

جمهورية الجزائرية الديمقراطية الشعبية
R_ E P U B L I Q U E A L G E R I E N N E D E M O C R A T I Q U E E T P O P U L A I R E
وزارة التعليم العالي و البحث العلمي
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
جامعة عمّار ثليجي بالأغواط
UNIVERSITE AMAR TELIDJI LAGHOUAT
كلية العلوم
FACULTE DES SCIENCES
DEPARTEMENT D'INFORMATIQUE

Mémoire de MASTER

Filière : Informatique
Option : Réseaux et systèmes répartis

Par :
Boudefla Mohamed

THEME

*Optimisation par critère de proximité géographique de
l'affectation des votants aux centres de vote, en
utilisant l'API de Service Web de géolocalisation de
Google Maps*

Soutenu publiquement devant le jury composé de:

M.
M.
M.
M. Ali Abbassene

Attaché de recherche

*Président
Examinateur
Examinateur
Encadreur*

Année Universitaire 2012/2013

Remerciements

Au terme de ce travail je tiens à remercier tout d'abord ALLAH le tout puissant de m'avoir donné le courage et d'avoir facilité la réalisation de ce travail.

J'exprime mon grande reconnaissance et remerciement à mon promoteur

Mr : Ali ABBASSENE pour son aide, ses conseils et de m'avoir orienté et mis à mon disposition tous les moyens nécessaires durant mon travail.

Je remercie aussi Monsieur le Directeur du Centre de Développement des Technologies Avancées(C.D.T.A).

Mes remerciements les plus vifs s'adressent aussi aux messieurs le président et les membres de jury d'avoir accepté d'examiner et d'évaluer mon travail.

J'exprime également mon gratitude à tous les enseignants qui ont collaboré à mon formation Master II et tous ceux qui ont participé de près ou de loin à la réalisation de mon mémoire.

Enfin, j'adresse mes plus sincères remerciements à mes parents pour leur contribution, leur soutien et leur patience et à tous mes proches.

Merci à tous et à toutes.

BOUDEFLA MOHAMEM

Résumé

Le eVOTE est un exemple d'application du domaine du eGouvernement(eAdministration) permettant de mettre en ligne les divers services et activités offerts par un service de vote classique d'une APC. Mais, le talent d'Achille d'un tel système est le module d'affectation des votants à leurs bureaux de vote respectifs, car elle doit se faire manuellement.

Pour résoudre le problème de l'affectation des votants aux divers centres de vote, l'utilisation des techniques géographiques peut constituer une alternative prometteuse.

Notre projet vise à développer un module d'affectation géographique des votants, intégré au système eVOTE, qui permet de réaliser une affectation semi-automatique.

La procédure d'affectation se fera selon des critères bien précis à savoir la proximité géographique du centre de vote. Pour ce faire on exploite l'API Web de Google Maps afin de retrouver de manière graphique, le plus court chemin entre le votant et le centre de vote.

Notre module d'affectation permettra de rapprocher les centres de votes des citoyens, et diminuera la charge sur les centres en équilibrant les affectations. Du côté des officiers service de vote, ce système va accélérer l'opération d'affectation des votants et la rendra plus fluide.

MOTS-CLES: eGouvernement/eAdministration, Système de eVOTE, module d'affectation des votants, services Web, API Google Maps, SIG.

Abstract

The eVOTE is an example of the different applications in the field of eGovernment (eAdministration) to put online the various services and activities offered by a vote service with a classical APC. But the problem of such a system is that the process of voter's assignment to their respective polling stations is done manually.

To solve the problem of assigning voters to different polling stations, the use of geographical techniques can be a promising alternative.

Our project aims to develop a geographic voter's assignment module integrated to eVOTE system, which allows a semi-automatic assignment.

The allocation process will be based on specific criteria as the geographic proximity of the polling center .To do this we use the Web API of Google Maps to find a graphical way, the shortest path between the voter and the voting center

Our assignment module will bring closer the voting centers to citizens, and reduce the pressure on certain centers by balancing the assignments. On the side of the voting service officers, the system will accelerate the voter's assignment operation and make it more fluid.

KEYWORDS: eGovernment / eAdministration, eVOTE System, Voter's assignment module, web services, Google Maps API, GIS

ملخص

eVOTE مثال تطبيقي في مجال الحكومة الإلكترونية يسمح بوضع على شبكة الإنترنت مختلف الخدمات والنشاطات المهداة من قبل خدمة الانتخاب الكلاسيكية للبلدية. لكن، نقطة الضعف لهذا النظام تتمثل في وحدة التعيين للناخبين الخاص بهم، لأنها تتم يدويا.

لحل مشكلة تعيين المنتخبين إلى مختلف مراكز الاقتراع، استعمال التقنيات الجغرافية يمكن أن تشكل حلا واعدة.

يهدف هذا المشروع إلى تطوير وحدة التعيين الجغرافية للناخبين ، المدمج في نظام eVOTE والذي يسمح بتحقيق التعيين النصف الآلي.

إجراءات التعيين ستنم حسب مقاييس مدققة بأخذ الاعتبار القرب الجغرافي لمركز الانتخابات. لتحقيق ذلك نستعمل واجهة برمجة تطبيقات الويب لخرائط جوجل من أجل إيجاد الطريق الأقصر ما بين الناخب و مركز الاقتراع.

تسمح وحدة التعيين بتقريب مراكز الاقتراع للمواطنين ، وتقلل من الاكتظاظ في مراكز الاقتراع بتوازن تعيينات الناخبين. من جهة ضباط خدمة الانتخاب ، هذا النظام يسرع عملية تعيين الناخبين ويجعلها أكثر مرونة.

Sommaire

Introduction générale.....	1
Chapitre I : Etude de l'existant	
I.1. Introduction.....	3
I.2. Gestion de l'opération de vote	4
I.2.1. 1 ^{er} Phase : Faire une demande de vote.....	4
I.2.1.1. Inscription.....	4
I.2.1.2. Radiation	5
I.2.1.3. Modification	5
I.2.2. 2 ^{ème} Phase : Validation de demandes	6
1. Validation par l'officier d'état civil valideur.....	6
2. Validation finale par la commission	7
I.2.3. 3 ^{ème} Phase : L'affectation des votants aux différents centres et bureau de vote.....	8
I.2.4. 4 ^{ème} Phase : Traitement des recours.....	9
I.3. Les problématiques.....	9
I.4. Conclusion	9
Chapitre II : Généralisation sur les services Web	
II.1. Introduction	10
II.2. Définition des services Web	11
II.3. Apports des services Web	11
II.4. Architectures orientées services.....	12
II.5. Les services Web classiques.....	13
II.5.1. Introduction.....	13
II.5.2. Infrastructure des services Web (SOAP).....	13
II.5.2.1. Communication avec SOAP	13
II.5.2.2. Description des services Web WSDL.....	13
II.5.2.3. Découverte des services Web.....	14
II.6. Les services Web REST	14
II.6.1. C'est quoi REST	14
II.6.2. Les concepts de REST	14
II.6.3. Les principes de REST	15
II.7. Les APIs Web	17
II.8. Conclusion.....	17

Chapitre III : Analyse et Conception

III.1. Introduction	18
III.2. Architecture de l'application	18
III.2.1. Les principes de l'API	18
III.2.2. Le fonctionnement de système	19
III.3. Présentation de la démarche utilisée	20
III.3.1. La modélisation du système	20
III.3.2. Méthodologie de conception	21
III.3.3. Le processus UP/XP	21
III.3.4. Caractéristiques du processus suivi	21
III.4. Les modèles du processus UP/XP	22
III.4.1. Modèle des cas d'utilisation	23
III.4.1.1. Les cas d'utilisation de l'application eVOTE	23
III.4.1.2. Les cas d'utilisation de notre module d'affectation	26
III.4.2. Modèle de conception préliminaire	31
III.4.2.1. Diagrammes de collaboration	31
III.4.2.2. Diagrammes de séquence système	34
III.4.2.3. Diagramme de classe de conception	39
III.4.2.4. Conception de la base de données (schéma relationnel)	45
III.4.3. Modèle de Navigation	46
III.4.4. Modèle MVC (Modèle Vue Contrôleur)	49
III.5. Conclusion	51

Chapitre IV : Implémentation

IV.1. Introduction	52
IV.2. Environnement de développement	52
IV.2.1. Les technologies utilisées	52
IV.2.1.1. Google Maps	52
IV.2.1.2. AJAX	54
IV.2.1.3. Autres technologies utilisées	55
IV.2.2. Outils de développement	55
IV.2.2.1. Struts2	55
IV.2.2.2. Langage de développement utilisé	56
IV.2.2.3. Environnement de développement intégré	57
IV.2.2.4. Choix du SGBD	57

IV.2.2.5. Choix du serveur Web.....	57
IV.3. Illustration de l'application développée.....	58
IV.3.1. Page d'accueil de l'application.....	58
IV.3.2. Espace officié d'état civil valideur (OECV).....	60
IV.3.3. Espace administrateur	62
IV.3.4. Espace officier du service de vote (OSV).....	62
IV.3.5. Espace citoyen	66
IV.4. Conclusion	68
Conclusion générale et perspectives.....	69

Liste des sigles et acronymes

AJAX	Asynchrones JavaScript and XML.
APC	Assemblée Populaire Communale.
API	Application Programming Interface.
DCU	Diagramme de Cas d'Utilisation.
EJB	Entreprise Java Bean.
HTML	Hyper Text Markup Language.
HTTP	HyperText Transfer Protocole.
IDE	Integrated Development Environment.
JDBC	Java Data Base Connectivity.
JSF	Java Server Faces.
JSP	Java Server Pages.
J2EE	Java 2 Entreprise Edition.
MVC	Model Vue Contrôleur.
OEKV	Officier d'Etat Civil Valideur.
OMG	Object Management Group.
OMT	Object Modeling Technique.
OSV	Officier du Service de Vote.
REST	Representational State Transfer.
ROA	Resource Oriented Architecture.
SIG	Système d'Information Géographique.
SOA	Service Oriented Architecture.
SOAP	Simple Object Access Protocol.
SGBD	Système de Gestion de Base de Donnée.
MySQL	Structured Query Language.
UDDI	Universal Description Discovery and Integration.
UML	Unified Modeling Language.
UP	Unified Process.
URI	Uniform Resource Identifier.
URL	Uniform Resource Locator.
XML	Extensible Markup Language.
WSDL	Web Services Description Language.

W3C World Wide Web Consortium.

XP eXtreme Programming.

Liste des figures

Figure 1: Diagramme général d'une APC	3
Figure 2: Architecture générale du module d'affectation.....	19
Figure 3: Diagramme de cas d'utilisation générale pour l'application eVOTE	24
Figure 4: DCU de création d'un compte.....	26
Figure 5: DCU de validation (refus) les informations de la création d'un compte.....	27
Figure 6: DCU de l'ajout d'un centre de vote.....	28
Figure 7: DCU d'affectation ou réaffectation les votants au centre de vote.....	29
Figure 8: Diagramme de collaboration pour la création d'un compte	31
Figure 9: Diagramme de collaboration pour la validation des informations d'un compte.....	32
Figure 10: Diagramme de collaboration pour l'ajout d'un centre de vote	32
Figure 11: Diagramme de collaboration pour l'affectation ou réaffectation des votants au centre de votes.....	33
Figure 12: Diagramme de séquence pour la création d'un compte.....	34
Figure 13: Diagramme de séquence pour la validation des comptes	35
Figure 14: Diagramme de séquence pour l'ajout d'un centre de vote	36
Figure 15: Diagramme de séquence pour l'affectation ou réaffectation des votants au centre de vote (par individu)	37
Figure 16: Diagramme de séquence pour l'authentification.....	38
Figure 17: Diagramme des classes de l'application eVOTE.....	40
Figure 18: Diagramme des classes pour le module d'affectation	42
Figure 19: Début des diagrammes de navigation des acteurs internautes et utilisateur	47
Figure 20: Diagramme de navigation de validation les informaions de création des comptes	48
Figure 21: Diagramme de navigation d'affectation ou réaffectation des votants au centre de vote par individu.....	49
Figure 22: Diagramme de classe participante d'ajout d'un centre de vote	50
Figure 23: Chemin d'une requête dans Struts 2	56
Figure 24: Page d'accueil de l'application eVOTE.....	58
Figure 25: Inscription d'un nouvel utilisateur	59
Figure 26: Consultation la liste des comptes à valider	60
Figure 27: Vérification les informations d'un compte	61
Figure 28: Localiser la position de citoyen sur la carte géographique	62

Figure 29: Localiser la position de centre de vote sur la carte géographique.....	63
Figure 30: Consultation la liste des cités	64
Figure 31: Affectation ou réaffectation des votants par cité « 1/2»	65
Figure 32: Affectation ou réaffectation des votants par cité « 2 /2»	66
Figure 33: Consultation les détails du chemin emprunté.....	67

Liste des Tableaux

Tableau 1: Le format du registre de préinscription	6
Tableau 2: Le format du registre de préradiation	6
Tableau 3: Le format du registre de duplicata	7
Tableau 4: Le format du registre d'inscription finale.....	8
Tableau 5: Le format du registre de radiation finale	8
Tableau 6: Les domaines métier de Google REST API	18
Tableau 7: Les modèles du processus UP/XP.....	22
Tableau 8: Description du DCU de création d'un compte	27
Tableau 9: Description du DCU de validation (refus) les informations de la création d'un compte.....	28
Tableau 10: Description du DCU de l'ajout d'un centre de vote.....	29
Tableau 11: Description du DCU d'affectation ou réaffectation des votants au centre de vote.....	30
Tableau 12: Dictionnaire de données pour le module d'affectation	43

Introduction générale

Aujourd'hui, Internet et les technologies de l'information et de communication deviennent de plus en plus indispensables dans la vie du citoyen. Car elles tissent des liens entre lui et les organismes de tout type d'activité que ce soit commerciale, culturelle, sociale, ou gouvernementale. Le citoyen devrait impérativement bénéficier à son tour de ces nouvelles technologies qui lui rendent la vie plus facile.

Dans cette optique, le projet eVOTE vise à mettre en place une application Web qui permet de publier sur Internet les divers services et activités offerts par un service de vote d'une APC (Assemblée Populaire Communale). Ce projet rentre dans le cadre du domaine du e-Gouvernement (e-Administration) et plus exactement dans la e-Commune. Il est associé à un autre projet e-EtatCivil, visant à informatiser les services d'état civil de l'APC. Ces services, vote et état civil, sont fortement sollicités par les citoyens d'une commune.

Actuellement, une première version du système eVOTE (réalisé par le CDTA) existe mais souffre de grandes lacunes parmi lesquels on peut citer: l'affectation manuelle fastidieuse des votants sans aucun critères (géographique) de sélection du centre de vote par rapport au domicile du votant.

A cet effet, et dans le but d'améliorer l'opération d'affectation des votants au niveau des services de vote, nous avons procédé à l'automatisation de ce processus à travers le développement d'un module d'affectation géographique des votants.

On peut dénombrer plusieurs critères d'affectation, mais la proximité du votant au centre de vote reste le critère principal et le calcul de la métrique "distance géographique" est primordial à toute bonne affectation. Notre solution consiste à réaliser une affectation semi-automatique qui se fera selon des critères de proximité géographique en utilisant l'API des services Web de Google Maps pour trouver le plus court chemin entre le votant et son centre de vote.

Le présent mémoire est structuré comme suit :

Dans chapitre I nous étudierons l'organisation et les différentes fonctions du service de vote de l'APC.

En ce qui concerne le chapitre II, on va enchaîner avec un état de l'art sur les services Web, l'architecteur SOA, les services web classique et les web service REST, leurs principes et leurs concepts.

Dans le chapitre III, nous présenterons dans sa première partie le processus suivi pour le développement. Dans la deuxième partie nous décrirons la conception du module d'affectation géographique.

Le chapitre IV est consacré à la réalisation de l'application. On y présente des illustrations de l'application développée.

Enfin, une conclusion et des perspectives cloront notre travail.

Chapitre I

Etude de l'existant

I.1. Introduction

Nous allons présenter dans ce chapitre, le service de vote qui appartient à chaque APC en Algérie.

Une assemblée populaire communale (APC) est une collectivité locale, dirigée par un maire dont il est élu par le peuple. Elle est composée d'un certain nombre d'élus géré par un secrétaire générale (SG) sous tutelle du ministre de l'intérieur.

L'organigramme suivant représente l'organisation de l'APC. Notre objectif d'étude se limite au service de vote.

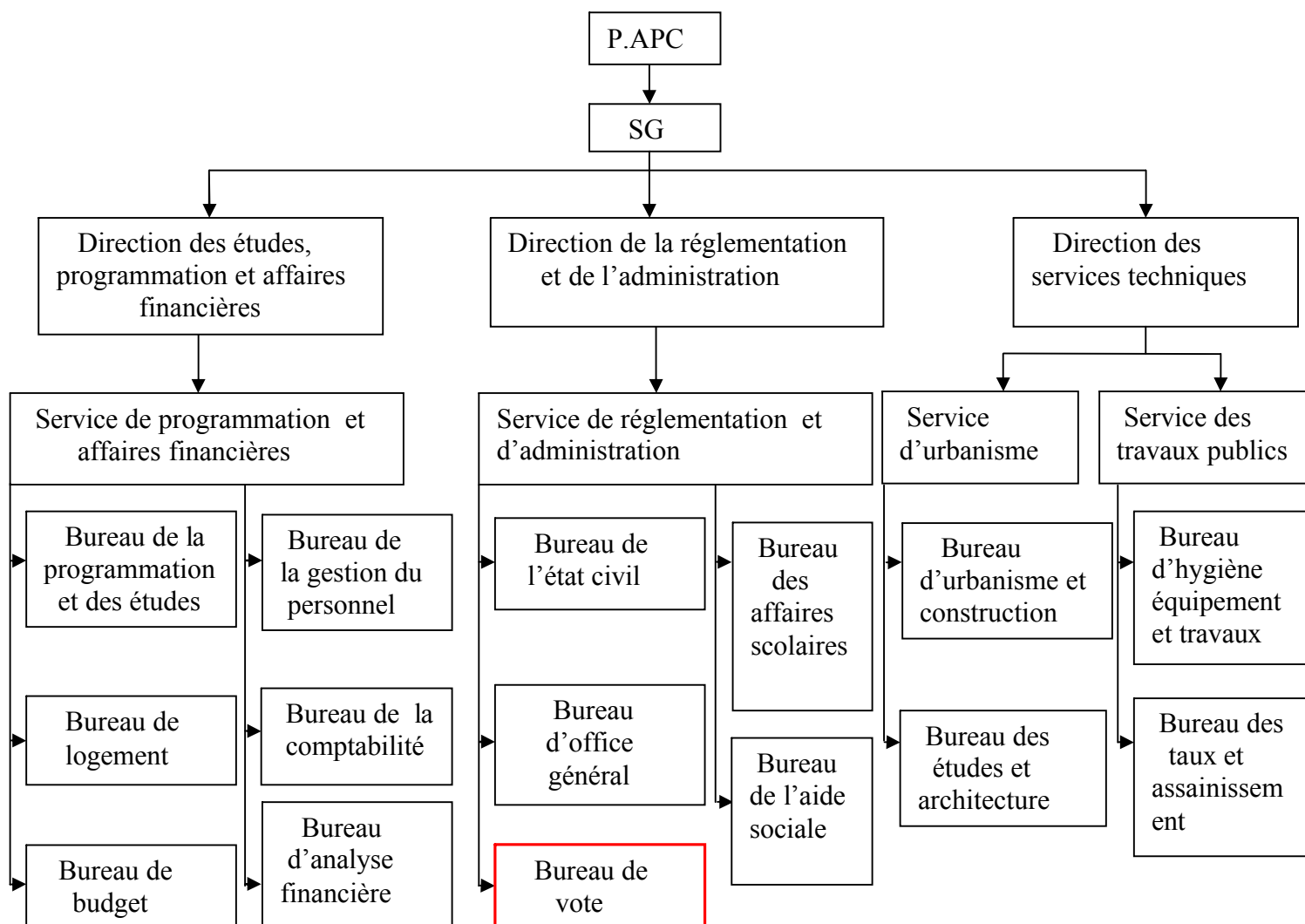


Figure 1:Diagramme général d'une APC

I.2. Gestion de l'opération de vote

En Algérie plusieurs acteurs participent à l'opération de vote (chef du service de vote, juge,.....etc.), Cette dernière est composée de quatre phases :

I.2.1. 1^{er} Phase : Faire une demande de vote

Les services de vote que l'on a visité offrent trois grandes opérations : inscription, radiation de la liste électorale et modification de la carte de vote.

I.2.1.1. Inscription

Le citoyen peut demander son inscription sur la liste électorale dans les cas suivants:

Cas 1 : Inscription pour la 1^{er} fois (âge 18ans)

Dans ce cas, l'inscription dans la liste électorale ne demande que:

- a) Une photocopie de la carte nationale ou passeport ;
- b) Un livret de famille ou extrait de naissance ;
- c) Une pièce justificative de la résidence (quittance d'électricité, eau, ...).

Cas 2 : Inscription pour la 1^{er} fois avec retard (âge + 18ans)

Dans ce cas, l'inscription dans la liste électorale ne demande que:

- a) Une photocopie de la carte nationale ou passeport ;
- b) Un livret de famille ou extrait de naissance ;
- c) Une pièce justificative de la résidence (quittance d'électricité, eau, hébergement ...) ;
- d) Une déclaration sur l'honneur de non inscription sur liste électorale.

Cas 3 : Changement de résidence

Dans ce cas, le citoyen doit commencer par demander sa radiation de la liste électorale d'origine et de s'inscrire dans la liste électorale de la nouvelle commune comme suit :

Etape1 : Dans la commune d'origine

- Il doit déposer sa carte de vote ;
- Il faut remplir un formulaire demande de radiation de la liste électorale ;
- Il faut déposer un certificat sa déclaration de changement de résidence.

Etape2 : Dans la nouvelle commune

- Il faut munir d'une pièce justificative de sa radiation de la liste électorale de la commune d'origine ;
- Il faut avoir un certificat de changement de résidence ;
- Il faut avoir une photocopie de la carte nationale ou passeport ;
- Il faut avoir un livret de famille ou extrait de naissance ;
- Il faut avoir une pièce justificative de la résidence (quittance d'électricité, ...).

I.2.1.2. Radiation

Le citoyen peut demander sa radiation de la liste électorale dans les cas suivants:

Cas 1 : Changement de résidence

Dans ce cas, la radiation de la liste électorale ne demande que:

- a) De déposer sa carte de vote ;
- b) De remplir un formulaire de demande de radiation de la liste électorale de l'ancien APC ;
- c) De déposé sa déclaration de changement de résidence.

Cas 2 : Décès

Dans ce cas, la radiation de la liste électorale ne demande que:

- a) De déposer la carte de vote du défunt(e) ;
- b) De déposer la déclaration de décès.

I.2.1.3. Modification

Le citoyen peut avoir une 2^{ème} carte (carte duplicata) avec les modifications faites en cas des problèmes suivants :

Cas 1 : Erreur de saisie de sa carte de vote

Dans ce cas, le citoyen remplit une demande pour rectification.

Cas 2 : Perte de la carte

Dans ce cas, le citoyen dépose une déclaration de perte de sa carte de vote.

Il y a un autre type de modification concerne l'APC au cas où celle-ci procède à certains changements comme modification du centre de vote ou du bureau de vote, elle envoie des nouvelles cartes de votes aux citoyens concernés.

I.2.2. 2^{ème} Phase : Validation de demandes

La validation des demandes se déroule en deux étapes qui comprennent la validation par l'officier d'état civil valideur et la validation par la commission comme suit:

1. Validation par l'officier d'état civil valideur

Le rôle de l'officier d'état civil valideur est de réceptionner les dossiers des citoyens que ce soit pour une demande d'inscription, radiation ou de modification au niveau de l'APC. Les informations contenues dans ces dossiers sont vérifiées par l'officier d'état civil valideur un par un.

Dans le cas, où les informations sont correctes. L'officier d'état civil valideur déclare le citoyen sur le registre de préinscription dans le cas d'inscription, ou sur le registre de préradiation s'il s'agit d'une radiation.

Dans ce cas, l'officier d'état civil valideur donne au citoyen un récépissé, attestant de son inscription ou radiation provisoire, contenant un numéro séquentiel qui doit être le même que celui du registre (préinscription ou préradiation). Le tableau 1 et le tableau 2 montre respectivement le format du registre de préinscription et de la préradiation.

Tableau 1: Le format du registre de préinscription

Numéro	Nom Prénom	Date et lieu de naissance	Prénom père	Nom Prénom Mère	Profession	Adresse
111	Bala Hamza	24 /08 /1982 Alger	Mourad	Dali Amina	Etudiant	Cité 20 Août Bordj el Behri

Tableau 2: Le format du registre de préradiation

Numéro	Nom Prénom	Date et lieu de naissance	Prénom père	Nom et Prénom Mère	Profession	Adresse
22	Meddid Ali	22 /05 /1981 Alger	Farid	Dali Khadija	Ingénieur	Cité Jolie Vue 2 Kouba

Dans le cas d'une demande de modification, l'officier d'état civil valideur corrige les informations personnelles du citoyen concerné, puis il donne au citoyen une autre carte (carte

duplicata) avec les modifications faites (changement de centre ou de bureau, changement de nom,....etc) sans passer par la commission mais sera inscrit dans le registre duplicata. Le tableau 3 montre le format du registre duplicata.

Tableau 3: Le format du registre de duplicata

Numéro	Nom Prénom	Date et lieu de naissance	Prénom père	Nom et Prénom Mère	Profession
75	Media Samir	25 /05 /1970 Alger	Mahdi	Amrani Halima	Sans profession
Adresse			Motif de modification		
Cité des 255 Logements Mohammadia			Ex : Perte de la carte		

2. Validation finale par la commission

Une fois tous les citoyens sont enregistrés dans le registre de préinscription ou de préradiation, une commission se réunit pour valider la liste électorale.

La commission est composée de trois membres :

- Le président de la commission : juge ou procureur.
- Le secrétaire de la commission : le chef du service de vote.
- Le secrétaire général de l'APC.

La commission s'organise dans les deux cas suivants :

- **Révision Annuelle** : chaque année du 1^{er} au 31 Octobre.
- **Révision exceptionnelle** : demandée par l'état avant chaque élection nationale pour une période de 10 jours.

Le juge a la responsabilité d'ouvrir et de fermer la révision que ce soit une révision annuelle ou exceptionnelle, pour traiter les demandes d'inscriptions ou radiations. Il peut refuser ou accepter les dossiers selon des clauses juridiques.

Le chef du service de vote et le secrétaire général de l'APC peuvent refuser ou accepter les dossiers selon des clauses administratives (exemple : manque d'un document).

Les citoyens acceptés par la commission sont inscrits sur le registre final (inscription ou radiation).

Le format du registre d'inscription finale et de radiation finale sont représentés respectivement dans le tableau 4 et le tableau 5.

Tableau 4: Le format du registre d'inscription finale

Numéro	Nom Prénom	Date et lieu de naissance	Prénom père	Nom Prénom Mère	Profession
23	Zebar Mohamed	22 /04 /1979 Alger	Halim	Amrani Jamila	Menuisier
Adresse		Documents Présentés		Verdict de la commission administrative	Observation
Cité 632 Logements Mohammadia		Livret de famille Carte nationale Certificat de résidence		Acceptée	-----

Tableau 5:Le format du registre de radiation finale

Numéro	Nom Prénom	Date et lieu De naissance	Prénom père	Nom Prénom Mère	Profession
45	Mehani Ahmed	22 /05 /1981 Alger	Hocine	Amrani Zohra	Médecin
Adresse		Motif de radiation	Verdict de la commission administrative		Observation
Cité Radieuse Oued Smar		Changement de résidence	Acceptée		-----

I.2.3. 3^{ème} Phase : L'affectation des votants aux différents centres et bureau de vote

Le rôle de l'officier du service de vote est d'affecté chaque citoyen accepté par la commission et qui est inscrit sur le registre final d'inscription à un centre de vote le plus proche qui est composé de plusieurs bureaux. Par la suite, il l'inscrit sur la liste électorale. Dans ce cas, le citoyen peut recevoir sa carte de vote.

Dans le cas de radiation, l'officier du service de vote radie tous les citoyens qui sont inscrits sur le registre de radiation finale. Dans ce cas le citoyen peut avoir l'accusé de radiation de l'ancien APC.

I.2.4. 4^{ème} Phase : Traitement des recours

Les citoyens dont les dossiers sont refusés par la commission ont le droit de présenter des recours après la fermeture de la commission pendant 8 jours, que ce soit des recours administratifs ou des recours juridiques et c'est le président d'APC qui a la responsabilité de traiter ces recours.

Dans le cas où un recours est accepté par le président de l'APC, celui-ci donne l'instruction à l'officier du service de vote pour inscrire ou radier le citoyen concerné dans la liste électorale.

I.3. Les problématiques

Actuellement les problèmes majeurs que rencontre le service de vote sont:

- Les traitements (inscription, radiation) sont faits manuellement sur des registres (registre préinscription, registre d'inscription finale, registre préradiation, registre radiation finale) ce qui rend difficile et lent les traitements des dossiers.
- La longue file d'attente au niveau de service du vote pour faire une demande d'inscription ou radiation dans la liste électorale et pour récupérer la carte de vote.
- Les déplacements inutiles des citoyens à l'APC pour voir l'état de validation.
- Le risque de perte du récépissé.

I.4. Conclusion

Pour l'instant le projet eVOTE permet de régler la plupart des problèmes de service de vote d'une APC.

Mais ce dernier, ne permet qu'une affectation manuelle et très lourde des votants aux centres de vote, puis aux bureaux de vote, car l'officier utilise le plan de masse de l'APC pour affecter les votants. Ce qui nécessite une affectation automatique basée sur l'information cartographique de Google Maps qui la rendrait plus efficace.

Chapitre II

Généralisation sur les services Web

II.1. Introduction

Les services Web sont une technologie permettant à des applications de dialoguer à distance via un réseau et ceci indépendamment des plates-formes et des langages sur lesquels elles reposent. Pour ce faire, les services Web s'appuient sur un ensemble de protocoles tel que HTTP (Hypertexte Transfert Protocole) et SOAP (Simple Object Access Protocol) standardisant les modes d'invocation mutuels de composants applicatifs.

Les services Web utilisent un système de requête/réponse sans état, qui prend ses racines dans HTTP et qui sont devenus le paradigme de facto pour les sites Web les plus sollicités.

Le Web a évolué en utilisant principalement des systèmes et des protocoles sans état, dans certains cas une certaine forme d'état ou session est nécessaire, ceci est fourni par des objets secondaires appelés Cookies; cela s'est avéré être beaucoup plus pratique et un paradigme hautement évolutif.

Contrairement à notre précédente utilisation du Web principalement dans l'interaction entre utilisateur et le navigateur, les services Web expriment l'interaction entre programmes. La puissance actuelle des services Web réside dans les applications composites, en combinant des éléments provenant de sources multiples derrière une interface graphique unifiée.

Les services Web sont publiés dans des annuaires. Nous trouvons de plus en plus des services Web orientés vers les SIG (Système d'Information Géographique) car aujourd'hui, l'évolution de ce système tend vers une accessibilité pour le Web avec les Web Mapping, en particulier avec Google Maps.

Il existe deux grands standards dans les services Web: REST (Représentationnel State Transfer) et SOAP.

Les services Web REST exposent entièrement ses fonctionnalités comme un ensemble de ressources (URI) identifiables et accessibles par la syntaxe et la sémantique du protocole HTTP.

Les services Web SOAP sont basés sur des technologies XML entre autres les standards SOAP, WSDL et UDDI qui sont utilisés pour l'encapsulation des messages, la description et la publication des services Web et ceci dans des environnements hétérogènes.

Dans ce chapitre nous allons nous attarder sur les services Web REST sur lesquels se base l'API Google Maps que nous avons utilisée.

II.2. Définition des services Web

Le Web est de plus en plus le support privilégié des applications. Les services Web constituent le développement ultime dans ce domaine. Le terme « service Web » est souvent utilisé de nos jours, mais pas toujours avec la même signification, car la signification est tributaire de l'application de cette technologie. Il existe plusieurs définitions pour les services Web :

- Selon W3C: « Un service Web est un logiciel identifié par une URI (Uniform Resource Identifier), (en détaillant plus tard), dont l'interface est publique et les liaisons sont définies et décrites en utilisant XML. L'environnement du service offre le moyen à d'autres logiciels de découvrir celui-ci. Ce service interagit avec les autres services Web en respectant cette définition, donc en utilisant des messages basés sur XML acheminés par des protocoles Internet.» [W3C_02].
- Selon Gilbert Babin : « Les services Web sont des applications qui relient des programmes, des objets, etc. à l'aide de XML et de protocoles Internet standards. Les services Web sont des compléments aux programmes et applications existantes, développées à l'aide de langages tel que Visual Basic, C, C++, C#, Java ou autre, et servent de pont pour que ces programmes communiquent entre eux » [YKH_11].
- Autre définition : « Un service Web est une application accessible à partir du Web. Il utilise les protocoles Internet pour communiquer et utilise un langage standard pour décrire son interface » [MEL_04].

De ces trois définitions on peut conclure qu'un service Web est une brique de base pour créer une architecture de communication, dont les entités sont les applications développées à l'aide des langages de programmation différents.

II.3. Apports des services Web

L'architecture modulaire des services Web, combinée au faible couplage des interfaces associées [AUS_04], permet l'utilisation et la réutilisation de services qui peuvent facilement

être recombinaés à différents autres applications. D'autre part l'utilisation des services Web engendre plusieurs autres avantages tels que **[BAB_03]** :

- L'interopérabilité : C'est la capacité des services Web d'interagir avec d'autres composantes logicielles via des éléments XML et utilisant des protocoles de l'Internet ;
- L'hétérogénéité : Les services Web permettent d'ignorer l'hétérogénéité entre les différentes applications. En effet, ils décrivent comment transmettre un message (standardisé) entre deux applications, sans imposer comment construire ce message ;
- L'auto-descriptivité : Les services Web ont la particularité d'être auto-descriptifs, c'est à dire capables de fournir des informations permettant de comprendre comment les manipuler. La capacité des services à se décrire par eux-mêmes permet d'envisager l'automatisation de l'intégration de services.

II.4. Architectures orientées services

L'architecture orientée service, connue sous l'acronyme SOA (Services Oriented Architecture), est une approche architecturale permettant la création des systèmes basés sur une collection de services développés dans différents langages de programmation, hébergés sur différentes plates-formes avec divers modèles de sécurité et processus métier **[BAR_03]**. L'idée maîtresse de l'architecture orientée service est que tout élément du système d'information doit devenir un service identifiable, documenté, fiable, indépendant des autres services, accessible, et réalisant un ensemble de tâches parfaitement définies **[BOO_04]**.

L'SOA est axée autour de trois concepts fondamentaux **[BEN_09]** :

- **Le fournisseur de services** : il permet l'accès à son service à travers une interface ;
- **Le client de services** : il désigne une personne, un serveur ou une autre application qui accède au service et l'invoque à travers son interface ;
- **L'annuaire de publication**: il joue le rôle d'intermédiaire entre le fournisseur et le client. Les fournisseurs y enregistrent leurs services, et les clients y cherchent le service satisfaisant leurs besoins.

L'SOA encapsule plusieurs avantages bénéfiques pour le domaine de la technologie d'information et de communication. Elle offre la simplicité à travers les concepts de décomposition, de découplage et de réutilisation. En plus, elle participe à la réduction du coût de développement des grands projets et rend le développement plus efficace **[WEE_05]**.

II.5. Les services Web classiques

II.5.1. Introduction

Les services Web classiques sont définis selon le type d'architecture SOA et reposent tous sur un ensemble des standards de base utilisés pour l'échange de données entre applications dans des environnements hétérogènes tel que:

Le protocole SOAP pour l'échange de messages, WSDL pour décrire le service aux utilisateurs et l'annuaire UDDI pour publier les services Web. Ces standards sont présentés dans la section suivante.

II .5.2. Infrastructure des services Web (SOAP)

II.5.2.1. Communication avec SOAP

SOAP (acronyme de Simple Object Access Protocol jusqu'à sa version 1.1) est une grammaire de XML développée par Microsoft, IBM et d'autres, elle est actuellement sous l'égide du W3C [AME_07].

Du fait qu'il est basé sur XML, il permet l'échange de données structurées indépendamment des langages de programmation ou des systèmes d'exploitation. SOAP permet l'échange d'informations dans un environnement décentralisé et distribué, comme Internet, indépendamment du contenu du message. Il peut être employé dans tous les styles de communication : synchrones ou asynchrones, point à point ou multi-point. SOAP utilise principalement les deux standards HTTP comme protocole de transport des messages et XML pour structurer les requêtes et les réponses [DEV_03].

Le protocole applicatif le plus utilisé pour transmettre les messages SOAP est HTTP, mais il est également possible d'utiliser les protocoles SMTP, FTP, etc [YKH_11].

II.5.2.2. Description des services Web WSDL

Le langage de description de services Web WSDL (Web Services Description Language) est également une grammaire de XML, développée par Microsoft et IBM sous les égides du W3C. Il permet de décrire de façon précise les services Web en incluant des détails tels que format du message prévu, le protocole de communication employé et l'adresse d'accès au service Web [AME_07]. Il permet la représentation d'un service Web de manière plus abstraite pour la réutilisation [GHA_11].

II.5.2.3. Découverte des services Web

UDDI (Universal Description, Discovery and Integration) est une norme d'annuaire de services Web appelée via le protocole SOAP [UDD_00]. Permettant de publier les interfaces des services Web. Il est possible d'effectuer des recherches en fonction de la description commerciale et technique d'un service. Il peut être privé (utilisé uniquement dans l'entreprise) ou public (ouvert à l'ensemble de l'Internet).

Les informations présentes dans le registre sont réparties en trois catégories :

- ✓ **pages blanches:** listent les entreprises ainsi que des informations associées à ces dernières.
- ✓ **pages jaunes :** recensent les services Web de chacune des entreprises sous le standard WSDL.
- ✓ **pages vertes :** fournissent des informations techniques précises sur les services fournis.

II.6. Les services Web REST

II.6.1. C'est quoi REST

REST est l'acronyme de Représentationnel State Transfer et a été décrit par Roy Thomas Fielding dans sa thèse « Architectural Styles and the Design of Network-based Software Architectures » en 2000.

REST n'est pas un protocole ou un format mais un style d'architecture. Il permet de construire une application pour les systèmes distribués comme le Web bien que REST ne soit pas un standard, il utilise des standards. En particulier :

- URI comme syntaxe universelle pour adresser les ressources.
- HTTP un protocole sans état (stateless).
- Des liens hypermédia dans des documents (X)HTML et XML pour représenter à la fois le contenu des informations et la transition entre états de l'application.
- Les types MIME comme text/xml, text/html, image/jpeg, application/pdf, video/mpeg pour la représentation des ressources.

II.6.2. Les concepts de REST

Dans le style d'architecture REST, toute information est une ressource et chacune d'elles est désignée par une URI généralement un lien sur le Web. La particularité de REST réside sur trois concepts : les ressources, leurs URI et leurs représentations.

a. Les ressources

Les ressources jouent un rôle central dans l'architecture REST. Une ressource est tout ce que peut désigner ou manipuler un client, toute information pouvant être référencée dans un lien hypertexte. Elle peut être stockée dans une base de données, un fichier, ect. On évite autant que possible d'exposer des concepts abstraits sous forme de ressources et l'on privilégie les objets simples [GON _10].

b. URI (Uniform Resource Identifier)

Une ressource Web est identifiée par une URI, qui est identifiant unique sur le système formé d'un nom et d'une adresse indiquant où il trouve la ressource. Les URIs devraient être aussi des descriptives que possible ne désigner qu'une seule ressource, Par contre, une ressource peut posséder une ou plusieurs URIs [GON _10].

Exemple :

`HTTP://www.ex.com/logiciel/versions/1.0.3.tar.gz` et `HTTP://www.ex.com/logiciel/versions/derniere.tar.gz` Sont deux URIs qui pointent sur la même ressource. La différence est le fait que la première désigne une version spécifique tandis que la seconde renvoie à la version qui est toujours la plus récente.

c. Représentation [GON _10]

On peut obtenir la représentation d'un objet sous forme de texte, de XML, de PDF ou sous un autre format. Un client trait toujours une ressource au travers de sa représentation ; la ressource elle-même reste toujours sur le serveur. La représentation contient toutes les informations utiles à propos de l'état d'une ressource.

II.6.3. Les principes de REST

Dans le style d'architecteur REST les ressources sont manipulées par un ensemble d'opérations simples et bien définies. L'architecture client-serveur de REST est conçue pour utiliser un protocole de communication sans état, le plus souvent HTTP. Ces principes encouragent la simplicité, la légèreté et l'efficacité des applications.

a. Adressabilité

Une application est dite adressable si elle présente des aspects intéressants de ses données en tant que ressources. Du fait que les ressources sont accessibles via les URIs, une application adressable présente une URI pour chaque information qu'elle peut être amenée à mettre à disposition.

Un exemple de ressources non adressable est par exemple les services Web utilisant SOAP. Les méthodes contenues dans le fichier SOAP ne sont pas adressables car nous ne pouvons pas les accéder via l'URI directement. Nous sommes obligés de passer par une requête POST sur un URI qui traite le message SOAP.

b. Absence d'état

L'absence d'état signifie que chaque requête HTTP se produit de manière totalement isolée. Lorsque le client réalise une requête HTTP, celle-ci comprend toutes les informations nécessaires au serveur pour y répondre. Le serveur ne s'appuie jamais sur des informations provenant de requêtes passées. Si une information est importante, le client doit la transmettre de nouveau, avec la requête en cours. Ainsi, chaque requête est entièrement décorrélée les unes des autres.

L'avantage est que nous pouvons nous permettre de répartir les tâches entre différents serveurs parce que chaque requête est autonome. C'est une solution très importante car cela permet de résoudre le problème en cas de panne d'un des serveurs. Deuxièmement, les applications sans état sont également fiables. Une requête envoyée par le client qui n'est pas arrivée peut être relancée une deuxième fois sans se soucier de savoir si sa session est dans un état anormal.

c. L'interface uniforme

Le Web est formé de ressources bien identifiées, reliées ensemble et auxquelles accéder au moyen de requêtes HTTP simples. Les requêtes principales de HTTP sont de type GET, POST. Ces types sont appelés verbes ou méthodes. Qui permettent aux composants de manipuler les ressources de manière simple et "intuitive". HTTP définit d'autres verbes plus rarement utilisés HEAD, PUT et DELETE.

- **GET**

L'utilisation de GET est "sûre" c'est à dire que l'état de la ressource ne doit pas être modifié par un GET. Ceci autorise les liens, la mise en cache, les favoris. Il faut donc utiliser GET pour des opérations qui ressemblent à des questions ou à des lectures de l'état de la ressource.

- **POST**

La sémantique de la méthode « POST » est le transfert d'informations du client vers le serveur. Si cette information est publiée sur le serveur et accessible au moyen d'une URI, il faut utiliser POST quand la demande ressemble à une commande, ou quand l'état de la ressource est modifié.

Le protocole HTTP comporte d'autres commandes moins souvent utilisées. HEAD qui est un GET qui ne renvoie que les métadonnées, pas les données. PUT et DELETE pour modifier et supprimer une ressource.

II.7. Les APIs Web

Une API (Application Programming Interface) 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. **[DIG_07]**

Il existe plusieurs APIs proposant parfois les mêmes services celles de : Google Maps, Yahoo Maps, Via Michelin Maps, Drive API.... etc. D'autres API proposent d'afficher la météo, ou encore d'avoir le module de recherche de livres d'Amazon.

La plupart des APIs sont offertes gratuitement à condition de respecter leurs conditions d'utilisation. Même les producteurs de services qui offrent des APIs payantes, offrent des APIs compatibles gratuites, mais avec moins de fonctionnalités que les APIs payantes.

II.8. Conclusion

Nous avons vu dans ce chapitre l'architecture orientée service SOA, les services web classique et les services Web REST, les APIs Web, etc. On a pu voir qu'il existe différentes APIs Web proposant parfois les mêmes services (comme les API Google Maps, Yahoo Maps,).

Chapitre III

Analyse et Conception

III.1. Introduction

Après avoir présenté dans la partie précédente la généralisation sur les services Web et l'étude d'existant de la gestion du service de vote, nous entamons dans ce chapitre la phase de conception de notre module d'affectation, qui va combiner un service cartographique externe (Google Maps).

Pour concevoir notre application nous avons choisi un processus spécifique à la modélisation des applications Web, basé sur le langage de modélisation UML dont l'objectif est de fournir une modélisation formelle, précise, à jour, mais néanmoins flexible pour mieux réaliser notre application.

III.2. Architecture de l'application

III.2.1. Les principes de l'API

Une API cartographique étant une bibliothèque de fonctions stockées sur le serveur qui s'exécutent dans le navigateur du client.

a. Les ressources et identifiants uniques

Google REST API permet d'accéder à des ressources et de les manipuler. Ces ressources se caractérisent toutes par un identifiant unique. Chaque domaine métier permet d'accéder à une ressource de type spécifique que les fonctions des autres domaines métiers peuvent parfois prendre en paramètre d'entrée au travers de cet identifiant unique. Le tableau suivant montre les domaines métier de Google REST API.

Tableau 6: Les domaines métier de Google REST API

Domaine métier	Ressource principale	Identifiant unique
Recherche de proximité	POI	couple (<i>dbID</i> , <i>poiID</i>)
Géocodage	Lieu	<i>locID</i>
Cartographie	Carte	<i>mapID</i>
Itinéraire	Itinéraire	<i>itiID</i>

b. Les paramètres

L'accès aux différents services de l'API REST de Google Maps se fait au travers d'une URL en positionnant les paramètres, soit dans le chemin d'accès, soit en paramètres de type GET suivant leur importance dans la définition ou dans la manipulation de la ressource accédée.

III.2.2. Le fonctionnement de système

L'objectif de notre travail est d'arriver à utiliser l'API de service Web de géolocalisation de Google Maps offert à travers le Web. Afin de créer un module d'affectation pour l'application eVOTE, l'architecture de ce système est une architecture orientée service (SOA Service Oriented Architecture) utilisant des services web (REST). L'API Maps dynamique permet de générer des cartes qui seront intégrées dans des pages web à l'aide de code JavaScript.

Et le schéma suivant montre le fonctionnement général du système prévu :

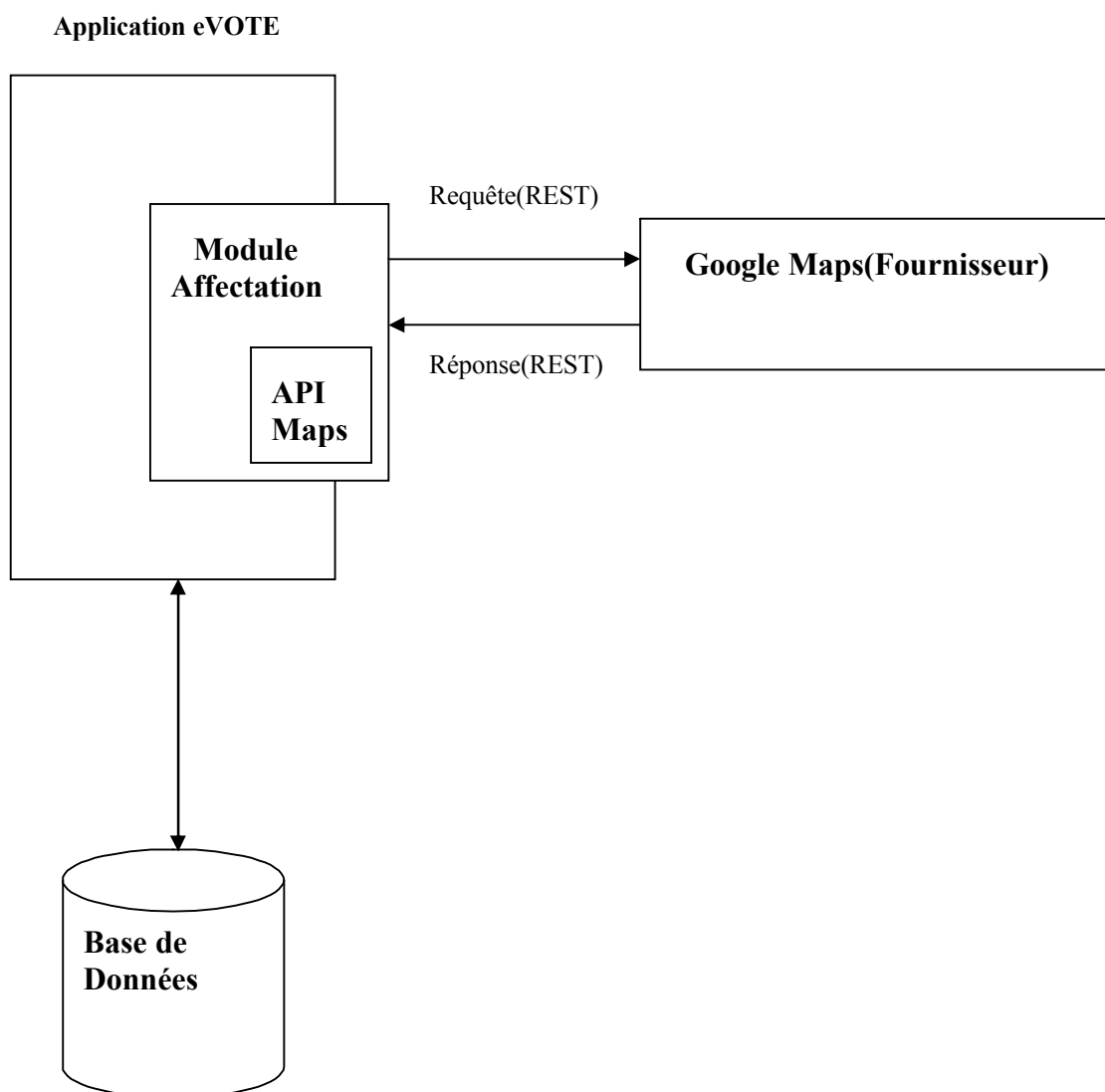


Figure 2: Architecture générale du module d'affectation

Google Maps fournit plusieurs services géo et sont utilisés par l'API de Google Maps tel que : un géocodeur et un service de calcul d'itinéraire, ...etc. On utilise ces services offerts par Google Maps pour :

- Localiser les positions des votants et des centres de vote et les citer sur la carte de Google Maps, et enregistrer leurs coordonnées et l'adresse postale sur la base de données.
- Afficher les différents trajets avec des différentes couleurs entre le votant et le centre des votes dans le cas d'une affectation par individu (ou groupe) ou entre le centre de la cité et les centres de vote dans le cas d'une affectation par cité (les deux types d'affectation sont expliqués dans le paragraphe de la cinquième DCU (**Diagramme de cas d'utilisation**) de la section III.4.1.2.2).
- Afficher la distance et la durée de chaque trajet que ce soit un trajet parcouru par un moyen de transport ou à pied, cet affichage permettra à l'officier du service de vote (OSV) de choisir le trajet le plus adéquat (en appliquant l'algorithme de plus court chemin pour afficher le trajet le plus adéquat).
- Afficher les détails d'un trajet que ce soit un trajet parcouru par un moyen de transport ou à pied.

III.3. Présentation de la démarche utilisée

III.3.1. La modélisation du système

Pour la conception de toutes les parties composant le système nous avons opté pour l'approche Orientée-Objet en utilisant le langage de modélisation UML « Unified Modeling Language », en français " langage de modélisation objet unifié " (**ANNEXE**). Ce choix a été alimenté par différentes raisons, parmi lesquelles :

- ✓ La possibilité de découper le système en sous-systèmes ce qui diminue la complexité.
- ✓ Un système conçu avec UML à une structure modulaire ce qui offre la possibilité de faire des modifications sur une partie du système sans porter atteinte au fonctionnement général de ce dernier.
- ✓ Sa notation graphique permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évaluation de solutions.
- ✓ L'aspect formel de sa notation limite les ambiguïtés et les incompréhensions.

- ✓ L'aspect formel de sa notation limite les ambiguïtés et les incompréhensions.
- ✓ Son indépendance par rapport aux langages de programmation, aux domaines d'application et aux processus, en font un langage universel.

Donc, la force d'UML réside dans sa façon de modéliser de manière claire et précise la structure (vue statique) et le comportement (vue dynamique) d'un système indépendamment de toute méthode ou de tout langage de programmation.

III.3.2. Méthodologie de conception

UML est un langage et non pas une méthode car il ne donne pas une démarche particulière de travail. Pour cette raison, il faut lui associer une méthode pour pouvoir l'utiliser. Pour la conception du système nous avons choisi de suivre la méthodologie de conception UP /XP.

III.3.3. Le processus UP/XP [ROQ_06]

Le processus que l'on va suivre pour le développement de notre application se situe à mi-chemin entre UP (Unified Process) (**les détails de ce processus est expliqué dans l'ANNEXE**), un cadre général très complet de processus de développement, et XP (eXtreme Programming) (**voir ANNEXE pour plus de détail**), une approche minimaliste centrée sur le code.

Le processus suivi (**UP/XP**) s'appuie sur UML tout au long du cycle de développement, car les différents diagrammes de ce dernier permettent de part leurs facilités et clartés, de bien modéliser l'application à chaque étape. Ce processus a été proposé par l'auteur du livre 'UML2 Modéliser une application Web', dans le cadre du développement des applications Web.

III.3.4. Caractéristiques du processus suivi

Le processus que nous allons suivre dans la modélisation de notre application est :

- Conduit par les cas d'utilisation, comme UP.
- Relativement léger et restreint, comme XP, mais sans négliger les activités de modélisation en analyse et conception.
- Fondé sur l'utilisation d'un sous-ensemble nécessaire et suffisant du langage UML.

III.4. Les modèles du processus UP/XP

Un modèle est une abstraction du système. Tous les modèles sont liés ensemble, ils représentent le système comme un tout. Ils utilisent UML. Le tableau ci-dessous montre les modèles du processus UP/XP.

Tableau 7: Les modèles du processus UP/XP

Modèle des cas d'utilisation	Expose les cas d'utilisation et leurs relations avec les utilisateurs.
Modèle de conception préliminaire	Détaille les cas d'utilisations et procède à une première répartition du système entre divers projets. Définit la structure statique du système sous forme de sous système, classes et interfaces.
Modèle de navigation	Détaille la navigation à l'intérieur du site Web.
Modèle vue contrôleur	Sépare la partie de programme et la l'interface utilisateur.

Le modèle des cas d'utilisation : Nous présenterons dans le modèle des cas d'utilisation : l'identification des acteurs, les cas d'utilisation.

Le modèle de conception préliminaire : Nous allons présenter les diagrammes de collaboration, de séquence, afin d'identifier les classes de conception nécessaires à la réalisation des cas d'utilisation.

Le modèle de navigation : Pour que la modélisation soit complète, il nous reste à détailler la navigation à l'intérieur du site Web, Ces diagrammes s'appellent des diagrammes de navigation. Nous nous intéressons aux deux cas d'utilisation majeurs: Valider les informations de la création d'un compte, affecter ou réaffecter les votants au centre de vote.

Le modèle vue contrôleur (MVC) : Le modèle vue contrôleur (MVC) est un moyen de décrire la séparation appropriée entre la partie d'un programme qui interagit avec l'utilisateur et celle qui s'occupe de la charge lourde.

III.4.1. Modèle des cas d'utilisation

Acteur et cas d'utilisation sont les concepts UML fondamentaux pour la spécification des exigences. Nous allons les identifier à partir de l'expression initiale des besoins de notre étude de cas. Ensuite, on va structurer, relier, et classer ces cas d'utilisation ainsi que leurs représentations graphiques UML associées.

Puisque notre module d'affectation est intégré dans l'application eVOTE donc on va représenter le cas d'utilisations générales et les tâches de chaque acteur de l'application eVOTE ensuite on va représenter les cas d'utilisations détaillés et les acteurs de notre module d'affectation.

III.4.1.1. Les cas d'utilisation de l'application eVOTE

Ce système possède deux types d'acteurs :

✓ **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 définis 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.
- Officier d'état civil valideur (OECV).
- Officier du service de vote (OSV).
- Chef du service de vote.
- Citoyen.
- Président de l'APC.
- Juge ou procureur.

✓ **Anonyme** : c'est une personne qui accède au système sans aucune authentification.

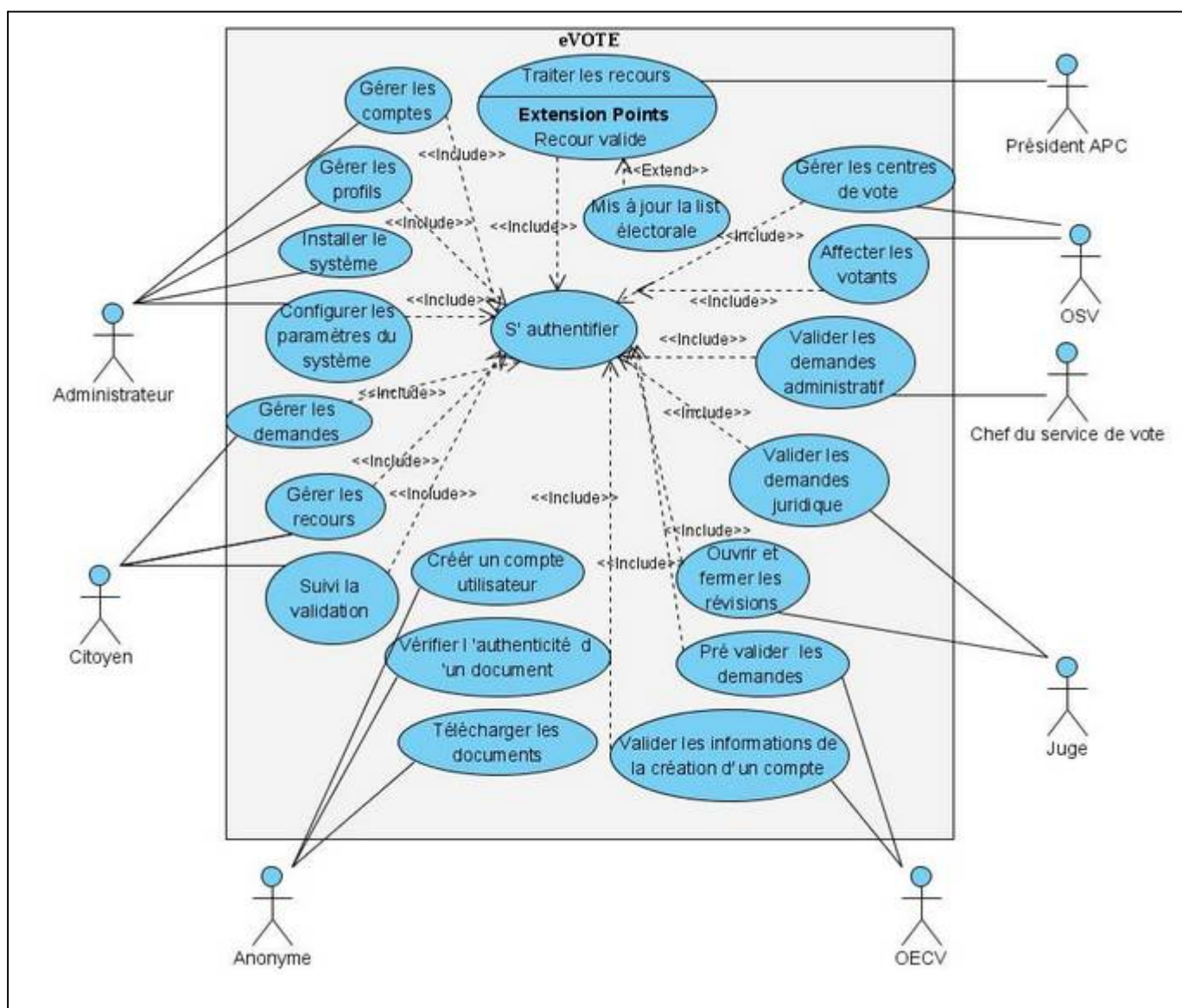


Figure 3: Diagramme de cas d'utilisation générale pour l'application eVOTE

Dans ce qui suit, nous définissons les tâches que peut réaliser chaque acteur.

➤ Administrateur du système

L'administrateur effectue les tâches suivantes

- Gestion des profils
 - Créer un profil.
 - Modifier et supprimer un profil.
- Gestion des comptes
 - Créer un compte utilisateur
 - Activer un compte

- Modifier et supprimer un compte
- Définir les droits d'accès.
- Paramétrage du système : l'administrateur peut définir des paramètres du système comme :
 - Préciser la méthode d'activation d'un compte (soit le compte sera activé automatiquement ou bien il sera activé par l'administrateur du système) .
 - Modifier l'année de découpage administratif.
- **Officier d'état civil validateur (OECV)**

Chaque information enregistrée dans le système doit être validée par l'officier d'état civil validateur.
- **Juge**

Le juge est le responsable de la commission. Pour cette raison il a la possibilité de :

 - Ouvrir et fermer la révision, que ce soit une révision annuelle ou exceptionnelle.
 - Valider les demandes acceptées par l'officier d'état civil validateur.
- **Chef du service de vote**

Cet acteur a le droit de valider les demandes acceptées par l'officier d'état civil validateur.
- **Officier du service de vote (OSV)**

Cet acteur a le droit d'affecter les votants au centre de vote le plus proche et de gérer les centres de vote.
- **Président d'APC**

Le rôle du président d'APC est de traiter les recours et prendre la responsabilité d'inscrire ou de radier les citoyens dans la liste électorale qui ont des recours valides.
- **Citoyen**

Le citoyen effectue les tâches suivantes:

 - Gestion des demandes (inscription, radiation, modification).
 - Suivi de l'état de validation d'une demande.
 - Gestion des recours.

Remarque :

Le chef du service de vote peut refuser ou accepter les dossiers selon des causes administratives, par contre le juge peut refuser ou accepter les dossiers selon des causes juridiques.

III.4.1.2. Les cas d'utilisation de notre module d'affectation**III.4.1.2.1. Identification des acteurs**

Les acteurs humains du système eVOTE qui influent sur le module d'affectation sont :

- Administrateur du système.
- Officier d'état civil validateur(OECV).
- Officier du service du vote(OSV).
- Citoyen.
- Anonyme.

On a aussi un acteur non humain :

- Le service de cartographie de Google Maps, qui permet de visualiser la position de tout votant inscrit et de l'affecter au centre le plus proche.

III.4.1.2.2. Diagramme de cas d'utilisation détaillé

Pour chaque acteur identifié précédemment, il convient de rechercher les différentes intentions « métier » selon lesquelles il utilise le module d'affectation.

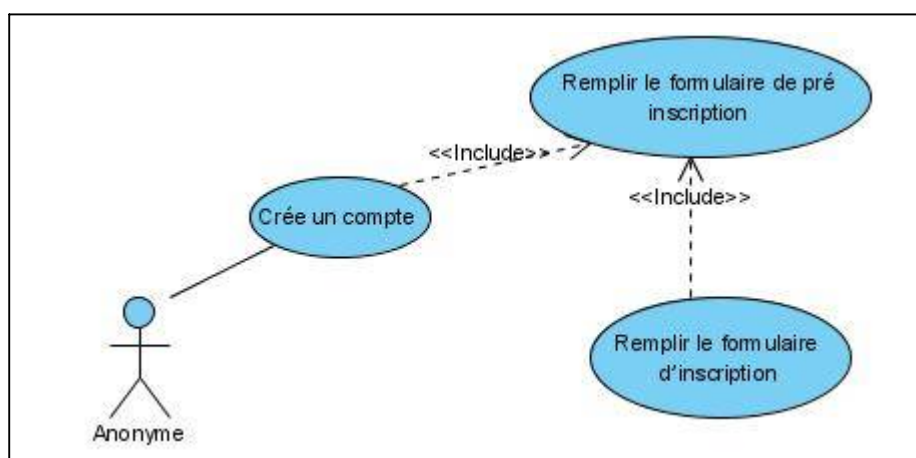
1. DCU (Diagramme de cas d'utilisation) de création d'un compte

Figure 4: DCU de création d'un compte

Dans le tableau 8 on définit la description du DCU de création d'un compte.

Tableau 8:Description du DCU de création d'un compte

Cas d'utilisation	Acteurs	Description
Crée un compte	Anonyme\ Administrateur	Chaque anonyme a le droit de faire une création d'un compte, qui sera validée ou refusée par OECV.
Remplir le formulaire de pré inscription	Anonyme\ Administrateur	L'anonyme doit remplir le formulaire de pré inscription pour avoir une réponse sur l'état de compte.
Remplir le formulaire d'inscription	Anonyme\ Administrateur	L'anonyme doit remplir le formulaire d'inscription pour avoir une réponse d'inscription.

Le formulaire de pré inscription contient: wilaya et daïra et commune de naissance, Date de Naissance, nom utilisateur, N° de séquence et le formulaire d'inscription contient: nom, prénom, date naissance, nom père,.....

2. DCU de validation (refus) les informations de la création d'un compte

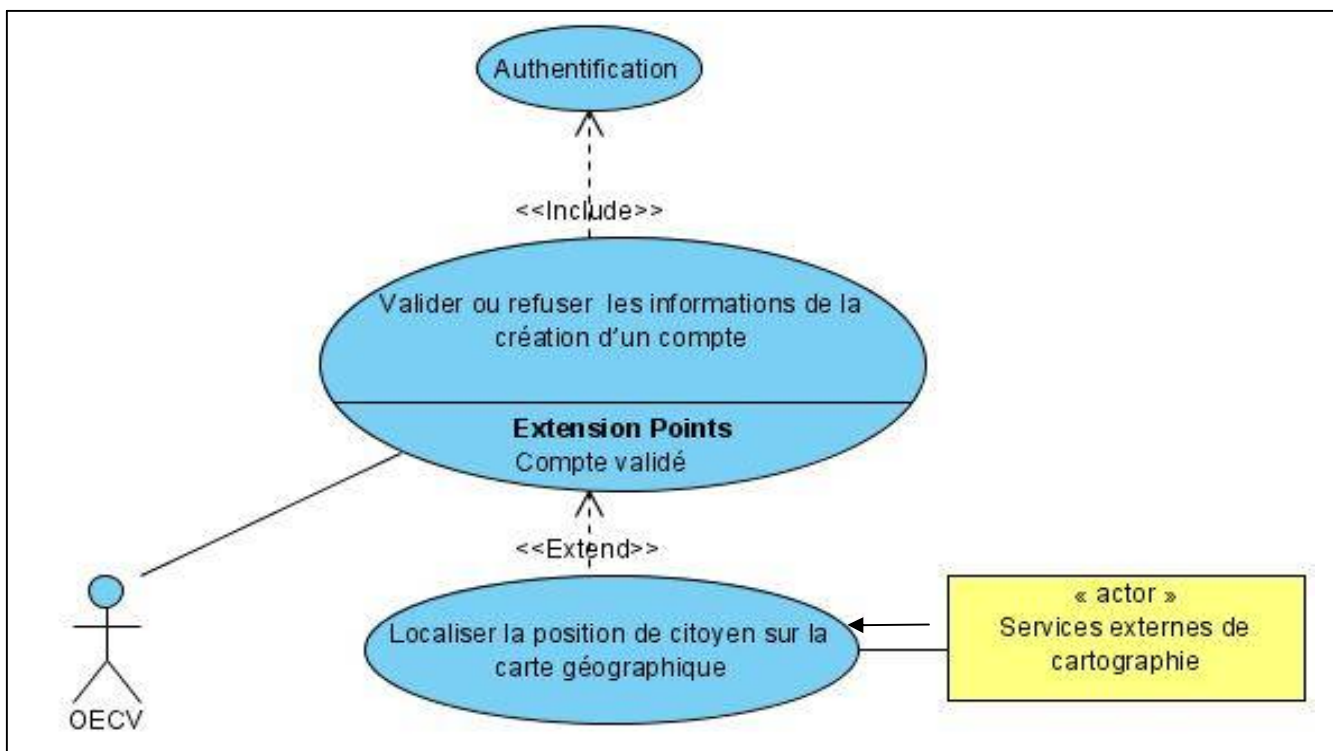


Figure 5: DCU de validation (refus) les informations de la création d'un compte

Le tableau 9 montre la description du DCU de validation (refus) les informations de la création d'un compte.

Tableau 9: Description du DCU de validation (refus) les informations de la création d'un compte

Cas d'utilisation	Acteurs	Description
Authentification	OECV	L'authentification est obligatoire pour valider ou refuser les informations de la création d'un compte.
Valider ou refuser les informations de la création d'un compte	OECV	L'OECV peut accepter ou refuser les informations de la création d'un compte après la vérification.
Localiser la position de citoyen sur la carte géographique	OECV	L'OECV a le droit de montrer la position de la pointeur sur la carte géographique correspond au lieu où réside le citoyen.

- L'OECV valide ou bien refuse les comptes des utilisateurs, il s'agit de l'authenticité des informations de ces comptes en contactant le service d'état civil.
- Si les informations sont valides, l'OECV peut mettre ensuite la position de pointeur sur la carte géographique correspond au lieu où réside le citoyen.

3. DCU de l'ajout d'un centre de vote

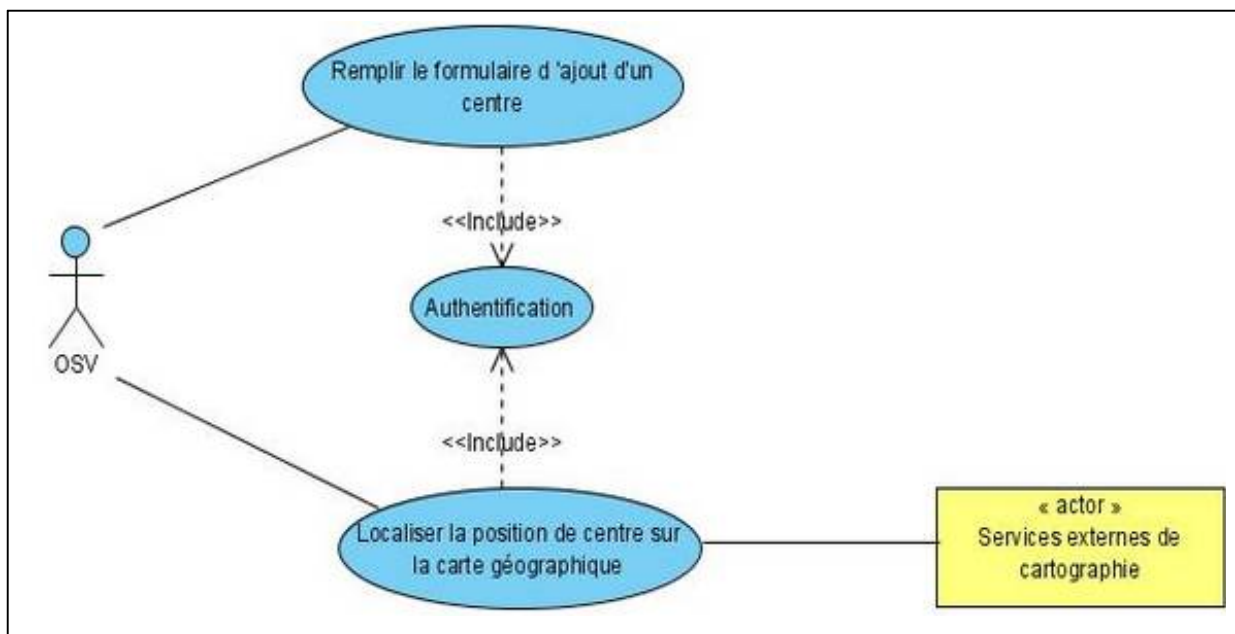


Figure 6: DCU de l'ajout d'un centre de vote

Le tableau 10 montre la description du DCU de l'ajout d'un centre de vote.

Tableau 10: Description du DCU de l'ajout d'un centre de vote

Cas d'utilisation	Acteurs	Description
Authentification	OSV	L'authentification est obligatoire pour faire l'ajout d'un centre.
Remplir le formulaire d'ajout un centre	OSV	L'OSV doit remplir le formulaire d'ajout d'un centre de vote.
Localiser la position de centre sur la carte géographique	OSV	L'OSV doit montrer le lieu du centre en posant le pointeur au dessus.

Le formulaire d'ajout d'un centre contient une carte géographique où l'OSV doit poser son pointeur sur cette carte correspond au lieu du centre de vote.

4. DCU de l'activation d'un compte

L'administrateur peut activer les nouveaux comptes des citoyens après leurs validations par l'OECV.

5. DCU de l'affectation ou réaffectation des votants au centre de vote

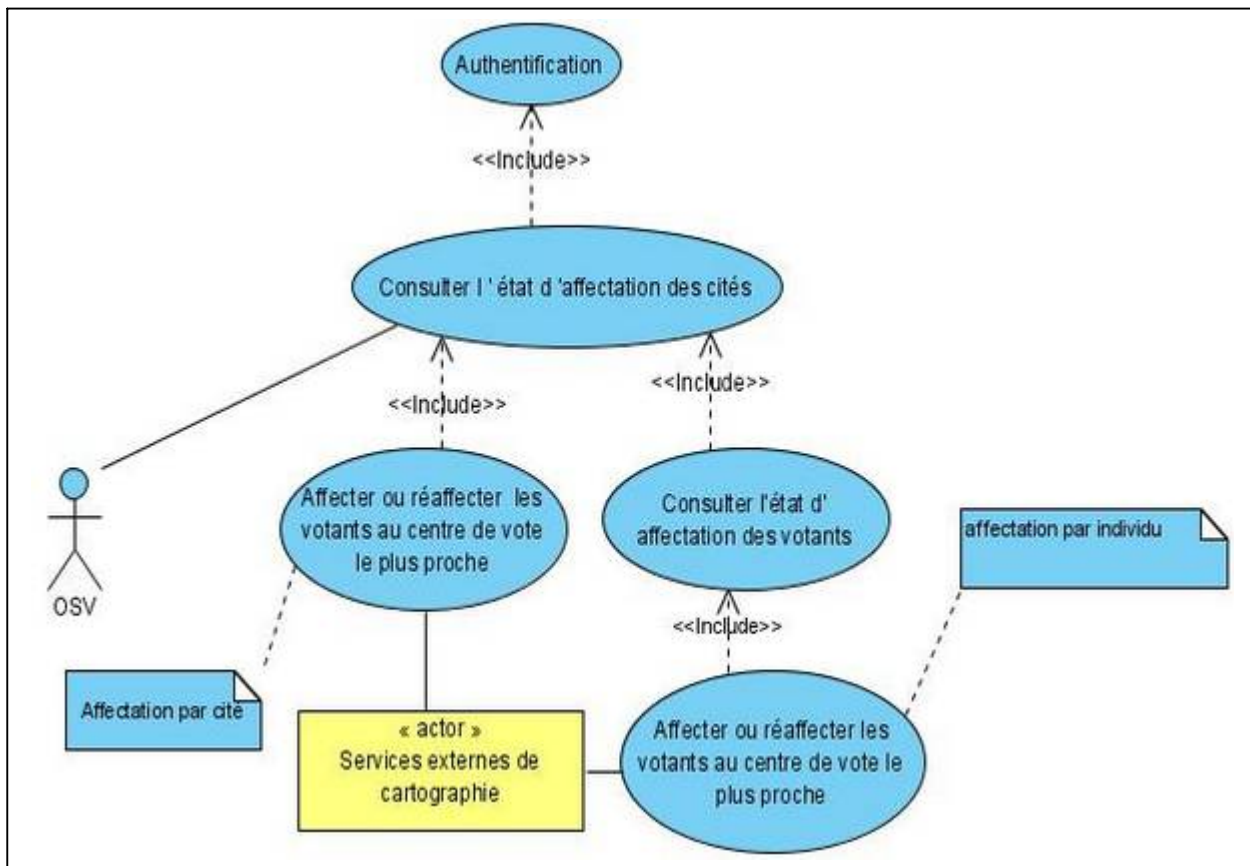


Figure 7: DCU d'affectation ou réaffectation les votants au centre de vote

Ci-dessous dans le tableau 11, la description du DCU d'affectation ou réaffectation des votants au centre de vote.

Tableau 11: Description du DCU d'affectation ou réaffectation des votants au centre de vote

Cas d'utilisation	Acteurs	Description
Authentification	OSV	L'authentification est obligatoire pour affecter ou réaffecter les votants au centre de vote.
Consulter l'état d'affectation des cités	OSV	L'OSV a le droit de consulter l'état d'affectation des cités (l'état d'affectation des cités est expliqué dans la deuxième remarque de la section III.4.2.3).
Affecter ou réaffecter les votants (par cités) au centre de vote le plus proche	OSV	L'OSV a le droit d'affecter ou réaffecter tous les votant qui appartient à une cités au même centre de vote le plus adéquat après voir le résultat d'affichage de tous les chemins entre le centre de la cité et les centres de votes.
Consulter l'état d'affectation des votants	OSV	L'OSV a le droit de consulter l'état d'affectation des votants (affecté ou non affecté).
Affecter ou réaffecter les votants (par individu (ou groupe)) au centre de vote le plus proche	OSV	L'OSV a le droit d'affecter ou réaffecter les votants au centre de vote le plus adéquat après voir le résultat d'affichage de tous les chemins entre l'adresse de votant et les centres de votes.

- L'affectation au centre de vote ce fait suivant les critères :
 - ✓ le citoyen doit faire une demande d'inscription dans la liste électorale.
 - ✓ .La demande doit être acceptée par l'OECV et la commission.
- L'OSV a le droit d'affecter ou réaffecter les votants au centre de vote le plus adéquat selon deux type :
 - ✓ Affectation par individu (ou groupe) : Dans ce cas l'affectation ce fait par individu d'une cité au centre de vote comme on peut affecter par groupe d'individus d'une même adresse au même centre de vote, ce choix dépend de l'OSV, c'est-à-dire que les votants qui appartiennent à la même cité ne seront pas forcément affectés au même centre.

- ✓ Affectation par cité : Dans ce cas l'affectation se fait par cité, c'est-à-dire que tous les votants qui appartiennent à la même cité sont affectés au même centre de vote le plus adéquat.
- Le résultat d'affichage de tous les chemins entre le centre de la cité (ou l'adresse de votant) et les centres de votes, nous donne les différents chemins avec différentes couleurs sur la carte de Google Maps et la durée et la distance de chaque chemin sous forme de tableau.

6. DCU de l'ajout d'une cité

DCU « Ajout d'une cité » est le même que DCU « Ajout d'un centre », la différence est que dans le DCU « Ajout d'une cité » l'OSV doit remplir le formulaire d'ajout d'une cité qui contient une carte géographique et en parallèle doit poser le pointeur sur cette carte qui correspond au centre de la cité.

III.4.2. Modèle de conception préliminaire

Le modèle de conception est un modèle objet décrivant la réalisation physique des cas d'utilisation en s'attachant aux effets des exigences fonctionnelles et des autres types de contraintes sur le système en question. En outre, il tient lieu de vision abstraite de l'implémentation du système et constitue par conséquent une entrée majeure pour les activités d'implémentation. Dans cette étape, nous allons modéliser les différents scénarios entre les acteurs et le système futur et puis identifier les classes de conception ainsi que les différentes tables constituant la base de données, nécessaires à la réalisation des cas d'utilisation.

III.4.2.1. Diagrammes de collaboration

Les différents diagrammes de collaboration pour chaque cas d'utilisation sont les suivants :

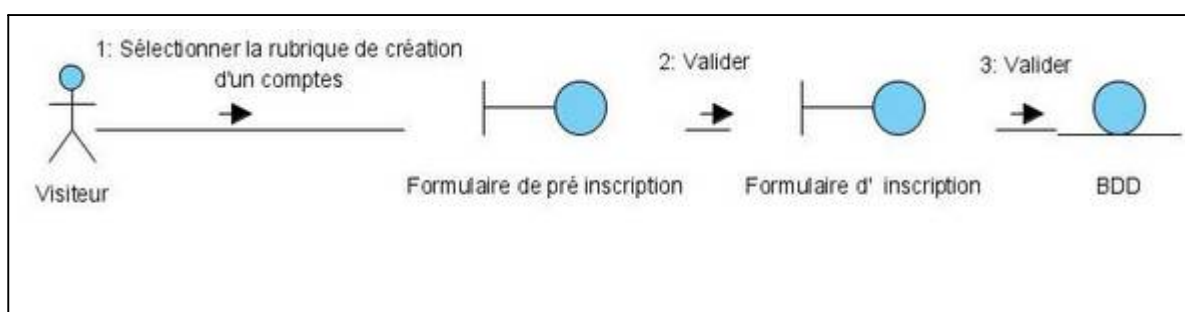


Figure 8: Diagramme de collaboration pour la création d'un compte

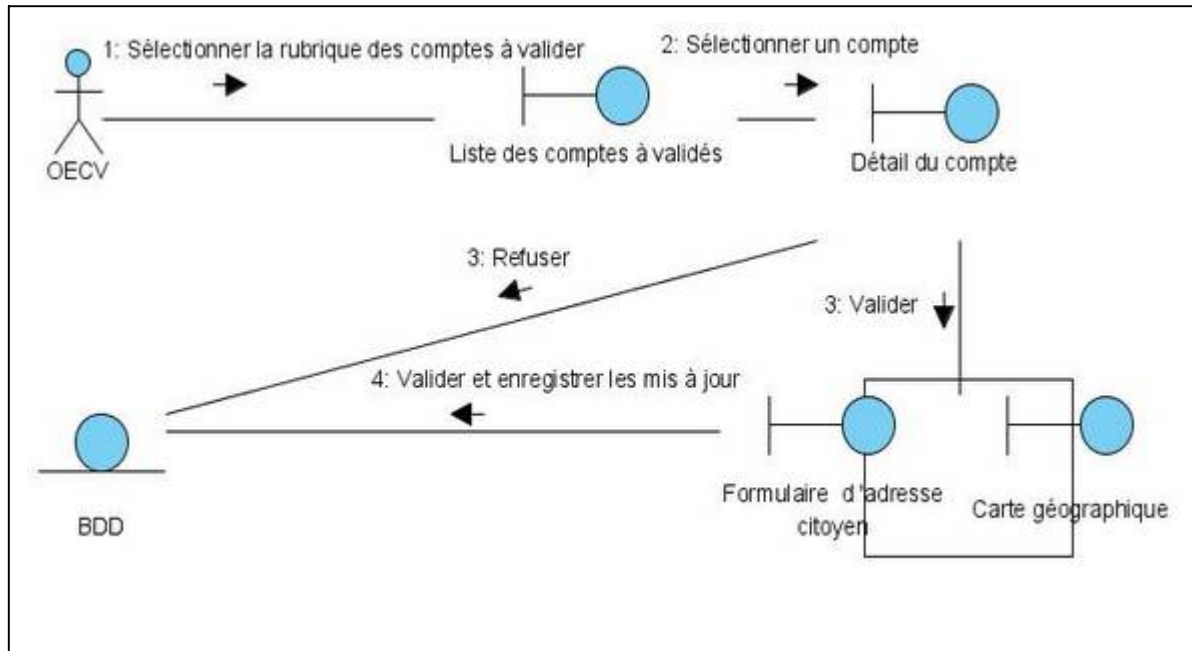


Figure 9: Diagramme de collaboration pour la validation des informations d'un compte

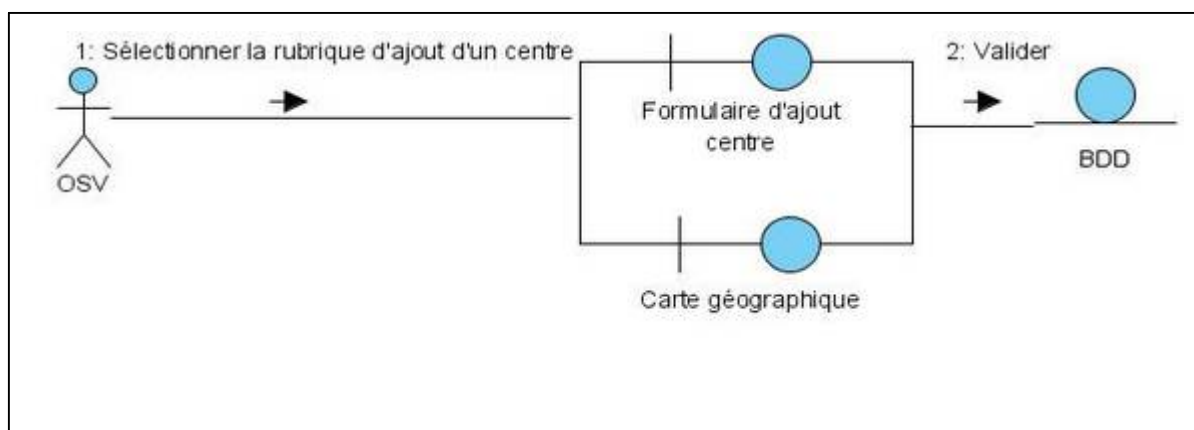


Figure 10: Diagramme de collaboration pour l'ajout d'un centre de vote

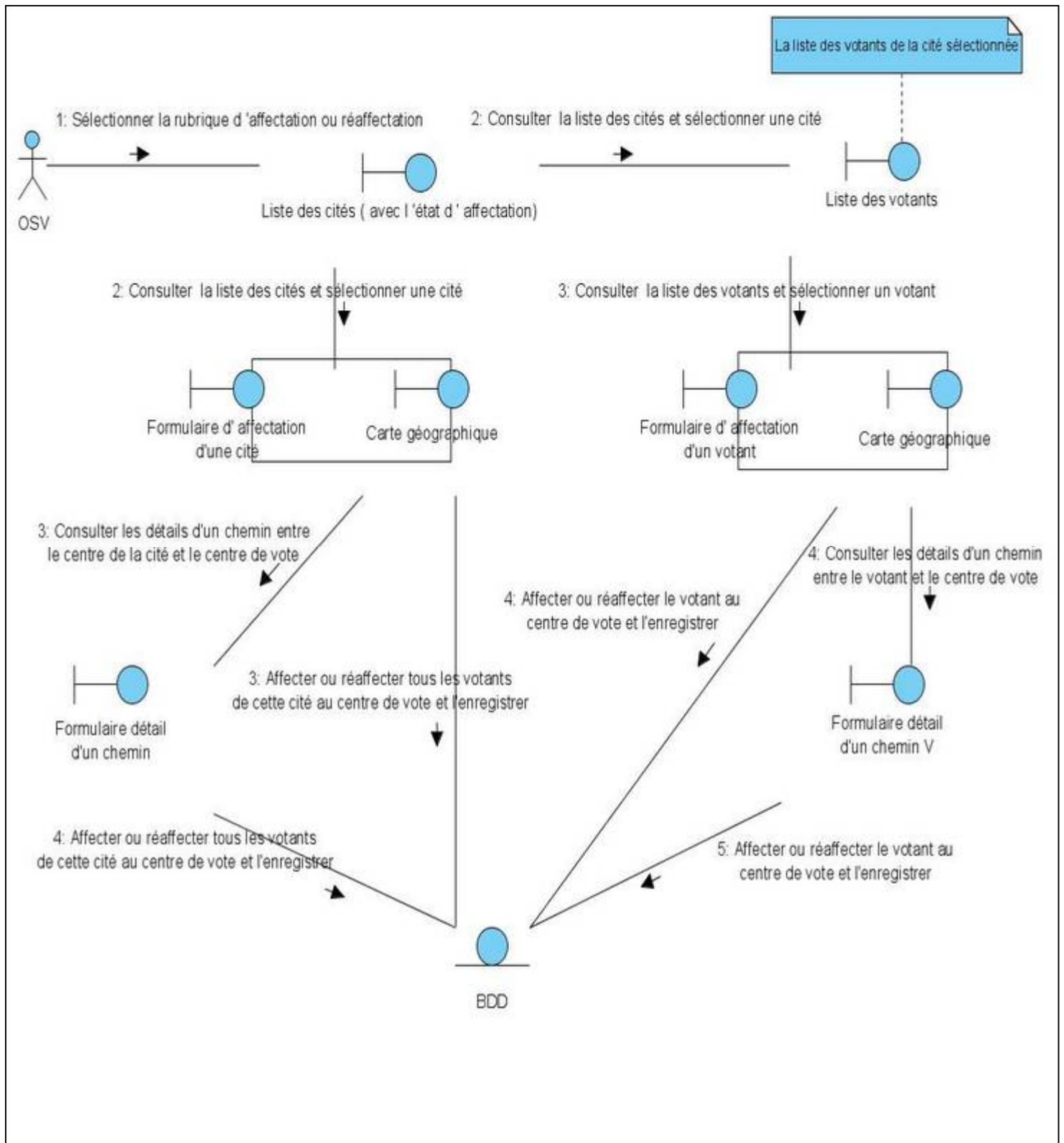


Figure 11: Diagramme de collaboration pour l'affectation ou réaffectation des votants au centre de votes

III.4.2.2. Diagrammes de séquence système

a). Création d'un compte

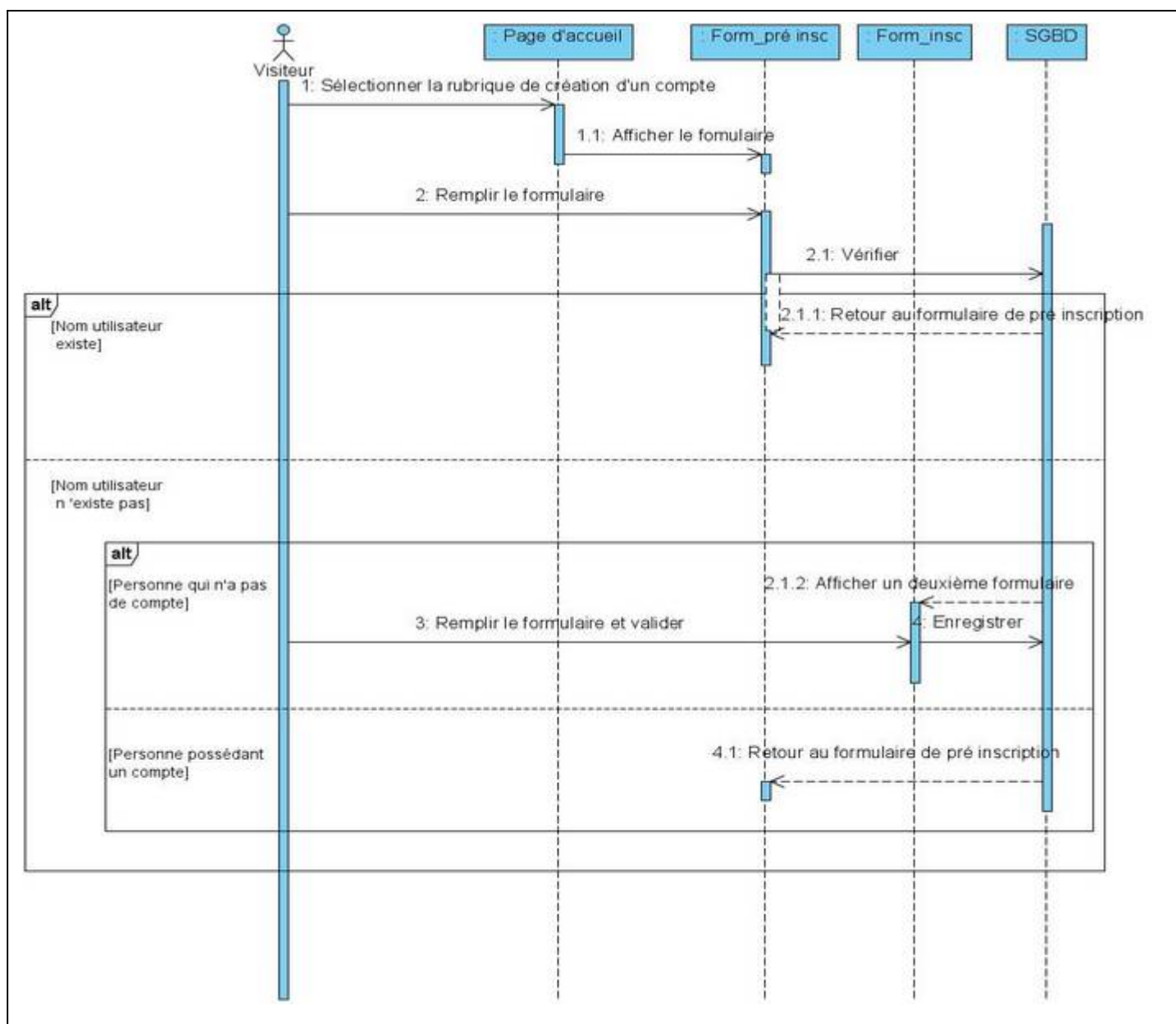


Figure 12: Diagramme de séquence pour la création d'un compte

Pour décrire ce scénario dans ce diagramme, nous allons faire intervenir les instances suivantes :

Interface	Signification
Form_pré insc	Formulaire de préinscription pour les nouveaux utilisateurs.
Form_insc	Formulaire d'inscription détaillé pour les nouveaux utilisateurs.

b) .Validation des comptes

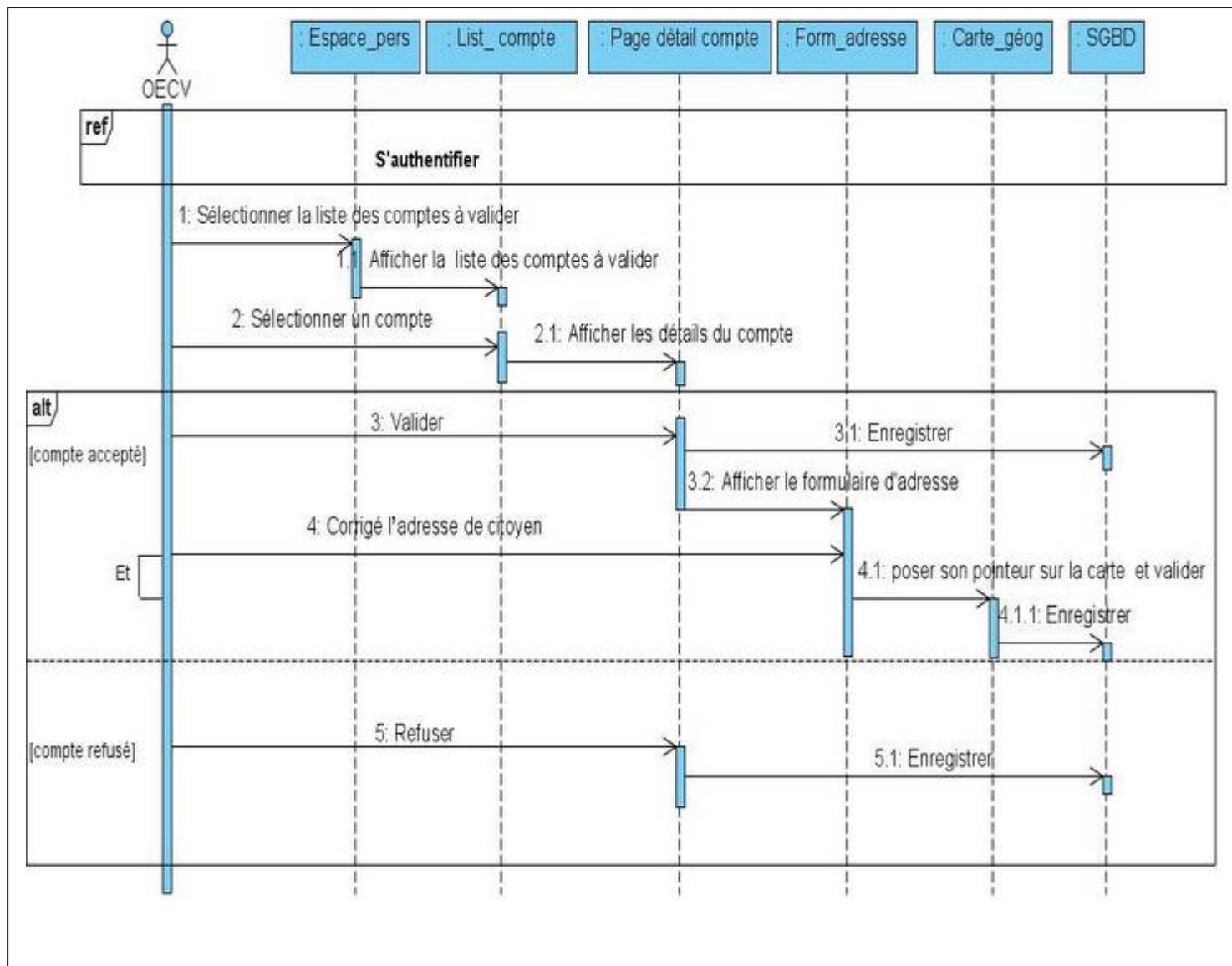


Figure 13: Diagramme de séquence pour la validation des comptes

Pour décrire ce scénario dans ce diagramme, nous allons faire intervenir les instances suivantes :

Interface	Signification
Espace_pers	Espace personnel d’officier d’état civil valideur.
List_compte	La liste des comptes à valider.
Page détail compte	Page détaillée pour un compte.
Form_adresse	Formulaire d’adresse qui contient l’adresse de citoyen et la carte géographique.
Carte_géog	la carte géographique.

c) .Ajout un centre de vote

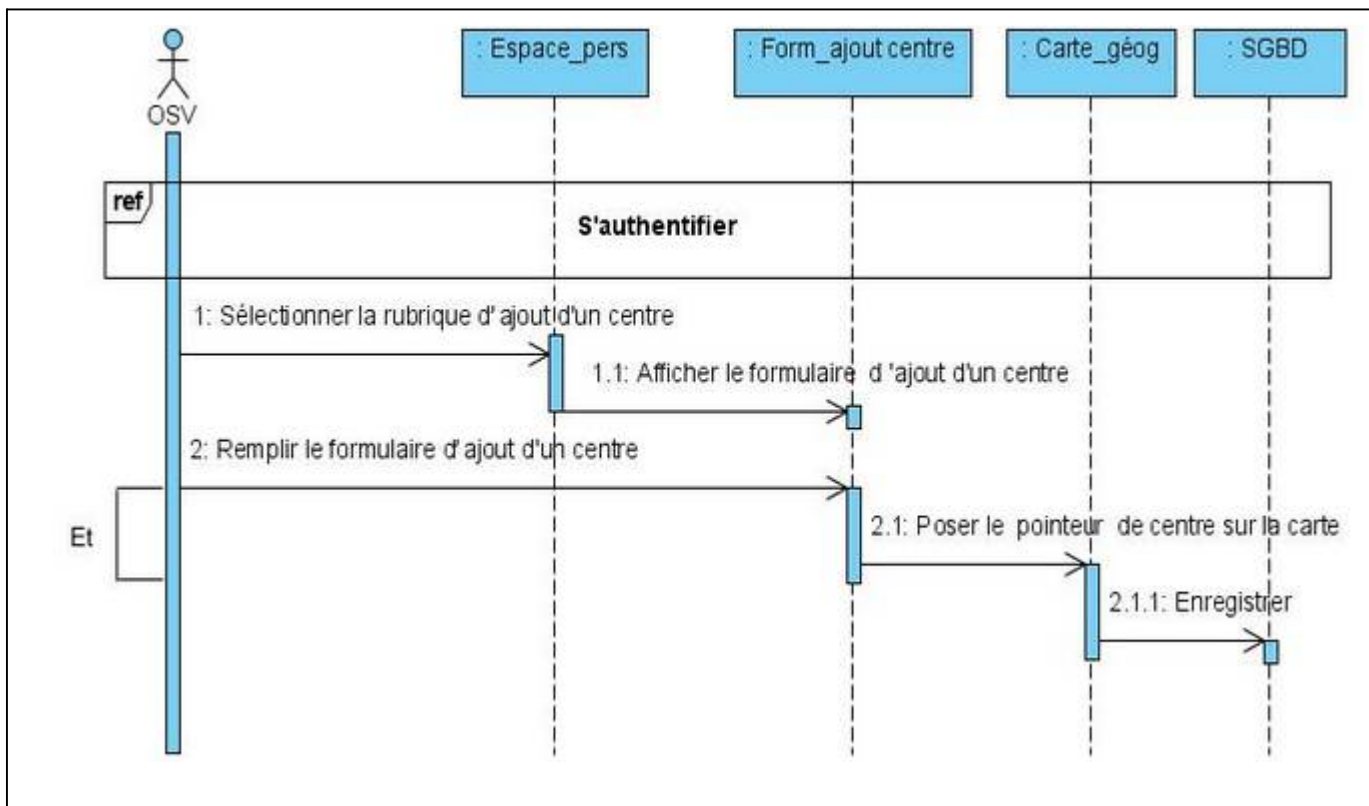


Figure 14: Diagramme de séquence pour l'ajout d'un centre de vote

Pour décrire ce scénario dans ce diagramme, nous allons faire intervenir les instances suivantes :

Interface	Signification
Espac_pers	Espace personnel d'officier du service de vote.
Form_ajout centre	Formulaire d'ajout d'un centre de vote qui contient la carte géographique.
Carte_géog	La carte géographique.

d). Affectation ou réaffectation des votants au centre de vote

1. Affectation ou réaffectation des votants au centre de vote (par individu)

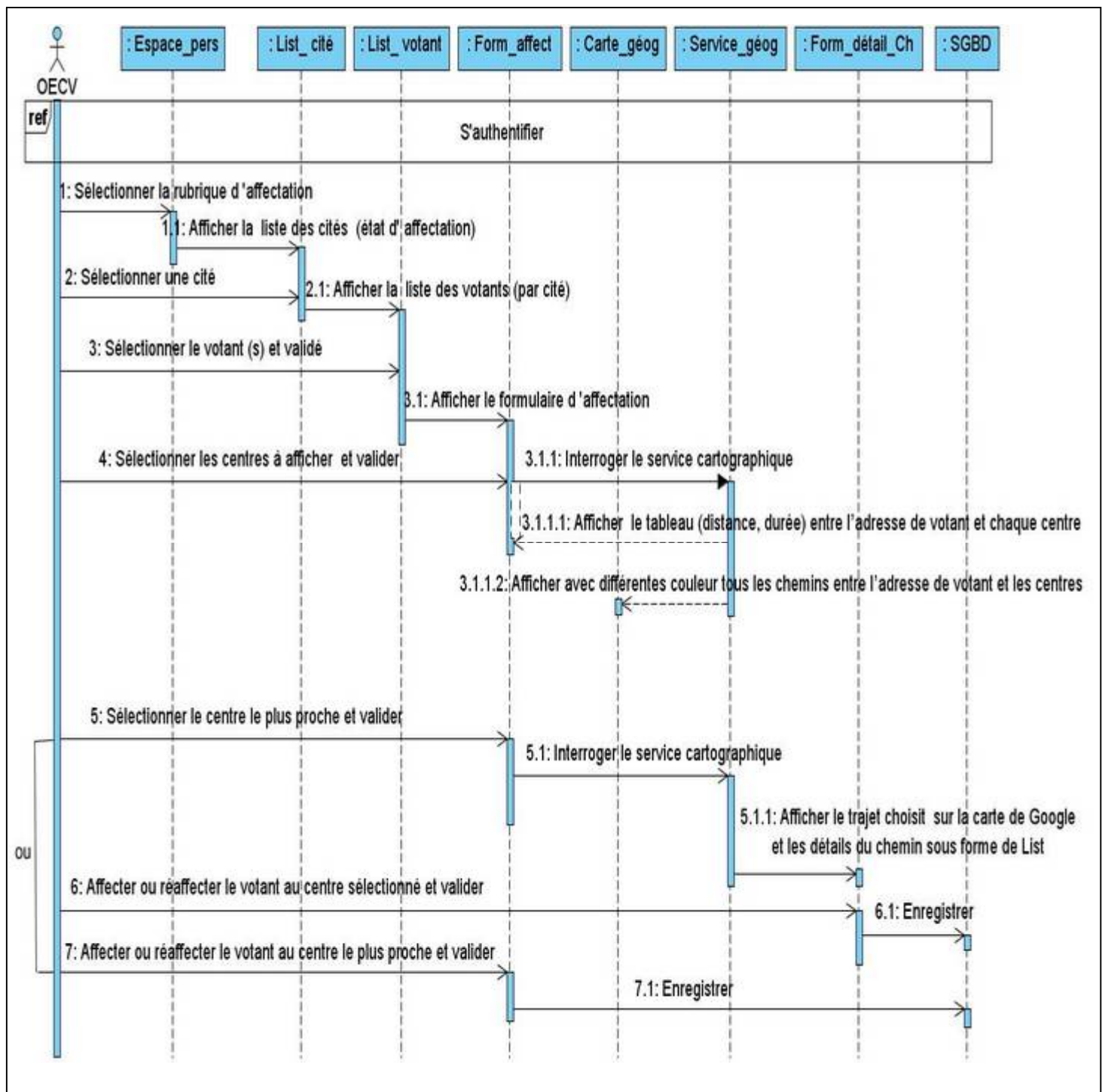


Figure 15: Diagramme de séquence pour l'affectation ou réaffectation des votants au centre de vote (par individu)

Pour décrire ce scénario dans ce diagramme, nous allons faire intervenir les instances suivantes :

Interface	Signification
Espace_pers	Espace personnel d'officier du service de vote.
List_cité	La liste des cités.
List_votant	La liste des votants à affectée ou réaffectées.
Form_affect	Formulaire d'affectation (contient : un tableau qui permet d'afficher la distance et la durée entre l'adresse du votant et les différents centres de vote, liste des centres (avec le pourcentage des votants à affecté de chaque centre), carte géographique).
Carte_géog	La carte géographique qui permet d'afficher tout les chemins entre l'adresse du votant et les centres de votes.
Service_géog	Service géographique (Google Maps).
Form_détail_Ch	Formulaire détails chemin contient une carte géographique ,cette formulaire permet l'affichage les détails du chemin sélectionné.

Remarque :

- ✓ Le formulaire d'affectation contient une carte géographique qui permet d'afficher tous les chemins entre l'adresse de votant et les centres de votes.
- ✓ L'OSV peut affecter ou réaffecter un votant ou plusieurs d'une même adresse au même centre de vote.

2. Affectation ou réaffectation des votants au centre de vote (par cité)

Le scénario «Affectation ou réaffectation des votants par cité au centre de vote » se déroule de la même manière que «Affectation ou réaffectation des votants par individu au centre de vote », la différence est que dans le scénario «Affectation ou réaffectation des votants par cité au centre de vote » l'OSV affecte ou réaffecte ou réaffecté tous les votants qui appartiennent à la même cité au même centre de vote le plus adéquat.

e) .Authentification

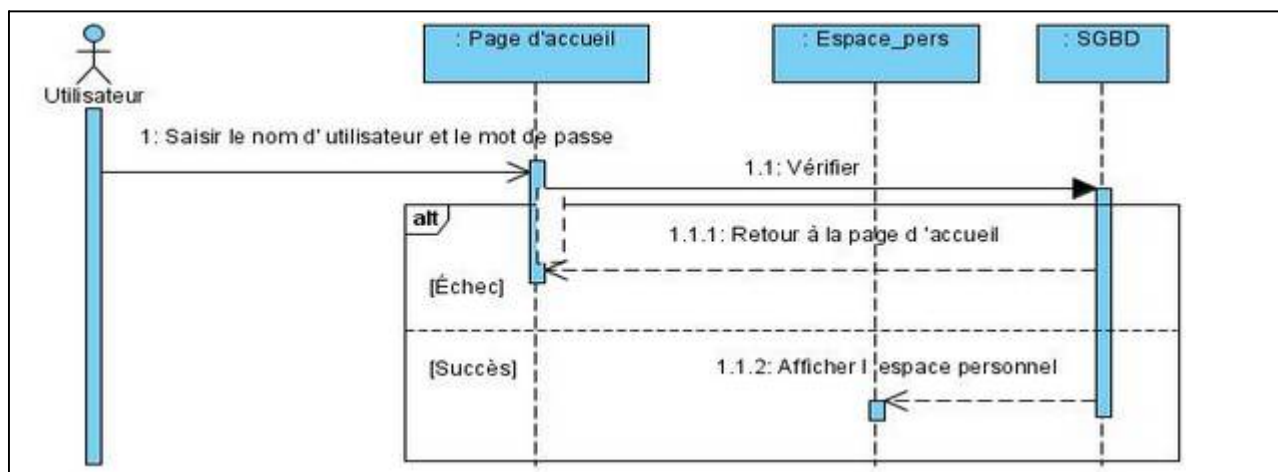


Figure 16: Diagramme de séquence pour l'authentification

Pour décrire ce scénario dans ce diagramme, nous allons faire intervenir les instances suivantes :

Interface	Signification
Espace_pers	Espace personnel d'utilisateur.

III.4.2.3. Diagramme de classe de conception

Dans cette étape, nous allons identifier les classes de conception et les relations entre ces classes pour l'application eVOTE et le module d'affectation.

1. Diagramme de classe de conception pour l'application eVOTE

Dans cette étape nous allons identifier les classes de conception nécessaire à la réalisation des cas d'utilisation de l'application eVOTE.

Remarque :

- La classe « Demande eVOTE »: le citoyen peut faire une demande de vote que ce soit une demande de radiation ou demande d'inscription ou demande de modification (type de demande).
- La classe « Demande eVOTE » possède une identifiant unique pour chaque demande de vote et date de demande et état de la demande, même pour la classe « Recours» qui possède un identifiant unique et date recours. La classe «Révision» possède une date début et date de fermeture de la révision.
- La date de recours est supérieure à date de fermeture de la révision.
- Ce diagramme de classe introduit un nouveau type de classeur possédant le stéréotype « énumération », ces classe de type « énumération » sont : « État Demande_E», « État Compte», « Sexe-P ».C'est-à-dire que l'attribut "Etat demande" de la classe « Demande eVOTE » permet d'exprimer l'état de validation de cette demande, le type de valeur de cet attribut est énuméré qui prend une valeur parmi l'ensemble des valeurs de la classe « État Demande_E ». Ces valeurs sont les suivant :
 - ✓ Pas encore: la demande n'est pas encore validé.
 - ✓ Validé1 : la demande est validée par L' OECV.
 - ✓ Validé2 : la demande est validée par le Juge.
 - ✓ Refusé : la demande est refusée.

- La même explication pour les deux autres attributs "Etat_compte" et "Sexe" comme suit :
 - ✓ L'attribut "Etat_compte" de la classe « Utilisateur » qui peut prendre une valeur parmi l'ensemble des valeurs de la classe « État Compte », c'est-à-dire soit Activé ou Désactivé.
 - ✓ L'attribut "Sexe" de la classe « Personne » qui peut prendre une valeur parmi l'ensemble des valeurs de la classe « Sexe-P », c'est-à-dire soit Homme ou Femme.
- [OECV] : présenté une contrainte sur la classe « Utilisateur » qui permet d'exprimer que le rôle de ce utilisateur appartient à la liste des rôles qui contient le rôle de l'officier d'état civil validateur, pAPC : exprimé le rôle de président d'APC.

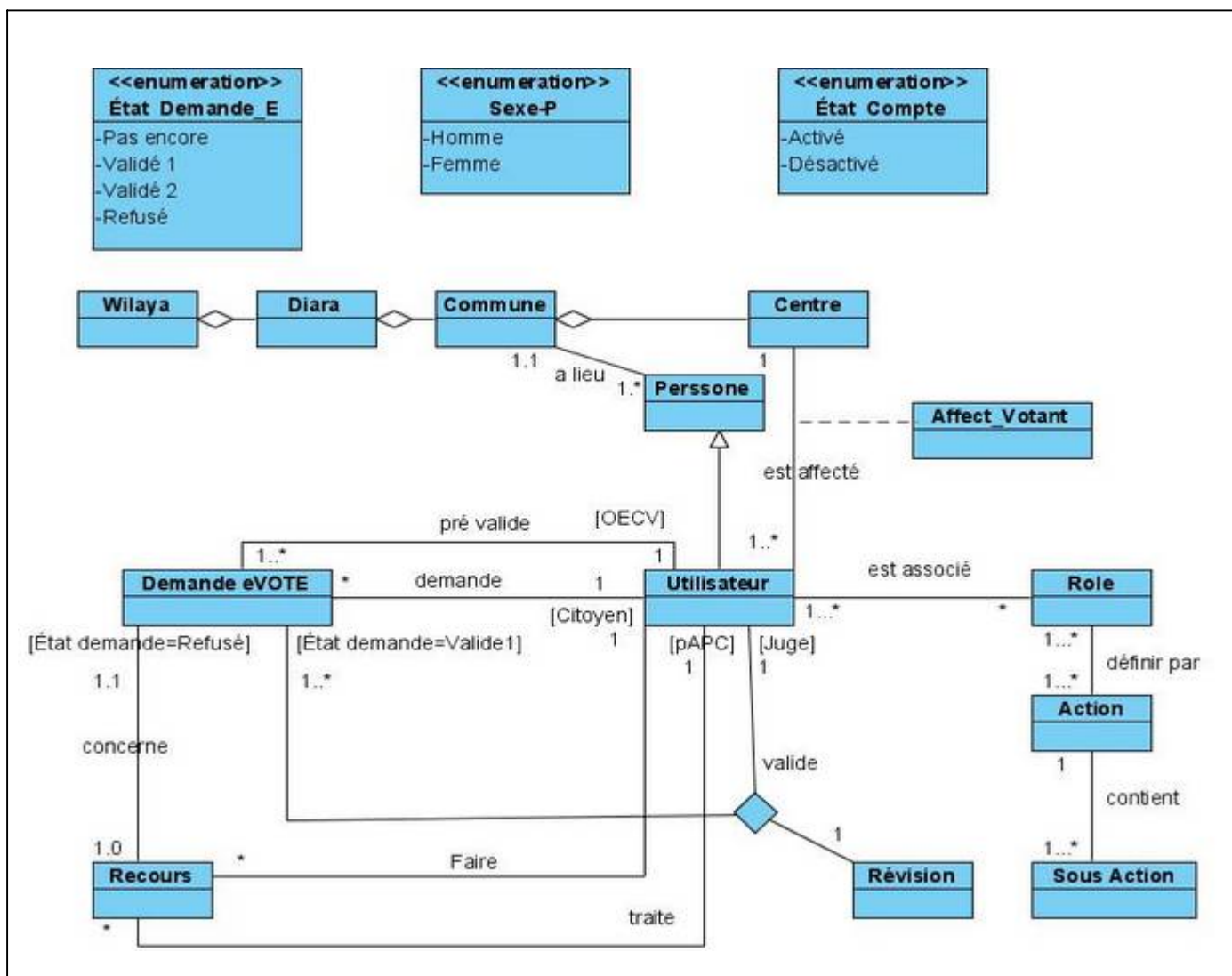


Figure 17: Diagramme des classes de l'application eVOTE

2. Diagramme de classe de conception pour le module d'affectation

Dans cette étape nous allons identifier les classes de conception nécessaire à la réalisation des cas d'utilisation de notre module d'affectation cela de manière détaillée et approfondie. Nous nous intéressant seulement à des classes qui sont des relations avec notre module d'affectation.

Remarque :

- ✓ Les deux champs la latitude et la longitude de la table « Adresse » sont automatiquement remplis quand l'OECV pose le pointeur correspond au lieu où réside le citoyen sur la carte lors de la création de son compte.
- ✓ Les deux champs la latitude et la longitude de la table « Cité » sont automatiquement remplis quand l'OSV pose le pointeur correspond au centre de la cité sur la carte lors de l'ajout d'une cité. Même principe pour le centre de vote.
- ✓ les deux attributs latitude et longitude de la classe « Adresse » présente le lieu de un ou plusieurs votants qui habite dans la même adresse.
- ✓ Ce diagramme de classe introduit un nouveau type de classeur possédant le stéréotype « énumération », ces classes de type « énumération » sont : « État Affect-Cité », « État Compte », « Sexe-P » (les deux dernières classe « énumérations » sont expliqué dans la remarque précédente).

C'est-à-dire que l'attribut calculable "Etat_affectation" de la classe « Cité » permet d'exprimer l'état d'affectation de la cité, le type de valeur de cet attribut est énuméré qui prend une valeur parmi l'ensemble des valeurs de la classe « État Affect-Cité ». Ces valeurs sont les suivants :

- La valeur 'Non affecté' signifie que les votants qui appartiennent à cette cité ne sont pas encore affectés.
- La valeur 'Affecté' signifie que les votants qui appartiennent à cette cité sont tous affectés au même centre.
- La valeur 'Affecté partie' signifie que les votants qui appartiennent à cette cité ne sont pas affectés au même centre (affectée par groupe).

- ✓ Le pourcentage des votants par rapport à la capacité du centre aide l’OSV pour bien organisé l’affectation des votants au centre de vote, si le pourcentage est 100% c'est-à-dire que le centre est rempli.
- ✓ La distribution des votants au bureau au niveau des centres de vote ce fait automatiquement et de façon équilibré pour ne pas surcharger certain bureau de vote, c'est pour ça la classe bureau n'existe pas.

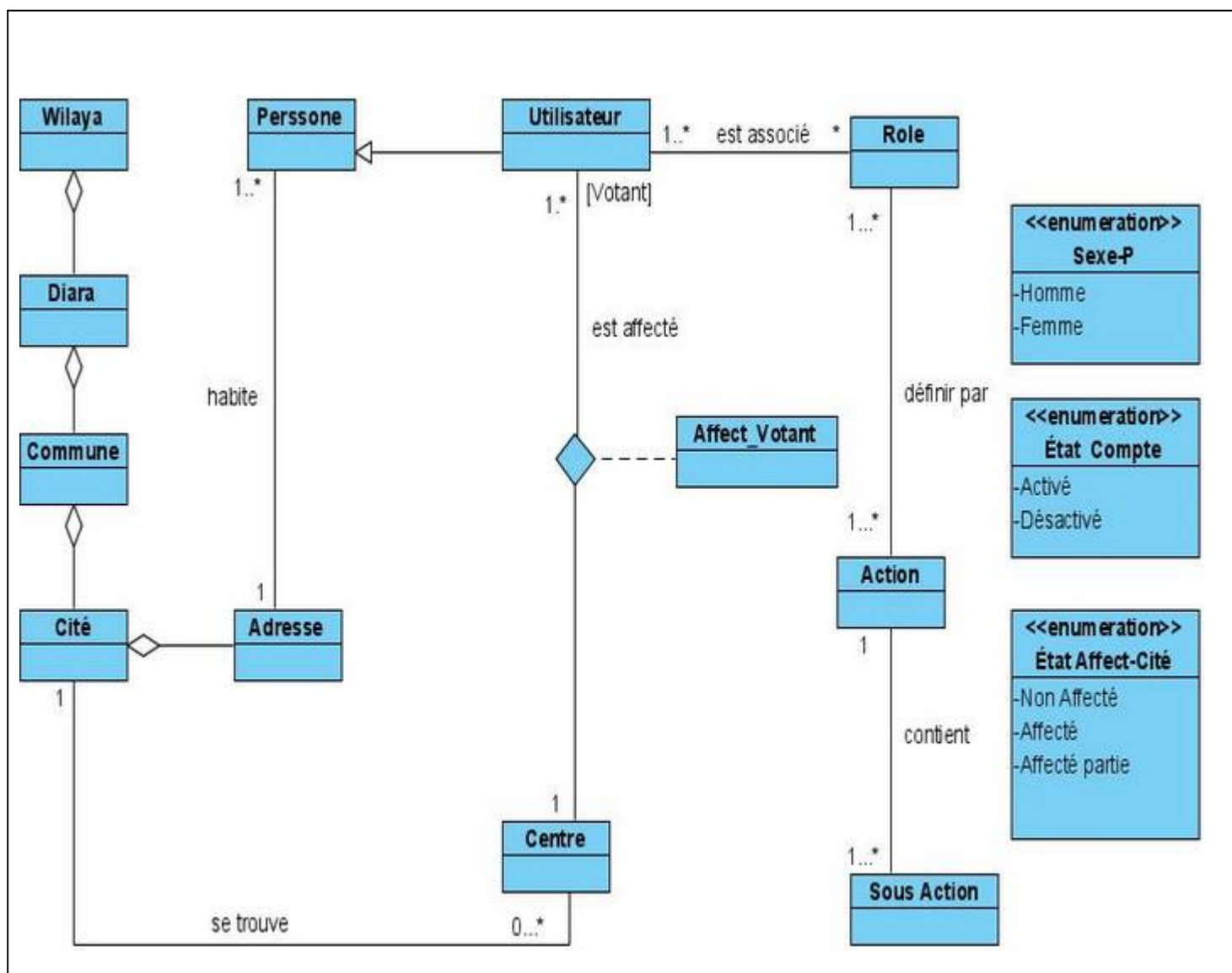


Figure 18: Diagramme des classes pour le module d’affectation

Dans le tableau 12 sont définies un ensemble d’attributs et des méthodes pour chaque classe dans notre module d’affectation.

Tableau 12:Dictionnaire de données pour le module d'affectation

Classe Personne			
Attributs	type	Signification	Méthodes
#Id personne	Entier	L'identifiant de la personne	Ajouter_p() Modifier_p() Supprimer_p() Rechercher_p() Consulter_p()
Nom_pers	Chaine	Le nom de la personne	
Prénom_pers	Chaine	Le prénom de la personne	
Date naissance	Date	Date de naissance de la personne	
Lieu naissance	Chaine	Lieu de naissance de la personne	
Sexe	Enum « Sexe-P »	Sexe de la personne (homme ou femme)	
Profession	Chaine	Profession de la personne	
Classe Utilisateur			
Attributs	Type	Signification	Méthodes
Nom_utilisateur	Chaine	Le nom de l'utilisateur	Ajouter_u() Modifier_u() Supprimer_u() Rechercher_u() Consulter_u() Activer_u() Désactiver_u()
Mot_passe	Chaine	Le mot passe de l'utilisateur	
Email	Chaine	L'adresse électronique	
Etat_compte	Enum « État Compte»	L'état de compte (activé ou désactivé)	
Classe Rôle			
Attributs	Type	Signification	Méthodes
#Id_rôle	Entier	L'identifiant de rôle	Ajouter_RI() Modifier_RI() Supprimer_RI() Rechercher_RI() Consulter_RI()
Nom_rôle	Chaine	Le nom de rôle	
Classe Action			
Attributs	Type	Signification	Méthodes
Id_action	Entier	L'identifiant de l'action	Rechercher_Ac() Consulter_Ac()
Nom_action	Chaine	Nom de l'action	
Classe Sous Action			
Attributs	type	Signification	Méthodes
Id_sousAction	Entier	L'identifiant de sous Action	Rechercher_Sac() Consulter_Sac()
Nom_sousAction	Chaine	Nom de sous action	
Classe Wilaya			
Attributs	Type	Signification	Méthodes
#Id_wilaya	Chaine	L'identifiant de la wilaya	Ajouter_W() Modifie_W() Supprimer_W()
Nom_wilaya	Chaine	Le nom de la wilaya	

Classe Daïra			
Attributs	Type	Signification	Méthodes
#Id_daïra	Chaine	L'identifiant de la daïra	Ajouter_D () Modifie_D () Supprimer_D ()
Nom_daïra	Chaine	Le nom de la daïra	
Classe Commune			
Attributs	Type	Signification	Méthodes
#Id_commune	Chaine	L'identifiant de la commune	Ajouter_C () Modifie_C () Supprimer_C ()
Nom_commune	Chaine	Le nom de la commune	
Classe Cité			
Attributs	Type	Signification	Méthodes
#Id_cité	Chaine	L'identifiant de la cité	Ajouter_Ct () Modifie_Ct () Supprimer_Ct ()
Nom_cité	Chaine	Le nom de la cité	
Lat_cité	Réel	Latitude du centre de la cité	
Logd_cité	Réel	Longitude du centre de la cité	
Etat_affectation	Enum « État Affect-Cité »	L'état d'affectation de la cité, qui est un attribut calculable (Expliqué en détail dans la remarque précédente).	
Classe Adresse			
Attributs	Type	Signification	Méthodes
#Id_adresse	Chaine	L'identifiant de l'adresse	Ajouter_Ar () Modifie_Ar () Supprimer_Ar ()
Libellé_adresse	Chaine	Libellé de l'adresse	
Lat_adresse	Réel	Latitude du lieu de l'adresse de votant (s)	
Logd_adresse	Réel	Longitude du lieu de l'adresse de votant (s)	
Classe Centre de vote			
Attributs	Type	Signification	Méthodes
#Id_centre	Chaine	L'identifiant du centre	Ajouter_Cr () Modifie_Cr () Supprimer_Cr ()
Adresse_centre	Chaine	L'adresse du centre	
Lat_centre	Réel	Latitude du lieu du centre	
Logd_centre	Réel	Longitude du lieu du centre	
Nomb_bureaux	Entier	Nombre de bureaux du centre	
Pourc_centre	Réel	Le pourcentage des votants par rapport à la capacité d'absorption du centre	

III.4.2.4. Conception de la base de données (schéma relationnel)

Nous présentons la conception du schéma relationnel de la base de données à partir du model objet qu'est le diagramme de classes. Pour ce faire, nous adoptons les règles de passage proposées par J.Rumbaugh.

PERSONNE (#Id_personne, Nom_pers, Prénom_pers, Sexe, Lieu_naissance, Date_naissance, Profession, #Id_adresse)

ROLE (#Id_rôle, Nom_rôle)

UTILISATEUR (#Id_personne, Nom_utilisateur, Mot_passe, Email, Etat_compte)

UTIL_ROLE (#Id_personne, #Id_rôle)

ACION (#Id_action, Nom_action).

ROLE_ACION (#Id_rôle, #Id_action)

SOUS ACTION (#Id_sousAction, Nom_sousAction, #Id_action).

COMMUNE (#Id_commune, Nom_commune, #Id_daïra)

DAIRA (#Id_daïra, Nom_daïra, #Id_wilaya)

WILAYA (#Id_wilaya, Nom_wilaya)

CENTRE DE VOTE (#Id_centre, Adresse_centre, Lat_centre, Logd_centre, Nomb_bureaux, Pourc_centre, #Id_cité)

CITE (#Id_cité, Nom_cité, Lat_cité, Logd_cité, Etat_affectation, #Id_commune)

ADRESSE (#Id_adresse, Libellé_adresse, Logd_adresse, Lat_adresse, #Id_cité)

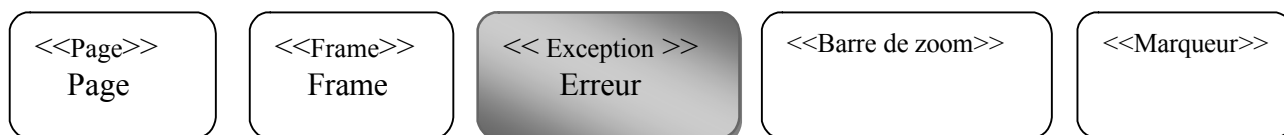
AFFECT_VOTANT (#Id_personne, #Id_centre, Date_affectation)

III.4.3. Modèle de Navigation

Nous avons commencé par identifier les cas d'utilisation puis nous avons poursuivi par une description détaillée avec les diagrammes de séquence système, puis on a identifié les diagrammes de classe. Pour que la modélisation soit complète, il nous reste à détailler la navigation à l'intérieur du site Web. Il s'agit de réaliser des diagrammes dynamiques représentant de manière formelle l'ensemble des chemins possibles entre les principaux écrans proposés à l'utilisateur. Ces diagrammes s'appellent des diagrammes de navigation.

1. Conventions spécifiques

Nous allons utiliser les conventions graphiques suivantes :



- Une page complète du site (« page »).
- Un frame particulier à l'intérieur d'une page (« frame »).
- Une erreur ou un comportement inattendu du système (« exception » avec un niveau de gris intermédiaire).
- La carte géographique (« carte »).

Ces types de convention graphique constituent un ensemble simple suffisamment complet, pour notre utilisation courante.

2. Structuration de la navigation

Il est clair que la navigation de visiteur dans le site, sera différente de celle de l'utilisateur (OSV, OECV,... etc.) qui ont plus de fonctionnalités.

La modélisation de la navigation peut donc se structurer tout d'abord par acteur. Le début du diagramme de navigation de chaque acteur est ainsi représenté sur la figure suivant.

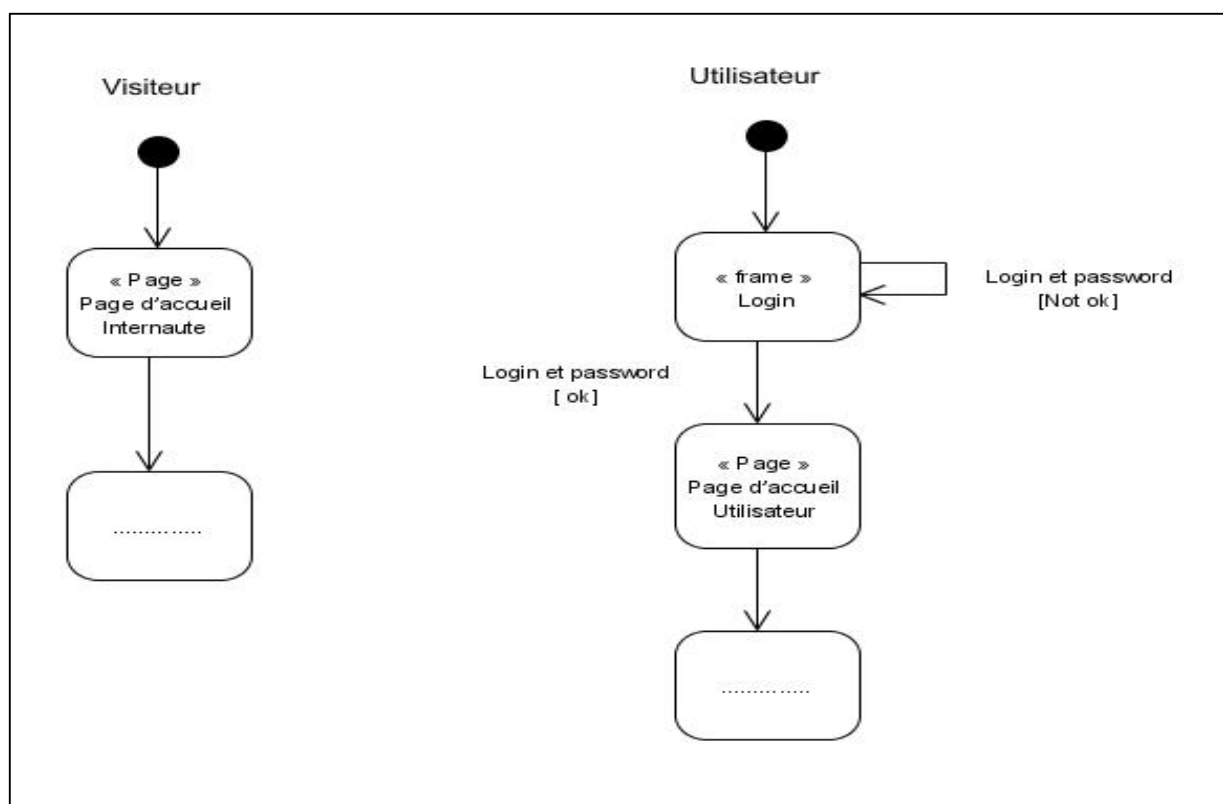


Figure 19: Début des diagrammes de navigation des acteurs internautes et utilisateur

L'internaute démarre par la page d'accueil. L'utilisateur (OSV, OECV, etc.), se connecte lui aussi au site. Mais il doit saisir son identifiant et son mot de passe dans un Frame. Suivant le résultat du contrôle effectué par le système, il se retrouve soit sur la page d'accueil avec un message d'erreur ou bien sur son espace personnel.

3 .Navigation d'OECV

Nous nous intéressons au cas d'utilisation « Valider les informations de la création des comptes» :

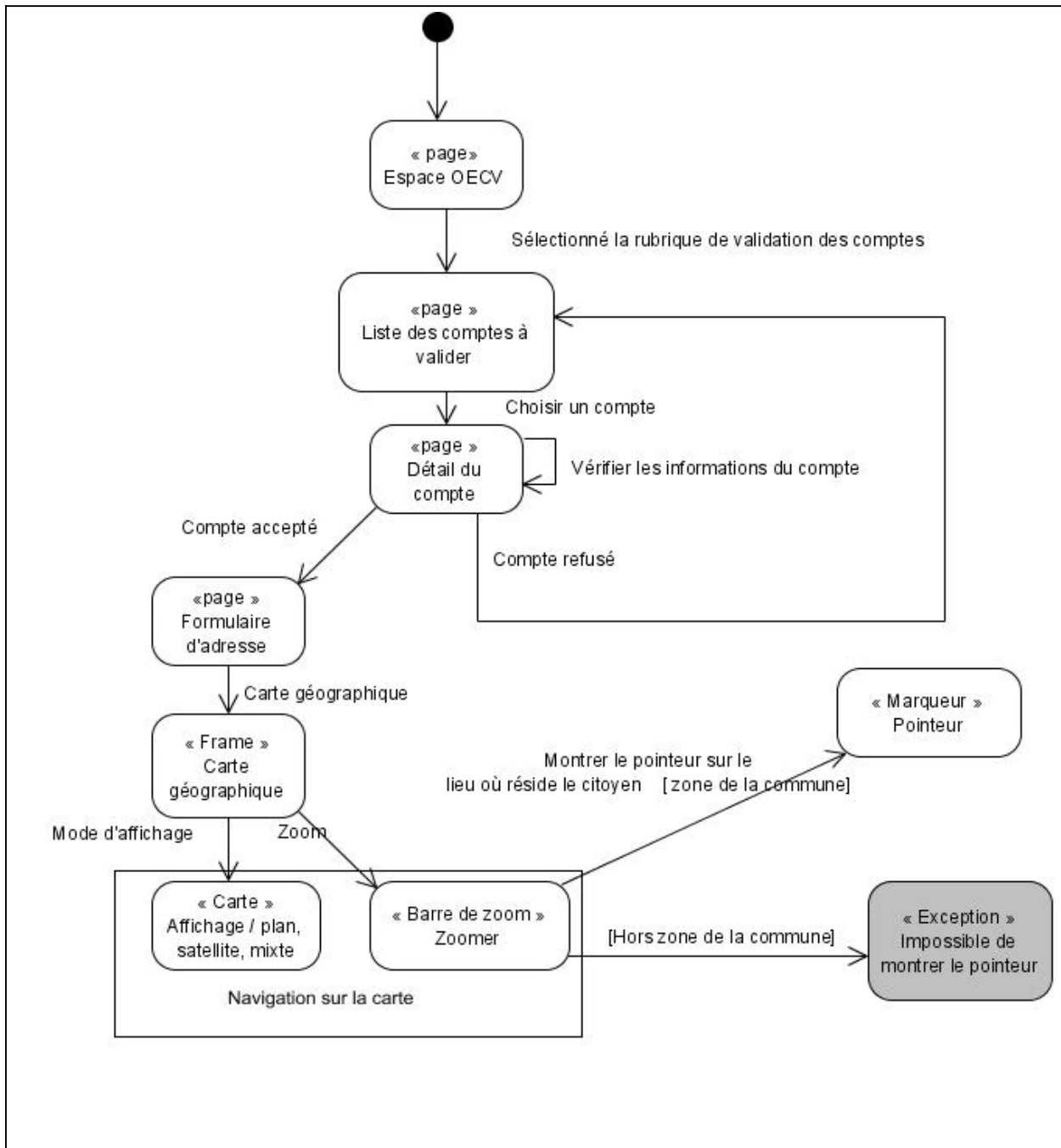


Figure 20: Diagramme de navigation de validation les informations de création des comptes

4 .Navigation d’OSV

Nous nous intéressons au cas d’utilisation « Affecter ou réaffecté les votants au centre de vote par individu » :

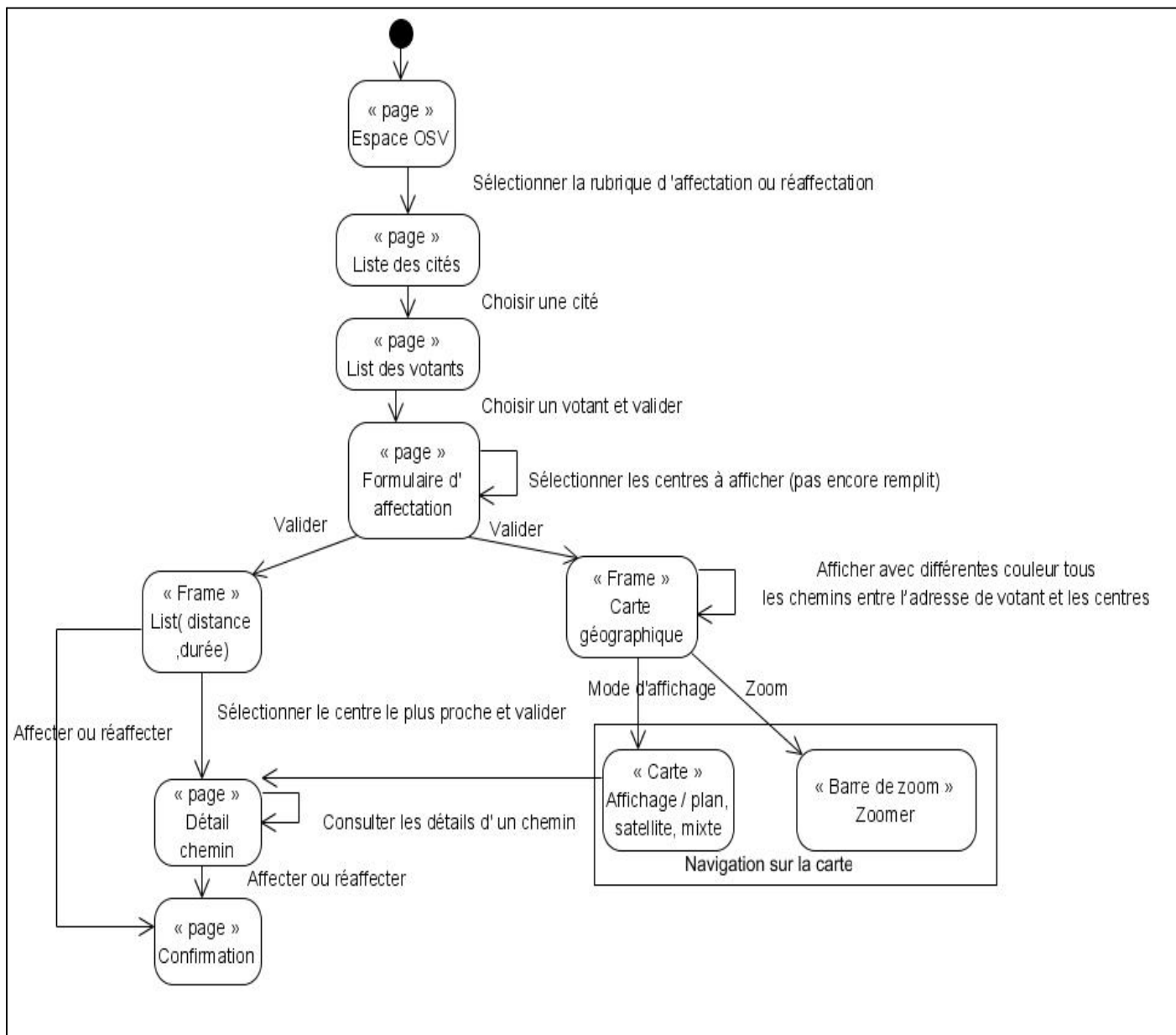


Figure 21: Diagramme de navigation d’affectation ou réaffectation des votants au centre de vote par individu

III.4.4. Modèle MVC (Modèle Vue Contrôleur)

Le modèle MVC (modèle vue contrôleur) est un moyen de décrire la séparation appropriée entre la partie d’un programme qui interagit avec l’utilisateur et celle qui s’occupe de la charge lourde, du traitement des calculs et des autres opérations métier de l’application.

Le modèle MVC identifie trois rôles qu'un composant du système peut remplir.

- ✓ Le modèle est la représentation du domaine des problèmes de l'application, c'est-à-dire de ce avec quoi elle est censée travailler.
- ✓ La vue est la partie du programme qui présente des éléments à l'utilisateur : formulaire de saisie, image, texte, éléments d'interface.
- ✓ La règle d'or du MVC stipule que la Vue et le Modèle ne doivent pas communiquer entre eux. La responsabilité de cet échange incombe en fait au contrôleur.

Lorsque l'utilisateur clique sur un bouton ou remplit un formulaire, la vue signale cet évènement au contrôleur. Le contrôleur manipule ensuite le modèle et décide si les changements dans le modèle requièrent une mise à jour de la vue. Si c'est le cas, il demande à la vue de se modifier elle-même.

Nous nous intéressons au cas d'utilisation « Ajouter un centre de vote », le diagramme correspondant est montré sur la figure 22.

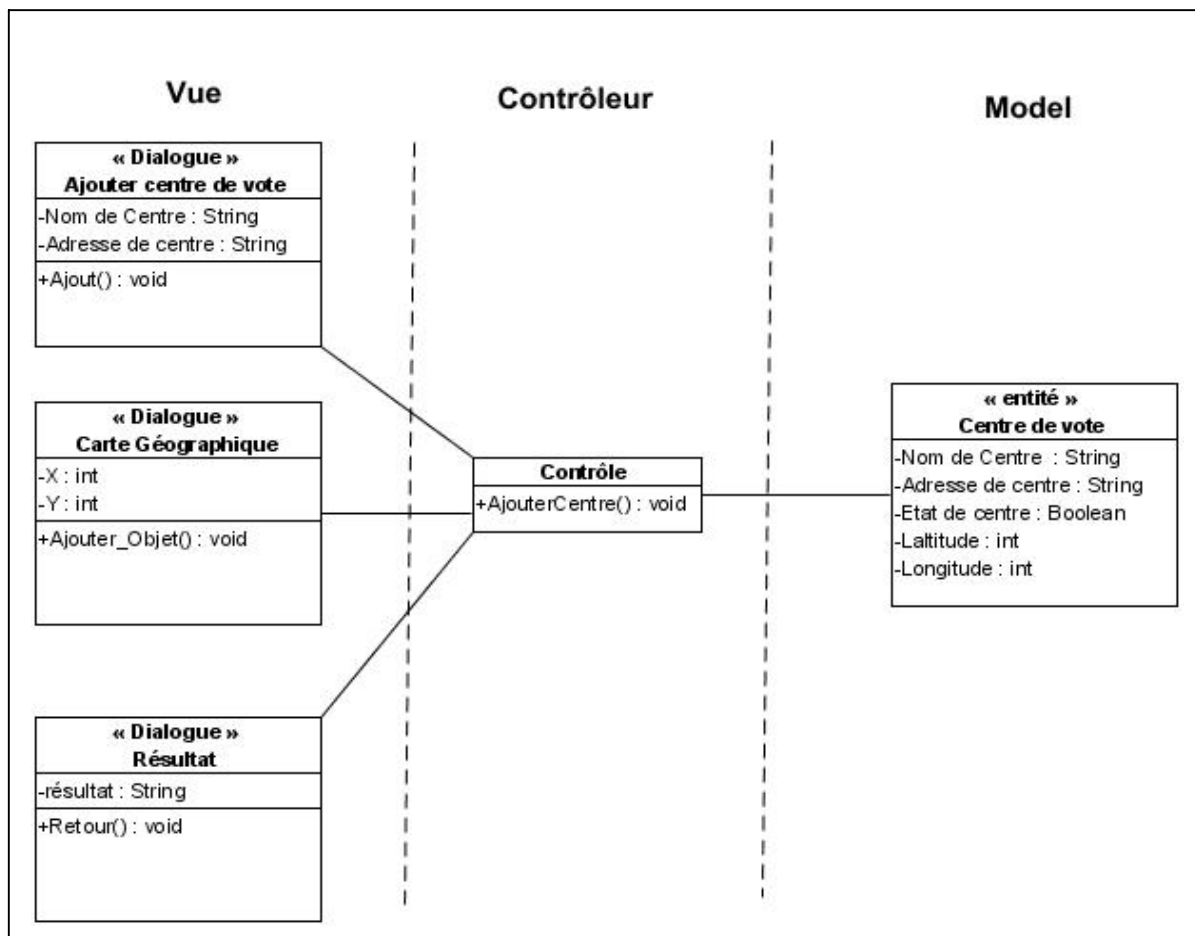


Figure 22: Diagramme de classe participant d'ajout d'un centre de vote

III.5. Conclusion

Nous avons présenté dans ce chapitre l'étude conceptuelle de notre application Web en concevant les différents modèles de conception et en présentant le fonctionnement de notre application Web.

Aussi nous avons mis en relief le fonctionnement, l'organisation et l'interaction entre les différents acteurs et le service externe utilisé.

Dans le chapitre qui va suivre, nous présenterons la réalisation de notre application en présentant le langage de programmation utilisé, le SGBD utilisé pour l'implémentation de notre base de données, les technologies et l'API utilisées, ainsi que quelques exemples d'utilisation de notre application Web.

Chapitre IV

Implémentation

IV.1. Introduction

Après l'étape de conception de la solution répondant aux objectifs fixés préalablement, nous entamons l'étape de réalisation de notre application Web, pour l'affectation des votants au centre de vote le plus adéquat qui se base sur une solution géographique.

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.

IV.2. Environnement de développement

Les technologies et outils de développement utilisées dans le module d'affectation sont les mêmes utilisées dans l'application eVOTE sauf l'API Google Maps qu'est utilisé seulement dans le module d'affectation.

IV.2.1. Les technologies utilisées

IV.2.1.1. Google Maps

Google Maps est un service web cartographique gratuit créé par Google en 2004 qui utilise la technologie SIG Web pour offrir un ensemble de fonctionnalités SIG qui s'exécutent au niveau du serveur.

Ce service a de particulier qu'il permet, à partir de l'échelle d'un pays, de pouvoir zoomer jusqu'à l'échelle d'une rue.

1. Implémentation

Google Maps est basé sur JavaScript. Lorsque l'utilisateur drague la carte, les images sont téléchargées depuis le serveur. Lorsque l'utilisateur cherche une location, une petite bulle est affichée sur la carte pour indiquer la position.

Google Maps utilise la technologie AJAX (Asynchrones JavaScript and XML) qui combine Javascript avec XML. Cette technologie a l'avantage de satisfaire les requêtes client sans rafraîchir à chaque fois la carte.

2. L'API Google Maps

L'API Google Maps était créée par Google afin de faciliter aux développeurs l'intégration de Google Maps au sein de leurs applications web avec leurs propres bases de données. En utilisant l'API Google Maps on peut avoir toutes les fonctionnalités Google Maps dans une application web externe.

L'utilisation des API de Google Maps 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) qui est indispensable pour l'utilisation.
3. La maîtrise du langage supporté par l'API (c'est le langage JavaScript). Cette étape est la plus importante.

Dans notre application nous avons utilisé l'API de Google Maps version 3 qui ne demande pas une clé d'utilisation fournie par le producteur de service par contre les autres versions (1 ou 2) demandent une clé d'utilisation.

1). Type de carte

Les différents types de carte disponible sur Google Maps sont :

- Plan : la carte routière par défaut de Google ;
- Satellite : la carte satellite obtenue par satellite ou par ortho-photographie aérienne ;
- Hybrid : superposition de la carte satellite avec des éléments de la carte routière classique ;
- Relief : la carte topographique représentant le relief et quelques informations importantes.

2). De nombreuses fonctionnalités

L'API Maps dynamiques permet de générer des cartes qui seront intégrées dans des pages Web à l'aide de code JavaScript. Elle propose 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 :

- Accéder à des services de données : image satellite/aérienne, carte topographique, Street View, etc.
- Personnaliser les fonds de carte (map styling).

- Ajouter des points, images, lignes, polygones, infos bulle.
- Ajouter ses propres données (data overlayed).
- Utiliser des **librairies** : géométrie, panoramique, Places, Static Maps...
- Utiliser des **services Web**: Calcul d'itinéraire, géocodage..., etc.

3). Les services Web géo de Google Maps

Il existe plusieurs service Web fournit par Google Maps et sont utilisés par l'API de Google Maps dont l'essentielles sont :

a. Géolocalisation

Le service Web de géolocalisation qui permet de retrouver la situation géographique d'un point donné à partir de l'adresse postale qui lui associée et le contraire.

b. Calcul d'itinéraires

Le service Web de calcul d'itinéraires qui permet de généré des itinéraires avec des informations étape par étape d'un point A à un point B .Ce service propose la génération d'itinéraire pour les véhicule légers (voiture) mais également les itinéraires piétons.

c. Streetview

Le service Web streetview est un mode cartographique qui permet de **naviguer virtuellement sur les routes** grâce à des prises de vues photographiques à 360°. Ce mode permet la visualisation des bâtiments, voies, circulation et représente au mieux la réalité du terrain. (Tout dépend de la disponibilité des données, toutes les routes du monde ou même des villes françaises ne sont pas encore prises).

IV.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 [CRA_06].

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 ».

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.

IV.2.1.3. Autres technologies utilisées

Les feuilles de style (CSS) sont utilisées pour la présentation des informations au sein des pages Web.

IV.2.2. Outils de développement

IV.2.2.1. Struts2

Apache Struts est un framework libre servant au développement d'applications Web J2EE. Il utilise et étend l'API Servlet Java afin d'encourager les développeurs à adopter l'architecture Modèle-Vue-Contrôleur [SED_11].

1. Caractéristique

- **Architecture MVC**

Struts 2.0 est basé sur l'architecture MVC. Le contrôleur peut être représenté par de nombreuses technologies standard comme Java Beans, XML, pour le model, le framework peut utiliser toutes les technologies d'accès aux données comme JDBC, EJB, Hibernante, pour la vue, le Framework Struts 2.0 n'est pas limitée à avoir JSP comme son point de vue seulement. Toute Technologie Vue peut être choisie Il peut être intégré avec JSP, JSF [SED_11].

- **Tags**

Les tags dans Struts 2 permettent la création d'applications Web dynamiques avec moins de nombres de codage. Non seulement ces balises contiennent des données de sortie, mais aussi fournissent la feuille de style de balisage qui conduit à son tour, contribue à créer les pages avec moins de code. Le nombre inférieur de codes rend aussi facile à lire et à entretenir [ROS].

2. Les actions dans Struts 2

Struts 2 est avant tout un Framework basé sur des actions. Une action est un traitement déclenché suite à une requête HTTP. Les actions seront toujours invoquées par l'intermédiaire

des données d'URL et envoyées au serveur via les paramètres de requête d'URL. Tous ces objets Servlet (requête, réponse, session) sont tous encore disponibles à l'action. L'action après le traitement effectué retourne un dénouement sous forme d'une chaîne de caractères. Une décision de navigation peut être prise en fonction de la valeur de ce dénouement. Il peut s'agir de l'affichage d'une autre page JSP, ou d'un message d'erreur [SED_11].

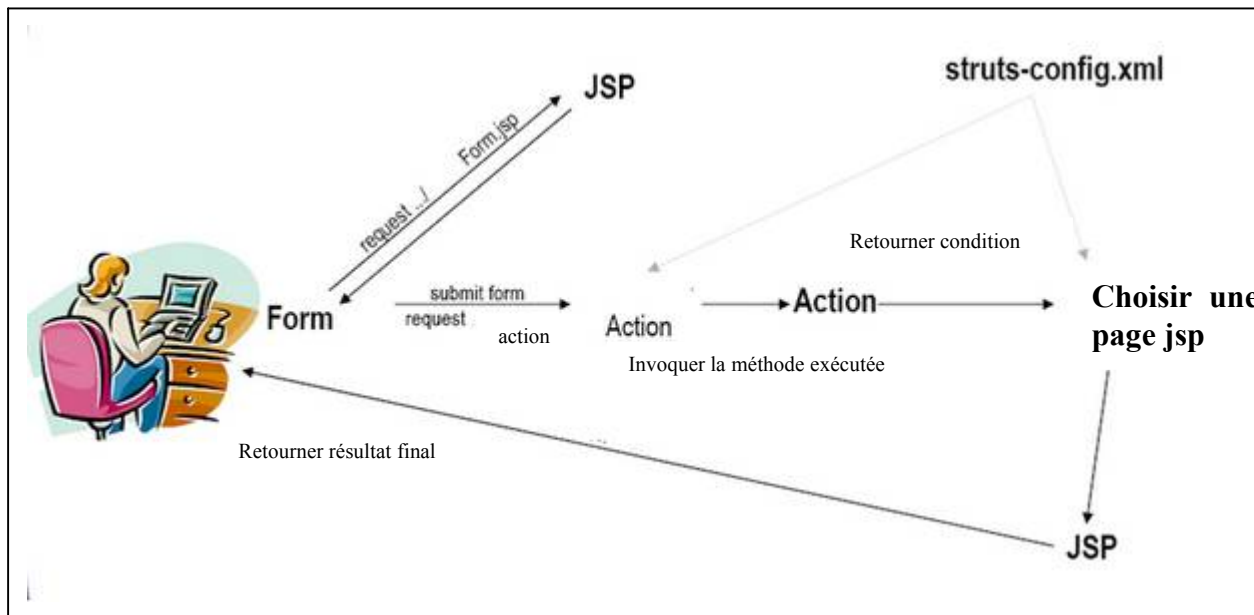


Figure 23: Chemin d'une requête dans Struts 2

IV.2.2.2. Langage de développement utilisé

Pour la réalisation de notre système on a choisis :

1. JSP

Les JSP sont une technique basée sur java qui permet la génération dynamique de page HTML. Les JSP permettent d'introduire du code Java dans des balises prédéfinies à l'intérieur d'une page HTML. Ils mélangent la puissance de java coté serveur et la facilité de mise en page d'HTML côté client.

Les JSP possèdent plusieurs avantages dont [AKA_11] :

- ✓ L'utilisation de Java qui permet une indépendance de la plate-forme d'exécution mais aussi du serveur Web utilisé ;
- ✓ La séparation des traitements de la présentation : la page Web peut être écrite par un designer et les balises JSP peuvent être ajoutées ensuite par le développeur. Les traitements peuvent être réalisés par des composants réutilisables (des Java beans, servlets).

2 .JAVA

Java est un langage de programmation orienté objet développé par Sun Microsystème. Le choix de ce langage dû aux avantages suivants [ANT_99] :

- Java est un langage orienté objets.
- Java est extensible à l'infini.
- Java est un langage à haute sécurité.
- Java est un langage simple à apprendre.
- Java est portable.

Pour programmer en java, nous avons besoin d'un environnement de développement adapté. Le JRE de Sun répond à cette exigence. Pour l'implémentation, c'est la version **jre1.5.0_16** qui a été utilisée.

IV.2.2.3. Environnement de développement intégré

L'utilisation d'un environnement de développement intégré (IDE) permet généralement d'accélérer le processus de développement. Des dizaines d'environnements de développement intégré pour java existent actuellement, tels que Eclipse, NetBeans et JDeveloper. Pour implémenter notre application, c'est l'IDE Eclipse qui a été utilisé.

IV.2.2.4. Choix du SGBD

Pour la manipulation de l'ensemble des tables constituant notre base de données, nous avons choisi le MySQL comme système de gestion de bases de données relationnelles.

MySQL utilise une architecture multiutilisateur, multitraitements. Cela permet d'établir des connexions rapides et d'utiliser la même mémoire cache pour plusieurs requêtes. De plus, parmi les fonctions de ce SGBD, celle de gérer de façon intégrée l'ensemble des données, et de les rendre accessibles à un nombre important d'utilisateurs tout en garantissant leur sécurité, leur cohérence et leur intégrité.

MySQL est un SGBD gratuit (Open Source), robuste, fiable, sécurisé, rapide et facile à utiliser. MySQL est disponible sur de nombreuses plateformes : Windows, Linux, ...

IV.2.2.5. Choix du serveur Web

Un serveur Web est un logiciel qui permet aux internautes d'accéder à des pages Web à partir d'un navigateur installé sur leurs ordinateurs distants. Un serveur Web est capable

d'interpréter les requêtes HTTP, et de fournir une réponse avec ce même protocole. On a choisi le serveur le plus répandu sur Internet actuellement : Apache Tomcat, disponible sur différents systèmes d'exploitation.

IV.3. Illustration de l'application développée

IV.3.1. Page d'accueil de l'application

Puisque notre application est une application d'eGouvernement, donc notre interface est basée sur 3 couleurs vert, blanc et rouge selon le drapeau algérien.



Figure 24: Page d'accueil de l'application eVOTE

L'application eVOTE englobe plusieurs fonctionnalités qui ne sont accessibles aux utilisateurs qu'après une inscription (page25). Dans notre cas on ce concentré sur le module d'affectation des votants du système eVOTE.



Figure 25: Inscription d'un nouvel utilisateur

L'inscription devra ensuite être validée par l'officier d'état civil valideur (OECV). Une fois l'inscription validée, le nouvel utilisateur pourra accéder à l'espace membre où il pourra avoir accès à des fonctionnalités supplémentaires.

Remarque :

Au moment de navigation des officiers (officier d'état civil valideur(OECV) et officier service de vote(OSV)) sur la carte géographique :

- ✓ La carte géographique va automatiquement se focalisée sur la zone de la commune enquêtions.
- ✓ L'utilisateur de l'application peut utiliser le pointeur uniquement dans la zone de la commune sélectionnée.

IV.3.2. Espace officié d'état civil validateur (OECV)

L'espace OECV regroupe toutes les fonctionnalités d'OECV, dans notre module d'affectation, nous nous intéressons seulement à la validation des informations de la création d'un compte.

L'OECV peut consulter la liste des comptes à valider (Figure 26).



Figure 26: Consultation la liste des comptes à valider

L'OECV doit ensuite vérifier les informations de la création d'un compte sélectionné (Figure 27).

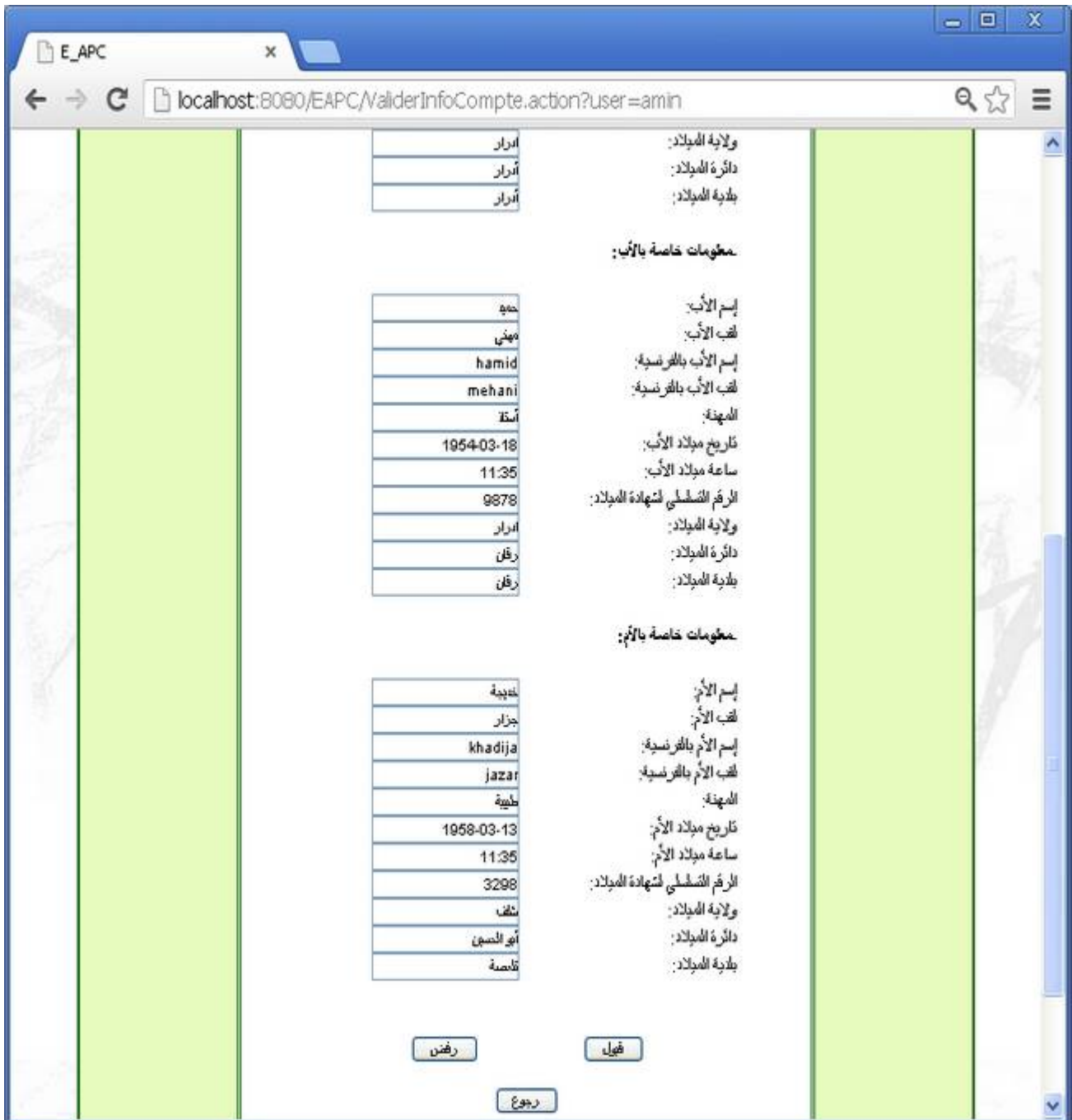


Figure 27: Vérification les informations d'un compte

Si les informations sont validées, L'OECV doit placer le pointeur sur le lieu exact où réside le citoyen sur la carte géographique (Figure 28).

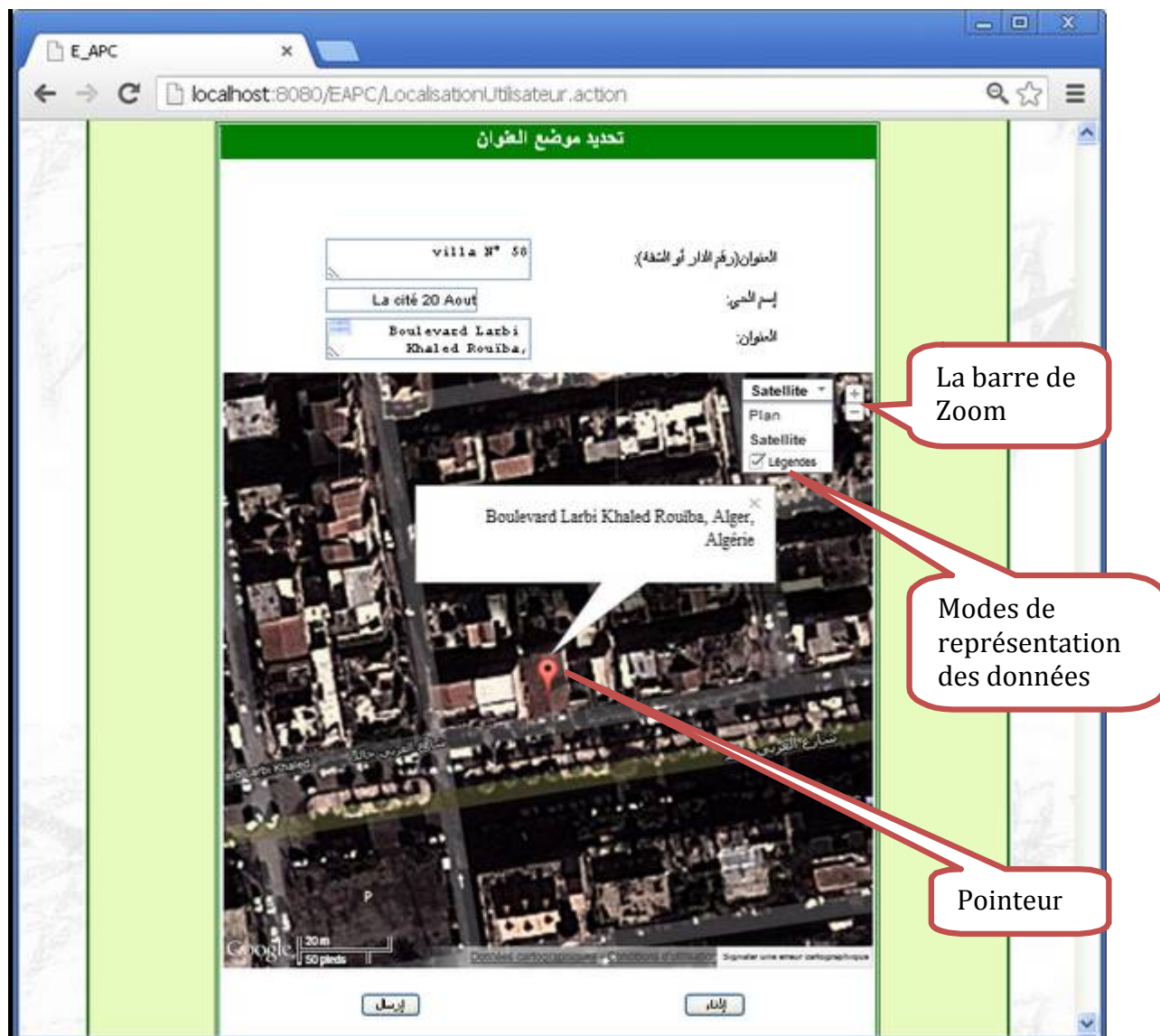


Figure 28: Localiser la position de citoyen sur la carte géographique

IV.3.3. Espace administrateur

L'administrateur est le responsable de la création des comptes des officiers d'état civil ainsi que l'activation des nouveaux comptes des citoyens après leur validation par l'OECV.

IV.3.4. Espace officier du service de vote (OSV)

L'OSV peut localiser et ajouter les nouveaux centres de vote comme illustré sur la figure 29, il s'agit du même formulaire d'ajout de cité, la différence est que dans le formulaire d'ajout d'une cité contient les informations de la cité qui sont remplies par l'OSV en parallèle doit poser le pointeur sur la carte géographique qui correspond au centre de la cité.

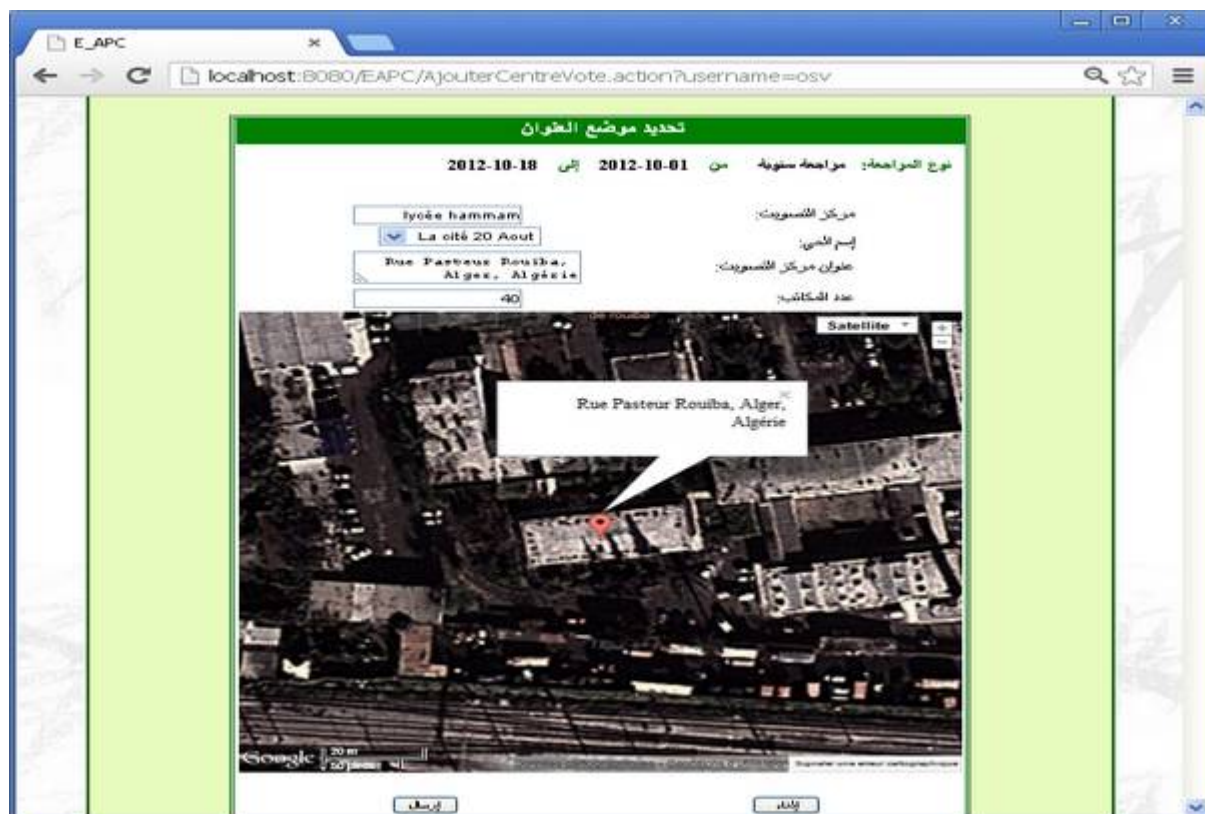




Figure 29: Localiser la position de centre de vote sur la carte géographique

L'OSV pourra également consulter la liste des cités avec l'état d'affectation de chaque cité (Figure 30).

L'état d'affectation de la cité est présenté par trois couleurs comme suit :

- La couleur blanche signifie que les votants qui appartiennent à cette cité ne sont pas encore affectés.
- La couleur rouge signifie que les votants qui appartiennent à cette cité sont tous affectés au même centre.
- La couleur verte signifie que les votants qui appartiennent à cette cité ne sont pas affectés au même centre (affectée par groupe d'individu).

Après la consultation de l'état d'affectation de la cité, l'OSV peut affecter ou réaffecter les votants au centre de vote le plus adéquat selon deux types : Affectation par cité et ce en cliquant sur le lien  ou par individu (ou groupe) en cliquant sur le lien  pour voir la liste des votants et ensuite faire une affectation par individus (ou groupe) (les deux types d'affectation sont expliquées dans le paragraphe de la cinquième DCU (**Diagramme de cas d'utilisation**) de la section III.4.1.2.2)).

Remarque :

Vu que l'affectation par cité et par individu (ou groupe) sont les même donc nous nous intéressons par l'affectation par cité



Figure 30: Consultation la liste des cités

L’OSV peut consulter le pourcentage d’affectation des votants dans chaque centre de vote afin de vérifier l’état des centre et de sélectionné les centres de vote pas encore saturé pour ensuite les affichés sur Google Maps (Figure 31).

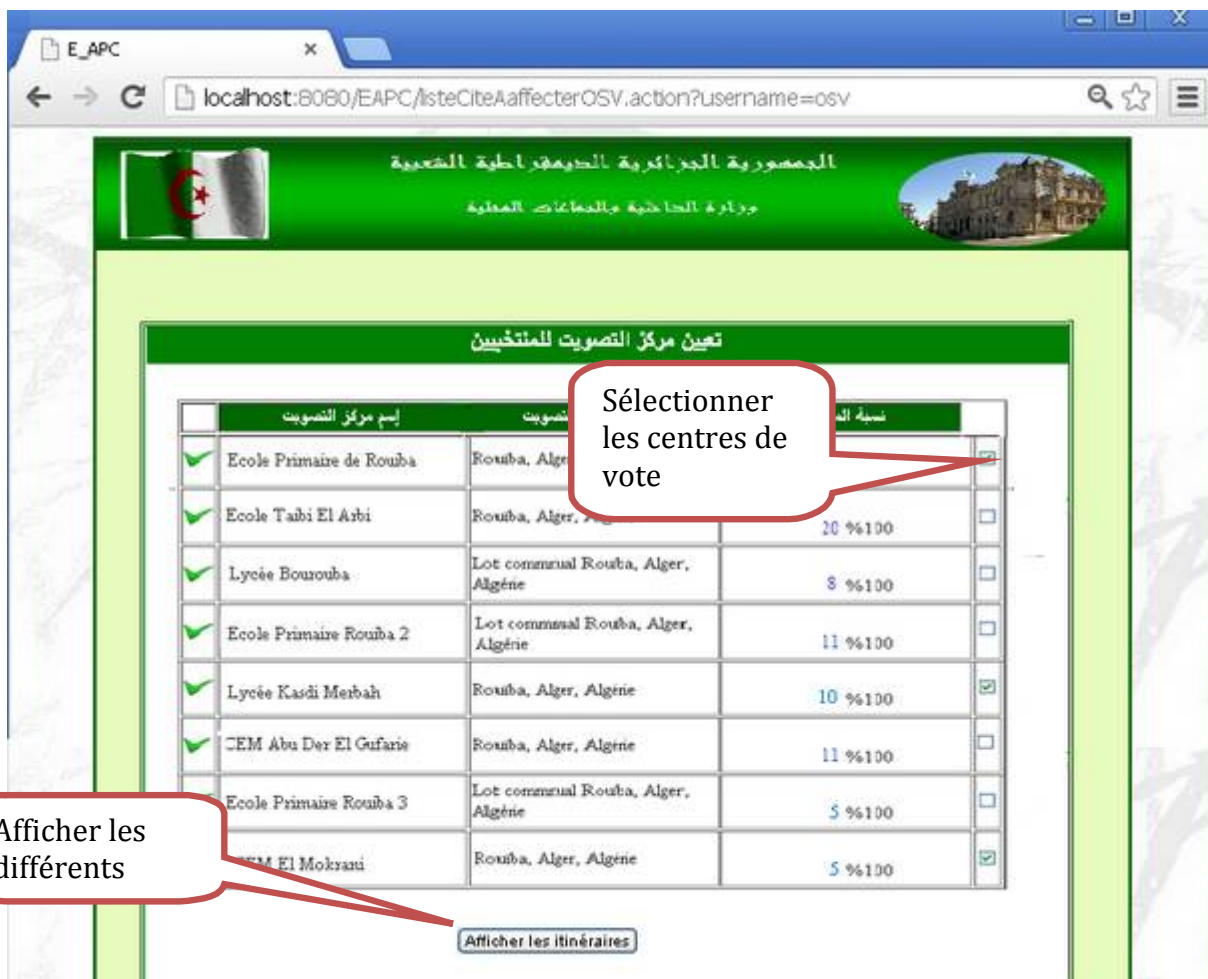


Figure 31: Affectation ou réaffectation des votants par cité « 1/2»

Ensuite l’OSV clique sur le bouton ‘Afficher les itinéraires’ pour consulter les résultats suivant (figure 32):

- L’affichage des différents trajets sera effectué avec des différentes couleurs entre le centre de la cité et les centres de vote sélectionnés sur la carte de Google Maps.
- L’affichage de la distance et la durée de chaque trajet entre le centre de la cité et les centres de vote sélectionnés que sa soit un trajet parcouru par un moyen de transport ou à pied permettra à l’OSV de choisir le trajet le plus adéquat et ce en sélectionnons le bouton radio situé dans le tableau.

Remarque :

Dans le cas d’une affectation par individu (ou groupe), les trajets afficher sur Google Maps seront entre le votant (s) et les centres de vote sélectionnés.



Figure 32: Affectation ou réaffectation des votants par cité « 2 /2»

Après la consultation des résultats d’affichage dans le tableau et sur la carte de Google Maps :

- L’OSV peut affecter les votants de cette cité vers le centre de vote le plus proche et ce en sélectionnant le bouton radio situé dans le tableau correspondant à la ligne du chemin le plus court et cliquer sur le bouton ‘affecté’.
- L’OSV peut aussi consulter les détails qui aident à l’orientation du citoyen sur le chemin sélectionné sur la carte géographique (Figure 33).

IV.3.5. Espace citoyen

Le citoyen peut consulter les détails du chemin emprunté qui aide à l’orienter depuis sa résidence jusqu’au centre de vote (Figure 33).

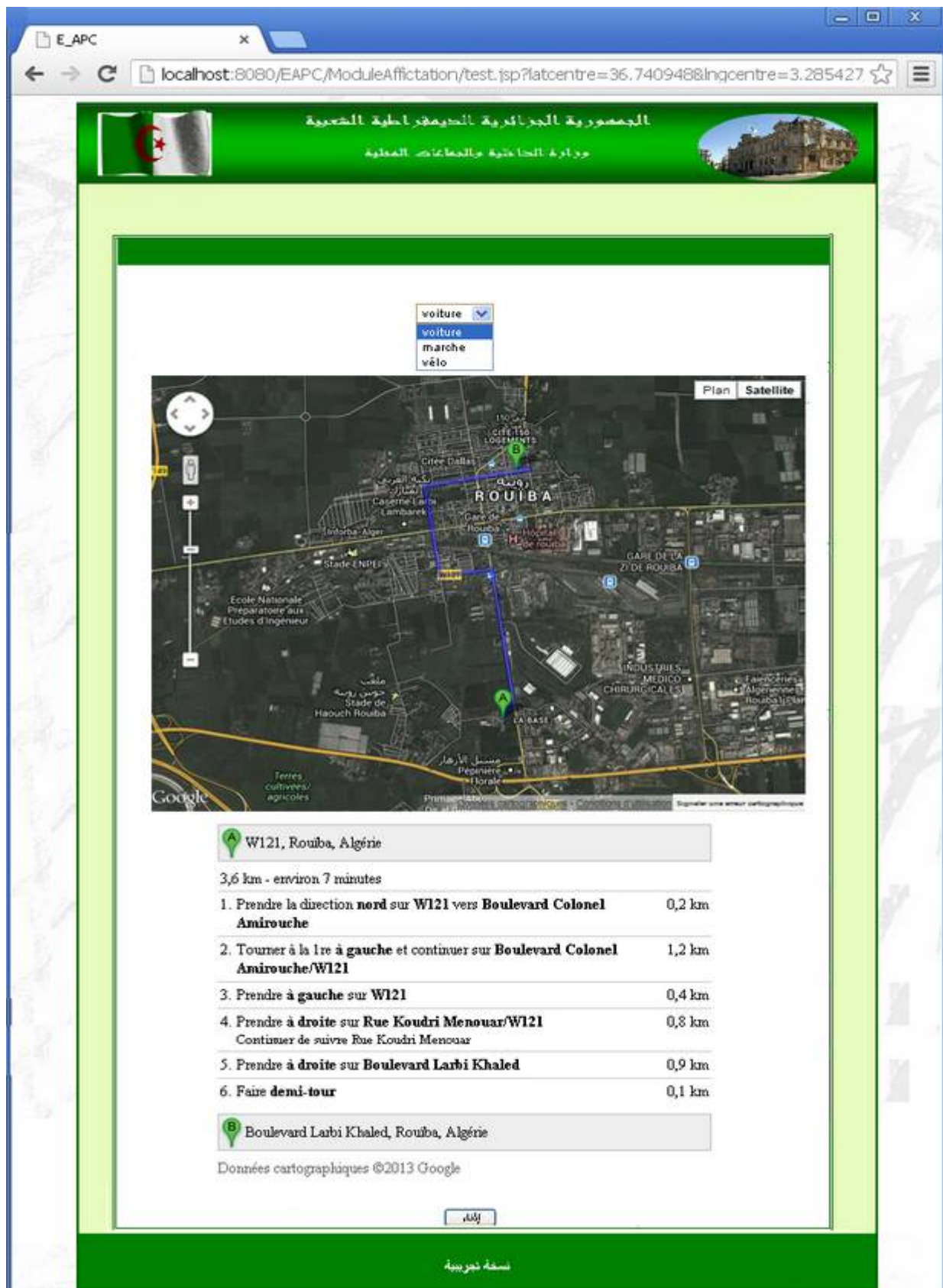


Figure 33: Consultation les détails du chemin emprunté

IV.4. Conclusion

Nous avons présenté à travers ce chapitre le module d'affectation que nous avons développée.

Nous avons présenté dans un premier temps les technologies utilisées, ainsi que l'environnement de développement de notre projet en présentant le langage et le Framework Struts 2, le SGBD.

Dans un second temps, nous avons montré quelques fonctionnalités de notre projet et les interactions des utilisateurs avec les différentes pages accompagnées des capture d'écrans correspondantes.

Lors de cette phase de réalisation, nous avons rencontré plusieurs difficultés pour utiliser les API Web de Google Maps dans notre projet. Le plus difficile était d'arriver à afficher plusieurs chemins avec différentes couleurs sur même carte géographique, ainsi que l'utilisation des technologies AJAX.

Conclusion générale et perspectives

Les données géographiques relatives à la position et à la distance séparant le votant de chaque bureau de vote de la commune ont un impact considérable sur la facilitation du mécanisme d'affectation des citoyens aux centres de vote les plus adéquats.

Le projet développement de ce module « d'affectation géographique des votants » nous a permis de découvrir de nouveaux domaines, comme celui des services Web et leur utilisation dans le eGouvernement. En effet, une longue étude théorique a été réalisée dans les deux premiers chapitres sur ces thèmes, dans le but de comprendre leurs caractéristiques et comment les exploiter dans une telle problématique.

Dans le troisième chapitre de notre mémoire, on a pu présenter le cadre conceptuel du projet en définissant l'approche de modélisation, la méthodologie de conception et les outils de design choisis dans cet objectif.

La méthode de développement utilisée dans notre travail, se situe à mi-chemin entre UP (Unifiedprocess), qui est un cadre général très complet de processus de développement, et XP (eXtremeProgramming), qui est une approche minimaliste centrée sur le code. Cette méthode utilise UML2 comme langage de modélisation. Elle nous a permis de modéliser les besoins des utilisateurs de façon claire et de représenter en détails notre système.

A la fin, Nous avons appliqué le résultat de notre conception pour réaliser le module d'affectation selon une approche MVC (Modèle Vue Contrôleur) mettant en œuvre le Framework Struts2 (contrainte technologique imposée par le système global eVOTE) et l'API Google Maps.

En conclusion, nous estimons que le système conçu a répondu aux besoins du projet proposé en permettant la gestion des données disponibles et en facilitant les traitements et les opérations effectuées sur la base de ces données.

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 :

- Personnalisation et paramétrage du module d'affection (ex : donner le choix aux votants de naviguer avec Google Maps pour indiquer sa position).
- Développer un forum de discussion pour renforcer la relation entre les votants et les responsables de service de vote.
- Développer un serveur de messagerie : Ceci permettrait de fournir une adresse électronique à chaque utilisateur inscrit.

- Utiliser notre application sans connexion Internet en exploitant un cache d'imagerie satellitaire de la commune concerné.

Il faut dire que ce projet a été riche par les connaissances qu'il nous a permis d'acquérir, d'une part, par la diversité des technologies auxquelles nous nous sommes intéressé et d'autre part par le fait qu'il nous a permis de travailler au sein d'une équipe de recherche.

Bibliographie

- [AKA_11] A. Akakpo, "Prototype d'un système de webmapping interactif avec les jsp et les servlets," Mémoire de Master en réseau et système d'information, Université d'Abomey-Calavi, Bénin, 2011.
- [AME_07] H. Ameraoui, A. Boudissa, "Applications Géographique Orientées Web Services," Mémoire d'Ingénieur d'Etat en Informatique, U.S.T.H.B, Bab Zouar, Alger, 2007.
- [ANT_99] M. Antoine, "Le développeur JAVA2," Édition Eyrolles ,1999.
- [AUS_04] D. Austin, A. Barbir, C. Ferris, G. Sharad, "Web Services Architecture Requirements," W3C, HTTP: //www.w3.org/TR/wsa-reqs, February 2004.
- [BAB_03] G. Babin, M. Leblanc, "Les web services et leur impact sur le commerce2b," CIRANO Burgundy Reports 2003rb-07, CIRANO, 2003.
- [BAR_03] D. K. Barry, "The Savvy Manager's Guide to Web Services and Service-Oriented Architectures," Morgan Kaufmann Publishers Inc, San Francisco, CA, USA, 2003.
- [BEN_09] R. Ben halima, " Conception, implantation et expérimentation d'une architecture en bus pour l'auto-réparation des applications distribuées à base de services web," Thèse de Doctorat, Université Toulouse III, 2009.
- [BOO_04] D. Booth, H. Haas, F. McCabe, E. Newcomer, M. Champion, C. Ferris, D. Orchard, "Web services architecture" Technical report, W3C, Web Services Architecture Working Group, February 2004.
- [CRA_06] D. Crane, " Ajax en pratique," Edition Campus Press, 2006.
- [DEV_03] K. Devaram, D. Andresen, "Soap optimization via parameterized client-side caching," In proc of International Conference on parallel and Distributed Computing System (ICPDCS), pp. 785–790, November 2003.
- [DIG_07] Digimind, "Le Web 2.0 pour la veille et la recherche d'information," Livre réalisé par la société Digimind . Juin 2007.
- (Digimind, spécialiste des logiciels de veille stratégique et éditeur de la plate-forme de veille Digimind Evolution).
- [GHA_11] M. Gharzouli, " Composition des Web Services Sémantiques dans les systèmes Peer-to-Peer," Thèse de Doctorat, Université Mentouri de Constantine, 2011.
- [GON_10] A. Goncalves " Java EE6 et Glassfish3," Éditions Pearson, juin 2010.
- [MEL_04] T. Melliti, " Interopérabilité des Services Web complexes," Thèse de Doctorat, Université Paris IX Dauphine, 2004.
- [ROQ_06] P. Roque, " UML2 modéliser une application Web," Edition Eyrolles, 2006.
- [ROS] [HTTP://www.roseindia.net/struts/struts2/struts-2-features.html](http://www.roseindia.net/struts/struts2/struts-2-features.html)

- [SED_11] W. Seddaoui, “ Conception Orientée Objet d’un composant gérant l’état Civil pour le système eAPC,” Mémoire de Master en Informatique, USDB, Blida, 2011.
- [UDD_00] UDDI, [HTTP://www.uddi.org/pubs/Iru_UDDI_Technical_White_Paper.pdf](http://www.uddi.org/pubs/Iru_UDDI_Technical_White_Paper.pdf), UDDI technical white paper, September 2000.
- [W3C_02] W3C “Web services Architecture Requirements,” Oct.2002, [HTTP://www.w3.org/TR/ws-gloss/](http://www.w3.org/TR/ws-gloss/)
- [WEE_05] S. Weerawarana, F. Curbera, F. Leymann, T. Storey, D. Ferguson, “Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging and More,” Prentice Hall PTR, Upper Saddle River, NJ, USA, 2005.
- [YKH_11] H. Ykhlef, “ Architecture Orientée Service du Système de communication de eAPC,” Mémoire de Master en Informatique, USDB, Blida, 2011.

Glossaire

Apache Tomcat : est un conteneur libre de servlet Java 2 Enterprise Edition. Issu du projet Jakarta, Tomcat est désormais un projet principal de la fondation Apache. Tomcat implémente les spécifications des servlets et des JSP de Sun Microsystems. Il inclut des outils pour la configuration et la gestion, mais peut également être configuré en éditant des fichiers de configuration XML. Comme Tomcat inclut un serveur HTTP interne, il est aussi considéré comme un serveur HTTP.

Application Web : En informatique une application Web (aussi appelée site Web dynamique) est un logiciel applicatif dont l'interface homme-machine imite un site Web. L'interface homme-machine est appelée interface Web. Une application Web se manipule avec un navigateur Web en parcourant les liens hypertexte et en actionnant des widgets contenus dans les pages.

EJB : la technologie Entreprise JavaBean (EJB) est une architecture de composants logiciels coté serveur pour la plateforme de développement J2EE, elle permet de faciliter la création d'applications distribuées pour les entreprises.

Framework : est un ensemble de bibliothèques, d'outils et de conventions permettant le développement d'applications.

Hibernante : est un Framework open source gérant la persistance des objets en base de données relationnelle .

HTML : L'Hypertext Markup Language, généralement abrégé HTML, est le format de données conçu pour représenter les pages Web. C'est un langage de balisage qui permet d'écrire de l'hypertexte, d'où son nom.

HTTP : Le HyperText Transfer Protocol, plus connu sous l'abréviation HTTP, littéralement le « protocole de transfert hypertexte », est un protocole de communication client-serveur développé pour le World Wide Web.

J2EE : Java Enterprise Edition, ou Java EE (anciennement J2EE), est une spécification pour la technologie Java de Sun plus particulièrement destinée aux applications d'entreprise.

Java Bean : sont des composants logiciels écrits en langage Java .

JDBC : La technologie JDBC (Java DataBase Connectivity) est une API fournie avec Java permettant de se connecter à des bases de données, c'est-à-dire que JDBC constitue un ensemble de classes permettant de développer des applications capables de se connecter à des serveurs de bases de données (SGBD).

JSF : (abrégé en JSF) est un Framework Java, pour le développement d'applications Web, est agnostique à la technologie de présentation. Il utilise JSP par défaut, mais peut être utilisé avec d'autres technologies, comme par exemple Facelets ou XUL.

Servlet : est une application Java qui permet de créer dynamiquement des données au sein d'un serveur HTTP .

SGBD : Un système de gestion de base de données (abrégé SGBD) est un ensemble de logiciels qui sert à la manipulation des bases de données. Il sert à effectuer des opérations ordinaires telles que consulter, modifier, construire, organiser, transformer, copier, sauvegarder ou restaurer des bases de données. Il est souvent utilisé par d'autres logiciels ainsi que les administrateurs ou les développeurs.

Spring : est un Framework open source J2EE pour les applications 3-tiers, dont il facilite le développement et les tests.

XML : (Extensible Markup Language « langage extensible de balisage ») est un langage informatique de balisage générique. Il sert essentiellement à stocker/transférer des données de type texte Unicode structurées en champs arborescents.

W3C : (World Wide Web Consortium) est un organisme international qui développe des standards pour le Web afin que les gens puissent communiquer efficacement à travers Internet, autour de formats ouverts garantissant une meilleure inter-opérabilité (c'est-à-dire une meilleure compréhension des systèmes hétérogènes à travers des données et langages standardisés).

Web Mapping : Le processus de distribution de cartes via un réseau tel que l'Internet ou l'Intranet et leur visualisation dans un navigateur.

Annexe

1. Processus Unifié (UP)

Le Processus Unifié (UP, pour Unified Process) est un processus de développement logiciel « itératif et incrémental, centré sur l'architecture, conduit par les cas d'utilisation et piloté par les risques »

1.1 .Les principes fondamentaux du Processus Unifié (UP)

- Itératif et incrémental : le projet est découpé en itérations de courte durée qui aident à mieux suivre l'avancement global. A la fin de chaque itération, une partie exécutable du système final est produite, de façon incrémentale.
- Centré sur l'architecture : tout système complexe doit être décomposé en parties modulaires afin de garantir une maintenance et une évolution facilitées. Cette architecture (fonctionnelle, logique, matérielle, etc.) doit être modélisée en UML et pas seulement documentée en texte.
- Piloté par les risques : les risques majeurs du projet doivent être identifiés au plus tôt, mais surtout levés le plus rapidement possible. Les mesures à prendre dans ce cadre déterminent l'ordre des itérations.
- Conduit par les cas d'utilisation : le projet est mené en tenant compte des besoins et des exigences des utilisateurs. Les cas d'utilisation du futur système sont identifiés, décrits avec précision et priorisés.

1.2. Les phases et les disciplines d'UP

La gestion d'un tel processus est organisée suivant les quatre phases suivantes : initialisation, élaboration, construction et transition.

La phase d'initialisation conduit à définir la « vision » du projet, sa portée, sa faisabilité, son "business cas ", afin de pouvoir décider au mieux de sa poursuite ou de son arrêt.

La phase d'élaboration poursuit trois objectifs principaux en parallèle :

- Identifier et décrire la majeure partie des besoins des utilisateurs,
- Construire (et pas seulement décrire dans un document !) l'architecture de base du système,
- Lever les risques majeurs du projet.

La phase de construction consiste surtout à concevoir et implémenter l'ensemble des éléments opérationnels (autres que ceux de l'architecture de base). C'est la phase la plus consommatrice en ressources et en effort.

Enfin, la phase de transition permet de faire passer le système informatique des mains des développeurs à celles des utilisateurs finaux. Les mots-clés sont : conversion des données, formation des utilisateurs, déploiement, bêta-tests.

Chaque phase est elle-même décomposée séquentiellement en itérations limitées dans le temps. Le résultat de chacune d'elles est un système testé, intégré et exécutable. L'approche itérative est fondée sur la croissance et l'affinement successifs d'un système par le biais d'itérations multiples, feed-back et adaptation cyclique étant les moteurs principaux permettant de converger vers un système satisfaisant. Le système croît avec le temps de façon incrémentale, itération par itération, et c'est pourquoi cette méthode porte également le nom de développement itératif et incrémental. Il s'agit là du principe le plus important du Processus Unifié.

Les activités de développement sont définies par cinq disciplines fondamentales qui décrivent la capture des exigences, l'analyse et la conception, l'implémentation, le test et le déploiement. La modélisation métier est une discipline amont optionnelle et transverse aux projets. Enfin, trois disciplines appelées de support complètent le tableau : gestion de projet, gestion du changement et de la configuration, ainsi que la mise à disposition d'un environnement complet de développement incluant aussi bien des outils informatiques que des documents et des guides méthodologiques.

2. Les pratiques de l'eXtreme Programming (XP)

L'eXtreme Programming (XP) est un ensemble de pratiques qui couvre une grande partie des activités de la réalisation d'un logiciel, de la programmation proprement dite à la planification du projet, en passant par l'organisation de l'équipe de développement et les échanges avec le client. Ces pratiques ne sont pas révolutionnaires : il s'agit simplement de pratiques de bon sens mises en oeuvre par des développeurs ou des chefs de projet expérimentés, telles que

- Un utilisateur à plein-temps dans la salle projet. Ceci permet une communication intensive et permanente entre les clients et les développeurs, aussi bien pour l'expression des besoins que pour la validation des livraisons.

- Écrire le test unitaire avant le code qu'il doit tester, afin d'être certain que le test sera systématiquement écrit et non pas négligé.
- Programmer en binôme, afin d'homogénéiser la connaissance du système au sein des développeurs, et de permettre aux débutants d'apprendre auprès des experts. Le code devient ainsi une propriété collective et non individuelle, que tous les développeurs ont le droit de modifier.
- Intégrer de façon continue, pour ne pas repousser à la fin du projet le risque majeur de l'intégration des modules logiciels écrits par des équipes ou des personnes différentes. Etc.

Pour résumer, on peut dire que XP est une méthodologie légère qui met l'accent sur l'activité de programmation et qui s'appuie sur les valeurs suivantes : communication, simplicité et feedback. Elle est bien adaptée pour des projets de taille moyenne où le contexte (besoins des utilisateurs, technologies informatiques) évolue en permanence.

3 .La modélisation avec UML 2.0 [ROQ_06]

3.1. Introduction

Le recours à la modélisation est depuis longtemps une pratique indispensable au développement logiciel, car un modèle est prévu pour arriver à anticiper les résultats du codage. Un modèle est en effet une représentation abstraite d'un système destiné à en faciliter l'étude et à le documenter. En outre, les systèmes devenant de plus en plus complexes, leur compréhension et leur maîtrise globale dépassent les capacités d'un seul individu. La construction d'un modèle abstrait aide à y remédier.

Aujourd'hui, UML (Unified Modeling Language) est devenue un standard incontournable dans le monde de la modélisation parce qu'il offre plusieurs avantages, et surtout à cause de ça façon de modéliser de manière claire et précise la structure (vue statique) et le comportement (vue dynamique) d'un système indépendamment de toute méthode ou de tout langage de programmation.

3.2. Unified modeling language (UML)

3.2.1. Définition

UML se définit comme un langage de modélisation graphique et textuelle destiné comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue.

UML unifie à la fois les notations et les concepts orientés objet. Il ne s'agit sémantique précise et sont porteurs de sens au même titre que les mots d'un langage.

3.2.2. Historique d'UML

Né de la fusion des méthodes objets dominantes **OMT** (Object Modeling Technique), **BOOCH**, **OOSE** en 1994, puis normalisé par l'**OMG** (Object Management Group) en 1997, UML est rapidement devenu un standard incontournable. L'année dernière a connue l'arrivée de la dernière version d'UML ; il s'agit de la version UML 2.0 très adaptée aux besoins de modélisation des applications web avec la création de nouveaux diagrammes.

3.3. Les diagrammes d'UML

Un diagramme est la représentation graphique d'un ensemble d'éléments qui constituent un système. Il forme d'un graphe connexe où les sommets correspondent aux éléments et les arcs aux relations.

UML 2.0 s'articule autour de treize types de diagrammes ; chacun d'eux étant dédié à la représentation d'un concept particulier du système. Ces types de diagrammes sont répartis en deux grands groupes :

3.3.1. Diagrammes structurels « Représentent la vue statique du système »

- ✓ **Diagramme de classes** : Il montre les briques de base statiques : Classes, associations, interfaces, attributs, opérations, généralisations, etc.
- ✓ **Diagramme d'objets** : Il montre les instances des éléments structurels et leurs liens à l'exécution.
- ✓ **Diagramme de packages** : Il montre l'organisation logique du modèle et les relations entre packages.
- ✓ **Diagramme de structure composite** : Il montre l'organisation interne d'un élément statique complexe.
- ✓ **Diagramme de composants** : Il montre des structures complexes, avec leurs interfaces fournies et requises.
- ✓ **Diagramme de déploiement** : Il montre le déploiement physique des "artefacts" sur les ressources matérielles.

3.3.2. Diagrammes comportementaux «Représentent la vue dynamique du système»

- ✓ **Diagramme de cas d'utilisation** : Il montre les interactions fonctionnelles entre les acteurs et le système à l'étude.
- ✓ **Diagramme de séquence** : Il montre la séquence verticale des messages passés entre objets au sein d'une interaction.
- ✓ **Diagramme d'activité** : Il montre l'enchaînement des actions et décisions au sein d'une activité.
- ✓ **Diagramme d'états** : Il montre les différents états et transitions possibles des objets d'une classe.
- ✓ **Diagramme de communication** : Il montre la communication entre objets dans le plan au sein d'une interaction.
- ✓ **Diagramme de vue d'ensemble des interactions** : Il fusionne les diagrammes d'activité et de séquence pour combiner des fragments d'interaction avec des décisions et des flots.
- ✓ **Diagramme de temps** : Il fusionne les diagrammes d'états et de séquence pour montrer l'évolution de l'état d'un objet au cours du temps.

Pour notre conception du système on a utilisé seulement 4 types de diagrammes en expliquant les mots clé utilisés et les abréviations correspondantes.

3.4. Diagramme de cas d'utilisation

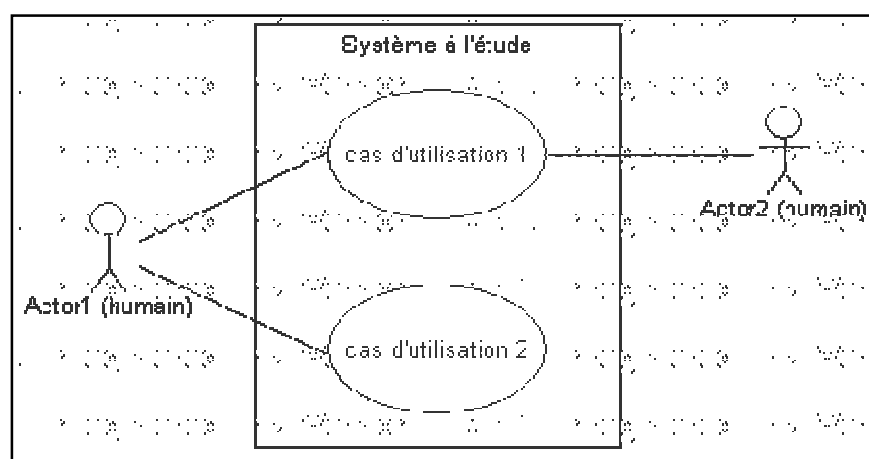


Figure 1: Diagramme de cas d'utilisation (bases)

- **Acteur** : rôle joué par un utilisateur humain ou un autre système qui interagit directement avec le système étudié. Un acteur participe à au moins un cas d'utilisation.

- **Cas d'utilisation (use case)** : ensemble de séquences d'actions réalisées par le système produisant un résultat observable intéressant pour un acteur particulier. Collection de scénarios reliés par un objectif utilisateur commun.
- **Association** : utilisée dans ce type de diagramme pour relier les acteurs et les cas d'utilisation par une relation qui signifie simplement « participe à »
- **Inclusion («include»)** : le cas d'utilisation de base en incorpore explicitement un autre, de façon obligatoire, à un endroit spécifié dans ses enchaînements.
- **Extension («Extend»)** : le cas d'utilisation de base en incorpore implicitement un autre, de façon optionnelle, à un endroit spécifié indirectement dans celui qui procède à l'extension.
- **Généralisation** : les cas d'utilisation descendants héritent de la description de leur parent commun. Chacun d'entre eux peut néanmoins comprendre des relations spécifiques supplémentaires avec d'autres acteurs ou cas d'utilisation. La généralisation d'acteurs est en revanche parfois utile.

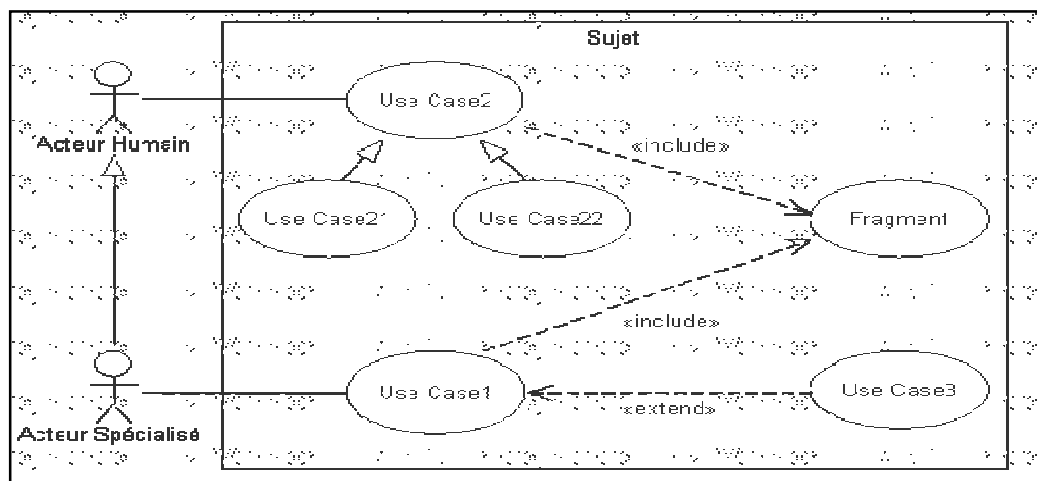


Figure 2 : Diagramme de cas d'utilisation (avancé)

3.5. Diagramme de séquence

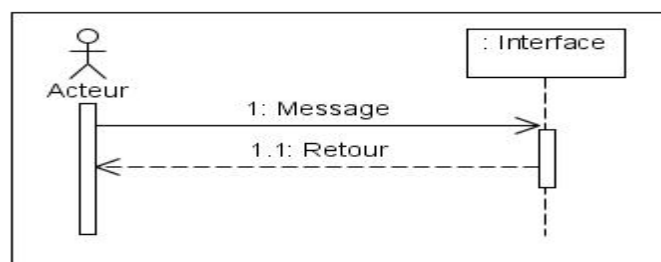


Figure 3 : Diagramme de séquence (bases)

- **Ligne de vie** : représentation de l'existence d'un élément participant dans un diagramme de séquence. Cela peut être un acteur ou le système en modélisation d'exigence, des objets logiciels en conception préliminaire ou conception détaillée.
- **Message** : élément de communication unidirectionnel entre objets qui déclenche une activité dans l'objet destinataire. La réception d'un message provoque un événement dans l'objet récepteur. La flèche pointillée représente un retour au sens UML. Cela signifie que le message en question est le résultat direct du message précédent.

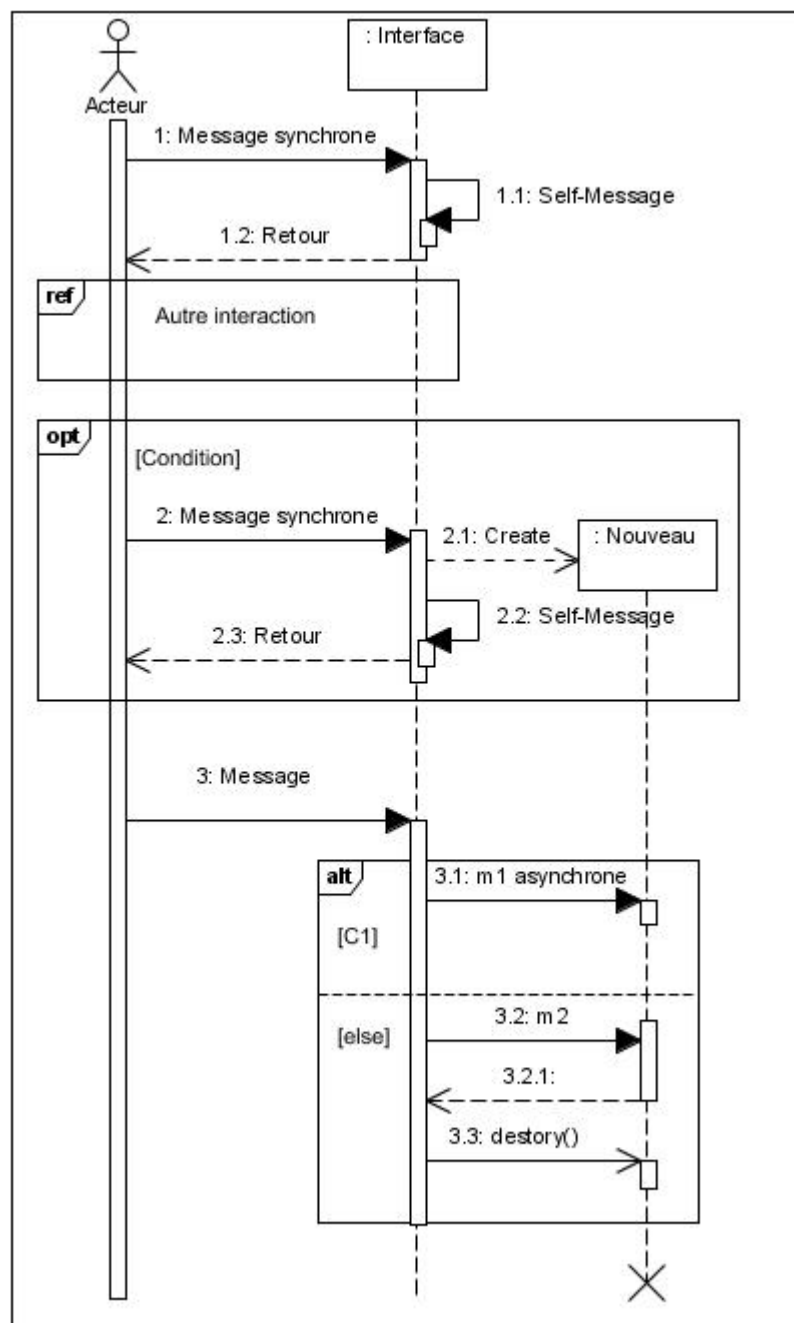


Figure 4 : Diagramme de séquence (avancé)

- **Spécification d'activation** : bande blanche qui représente une période d'activité sur une ligne de vie.
- **Message synchrone** : envoi de message pour lequel l'émetteur se bloque en attente du retour et qui est représenté par une flèche plein. Un message asynchrone, au contraire, est représenté par une flèche ouverte.
- **Occurrence d'interaction** : une interaction peut faire référence explicitement à une autre interaction grâce à un cadre avec le mot-clé «**ref**» et indiquant le nom de l'autre interaction.

UML 2.0 a ajouté une nouvelle notation très utile : les cadres d'interaction. Chaque cadre possède un opérateur et peut être divisé en fragments. Les principaux opérateurs sont :

- **loop** : boucle. Le fragment peut s'exécuter plusieurs fois, et la condition de garde explicite l'itération.
- **opt** : optionnel. Le fragment ne s'exécute que si la condition fournie est vraie.
- **alt** : fragment alternatifs. Seul le fragment possédant la condition vraie s'exécutera.

3.6. Diagramme de collaboration

Le diagramme de collaboration représente la même chose, mais différemment. Les messages entre objets sont numérotés en fonction de la ligne de vie. Il permet une vision différente. Le diagramme de séquence montre le déroulement des opérations tandis que le diagramme de collaboration insiste plus sur la structure.

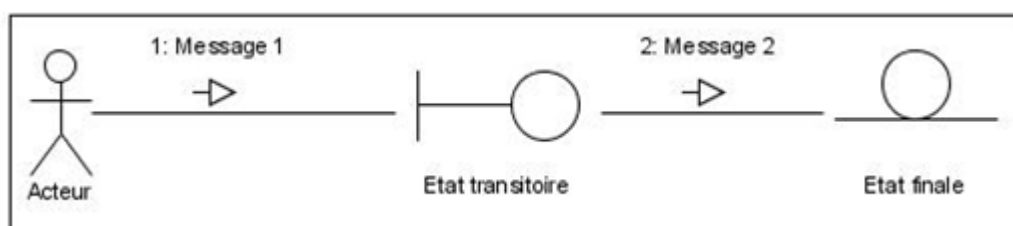


Figure 5 : Diagramme de collaboration

- **Acteur** : Un acteur représente un rôle joué par une personne qui interagit avec le système.
- **Objet** : Un objet dans le diagramme de collaboration est présenté par deux formes d'état : état transitoire ou état finale.
- **Message**: Message émis par un objet, flèche orientée de l'émetteur vers le récepteur. Nom du message s'écrit sur le trait continu qui relie les objets.

3.7. Diagramme de classes

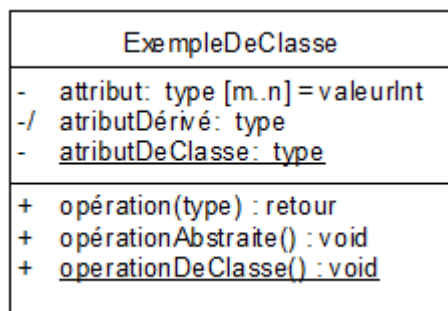


Figure 6 : Diagramme de classes (bases)

- **Classe** : description abstraite d'un ensemble d'objets qui partagent les mêmes propriétés (attribut et associations) et comportements (opérations et états).
- **Attribut** : donnée déclarée au niveau d'une classe, éventuellement typée, à laquelle chacun des objets de cette classe donne une valeur. Un attribut peut posséder une multiplicité et une valeur initiale. Un attribut dérivé (« / ») est un attribut dont la valeur peut être déduite d'autres informations disponibles dans le modèle.
- **Opération** : élément de comportement des objets, défini de manière globale dans leur classe. Une opération peut déclarer des paramètres (eux-mêmes types) ainsi qu'un type de retour.
- **Visibilité des attributs et des opérations**: Il existe trois visibilités prédéfinies.
 - ✓ **Publique(+)** : Toutes les autres classes ont accès à cet attribut (ou opération).
 - ✓ **Protégé (#)**: Seules la classe elle-même et les classes filles (héritage) ont accès à cet attribut (ou opération).
 - ✓ **Privée (-)** : Seule la classe elle-même a accès à cet attribut (ou opération).

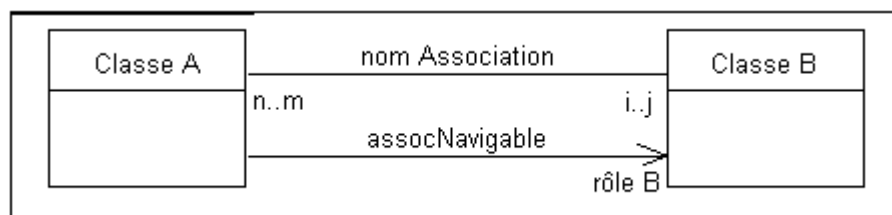


Figure 7 : Diagramme de classes (bases-suite1)

- **Association** : relation sémantique durable entre deux classes, qui décrit un ensemble de liens entre instances. Une association est bidirectionnelle par défaut, sauf si l'on restreint sa navigabilité en ajoutant une flèche.

- **Rôle** : nom donné à une extrémité d'une association ; par extension, manière dont les instances d'une classes voient les instances d'une autre classes au travers d'une association.
- **Multiplicité** : le nombre d'objets (Min...Max) qui peuvent participer à une relation avec un autre objet dans le cadre d'une association. Multiplicités fréquentes :

1 = exactement 1

0...1= optionnel (mais pas multiple)

0...* = * = quelconque

1...*= au moins 1

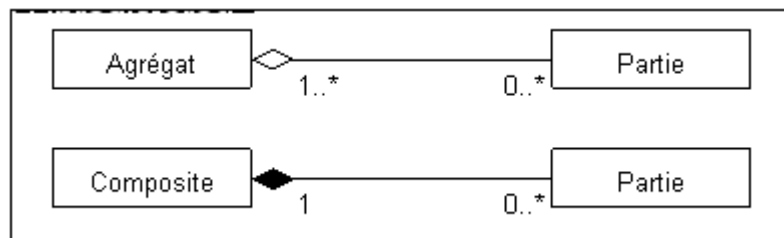


Figure 8 : Diagramme de classes (bases-suite2)

- **Agrégation** : cas particulier d'association non symétrique exprimant une relation de contenance.
- **Composition** : forme forte d'agrégation, dans laquelle les parties ne peuvent appartenir à plusieurs agrégats et où le cycle de vie des parties est subordonné à celui de l'agrégat.

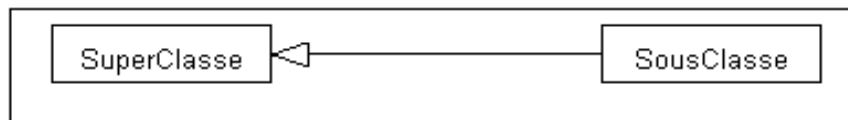


Figure 9 : Diagramme de classes (bases-suite3)

- **Super-classe** : classe générale reliée à d'autre plus spécialisées (sous-classes) par une relation de généralisation.
- **Généralisation** : relation entre « classifieurs » ou les descendants héritent des propriétés de leur parent commun. Ils peuvent néanmoins comprendre chacun des propriétés spécifiques supplémentaires, mais aussi modifier les comportements hérités.

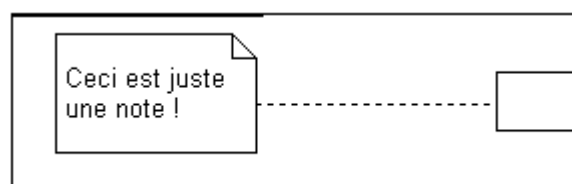


Figure 10 : Diagramme de classes (bases-suite4)

- **Note** : commentaire visible dans le diagramme. Peut être attaché à un élément du modèle (ou plusieurs) par trait pointillé. Utilisable dans tout type de diagramme UML.

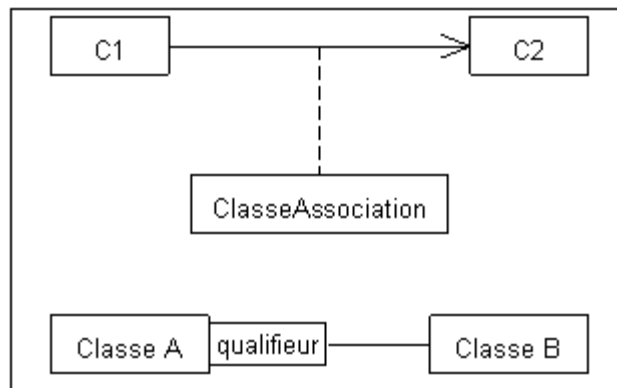


Figure 11: Diagramme de classes (avancé)

- **Classe d'association** : association promue au rang de classe. Elle possède tout à la fois les caractéristiques d'une association et celles d'une classe et peut donc porter des attributs qui prennent des valeurs pour chaque lien entre objets.
- **Qualifieur (ou qualificatif)** : attribut qui permet de « partitionner » l'ensemble des objets en relation avec un objet donné dans le cadre d'une association multiple.

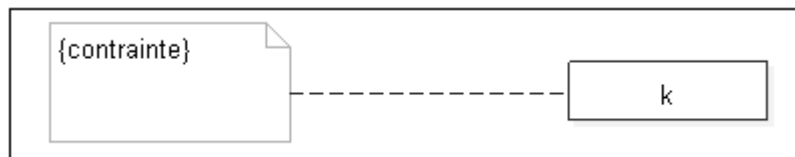


Figure 12 : Diagramme de classes (avancé-suite1)

- **Contrainte** : condition portant sur un ou plusieurs élément(s) du modèle qui doit être vérifiée par les éléments concernés. Elle est notée entre accolades { }, et souvent insérée dans une note graphique (le post-it).

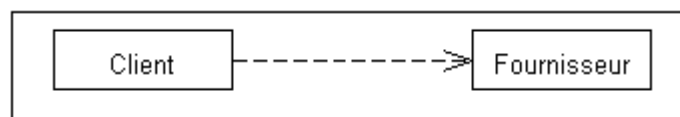


Figure 13 : Diagramme de classes (avancé-suite2)

- **Dépendance** : relation sémantique entre deux éléments, dans laquelle la modification d'un des éléments (l'élément indépendant) peut affecter la sémantique de l'autre élément (l'élément dépendant).

- **Énumération** : est un type de donné UML, possédant un nom, et utilisé pour énumérer un ensemble de littéraux correspondants à toutes les valeurs possibles que peut prendre une expression de ce type.

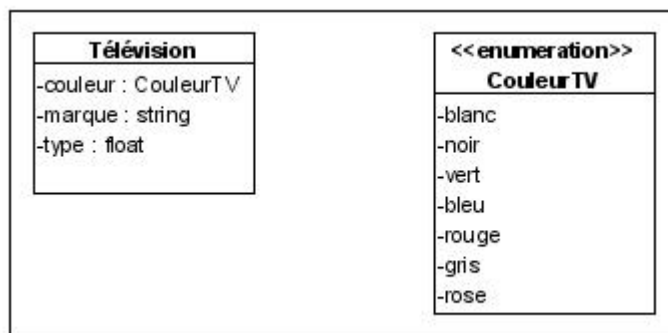


Figure 14 : Diagramme de classes (avancé-suite3)