

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Université Saad Dahleb de Blida



Faculté des sciences

Département d'Informatique

En vue d'obtenir le diplôme de Master

Domaine : Mathématique et informatique

Filière : Informatique

Spécialité : Ingénierie du logiciel

Sujet :

Affectation automatique et semi-automatique des votants aux centres de vote

Mémoire Présenté par :

Mlle MAKHLOUFI Amina

Mlle ZAIDI Louiza

Sous la direction du promoteur:

BENNOUAR Djamal

Encadré par: :

Mr. BOUDEFLA Mohamed

2011/2012

Remerciements

*Avant tout, nous remercions Dieu, le miséricordieux,
Le Tout Puissant
Pour nous avoir donné la force et le courage d'accomplir ce travail avec beaucoup de
courage .*

*Nos vifs et sincères remerciements
S'adressent spécialement à,
MR, Djamal Bennouar*

*pour l'orientation, la confiance, la patience qui ont constitué un apport considérable
sans lequel ce travail n'aurait pas pu être mené au bon port, pour ses bonnes
explications qui nous ont éclairé le chemin de la recherche et sa collaboration avec
nous dans l'accomplissement de ce travail*

Aussi,

*Nous adressons nos vifs remerciements à
Mr Boudefla Mohamed, Melle ferhi faiza pour leurs aides...*

Nous remercions également,

*Les membres du jury de nous avoir fait l'honneur d'expertiser ce mémoire.
Veuillez accepter l'expression de notre vive gratitude et notre parfaite déférence.*

à tout nos Professeurs,

*Nous saisissons cette occasion pour vous exprimons notre profonde reconnaissance
tout en vous témoignant notre respect et notre humble dévouement.*

Enfin,

*A toutes les personnes qui ont contribué de près ou de loin, d'une
manière directe ou indirecte à l'élaboration de ce travail de fin d'études.*

Merci 

Dédicaces

A ma mère,

*Du moment que tu es là maman je n'ai besoin de rien, ta présence seule me suffit, et ton sourire seule me comble.
Je ne sais ce que serais ma vie sans toi, t'avoir à mes côtés vaut pour moi tous l'or du monde, et toute les joies de cette vie.
merci pour tous ce que vous avais fait pour moi.*

A mon père,

*Tous les mots du monde ne sauraient exprimer l'immense amour que je te porte, ni la profonde gratitude que je te témoigne pour tous les efforts et les sacrifices que tu n'as cessé de consentir pour mon instruction et mon bien-être.
merci pour tous ce que vous avais fait pour moi.*

J'espère

*Avoir répondu aux espoirs que vous avez fondés en moi.
Je vous rends hommage ici par ce modeste travail en guise de ma reconnaissance éternelle à jamais...
Que Dieu, le Tout Puissant vous garde et vous procure santé, bonheur et longue vie.*

J'ai un grand plaisir a dédier ce modeste travail

*A Mon grand-père maternel abderahman et ma grand-mère maternelle zineb
Que Dieu leur accorde une longue vie.
Mes très chers frères Nabil, abdeallah, Sohaib
Ma chère sœur Hadjer
Ma tante Akila, Fatiha, Meriem et sa fille, Dahbiya, Faiza
A ma chère binôme Louiza
Et ma belle Mahdia
Que Dieu vous garde auprès de moi.*

Résumé

L'affectation de votants aux centres de vote se fait actuellement de manière manuelle. Elle est réalisée par le service de vote qui se base sur une connaissance parfois non précise. Et elle nécessite d'un temps très long et très lourde, pour cela il est nécessaire de mettre en place un outil pour rendre l'affectation automatique

Ce travail se concentre sur la mise en place d'un logiciel qui permet de gérer les processus d'affectation des votants. En se basant sur l'information cartographique. Une telle qualité d'information rendre le logiciel plus efficace. Ce composant sera conçu et réalisé selon une approche MVC.

Mots clé : affectation, information cartographique,

ملخص

تعيين الناخبين إلى مراكز الاقتراع حاليا يدويا. يتم تنفيذ من قبل محطة الخدمة التي تقوم على المعرفة غير دقيقة في بعض الأحيان. وتتطلب فترة طويلة جدا وثقيلة جدا، لهذا لا بد من تطوير أداة لجعل التعيين التلقائي يركز هذا العمل على وضع البرامج التي تسمح لك لإدارة عملية التصويت تكليف. استنادا إلى معلومات التعيين. هذه المعلومات نوعية لجعل البرنامج أكثر فعالية. وسيتم تصميم هذا المكون وتقدم وفقا لنهج MVC.

Abstract

The assignment of voters to polling centers is currently manually. It is performed by the service station which is based on knowledge sometimes inaccurate. And it requires a very long and very heavy, for this it is necessary to develop a tool to make the automatic assignment.

This work focuses on the development of software that allows you to manage the process of assigning voting. Based on the mapping information. Such quality information to make the software more effectively. This component will be designed and made according to a MVC approach

Table des Matières

Chapitre I : Introduction Générale

1. Introduction	2
2. Problématique.....	2
3. Objectifs.....	3

Chapitre II : Google Maps : Généralités e Définitions

1. Introduction	6
2. Généralités sur Google Maps.....	6
2.1. Qu'est que l'information géographique ?.....	6
2.2. Qu'est-ce qu'une carte ?.....	6
2.3. Les composantes d'une carte	7
2.4. Les types de cartes.....	7
a) Les cartes topographiques.....	7
b) Les cartes thématiques.....	8
3. Cartographie dynamique dans le web.....	8
4. Définition de Google Maps.....	9
4.1. Présentation de Google Maps.....	10
5. API GOOGLE MAPS.....	10
5.1. Retour historique sur L'API	10
5.2. Qu'est ce qu'une API ?.....	10
5.2.1 Définition 1.....	10
5.2.2. Définition 2.....	11
5.3. Que peut faire l'api Google maps?.....	11
5.4. API Google Maps version 3	12
5.4.1.Définition.....	13
5.4.2. les caractéristiques de l'API (Google Maps) version3	13
5.4.3. Les nombreuses fonctionnalités de l'API v3.....	14
5.4.3.1 Création de la carte	14
5.5. Description des différents services	16
5.5.1 Service de géo localisation.....	17
5.5.1.1 Définition1.....	17
5.5.1.2 Definition2.....	17
5.5.1.3 Construction du service de géo localisation.....	18
5.5.2. Service d'itinéraire.....	19
5.5.2.1. Construction du service d'itinéraire.....	20
5.5.3. Service de STREETVIEW.....	22
5.5.3.1.Utilisation du panorama en dehors de la cartographie.....	22
5.6. Différence entre API V2 et API V3 de Google MAPS.....	24
Conclusion	29

Chapitre III : Analyse des besoins

1. Introduction.....	31
2. Acteur de système.....	31
3. Diagrammes de cas d'utilisation global.....	31
3.1. Diagramme cas d'utilisation générale	31
3.2. Gestion des comptes	32
3.3. Gestion des votants	33
3.4. Gestion des centres et bureaux de vote	33
3.5. Navigation géographique	34

Géo localisation automatique.....	35
Géo localisation manuelle.....	35
5.6. Affectation automatique.....	36
5.7. Réajustement d'affectation.....	36
5.8. Gestion des staffs.....	37
Conclusion.....	38
Chapitre IV : Conception	
1. Introduction.....	40
2. Conception du système.....	40
3. Description des cas d'utilisation.....	41
3.1. Gestion des comptes.....	42
3.1.1 Cas d'utilisation : "créer un compte utilisateur".....	42
Généralité.....	42
Scenario.....	42
3.1.1.1 Diagramme de séquence : « créer compte utilisateur».....	43
3.1.2. Cas d'utilisation : « modifier un compte utilisateur».....	44
Généralité.....	44
Scenario.....	44
3.1.2.1 Diagramme de séquence : « modification d'un compte utilisateur».....	45
3.1.3 Cas d'utilisation : « supprimer un compte».....	46
Généralité.....	46
Scenario.....	46
3.1.3.1 Diagramme de séquence : « suppression d'un compte utilisateur».....	47
3.1.4. Diagramme d'activité : « gestion des comptes».....	48
3.2 Cas d'utilisation : "Authentification".....	49
Généralité.....	49
Scenario.....	49
3.2.1 Diagramme de séquence : « Authentification».....	50
3.2.2 Diagramme d'activité : « Authentification».....	50
3.3. Gestion des votants.....	51
3.3.1 Cas d'utilisation : "Ajout de votant".....	51
Généralité.....	51
Scenario.....	51
3.3.1.1 diagrammes de séquence : « ajout d'un votant».....	52
3.3.1.2 diagramme d'activité : « ajout d'un votant».....	53
3.3.2 Cas d'utilisation : "Modifier les informations d'un votant".....	54
Généralité.....	54
Scenario.....	54
3.3.2.1 diagrammes de séquence : « Modifier les informations d'un votant».....	55
3.3.2.2 diagramme d'activité : « Modifier les informations d'un votant».....	56
3.3.2 Cas d'utilisation : "supprimer un votant".....	57
Généralité.....	57
Scenario.....	58
3.3.2.1 diagrammes de séquence : « supprimer un votant ».....	59
3.3.2.2 diagramme d'activité : « supprimer un votant ».....	59
3.4. Gestion des centres et bureaux de vote.....	60
3.4.1 Cas d'utilisation : « ajouter un Centre de vote».....	60
Généralité.....	60
Scenario.....	60
3.4.1.1 diagrammes de séquence : « ajouter un centre de vote ».....	61

3.4.2	Cas d'utilisation : « ajouter un Bureau de vote »	62
	Généralité.....	62
	Scenario	62
3.4.2.1	diagrammes de séquence : « ajouter un bureau de vote »	63
3.4.2.2	diagramme d'activité : « ajouter un centre et bureau de vote »	64
3.4.3	Cas d'utilisation : « modifier un centre de vote ».....	64
	Généralité.....	64
	Scenario	65
3.4.3.1	diagrammes de séquence : « modifier un centre de vote »	65
3.4.3.2	diagrammes d'activité : « modifier un centre de vote »	66
3.4.4	Cas d'utilisation : « supprimer un centre de vote»	67
	Généralité.....	67
	Scenario	67
3.4.4.1	diagrammes de séquence : « supprimer un centre de vote»	68
3.4.4.2	diagrammes d'activité : « supprimer un centre de vote»	69
3.6.	Navigation géographique.....	70
3.6.1.	Cas d'utilisation : « géo localisation automatique des adresses».....	71
	Généralité.....	71
	Scenario	71
3.6.1.1	diagrammes de séquence : « géo localisation automatique des adresses».....	71
3.6.1.2	diagrammes d'activité : « géo localisation automatique des adresses».....	72
3.6.2.	Cas d'utilisation : « géo localisation manuelle des adresses».....	73
	Généralité.....	73
	Scenario	73
3.6.2.1	diagrammes de séquence : « géo localisation manuelle des adresses»	74
3.6.2.2	diagrammes d'activité : « géo localisation manuelle des adresses».....	75
3.7.	Cas d'utilisation : « affectation automatique ».....	76
	Généralité.....	77
	Scenario	77
3.7.1.	Diagramme de séquence : « affectation automatique».....	78
3.7.2.	Diagramme d'activité : « affectation automatique».....	79
3.8.	Gestion des staffs.....	80
3.8.1	Cas d'utilisation : « ajouter les éléments du staff».....	80
	Généralité.....	81
	Scenario	81
3.8.1.1.	Diagramme de séquence : « ajouts des staffs».....	82
3.8.1.2.	Diagramme d'activité : « ajouts des staffs».....	82
3.8.2	Cas d'utilisation : « Modifier staff».....	82
	Généralité.....	82
	Scenario	82
3.8.2.1.	Diagramme de séquence : « modifier staffs».....	83
3.8.2.2.	Diagramme d'activité : « modifier staffs».....	84
3.8.3	Cas d'utilisation : « supprimer staff».....	84
	Généralité.....	84
	Scenario	84
3.8.3.1.	Diagramme de séquence : « supprimer staffs».....	85
3.8.3.2.	Diagramme d'activité : « supprimer staffs».....	86
3.9.1	Cas d'utilisation : « réajustement d'affectation»	86
	Généralité.....	86

Scenario	86
3.9.1.1. Diagramme de séquence : « réajustement d'affectation».....	87
3.9.1.2. Diagramme d'activité : « réajustement d'affectation».....	88
4. Diagramme de classe.....	89
4.1. Dictionnaire de données.....	89
4.2. Diagramme de classe.....	91
4.3. Codification.....	92
Conclusion.....	92
Chapitre V : Implémentation.....	
1. Introduction.....	
2. Environnement de développement.....	
2.1 Les technologies utilisées.....	
2.1.1. Les APIs Web.....	
2.1.1.1 L'utilisation des APIs Web.....	
2.1.1.2 Avantages.....	
2.2 Outils de développement.....	
2.2.1 Le langage JAVA.....	
2.2.2 Le langage JAVASCRIPT.....	
2.2.3 l'environnement eclipse.....	
2.2.4 Choix du SGBD.....	
2.2.4.1 Présentation de MYSQL.....	
2.2.4.2 Fonctionnalités de MYSQL.....	
2.3 Architecture utilisée	

Table des Figures

Chapitre II : Google Maps :Généralités et Définitions

Figure 1:Organisation d'une application de cartographie numérique.....	9
Figure 2:Géocodage simple en utilisant l'API.....	18
Figure 3:calcul d'itinéraire simple en utilisant l'API.....	20
Figure 4:Streetview en utilisant l'API.....	22

Chapitre III : Analyse des besoins

Figure 5:Diagramme de cas d'utilisation générale.....	32
Figure 6:Diagramme de cas d'utilisation pour la gestion des comptes	33
Figure 7:Diagramme cas d'utilisation pour la gestion des votants.....	34
Figure8:diagramme de cas d'utilisation pour la gestion des centres et bureaux de vote...	34
Figure 9:Diagramme de cas d'utilisation pour la navigation géographique.....	35
Figure 10:Diagramme de cas d'utilisation pour l'affectation automatique.....	36
Figure 11:diagramme de cas d'utilisation pour la réajustement d'affectation.....	37

Chapitre IV : Conception de l'application

Figure 12:l'architecture MVC.....	41
Figure 13:diagramme de séquence pour la création d'un compte.....	43
Figure 14:diagramme de séquence pour la modification des comptes utilisateur.....	45
Figure 15:diagramme de séquence pour la suppression d'un compte utilisateur.....	47
Figure 16:diagramme de séquence pour la gestion des comptes.....	48
Figure 17:diagramme de séquence pour l'authentification.....	50
Figure 18:diagramme d'activité pour l'authentification.....	50
Figure 19:diagramme de séquence pour l'ajout d'un votant.....	52
Figure 20:diagramme d'activité pour ajouter un votant.....	53
Figure 21:diagramme de séquence pour la modification d'un votant.....	55
Figure 22:diagramme d'activité pour la modification d'un votant.....	56
Figure 23:diagramme de séquence pour a suppression d'un votant.....	58

Figure 24:diagramme d'activité pour la suppression d'un votant.....	59
Figure 25:Diagramme de séquence pour l'ajout d'un centre.....	61
Figure 26:diagramme de séquence pour le l'ajout d'un bureau de vote.....	63
Figure 27:diagramme d'activité pour l'ajout d'un centre et bureau de vote.....	64
Figure 28:diagramme de sequence pour modifier un centre de vote.....	65
Figure 29:diagramme d'activité pour la modification d'un centre.....	66
Figure 30:diagramme de séquence pour la suppression d'un centre de vote.....	68
Figure 32:diagramme d'activité pour a suppression d'un centre de vote.....	69
Figure 33:diagramme de séquence pour la géo localisation automatique.....	72
Figure 34:diagramme d'activité pour la géo localisation automatique.....	72
Figure 35:diagramme de séquence pour la géo localisation manuelle.....	74
Figure 36:diagramme d'activité pour la géo localisation manuelle.....	75
Figure 37 :	
Figure 38:diagramme de séquence d'affectation automatique.....	78
Figure39:diagramme d'activité pour l'affectation automatique.....	79
Figure 40:diagramme de séquence pour l'ajout des staffs.....	81
Figure 41:diagramme d'activiste pour l'ajout des staffs.....	82
Figure 42:diagramme de séquence pour modifier staff.....	83
Figure 43:diagramme d'activité pour modifier staff.....	84
Figure 43:diagramme d'activité pour modifier staff.....	85
Figure 44:diagramme de séquence pour la suppression d'un staff.....	86
Figure 45: diagramme d'activité pour la suppression d'un staff.....	87
Figure 46:diagramme de séquence pour réajustement d'affectation.....	88
Figure 47:diagramme d'activité pour le réajustement d'affectation.....	91

Chapitre V : implémentation

Figure 48: ajouter un centre de vote.....97

Figure 49:supprimer un votant de la liste.....98

Figure 50:géo localisation manuelle.....99

Figure 51:géo localisation automatique100

Figure 52:calculer itinéraire pour affecter les votants au centre de vote..... 101

Figure 53:itinéraire choisi.....101

Figure 54:

Introduction Générale

Introduction
Problématique
Objectifs

1. Introduction :

Les technologies de l'information et de la communication sont devenues un puissant levier de développement économique et social. Elles sont la source d'innovations continues et créent des opportunités nouvelles de développement.

À cet effet, et dans le souci de généraliser Internet plusieurs essais ont été faits afin de mettre sur internet les divers services gérés par les institutions étatiques.

Le projet intitulé E-APC a pour objectif de mettre sur pied un système qui permettrait de mettre sur internet les divers services et activités que nous trouvons dans une APC et qui sont fortement sollicités par les citoyens d'une commune tels que le service d'état civil, le service de vote et les divers aspects liés aux délibérations de l'APC.

Le travail s'inscrit dans le contexte de la réalisation de l'aspect service vote pour l'application de eGouvernement eAPC 2.5.

Actuellement le sous système de gestion du service de vote ne permet qu'une affectation manuelle et très lourde des votants au centre de vote puis aux bureaux de vote. Une affectation automatique ou assistée est nécessaire. Cette affectation se fera selon des critères bien précis qu'il faudrait déterminer et appliquer.

2. Problématique :

Le problème d'affectation de votants aux bureaux de vote se fait actuellement de manière manuelle. Elle est réalisée par le service de vote qui se base sur une connaissance, parfois non précise, de la proximité des adresses de votant par rapport au centre de vote. Sur une connaissance acquise au préalable, mais non du voisinage. La connaissance non précise peut mener parfois à des affectations aberrantes. Cette affectation est souvent faite une fois pour toute et ne considère pas les mouvements des personnes. Même si l'adresse est mise à jour au niveau des services qui délivrent les résidences, cette information n'est pas prise en considération par les services de vote et ainsi les listes de votant ne sont pas mises à jour suite à un changement de résidence. Les listes sont souvent mises à jour pour refléter l'inscription d'un nouveau votant ou pour la radiation de votants suite à un changement de commune (radiation d'une commune et inscription dans une autre commune).

Un autre problème apparaît lors de la conduite d'une opération de vote. Ainsi il est fort probable que des centres sont ajoutés ou supprimés, des bureaux dans un centre sont ajoutés ou supprimés. De ce fait, il devient nécessaire de réorganiser les listes de votant de manière très

rapide pour refléter la nouvelle organisation de l'infrastructure du vote, qui aurait été décidé après établissement d'une première liste de votants. L'approche manuelle qui sévit actuellement mènera à beaucoup d'imprécision voir de problème tels que la difficulté de retrouver un votant dans la liste, voir la difficulté de retrouver dans quel centre ou quel bureau aurait été affecté un votant.

3. Objectifs :

Une gestion informatisée des opérations d'affectation de votant aux centres de vote, l'établissement rapide de liste de votant selon la situation de l'infrastructure du vote et de l'opération de vote elle-même est nécessaire.

L'objectif du travail présenté dans ce mémoire est de réaliser un logiciel qui permet d'affecter de manière automatique et assistée les votants aux centres de vote. Ce système doit se baser sur des données très précises de la position des votants par rapport aux centres de vote. A titre d'exemple chaque adresse devra disposer d'au moins d'une coordonnées qui permettra de la situer par rapports aux divers bureaux de votes, qui eux-mêmes possèdent des adresses doté de coordonnée les positionnant dans un système de coordonnée bien précise.

Aujourd'hui, il n'est plus un secret pour personne que Google offre un certain nombre de services très efficace pour reconnaître la position géographique de n'importe quel point sur le globe terrestre. L'utilisation des services de Google pour affecter aux diverses adresses leur coordonnées ne ferait que rendre plus efficace l'opération d'affectation automatique.

L'affectation automatique doit se baser fondamentalement sur les positions géographiques des diverses adresses par rapport aux adresses des divers centre de vote. Les critères d'affectation automatique doivent être déterminés D'emblée, les critères les plus évidents sont la distance entre le centre de vote et le domicile du votant, la capacité du centre de vote, la pondération du vœu d'un votant pour un centre de vote etc..

L'objectif du travail ne se restreint pas seulement à définir les critères d'affectation et appliquer un algorithme. Il faudrait faire un logiciel complet dans lequel cet algorithme serait mis en œuvre. Ainsi à titre d'exemple, l'affectation automatique peut donner des résultats optimaux, mais qui ne seront pas satisfaisant pour plusieurs citoyens. Une affectation manuelle devra être possible. L'affectation manuelle permettrait de réaffecter un votant à un centre de vote. Une interface adaptée à ce type d'opération est nécessaire.

L'interface devra entre autre montrer comment Google Map est exploité pour affecter à chaque adresse ou groupe d'adresse une longitude et une latitude.

Le logiciel à réaliser devra permettre la gestion des votant (notamment la saisie ou leur transfert d'une autre base de donnée), la gestion du positionnement géographique des diverses adresse, la gestion de l'infrastructure de vote (centre, bureau, urnes, les caractéristiques du vote lui-même, les responsables aux niveaux de chaque urnes etc..) et l'établissement des diverses listes de vote avec une précision allant au niveau de l'urne. Une liste de votant au niveau de l'urne devra contenir entre autre les responsables de l'opération de vote au niveau de l'urne. Le logiciel doit être conçu et réalisé dans une optique d'appartenance à un ensemble d'application qui collaborent dans le cadre d'une infrastructure pour le eGouvernement .

II

Google Maps : Généralités & Définitions

Introduction

Généralité sur Google maps

Cartographie dynamique sur le web

Api Google maps

Conclusion

1. Introduction :

La carte a toujours été un outil de pouvoir – que ce soit à une échelle locale pour calculer le montant de l'impôt chez les Égyptiens ou encore à grande échelle pour la domination maritime mondiale lors des explorations maritimes du XV e siècle.

L'évolution de la technique, l'apparition de l'informatique le siècle dernier et L'impressionnante démocratisation de l'Internet a encore renforcé la place centrale de la carte dans notre société. Google, en mettant à disposition du monde entier et de manière gratuite une cartographie mondiale détaillée, a grandement favorisé l'essor de cette nouvelle technique qu'est la cartographie dynamique (ou webmapping). Google Maps en est l'exemple le plus connu et représentatif.

La force de Google a été de donner aux développeurs web, dès juin 2005, la possibilité d'ajouter leurs propres cartes et créations via l'utilisation d'une interface de programmation JavaScript. Le succès fut alors immédiat, et des milliers de sites web utilisent ce service pour enrichir d'une touche géographique leur contenu.

2. Généralités sur Google Maps:

Avant de définir ce que Google Maps, il faut d'abord éclaircir quelques notions importantes relatives à ce sujet, comme la cartographie... etc.

2.1. Qu'est que l'information géographique ?

L'information géographique qui est le mariage de la géographie et de l'informatique est une information reliée à un territoire, c'est un phénomène réel, localisé dans l'espace, à un moment donné.

« L'information géographique est définie comme étant l'objet d'une discipline scientifique qui se rapporte à l'étude des phénomènes physiques, biologiques et humaines localisés à la surface du globe ». (CANCES. M, 92)

2.2. Qu'est-ce qu'une carte ?

Une carte géographique est un document représentant une étendue géographique et mettant en valeur la localisation des phénomènes qu'il décrit. Et elle peut être définie de plusieurs manières :

Selon le comité français de la cartographie, la carte est : « une représentation conventionnelle généralement plane, en positions relatives, de phénomènes concrets ou abstraits, localisables dans l'espace ». (CUENIN. R, 72).

Ou bien: la carte est la : « représentation classique à une échelle donnée, de l'information géographique ». (MOUSSA. M, 93).

« La carte permet de relier une grande variété de données qualitatives que quantitatives destinées à être organisées, analysées, présentées, communiquées et utilisées de façon à ce qu'aucun produit ne puisse l'égaliser.

N'empêche que la carte reste une abstraction de la réalité, elle ne représente pas la réalité elle-même ». (TAYLOR. D. R. F, 91).

La carte est le meilleur outil de présentation de l'information géographique car elle transmet un grand nombre de connaissances et elle se lit facilement.

2.3. Les composantes d'une carte :

Les composantes d'une carte sont (SIOUANI. N & DAOUD. N, 97):

- **La toponymie** : représente le nom de l'entité.
- **La légende** : décrit les entités géographiques représentées sur une carte à l'aide de symboles linéaires surfaciques, ponctuels et textuels.
- **La barre d'échelle** : est le rapport des longueurs d'un élément linéaire mesurées en projection et sur la surface projetée.
- **La rose des vents** : décrit l'orientation de la carte.

2.4. Les types de cartes : (WEGER. G, 99)

De nos jours, l'étendue du domaine de l'information géographique a produit une multitude de cartes de toutes natures. Weger propose une classification des cartes en les regroupant selon leur contenu :

a) Les cartes topographiques :

Les résultats d'observations directes concernant la position planimétrique, altimétrique, la forme, la dimension et l'identification des phénomènes concrets, durables existant à la surface du sol figurent sur ces cartes.

L'objectif de ce type de carte est par conséquent de donner des informations claires et complètes sur les structures d'un territoire (relief, réseau routier et

hydrographique, limites administratives, ...). Échelle du 5 000 au 100 000 selon le degré de développement du pays.

b) Les cartes thématiques :

Elles représentent, sur un fond repère (généralement issue de carte topographique), des phénomènes qualitatifs ou quantitatifs concrets ou abstraits circonscrits et limités par le choix d'un ou plusieurs sujets particuliers.

Les cartes thématiques fournissent des informations sur des thèmes particuliers à un instant donné. Parmi ces cartes, on peut effectuer un classement par thèmes, par exemple des cartes :

- **Physiques:** Géophysique, Géologique, Hydrologique, Climatologique, Météorologique, ...
- **Biogéographiques:** Zoologique, Écologique, ...
- **Géographie humaine:** Démographique, Politique, Administrative, Historique, ...
- **Économique:** Agricole, Industrielle, Transport, Commerce, ...

« Le fond de plan peut être riche ou dépouillé, seuls importent les éléments de repérage permettant de faire la superposition entre les éléments topographiques concernés et les objets nouveaux qu'il faut visualiser. En fait, il s'agit de superposer deux ou plusieurs cartes issues de domaines différents. La carte thématique outre son rôle descriptif, est aussi un outil d'aide à la décision, à la gestion et à la planification. »

(DIDIER. M & BOUV. C, 93)

3. Cartographie dynamique dans le web:

La cartographie dynamique sur Internet est une nouvelle technique de représentation de données géographiques. Elle s'appuie sur les technologies web pour permettre à l'internaute de naviguer dans des cartes et d'aller chercher lui-même l'information dont il a besoin.

L'API Google Maps a permis de démocratiser cette nouvelle technologie, que l'on nomme également « Web mapping ».

La cartographie dynamique sur le Web est une forme de cartographie récente dans l'histoire de la géographie. Il s'agit d'une cartographie où l'utilisateur est acteur de sa découverte d'informations : il zoome, il change de fond de carte, il ajoute ou modifie des informations.

D'une manière simplifiée, la cartographie dynamique regroupe l'ensemble des technologies permettant d'afficher une carte sur le Web.

Ces technologies reposent principalement sur les trois composantes que sont le client, le serveur et les données. La cartographie dynamique permet donc, en fonction d'une requête d'un client envoyée au serveur cartographique, de retourner les données désirées sous la forme d'une carte (voir figure 1).

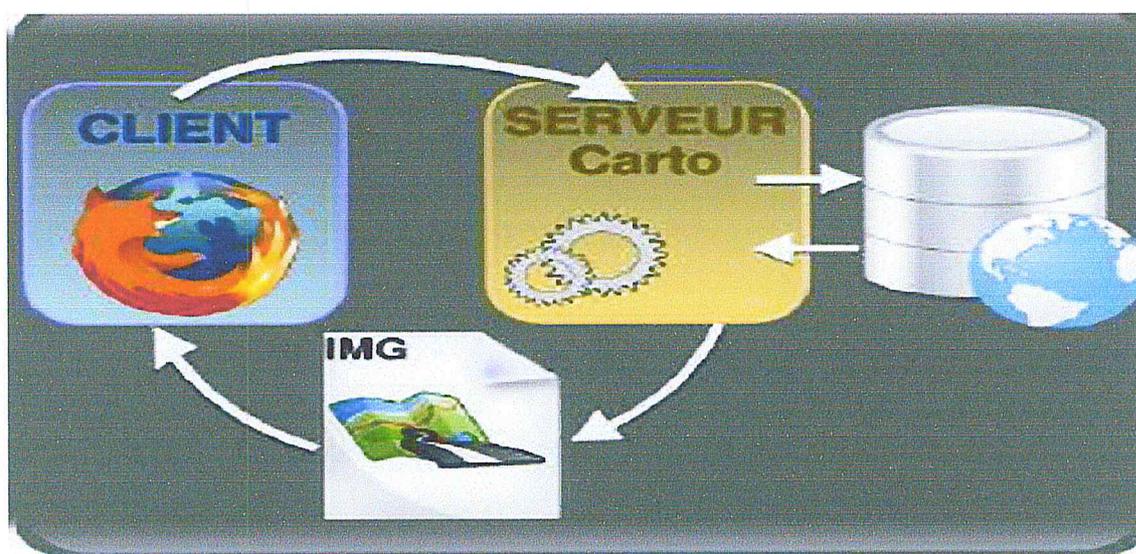


Figure 1: Organisation d'une application de cartographie numérique

Une application reposant sur l'API Google Maps est un exemple représentatif de ces nouvelles technologies de cartographie dynamique. L'internaute, via son client web – son navigateur – demande aux serveurs Google les cartes qu'il désire visualiser à travers l'interface de navigation. Google lui restitue cette information sous la forme d'images, appelées « tuiles », assemblées dans l'interface de navigation. À chaque nouvelle demande – un déplacement par exemple – l'application web repose une requête aux serveurs, qui envoient l'information correspondante ; cet échange se fait de manière transparente, les interactions entre le client et le serveur devant être les moins intrusives possible.

4. Définition de Google Maps:

Tout d'abord nous allons commencer par présenter Google Maps, à quoi ça sert, quelles sont les fonctionnalités.

4.1. Présentation de Google Maps:

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 géo portail. 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 permet, à partir de l'échelle d'un pays, de pouvoir zoomer jusqu'à l'échelle d'une rue. Deux types de vue sont disponibles : une vue en plan classique, avec nom des rues, quartier, villes et une vue en image satellite, qui couvre aujourd'hui le monde entier. Ce service n'est plus en version bêta depuis le 12 septembre 2007, et a été ajouté aux liens de la page d'accueil de Google.

5. API GOOGLE MAPS:

5.1. Retour historique sur L'API :

En premier temps lorsque Google Maps (<http://maps.google.com>) d'abord fait ses débuts, il a été victime de hackers nombreux les gens ont été captivés par cette application de la prochaine génération Ajax qui a fait des choses que personne n'avait jamais vues auparavant. Les développeurs de partout dans le monde voulaient savoir comment ça fonctionnait et comment ils pourraient l'utiliser à leur avantage. Bien que ne pas endommager de quelque façon, ces premiers hackers Google Cartes ont ouvert les yeux des gens à Mountain View, et bientôt l'API Google Maps a été rendu public.

5.2. Qu'est ce qu'une API ?

Definition1 :

Une API est une interface fournie par un programme informatique. Elle permet l'interaction des programmes les uns avec les autres.

D'un point de vue technique c'est un ensemble de fonctions, procédures ou classes mises à disposition par une bibliothèque logicielle, un système d'exploitation ou un service.

Toute application web peut autoriser ou non des développeurs tiers à utiliser une partie de ses fonctionnalités, et ce de plusieurs façons :

- En téléchargeant une bibliothèque de fonctions pour l'inclure directement sur son site.
- En récupérant une donnée précise en construisant une URL comme avec Google
`chartshttps://chart.googleapis.com/chart?cht=chart_type&chd=chart_data&chs=cha...`
- Ou comme Google Maps, en incluant directement la bibliothèque en ligne.

Definition2 :

Une API est une interface de programmation. Dans le cas de Google Maps, il s'agit d'un ensemble de fonctions et classes JavaScript permettent de manipuler une carte dynamiquement au sein d'un site web.

Cette manière de visualiser des informations laisse souvent libre court à l'internaute afin que celui-ci navigue au grès de ses envies à travers la carte. Le fait de pouvoir ajouter des objets ou de l'information permet aux développeurs web de guider les visiteurs sur la carte et sur le site.

La première version destinée aux seuls sites web s'est vue agrémentée au fil des années de plates-formes supplémentaires : API pour Flash, API statique (cf. annexe B), API pour les smartphones (cf. seconde partie). De plus, au fur et à mesure de l'exploitation de cette API, Google a proposé de nouveaux services : géolocalisation, calcul d'itinéraires, calcul d'altitude, etc.

5.3. Que peut faire l'api Google maps?

- Accéder à des services de données : image satellites/aérienne, carte topographique, Streets View, image à 45 degré, ect
- Personnaliser les fonds de carte.
- Ajouter des points, images, lignes, polygones, infos bulles
- Ajouter ses propres données (data overlaid)

- Utiliser des **librairies** : géométrie, panoramique, Places, Static Maps...
- Utiliser des **Web services** : Calcul d'itinéraire, géocodage, Altitude..

5.4. API Google Maps version 3:

La conception d'applications cartographiques pour le Web est passionnante. Elle oblige à être au fait des nouvelles technologies et donc de faire une veille permanente.

C'est ainsi que les solutions de développement proposées par Google deviennent intéressantes. En effet, depuis mai 2010, la firme de Mountain View propose une nouvelle API de Google Maps qui a la particularité de pouvoir facilement être utilisée sur plusieurs plates-formes : ordinateurs classiques et périphériques mobiles (smartphones).

Trois approches de développement sont possibles lors du développement d'une application cartographique pour le Web :

- la première est une approche dite « full web » : on n'utilise que le navigateur et la connexion Internet de la plate-forme pour développer l'application : c'est celle que nous découvrirons dans cette première partie de l'ouvrage ;
- la seconde est une approche dite « native » : on utilise alors les fonctionnalités des plates-formes cibles pour développer les applications – dans le cas de l'iPhone ou d'Android, cela passe par leurs SDK respectifs. Cette approche sera abordée dans la seconde partie de l'ouvrage ;
- la troisième approche est à mi-chemin entre les deux premières : basée sur les webviews, elle délègue le développement de la cartographie dans une page Internet qui sera intégrée telle qu'elle dans l'application.

La troisième version est actuellement conseillée par Google pour tout nouveau développement et mise en production. La version 2 est passée en mode « deprecated », soit déconseillé pour la production de nouveaux mashups.

5.4.1. Définition :

Un mashup permet d'ajouter sur une page web du contenu provenant d'un autre site. En l'occurrence dans notre cas, il s'agit de Google Maps.

Cette nouvelle API, à la différence de la version 2, repose sur le concept MVC (modèle-vue-contrôleur) qui a permis d'alléger considérablement la taille du code JavaScript et par conséquent d'améliorer la fluidité de la navigation.

Elle est également spécialement développée pour pouvoir être affichée sur de petits écrans, tels que ceux qui équipent les Smartphones (i Phone et Android en tête). Ainsi, plus besoin de développement spécifique pour chacune des plates-formes dans le cas d'application ne nécessitant pas l'usage des capteurs spécifiques aux Smartphones : GPS, accéléromètres, etc. (dans ce dernier cas, il faudra développer avec les API spécifiques à ces plates-formes) .

De nombreuses simplifications ont été également apportées à cette nouvelle mouture afin de rendre le code plus lisible.

Note : avec cette version, il n'est plus nécessaire de s'enregistrer auprès de Google pour obtenir une clé d'utilisation de l'API.

5.4.2. les caractéristiques de l'API (Google Maps) version3:

Google a annoncé une mise à jour majeure de son API Google Maps. Alors que Google ajoutait des fonctionnalités à l'API à un rythme assez régulier, l'API Google Maps n'a jamais connu de bouleversements en près de trois ans.

La nouvelle API de Google Maps v3 promet d'être plus simple, plus rapide et optimisé pour les petits écrans comme celui de l'iPhone. Google Chrome fera (officiellement) parti de la liste des navigateurs pris en charge.

Bien que il y ait plusieurs changements importants qui nécessiteront une mise à niveau pour les développeurs. Voici les principales nouveautés :

- Vous n'avez plus besoin de la clé API (google maps api key).

- La taille du JavaScript a été réduite et l'API a été remaniée avec en utilisant l'architecture MVC. Du coup, on se retrouve avec un JavaScript moins lourd et un temps de chargement amélioré.
- L'interface utilisateur par défaut est celle accessible sur le site de Google maps. Cependant, comme avant, il est possible d'interagir sur tous les éléments de l'interface.
- Chrome et Safari Mobile ont été ajoutés à la liste des navigateurs pris en charge. Pour Android c'est en cours de validation.

5.4.3. Les nombreuses fonctionnalités de l'API v3:

Les API cartographiques dynamiques permettent de générer des cartes qui seront intégrées dans des pages web à l'aide de code JavaScript. Elles proposent donc toute une panoplie de services et d'outils à destination de l'internaute afin de l'aider dans la navigation au sein d'une carte :

- contrôles de zoom,
- contrôles de navigation,
- contrôles de choix du type de carte (carte normale, carte relief, carte satellite ou hybride),
- contrôle d'échelle,
- marqueurs et infobulles associées,
- polygones et polygones,
- événements pour tout type d'actions sur les éléments de la carte,
- service de géocodage à l'adresse,
- StreetView,
- itinéraires,
- superposition de couches.

❖ CREATION DE LA CARTE:

L'API Google Maps JavaScript version 3 nous permet de créer, modifier et personnaliser des **cartes géographiques** à l'aide de la class google maps. Plusieurs types de cartes existent :Plan,satellite,mixte, relief.

L'API Google Maps est disponible sous plusieurs déclinaisons, que l'on peut choisir en ajoutant un paramètre à cette URL.

Parmi ces paramètres on retrouve :

- L'utilisation ou non du GPS : "sensor"
- La langue des textes affichés sur la carte : "language"
- Le pays : "region"

Grâce aux caractéristiques de l'objet `google.maps.MapOptions` nous pouvons définir les **propriétés d'une carte** selon vos besoins : centrer la **carte**, définir le type de **carte**, activer ou désactiver le contrôle panoramique, etc.

Ces **cartes** sont interactives. nous pouvons zoomer ou les déplacer à volonté. Il est également possible d'ajouter des **observateurs d'événements** afin de faire réagir la **carte** à certains événements.

différents types d'informations peuvent venir s'afficher sur la **carte Google**, comme par exemple un **itinéraire**, une **image**, un **polygone**, une **polyline**, une **info bulle**, un **Street view**, un **fichier KML KMZ**.

Les contrôles de navigation sont définis par l'objet `NavigationControlOptions` qui possède deux propriétés : `style` et `position`.

Les contrôles de navigation différents proposés par l'API sont définis par la propriété `style` de type :

- **DEFAULT** : les contrôles par défaut de l'API. Celle-ci choisit automatiquement l'interface de contrôle la plus appropriée en fonction de la taille de la fenêtre
- **SMALL** : affiche seulement les boutons de zoom Plus et Moins .
- **ZOOM_PAN** : affiche le contrôle de navigation le plus complet avec flèche de direction Nord, Sud, Est et Ouest, et boutons de zoom Plus et Moins. Ce contrôle complet

ressemble fortement à l'outil de navigation dans le logiciel Google Earth.

- **ControlPosition** : La position sur la carte du contrôle de navigation et prend les valeurs suivantes :
 - ✓ **BOTTOM** : l'élément est placé en bas au milieu de la carte
 - ✓ **BOTTOM_LEFT** : l'élément est placé en bas à gauche de la carte juste à droite du logo Google ;
 - ✓ **BOTTOM_RIGHT** : l'élément est placé en bas à droite de la carte juste à gauche des copyrights ;
 - ✓ **LEFT** : l'élément est placé à gauche, juste au-dessous des éléments positionnés en haut à gauche ;
 - ✓ **RIGHT** : l'élément est placé à droite, juste au-dessous des éléments positionnés en haut à droite ;
 - ✓ **TOP** : l'élément est placé au milieu en haut de la carte
 - ✓ **TOP_LEFT** : l'élément est placé en haut à gauche de la carte ;
 - ✓ **TOP_RIGHT** : l'élément est placé en haut à droite de la carte.

5.5. Description des différents services:

Les services sont la deuxième principale fonctionnalité de l'API Google Maps après l'affichage de primitives (points, lignes, polygones).

Ceux-ci utilisent toute l'intelligence des serveurs de Google pour proposer des outils de géocodage, d'itinéraires, de calcul d'altitude ainsi qu'un outil d'exploration de photos 3D, StreetView.

Nous verrons dans ce qui suit comment utiliser ces différents services et comment les piloter à partir de l'API Google Maps.

5.5.1 Service de géo localisation:

5.5.1.1. Définition 1 :

La géo localisation ou géo référencement est un procédé permettant de positionner un objet (une personne, etc) sur un plan ou une carte à l'aide de ses coordonnées géographiques.

Cette opération est réalisée à l'aide d'un terminal capable d'être localisé (grâce à un (et un récepteur GPS par exemple) ou à d'autres techniques) et de publier (en temps réel ou de façon différée) ses coordonnées géographiques (latitude/longitude).

Les positions enregistrées peuvent être stockées au sein du terminal et être extraites postérieurement, ou être transmises en temps réel vers une plateforme logicielle de géo localisation.

La transmission temps réel nécessite un terminal équipé d'un moyen de télécommunication de type GSM, GPRS, UMTS, radio ou satellite lui permettant d'envoyer les positions à des intervalles réguliers. Ceci permet de visualiser la position du terminal au sein d'une carte à travers une plateforme de géo localisation le plus souvent accessible depuis internet.

5.5.1.2. Définition 2 :

Le géocodage est le processus de conversion des adresses (comme " 1600 Amphithéâtre Parkway, Mountain View , CA") en coordonnées géographiques (latitude 37,423021 comme et la longitude -122,0837339), que vous pouvez utiliser pour placer des marqueurs de position ou de la carte. En outre, le service vous permet d'effectuer l'opération inverse (en tournant coordonnées en adresses), ce processus est connu sous le nom "géocodage inverse.

Donc Le géocodage est la possibilité d'obtenir les coordonnées géographiques d'une adresse Donnée, Mais comment faire pour afficher une adresse dont nous ne connaissons pas ces coordonnées géographiques ? C'est là qu'intervient le géocodage.

lorsque sur la page d'accueil de Google Maps vous saisissez une adresse et cliquez sur « Recherche », Google va chercher dans sa base de données les coordonnées géographiques correspondant à cette requête.

L'API Google Maps fournit exactement le même service : nous envoyons une Adresse et Google nous renvoie les coordonnées géographiques si ces dernières lui sont connues.



Figure 2: Géocodage simple en utilisant l'API

5.5.1.3. Construction du service de geolocalisation:

Une requête de géocodage passe par un objet de la classe Géocoder. Celle-ci ne possède qu'une seule méthode, *géocode*, qui permet d'envoyer la requête :

- **géocode** : prend en argument la question – un objet de type GeocoderRequest –, une fonction de retour qui nous permettra de faire l'appel AJAX vers le service Google et un code de retour.

Un objet de type GeocoderRequest prend quant à lui les arguments suivants :

- **address** : obligatoire si l'argument latLng n'est pas renseigné. Il s'agit de l'adresse à géocoder. C'est une chaîne de caractères .

- **latLng** : obligatoire si address ci-dessus est non renseigné. Il s'agit d'un point dont on veut trouver l'adresse. On parle alors de « géocodage inverse ». C'est un objet de type LatLng .
- **language** : la langue dans laquelle on souhaite obtenir le résultat.

Le code de retour peut prendre les valeurs suivantes :

- **ERROR** : indique un problème inattendu .
- **INVALID_REQUEST** : indique que la question posée est erronée .
- **OK** : la réponse est valide et donc exploitable .
- **OVER_QUERY_LIMIT** : indique que l'application a atteint son quota de questions qui est actuellement de 2 500 requêtes par jour .
- **REQUEST_DENIED** : indique que la requête a été rejetée .
- **UNKNOWN_ERROR** : indique que la requête a échoué. Cependant il est possible qu'elle fonctionne lors d'un essai ultérieur. Indique donc un problème serveur .
- **ZERO_RESULTS** : indique que la question ne renvoie aucun résultat.

Spécifications de la réponse:

Lorsque l'on envoie une question de géocodage via l'API, le service répond par un objet de type GeocoderResult.

On peut extraire de cette réponse plusieurs informations :

- **les coordonnées géographiques** : propriété geometry.location de type LatLng .
- **la précision du retour** : propriété geometry.location_type de type chaîne de caractères .
- **les coordonnées des coins de la carte recommandées pour visualiser le résultat** : propriété geometry.viewport de type LatLngBounds.

5.5.2. Service d'itinéraire:

Le deuxième service de l'API Google Maps est le calcul d'itinéraire. Très abouti, il est en concurrence avec des outils similaires chez Mappy, ViaMichelin ou encore Bing Maps.

L'API dans sa version 3 continue de mettre à disposition des développeurs cette fonctionnalité très populaire. En effet, qui n'a jamais calculé d'itinéraire sur le site de Google Maps ?

La méthode de fonctionnement de ce service est très similaire à celle du géocodage.

En effet, il faut poser une question bien formatée au service de Google qui répond par un objet directement exploitable par l'API.

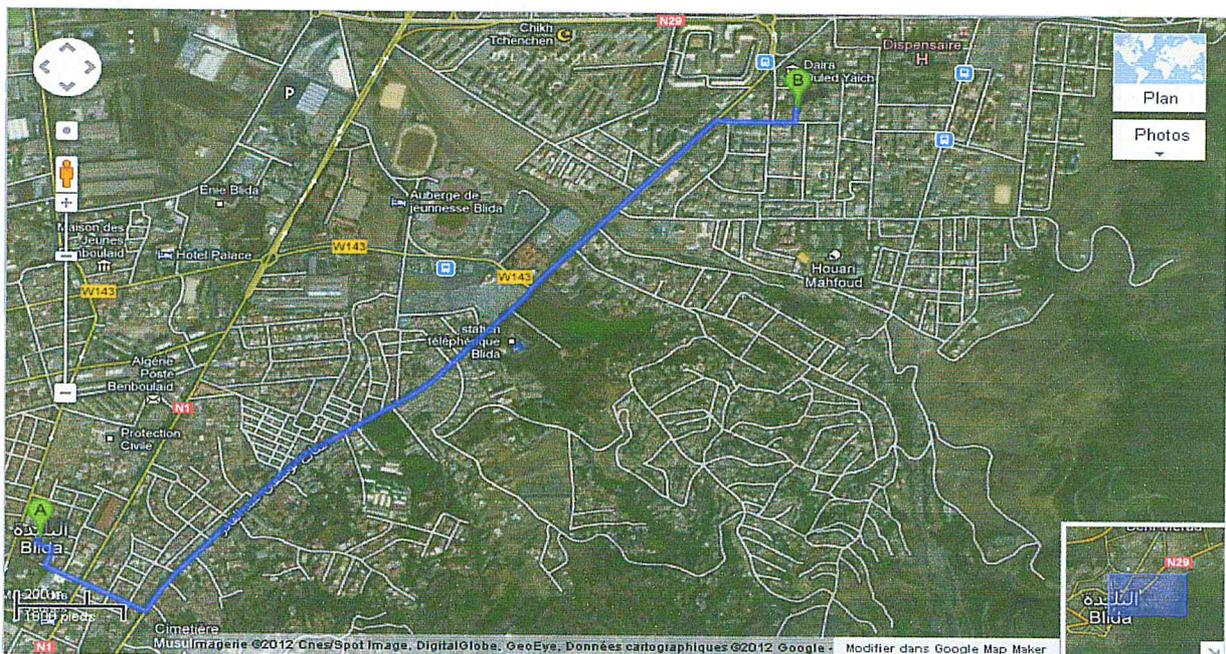


Figure 3: calcul d'itinéraire simple en utilisant l'API

5.5.2.1. Construction du service d'itinéraire :

Le service d'itinéraire de l'API passe par les classes `DirectionsService` et `Directions-Rendered`.

La première étant le constructeur du service et la seconde la classe qui exploite le résultat de la requête.

Une demande d'itinéraire doit comporter les paramètres suivants :

- **origin** : point de départ de l'itinéraire. Peut être un objet LatLng ou une adresse que le service se chargera de géocoder .
- **destination** : point d'arrivée de l'itinéraire. Peut être un objet LatLng ou une adresse que le service se chargera de géocoder.
- **unitSystem** : le type de système de mesure. Deux possibilités : métrique ou système Anglophone.
- **travelMode** : le type de calcul d'itinéraire désiré. Il est possible de calculer des itinéraires en voiture ou à pied.
- **avoidHighways** : autorise le service à calculer un itinéraire empruntant les Autoroutes.
- **avoidTolls** : autorise le service à calculer un itinéraire empruntant des routes à Péage.
- **waypoints** : tableau de points intermédiaires. Peuvent être de type LatLng ou une chaîne de caractères que le service se chargera de géocoder.
- **optimizeWaypoints** : autorise le service à modifier le tableau de points intermédiaires si le coût de l'itinéraire est moins élevé.

Comme le service de géocodage, la construction d'une requête passe par une fonction de retour qui exploitera par AJAX la réponse du service. Le résultat de la requête est un tableau de routes envoyé au format JSON. Il est possible d'en tirer des informations telles que la distance entre deux intersections, le fait de tourner à gauche ou à droite, etc.

Nous verrons ici simplement comment afficher la totalité de l'itinéraire sur la carte et la totalité des informations textuelles dans un bloc, grâce à la classe DirectionsRendered qui possède les méthodes suivantes :

- **setDirections** : prend en argument le résultat d'une requête au service d'itinéraire et affiche sur la carte l'itinéraire en question ainsi que l'ensemble des informations textuelles dans le bloc HTML passé en paramètre avec la méthode setPanel ;
- **setMap** : indique la carte sur laquelle sera affiché l'itinéraire ;
- **setPanel** : indique le bloc HTML où les informations de l'itinéraire seront affichées.

5.5.3. Service de STREETVIEW:

Google StreetView est un service qui propose des photographies panoramiques à 360° d'un grand nombre de routes de part le monde.

Ajouter le service StreetView dans une carte Google Maps c'est avec le contrôle de streetViewControl, Ce contrôle se caractérise par l'ajout d'un petit bonhomme orange au-dessus du contrôle de navigation, Il est alors possible de cliquer sur ce bonhomme et de le déplacer sur la carte, sur une route couverte par le service – surbrillance en bleu. La carte s'efface alors pour laisser place à la navigation StreetView dans des photographies panoramiques à 360 (voir figure):

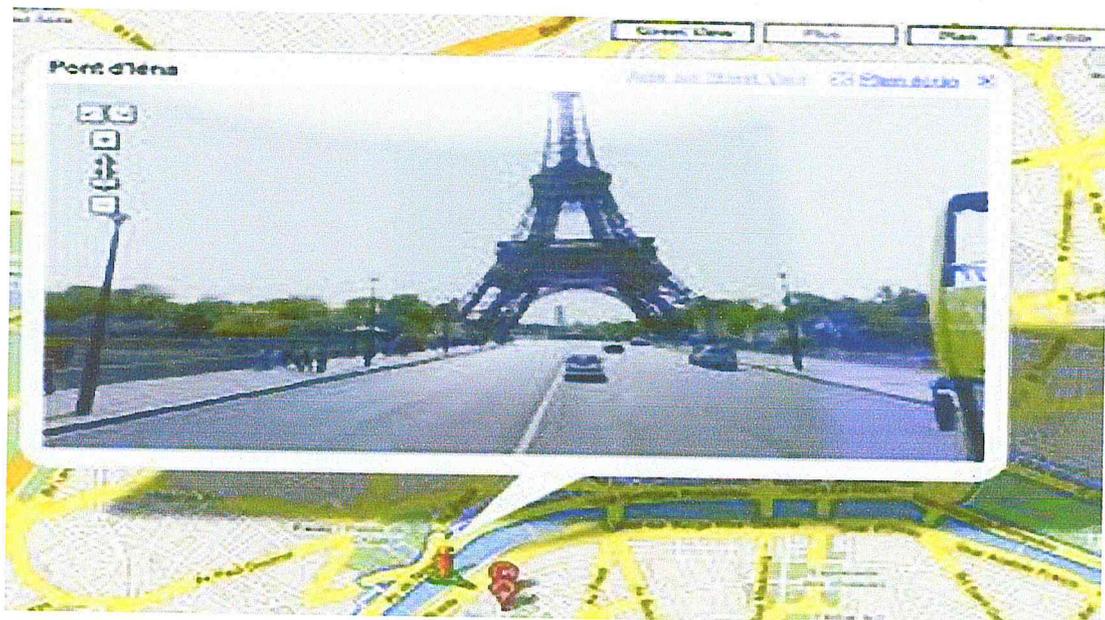


Figure 4: Streetview en utilisant l'API

5.5.3.1. Utilisation du panorama en dehors de la cartographie:

L'utilisation précédente du service StreetView ne permet cependant pas d'avoir le panorama en correspondance avec la carte.

Pour cela, il est nécessaire de déclarer un objet `StreetViewPanorama` qui permet d'externaliser le panorama en-dehors de la carte.

Les options du panorama sont les suivantes :

- **addressControl** : de type booléen. Contrôle l'affichage de l'adresse dans le panorama .
 - **enableCloseButton** : de type booléen. Affiche ou non la croix de fermeture de la fenêtre.
 - **linksControl** : de type booléen. Contrôle l'affichage des liens de contrôle. Si non, il n'est plus possible de se déplacer vers les panoramas suivant et précédent.
 - **navigationControls** : de type booléen. État du contrôle de navigation ;
 - **position** : de type `LatLng`. Il s'agit de la position géographique du panorama ;
 - **pov** : de type `StreetViewPov`. Il s'agit du contrôle de l'inclinaison, du zoom et la direction de l'angle de la caméra ;
-
- **visible** : de type booléen. Contrôle la visibilité du panorama.

Un objet de type `StreetViewPanorama` peut réagir à tous les changements d'état des propriétés précédentes : `links_changed`, `pano_changed`, `position_changed`, `pov_changed`, `visible_changed` ; ainsi que les événements `closeclick` et `resize`. Cet objet panorama possède donc des méthodes qui peuvent piloter toutes ces propriétés, il s'agit des accesseurs d'attributs :

- **getPano** : retourne l'identifiant du panorama courant ;
- **setPano** : édite l'identifiant du panorama ;
- **getPosition** : retourne la position `LatLng` du panorama courant ;
- **setPosition** : modifie la position du panorama. Reçoit un objet de type `LatLng` en paramètre ;
- **getPov** : retourne le point de vue de la caméra sous la forme d'une chaîne de caractère (inclinaison, angle et zoom de la caméra) ;

- **setPov** : modifie le point de vue de la caméra. Prend un objet de type StreetViewPov en paramètre ;
- **getVisible** : retourne la visibilité du panorama ;
- **setVisible** : modifie l'état de visibilité du panorama avec un paramètre vrai ou faux.

Finalement, Le géocodage et le calcul d'itinéraire peuvent être utilisés dans des applications de suivi de flotte et d'optimisation de parcours

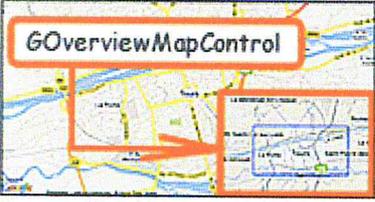
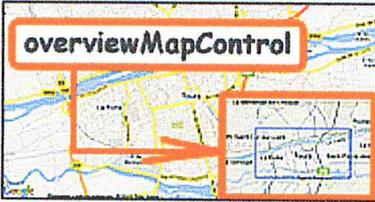
Le service StreetView permet, quant à lui, une nouvelle forme d'exploration du territoire en immersion dans un panorama en trois dimensions.

5.6. Différence entre API V2 ET API V3 de Google MAPS:

Un tableau récapitulatif des principales différences et évolutions, entre l'API Google Maps JavaScript Version 2 et 3.

Version 2	Version 3
<p>Etat de la version Version déclarée obsolète le 19 mai 2010. Cependant, à compter de cette date, elle restera encore active durant 3 ans. Plus aucune évolution ne sera apportée à cette version.</p>	<p>Etat de la version Version déclarée officiellement active le 19 mai 2010. Elle remplace désormais la version 2. Google vous invite désormais à migrer votre code vers la version 3.</p>
<p>Navigateurs compatibles Internet Explorer 6+, FireFox 2+, Safari 3.1+, Chrome</p>	<p>Navigateurs compatibles Internet Explorer 6+, Internet Explorer 7+, FireFox 3+, Safari 4+, Chrome, Android</p>
<p>Versions v=2.s v=2 v=2.x</p>	<p>Versions v=3.8 (Actuelle) v=3.7 (Stable) v=3.6 (Gelée)</p>

<p>Mobiles Non optimisée pour les mobiles.</p>	<p>Mobiles Optimisée pour les mobiles, IPhone et Android par exemple.</p>
<p>Taille des fichiers Importante.</p>	<p>Taille des fichiers Réduite.</p>
<p>Clé API Google Elle est obligatoire.</p>	<p>Clé API Google La clé API Google Maps n'est plus nécessaire.</p>
<p>Espace de nommage G* - Exemple : <code>GLatLng</code></p>	<p>Espace de nommage google.maps.* - Exemple : <code>google.maps.LatLng</code></p>
	<p>MVC l'API Google Maps JavaScript version 3 utilise MVC : Model, View, Controller.</p>
<p>sensor Ce paramètre est apparu et devenu obligatoire dans la V2</p>	<p>sensor Le paramètre sensor est obligatoire. Exemple : sensor = true ou sensor = false</p>
<p>GBrowserIsCompatible Permet de s'assurer que le navigateur utilisé par le visiteur est compatible avec l'API Google Maps JavaScript version 2.</p>	<p>GBrowserIsCompatible n'existe plus.</p>
<p>Type de carte par défaut Par défaut le type de carte est le type Plan. Il est possible d'insérer une carte Google Earth dans une carte Google Maps API JavaScript V2 à l'aide de <code>GMapType.G_SATELLITE_3D_MAP</code>.</p>	<p>Type de carte par défaut Il n'existe plus de type de carte par défaut, il faut obligatoirement l'indiquer à l'aide de MapTypeId :</p> <p style="text-align: center;"> Plan : <code>google.maps.MapTypeId.ROADMAP</code> Satellite : <code>google.maps.MapTypeId.SATELLITE</code> Mixte : <code>google.maps.MapTypeId.HYBRID</code> Relief : <code>google.maps.MapTypeId.TERRAIN</code> </p> <p>Pour l'instant, l'API Google Maps JavaScript V3 ne permet pas de façon native de basculer vers une carte de type Google Earth.</p>
<p>GControl Permet la création de boutons et contrôles</p>	<p>GControl Il n'existe plus. Désormais, les contrôles sont de simples éléments HTML, du type balise <code><div/></code>, créés à l'aide de : <code>document.createElement('div')</code>.</p>
<p>Contrôle Directionnel</p>  <p>Ce contrôle permet de déplacer la carte dans les 4 directions cardinales Nord, Sud, Est, Ouest. La main située au centre du bouton permet de repositionner la carte sur son emplacement initial.</p>	<p>Contrôle Directionnel</p>  <p>Ce contrôle permet de déplacer la carte dans les 4 directions cardinales Nord, Sud, Est, Ouest. Par contre, la main située au centre du bouton, bien que toujours présente, ne permet plus de repositionner la carte sur son emplacement initial.</p>

<p>GOverviewMapControl</p>  <p>Permet d'insérer une mini-carte en bas à droite de la carte principale, afin de géolocaliser celle-ci.</p>	<p>overviewMapControl</p>  <p>Deux nouvelles propriétés ont été ajoutées à l'objet <code>google.maps.MapOptions</code></p> <p><code>overviewMapControl</code></p> <p><code>overviewMapControlOptions</code></p> <p>ainsi qu'un objet <code>google.maps.OverviewMapControlOptions</code> et sa propriété <code>opened</code></p>
<p>Itinéraires à vélo Non disponible.</p>	<p>Itinéraires à vélo Bicycling Layer, permet d'afficher des pistes cyclables. Indisponible en France pour l'instant.</p>
<p>Altitude d'un point Indisponible dans la V2</p>	<p>Altitude d'un point Elevation Service, permet de calculer l'altitude de points.</p>
<p>Polyline Il est possible d'encoder, en amont, les Polylines à l'aide <code>defromEncoded()</code> afin d'améliorer les performances d'affichage.</p>	<p>Polyline L'encodage des Polylines, en amont, n'est plus possible dans l'API Google Maps version 3.</p>
<p>Rectangle La création de Rectangles nécessite la création d'une fonction particulière</p>	<p>Rectangle la création de Rectangles se fait à l'aide d'une classe intégrée dans l'API Google Maps JavaScript version 3 : classe <code>Rectangle</code></p>
<p>Cercle La création de Cercles nécessite la création d'une fonction particulière</p>	<p>Cercle la création de Cercles se fait à l'aide d'une classe intégrée dans l'API Google Maps JavaScript version 3 : classe <code>Circle</code></p>
<p>Géocodeur Nombre de requêtes autorisées : 15.000 requêtes/IP/24 Heures (Version Gratuite).</p>	<p>Géocodeur Il a été amélioré. Nombre de requêtes autorisées : 2.500 requêtes/IP/24 Heures (Version Gratuite). 100.000 requêtes/IP/24 Heures pour les utilisateurs de Google Maps API Premier (Version Payante)</p>
<p>Calcul d'itinéraire</p>	<p>Calcul d'itinéraire Il a été amélioré. Possibilité d'optimiser un itinéraire.</p>
<p>Les info-bulles</p>	<p>Les info-bulles</p>

<ul style="list-style-type: none"> • Les Info-bulles à onglets sont intégrées dans l'API V2. • Une seule info-bulle peut-être ouverte et affichée à la fois. • Lorsqu'une info-bulle est ouverte, un click sur la carte permet de la fermer. 	<p>Les Info-bulles à onglets ne sont plus intégrées dans l'API V3. Plusieurs info-bulles peuvent-être ouvertes et affichées simultanément.</p> <p>Un clic sur la carte ne ferme plus une info-bulle ouverte.</p>
<p>Adsense et GoogleBar insertion possible avec :</p> <ul style="list-style-type: none"> • GAdsManager • GAdsManagerOptions • GAdsManagerStyle • GMap2.enableGoogleBar() 	<p>Adsense avec AdSense Library Une bibliothèque nommée AdSense Library permet d'insérer des publicités Adsense avec :</p> <p>google.maps.adsense.AdFormat google.maps.adsense.AdUnit google.maps.adsense.AdUnitOptions</p> <p>GoogleBar Absent de l'API V3</p>
<p>GDownloadUrl Permet de récupérer, de façon asynchrone, une ressource identifiée par une URL</p>	<p>GDownloadUrl Pas d'équivalent. Il faut utiliser une solution externe à l'API Google Maps Javascript V3</p>
<p>GXmlHttp Permet de créer une instance de XmlHttpRequest</p>	<p>GXmlHttp Pas d'équivalent. Il faut utiliser une solution externe à l'API Google Maps Javascript V3. Une solution de substitution est proposée par Google qui utilise XmlHttpRequest via le fichier utils.js : Retrouvez différents exemples ici : XML parsing</p>
<p>Calculs géométriques Aucun utilitaire permettant d'effectuer des calculs.</p>	<p>Calculs géométriques Une bibliothèque nommée Geometry Library a été ajoutée à la version 3.</p> <p>Celle-ci contient des utilitaires permettant d'effectuer des calculs (angles, distances, zones).</p> <p>Elle comporte deux espaces de nommage :</p> <p>google.maps.geometry.encoding</p> <p>google.maps.geometry.spherical</p>
<p>Panoramio Library Néant</p>	<p>Insérez des images Panoramio avec Panoramio Library Permet d'insérer des photos provenant de Panoramio dans une carte.</p> <p>google.maps.panoramio.PanoramioFeature</p> <p>google.maps.panoramio.PanoramioLayer</p> <p><u>google.maps.panoramio.PanoramioLayerOptions</u> <u>google.maps.panoramio.PanoramioMouseEvent</u></p>
<p>Animation de Marqueurs Néant.</p>	<p>Animation de Marqueurs Rendue possible grâce à la classe google.maps.Animation et ses deux constantes :BOUNCE et DROP</p>
<p>SSL Indisponible dans la version gratuite.</p>	<p>SSL L'accès à l' API Google Maps v3 (mais aussi l'API Static Maps et l'API Web Services Google Maps) dans sa version gratuite est désormais possible</p>

A noté:

L'API Google Maps comporte de nombreuses fonctionnalités, pour retrouver les multiples classes et objets ainsi que leurs propriétés d'utilisation, il est nécessaire de se référer en permanence au document qui liste toutes les fonctionnalités de l'API : « Google Maps JavaScript API V3 Reference ». Il se trouve à cette adresse :

<http://code.google.com/intl/fr/apis/maps/documentation/javascript/reference.html>.

Conclusion :

Ce chapitre a été l'occasion de voir en détail l'objet Map qui est le socle de toute application Google Maps.

Une carte est donc définie par ses options ou propriétés, par des méthodes propres permettant de manipuler la carte (obtenir des informations sur celle-ci ou se mouvoir géographiquement), par des types de contrôle sur la navigation, le choix du type de carte ou le choix d'une échelle, et enfin sur des événements qui peuvent être interceptés afin de rendre l'application interactive.

III

Analyse des besoins

Introduction

Diagramme de cas d'utilisation

Conclusion

1. Introduction :

Dans le cadre de notre travail et pour répondre à l'objectif de notre travail qui est la réalisation d'une application d'affectation automatique de votants au centres et bureaux de vote on se basant sur l'information géographique de GOOGLE, Était nécessaire de passer par cette étape qui à une extrême importance dans la modélisation de système, elle consiste à déterminer de manière explicite les besoins de l'utilisateur.

Nous avons utilisé la technique des cas d'utilisation proposé par UML.

Cette technique va nous permettre de représenter les relations entre les acteurs et les fonctionnalités du système, ainsi que les diagrammes de cas d'utilisation permettent de recueillir, d'analyser et d'organiser les besoins et de recenser les grandes fonctionnalités d'un système, ainsi c'est une vue externe de la façon d'utiliser se système autrement dit guide d'utilisation de système, que se soit l'application, un sous système, une fonction, un composant.

2. Acteur de système :

Utilisateur : c'est une personne qui possède une session de travail dans le système.

L'accès à sa session

nécessite toujours une authentification. Un utilisateur possède des droits d'accès aux informations personnelles. Ces droits d'accès sont explicitement définies pour chaque utilisateur par l'administrateur du système. Il a la possibilité d'effectuer différentes tâches selon son profil qui peut être :

- Administrateur du système
- Agent d'affectation
- Agent de service de vote

3. Diagrammes de cas d'utilisation global:

Ces diagrammes permettent d'identifier le comportement et les fonctionnalités que pourra effectuer l'acteur de se système d'affectation donc les relations fonctionnelles entre l'acteur et le système étudié, ainsi cela permet de voir comment ce module d'affectation gère ces fonctionnalités.

3.1. Diagramme cas d'utilisation générale :

Un **cas d'utilisation** permet de mettre en évidence les relations fonctionnelles entre les acteurs et le système étudié. [W5]

Ce diagramme représente la vue générale de système, ainsi que les fonctionnalités que pourra effectuer l'acteur de se système d'affectation, et comment les gérer

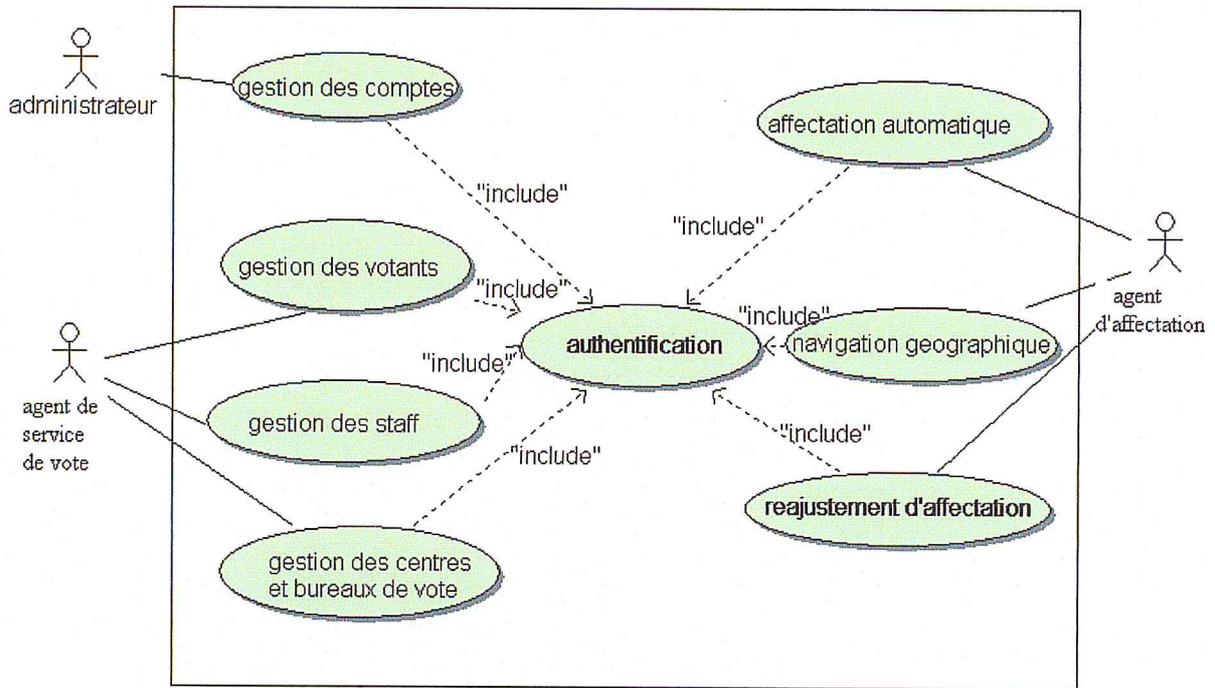


Figure 5:diagramme de cas d'utilisation générale

3.2. gestion des compte :

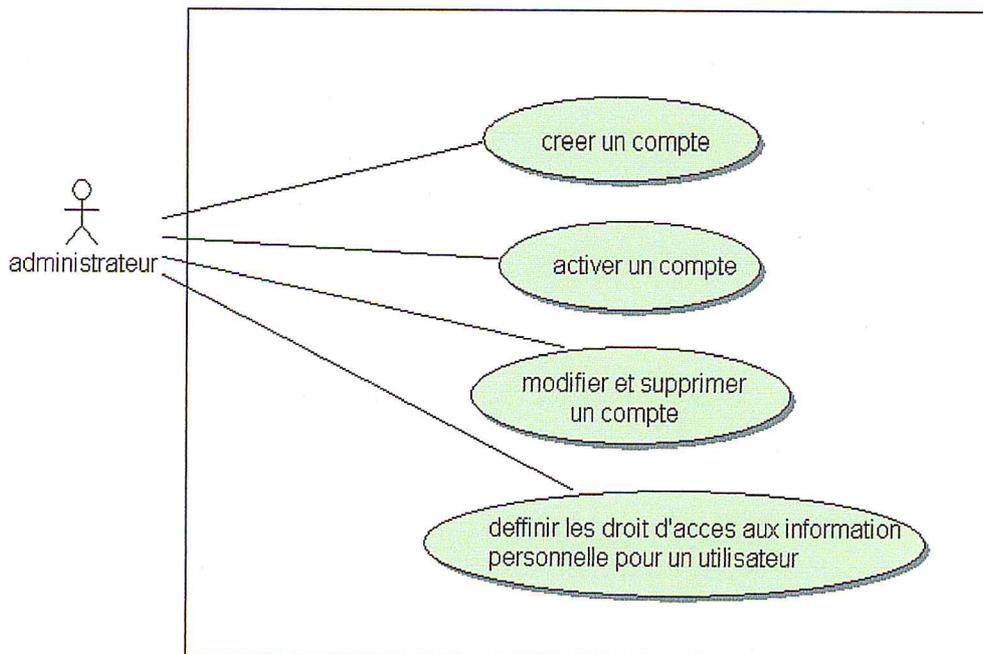


Figure 6:cas d'utilisation pour la gestion des comptes

3.3. Gestion des votants :

Avant toute implémentation de la partie affectation des votants aux centres de votes, il était nécessaire de mettre en place les différentes opérations qui gèrent le votant et compris les opérations suivantes:

- ✓ l'ajout de votant.
- ✓ mise à jour d'un votant.
- ✓ La suppression d'un votant.

Et voici le diagramme cas d'utilisation qui présente cette gestion.

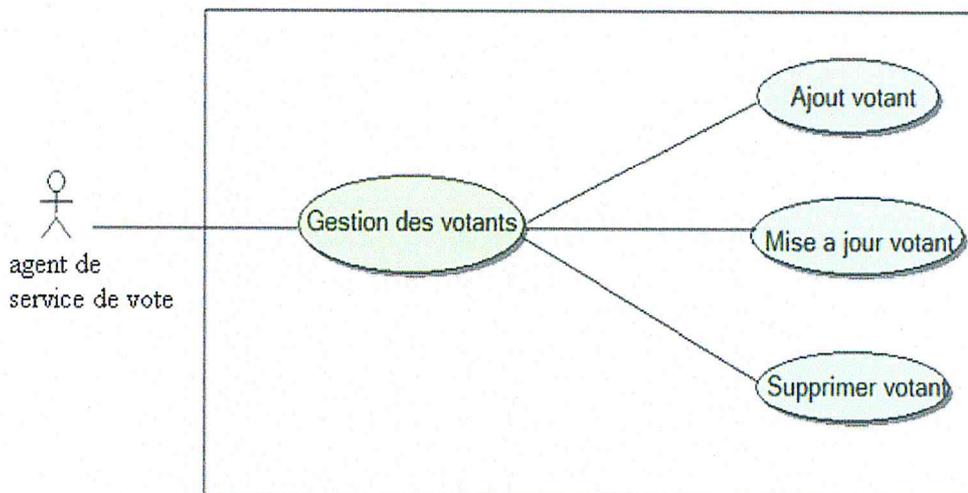


Figure 7:Diagramme cas d'utilisation pour la gestion des votants

3.4. Gestion des centres et bureaux de vote :

Avant toute implémentation de la partie affectation des votants aux centres de votes, il était nécessaire de mettre en place les différentes opérations qui gèrent les centres et bureaux de vote et compris les opérations suivantes:

- ✓ l'ajout d'un centre et bureau de vote.
- ✓ mise à jour d'un centre et bureau de vote.
- ✓ La suppression d'un centre et bureau de vote.

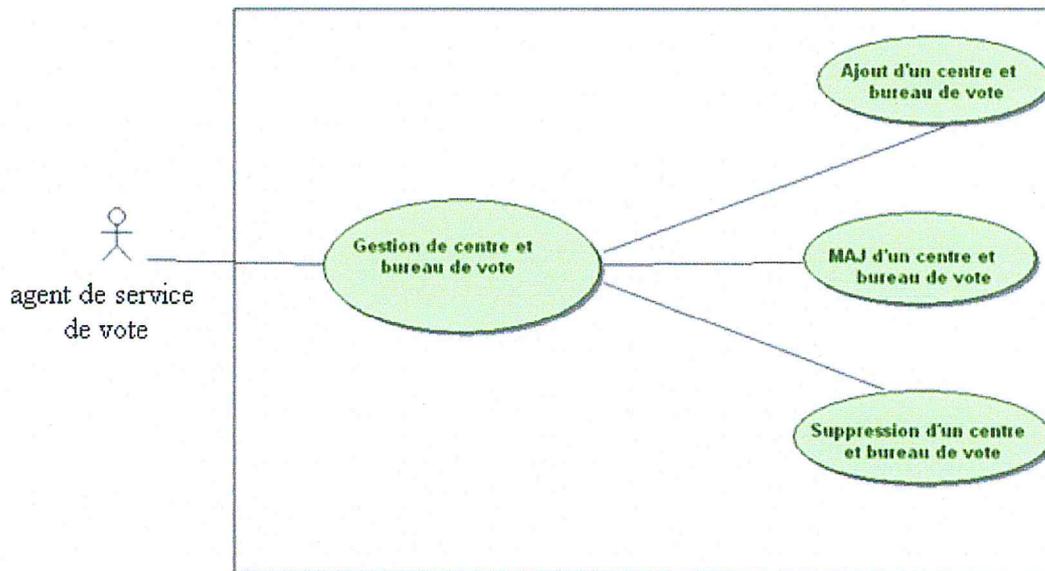


Figure 8:diagramme de cas d'utilisation pour la gestion des centres et bureaux de vote

3.5. Navigation géographique :

Dans ce niveau d'analyse, et dans ce cas d'utilisation on touche la relation entre l'acteur et le service de cartographie (la carte GOOGLE MAPS) et les différents services qui peuvent les exploiter, donc les grandes fonctionnalités de ce système et qui peut nous aider pour atteindre nos objectifs, on parle ici de service de navigation géographique, de parcourir la carte avec une forte souplesse et flexibilité était le premier choix de l'excellente API de Google maps et de leur service de cartographie ou l'acteur peut parcourir la carte manuellement ou bien utiliser le moteur de recherche de la carte géographique, afin de visualiser un lieu bien précise ,on parle aussi d'un puissant service de cette API qui est la géo localisation et qui permet de transformer une adresse on un couple de coordonnées géographique (latitude, longitude), ainsi que le service de calcul d'itinéraire qui représente la pièce maîtresse de notre travail, et qui permet de donner la distance entre deux points géographiques avec le chemin le plus intéressant entre eux.

❖ **Géo localisation automatique :**

Avant d'entamer l'opération de l'affectation selon la position géographique, il est nécessaire de passer sur cette étape de géo localisation afin de connaître la position géographique de diverses adresses (adresses de domiciles et les adresses des différents centres de vote). Et alimenter la BDD avec cette primordiale information qui est les coordonnées géo graphique (latitude, longitude).

❖ **Géo localisation manuelle:**

cette méthode est utiliser lors du rencontre d'un problème de mal reconnaissance d'adresse avec le moteur de recherche de la carte géographique d'où la nécessité de chercher cette adresse manuellement en parcourant la carte et pointer le curseur sur le point qui correspond à l'adresse demandée, puis alimenter la BDD les coordonnees geographiques.

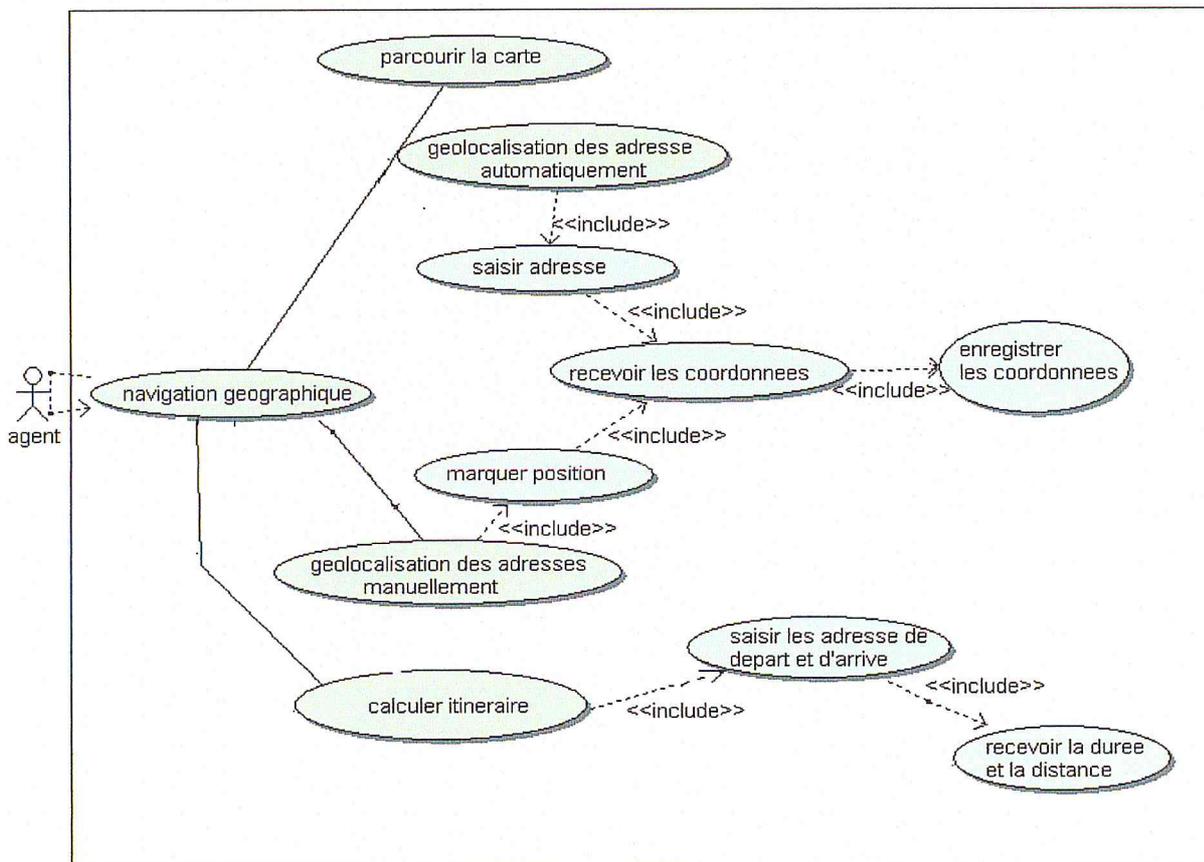


Figure 9:Diagramme de cas d'utilisation pour la navigation géographique

3.6. Affectation automatique :

Dans cette partie on s'intéresse au différentes taches exécuter par l'acteur principale pour arriver à l'objectif souligné qui est affecter le citoyen votant au centre de vote le plus proche à lui, et après avoir fait l'inscription on temps que citoyen votant et la reconnaissance de l'adresse du domicile de citoyen, il y à l'opération d'affectation qui inclus la géo localisation de tous les adresses concernées, soit automatiquement ou manuellement , et le calcul d'itinéraire entre les différents acteurs(citoyen et centres),et le choix de plus court chemin on question de temps et de parcours, le citoyen votant est affecter au centre de vote le plus approprié pour lui.

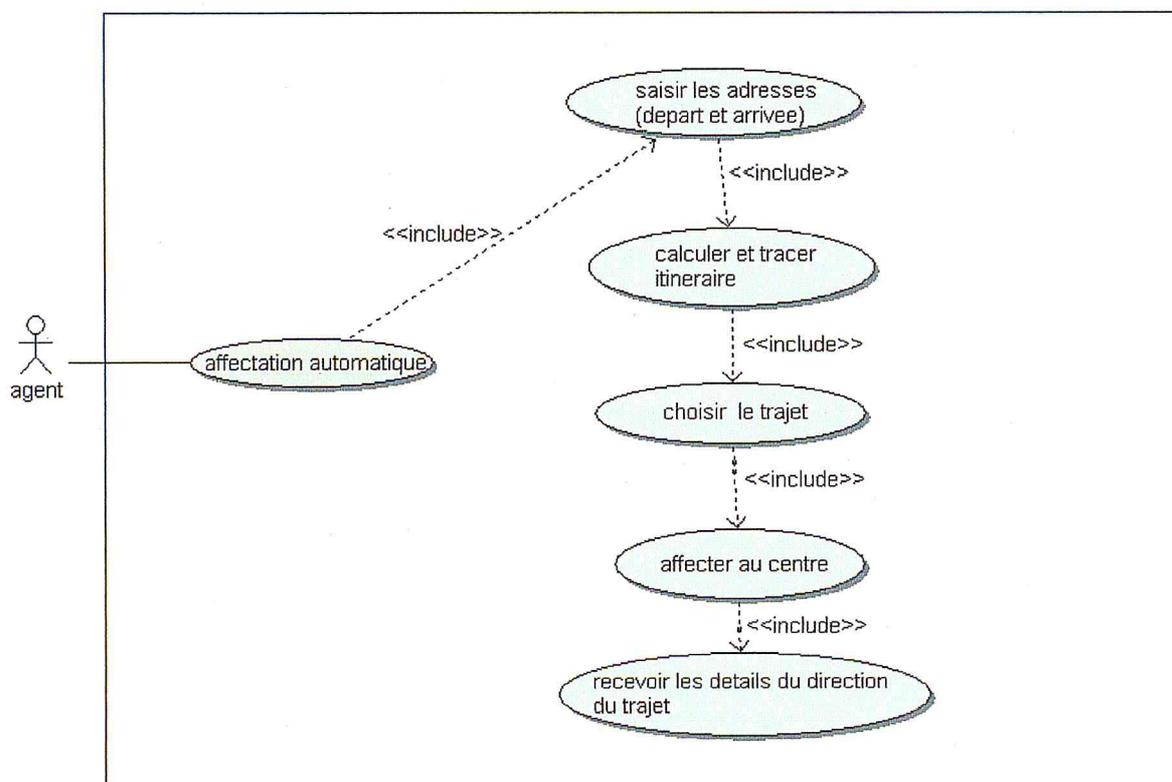


Figure 10:diagramme de cas d'utilisation pour l'affectation automatique

3.7. Réajustement d'affectation :

A ce niveau d'analyse et pour assurer des réponse à toutes les questions vis-à-vis les lacunes qu'on peut les faire face, et après une étude approfondie de système de vote; on particulier le module d'affectation des votants au centres et bureaux de vote, il était nécessaire de penser à gérer tous les cas de figure qui peuvent arriver, et compris

le cas de demande de changement de lieu de vote de la part de citoyen votant, à cause de plusieurs raisons, comme le changement de résidence par exemple, d'où le processus de réajustement qui permet le re-routage du votant à un nouveau centre de vote et effacement de ce dernier de la liste des votants dans l'ancien centre, un autre point qui semble à être aussi important qui est la désaffectation globale et qui permet de relancer le processus d'affectation pour établir des listes selon ce qui existe réellement. Par exemple augmentation du nombre de centre ou réduction, augmentation du nombre de bureau et du nombre d'Urne ou réduction... Etc., sans oublier le cas d'utilisation d'affectation automatique des votants aux centres et aux bureaux de vote.

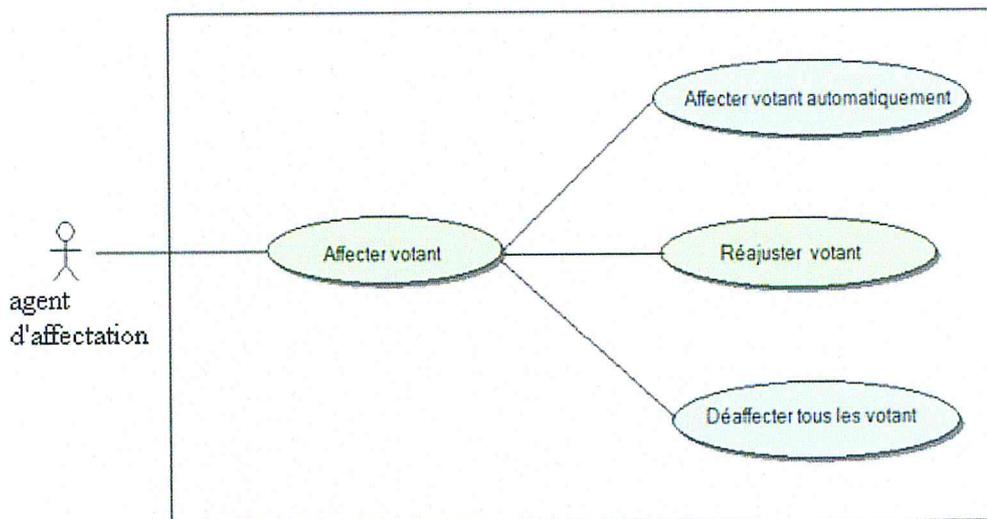


Figure 11:diagramme de cas d'utilisation pour la réajustement d'affectation

3.8. Gestion des staffs :

Dans cette partie de gestion, on s'intéresse à la gestion des personnes censé gérer le processus électoral, d'un coté administratif on parle des personne dédiés a l'organisation logistique du processus électoral et l'arrangement des votants ainsi que l'assurance du bon fonctionnement du processus, et aussi d'un coté de supervision des élections, donc ce qui concerne les organisations locales et internationales intéressées par la supervision du processus électoral.

Dans cette partie, nous allons détailler les différentes taches exécutées par l'agent de service de vote pour gérer le staff de processus électoral, qui sont:

- ✓ Ajout de nouvel élément de staff.
- ✓ mise à jour d'élément de staff.

- ✓ Suppression d'un élément de staff.

Voici le diagramme cas d'utilisation qui présente cette gestion.

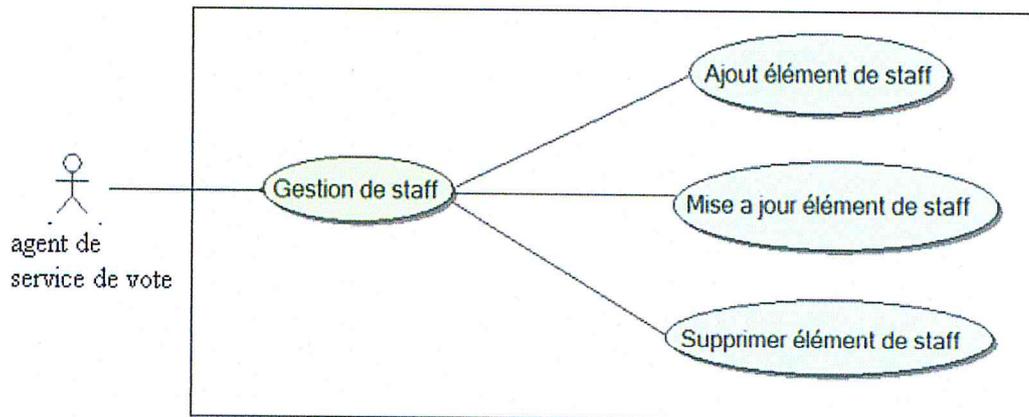


Figure 12:diagramme de cas d'utilisation pour la gestion des staffs

Conclusion :

Dans ce chapitre nous avons donné une vue générale sur notre système à l'aide des Diagrammes de cas d'utilisation de langage UML et différents acteurs du logiciels.

IV

Conception de l'application

Introduction

Diagramme de séquence

Diagramme d'activité

Diagramme de classe

Conclusion

1. Introduction :

Modéliser un système avant sa réalisation permet de mieux comprendre son fonctionnement. Après avoir spécifié les besoins de notre système d'une manière formelle sous forme de diagrammes de cas d'utilisation, nous présenterons dans ce chapitre à l'aide des diagrammes UML toutes les étapes pour la conception et la réalisation de notre système.

On va essayer d'associer chaque cas d'utilisation par un diagramme de séquence pour montrer la succession chronologique des opérations réalisées par un acteur, et par un diagramme d'activité pour montrer l'enchaînement des activités et de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation.

Enfin nous arriverons au diagramme de classe qui représente la structure statique de notre système.

2. Conception du système :

Notre système sera conçu en respectant le modèle de conception MVC (Modèle Vue Contrôleur). MVC a pour objectif d'organiser la réalisation de l'application et à séparer complètement la présentation (Vue) des données (Modèle) et de la manière de leur enregistrement (Contrôleur).

On définit le rôle des trois entités de la façon suivante :

Le modèle : Il représente les données de l'application. Il définit aussi l'interaction avec la base de données.

La vue : qui s'occupe de l'interaction homme machine.

Le contrôleur : est chargé de la synchronisation du modèle et de la vue. Il reçoit toutes les informations de l'utilisateur, et déclenche les traitements correspondants. Les traitements peuvent à leur tour solliciter la vue ou le modèle.

Donc l'architecture MVC peut être représentée par le schéma suivant :

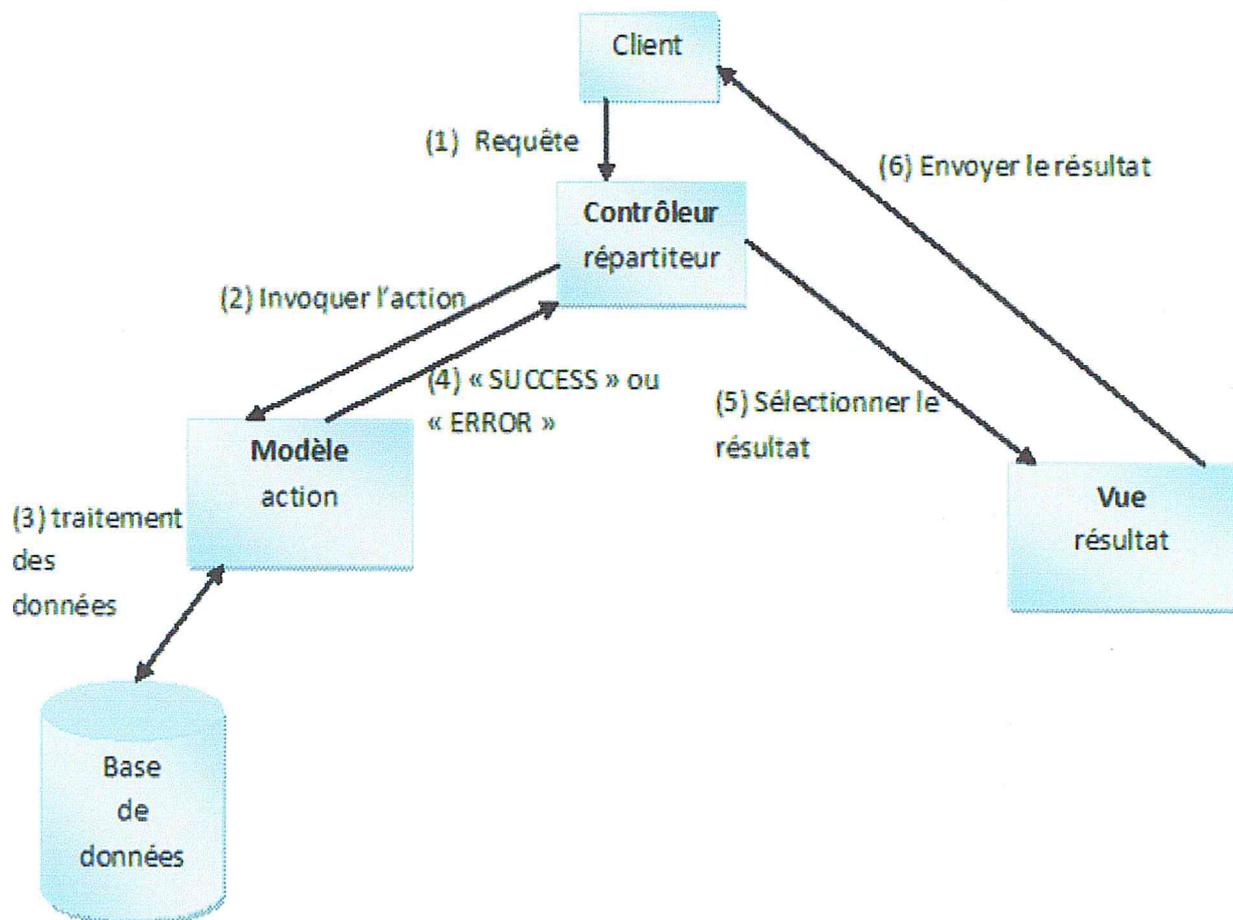


Figure 13:l'architecture MVC

3. Description des cas d'utilisation:

3.1. Gestion des comptes

3.1.1 cas d'utilisation : "créer un compte utilisateur"

Généralité :

Cas d'utilisation	Créer un compte utilisateur
Acteur	Utilisateur du système
But	Accès aux différentes fonctions du système
Résumé métier	Avant de faire n'importe quelle fonction du système l'utilisateur doit d'abord être authentifié
Pré condition	L'utilisateur remplit le formulaire pour créer le compte
Post condition	Création effectuée

Scenario :

1. L'administrateur demande la création d'un compte.
2. Le contrôleur demande au vue d'afficher le formulaire de création d'un compte
3. Le vue affiche le formulaire
4. L'administrateur remplit le nom d'utilisateur et les information et envoie le formulaire
5. Le contrôleur demande au model de faire la vérification des champs
6. Le model fait la vérification demandé
7. S'il trouve que le nom d'utilisateur et déjà utiliser pour un autre compte
 - ✓ Il retourne l'erreur au contrôleur
 - ✓ Le contrôleur demande au vue d'afficher un message d'erreur
 - ✓ La vue présente le formulaire à l'administrateur avec un message d'erreur informant qu'il faut changer le nom d'utilisateur car il est utilisé
8. Sinon
 - ✓ Le contrôleur demande au vue d'afficher un message de succès
 - ✓ La vue affiche le message de succès

3.1.1.1 Diagramme de séquence : « créer compte utilisateur »

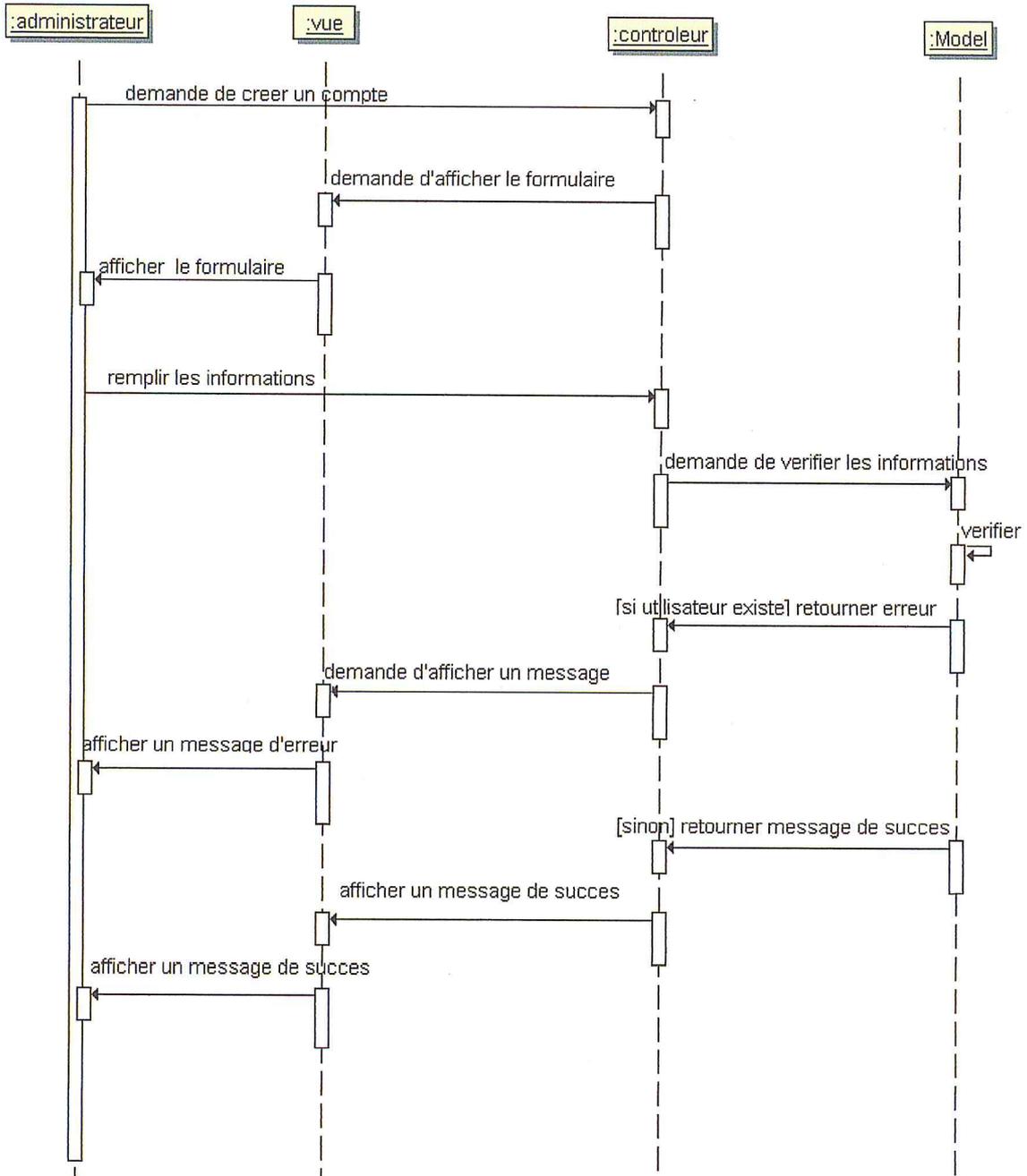


Figure 14:diagramme de séquence pour la création d'un compte

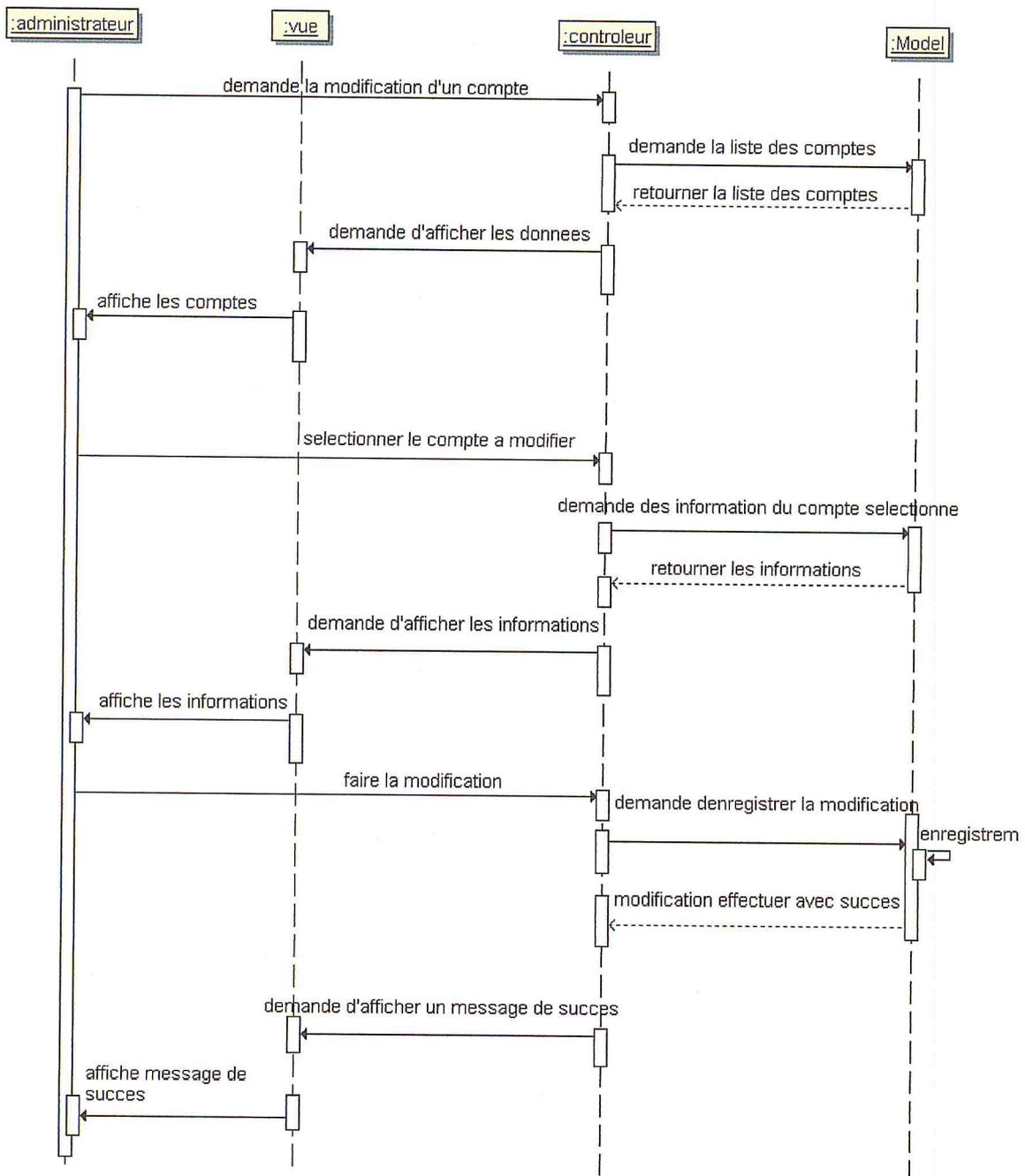
3.1.2. Cas d'utilisation : « modifier un compte utilisateur »

Généralité :

Cas d'utilisation	modifier un compte utilisateur
Acteur	Utilisateur du système
But	modifier les informations du compte utilisateur
Résumé métier	Faire les MAJ d'un compte
Pré condition	L'utilisateur remplit le formulaire pour modifier le compte
Post condition	modification effectuer

Scenario :

1. L'administrateur demande la modification d'un compte
2. Le contrôleur reçoit la demande et demande au model de donner la liste des comptes qui existent dans la base de données.
3. Le model retourne au contrôleur la liste des compte demandes
4. Le contrôleur demande au vue d'afficher les données retournées
5. La vue affiche les comptes existe dans la base de données
6. L'administrateur sélectionne la modification d'un compte et envoie au contrôleur
7. Le contrôleur demande au model de donner les informations qui concerne le compte sélectionné
8. Le model retourne au contrôleur les informations retournées
9. Le contrôleur demande au vue d'afficher les données retournées
10. La vue affiche le formulaire
11. L'administrateur effectue la modification et envoie le formulaire
12. Le contrôleur demande au model d'enregistrer la modification
13. Le model effectue l'enregistrement
14. Le model envoie le message de succès au contrôleur
15. Le contrôleur demande au vue d'afficher un message de succès
16. Le vue affiche le message de succès



3.1.2.1.diagramme de séquence : « modification d'un compte utilisateur »

Figure 15:diagramme de séquence pour la modification des comptes utilisateur

3.1.3 Cas d'utilisation : « supprimer un compte »

Généralité :

Cas d'utilisation	supprimer un compte utilisateur
Acteur	Utilisateur du système
But	supprimer les informations du compte utilisateur
Résumé métier	Faire les MAJ d'un compte
Pré condition	L'utilisateur remplit le formulaire pour supprimer le compte
Post condition	suppression effectuer

Scenario :

1. L'administrateur demande la suppression d'un compte
2. Le contrôleur reçoit la demande et demande au model de donner la liste des comptes qui existent dans la base de données
3. Le model retourne au contrôleur la liste des comptes demandés
4. Le contrôleur demande au model d'afficher la liste des comptes
5. La vue affiche une page contient un tableau des comptes qui existent dans la base de données
6. L'administrateur sélectionne la suppression d'un compte bien précis et envoie la requête au contrôleur
7. le contrôleur demande au model de supprimer le compte sélectionné
8. Le model effectue la suppression avec succès
9. Il informe le contrôleur que tout c'est déroulé avec succès
10. Le contrôleur demande au vue d'afficher un message de succès
11. La vue affiche un message à l'administrateur informant que la suppression du compte a effectué avec succès

3.1.3.1 diagramme de séquence : « suppression d'un compte utilisateur »

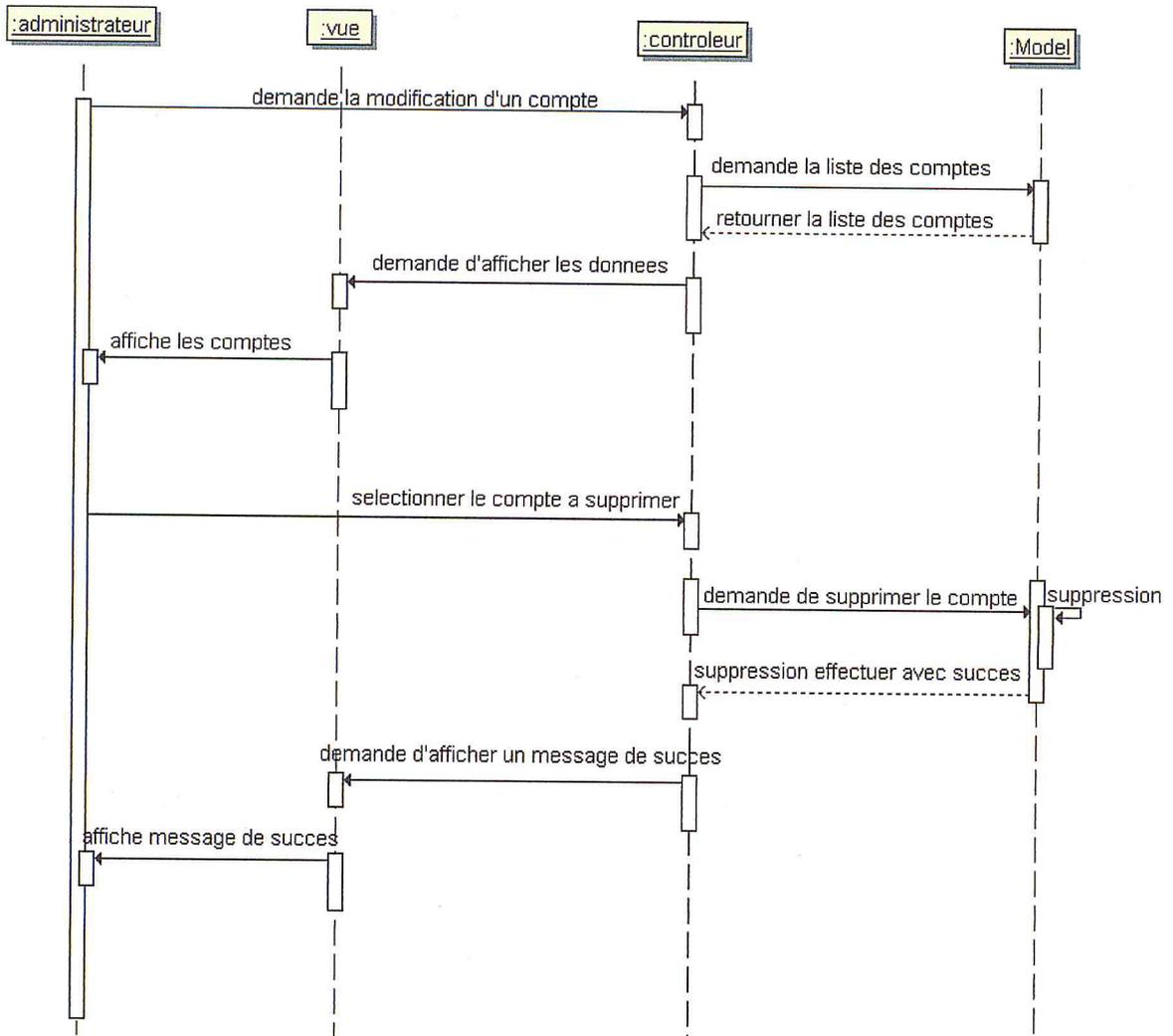


Figure 16:diagramme de séquence pour la suppression d'un compte utilisateur

3.1.4. diagramme d'activité : « gestion des comptes»

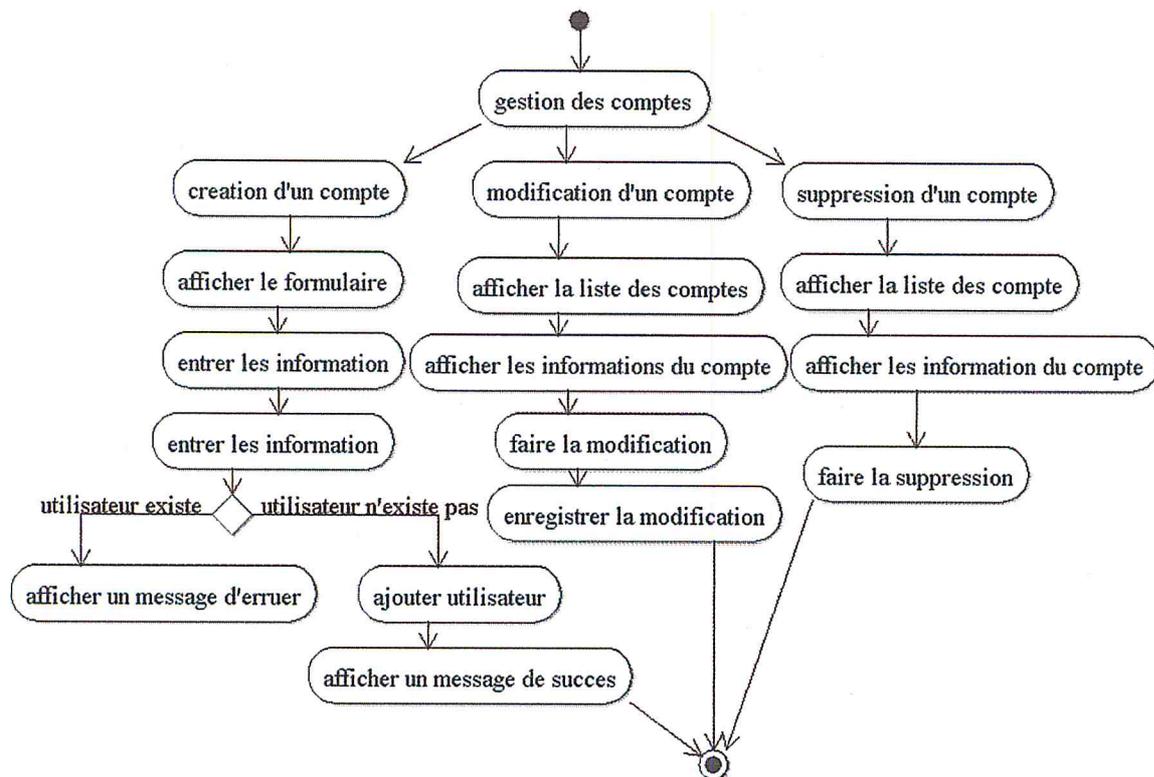


Figure 17:diagramme de séquence pour la gestion des comptes

3.2 Cas d'utilisation : "Authentification"

Généralité :

Cas d'utilisation	Authentification
Acteur	Utilisateur du système
But	Accès aux différentes fonctions du système
Résumé métier	Avant de faire n'importe quelle fonction du système l'utilisateur doit d'abord être authentifier
Pré condition	L'utilisateur entre son nom d'utilisateur et son mot de passe
Post condition	L'ouverture de session utilisateur

Scénario :

1. Le contrôleur demande au vue d'afficher la page d'accueil
2. La vue affiche une page d'accueil qui contient deux champs à remplir pour réaliser l'opération d'authentification (nom d'utilisateur et le mot de passe)
3. L'utilisateur entre ses informations et envoie le formulaire (reçus par le contrôleur)

4. Le contrôleur envoie les informations au model pour valider les informations donné par l'utilisateur
5. Le model lance la recherche selon le nom d'utilisateur et le mot de passe fournit par l'utilisateur
6. Si le résultat de la recherche est positive
 - ✓ Le model informe le contrôleur que les informations fournit sont correctes et retourne le profil associes á ces informations
 - ✓ Le contrôleur demande au vue d'ouvrir une session
 - ✓ La vue ouvre une session et affiche á l'utilisateur une page qui contient les actions associer au profil retourné
- 6 .Si non
 - ✓ Le model informe le contrôleur que les informations fournit ne sont pas correctes et retourne l'erreur au contrôleur.
 - ✓ Le contrôleur demande au vue de présenter une page avec un message d'erreur
 - ✓ La vue présente le formulaire avec un message d'erreur informant qu'un aucun utilisateur a été trou

3.3. Gestion des votants

3.3.1 Cas d'utilisation : "Ajout de votant"

Généralité :

Cas d'utilisation	Ajout de votant
Acteur	Agent de service de vote
description	Lorsque la période de révision annuelle termine et après mise à jour des listes des votants l'acteur peut faire l'opération de l'ajout de l'ensemble finale des votants.
Précondition	L'agent de service de vote est identifié.
Post condition	L'opération de l'ajout des votants est effectuée.

Scenario :

1. l'agent de service de vote s'identifie pour accéder à son espace.
2. Le système accepte l'accès de l'agent de service de vote.
3. L'agent de service de vote demande au système d'ajouter les votants inscrits dans la nouvelle opération de vote.
4. Le contrôleur demande au vue d'afficher le formulaire d'ajout d'un nouveau votant.
5. La vue affiche le formulaire.
6. L'agent remplit le formulaire qui contient des informations concernant le nouveau votant concerné par le nouveau processus de vote tel que (numéro d'extrait de naissance, lieu de naissance, nom de votant, prénom de votant...) et envoie le formulaire l'ajout d'un nouveau votant.
7. S'il trouve que la codification n'a été pas respectée.
 - ✓ Il retourne l'erreur au contrôleur.
 - ✓ Le contrôleur demande au vue d'afficher un message d'erreur.
 - ✓ La vue présente un message d'erreur indiquant que la codification prédéfinie n'a été pas respecté.
8. Sinon
 - ✓ L'agent vérifie si l'adresse existe dans la liste déroulante.
 - Il termine la saisie des autres informations.
Sinon
 - L'agent doit saisir la nouvelle adresse pour l'enregistrer dans la BDD.
9. Si l'agent saisie toutes les informations
 - ✓ L'agent de service de vote doit valider.

10. Le contrôleur reçoit le formulaire et demande au model de chercher si le votant est déjà ajouter.
11. Le model fait la recherché demandé.
12. S'il trouve que le votant existe déjà
 - ✓ Il retourne l'erreur au contrôleur.
 - ✓ Le contrôleur demande au vue d'afficher un message d'erreur.
 - ✓ La vue présente un message d'erreur indiquant que la déclaration existe.
13. Sinon
 - ✓ La vue présente un message indiquant que l'opération est affectée avec sucée.

3.3.1.1 diagramme de séquence : « ajout d'un votant »

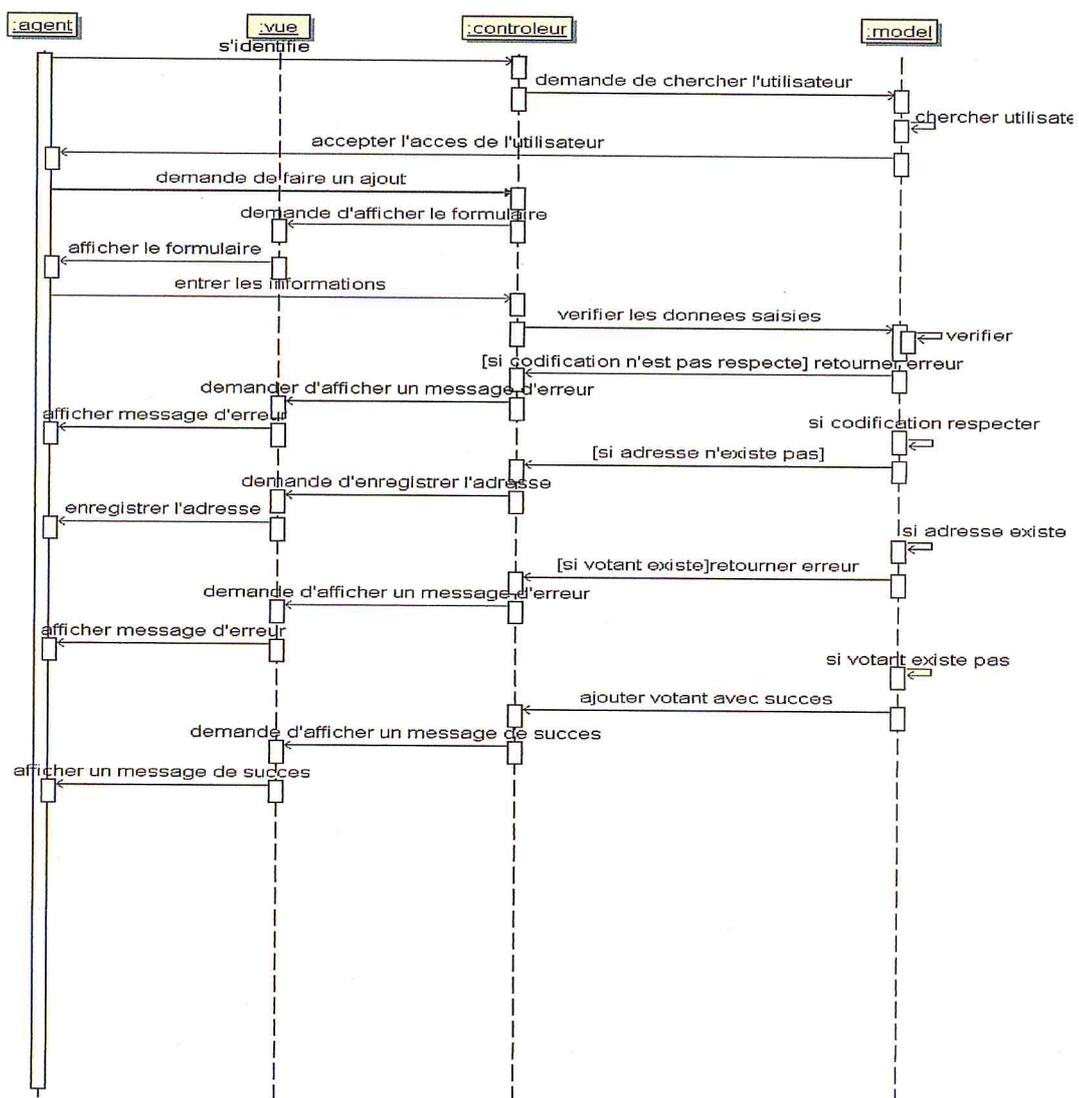


Figure 20:diagramme de séquence pour l'ajout d'un votant

3.3.1.2 Diagramme d'activité : « ajout d'un votant »

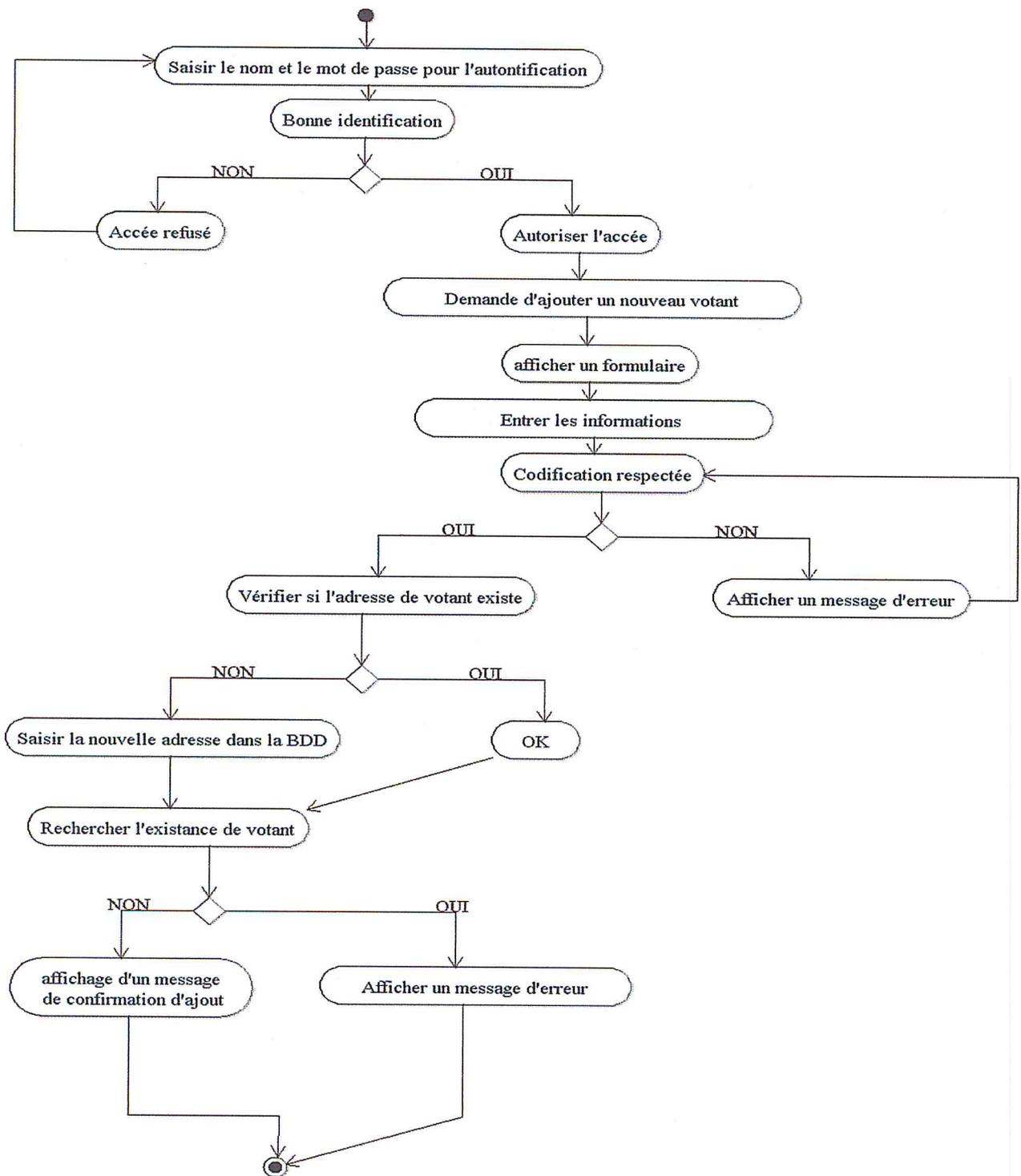


Figure 21:diagramme d'activité pour ajouter un votant

3.3.2. Cas d'utilisation : « Modifier les informations d'un votant »

Généralité :

Cas d'utilisation	Modifier les informations concernant un votant
Acteur	L'agent de service de vote
description	Après avoir ajouté le nouveau votant ; et a cause de changement de résidence par le votant, les informations peuvent être modifiées (la modification est limitée sur des champs bien précises.)
Pré condition	L'agent de service de vote est identifié.
Post condition	Les informations après modification sont enregistrées.

Scenario :

1. l'agent de service de vote s'identifie pour accéder à son espace.
 2. Le système accepte l'accès de l'agent de service de vote.
 3. L'agent de service de vote demande au système de modifier des informations bien précises d'un votant.
 4. Le contrôleur demande d'afficher le formulaire adéquat.
 5. La vue affiche le formulaire.
 6. L'agent remplit le formulaire qui contient des informations concernant le votant (identifiant, prénom...) après modification envoie le formulaire.
 7. S'il trouve que la codification n'a été pas respectée.
 - ✓ Il retourne l'erreur au contrôleur.
 - ✓ Le contrôleur demande à la vue d'afficher un message d'erreur.
 - ✓ La vue présente un message d'erreur indiquant que la codification prédéfinie n'a été pas respectée.
 8. Sinon
 - ✓ L'agent vérifie si l'adresse existe dans la liste déroulante.
 - Il termine la saisie des autres informations.
- Sinon
- L'agent doit saisir la nouvelle adresse pour l'enregistrer dans la BDD.
9. Si l'agent saisit toutes les informations.
 - L'agent de service de vote doit valider.
 10. Le contrôleur reçoit le formulaire et demande au modèle de chercher si le votant existe.
 11. Le modèle fait la recherche demandée.
 12. S'il trouve que le votant existe.

- ✓ Il fait la modification demande.
- ✓ La vue affiche un tableau contient les détails retourné par le model et qui sont enregistrés dans la BDD.

13. Sinon

- ✓ Il retourne l'erreur au contrôleur.
- ✓ Le contrôleur demande au vue d'afficher un message d'erreur.
- ✓ La vue présente un message d'erreur indiquant que le votant n'existe pas.

3.3.2.1. Diagramme de séquence : « modifier les informations d'un votant »

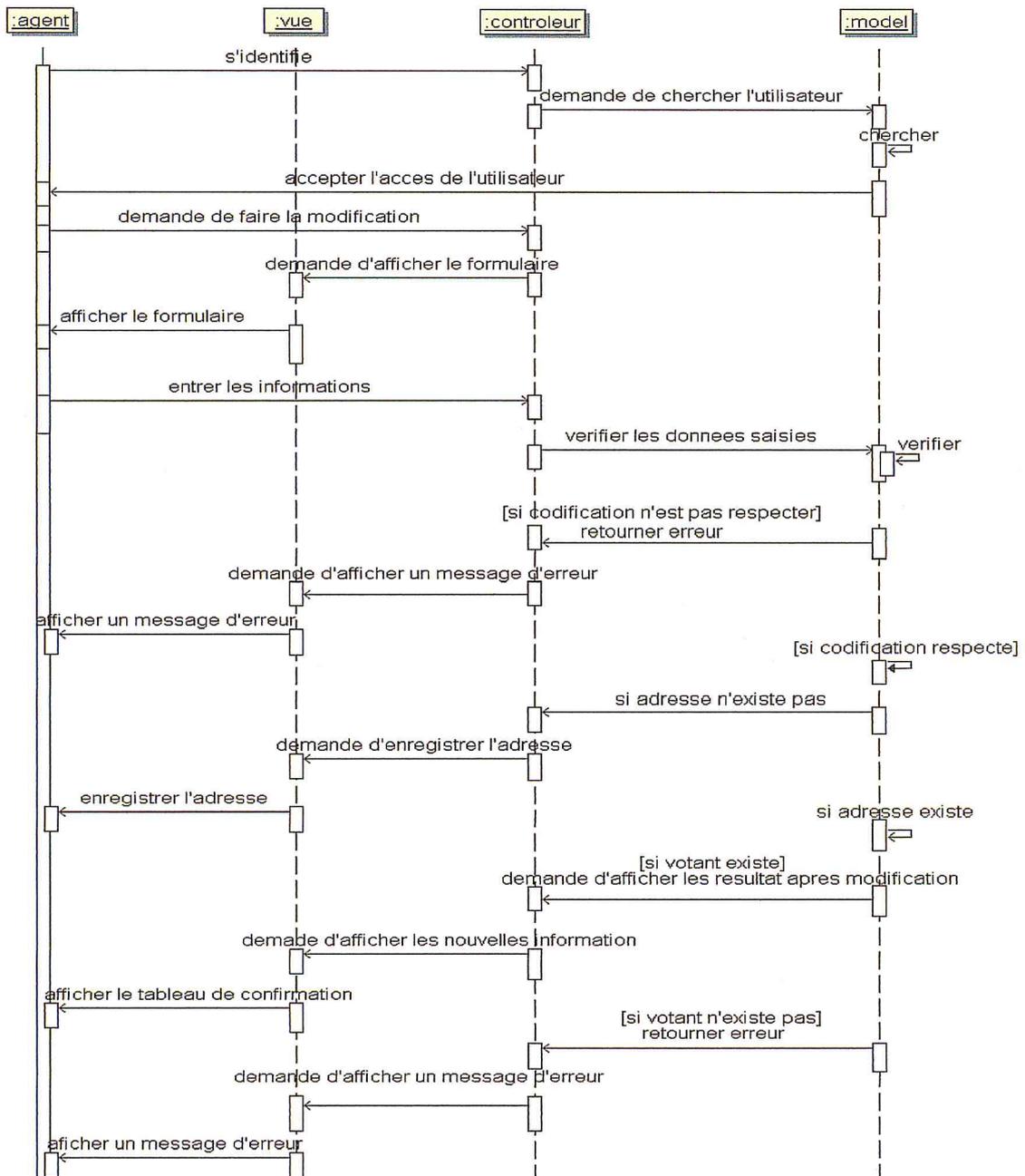


Figure 22:diagramme de séquence pour la modification d'un votant

3.3.2.2 Diagramme d'activité : « modifier les informations d'un votant »

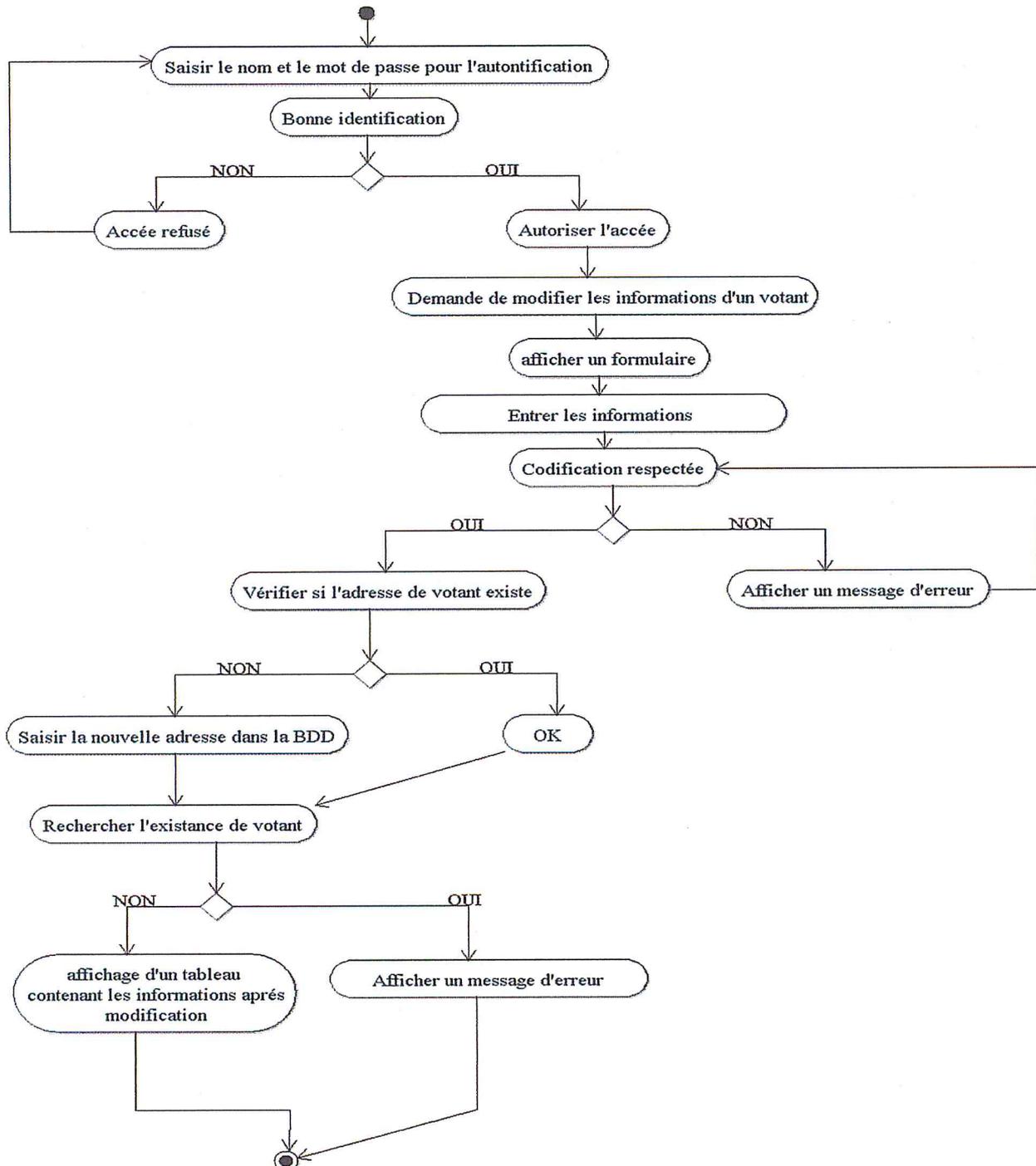


Figure 23:diagramme d'activité pour la modification d'un votant

3.3.3 Cas d'utilisation : « supprimer un votant »

Généralité :

Cas d'utilisation	Supprimer un votant
Acteur	L'agent de service de vote
Description	L'opération est effectuée dans des cas qui sont vraiment limité; et en premier lieu après la déclaration de décès de votant ou on va le radier temporairement avant de le radier définitivement.
Pré condition	L'agent de service de vote est identifié.
Post condition	Le votant est supprime.

Scenario :

1. l'agent de service de vote s'identifie pour accède a son espace.
 2. Le système accepte l'accède de l'agent de service de vote.
 3. L'agent de service de vote demande au système de supprimer un votant.
 4. Le contrôleur demande d'afficher le formulaire adéquat.
 5. La vue affiche le formulaire.
 6. L'agent remplit le formulaire qui contient l'information concernant le votant (nom) et envoi le formulaire.
 7. Le contrôleur demande au model de chercher tous les votants qui ont le même nom qui est fournit par l'agent en vue de voir tous les résultats.
 8. Si le model trouve les votants demandé
 - ✓ L'agent Sélectionne le votant concerné.
 - ✓ Supprimer.
- Sinon
- ✓ Il retourne l'erreur au contrôleur.
 - ✓ Le contrôleur demande au vue d'afficher un message d'erreur.
 - ✓ La vue présente un message d'erreur indiquant que le votant n'existe pas.

3.3.3.1. Diagramme de séquence : « supprimer un votant »

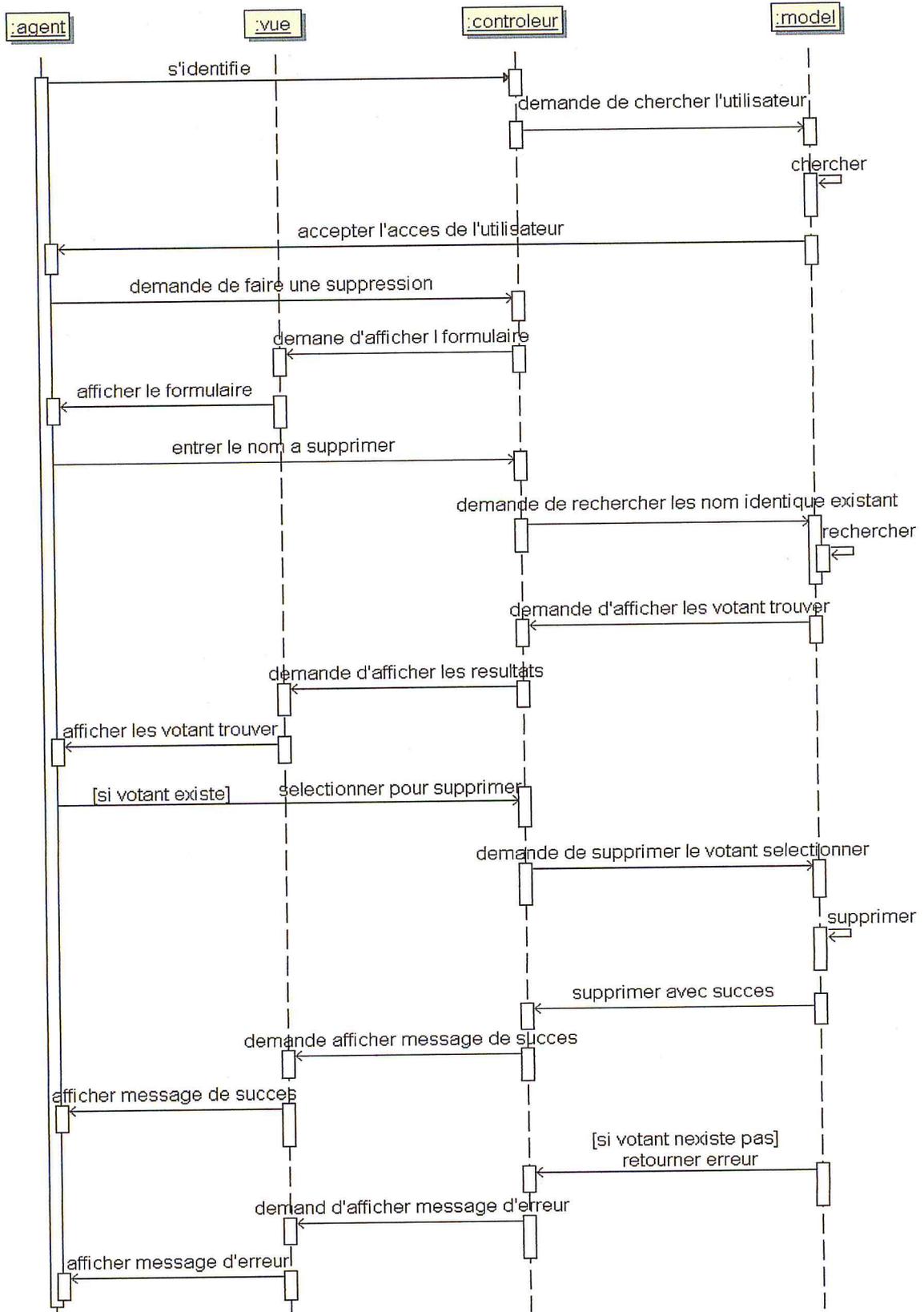


Figure 24:diagramme de séquence pour a suppression d'un votant

3.2.3.2. Diagramme d'activité : « supprimer un votant »

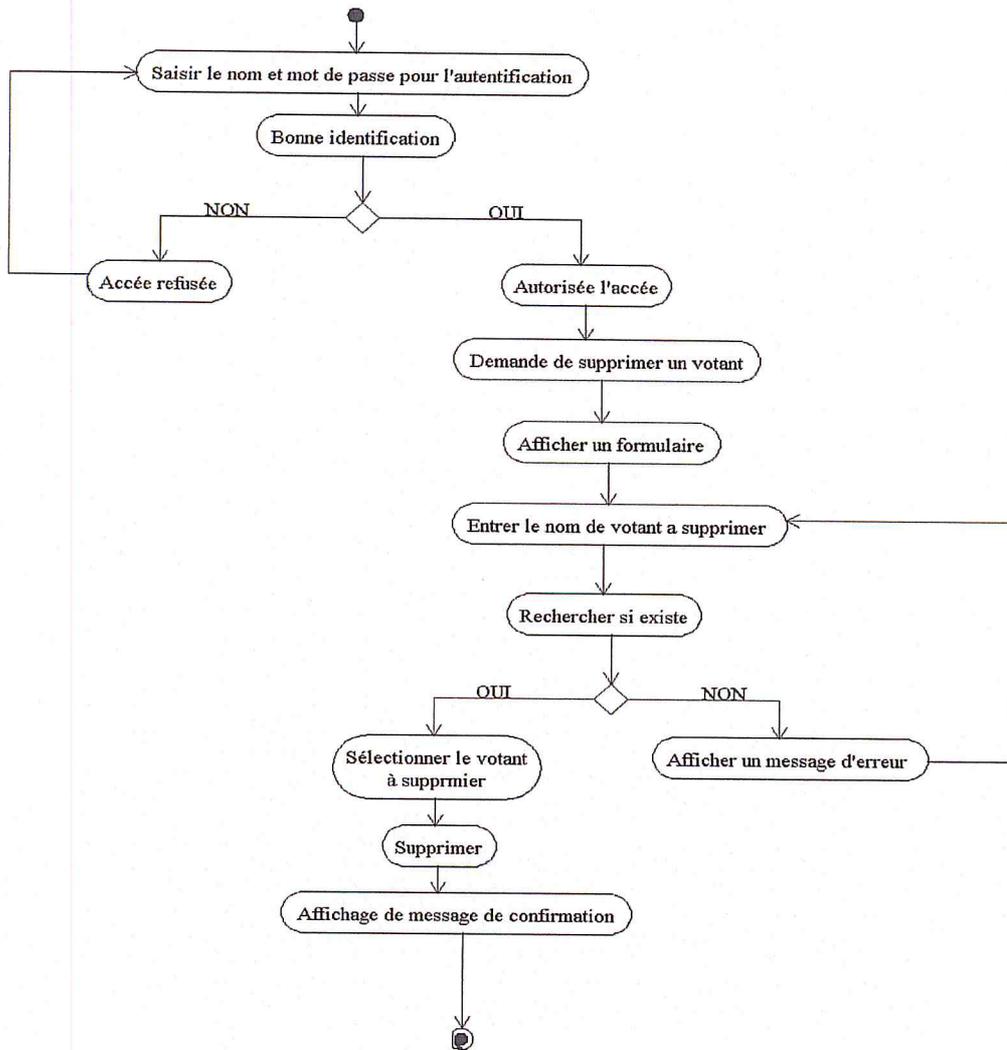


Figure 25:diagramme d'activité pour la suppression d'un votant

3.4. Gestion des centres et bureau de vote

3.4.1 Cas d'utilisation : « ajouter un Centre de vote »

Généralité :

Cas d'utilisation	Ajouter un centre de vote
Acteur	L'agent de service de vote
Description	Lorsqu'il y a un nombre très important de votant
Pré condition	L'agent de service de vote est identifié.
Post condition	Les informations sont enregistrées.

Scenario :

- 1- l'agent de service de vote s'identifie pour accéder à son espace.
- 2- le système accepte l'accès de l'agent.
- 3- l'agent demande au système d'afficher le formulaire pour ajouter un centre.
- 4- le contrôleur demande au vue d'afficher le formulaire.
- 5- la vue affiche le formulaire
- 6- l'agent saisie les informations du centre.
- 7- le contrôleur reçoit le formulaire et demande au model de chercher si le centre existe déjà :
- 8- le model fait la recherche demande
- 9- s'il trouve que le centre existe
 - ✓ Il tourne l'erreur au contrôleur
 - ✓ Le contrôleur demande au vue d'afficher un message d'erreur
 - ✓ La vue présente le formulaire avec un message d'erreur indiquant que le centre existe.
- 10- sinon
 - ✓ Le contrôleur demande au vue d'afficher le formulaire pour ajouter les bureaux de vote au centre

- ✓ La vue affiche le formulaire
- ✓ L'agent saisie les information du bureau.

3.4.1.1 Diagramme de séquence : « ajouter un centre de vote »

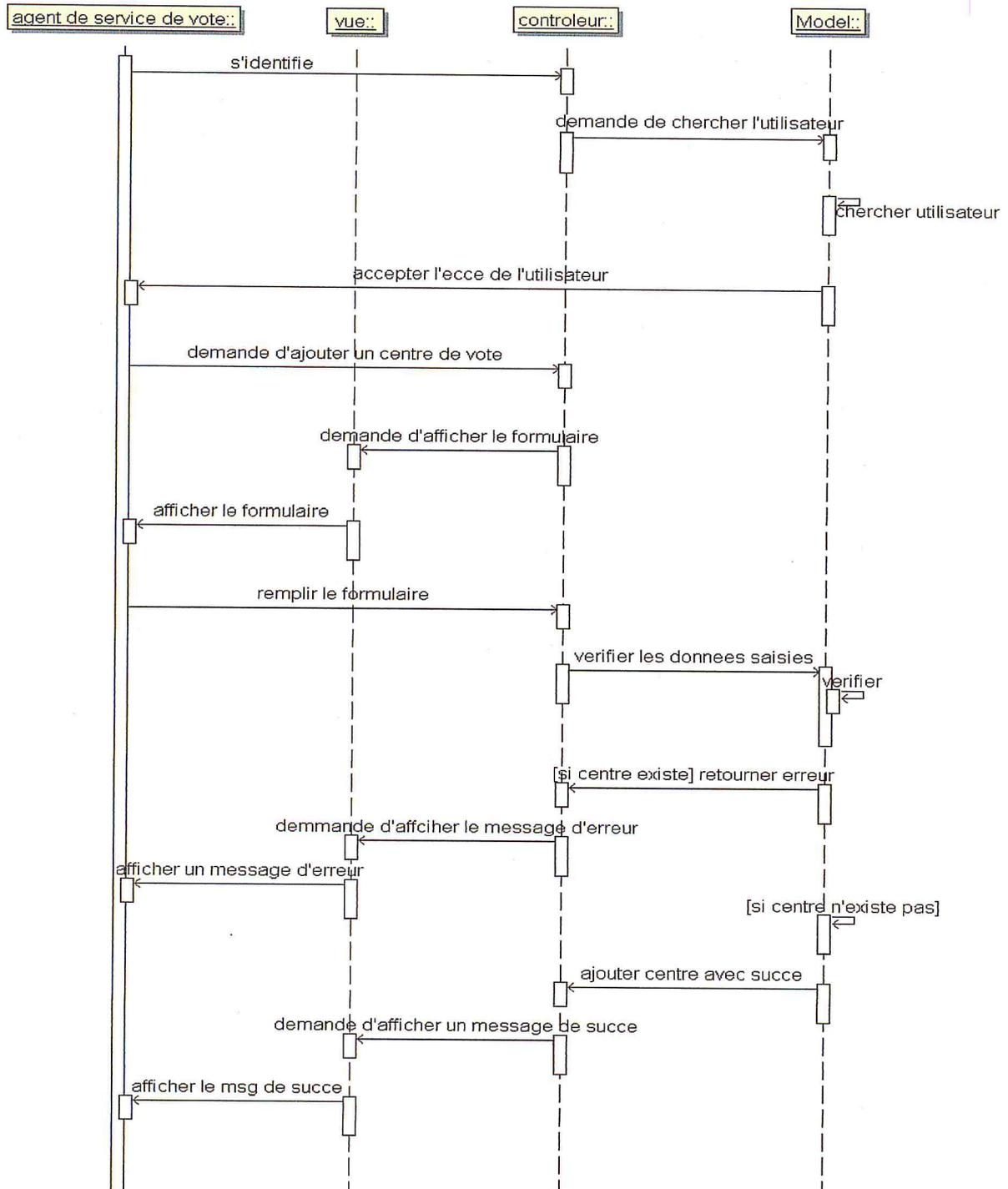


Figure 26: Diagramme de séquence pour l'ajout d'un centre

3.4.2 Cas d'utilisation : « ajouter un bureau de vote »

Généralité :

Cas d'utilisation	Ajouter un centre de vote
Acteur	L'agent de service de vote
Description	Lorsqu'il y a un nombre très important de votant
Pré condition	L'agent de service de vote est identifié.
Post condition	Les informations sont enregistrées.

Scenario :

1-l'agent de service de vote s'identifie pour accéder a son espace.

2-le système accepte l'accès de l'agent.

3-l'agent demande au système d'afficher le formulaire pour ajouter un bureau au centre de vote.

4-le contrôleur demande au vue d'afficher le formulaire.

5-la vue affiche le formulaire

6-l'agent saisie les informations du bureau.

7-le contrôleur reçoit le formulaire et demande au model de chercher si le bureau existe déjà :

8- le model fait la recherche demande

9- s'il trouve que le bureau existe

- ✓ Il tourne l'erreur au contrôleur
- ✓ Le contrôleur demande au vue d'afficher un message d'erreur
- ✓ La vue présente le formulaire avec un message d'erreur indiquant que le bureau existe.

10- sinon

- ✓ Le contrôleur demande au vue d'afficher un message de succès
- ✓ La vue présente le formulaire avec un message de succès.

3.4.2.1 Diagramme de séquence : « ajouter un Bureau de vote »

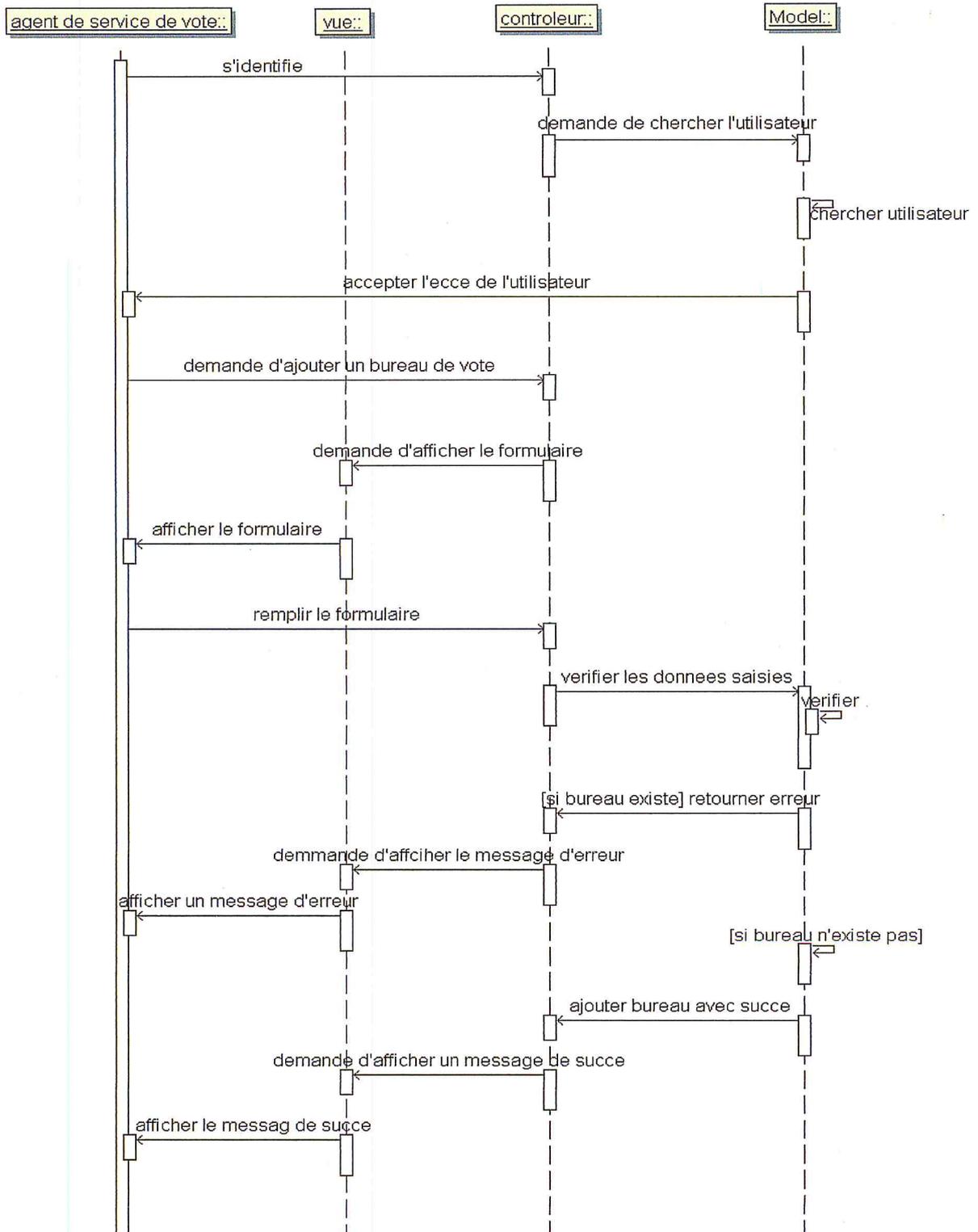


Figure 27:diagramme de séquence pour le l'ajout d'un bureau de vote

3.4.2.2 Diagramme d'activité pour l'ajout d'un centre et bureau de vote :

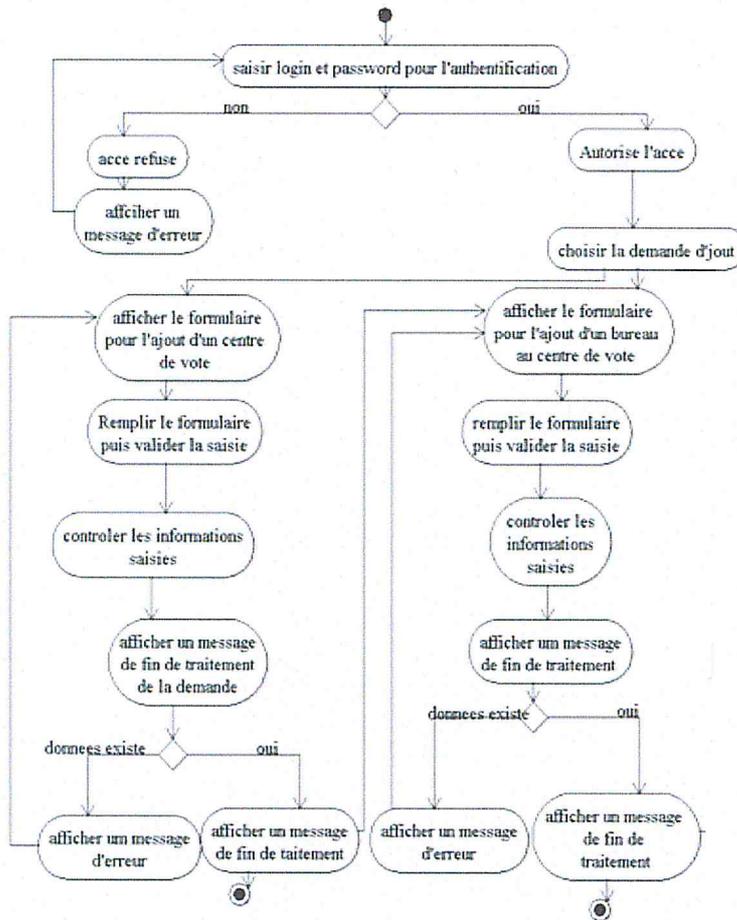


Figure 28:diagramme d'activité pour l'ajout d'un centre et bureau de vote

3.4.3 Cas d'utilisation : « modifier un centre de vote »

Généralité :

Cas d'utilisation	Modifier les informations concernant un centre
Acteur	L'agent de service de vote
Description	Après avoir ajouter un nouveau centre de vote les informations peuvent être modifier tel que (capacité d'un centre de vote)
Pré condition	L'agent de service de vote est identifié.
Post condition	Les informations après modification sont enregistrés.

Scenario :

- 1-l'agent de service de vote s'identifie pour accéder a son espace.
- 2-le système accepte l'accès de l'utilisateur.
- 3-l'agent demande au système de faire une demande de modification
- 4-le contrôleur demande au vue d'afficher le formulaire
- 5-la vue affiche le formulaire
- 5-le système contrôle les informations saisies
- 6-le système affiche un message que la demande est faite

3.4.3.1 Diagramme de séquence : « modifier un centre de vote »

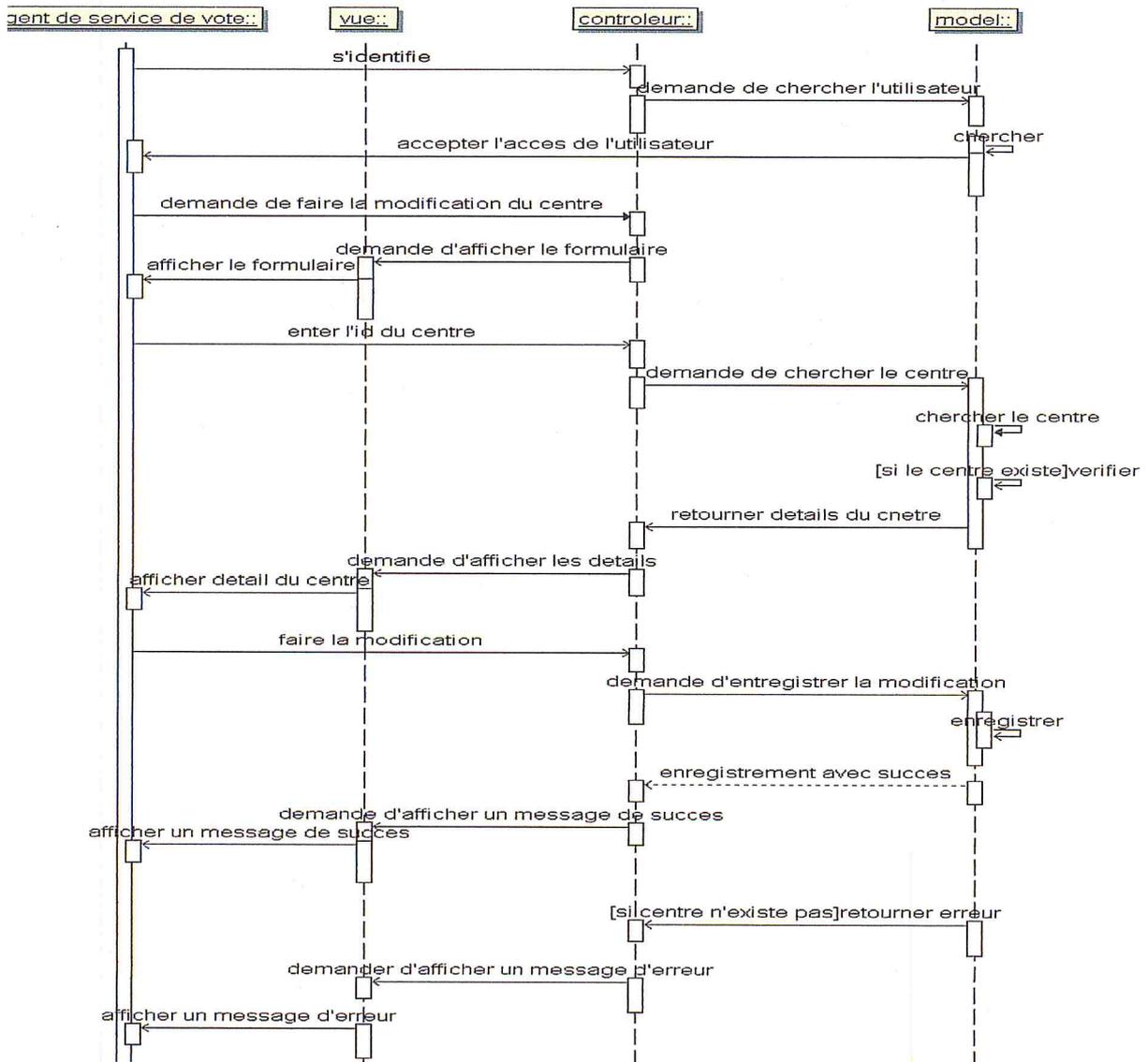


Figure 29:diagramme de sequence pour modifier un centre de vote

3.4.3.2. Diagramme d'activité : « modifier un centre de vote »

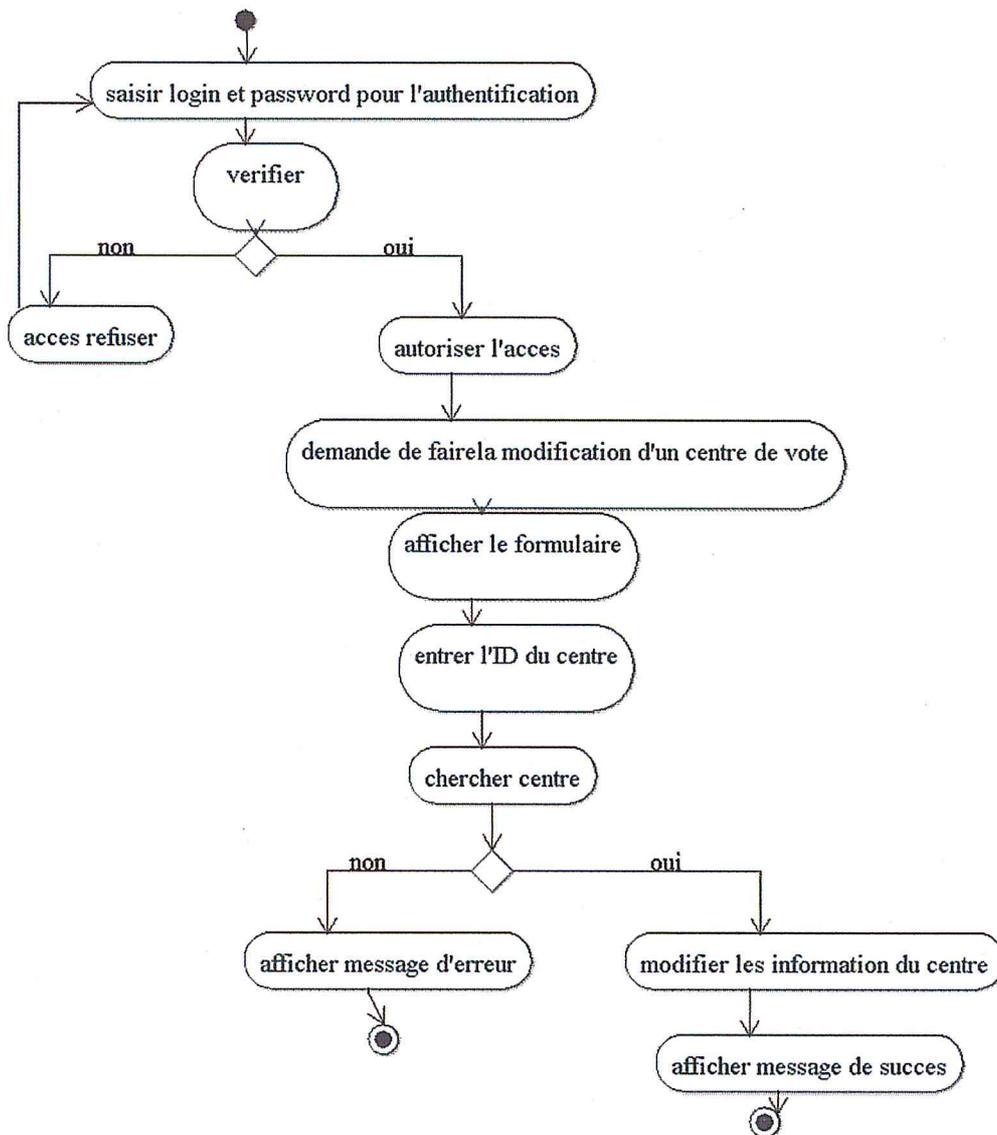


Figure 30:diagramme d'activité pour la modification d'un centre

3.4.4. Cas d'utilisation : « supprimer un centre de vote »

Généralité :

Cas d'utilisation	Supprimer un centre de vote
Acteur	L'agent de service de vote
Description	Cette opération se fait dans des cas très limite
Pré condition	L'agent de service de vote est identifié.
Post condition	Les enregistrements ont été supprimés.

Scenario :

1. l'agent de service de vote s'identifie pour accède a son espace.
2. Le système accepte l'accès de l'agent de service de vote.
3. L'agent de service de vote demande au système de supprimer un centre de vote
4. Le contrôleur demande au vue d'afficher le formulaire de suppression d'un centre de vote
5. La vue affiche le formulaire.
6. L'agent entre l'ID du centre
7. le contrôleur demande au model de chercher le centre pour le supprimer
8. si centre n'existe pas afficher un message d'erreur
9. sinon
 - ✓ Supprimer le centre et ses bureaux
 - ✓ Afficher un message de succès

3.4.4.1 Diagramme de séquence : « supprimer un centre de vote »

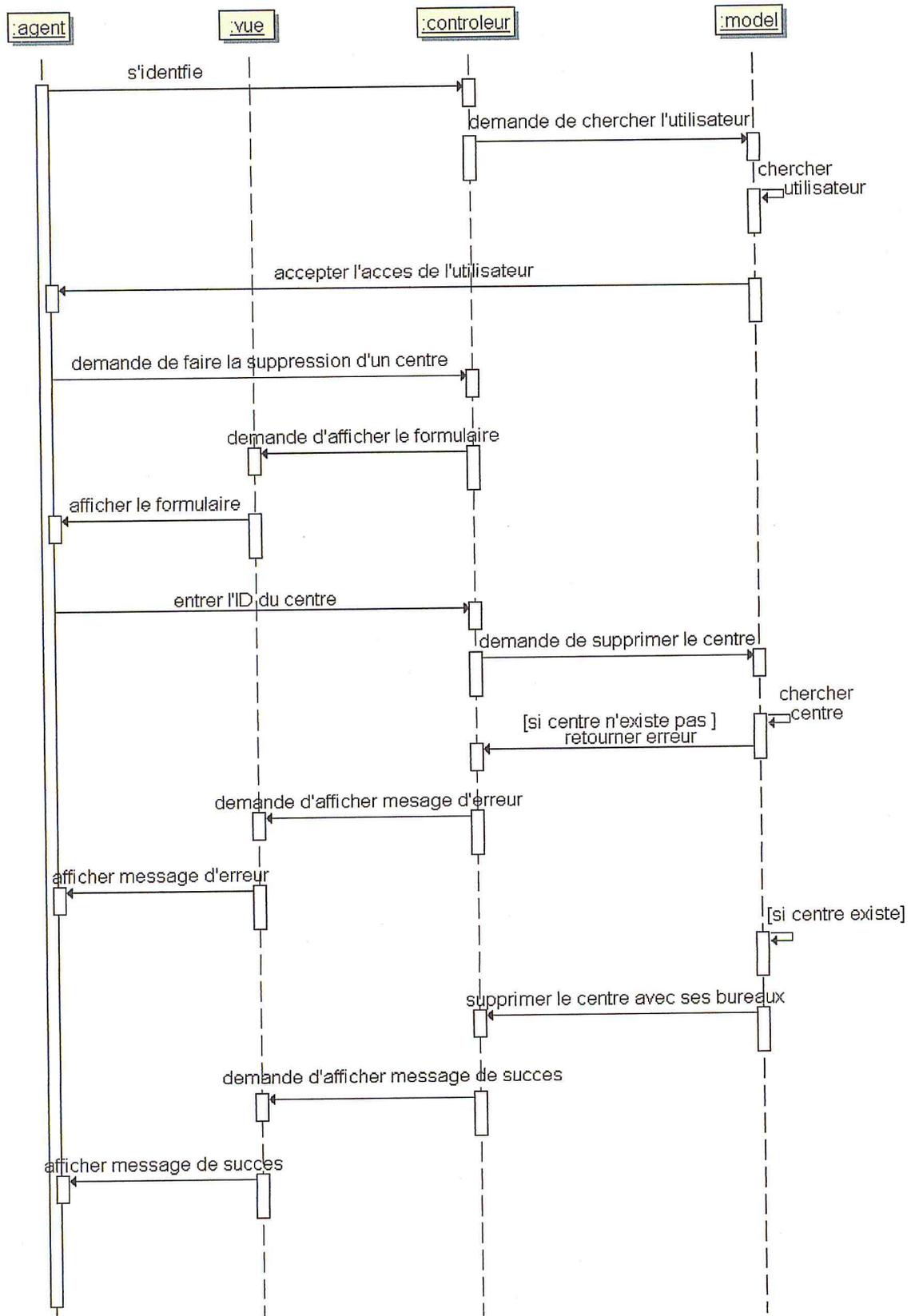


Figure 31:diagramme de séquence pour la suppression d'un centre de vote

3.4.4.2. Diagramme d'activité : « supprimer un centre de vote »

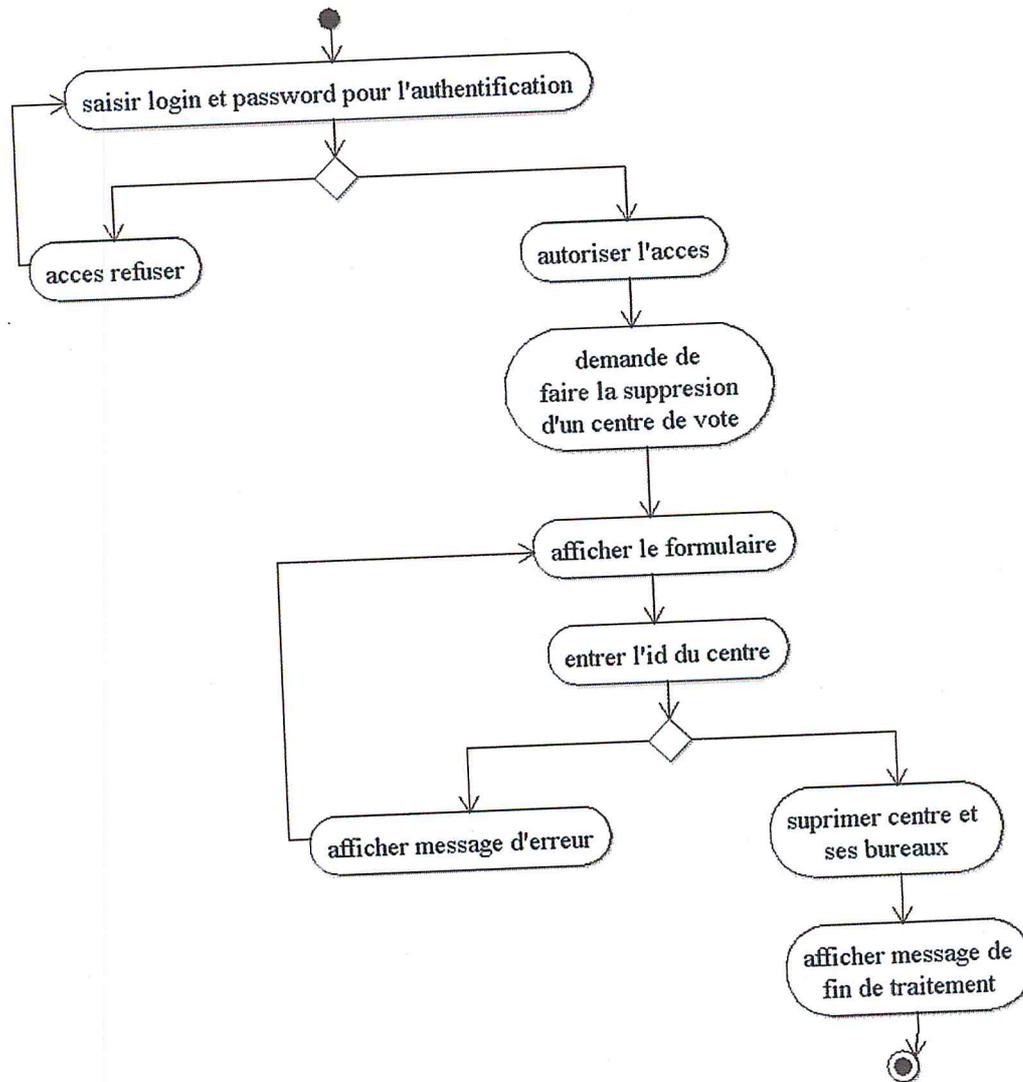


Figure 32:diagramme d'activité pour a suppression d'un centre de vote

3.6. navigation géographique

3.6. 1.Cas d'utilisation : « géo localisation automatique des adresses»

Généralité :

Ca s d'utilisation	Géo localisation automatique des adresses.
Acteur	L'agent de service de vote.
Description	Dans le but d'identifier chaque adresse comme étant un point géographique, avec une paire de latitude et longitude, et après avoir sélectionner les adresses en attente de géo localisation, l'adresse peut se transformée en coordonnées géo graphiques qui définit leur position.
Pré condition	vote saisi l'adresse à géo localisé.
Post condition	Ajout de coordonnées à la base de données.

Scenario :

1. l'agent d'affectation s'identifie pour accède a son espace.
2. Le système accepte l'accès de l'agent d'affectation.
3. L'agent de service de vote demande au contrôleur de géo localiser des adresses qui non pas encore être géo localisé.
4. L'agent de service de vote demande au contrôleur de géo localiser des adresses qui non pas encore être géo localisé.
5. Le contrôleur demande au vue d'afficher le formulaire de géo localisation
6. La vue affiche le formulaire.
7. L'agent sélectionne une adresse et de mande au contrôleur de geolocaliser .
8. Le contrôleur demande a la carte Google Maps de geolocalisaer
9. La carte Google maps interroge le service pour géo localiser l'adresse
10. Si l'adresse est reconnue par le service de GOOGLE MAPS
 - ✓ Le service de géo localisation envoie les coordonnées géographiques a la carte Google maps.
 - ✓ La carte Google maps envoie les coordonnées au contrôleur
 - ✓ Le contrôleur demande au vue d'afficher les coordonnées
 - ✓ La vue affiche les coordonnées
 - ✓ L'agent vérifié si le pointeur affiché sur la carte correspond au lieu demandé.
 - L'agent doit alimenter la base de données avec la nouvelle position géographique de l'adresse.

Sinon

L'agent accède à la géo localisation manuelle.

3.6.1.1 Diagramme de séquence : « géo localisation automatique des adresses »

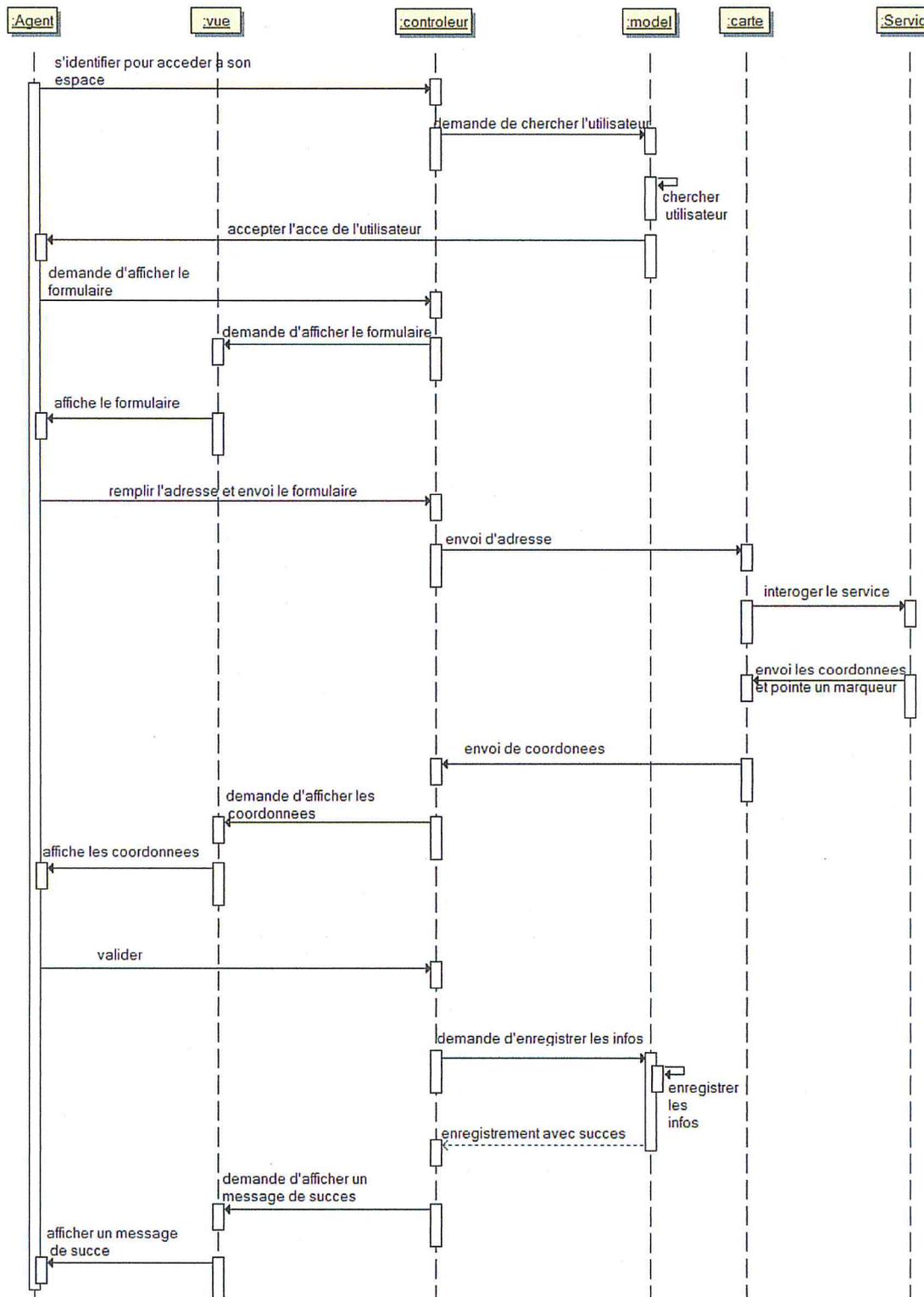


Figure 33:diagramme de séquence pour la géo localisation automatique

3.6.1.2 Diagramme d'activité : « géo localisation automatique »

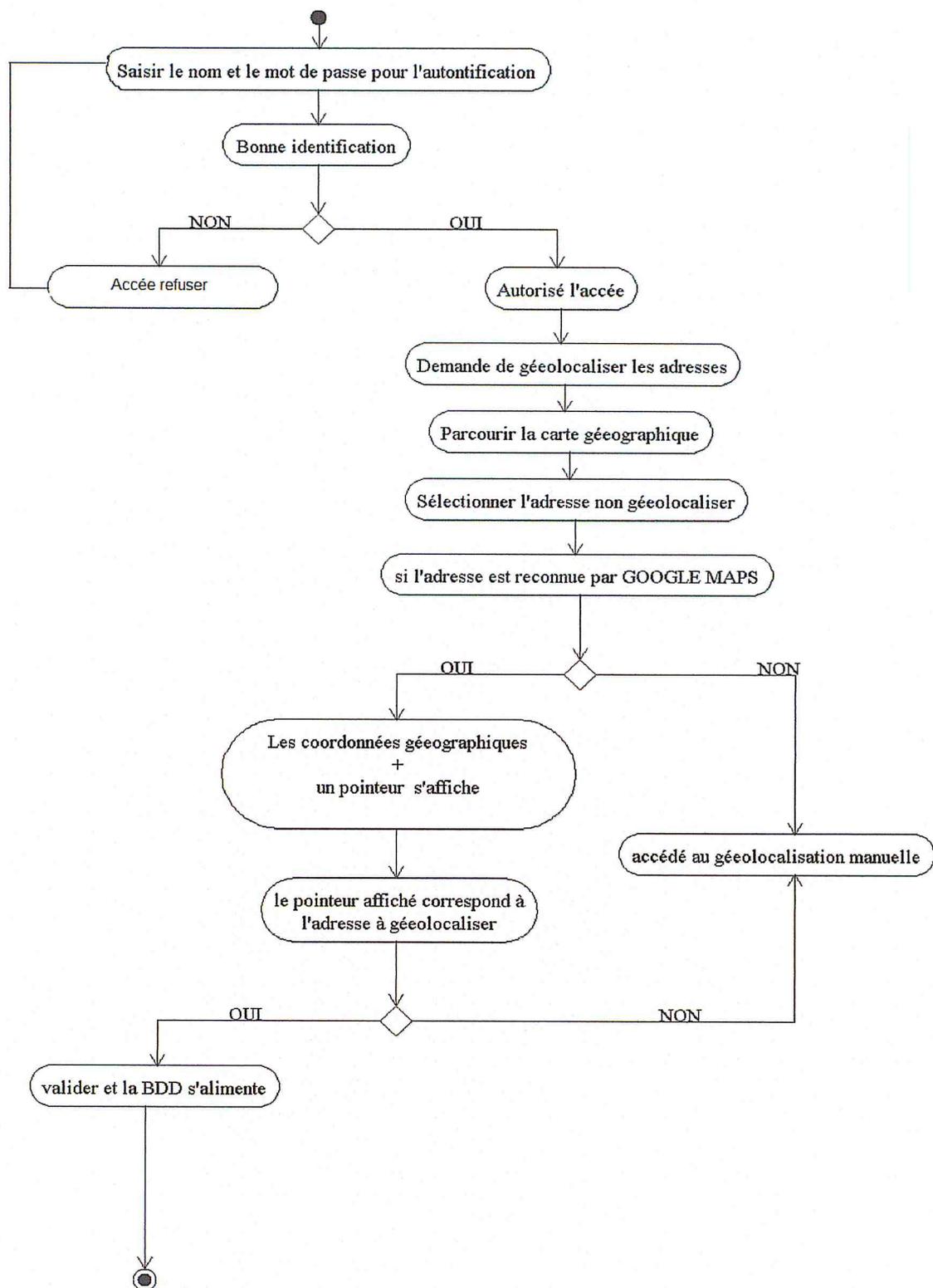


Figure 34:diagramme d'activité pour la géo localisation automatique

3.6.2 Cas d'utilisation : « géo localisation manuelle des adresses »

Généralité :

Cas d'utilisation	Géo localisation manuelle d'adresses.
Acteur	L'agent de service de vote.
description	Dans le but d'identifier une adresse qui n'été pas reconnue par GOOGLE MAPS et pour géo localiser les divers adresses de la base de données; une géo localisation est effectuée.
Pré condition	L'agent de service de vote parcourt la carte géographique; et point sur l'adresse concernée.
Post condition	Ajout de coordonnées à la base de données.

Scenario :

1. L'agent de service de vote s'identifie pour accède a son espace.
2. Le système accepte l'accédé de l'agent de service de vote.
3. L'agent de service de vote demande au contrôleur de géo localiser des adresses qui non pas encore être géo localisé.
4. Le contrôleur demande au vue d'afficher le formulaire
5. La vue affiche le formulaire
6. L'agent parcourir la carte Google maps.
7. L'agent pointe le marqueur sur le point d'adresse concernée par la géo localisation.
8. La carte Google maps interroge le service
9. Le service envoie les coordonnées a la carte Google maps
10. Carte Google maps envoie les coordonnées au contrôleur
11. Le contrôleur demande au vue d'afficher les coordonnées
12. La vue affiche les coordonnées
13. L'agent saisie l'adresse et alimenter la base de données

3.6..2.1 Diagramme de séquence : « géo localisation manuelle des adresses »

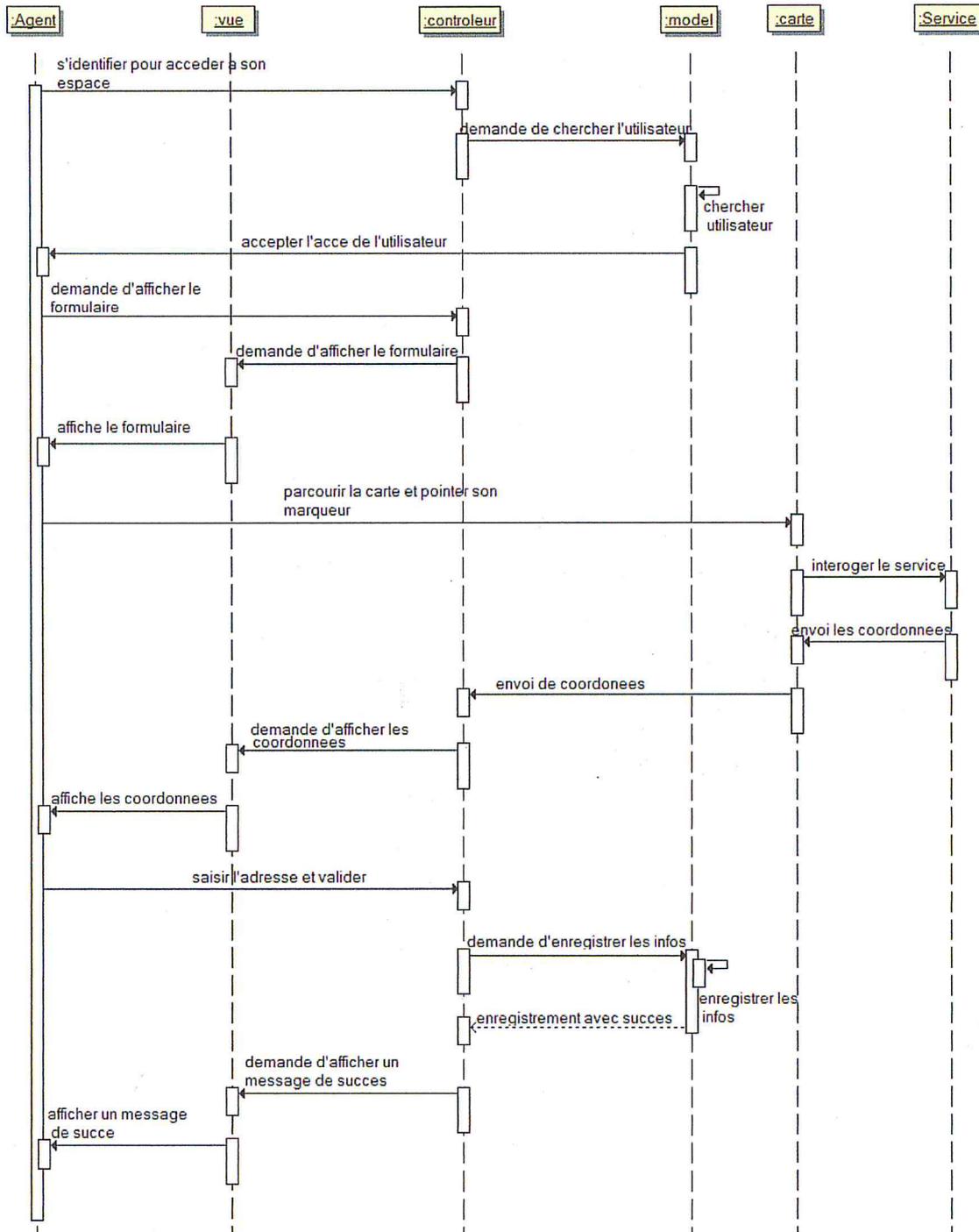


Figure 35:diagramme de séquence pour la géo localisation manuelle

3.6.2.2 Diagramme d'activité : « géo localisation manuelle des adresses »

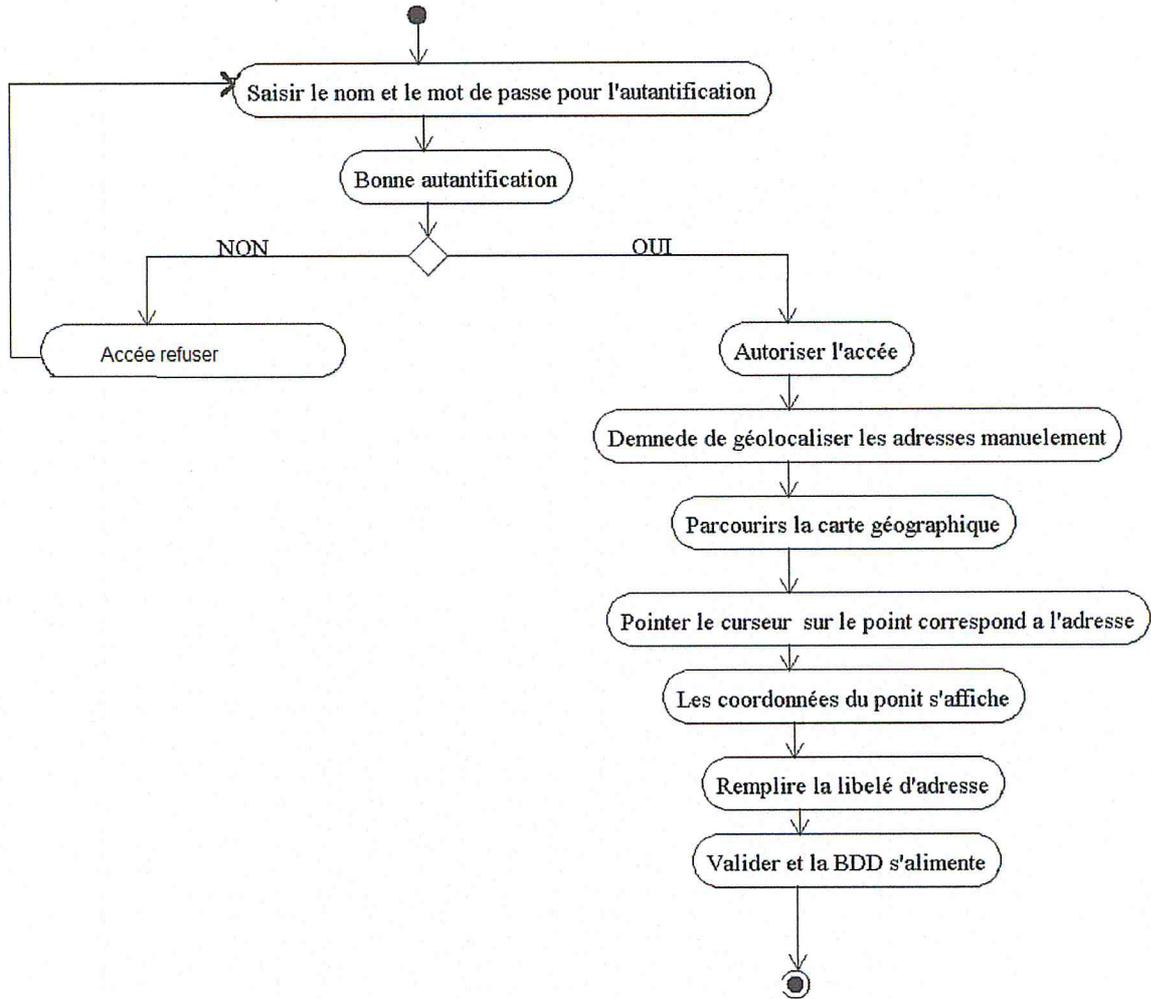


Figure 36:diagramme d'activité pour la géo localisation manuelle

3.7. Cas d'utilisation : « affectation automatique »

Généralité :

Cas d'utilisation	Affectation automatique
Acteur	Agent d'affectation
description	Dans le but d'affecter les votants au centre de vote le plus proche en utilisant Google Maps
Pré condition	L'agent de service de vote est identifié.
Post condition	L'opération de le l'affectation est effectuer.

Scenario :

1. L'agent d'affectation s'identifie pour accéder à son espace.
2. Le système accepte l'accès de l'agent d'affectation.
3. L'agent d'affectation demande au contrôleur d'afficher le formulaire d'affectation automatique
4. Le contrôleur demande au vue d'afficher le formulaire
5. La vue affiche le formulaire.
6. L'agent saisi l'adresse du départ
7. Le contrôleur demande au model de récupérer les coordonnées
8. Le model recherche les coordonnées et les envoie au contrôleur
9. Le contrôleur demande au vue d'afficher les coordonnées
10. La vue affiche les coordonnées
11. L'agent saisi l'adresse d'arrivée
12. Le contrôleur demande au model de récupérer les coordonnées
13. Le model recherche les coordonnées et les envoie au contrôleur
14. Le contrôleur demande au vue d'afficher les coordonnées
15. La vue affiche les coordonnées
16. L'agent d'affectation valide
17. Le contrôleur les adresses du trajet a la carte Google maps
18. Carte Google maps interroge le service
19. Le service envoie les distances et les durées et les traces des trajets

20. La carte envoie les informations récupérées au contrôleur
21. Le contrôleur demande au vue d'afficher les informations
22. La vue affiche les informations
23. L'agent d'affectation sélectionne le centre choisi et affecter les votants
24. Le contrôleur envoie les coordonnées de trajet choisi à la carte Google maps
25. La carte Google maps interroge le service
26. Le service envoie les détails du trajet sur un panel
27. La carte envoie au contrôleur les détails
28. Le contrôleur demande au vue d'afficher les détails
29. La vue affiche les détails.

3.7.1. Diagramme de séquence : « affectation automatique »

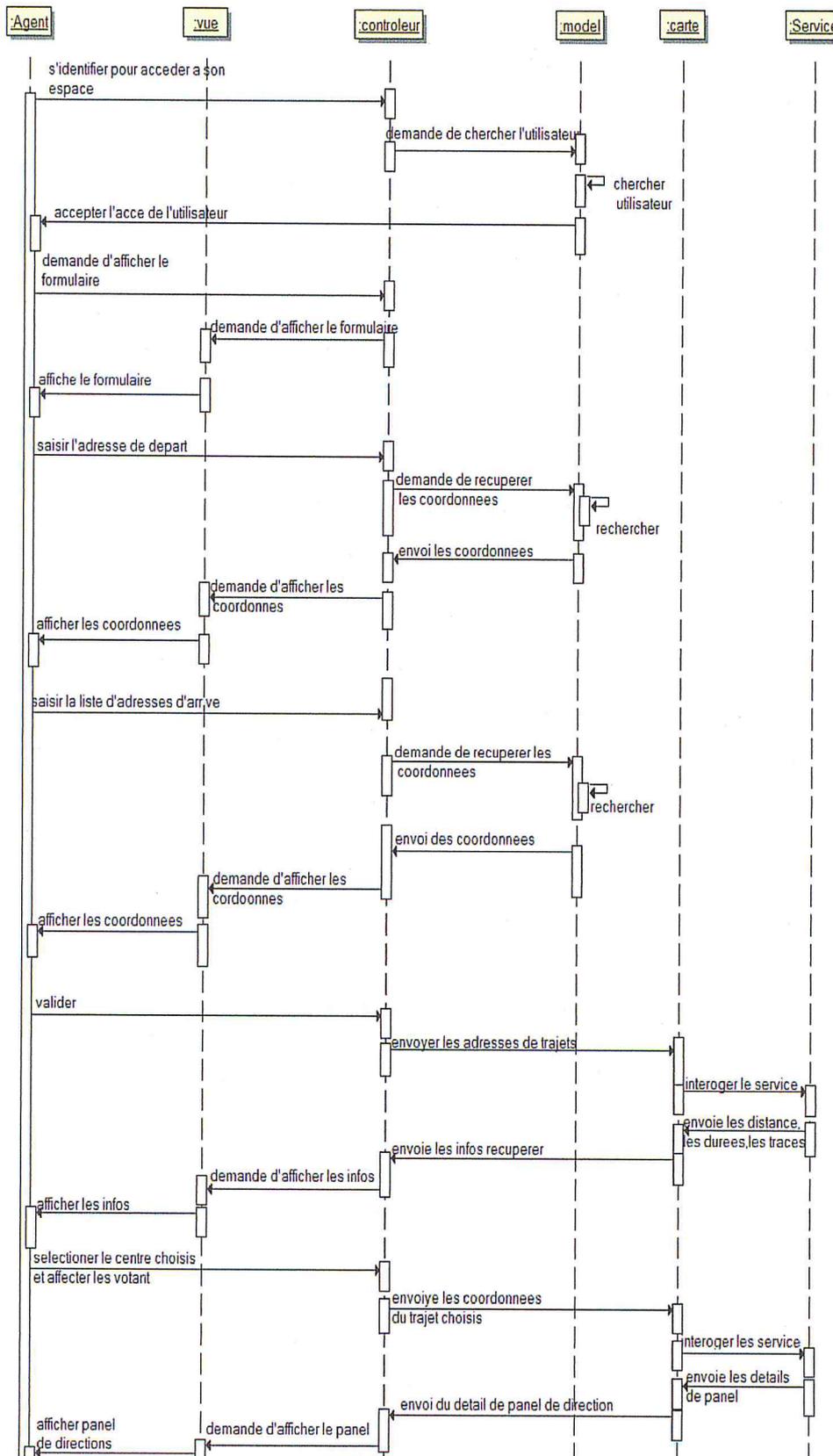


Figure 38:diagramme de séquence d'affectation automatique

3.7.2. Diagramme d'activité : « affectation automatique »

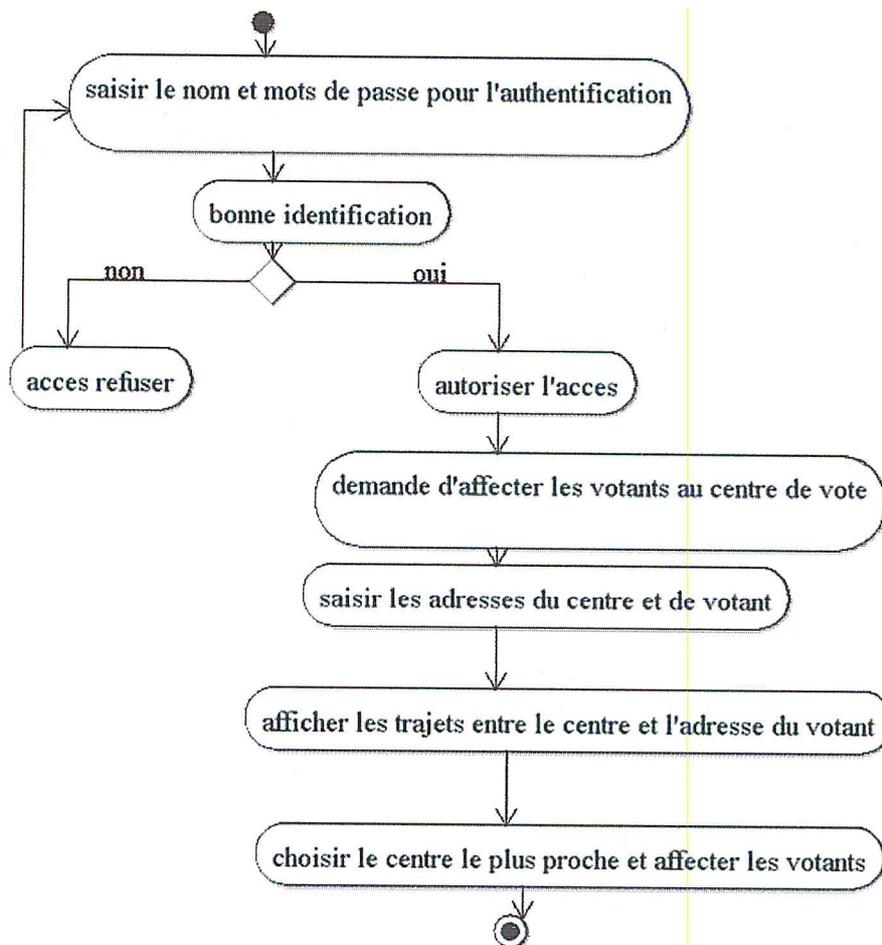


Figure39:diagramme d'activité pour l'affectation automatique

3.8. Gestion des staffs

3.8.1 Cas d'utilisation : « ajouter les éléments du staff »

Généralité :

Cas d'utilisation	Ajout des éléments de staff
Acteur	Agent de service de vote
déscription	Lorsque la date du nouveau processus de vote est declare, l'inscription des éléments de staff débutée l'agent peut ajouter les gents cencé de superviser l'opération du vote: les administrateurs et les observateurs
Pré condition	L'agent de service de vote est identifié.
Post condition	L'opération de l'ajout des éléments de staff est effectuer.

Scenario :

- 1- L'agent de service de vote s'identifie pour accéder a son espace.
- 2- Le système accepte l'accès de l'agent de service de vote.
- 3- L'agent de service de vote demande au système d'ajouter les éléments de staff inscrit dans la nouvelle opération de vote.
- 4- Le contrôleur demande au vue d'afficher le formulaire d'ajout d'un nouvel élément de staff.
- 5- La vue affiche le formulaire.
- 6- L'agent choisir le type de staff (administrateur, observateur).
- 7- L'agent remplit le formulaire qui contient des informations concernant le nouveau élément de staff concerné par le nouveau processus de vote tel que (nom d'élément, prénom d'élément, nom du centre, adresse du centre ...) et (le nom de l'organisation, origine...) pour les observateurs et envoie le formulaire l'ajout d'un nouveau élément.
- 8- La vue présente un message indiquant que l'opération est affectée avec sucée.

3.8.1.1 Diagramme de séquence : « ajout des staffs »

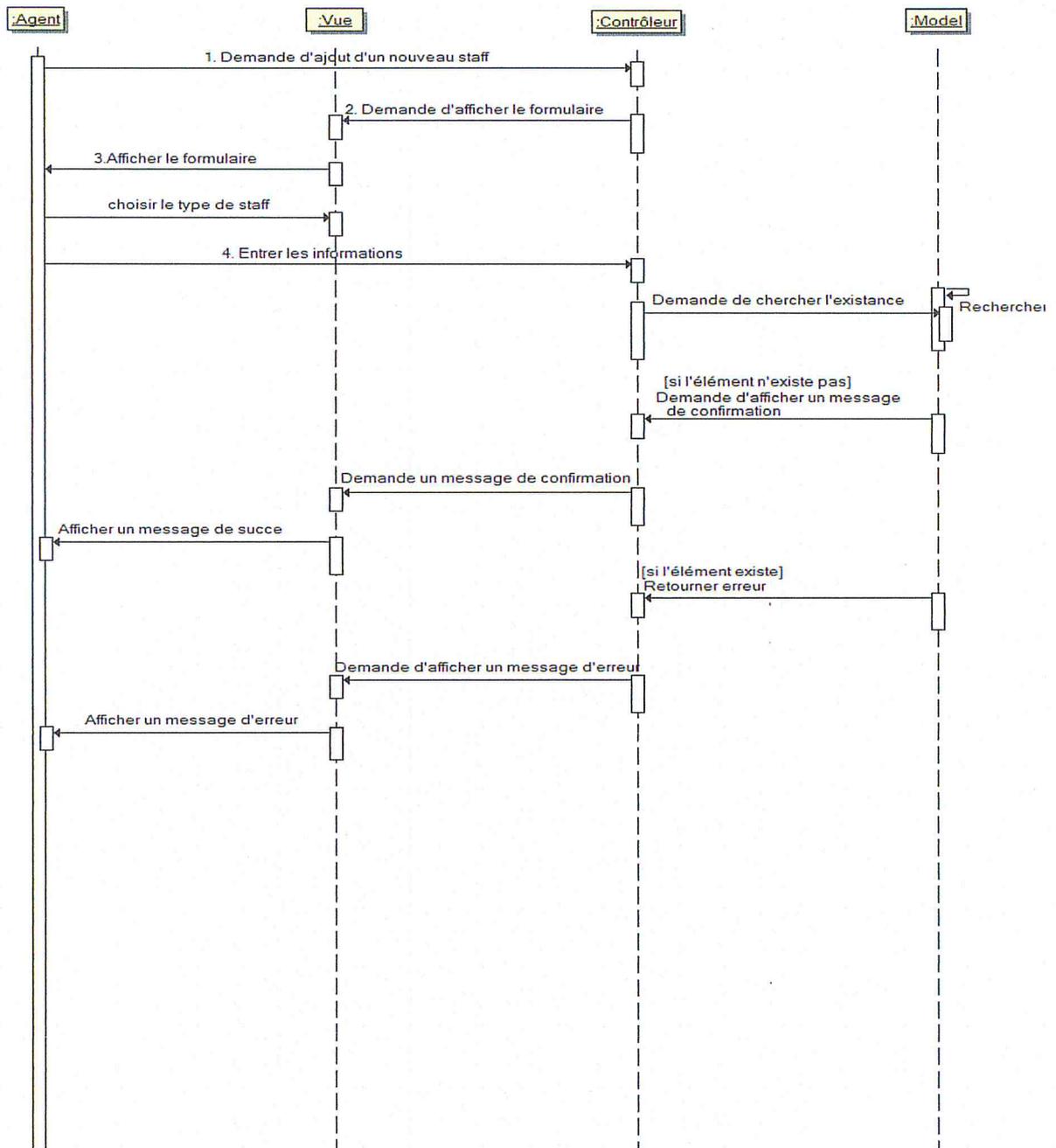


Figure 40:diagramme de séquence pour l'ajout des staffs

3.8.1.2 Diagramme d'activité : «ajout des staffs»

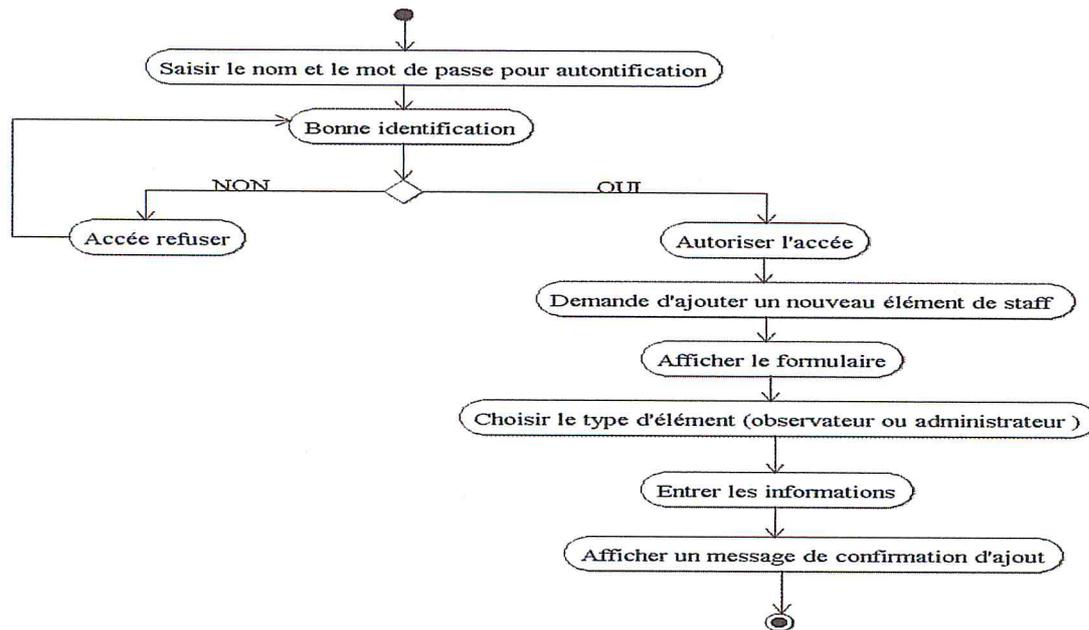


Figure 41:diagramme d'activiste pour l'ajout des staffs

3.8.2. Cas d'utilisation : « modifier staff »

Généralité :

Cas d'utilisation	Modifier les informations concernant un les staff
Acteur	L'agent de service de vote
Description	Après avoir ajouté le nouveau élément ; et a cause de changement de centre à qui est affecter, les informations peuvent être modifié (la modification est limité sur des champs bien précises.)
Pré condition	L'agent de service de vote est identifié
Post condition	Les informations après modification sont enregistrés.

Scenario :

1. l'agent de service de vote s'identifie pour accède a son espace.
2. Le système accepte l'accès de l'agent de service de vote.
3. L'agent de service de vote demande au système de modifier des informations bien précises d'un élément de staff.
4. Le contrôleur demande d'afficher le formulaire adéquat.
5. La vue affiche le formulaire.
6. L'agent choisir le type de staff (administrateur, observateur).
7. L'agent remplit le formulaire qui contient des informations concernant l'élément staff (identifiant, prénom...) après modification envoie le formulaire.

- 8. S'il trouve que élément existe.
Il fait la modification demande.
La vue affiche un tableau contient les détails retourné par le model et qui sont enregistrés dans la BDD.

Si non

- Il retourne l'erreur au contrôleur.
- Le contrôleur demande au vue d'afficher un message d'erreur.
- La vue présente un message d'erreur indiquant que l'élément n'existe pas.

3.8.2.1 Diagramme de séquence : « modifier les staff »

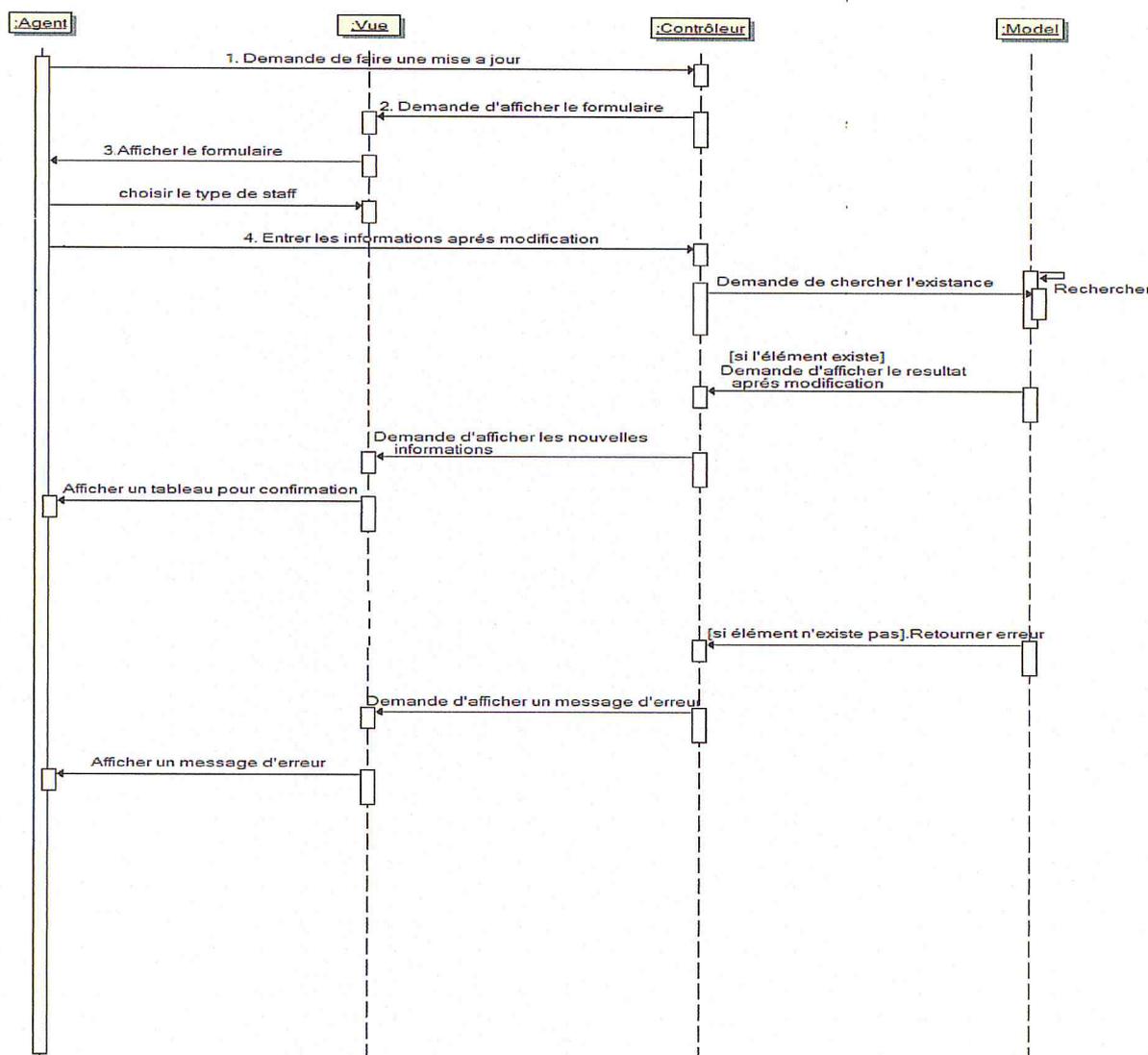


Figure 42:diagramme de séquence pour modifier staff

3.8.2.2 Diagramme d'activité : « modifier staff »

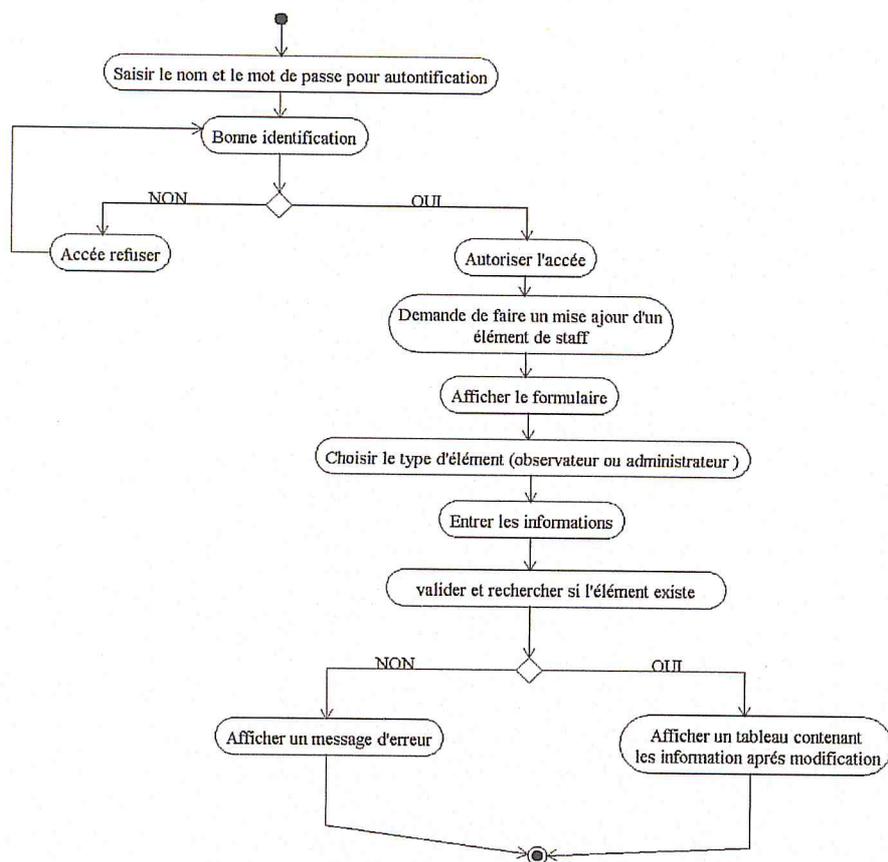


Figure 43:diagramme d'activité pour modifier staff

3.8.3 Cas d'utilisation : « supprimer staff »

Généralité :

Cas d'utilisation	Supprimer un élément de staff.
Acteur	L'agent de service de vote.
Description	L'opération est effectuée dans des cas qui sont vraiment limité; et en premier lieu après la déclaration de suspension ou un arrêt de travail.
Pré condition	L'agent de service de vote est identifié.
Post condition	L'élément de staff est supprimer.

Scenario :

1. l'agent de service de vote s'identifie pour accède a son espace.
2. Le système accepte l'accède de l'agent de service de vote.
3. L'agent de service de vote demande au système de supprimer un élément de staff.
4. Le contrôleur demande d'afficher le formulaire adéquat.

5. La vue affiche le formulaire.
6. L'agent remplit le formulaire qui contient l'information concernant l'élément à supprimer(nom) et envoi le formulaire.
7. Le contrôleur demande au model de chercher tous les éléments qui ont le même nom qui est fournit par l'agent en vue de voir tous les résultats.
8. Si le model trouve l'élément demandé
9. L'agent Sélectionne l'élément concerné.
Supprimer.

Sinon

10. Il retourne l'erreur au contrôleur.
11. Le contrôleur demande au vue d'afficher un message d'erreur.
12. La vue afficher un message d'erreur indiquant que l'élément n'existe pas

3.8.3.1 Diagramme de séquence : « supprimer staff »

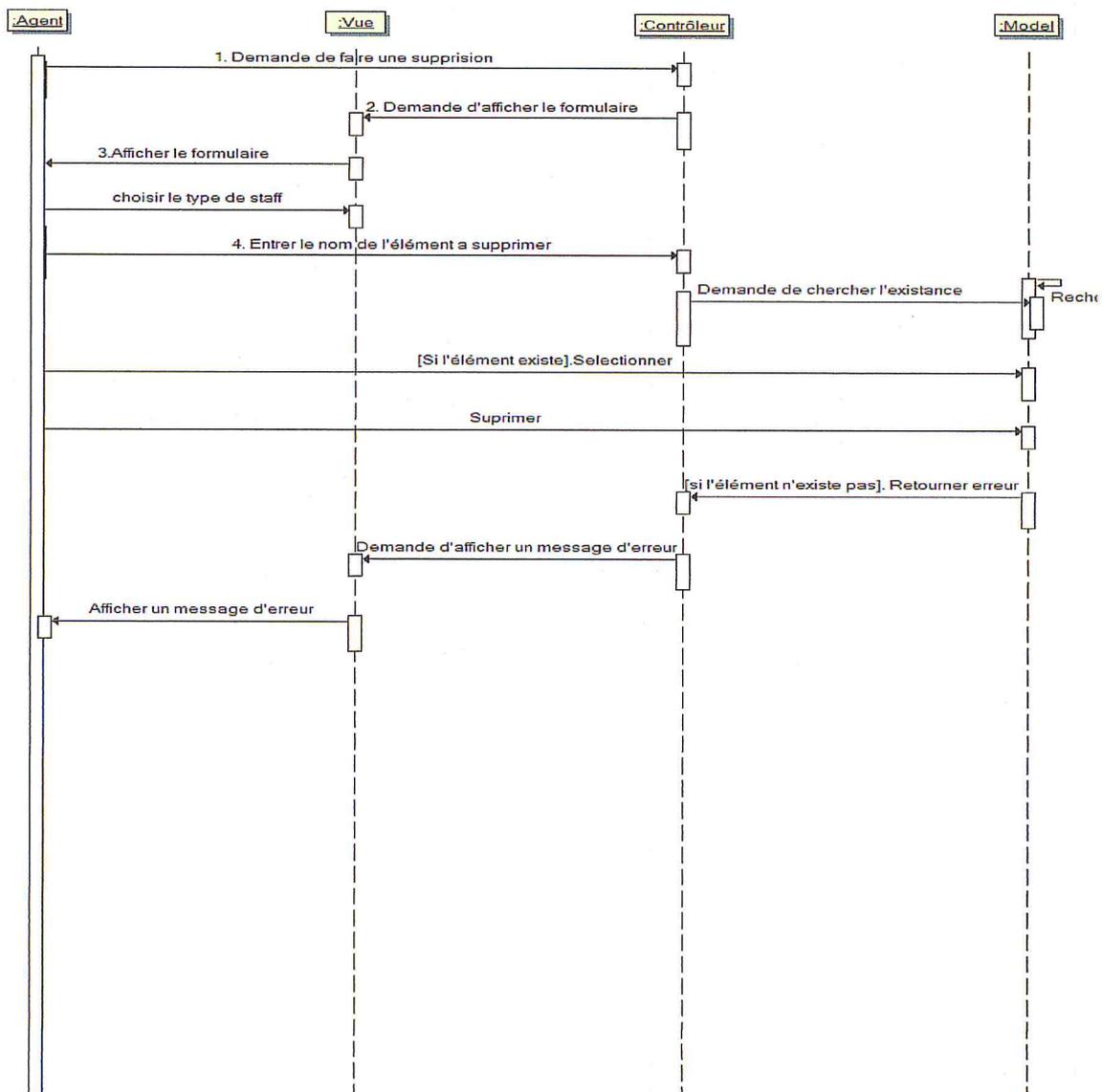


Figure 44:diagramme de séquence pour la suppression d'un staff

3.8.3.2. Diagramme d'activité : « supprimer staff »

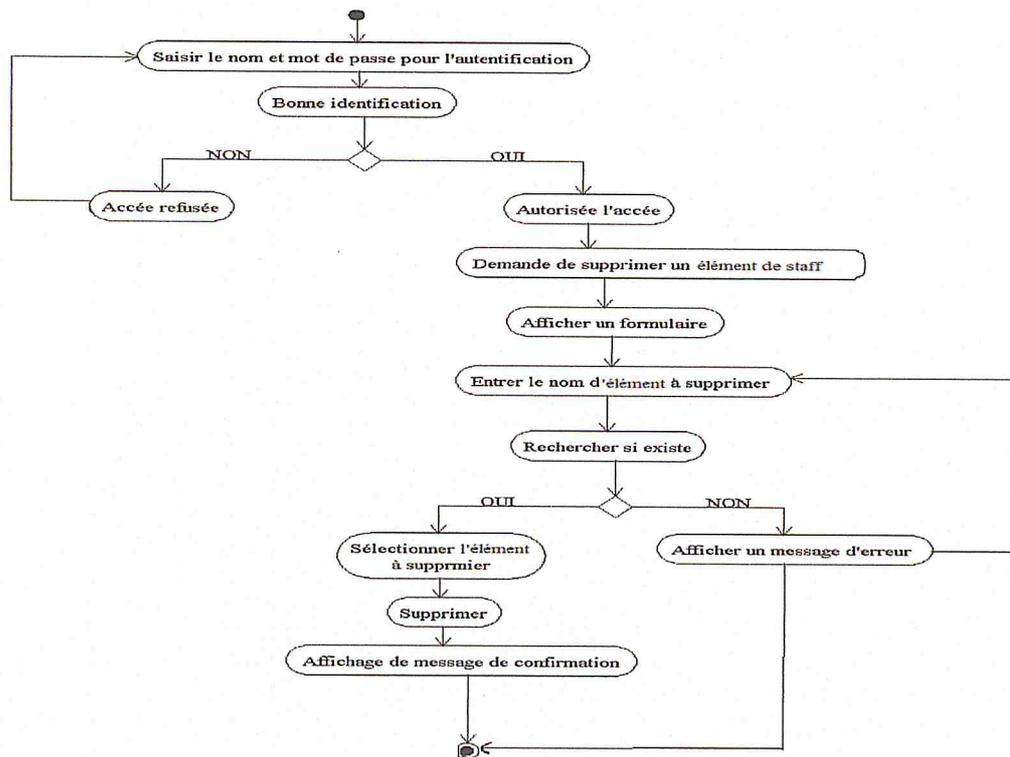


Figure 45: diagramme d'activité pour la suppression d'un staff

3.9.1 Cas d'utilisation : « réajustement d'affectation »

Généralité :

Cas d'utilisation	Réajustement d'affectation
Acteur	L'agent d'affectation.
Description	L'opération est effectuée dans des cas où le votant veut changer le centre de vote
Pré condition	L'agent d'affectation est identifié.
Post condition	Le votant et réaffecter

Scenario :

1. l'agent d'affectation s'identifie pour accéder à son espace.
2. Le système accepte l'accès de l'agent d'affectation.
3. L'agent d'affectation demande au système de réaffecter le votant

4. Le contrôleur demande d'afficher le formulaire adéquat.
5. La vue affiche le formulaire.
6. L'agent remplit le formulaire qui contient des informations concernant le votant (identifiant, prénom...) après réaffectation envoie le formulaire.
7. La vue affiche un message indiquant que l'opération est affectée avec succès.

3.9.1.1 Diagramme de séquence : « réajustement d'affectation »

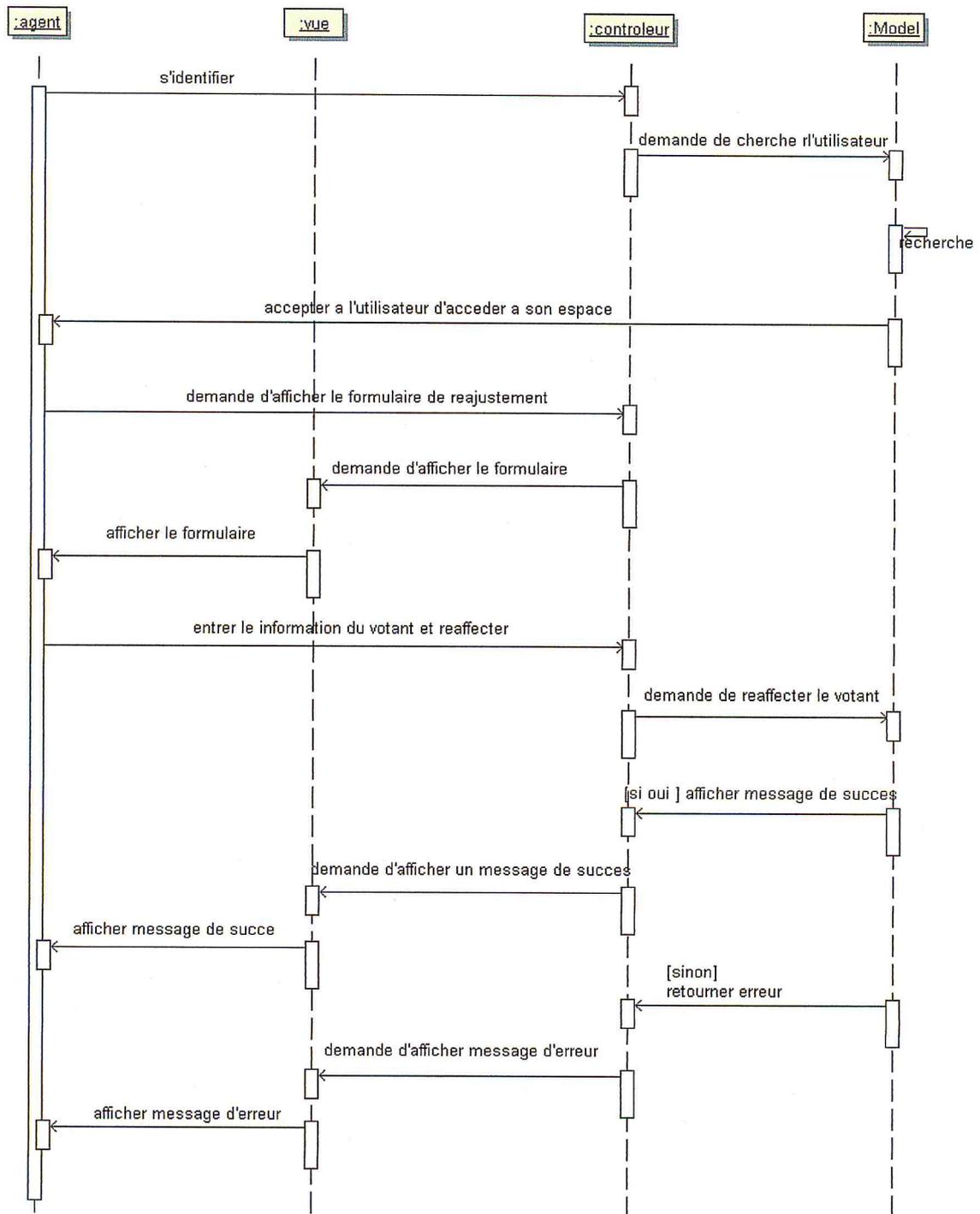


Figure 46:diagramme de séquence pour réajustement d'affectation

3.9.1.2. Diagramme d'activité : « réajustement d'affectation »

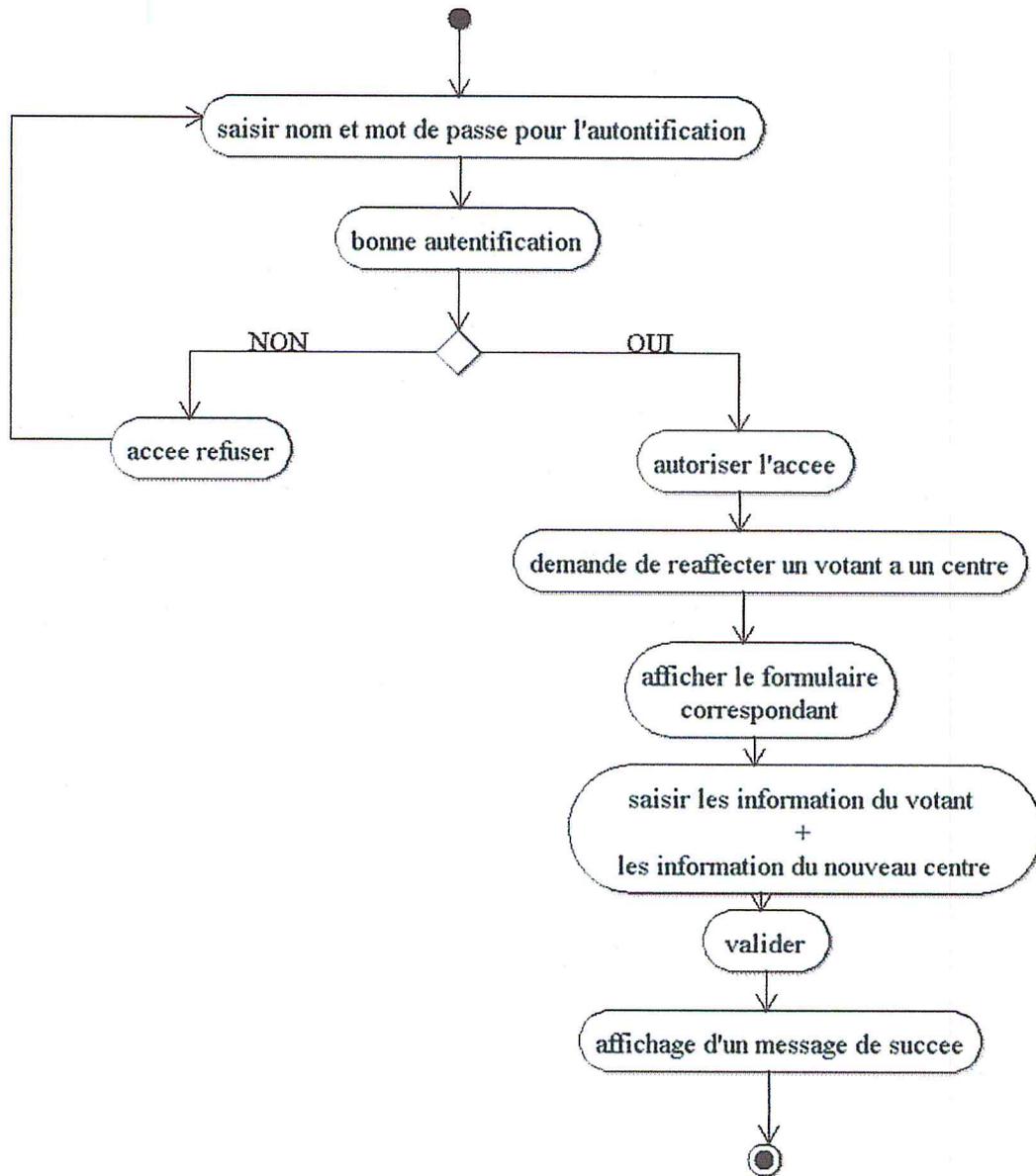


Figure 47:diagramme d'activité pour le réajustement d'affectation

4 .diagramme de classe :

Jusqu'a ici nous avons défini les différentes tâches constituant notre système. L'ensemble des cas d'utilisation avec l'analyse détaillée permet d'avoir les données Manipulables par le système. Ces données permettent de définir des classes et les relations Entre ces classes.

4.1. Dictionnaire de données :

Classe	Attribut	Code : type	Valeur	Les méthodes
Personne	identifiant de personne Nom de personne Prenom de personne Numero extrait naissanc Commune de naissance Date de naissance N_carte_national Identifiant del'adresse	#Id_personne :varchar Nom_personneF : varchar Prenom_personneF : varchar numero_extrait_naissance commune_naissance date_naissance:varchar N_carte_national :varchar #id_adresse :varchar	Not null Not null Not null Not null Not null Not null Not null	/
Votant	Identifiant de votant Nom de votant Prenom de personne Date de naissance Wilaya de naissance Daira de naissance Commune de naissance Sexe Lieu de vote Num de bureau de vote affecter identifiant de personne	#id_votant :varchar Nom_votant:varchar Prenom_votant:varchar date_naissance:varchar wilaya_naissance:varchar daira_naissance:varchar commune_naissance sexe:varchar Lieu_vote :varchar n_bur_vote:varchar Affecter :booleen Id_personne :varchar	Not null Not null	Ajouter votant() Modifier votant() Supprimer votant() Rechercher votant() Consulter()
Centre vote	Identifiant centre vote Nom du centre Adresse du centre Identifiant de commune	#Id_CentreEvote :varchar NameCentreEvote:varchar id_adresse:varchar Id_commune	Not null Not null Not null Not null	Ajouter centre() Modifier centre() Supprimer centre() Rechercher centre() Consulter()

Bureau vote	Identifiant de bureau Identifiant centre vote	#Id_BureauVote:varchar Id_CentreEvote:varchar	Not null Not null	Ajouter bureau() Modifier bureau() Supprimer bureau() Rechercher bureau() Consulter()
Adresse	Identifiant adresse Latitude Longitude Identifiant commune	#id_adresse :varchar Latitude :décimal Longitude :décimal	Not null Not null Not null Not null	Geolocaliser()
Administrateur	Identifiant votant Identifiant role	#id_votant :varchar Id_role:varchar	Not null Not null	Ajouter Admin () Modifier Admin () Supprimer Admin () Rechercher Admin () Consulter()
Observateur	Identifiant observateur Nom Prenom Num pièce d'identité Nom d'organisation Identifiant centre vote	#id_observateur:varchar nom prenom n_piece-ident nom_organis #Id_CentreEvote :varchar	Not null Not null Not null Not null Not null	Ajouter obser() Modifier obser () Supprimer obser () Rechercher obser () Consulter()
Wilaya	Identifiant wilaya Nom wilaya	#id_wilaya :varchar wilaya_F :varchar	Not null Not null	/
Daira	Identifiant daïra Nom daïra Identifiant wilaya	#id_daira :varchar Nom_daira_F :varchar id_wilaya :varchar	Not null Not null Not null	/
Commune	Identifiant commune Nom commune Identifiant daïra	#id_commune :varchar Nom_commune_F :varchar id_daira :varchar	Not null Not null Not null	/
Utilisateur	Identifiant utilisateur Nom de l'utilisateur Mode de pase E-Mail Identifiant profil	#id_utilisateur:varchar Nom_Utilisateur :varchar Mot_de_pass:varchar E_mail:varchar id_profil:varchar	Not null Not null Not null Not null Not null	Ajouter utili() Modifier utili () Supprimer utili () Rechercher utili () Consulter()
Profil	Identifiant profil Nom du profil	id_profil:varchar Nom_profil:varchar	Not null Not null	/

4.2 Diagramme de classe :

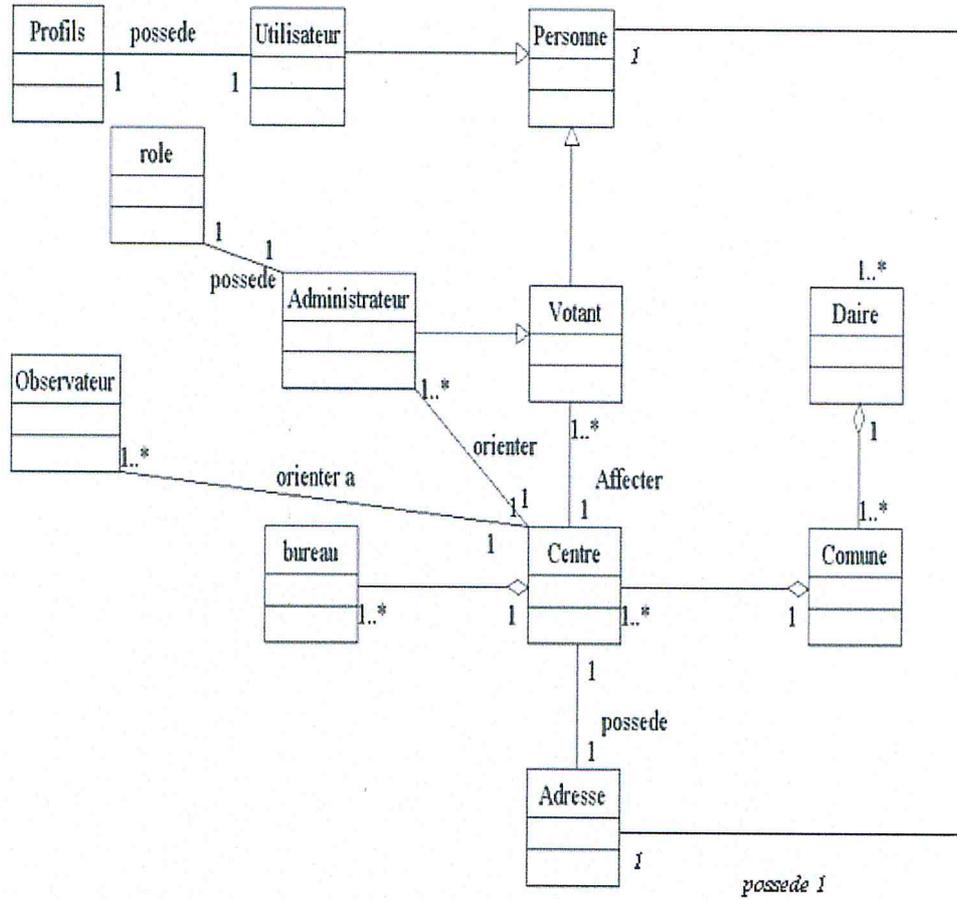


Figure 44:diagramme de classe générale

4.3 Codification :

a) Wilaya

|_ Numéro de wilaya _|

|_ 2 Cases _|

b) Daïra

|_ Numéro de wilaya _|_ Numéro de daïra _|

|_ 2 Cases _|_ 3Cases _|

c) Commune

|_ Numéro de wilaya _|_ Numéro de daïra _|_ Numéro de commune _|

|_ 2 Cases _|_ 3 Cases _|_ 4 Cases _|

d) centre de vote :

|_ code commune _|_ 2 Case _|

e) bureau de vote :

|_ code centre _|_ 2 Case _|

f) votant :

|_ année de naissance _|_ code commune _|_ date inscription _|_ nu séquentiel _|

Conclusion :

Dans ce chapitre nous avons faits une description détaillée de chaque cas d'utilisation à l'aide des diagrammes de séquence et des diagrammes d'activité selon le langage UML et enfin nous avons arrivé à construire notre diagramme de classe .

Cette conception permet la réalisation de notre système qui sera montré dans le chapitre suivant.

V

Implémentation

Introduction

Presentation de l'application

Conclusion

1. Introduction :

Après l'étape de conception de la solution répondant aux objectifs fixés préalablement, nous entamons l'étape de réalisation de notre application dédié à l'affectation des votants aux centres de vote.

Cette étape met en évidence plusieurs choix techniques (choix du langage et du SGBD...) que nous justifierons dans ce présent chapitre.

Nous donnerons aussi une illustration de l'utilisation de notre application, pour montrer les différentes fonctionnalités qu'elle propose.

2. Environnement de développement:

2.1 Les technologies utilisées:

2.1.1 Les APIs Web:

Une API (*Application Programming Interface* ou *API*) est une interface de code source fournie par un système informatique, en vue de répondre à des requêtes pour des services qu'un autre programme informatique pourrait lui faire. [Digimind07].

2.1.1.1 L'utilisation des APIs Web:

L'utilisation des APIs Web nécessitent :

1. Une clé d'utilisation fournie par le producteur de service
2. Avoir la documentation de l'API (fournie par le producteur de l'API). La documentation est indispensable pour l'utilisation.
3. L'étape la plus importante c'est de maîtriser le langage supporté par l'API (en général c'est le langage JavaScript).

Remarque :

1. L'API Web utilisées dans notre application est l'API de Google Map v3.
2. la plupart des APIs sont offertes gratuitement. Même les producteurs de services qui offrent des APIs payantes, offrent des APIs compatibles gratuites, mais avec moins de fonctionnalités que Les APIs payantes.

2.1.2 AJAX:

AJAX est une technologie utilisée pour construire des pages web dynamiques côté client. Les données sont échangées avec le serveur par des requêtes Javascript, et le serveur effectue des traitements sur ces données. Le moteur AJAX permet de supprimer l'attente pour interagir avec le serveur. [Crane06].

L'utilisation des techniques AJAX, permettent d'envoyer des requêtes au serveur

HTTP pour récupérer uniquement les données nécessaires en utilisant la requête XMLHttpRequest ; ces requêtes sont dites « asynchrones ».

2.1.2.1 Avantages:

L'avantage de cette technologie est d'abord la vitesse à laquelle une application AJAX répond aux actions de l'utilisateur. Les applications sont alors plus réactives, la quantité de données échangées entre le navigateur et le serveur HTTP étant fortement réduite. Le temps de traitement de la requête côté serveur est également réduit.

2.2 Outils de développement :

Dans cette partie nous allons présenter les outils et technologies de développement que nous avons utilisées :

2.2.1 Le langage JAVA:

Est un langage de programmation orienté objet développé par Sun Microsystems . Pour programmer en java nous avons besoin d'un environnement de développement adapté. Le JRE de Sun répond à cette exigence. Le JRE ou Java Runtime Environment est un environnement comprenant une machine virtuelle java grâce à laquelle des programmes java non compilés peuvent s'exécuter . Pour l'implémentation ,c'est la version **jre1.5.0_16** qui à été utilisé .

2.2.2 Le langage JAVASCRIPT:

Javascript est un langage de script orienté objet principalement utilisé dans les pages HTML, ce langage permet une interaction avec l'utilisateur en fonction de ses actions (lors du passage de la souris au dessus d'un élément, du redimensionnement de la page...). La version standardisée de Javascript est le ECMAScript

2.2.3 l'environnement eclipse:

Eclipse est un environnement de développement intégré (Integrated Development Environment) dont le but est de fournir une plate-forme modulaire pour permettre de réaliser des développements informatiques.

2.2.4 Choix du SGBD :

Pour l'implémentation de la couche stockage des données, le système de gestion de base de données sera MYSQL, et cela pour faciliter l'extraction des données.

2.2.4.1 Présentation de MYSQL:

MySQL est un serveur de base de données relationnelles Open Source.

Un serveur de bases de données stocke les données dans des tables séparées plutôt que de tout rassembler dans une seule table. Cela améliore la rapidité et la souplesse de l'ensemble. Les tables sont reliées par des relations définies, qui rendent possible la combinaison de données entre plusieurs tables durant une requête. Le SQL dans "MySQL" signifie "Structured Query Language": le langage standard pour les traitements de bases de données.

2.2.4.2 Fonctionnalités de MYSQL:

Ce choix est justifié par les avantages suivants que MYSQL offre :

- ✓ Sa facilité d'utilisation avec le langage de développement java, qui offre un accès direct et donc rapide au SGBD.
- ✓ Sa puissance pour gérer les très grandes bases de données.
- ✓ Ses puissants de sécurité et de gestion des droits d'accès.
- ✓ Intégration du langage SQL.
- ✓ Proposer aux développeurs un environnement riche, souple et intégré permettant de créer efficacement des applications adaptées aux besoins de chaque organisation.
- ✓ fournir des solutions décisionnelles robustes et intégrées favorisant la prise de décisions et permettant d'accroître l'efficacité de l'ensemble de l'organisation.

2.3 L'architecture utilisée:

Dans cette architecture les applications au niveau serveur sont délocalisées, c'est à dire que chaque serveur est spécialisé dans une tâche (serveur web ou serveur de base de données par exemple), en interrogeant u service externe qui est Google maps c'est à dire que l'architecture est partagée entre :

- ❖ **Le client:** le demandeur de ressource.
- ❖ **Le serveur de données** MYSQL.
- ❖ **Le serveur web.** Apache Tomcat.

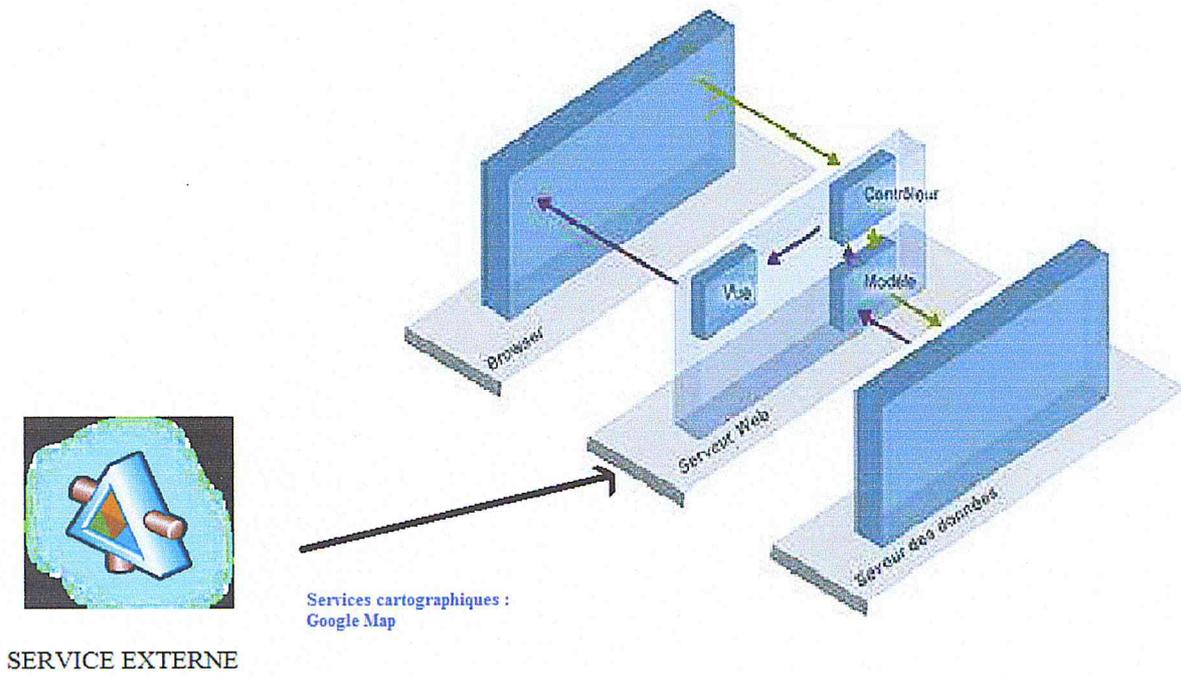


Figure45:architecture de l'application

3.Présentation de l'application:

On va prendre quelque exemple concernant les différentes gestions :

3.1 Gestion des votants :

La liste des Votants

	Nom votant	Prenom votant	date naissance	wilaya_naissance	daira_naissance	commune_naissance	sexe
<input type="checkbox"/>	benmeddour	nabil	06-09-1989	blida	blida	blida	masculin
<input type="checkbox"/>	benmeddour	samah	16-09-1987	blida	blida	blida	feminin
<input type="checkbox"/>	benmeddour	mohamed	16-12-197	blida	blida	blida	masculin
<input type="checkbox"/>	benmeddour	nawal	16-12-1977	blida	blida	blida	feminin
<input type="checkbox"/>	benmeddour	zineb	12-11-1988	blida	blida	blida	feminin

Soumettre la requête

Figure 47:suppression des votants

3.2 Gestion des centres et bureaux de vote :

❖ Ajouter un centre de vote :

Ajouter un centre de vote

Wilaya du commune blida ▾

Daira du cevtre ouled yaich ▾

commune du centre ouled yaich ▾

Nom du centre centre mahieddine derb:

adresse du centre cite 1er mai

Envoyer

retour

Figure 48:ajout d'un centre de vote

3.3 Navigation géographique :

❖ Géo localisation :

Construction : Une requête de géocodage passe par un objet de la classe Geocoder.

Celle-ci ne possède qu'une seule méthode, *geocode*, qui permet d'envoyer la requête :

- ✓ *geocode* : prend en argument la question – un objet de type GeocoderRequest –, une fonction de retour qui nous permettra de faire l'appel AJAX vers le service Google et un code de retour

Spécification de la réponse :

Lorsque l'on envoie une question de géocodage via l'API, le service répond par un objet de type GeocoderResult. On peut extraire de cette réponse plusieurs informations par exemple :

- les coordonnées géographiques : propriété *geometry.location* de type LatLng .
et voici l'extrait de code

```
<? page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1">
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>geolocalisation automatique</title>
<script type="text/javascript" src="http://maps.google.com/maps/api/js?sensor=false">
</script>
```

L'appel a l'API Google Maps

Figure 49: appel a l'api Google maps

```

var geocoder = new google.maps.Geocoder();
var addr, latitude, longitude;

function geolocalise(){
    addr = document.getElementById('check').value;
    geocoder.geocode( { 'address': addr}, function(results, status) {
        if (status == google.maps.GeocoderStatus.OK)
            latitude = results[0].geometry.location.lat();
            longitude = results[0].geometry.location.lng();
            document.getElementById('lat').value = latitude;
            document.getElementById('lng').value = longitude;});
        var marker = new google.maps.Marker({
            map: map,
            position: results[0].geometry.location
        });
    }
}

```

la classe de geocodage

Recuperation des coordonnees geographique

Figure 50:extrait de code pour la géo localisation

➤ géo localisation automatique :

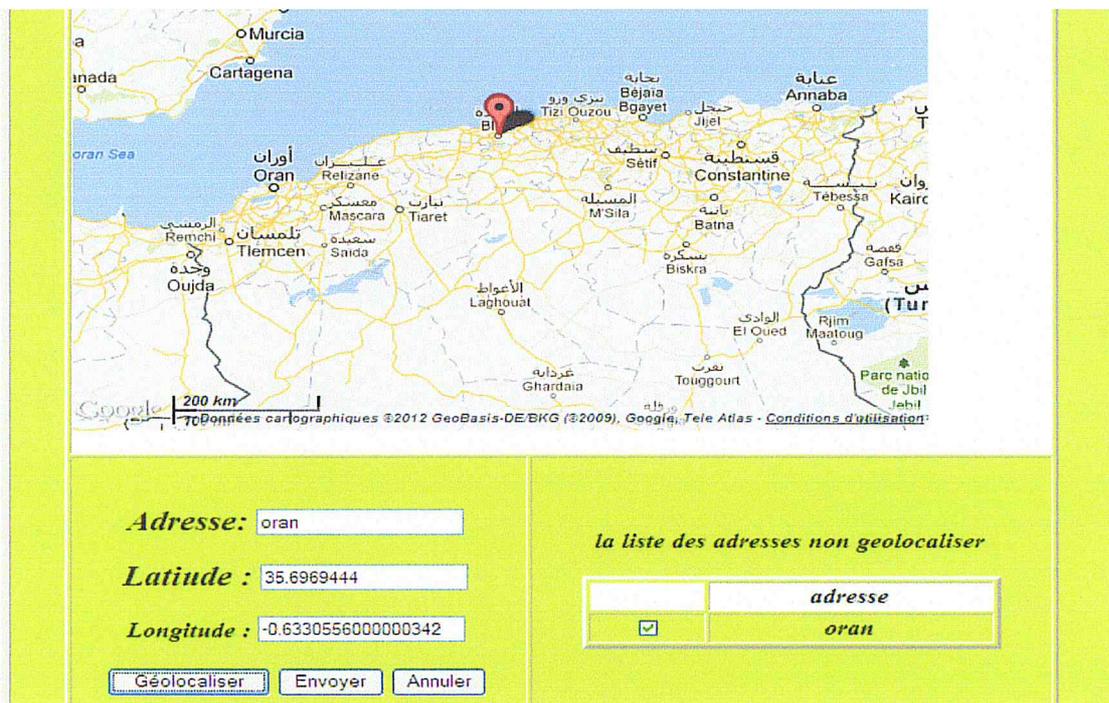


Figure 51:geo localisation automatique



géo localisation manuelle :

cette tache se fait lorsque les adresses ne sont pas reconnue par Google maps

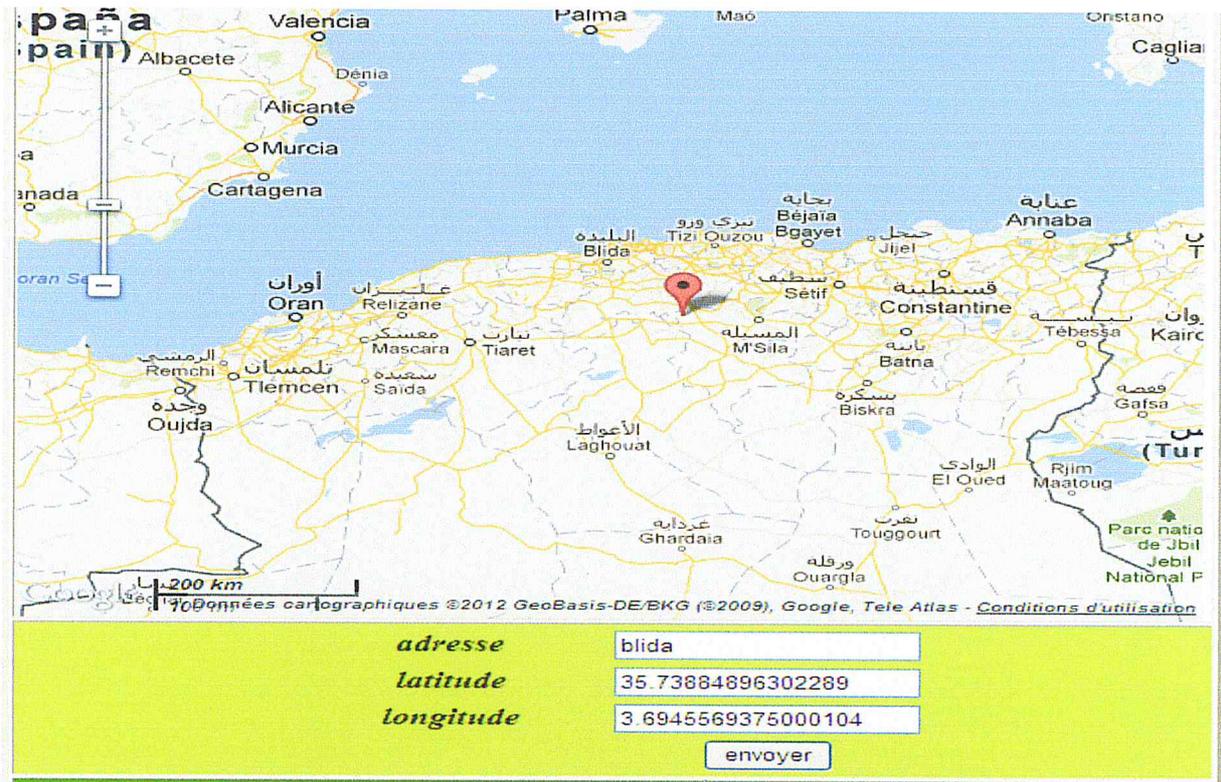


Figure 52:geo localisation manuelle

3.4 Affectation automatique :

Après avoir toute les adresses, l'acteur effectue cette tache qui consiste a choisir le centre le plus proche en utilisant le service d'itinéraire

La méthode de fonctionnalité de ce service est très similaire a celle du géocodage

Construction du service :

Le service d'itinéraire de l'API passe par les classes DirectionsService et Directions-Rendered. La première étant le constructeur du service et la seconde la classe qui exploite le résultat de la requête.

Et voici un extrais de code :

```

    });
    idColor++;

    directionsDisplay = new google.maps.DirectionsRenderer(renderOptions);
    directionsDisplay.setMap(map);
    directionsService.route(request, function(response, status) {
        if (status == google.maps.DirectionsStatus.OK) {
            directionsDisplay.setDirections(response);
            var distance = response.routes[0].legs[0].distance.value;
            var duree = response.routes[0].legs[0].duration.value;
            var outputDiv = document.getElementById('outputDiv');

            outputDiv.innerHTML+= '<table id=tableau border = 1 style="float:right;width:300px height:300px">'
                + '<tr><td> trajet </td>'
                + ' <td><a href="#" onclick="sortTable(this,1); return false;">Distance : </td> </a>'
                + ' <td> Durée estimée en voiture : </td>'
                + '<td> latitude:</td>'
                + '<tr><td>+ trajet[0] + ' vers ' + trajet[1]+</td>'
                + ' <td>+response.routes[0].legs[0].distance.text + ' </td>'
                + ' <td>+ response.routes[0].legs[0].duration.text + ' </td>'
                + ' <td><a href="test.jsp?dec='+trajet[0].split(",")[0]+'">+trajet[0].split(",")[1]+ ' </a></td>'
    
```

Classe de service d'itinéraire

Récupération de calcul

Figure 53:extrait de code pour le calcul d'itinéraire

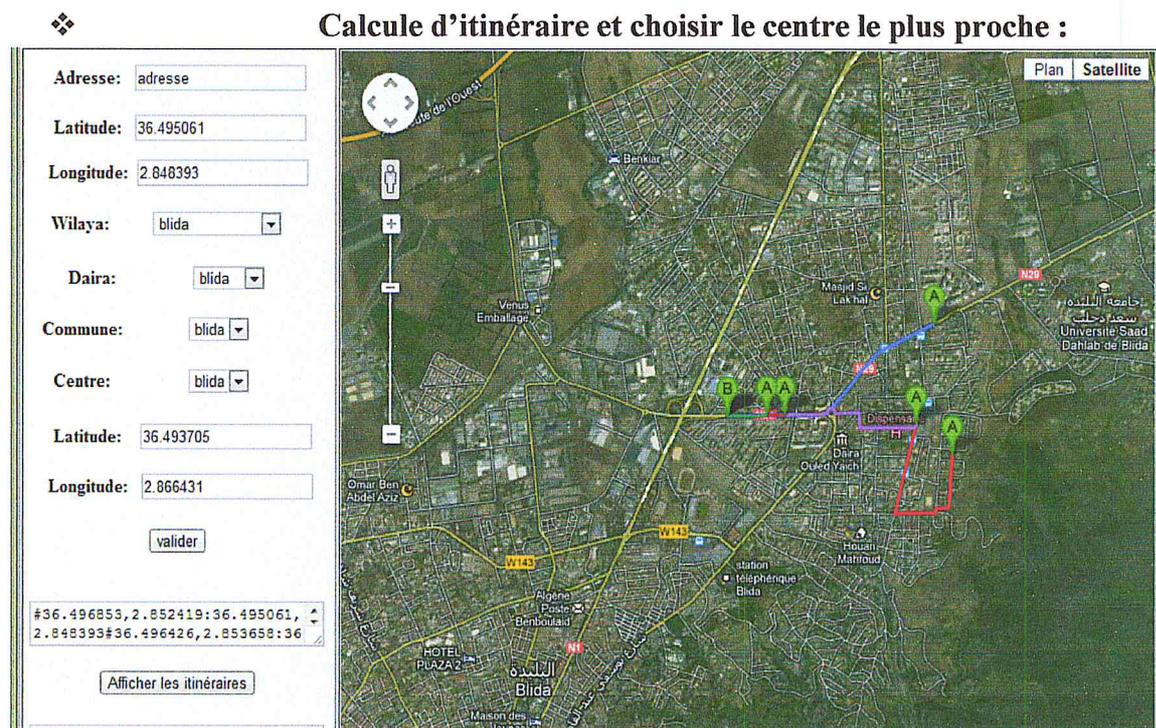


Figure 54:tracer itinéraire et choisir le centre

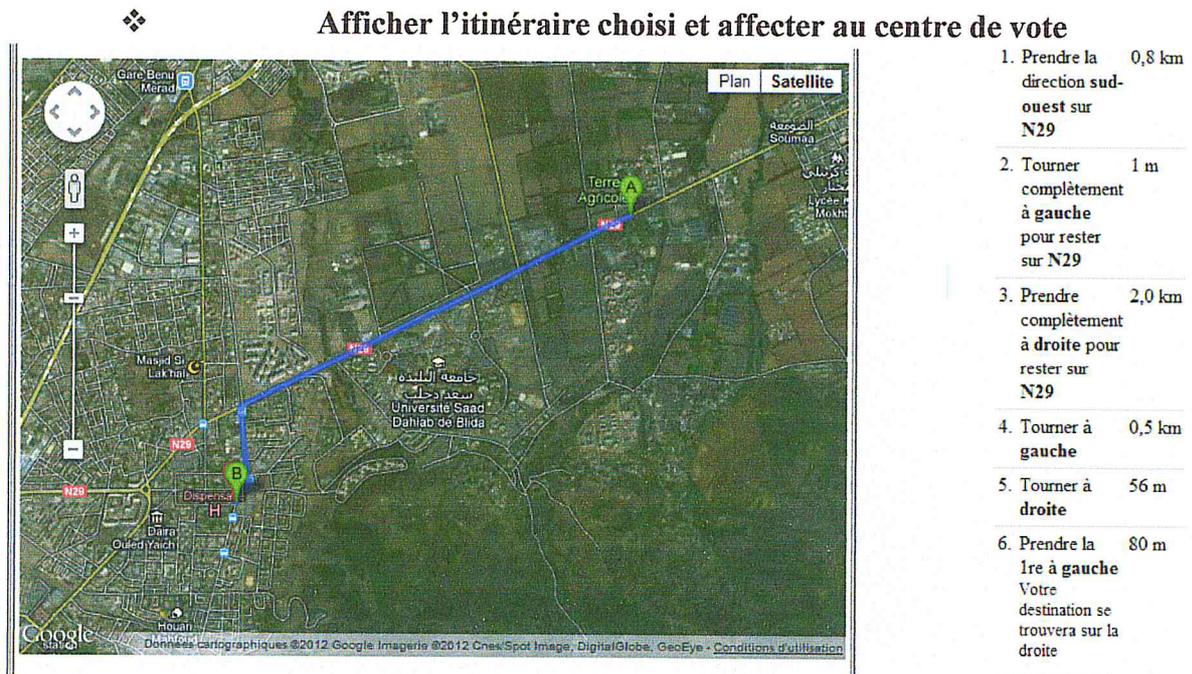


Figure55:afficher itinéraire choisi et affecter au centre de vote

Conclusion :

Dans ce chapitre nous avons vu un aperçu général sur les fonctionnalités des services de l'API Google Maps et nous avons montré quelques fonctionnalités de notre système.

Conclusion et perspective:

Les services de cartographie on ligne occupent aujourd'hui une place de plus en plus importante pour les développeurs d'application cartographique; l'objectif de notre travail été de concevoir un e application qui exploite ces nouvelles services de géocodage et calcul d'itinéraire grâce à une api d'un service cartographique qui est pour notre cas Google Maps pour assurer l'affectation des votants au centres de vote.

Ce travail ne c'est pas fait sans difficultés; en effet; il à fallu apprendre à utiliser les différentes technologies web 2.0(ajax, css...)et de les manipuler avec un service externe qui est Google maps.

Toute la complexité à résidé dans le choix d'une méthode ou d'un processus pour faire une bonne conception. Vient ensuite la conception et réalisation de cette application qui à également été une tache très intéressante.

Ce champ d'étude étant très récent, son étude c'est donc imposée. Nous avons développé une application d'affectation des votants aux centres de vote à l'aide de l'information cartographique de Google.

Même si l'objectif de notre travail a été atteint, il faut cependant rappeler que cette Application est amenée à évoluer et à prendre en compte les extensions suivantes :

- Enregistrer les données métier dans la base de données de Google maps
- Calculer un itinéraire en évitant les obstacles, trouver un raccourci, ce qui permettrait de créer d'autre itinéraire sur Google maps et aussi enrichir la base de données.

Bibliographie

(CANCES. M, 92): http://fr.wikipedia.org/wiki/L'Information_g%C3%A9ographique

(CUENIN. R, 72): http://fr.wikipedia.org/wiki/L'Information_g%C3%A9ographique

(MOUSSA. M, 93): www.programmez.com/livres/17_chapitre2.pdf

(TAYLOR. D. R. F, 91): <http://www.touraineverte.com/google-maps-api-version-3/migration-code-v2-v3/migration-evolution-difference-api-google-maps-v2-v3.html>

(WEGER. G, 99): www.programmez.com/livres/17_chapitre2.pdf

<https://developers.google.com/maps/>

<http://blog.thecodingmachine.com/fr/content/google-maps-api>

<http://www.touraineverte.com/google-maps-api-version-3/migration-code-v2-v3/migration-evolution-difference-api-google-maps-v2-v3.html>

<http://www.html5-css3.fr/html5/tutoriel-api-google-maps-geolocalisation-html5>

Pierre –Alain Muller, Nathalie Gartner, Modélisation objet avec UML EYROLLES, 2002

<http://www.commentcamarche.net>

http://www.programmez.com/livres/17_chapitre2.pdf

<http://www.cyber06.com/article/mvc.php>

Conception et réalisation d'une application MASHUP pour les chercheurs Algériens à travers le monde. DAHMANE HAMZA, MAHFOUD SAMI , Institut National de formation en Informatique (I.N.I)Oued-Smar, Alger, promotion 2007/2008.

Conception et réalisation selon une approche MVC mettant en oeuvre le framework Struts2, d'un composant représentant l'état civil d'une APC, SEDDAOUI Wassila et YAHIAOUI Madina, Promoteur.