in » qui se constitue de trois modules, un module dédié aux clients des logisticiens à travers lequel on peut déclencher les demandes d'envoi de colis/courrier et en même temps suivre l'avancement de la livraison. Un module dédié à enregistrer les différents événements du terrain : collecte colis, signature destinataires du colis. Un module dédié à la gestion des agents (transporteur) sur le terrain et la gestion des clients, ce dernier a fait l'objet de notre travail dans le cadre de ce projet.

Abstract

Logistics companies are increasingly forced by their customers to have reliable Tracking solutions package or other type of product. As part of this market and Responding to logistics companies requests, the company Eurequat launched the "Trak'in" project which consists of three modules, a module dedicated to customer logistics through which we can trigger requests sending parcels / mail and simultaneously monitor the progress of the delivery. A module dedicated to record the various events of the field: collecting parcels, recipients signing the package. A module dedicated to managing agents in the field and customer management, it has been our work on this project.

ملخص

إن الشركات اللوجستية في قيود مستمرة مع زبائنها من اجل الحصول على حلول موثوقة لاتباع الطرود، أو منتوجات أخرى. في إطار هذا المجال، واتباعا لرغبات الشركات اللوجستية، أطلقت شركة EUREQUAT مشروع Tracking الذي يحتوي على ثلاث وحدات: وحدة الى الزبائن، وحدة اخرى الى موظفي الميدان وحدة إلى المسؤولين الشركة. في هذه المذكرة، نهتم بتطوير وحدة لتكمل الودعتين السابقتين. وتمكين المدراء والمشرفين من وضع علاقات بين مختلف طلبات الزبائن وموظفي الميدان، اعتمادا على موقعهم الجغرافي، وإدارة مختلف القطاعات وحسابات الزبائن والموظفين، مع إمكانية توفير خدمة بعد التوصيل في حالة شكوى الزبائن