الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

N° Réf :……

Centre Universitaire de Mila

Institut des sciences et de la technologie **Département de Mathématiques et Informatique**

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

En: - Filière informatique générale

Spécialité sciences et technologies de l'information et de la communication STIC

Développement d'une application web pour le calcul des horaires de prière et la détermination des prévisions météo

Préparé par : Belmenai Nawel
Djoual Sara

Soutenue devant le jury :

Président : Dib Abderrahim	**Grade : MAA, C.U.Mila.**
Examinateur: Hattab Abdelkamel	**Grade: MAA, C.U.Mila.**
Promoteur : Mme F.Benabderrahmane	**Grade : MAA, C.U.Mila.**

Année universitaire: 2013/2014

Remerciements

Dans le cadre de la réalisation de notre projet de master

Nous tenons tout d'abord à remercier ALLAH tout puissant et miséricordieux, qui m'a donné la force et la patience d'accomplir ce modeste travail.

En second lieu, Nous tenons à remercier notre encadreur Mme : Benabderrahmane Fatiha, pour sa présence, son aide et surtout pour ses précieux conseils qui nous ont assistés pour l'accomplissement de notre projet.

Nos remerciements vont aussi aux membres de jury pour avoir accepté d'examiner et d'évaluer notre travail.

Nos remerciements à nos très chers parents, frères, soeurs, amis respectives qui nous ont encouragés, soutenu durant tout notre parcours.

Enfin nous remercions toutes les profs qui enseigner de primaire jusqu'à la dernier année de l'université, et tous les personnes qui de près ou de loin on contribués à l'élaboration de cette étude surtout Mr. Mohamed Amine.

NAWEL & SA

Dédicace

Je dédie ce travail à mes eyeux mon père Foudil et ma mère Oum assaad

A mes très chers soeurs : Zhour, Randa, Fouzia

A mes très chers frères : Moussa, Chamss eddine, Youcef et petit frère Haroun

A mes chers grand pères et grand-mères

A mes tantes et mes oncles

A mes cousins et mes cousines

A tous les membres de la famille Djoual et Bouchaffa

A les étudiants de la CUM surtout G master2 STIC

A tous mes enseignants

A ma binôme et m'amie Nawel

A tous mes amis Lamia b, Wiam dj, Yamina b, Dalila b, Roqa k, Nora b, Wafa r, Wafab , Imane b ,Soumia b,

Mereim b, Nadjet L, Nadjet a, Samira m, Kanza h, Nessrin b,

Ibtissam, Houda,Nadia, Imane, Wassila, Amina, Halima

Sara.

Dédicace

Je dédie ce travail à mon très cher père Abd Elmoumen et ma très cher mère Houria

A mes très chers frères: Saad, Ayoub, Lokmen

A mes très cher marie kamel et leur famille

A mes chers grand pères et grand-mères ‘’Aicha’’

A mes tantes Fatiha, Messouda, Houria, Elghalia

A mes oncles NourEddin, Mouhamed, Oumar

A mes cousins et mes cousines

Leai, Yacin, Meriem

A tous les membres de la famille Belmenai et Djoual

A les étudiants de la CUM surtout M2 stic

A tous mes enseignants

A ma binôme et m’amie Sara

A mes amis Lamia , Wiam dj, Nora b, Imane b, Yamina b, Roqa k, Wafa r, kenza h, fatima b, Amina dj, Dalila b, Wafa b , Soumia b, Samira m, Nessrin b

Nawel.

Résumé

Notre projet s'inscrit dans le contexte des applications basées sur le web qui offrent une technologie importante vis-à-vis l'utilisateur final d'une part, et du concepteur d'application d'autre part. Ce dernier peut profiter des services et applications offerts en les intégrant ou les invoquant afin de réaliser des tâches nécessitant des ressources disponibles sur le web. Citons comme exemple, les applications et services web offerts par la firme Google. C'est dans cette optique que se situe notre projet de stage de fin d'études. Il vise à approfondir nos connaissances informatiques dans le domaine des applications basées sur le Web.

Notre objectif à travers ce travail est de développer une application web qui réponde aux besoins de l'utilisateur. **HoPriMet « Horaire Prière Météo »** est l'application web qu'on propose et qui exploite un nombre de services web, destinés aux simples utilisateurs musulmans du web pour leur offrir deux services à savoir : le calcul des horaires de prière et la détermination des prévisions de la météo, étant donné que la majorité des musulmans dans des pays étrangers ou des États non-musulmans estiment qu'il est difficile de savoir les horaires de prière. **HoPriMet** garantit ainsi la disponibilité de l'information chez le musulman pratiquant son culte en lui offrant la possibilité de pointer sur une carte n'importe quelle région du monde, d'en calculer les horaires de prière à la date de navigation, d'en afficher le calendrier annuel des horaires de prière et enfin d'afficher les prévisions météo de la région pointée sur la carte. Nous exploitons, à cet effet, les services web Google Maps et Google Weather, afin de pouvoir profiter des ressources disponibles sur le web.

Abstract

This project is situated in the context of web-based applications that provide an important technology on the one hand to the end user and the on the other to the application designer. In fact, the latter can benefit from the web services and applications offered by integrating or invoking them to perform tasks requiring resources available on the web. As an example of such services, applications and web services offered by the company Google. It is in this context that our internship project graduation is located. It aims to deepen our IT knowledge in the field of web-based applications.

Our goal through this work is to develop a web application that meets the needs of the user. HoPriMet is the web application that offers and operates a number of web services, for the simple Muslim web users to provide them two services namely the calculation of prayer times and the determination of weather forecasts, as given that the majority of Muslims in

foreign countries or non- Muslim states find it difficult to know the prayer times. HoPriMet guarantees the availability of information to the practicing Muslim worship by offering them the opportunity to point on a map any region of the world, to calculate the prayer times at the time of sailing, to view the annual calendar of prayer times and finally display weather forecasts pointed on the map region. We use for this purpose web services Google Maps and Google weather, to take advantage of resources available on the web.

Sommaire :

Introduction générale : 1

Plan du mémoire : 2

Chapitre 1 : Notions et concepts des applications basées sur le web

Introduction : 3

1. Application web : 3
   1.1. Architecture des applications basées sur le Web : 4
      1.1.1. Le réseau Internet et ses protocoles : 4
      1.1.2. Composants du client Web : 4
      1.1.3. Composants du serveur web: 7
   1.2. Les avantages d'une application web : 8
2. les services web : 9
   2.1. Définitions : 9
   2.2. Caractéristiques des services Web : 10
   2.3. Architecture des services Web : 10
   2.4. Standards des services Web 13
      2.4.1. SOAP 13
      2.4.2. UDDI 14
      2.4.3. WSDL 16

   Conclusion : 17

Chapitre 2: les services web Google Maps et Google Weather

Introduction : 19

1. Cartographie dynamique sur le web : 19
2. Le service web Google Maps : 20
   2.1. Définition du service web Google Maps : 20
   2.2. La structure de la base de données du Google Maps : 21
   2.3. Les Coordonnées dans Google Maps : 22
   2.4. API de Google Maps (Application Programme Interface) : 23
      2.4.1. Les APIs Google Maps : 24
      2.4.2. Les APIs Google Maps Web Services: 24
   2.5. Navigation dans Google Maps: 25
   2.6. Récupération des coordonnées d'une adresse avec google Maps : 26
   2.7. Les traitements sur carte offerts par L'API Google Maps JavaScript version 3 : 27

3. Le service web Google weather: 27
3.1. Qu'est-ce que Google Weather ? 27
Conclusion : 29
Introduction : 30

Chapitre 3: la démarche d'analyse et de conception

1. Présentation de la démarche : 30
2. La démarche de développement : 30
2.1. L'Etude préliminaire : 31
2.2. Le modèle des besoins : 31
2.3. La spécification détaillée des besoins : elle consiste à 31
2.4. Le modèle de domaine : 31
2.5. Le diagramme des classes participantes : 32
2.6. La conception détaillée du système : 32
3. Spécificités UML pour les applications Web : 33
Conclusion : 35

Chapitre 4: Analyse et conception de l'application

Introduction 36
I. Analyse et conception de l'application : 36
1. Etude préliminaire : 36
1.1. Objectif de notre projet : 36
1.2. Notions astronomiques et calculs mathématiques : 36
2. Le modèle des besoins : 41
2.1. Le diagramme de cas d'utilisation : 42
2.2. La description détaillée des Cas d'Utilisations par les fiches descriptives: 42
2.3. Description de Cas d'Utilisation par les diagrammes d'activités : 46
3. La Spécification détaillée des besoins : 51
3.1. Description des Cas d'Utilisation par les diagrammes de séquence : 51
4. Conception détaillée du système: 55
4.1. Diagramme des classes de conception : 57
4.2. Diagramme de navigation: 58
5. Architecture d'application : 58
Conclusion : 59

Chapitre 5: Implémentation

Introduction: 60
1. Langages de programmation: 60
1.1. Le Langage HTML : 60
1.2. Le Langage PHP : 60
1.3. Le Langage JavaScript: 61
2. Outils de développement: 61
2.1. Wamp Server 61
2.2. Un navigateur Web : 62
3. Implémentation: 63
3.1. Le service « Calcul des Horaires de prière » : 63
Le calcul des horaires : 64
4. Les interfaces de l'application: 66
4.1. Page d'Accueil : 66
4.2. Obtenir Carte : 66
4.3. Résultat de calcul du jour actuel : 68
4.4. Résultat de calcul de l'année actuelle : 68
4.5. Déterminer les prévisions météo : 69
4.6. Résultat des prévisions météo : 69
Conclusion : 70
Conclusion générale 71

Liste des figures:

Figure 1:« Transfert des données à travers la pile de protocoles d'Internet »..... 4
Figure 2: « Mode de fonctionnement des applications Web »..... 5
Figure 3: « Mode de fonctionnement des applications Web 2.0 »..... 6
Figure 4 : « le fonctionnement d'une architecture de services Web »..... 11
Figure 5:« Architecture des services Web »..... 12
Figure 6:« Structure du message SOAP »..... 14
Figure 7 : « les trois facettes de l'annuaire UDDI [Hubert et al., 2003] »..... 15
Figure 8: « Structure d'une description WSDL »..... 17
Figure 9: « Organisation d'une application de cartographie numérique »..... 20
Figure 10: « La structure de stockage des images (données) de la carte Google Maps »..... 21
Figure 11: « Référencement par Pixel »..... 22
Figure 12: « Référencement par Carrelage »..... 23
Figure 13: « page d'accueil de google maps »..... 25
Figure 14: « géo localisation d'un lieu grâce à l'API Google Maps»..... 26
Figure 15: « Image fournie par le blog Google LatLong (Ouragan Alex 2010) »..... 29
Figure 16: stéréotype page «contrôle»..... 33
Figure 17: stéréotype page client «interface»..... 34
Figure 18: stéréotype page formulaire «form»..... 34
Figure 19: stéréotype objet script « script object»..... 35
Figure 20: « le diagramme de cas d'utilisation »..... 42
Figure 21:: « Diagramme d'activité du cas d'utilisation «Calculer et afficher les horaires de prière »..... 47
Figure 22: Diagramme d'activité du cas d'utilisation «Déterminer et afficher les prévisions météo »..... 48
Figure 23: «Diagramme d'activité du cas d'utilisation « Calculer les mesures astronomiques et les fonctions mathématiques »..... 49
Figure 24: « Diagramme d'activité du cas d'utilisation Obtenir carte »..... 50
Figure 25 : « Diagramme de séquence du cas d'utilisation Calculer et afficher les horaires de prière »..... 51
Figure 26:« Diagramme de système du cas d'utilisation Déterminer et afficher les prévisions météo »..... 52
Figure 27:« Diagramme de séquence système du cas d'utilisation »..... 53
Figure 28: « Diagramme de séquence système du cas d'utilisation »..... 54

Figure 29: Pages «contrôle» 55
Figure 30: Pages «interface» 56
Figure 31: Formulaires «form» 56
Figure 32: « Diagramme des classes de conception » 57
Figure 33: « Diagramme de navigation» 58
Figure 34: « Diagramme de déploiement » 59
Figure 35: « Page d'Accueil » 66
Figure 36: « Obtenir Carte » 67
Figure 37: «Résultat de calcul du jour actuel » 68
Figure 38: «Résultat de calcul de l'année actuelle » 68
Figure 39: « Déterminer les prévisions météo » 69
Figure 40: « Résultat des prévisions météo » 69

Liste des tableaux:

Tableau 1 : « les huit points des temps durant le jour » 37
Tableau 2: «Les formules pour calculer les horaires de prière » 40
Tableau 3: «les conventions de calcul des horaires de Fajr et Isha dans les régions du monde» 41
Tableau 4 : « Description textuelle du cas d'utilisation 43
Tableau 5 : « Description textuelle du cas d'utilisation 44
Tableau 6 : « Description textuelle du cas d'utilisation Calculer les mesures astronomiques et les fonctions mathématiques » 45
Tableau 7« Description textuelle du cas d'utilisation Obtenir carte » 46

Introduction générale

Introduction générale :

Dans un monde actif et continuellement évolutif, la motivation d'avoir des moyens performants et efficaces de communication et d'échange d'informations devient de plus en plus fondamentale. Cette motivation donne naissance à une révolution favorisant le travail à distance et l'accès aux besoins en temps réduit à l'aide d'Internet qui a bouleversé les habitudes de travail dans de nombreux métiers mais également chez les particuliers.

Les services Web en particulier, (en anglais Web services) représentent un mécanisme de communication entre applications distantes à travers le réseau Internet, telle que ces applications peuvent être vues comme un ensemble de services métiers, structurés et correctement décrits, dialoguant selon un standard international plutôt qu'un ensemble d'objets et de méthodes entremêlés.

Les services Web sont considérés comme une fonction accessible à distance par une requête qui exprime les besoins d'un utilisateur ayant comme entrée un ensemble de concepts indiqués dans une ontologie spécifique et donne comme sortie un autre ensemble de concepts.

Les services Web, tels qu'ils sont conçus, peuvent également se voir employer dans le cadre d'applications distribuées. En effet, en exploitant les standards ouverts du Web et en assurant un couplage faible des composants, la technologie service Web présente une approche flexible et universelle pour l'interopérabilité de systèmes hétérogènes.

Dans ce contexte s'inscrivent les applications basées sur le web qui offrent une technologie importante vis-à-vis l'utilisateur final d'une part, et du concepteur d'application d'autre part. Ce dernier peut profiter des services et applications offerts en les intégrant ou les invoquant afin de réaliser des tâches nécessitant des ressources disponibles sur le web. Citons comme exemple, les applications et services web offerts par la firme Google, tels que Google Maps et Google Weather.

C'est dans cette optique que se situe notre projet de stage de fin d'études. Il vise à approfondir nos connaissances informatiques dans le domaine des applications basées sur le Web. Notre objectif à travers ce travail est de développer une application web qui réponde aux besoins de l'utilisateur. **HoPriMet** est l'application web qu'on propose et qui exploite un nombre de services web, destinés aux simples utilisateurs musulmans du web pour leur offrir deux services à savoir : le calcul des horaires de prière et la détermination des prévisions de la météo, étant donné que la majorité des musulmans dans des pays étrangers ou des États non-musulmans estiment qu'il est difficile de savoir les horaires de prière. **HoPriMet** garantit ainsi

la disponibilité de l'information chez le musulman pratiquant son culte en lui offrant la possibilité de:

- Pointer sur une carte n'importe quelle région du monde.
- Calculer les horaires de prière de cette région à la date de navigation
- Afficher le calendrier annuel des horaires de prière de cette région
- Prévoir une alarme sonore qui prévient l'internaute du moment de la prière
- Afficher les prévisions météo de la région pointée sur la carte.

Nous exploitons, à cet effet, les services web Google Maps et Google Weather, afin de pouvoir profiter des ressources disponibles sur le web

Plan du mémoire :

Ce manuscrit est composé de quatre chapitres et une conclusion générale qui sont organisés comme suit :

- ✓ Le premier chapitre aborde les concepts liés aux applications basées sur le web mais aussi les standards des services web.
- ✓ Le deuxième chapitre donne un bref aperçu sur le service web Google Maps, et ce qui le lie à l'application du calcul des horaires de prière.
- ✓ Le chapitre 3 présente une démarche simplifiée adaptée aux applications web issue de [7].
- ✓ Le chapitre 4 constitue l'essentiel de notre travail il détaille l'analyse et la conception de notre système, Nous expliquons également tous les éléments théoriques et les mesures astronomiques mis en jeu dans le calcul des horaires de prière. Nous modélisons par des diagrammes UML ces différents composants et fonctionnalités.
- ✓ Finalement, nous clôturons notre mémoire par une conclusion qui offre une synthèse du travail réalisé.

Chapitre 1 : Notions et concepts des applications basées sur le web

Introduction :

Actuellement nous avons une large propagation et utilisation des réseaux informatiques particulièrement l'Internet. Ce réseau est un environnement sur lequel devient possible de faire fonctionner des applications sur des machines distantes, et leur permettre de communiquer entre elles. Ces applications distantes, selon leur type, peuvent répondre à plusieurs problématiques et besoins. Dans ce contexte s'inscrivent les applications basées sur le web qui offrent une technologie importante vis-à-vis l'utilisateur final d'une part, et du concepteur d'application d'autre part. Ce dernier peut profiter des services et des applications offerts en les intégrant ou les invoquant afin de réaliser des tâches nécessitant des ressources disponibles sur le web. Citons comme exemple, les applications et services web offerts par la firme Google, tels que Google Maps et Google Weather.

Dans le cadre de notre projet de fin d'étude, nous proposons une application webpour le calcul des horaires de prière et la détermination des prévisions de météo. Nous exploitons, à cet effet, les services web Google Maps et Google Weather, afin de pouvoir profiter des ressources disponibles sur le web.

Ce chapitre présente quelques notions concernant les applications web ainsi que les services web.

1. Application web :

Une application Web est un logiciel applicatif manipulable grâce à un navigateur Web, de la même manière que l'accès à un site Web. Elle est généralement hébergée sur un serveur et se manipule en actionnant des composants d'interface graphique (widgets) à l'aide d'un navigateur Web, via un réseau informatique (Internet, intranet, réseau local, etc.). Ainsi, le client Web doit donc pouvoir s'exécuter dans un environnement logiciel déjà présent sur les postes des utilisateurs (la Plateforme d'exécution). Une architecture d'application Web peut être divisée en modules ou composants, dont les principaux sont: la présentation, la logique métier, l'interaction et la gestion des données.

1.1. Architecture des applications basées sur le Web :[1]

1.1.1. Le réseau Internet et ses protocoles :

Le Web repose sur le réseau Internet et comme tous les réseaux informatiques, celui-ci repose sur des couches de protocoles de communication. Le paquet est l'unité de base de la transmission de données sur Internet.

La figure qui suit présente la couche de protocoles pour Internet dont le couple TCP/IP est la fondation :

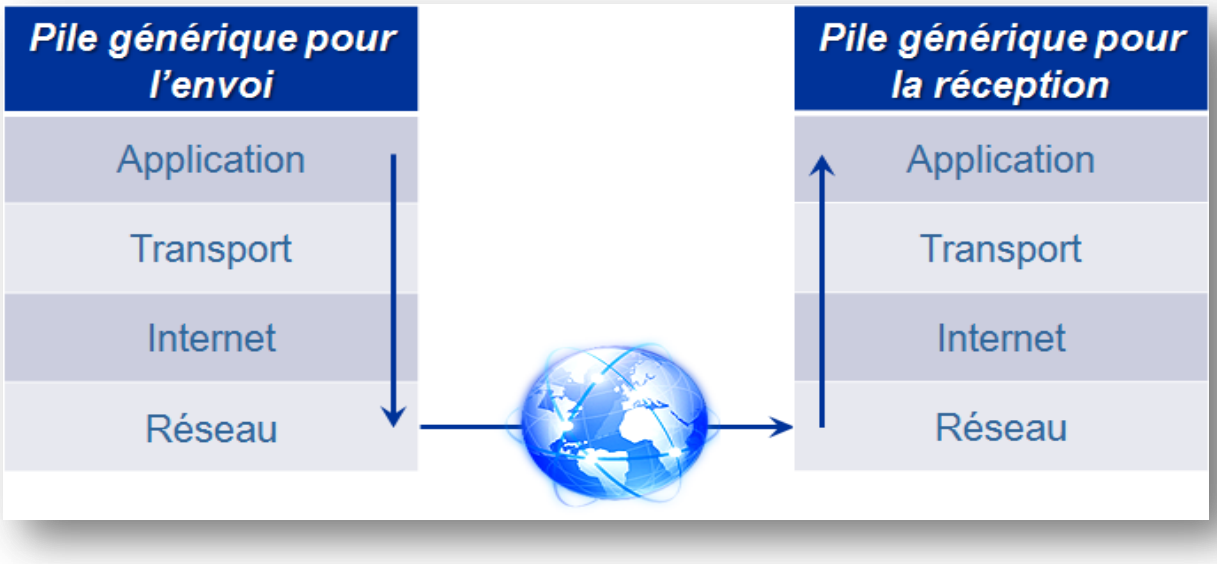

Figure 1:« Transfert des données à travers la pile de protocoles d'Internet »

1.1.2. Composants du client Web :

a. Le navigateur et les langages de description des pages Web:

Dans les architectures citées précédemment, le navigateur est une application cliente. Il permet d'envoyer des requêtes http au serveur Web et d'en interpréter la réponse. Les navigateurs sont aujourd'hui capables de travailler également avec le protocole FTP et d'afficher d'autres formats tels que XML.

Il existe plusieurs méthodes http pour envoyer des requêtes au serveur. Les plus répandues sont GET, HEAD et POST. GET permet de demander le téléchargement du contenu d'un document, HEAD permet de n'en récupérer que l'en-tête et POST permet d'envoyer des informations au serveur pour traitement. Lorsque l'utilisateur saisit une adresse ou clique sur

un lien hypertexte, le navigateur envoie une requête GET au serveur qui ne comprend qu'un en-tête. Les requêtes POST ont un corps de message qui comporte l'ensemble des informations saisies dans un formulaire, alors qu'avec GET, ces informations sont transmises en ajoutant des paramètres à l'adresse.

HTML est un langage qui permet de décrire le contenu et la structure d'une page Web. Il est composé d'environ 50 instructions de mise en forme. La version 5 prévue pour 2012 permettra d'intégrer et de standardiser toutes les innovations développées pour compléter la version actuelle qui ne permet pas de représenter des graphiques, de représenter une animation ou d'interagir avec l'utilisateur. Dans un document HTML, la structure de la page peut être représentée par un arbre appelé DOM (Document Object Model).

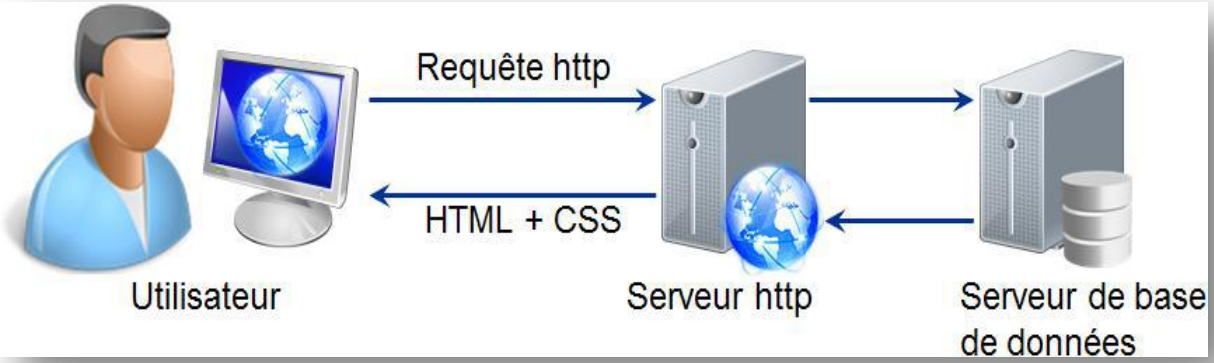

Figure 2: « Mode de fonctionnement des applications Web »

XHTML est une extension d'HTML basée sur le langage structuré de description de données XML (eXtensible Markup Language).

CSS (Cascading Style Sheets) est un complément d'HTML dont la version 2 actuelle a été développée en 1998. Une feuille de style (style sheet) est un ensemble de règles à appliquer aux éléments HTML. L'utilisation de CSS permet de séparer les responsabilités dans la présentation des données à l'utilisateur. HTML peut ainsi être responsable du contenu de la page affichée alors que CSS sera responsable de la mise en page. En modifiant seulement le CSS, il est ainsi possible de modifier l'apparence d'une application Web. La future version CSS3 est prévue pour 2012.

b. Les scripts

HTML n'offrant pas de comportement dynamique aux pages Web, les éditeurs ont créé des langages de scripts et des extensions aux navigateurs.

JavaScript est un langage de scripts inventé en 1995 pour le navigateur Netscape. Il permet de manipuler les objets de l'arbre DOM et de gérer la réaction à des évènements générés par les objets de la page. Il permet en fait de modifier la page HTML sans envoyer de requête http. Bien qu'il existe une norme ECMA Script gérée par l'organisme de spécifications ECMA, chaque navigateur a développé son propre interpréteur, ce qui pose des problèmes de portabilité des applications Web.

Afin de compléter JavaScript pour permettre de modifier la page HTML affichée en communiquant avec le serveur http, sans recharger toute la page, Microsoft a développé en 2002 un nouvel objet JavaScript : XML http Request. Aujourd'hui la plupart des navigateurs intègre cet objet. Cela permet de communiquer de manière asynchrone avec le serveur, ce que ne permet pas actuellement HTML. Ce nouvel outil a été la base du développement du Web 2.0 et sera intégré à la future norme HTML 5.

AJAX (Asynchronous JavaScript And XML) est un terme inventé en 2005 par Jesse James Garrett. Il ne s'agit pas d'une nouvelle technologie mais d'une façon d'utiliser conjointement les technologies XHTML, JavaScript et CSS. Les applications Web développées selon le paradigme Ajax utilisent massivement les requêtes GET pour mettre à jour l'interface graphique.

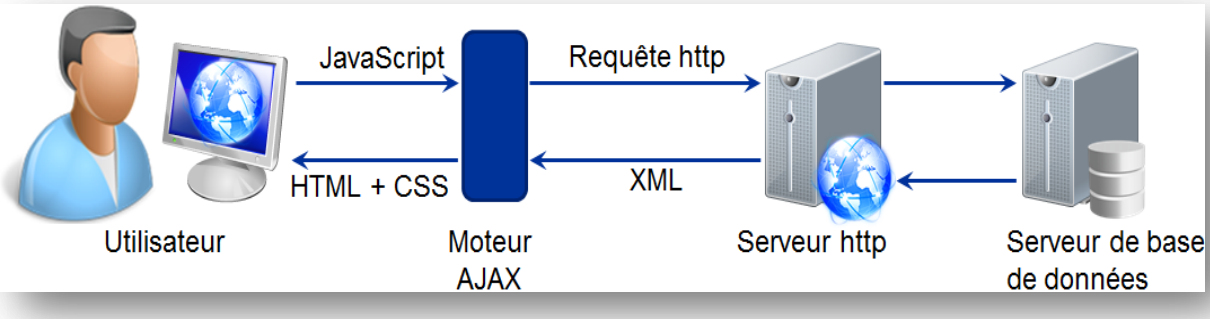

Figure 3: « Mode de fonctionnement des applications Web 2.0 »

1.1.3. Composants du serveur web:

a. Serveurs Web et serveurs d'application

Comme évoqué précédemment, le navigateur et le serveur communiquent en utilisant le protocole http. Les serveurs ne sont pas obligés d'implémenter toutes les méthodes http, seulement GET et HEAD. Bien qu'optionnelle, peu de serveurs actuels n'implémentent pas la méthode POST.

Sur Internet le navigateur et le serveur http communiquent rarement directement. Le plus souvent un serveur intermédiaire est présent : le serveur proxy. Les requêtes à destination du serveur sont interceptées par le serveur proxy qui peut leur faire subir un traitement, avant de les retransmettre au serveur. Ce principe est également appliqué aux réponses. Le serveur proxy peut servir de cache pour moins solliciter le serveur http. Il est possible de faire agir plusieurs serveurs proxy en cascade.

La seule fonction du serveur Web étant d'envoyer le contenu des fichiers au client, des extensions peuvent y être ajoutées, permettant de faire appel à des services pour générer dynamiquement les informations à transmettre. Ils traitent les requêtes http que le serveur http leur a fait suivre, interprètent et exécutent le code de l'application, puis génèrent une réponse qu'ils renvoient au serveur http qui l'enverra au navigateur de l'utilisateur. Si ces services fonctionnent indépendamment du serveur http, ils sont appelés serveurs d'applications.
L'extension CGI (Common Gateway Interface) permet l'exécution d'un programme extérieur appelé « gateway » dont la sortie standard d'affichage sera renvoyée au client par le serveur http. Le langage utilisé pour le développement du programme n'a pas d'importance. Cela peut être du C++, du Perl ou même Java. Il faut seulement que le programme puisse être exécuté sur la machine hébergeant le serveur http.

De même que CGI, l'extension Java Servlet développée par Sun permet d'intercepter les requêtes, de générer les réponses pour exécuter des applications Java. La différence est que la machine virtuelle Java peut être lancée sur un serveur différent du serveur http. Cette extension permet de conserver des informations entre les requêtes.
JSP (Java Server Pages) est un langage qui permet d'insérer des blocs de script basés sur Java dans un contenu HTML. Les pages JSP sont interprétées et transformées en Servlet par un serveur d'application pour être exécutées.

ASP (Active Server Pages) est le concurrent de JSP développé par Microsoft. Etant basé sur le langage VBScript, ASP est devenu très populaire dans le milieu des développeurs Visual-Basic dont il est très proche. Tout comme JSP, ASP permet d'insérer des blocs de script dans du contenu HTML. Pour combler les lacunes d'ASP, Microsoft a développé la Plate-forme « .NET » qui permet d'utiliser les autres langages de développement du monde Microsoft tel que C# pour la génération dynamique de contenu HTML. L'exécution de code ASP ou .Net est limitée aux serveurs Windows.

PHP (acronyme récursif pour « PHP: Hyper text Preprocessor ») est un langage proche de Perl et des scripts Shell, ce qui a fait son succès auprès de la communauté du monde Unix. C'est ce langage que nous avons utilisé pour implémenter notre système.

b. Serveurs de données

Les données étant principalement gérées par des serveurs dédiés, les langages cités précédemment offrent des moyens d'interagir avec eux. Les systèmes de gestion de bases de données permettent d'interroger les données et de les mettre à jour. Le langage le plus répandu est SQL (Structured Query Language) pour les bases de données relationnelles.

Les services Web sont des applications Web dont le but est de fournir des données selon une structure prédéfinie et des services à une autre application en utilisant les protocoles standards d'Internet.

1.2. Les avantages d'une application web :

L'application est placée sur un serveur et les utilisateurs y accèdent par un simple navigateur : il suffit d'une connexion à Internet.

Il en découle les avantages suivants :

- il n'est plus nécessaire d'installer un logiciel sur son poste, ce qui diminue en grande partie les frais de maintenance et élimine certaines incompatibilités.
- une application bien conçue peut être utilisée par différents types de terminaux (ordinateurs, la ptops, tablettes, smartphones).
- il n'existe plus de contrainte géographique et on peut envisager un accès pour les utilisateurs nomades en toute sécurité grâce au cryptage des données à travers d'Internet.
- des centaines de personnes réparties dans des lieux différents travaillent simultanément sur ce type d'applications car elles sont conçues pour supporter une grande charge qui est souvent répartie sur plusieurs serveurs.

- certaines parties de ces applications sont ouvertes aux clients et fournisseurs et facilitent ainsi les échanges d'informations entre les partenaires (B2B).
- ces applications sont capables de gérer des données multimédia (textes, photos, vidéos) et les utilisateurs sont à l'aise car habitués à utiliser un navigateur dans le cadre privé.
- pour mettre à jour l'application, il suffit de modifier l'application sur le serveur et tous les postes accèdent instantanément à la nouvelle version.

En résumé, cette technologie facilite la mise à disposition d'applications au sein de l'entreprise, de ses filiales ou pour les utilisateurs itinérants. Elle permet souvent de diminuer les coûts et d'améliorer la rapidité d'accès à l'information.

2. les services web :

Les Services Web sont un type d'architecture reposant sur les standards d'Internet. De ce fait, il s'agit d'une architecture orientée service permettant à des applications de communiquer entre elles directement sans se préoccuper des technologies sur lesquelles elles sont implémentées. Les Services Web se présentent comme étant la solution pour répondre à l'interopérabilité des systèmes.

Il existe trois grands standards dans les Web Service : REST, XML-RPC et SOAP

Parmi ces trois méthodes nous avons d'un côté XML-RPC et SOAP qui sont basés sur des technologies XML et qui utilisent le protocole HTTP, mais qui peuvent s'appuyer sur d'autres protocoles de transport. REST à l'inverse est le mode natif de HTTP.

2.1. Définitions :

Un service web est une agrégation de fonctionnalités publiées pour être utilisées. Il utilise Internet comme conduit pour réaliser une tâche. Il est semblable à un processus métier virtuel qui définit des interactions au niveau application. [Justin, 2002]

Les services web sont la nouvelle vague des applications Web. Ce sont des applications modulaires, auto-contenues et auto-descriptives qui peuvent être publiées, localisées et invoquées de puis le web. Les services web effectuent des actions allant de simples requêtes à des processus complexes. Une fois qu'un service Web est déployé, d'autres applications peuvent le découvrir et l'invoquer IBM [Colan, 2003].

En d'autres mots, les services Web sont des applications auto descriptives, modulaires et

faiblement couplées fournissant un modèle simple de programmation et de déploiement d'applications. Ils reposent principalement sur des technologies basées sur XML pour la structure et le contenu de messages échangés entre services(SOAP), pour la description des services (WSDL) et pour la découverte des services (UDDI).

2.2. Caractéristiques des services Web : [2]

- **basé sur le web**: les Web services sont basés sur les protocoles et les langages du Web, en particulier http et XML.
- **auto-descriptifs, auto-contenus**: le cadre des services Web contient en lui-même toutes les informations nécessaires à l'utilisation des applications, sous la forme de trois fonctions : trouver, décrire et exécuter.
- **Modulaires**: les services Web fonctionnent de manière modulaire et non pas intégrée.

Cela signifie qu'au lieu d'intégrer dans une seule application globale toutes les fonctionnalités, on crée (ou on récupère) plusieurs applications spécifiques qu'on fait inter-opérer entre elles, et qui remplissent chacune une de ces fonctionnalités. Une fonctionnalité développée sous forme de Web services peut dorénavant être réutilisée et recombinée à une suite d'autres fonctionnalités pour composer une nouvelle application.

2.3. Architecture des services Web :

La figure suivante résume l'architecture des services Web :

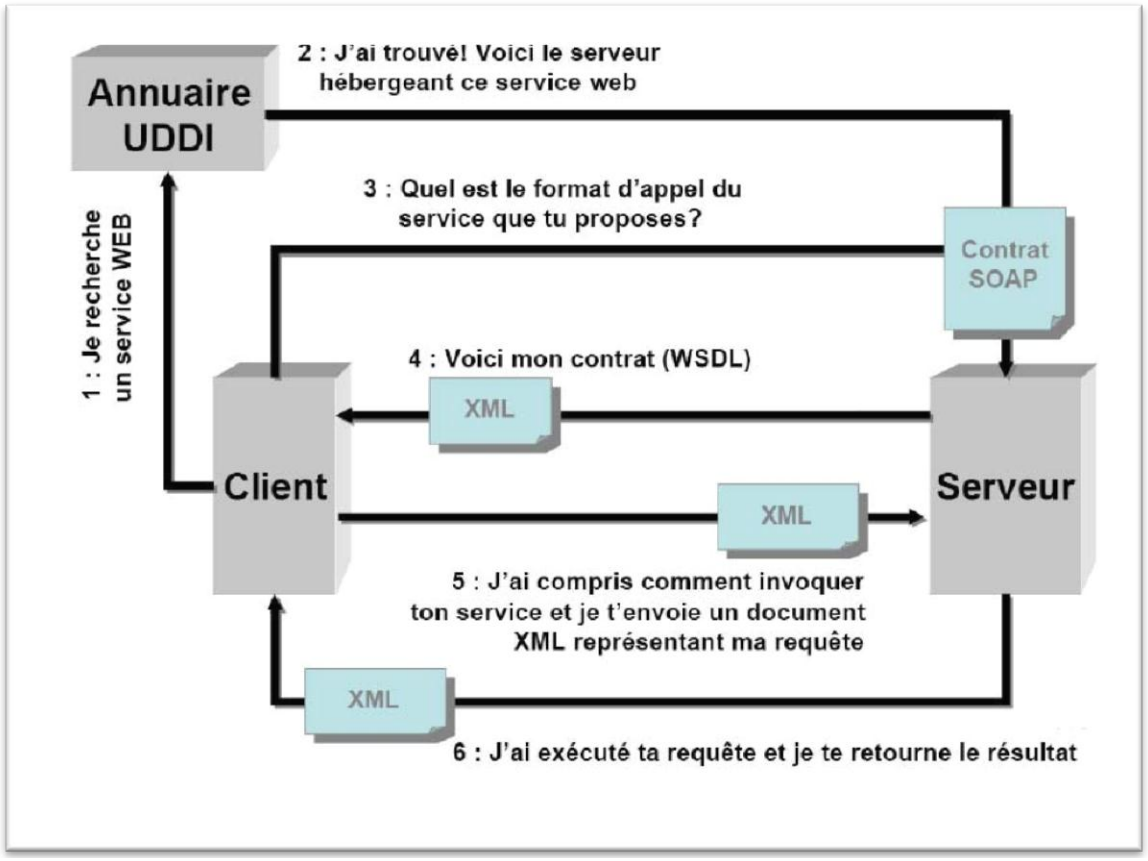

Figure 4 : « le fonctionnement d'une architecture de services Web »

1. Le client envoie une requête à l'annuaire de Service pour trouver le service Web dont il a besoin.
2. L'annuaire cherche pour le client, trouve le service Web approprié et renvoie une réponse au client en lui indiquant quel serveur détient ce qu'il recherche.
3. Le client envoie une deuxième requête au serveur pour obtenir le contrat de normalisation de ses données.

Les technologies utilisées par les services Web sont HTTP, WSDL, XML- RPC, SOAP et UDDI. L'architecture standard d'un service Web est organisée en plusieurs couches.

Chacune d'elles répond à des préoccupations fonctionnelles différentes telles que la publication, la description, la messagerie et le transport. Comme illustré sur la FigureI.2 [Hubertetal.,2003].

Les différentes couches de l'architecture d'un service Web s'interfacent avec des standards, comme suit:

- **La couche de publication** repose sur le protocole UDDI (Universal Description, Discovery and Integration), qui assure le regroupement, le stockage et la diffusion des descriptions des services Web.

- **La couche description :** est prise en charge par le langage WSDL (Web Service Description Language) [Christensen et al., 2001], qui décrit les fonctionnalités fournies par le service Web, les messages reçus et envoyés pour chaque fonctionnalité, ainsi que le protocole utilisé pour la communication.
- **La couche communication :** propose différents mécanismes liés à l'acheminement des messages (format de communication des messages, adressage, routage...etc.). Et utilise des protocoles reposants sur le langage XML. Actuellement, SOAP (Simple Object Access Protocol) est le protocole le plus utilisé pour cette couche.
- **La couche transport:** le protocole le plus utilisé dans cette couche est l'HTTP (Hyper Text Transfert Protocol) .Cependant, d'autres protocoles peuvent être utilisés, tels que le SMTP (Simple Mail Transfer Protocol) ou le FTP (File Transfer Protocol), permettant ainsi aux services Web de rester indépendants du mode de transport utilisé.

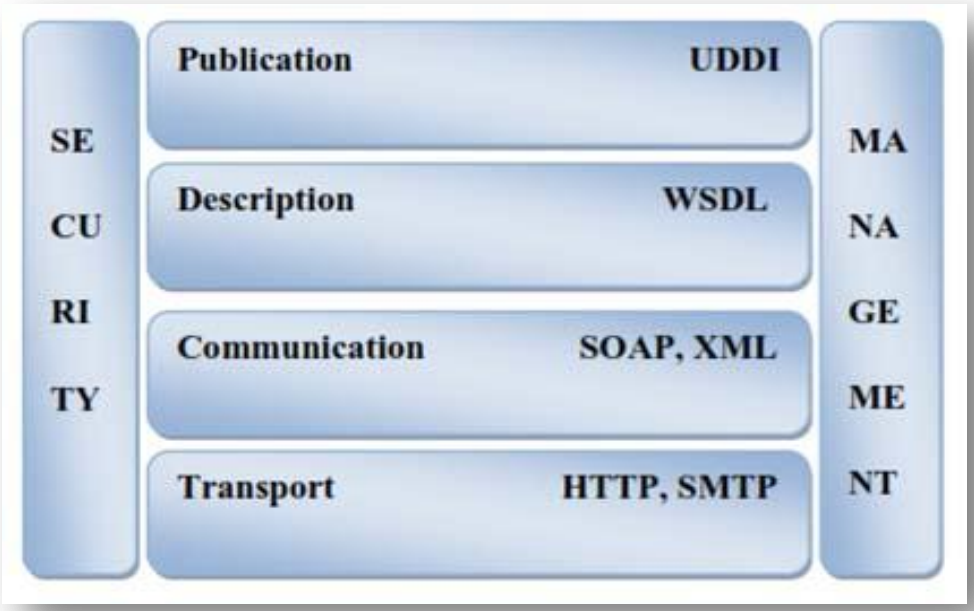

Figure 5:« Architecture des services Web »

2.4. Standards des services Web :[2]

Nous présentons dans cette section de manière succincte les trois standards de base, à savoir, SOAP, WSDL et UDDI. Ces trois standards sont basés sur le langage XML.

2.4.1. SOAP

SOAP est un standard du Consortium W3C définissant un protocole de transmission de messages permettant la normalisation des échanges de données. Il présente un ensemble des règles pour structurer des messages, qui peuvent être utilisées dans de simples transmissions unidirectionnelles, mais il est particulièrement utile pour exécuter des dialogues requête réponse, RPC (Remote Procédure Call) en utilisant http comme protocole de communication, mais aussi avec SMTP (Simple Mail Transport Protocol) ?

SOAP assure l'interopérabilité entre composants tout en restant indépendant des systèmes d'exploitation et des langages de programmation, donc, théoriquement, les clients et serveurs de ces dialogues peuvent fonctionner sur n'importe quelle plate-forme et être écrits dans n'importe quel langage à partir du moment où ils peuvent formuler et comprendre des messages SOAP. Il représente donc un composant de base pour développer des applications distribuées, qui exploitent des fonctionnalités publiées comme services par des intranets sous Internet.

SOAP utilise principalement les deux standards HTTP et XML:

- HTTP comme protocole de transport des messages SOAP. Il constitue un bon moyen de transport en raison de sa popularité sur le web.
- XML pour structurer les requêtes et les réponses, indiquer les paramètres des méthodes, les valeurs de retours, et les éventuelles erreurs de traitements.

➢ **Structure du message SOAP :**

Les messages sont composés de deux parties, l'en-tête de protocole de transport et l'enveloppe SOAP.

Figure 6:« Structure du message SOAP »

1. **L'en-tête du protocole de transport:** qui dépend de protocole de transport utilisé, par exemple si le protocole HTTP est utilisé, l'en-tête contient :
   - La version de protocole http utilisée.
   - La date de génération de message SOAP.
   - Le type d'encodage du contenu (généralement de type XML).
2. **L'enveloppe SOAP:** la partie principale d'un message SOAP est l'enveloppe (symbolisée par la balise enveloppe), cette dernière est subdivisée en deux sous-parties: la partie en-tête (Header) et la partie corps du message (Body).

2.4.2. UDDI

Un service d'annuaire UDDI est un service Web qui gère les méta-données des services, l'information sur les fournisseurs de services et les implémentations des services. Afin de trouver un service Web, il est possible d'utiliser un annuaire UDDI en précisant des exigences concernant le service requis. On cherche le service par son nom et/ou par des mots clés. Consultation de l'annuaire

L'annuaire UDDI se concentre sur le processus de découverte de l'architecture orientée services (SOA), et utilise des technologies standards telles que XML, SOAP et WSDL qui permettent de simplifier la collaboration entre partenaires dans le cadre des échanges commerciaux. L'accès au référentiel s'effectue de différentes manières.

Les informations sur un service publié dans un annuaire UDDI se présentent sous trois facettes comme illustre dans la figure 7 [Hubert et al, 2003] :

- Les pages blanches comprennent la liste des entreprises ainsi que des informations associées à ces dernières (coordonnées, description de l'entreprise, identifiants...).
- Les pages jaunes recensent les services Web de chacune des entreprises sous le standard WSDL.
- Les pages vertes fournissent des informations techniques précises sur les services fournis

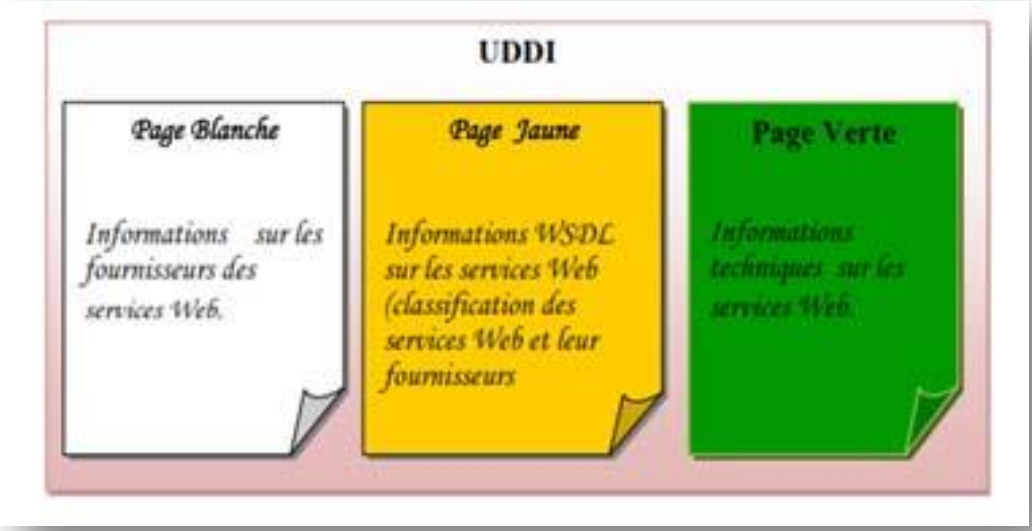

Figure 7 : « les trois facettes de l'annuaire UDDI [Hubert et al., 2003] »

Un annuaire UDDI est conçu pour être interrogé par des messages SOAP et afin de pouvoir stocker et fournir des informations permettant l'accès aux documents WSDL (Web Services Description Language) décrivant les protocoles et les formats de messages nécessaires pour interagir avec les services Web répertoriés dans l'annuaire. Les outils de recherche disponibles sont basés sur des mots-clés et ne prennent pas en considération les relations entre les services Web et les caractéristiques sémantiques de chaque service Web, forçant l'utilisateur à recommencer la recherche depuis le début en utilisant de nouveaux termes clés.

Les spécifications UDDI incluent notamment :

- ✓ Des APISOAP qui permettent l'interrogation et la publication d'informations,
- ✓ La représentation XML du modèle de données de l'annuaire et des formats de message

SOAP.

- ✓ Des définitions de l'interface WSDL de SOAP.
- ✓ Des définitions d'APIs de différents modèles techniques qui facilitent le travail des systèmes d'identification et de catégorisation des enregistrements UDDI.

2.4.3. WSDL

WSDL [Christensenetal.,2001], est un standard du W3C, qui permet de définir une syntaxe XML pour décrire les méthodes et les paramètres des services Web invocables par le biais de messages au format SOAP. Il permet de définir qu'est-ce qu'un service Web est capable de faire, son emplacement et comment l'invoquer.

Le standard offre une description en deux parties: une description abstraite et une description des concepts du modèle WSDL.

Les éléments de la partie abstraite décrivent principalement le service en termes de messages (envoyés et reçus). Ils sont indépendamment décrits par un schéma XML de description concrète.

Au niveau d'une balise XML notée "types" ,la partie abstraite inclut des éléments notés "opération" dont chacune associe un ou plusieurs messages à un modèle d'échange de messages. Le modèle spécifie le nombre et l'ordre ainsi que la cible et la source abstraites des messages. Les opérations sont groupées dans un élément noté "interface" ou aussi port Type" sans préciser le moyen (le protocole) d'échange des messages.

Techniquement, la structure d'un élément "bin ding" ressemble à la structure d'un élément "interface/ port Type" mais avec des détails en plus.

La spécification WSDL offre deux types de "bin ding " pour l'envoi des messages entre client et services Web :

- Soit intégrés dans des messages du protocole SOAP transmis par exemple en HTTP,
- Soit directement par exemple dans des messages du protocole HTTP.

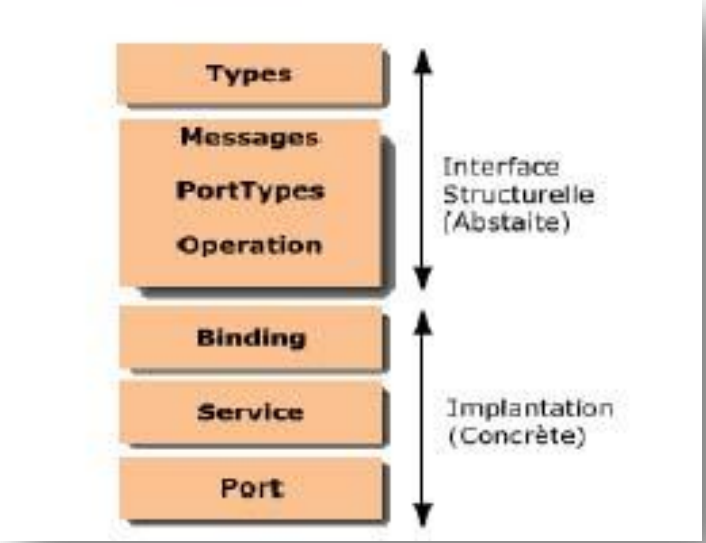

Figure 8: « Structure d'une description WSDL »

En résume un fichier WSDL contient donc sept éléments.

- **Types**: fournit la définition de types de données utilisés pour décrire les messages échangés.
- **Messages** : représente une définition abstraire (noms et types) des données en cours de transmission.
- **Port Types**: décrit un ensemble d'opérations. Chaque opération à zéro ou un message En entrée, zéro ou plusieurs messages de sortie ou d'erreurs.
- **Bin ding**: spécifie une liaison entre un <port Type> et un protocole concret (SOAP, HTTP...).
- **Service**: indique les adresses de port de chaque liaison.
- **Port**: représente un point d'accès de services défini par une adresse réseau et une liaison.
- **Opération**: c'est la description d'une action exposée dans le port.

Conclusion :

Nous avons présenté dans ce chapitre les concepts de base des applications web. L'utilisation de la notion d'application web peut nous assurer que, le calcul des horaires soit selon une seule méthode, celle qu'on a défini, où on peut contrôler ce calcul et l'adapter, en plus elle permet l'utilisation de l'application simultanément par différents utilisateurs. L'utilisation de service web de Google Maps, est une solution pour assurer une qualité de

service que l'on ne peut pas réaliser par nous-mêmes, car c'est inutile et difficile de réaliser une application de cartographie de la qualité et de la puissance de Google Maps.

Ainsi, ces deux notions d'applications web et de services web, nous offrent une diffusion, une sécurité et un coût minimal pour notre application.

Chapitre 2 : Les services web Google Maps et Google Weather

Introduction :

La conception d'applications cartographiques pour le Web est en plein essor. C'est ainsi que les solutions de développement proposées par Google deviennent de plus en plus utilisées. En effet, depuis mai 2010, est proposée une nouvelle API de Google Maps qui a la particularité de pouvoir facilement être utilisée sur plusieurs plates-formes : ordinateurs classiques et périphériques mobiles.

Apparu en 2005, Google Maps est un service gratuit de cartographie en ligne. En effet, La possibilité de naviguer à travers le monde entier de manière fluide avec une simple connexion Internet a fait de ce projet un véritable succès.

En 2010, le Web compte près de 350000 sites proposant une composante de cartographie utilisant la technologie Google Maps. Le succès de cette API a été et reste toujours la mise à disposition d'une interface de programmation gratuite à destination des développeurs, leur permettant d'intégrer de la cartographie dynamique dans leurs propres applications web.
Dans cette partie nous présentons de façon plus détaillée le service web google Maps.

1. Cartographie dynamique sur le web :[3]

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

D'une manière simplifiée, la cartographie dynamique regroupe l'ensemble des technologies permettant d'afficher une carte sur le Web. Ces technologies reposent principalement sur les trois composantes que sont le client, le serveur et les données. La cartographie dynamique permet donc, en fonction d'une requête d'un client envoyée au serveur cartographique, de retourner les données désirées sous la forme d'une carte (voir figure 9).

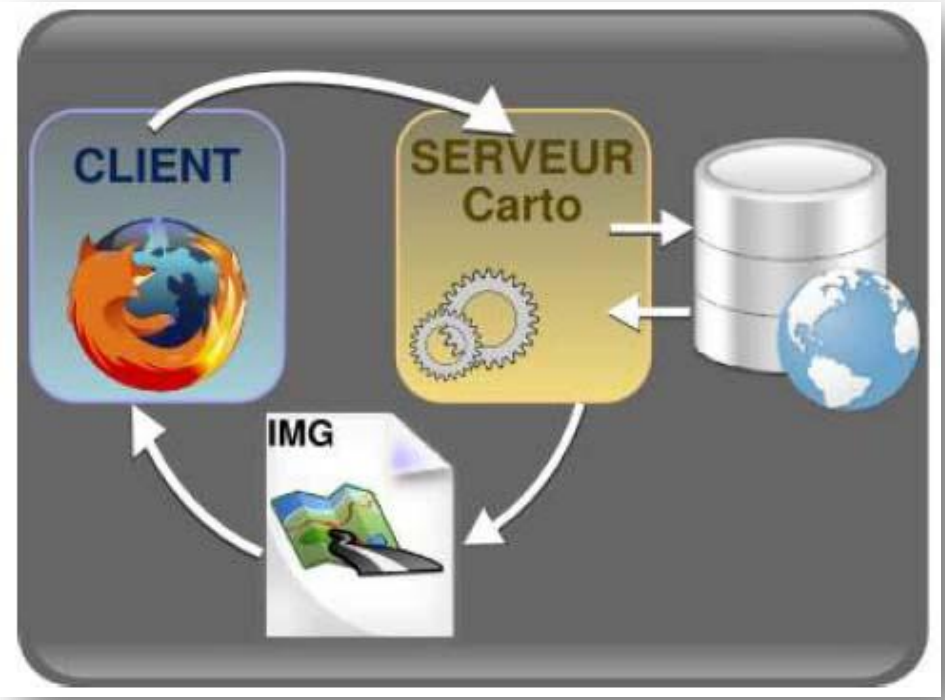

Figure 9: « Organisation d'une application de cartographie numérique »

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

2. Le service web Google Maps :[4]

2.1. Définition du service web Google Maps :

Google Maps est un service gratuit de cartographie en ligne. Le service a été créé par Google et lancé en 2004 aux États-Unis et au Canada et en 2005 en Grande-Bretagne (sous le nom de Google Local). Ensuite, Google Maps a été lancé jeudi 27 avril 2006, simultanément en France, Allemagne, Espagne et Italie.

Ce service permet, à partir de l'échelle d'un pays, de zoomer jusqu'à l'échelle d'une rue. Des prises de vue fixes montrant les détails de certaines rues sont également accessibles grâce à une passerelle vers Google Street View, un autre service de cartographie sur le web.
Deux types de vue sont disponibles dans Google Maps : une vue en plan classique, avec nom des rues, quartier, villes et une vue en image satellite, qui couvre aujourd'hui le monde entier.

2.2. La structure de la base de données du Google Maps :

La surface de la carte, qu'on voit, dans Google Map est un ensemble d'images, où chaque image représente une surface de la terre, et où pour chaque déplacement sur la carte, le serveur de Google Maps apporte les images correspondantes. La structure de stockage de ces images est celle présentée dans la figure10.

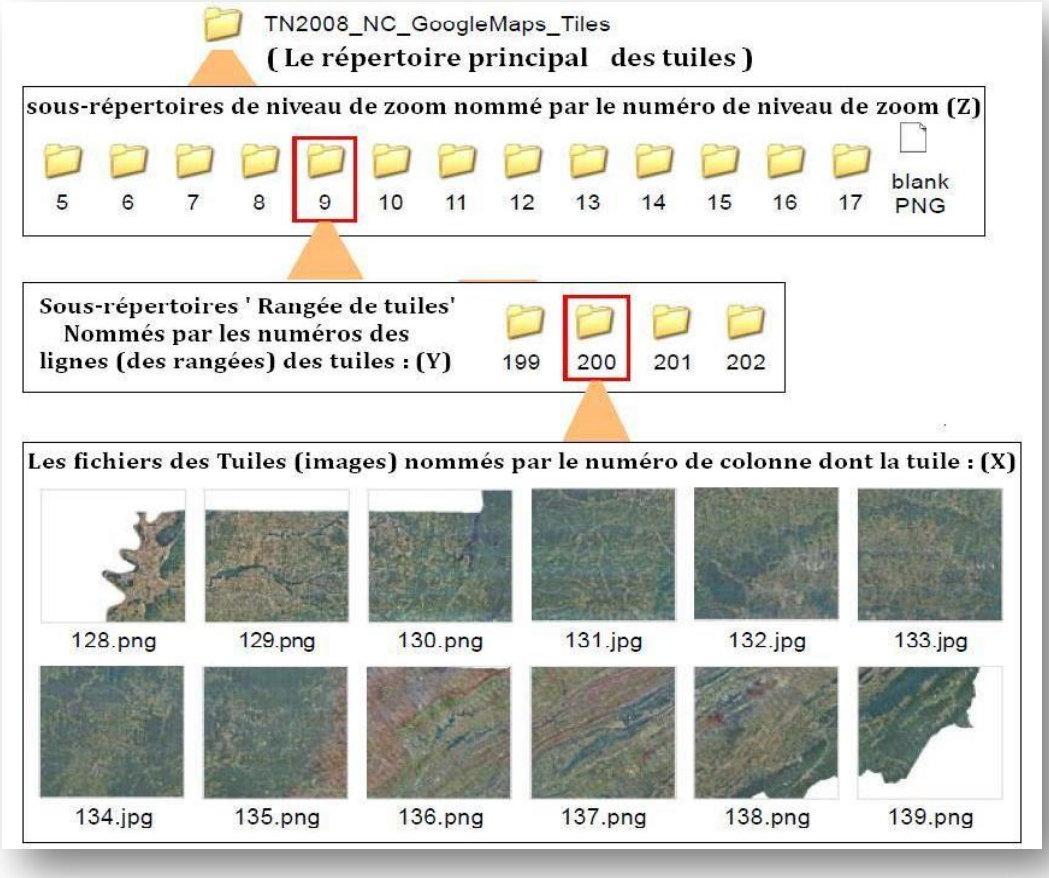

Figure 10: « La structure de stockage des images (données) de la carte Google Maps »

Définition : la tuile (dite Tile en Anglais) dans la carte Google Maps est une portion carrée (comme un Carreau) de longueur du côté égale à 256 pixels de la surface d'affichage de la carte c'est-à-dire la tuile est une image carrée qui représente une portion de la carte à un

niveau de zoom donné, et alors toute la carte est composée d'un ensemble de tuiles. Les tuiles existent pour chaque type de carte et pour chaque niveau de zoom en cours d'utilisation.

En se basant sur cette structure, on peut référencer chaque point sur la carte par ses coordonnées, et nous détaillons par la suite les différents systèmes de coordonnées.

2.3. Les Coordonnées dans Google Maps :

Il existe trois systèmes de coordonnées que l'API Google Maps utilise:

❖ **Par Pixel :**

C'est le référencement d'un point sur une tuile d'images, où chaque tuile dans Google Maps se compose de 256 x 256 pixels. Un point sur une tuile en particulier peut donc être référencé par rapport à l'origine (0,0) pour chaque tuile qui est notée à l'angle nord-ouest de la tuile. Pour chaque niveau de zoom, on peut référencer les valeurs de x et y d'un pixel sur la carte, par exemple au niveau de zoom 1 par des valeurs entre 0 et $256*2^1$, et au niveau de zoom 19 par des valeurs entre 0 et $256*2^{19}$.

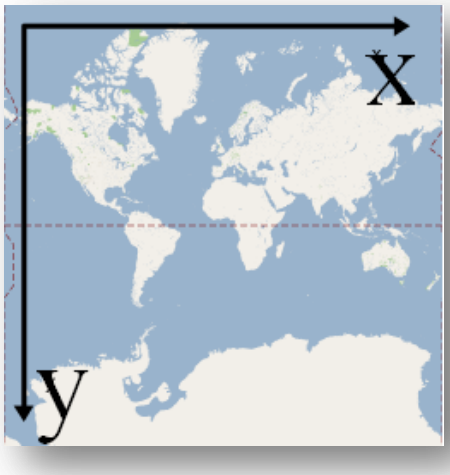

Figure 11: « Référencement par Pixel »

❖ **Par Carrelage :**

C'est le référencement d'une tuile dans une couche de tuiles. Au niveau de zoom élevé, l'API Google Maps n'utilise pas un seul fichier image pour afficher la terre entière, il est donc utile de déterminer quelle tuile (images) est en cours d'utilisation, puis calculer les coordonnées d'un pixel par rapport aux coordonnées de l'origine de cette tuile.

Les tuiles dans Google Maps sont numérotées à partir de la même origine que les pixels, de sorte que la tuile d'origine est toujours à l'angle nord-ouest de la carte. Les tuiles sont indexées à l'aide des coordonnées x, y depuis cette origine. Par exemple, au niveau de zoom 2,

lorsque la terre est divisée en 16 tuiles, chaque tuile peut être référencée par une paire unique (x,y) où *x* représente le numéro de la tuile et *y* représente le numéro de la ligne (rangée) dans cette tuile, comme c'est présenté sur la carte

Figure 12: « Référencement par Carrelage »

❖ **Par la couche de zoom :**

A chaque niveau de zoom, la carte est divisée en 4^Ntuiles, où N désigne le niveau de zoom, alors on peut définir le nombre total des tuiles pour chaque niveau de zoom.

Pour les types de carte satellite (*G_SATELLITE_MAP* et *G_HYBRID_MAP*), on peut déterminer le niveau de zoom maximum, où les images satellites sont disponibles à un endroit donné.

Il existe des méthodes offertes par l'API Google Maps, pour convertir une valeur de type *GLatLng* (paire des réels représentant la latitude et la longitude d'une région pointée sur la carte) vers une coordonnée pixel sur un niveau de zoom donné.

Et de la même manière, pour convertir une coordonnée pixel sur un niveau de zoom donné vers une valeur *GLatLng*. Dans notre projet, nous avons utilisé ces méthodes pour obtenir sur une carte la latitude et la longitude de la ville dont on veut calculer les horaires de prière.

2.4. API de Google Maps (Application Programme Interface) :

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

Google Maps API (Application Programme Interface) permet d'accéder aux images satellite de la base de données de Google depuis une page Web connue pour offrir cette possibilité (par exemple: météo.fr) ou depuis une page perso à créer (il suffit d'ouvrir un compte Google pour obtenir la « clé API » nécessaire). Elle constitue une interface servant de fondement à un travail de programmation plus poussé.

On peut classer ces APIs qui servent à notre objectif en deux catégories comme suit :

2.4.1. Les APIs Google Maps :[5]

a. API Google Maps JavaScript V3 :

L'API JavaScript Google Maps est un script JavaScript qui est appelé grâce à la balise <script>. L'appel au code doit être fait avec un paramètre "key". Elle nous permet d'intégrer une carte Google sur une page web en utilisant JavaScript, de la manipuler et d'y ajouter du contenu à travers une variété de services.

b. API Google Maps Flash :

Nous utilisons cette API Action Script pour intégrer une carte Google dans une page Web ou une application basée sur Flash, manipuler la carte en trois dimensions et ajouter du contenu à travers de nombreux services.

Elle nécessite :

- une clé fournie par Google.
- une plateforme (environnement) spécifique soit, sur Flex SDK (exige Java 1.4 ou 1.5), Flex Builder (Adobe's IDE for Flex développement) :

Sur Flex SDK : il suffit de créer un simple fichier MXML contenant une carte en utilisant le code Action Script, compiler ce fichier dans un fichier SWF à l'aide du Flex SDK, inclure ce fichier SWF dans une page HTML, et l'afficher.

Sur Flex Builder : utilisant Flex Builder, on crée un simple fichier MXML, auquel on ajoute du code Action Script et ce fichier sera compilé par la suite dans un fichier SWF à l'aide du SDK Flex, et il pourra être lancé pour l'inspection visuelle.

2.4.2. Les APIs Google Maps Web Services:

Dans ce cas les services offerts par Google Maps sont des informations sur le géocodage (Géocoding), les directions, l'altitude (Elévation) et les Places.

L'API Google Maps offre ces Services Web, comme une interface pour demander et interroger des données API des cartes depuis des services (client) externes et les utiliser dans

nos applications Maps (i.e. des applications client). Ces services sont conçus pour être utilisés en conjonction avec une carte.

Ces Web Services utilisent des requêtes HTTP à des URL spécifiques, en passant des paramètres d'URL comme arguments pour les services et en général, ces services retournent les données dans la requête HTTP soit comme JSON ou XML pour l'analyse et /ou le traitement par notre application. Une requête typique de Service Web est généralement de la forme suivante:

Où service indique le service demandé et output précise la forme de résultat retourné (habituellement JSON ou XML).

2.5. Navigation dans Google Maps:

Il existe plusieurs façons de naviguer dans Google Maps. La première, la plus intuitive, consiste à cliquer sur un endroit de la carte, à maintenir le clic gauche enfoncé, et à déplacer la souris dans le sens ou vous désirez faire défiler la carte, comme si vous faisiez tourner une mappemonde. (Si possible, éviter de prendre comme « point d'appui » une extrémité de la carte, cela réduit l'amplitude de mouvement, et multiplie donc les manipulations.) La roulette de la souris est utilisée pour le zoom avant et arrière. Vous disposez également d'une échelle de distance, en bas a gauche de la carte.

Figure 13: « page d'accueil de google maps »

2.6. Récupération des coordonnées d'une adresse avec google Maps :

On peut récupérer grâce à un script développé avec l'API Google Maps les coordonnées géographiques d'un point à partir de son adresse en l'utilisant l'api Google ; Cette méthode peut s'avérer très efficace surtout lorsque l'on a besoin de récupérer beaucoup de coordonnées.

La réponse retournée par Google est au format CSV. Chaque coordonnée renvoyée est précédée d'un code permettant de savoir si l'api est parvenue à géo-localiser l'adresse en question (exemple: 200 = ok, 602 = erreur, …). Associée à un script PHP et à une connexion à une BDD, on peut alors automatiser cette tâche de récupération de coordonnées géographiques. L'outil proposé permet de récupérer les coordonnées d'une adresse, d'une ville, d'un département,

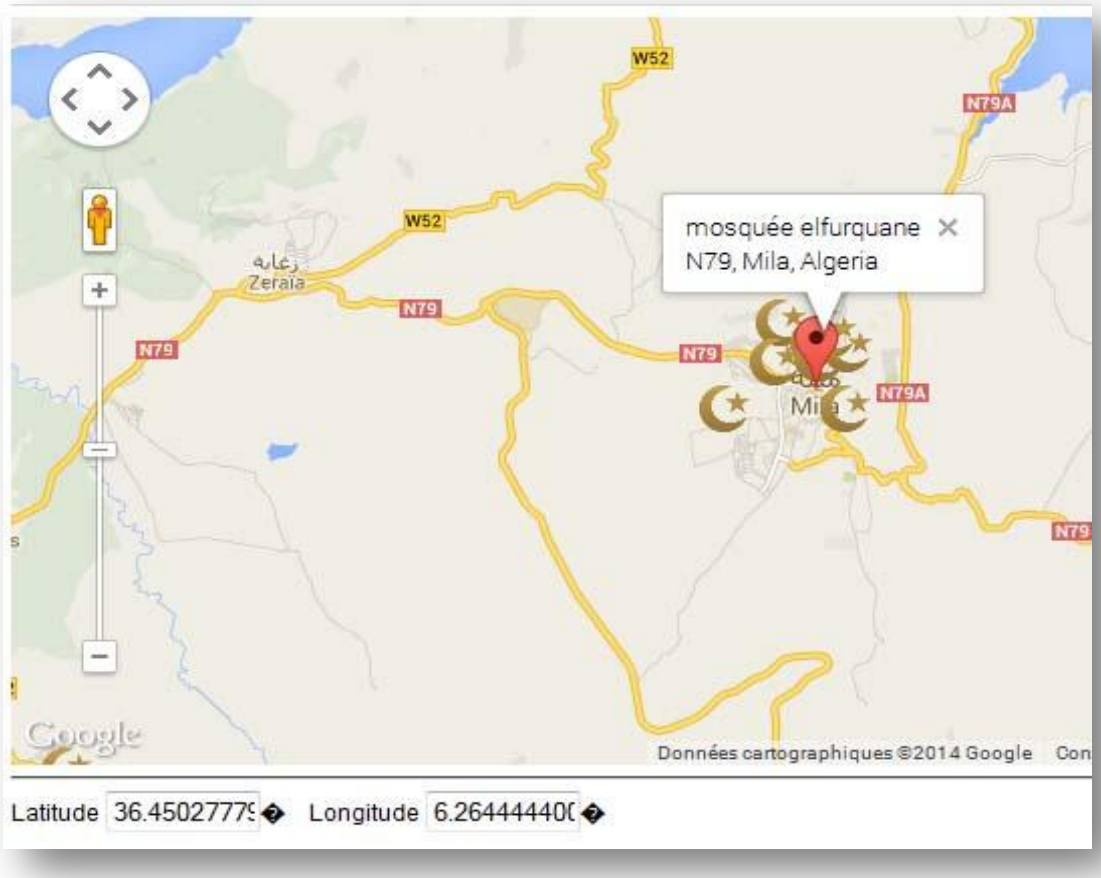

Figure 14: « géo localisation d'un lieu grâce à l'API Google Maps»

2.7. Les traitements sur carte offerts par L'API Google Maps JavaScript version 3 :

L'**API Google Maps JavaScript version 3** nous permet entre autre de :

❖ Créer, modifier et personnaliser des **cartes géographiques** à l'aide de la **classe google.maps.Map**. Plusieurs **types de cartes** existent : **Plan**, **Satellite**, **Mixte**, **Relief**. Grâce aux caractéristiques de l'objet **google.maps.MapOptions** on peut définir les propriétés d'une carte selon nos besoins : centrer la carte, définir le type de carte, activer ou désactiver le contrôle panoramique, etc. Ces cartes sont interactives. Nous pouvons zoomer ou les déplacer à volonté. Il est également possible d'ajouter des **observateurs d'événements** afin de faire réagir la carte à certains événements. Enfin, nous pouvons afficher différents types d'informations sur la **carte Google**, comme par exemple un **itinéraire**, une **image**, un **polygone**, une **polyline**, une **info bulle**, un **Street view**….

❖ Géo-localiser sur une **carte** un **point** particulier. Celui-ci est identifié à partir de ses **coordonnées GPS** grâce à un **marqueur** créé via la **classe google.maps.Marker**. Ainsi on peut obtenir la l**atitude et la longitude du point** grâce à la **classe google.maps.LatLng**. Les **marqueurs** sont représentés à l'aide de deux **images** (icône et ombre) et sont personnalisables à l'aide de **la classe google.maps.MarkerOptions**. Enfin, les **marqueurs** sont interactifs. Ils peuvent être utilisés, via des **observateurs d'événements**, pour réagir à certaines actions et déclencher, par exemple, l'ouverture d'une **info-bulle**.

3. Le service web Google weather:[6]

Google Weather fournit des informations météorologiques en temps réel via Internet, des bulletins météo pour les villes à travers le monde ainsi que des rapports météorologiques locales pour les journaux et les sites Web. Elle offre notamment l'accès aux informations météo des villes et des régions, la récupération des informations prévisionnelles, l'accès à des images satellite, des conditions actuelles et par emplacement.

L'API Weather permet aux développeurs et aux utilisateurs d'accéder aux données du climat pour intégrer ces données et ces fonctionnalités dans d'autres applications.

3.1. Qu’est-ce que Google Weather ?

Google Weather permet d'afficher les conditions météorologiques actuelles. On peut le développer pour afficher les catégories suivantes :

- ***Nuages* :** affiche les nuages au-dessus de la Terre. (Si vous ne les voyez pas, essayez de faire un zoom arrière.)
- ***Radar* :** affiche les images radar des conditions de pluie, de neige et de gel, selon la légende suivante :

- ***Conditions et prévisions* :** affiche la température actuelle en degrés Fahrenheit et Celsius avec une icône représentant les conditions météorologiques dans la visionneuse 3D. on peut Cliquer sur cette icône pour faire apparaître une info-bulle avec des prévisions détaillées.
- ***Informations* :** affiche des détails concernant les calques Google Weather, y compris l'heure des images de nuages et de radar actuelles.

Les données Google Weather sont fournies par weather.com et Weather Services International.

Affichage de la pluie et de la neige :

Pour certaines régions d'Amérique du Nord et d'Europe, Google Earth peut projeter des images de pluie et de neige au-dessus des zones ayant de telles conditions météorologiques, comme si cela était vraiment réel. Nous pouvons déterminer si la zone qui nous intéresse est couverte en activant les données géographiques du radar.

Figure 15: « Image fournie par le blog Google LatLong (Ouragan Alex 2010) »

Conclusion :

Ce chapitre a présenté les deux services Web Google Maps et google Whether. LAPIs JavaScript V3 en particulier a été choisie pour le développement de notre application étant plus légère et facile d'utilisation, (par rapport aux autres API surtout l'API Flash).

Chapitre 3 : La démarche d’analyse et de conception

Introduction :

Ce chapitre présente de façon brève une démarche simplifiée adaptée aux applications Web issue de [7]. Le processus définit une séquence d'étapes, partiellement ordonnées, qui concourent à l'obtention d'un système logiciel. Des précisions sont apportées pour présenter les stéréotypes UML spécifiques aux applications Web.

1. Présentation de la démarche :

La démarche de développement que nous avons adoptée pour l'application constitue une adaptation simplifiée du processus UP pour des projets de logiciel de petite taille. En plus elle introduit au cours de l'analyse et de la conception, des concepts spécifiques aux applications web [7].

Le Processus Unifié UP (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 :

Itératif et incrémental : le projet est découpé en itérations de courte durée (environ 1 mois) qui aident à mieux suivre l'avancement global. À 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.

2. La démarche de développement :

La démarche de développement s'articule autour des étapes citées ci-dessous :

1) L'étude préliminaire.
2) Le modèle des besoins.
3) La spécification détaillée des besoins.
4) Le modèle de domaine.

5) Le diagramme de classes participantes.
6) La conception détaillée du système.
7) L'implémentation.

2.1. L'Etude préliminaire :

Cette étape prépare la description des besoins par les cas d'utilisation. Ses objectifs peuvent être résumés comme suit :

- Bien comprendre le système.
- Recenser tous les besoins.
- Décrire textuellement ces besoins.

2.2. Le modèle des besoins :

Le modèle des besoins sert à :

- déterminer tous les acteurs,
- recenser les cas d'utilisation,
- établir le diagramme de cas d'utilisation
- pour chaque cas d'utilisation le diagramme d'activité représentant le scénario de son exécution.

2.3. La spécification détaillée des besoins : elle consiste à

- Déterminer les scénarios pour chaque cas d'utilisation.
- Représenter chaque scénario par un diagramme de séquence.

2.4. Le modèle de domaine :

C'est est une étape totalement dissociée de l'analyse des besoins, elle se divise par plusieurs étapes:

- identifier les classes métier.
- Identifier les entités ou concepts du domaine.
- Identifier et ajouter les associations et les attributs.
- Organiser et simplifier le modèle en éliminant les classes en paquetage selon les principes de cohérence et d'indépendance.

2.5. Le diagramme des classes participantes :

Il effectue la jonction entre, d'une part les cas d'utilisation et le modèle de domaine, et d'autre part les diagrammes de conception logicielle. Il modélise trois types de classe d'analyse : les dialogues, les contrôles et les entités ainsi que leurs relations :

Les classes de dialogue : 

Elles permettent les interactions entre l'IHM (Interface Homme/Machine) et les utilisateurs. Ces classes sont directement issues de l'analyse de la maquette. En général les dialogues vivent seulement le temps du déroulement du cas d'utilisation concerné.

Les classes de contrôle : 

Ces classes modélisent la cinématique de l'application. Elles font la jonction entre les dialogues et les classes métier en permettant aux différentes vues de l'application de manipuler des informations détenues par un ou plusieurs objets métier. Elles contiennent les règles applicatives et les isolent à la fois des dialogues et des entités.

Les classes entités : 

Ce sont les classes métier qui proviennent directement du modèle du domaine, elles sont généralement persistantes, c'est-à-dire qu'elles surviennent à l'exécution d'un cas d'utilisation particulier et qu'elles permettent à des données et des relations d'être stockées dans des fichiers ou des bases de données. Lors de l'implémentation, ces classes peuvent ne pas se concrétiser par des classes mais par des relations, au sens des bases de données relationnelles.

2.6. La conception détaillée du système :

Cette étape consiste à déterminer comment les objets logiciels vont interagir entre eux pour réaliser telle ou telle opération. Elle contient plus de détail en prenant en compte les objets systèmes en plus des objets métier. Elle consiste à élaborer :

- Le diagramme des classes de conception (détaillée).
- Le diagramme de navigabilité.
- l'architecture logicielle du système, cette architecture est modélisée par un diagramme de composants où sont spécifiés les différents modules du système, ainsi que leurs interactions.

3. Spécificités UML pour les applications Web :

La conception des applications Web se distingue de la conception d'autres systèmes par deux activités majeures :

- La répartition des objets sur le client ou le serveur.
- La définition de l'interface utilisateur sous forme de pages Web.

L'extension d'UML pour les applications Web consiste à définir de nouveaux stéréotypes, étiquettes et contraintes, ce qui permet de fournir une notation pour exprimer les composants de technologie Web du système dans le modèle.

Le principal élément spécifique des applications Web étant la page Web, plusieurs stéréotypes lui sont destinés comme les formulaires, les frames, les pages,…etc.

Avant d'utiliser ces stéréotypes dans nos modèles, nous allons expliquer dans cette section leur signification et les icônes utilisées pour leur représentation.

Définition du stéréotype :

Un stéréotype est une extension d'une classe, il a pour objectif de définir une utilisation particulière des éléments modélisés ou pour modifier la signification d'un élément.

a) Le stéréotype page serveur «contrôle» :

Une page serveur représente une page Web qui possède des scripts exécutés par le serveur. Ces scripts interagissent avec des ressources du serveur, telles que bases de données, le logique métier ou des systèmes externes. Les opérations de l'objet représentent les fonctions dans le script et ses attributs représentent les variables qui sont visibles dans la portée de la page (c'est-à-dire accessibles par toutes les fonctions de la page).

Icône :

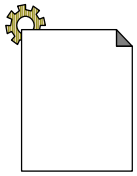

Figure 16: stéréotype page «contrôle»

b) Le stéréotype page «interface»

Une instance d'une page client est une page Web formatée en HTML, mélange de données, de présentation et même de logique. Les pages client, restituées par des navigateurs client, peuvent contenir des scripts interprétés par les navigateurs. Les fonctions d'une page client correspondent aux variables déclarées dans les scripts qui sont accessibles à toute fonction de la page (portée de la page). Les pages client peuvent entretenir des associations avec d'autres pages client ou serveur.

Icône :

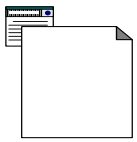

Figure 17: stéréotype page client «interface»

c)Le stéréotype page formulaire «form» :

Une classe stéréotypée «form» est un ensemble de champs de saisie faisant partie d'une page client. A une classe formulaire correspond une balise HTML « form ». Les attributs de cette classe correspondent aux éléments de saisie d'un formulaire HTML (zones de saisie, zones de texte, bouton d'option, cases à cocher et éléments cachés).

Un formulaire n'a pas d'opération, puisqu'il ne peut les encapsuler. Toute opération qui interagit avec le formulaire appartient à la page qui la contient.

Icône :

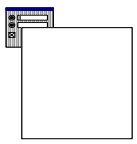

Figure 18: stéréotype page formulaire «form»

d) Le stéréotype objet script «script object» :

Un objet script client est un ensemble qui regroupe des scripts client particuliers dans un fichier, lequel est inclu par une requête distincte du navigateur client.

Icône :

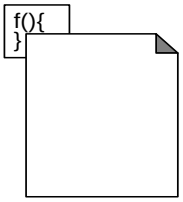

Figure 19: stéréotype objet script « script object»

e) Le stéréotype lien «link» :

Un lien est un pointeur d'une page vers une autre page. Dans un diagramme de classes, un lien est une association entre une page client et une autre page client ou une page serveur.

Icône : Aucune.

f) Le stéréotype soumettre «submit» :

Une association de soumission se trouve toujours entre un formulaire et une page serveur. Les formulaires soumettent les valeurs de leurs champs au serveur, par l'intermédiaire de pages serveur, pour qu'il les traite.

Icône : Aucune.

g) Le stéréotype construit «build» :

L'association «build» est une relation particulière qui fait le pont entre les pages client et les pages serveur. Les pages serveur n'existent que sur le serveur, ou elles sont employées à construire les pages client.

Une page serveur peut construire plusieurs pages client, en revanche, une page client ne peut être construite que par une seule page serveur.

Icône : Aucune.

Conclusion :

Ce chapitre a présenté brièvement les aspects de modélisations que nous allons appliquer pour le développement de notre application. Le chapitre suivant résume les étapes d'analyse et de conception que nous avons suivies pour sa réalisation.

Chapitre 4 : Analyse et conception de l’application

Introduction

Ce chapitre constitue l'essentiel de notre travail il détaille l'analyse et la conception de notre système, Nous expliquons également tous les éléments théoriques et les mesures astronomiques mis en jeu dans le calcul des horaires de prière. Nous modélisons par des diagrammes UML ces différents composants et fonctionnalités.

Avec le développement technologique, qui traverse le monde dans tous les domaines, en particulier l'internet nous nous trouvons contraints de suivre ce développement en profitant les services web et d'essayer de développer d'autres applications.

Mais la question qui se pose est que la majorité des musulmans dans des pays étrangers ou des États non-musulmans estiment qu'il est difficile de savoir les heures de prière. C'est pour cela qui nous est dédiée cette application qui calcule les horaires de prière de n'importe quelle région dans le monde et qui offre aussi un service supplémentaire de calcul et d'affichage des prévisions météo.

Cette application utilise les services web Google Maps et Google Weather.

I. Analyse et conception de l'application :

1. Etude préliminaire :

1.1. Objectif de notre projet :

Notre objectif est le développement d'une application web pour le calcul des horaires de prière et la détermination des prévisions météo nommée « HoPriMet », en effectuant des calculs mathématiques basés sur des notions astronomiques.

Notons que pour réaliser cette tâche, notre application invoque les services web fournis par Google Maps et Google Weather.

1.2. Notions astronomiques et calculs mathématiques :

Comme nous l'avons déjà précisé, le calcul des horaires de prière nécessite plusieurs notions et calculs astronomiques, nous présentons dans ce qui suit les calculs mathématiques utilisés dans ce but.

Pour déterminer l'horaire de chaque prière il faut qu'on détermine les huit points des temps durant le jour :

Temps/Prière	Définition
Imsak	Le temps d'arrêter de manger Sahur (pour le jeûne), 10 minutes avant El-Fajr.
Fajr	Lorsque le ciel commence à éclaircir (l'aube).
Lever (Shorouk)	L'heure à laquelle la première partie du Soleil apparaît au-dessus de l'horizon.
Dhuhr	Quand le soleil commence à décliner après avoir atteint son point culminant dans le ciel.
Asr	Le moment où la longueur de l'ombre de tout objet atteint un facteur (généralement 1 ou 2) de la longueur de l'objet lui-même plus la longueur de l'ombre de cet objet à midi.
Coucher	Le temps où le soleil disparaît sous l'horizon.
Maghrib	Peu après le coucher du soleil.
Isha	Le moment où la nuit tombe, et il n'ya pas de lumière diffuse dans le ciel.

Tableau 1 : « les huit points des temps durant le jour »

Pour le calcul des horaires des prières, en plus de la latitude et la longitude de l'endroit, nous avons besoin de deux autres mesures astronomiques essentielles, qui sont :

1) L'équation du temps : qui est la différence entre le temps lu à partir d'une horloge solaire et celui lu par une horloge.

2) La déclinaison solaire : C'est l'angle entre les rayons du soleil et le plan de l'équateur terrestre.

Les astronomes (du Département de Défense « DoD » et l'Orientation de la Marine de l'USA du « US Naval Océanographie ») nous ont fourni un algorithme mathématique pour calculer ces deux mesures : l'équation du temps (Edt) et la déclinaison solaire (D), l'algorithme est présenté dans ce qui suit :

- ✓ En premier lieu, calculer ***d***, le nombre de jours et de la fraction (heures et minutes) de l'époque dénommée "J2000.0", qui est la date julienne (2451545,0) correspondante à la date grégorienne le 1ier Janvier 2000 à 12:00:00, ***d*** se calcul par : **d = jd**- 2451545.0**.**

On va par la suite définir comment calculer la date julienne **jd,** du jour en question, de puis une date simple du calendrier Grégorien.

- ✓ Calculer les constantes : **g** et **q** qui sont en degrés comme suit :

g = 357.529 + 0.98560028 ×**d**;

q = 280.459 + 0.98564736 × **d**;

- ✓ Calculer la constante **L** (l'approximation de la longitude écliptique géocentrique apparente du Soleil) qui est en degrés comme suit :

L = q + 1.915 × sin (**g**) + 0.020 × sin (2×**g**);

La latitude écliptique du Soleil, b, peut être approchée par b = 0.

- ✓ Calculer la distance du Soleil de la Terre, **R**, en unités astronomiques (UA), par :

R = 1.00014 - 0.01671 × cos(**g**) - 0.00014 × cos (2×**g**);

- ✓ Puis calculer l'obliquité de l'écliptique moyen **e**, en degrés:

e = 23.439 - 0.00000036 × **d**;

- ✓ Ensuite, l'ascension droite du Soleil **RA**, et la déclinaison solaire **D**, peuvent être obtenues comme suit :

RA = arctan2 (cos(**e**) × sin(**L**), cos(**L**)) / 15;

D = arcsin(sin(**e**) × sin(**L**));

Où **RA** est obtenu en degrés, et on la diviser par 15 pour la convertir en heures, et **RA** sera alors réduite à la gamme 0h à 24h.

- ✓ Et en fin calculer l'équation du temps :

EdT = q/15 - **RA;**

Où **EdT** et **RA** sont en heures et **q** est exprimée en degrés.

Pour le calcul des horaires de prière nous avons besoin de plus des fonctions citées ci-après :

3) Le fuseau horaire local (FHL) :

La zone couverte par un fuseau, limitée par deux méridiens distants de 15°, s'étend du pôle nord au pôle sud. Elle est centrée sur un méridien dont la longitude est multiple de 15°. Le premier fuseau est centré sur le méridien de Greenwich (GMT). Au passage d'un fuseau à l'autre l'heure augmente ou diminue d'une heure.

Cependant, certains conflits peuvent apparaitre, comme il est le cas de Constantine (6.37 E), par exemple qui est située à l'intérieur de la zone couverte par le fuseau horaire GMT ([7.5 W, 7.5 E]) du point de vue géographique, mais pratiquement elle suit le fuseau horaire GMT+1 ([7.49 E, 22.5 E]). Pour résoudre ce conflit nous proposons dans notre cas de laisser le choix à l'utilisateur à entrer la zone horaire qu'il veut.

4) La fonction T(α) :

Pour le calcul du lever, coucher, Fajr, Maghrib et Isha, cette fonction représente la différence de temps entre la mi-journée et le moment où le soleil atteint un angle α dessous de l'horizon, et est calculée par la formule suivante:

$$\boldsymbol{T(\alpha) = \frac{1}{15} \times \arccos\left(\frac{-\sin(\alpha) - \sin(Lat) \times \sin(D)}{\cos(Lat) \times \cos(D)}\right)}$$

5) La fonction A(t) : pour le calcul de l'Asr, cette fonction calcule la différence de temps entre la mi-journée et le moment où l'ombre de l'objet est égale à t fois la longueur de l'objet lui-même plus la longueur de l'ombre de cet objet à midi:

$$\boldsymbol{A(t) = \frac{1}{15} \times \arccos\left(\frac{\sin(\operatorname{arccot}(t + \tan(Lat - D))) - \sin(Lat) \times \sin(D)}{\cos(Lat) \times \cos(D)}\right)}$$

6) Le calcul de la date julienne : La date julienne est une forme d'identification des jours, représentée sous forme d'un nombre réel qui précise le nombre des jours depuis le1er janvier 4713 av. J.C. Alors la partie décimale de la date julienne peut identifier des heures, des minutes et des secondes dans ce jour.

7) Madhab (la doctrine) : notons de plus que le calcul de Asr dépend aussi d'un paramètre précisant el Madhab (la doctrine).

Les formules pour calculer les horaires de prière sont présentées dans le tableau suivant:

Temps/Prière	Formule	Précisions
Imsak	= **Fajr – x.**	X=10 minutes.
Fajr	**= Dhuhr – T(β1).**	**β1** : est obtenir par le tableau des conventions de calcul de Fajr et l'Isha.
Lever	= Dhuhr – T(γ).	γ = 0.833 ou γ =0.0347 × √h h : est la hauteur de l'observateur en mètres.
Dhuhr	= **12 + FHL - Lng/15 – EdT(N)**	**N**: date julienne.
Asr	**=Dhuhr+A(madhab)**	- **madhab**=1, pour Shafii, Maliki, Jaafari et Hanbali. - **madhab**=2, pour Hanafi.
Coucher	= **Dhuhr + T(γ).**	γ = 0.833 ou γ =0.0347 × √h h : est la hauteur de l'observateur en mètres.
Maghrib	= Coucher + Y	**Y=** 2 minutes (ou 1 ou 3).
Isha	= Dhuhr + T(β2).	**β2** : est obtenu par le tableau des conventions de calcul de Fajr et l'Isha.

Tableau 2: «Les formules pour calculer les horaires de prière »

Le tableau ci-après présente les conventions de calcul des horaires de Fajr et Isha dans les régions du monde :

/	**Convention**	**Angle du Fajr (β1)**	Angle de l'Isha (â2)
1	Algérie, Tunis, Maroc et l'Union des Organisations Islamiques de France (UIOF).	12	12
2	Umm al-Qura University, Makkah.	18,5	90 mins (120 mins pendant le Ramadan) après le Maghrib.
3	Muslim World League (MWL).	18	17
4	Islamic Society of North America (ISNA).	15	15
5	Egyptian General Authority of Survey.	19,5	17,5
6	University of Islamic Sciences, Karachi	18	18
7	Institut de géophysique, Université de Teheran	17,7	15
8	Shia Ithna Ashari, Leva Research Institute, Qum	16	14

Tableau 3: «les conventions de calcul des horaires de Fajr et Isha dans les régions du monde»

2. Le modèle des besoins :

La capture des besoins fonctionnels nous a permis d'identifier les différents cas d'utilisation d'application qui nous avons spécifiés en utilisant un diagramme de cas d'utilisation.

2.1. Le diagramme de cas d'utilisation :

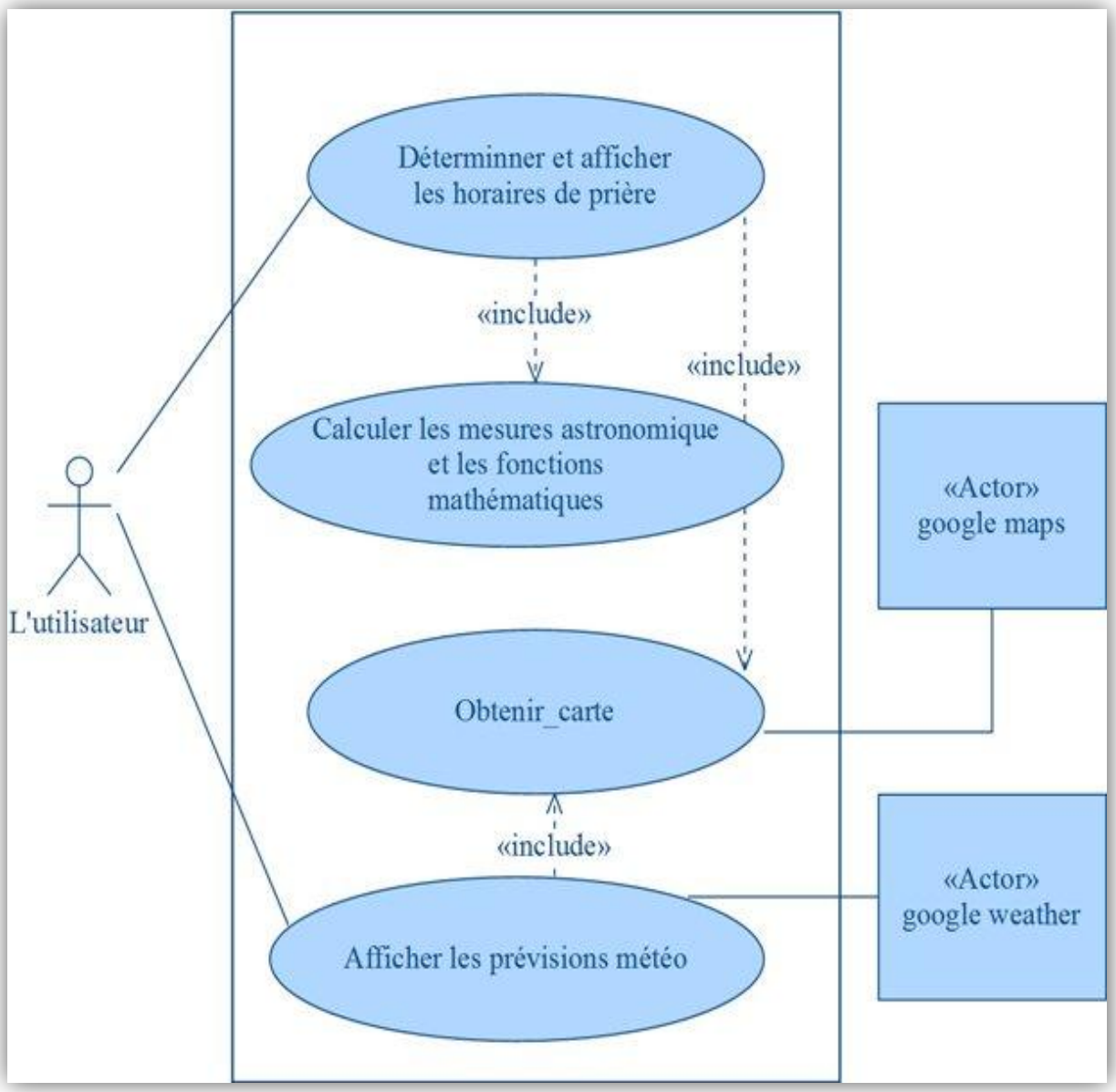

Figure 20: « le diagramme de cas d'utilisation »

2.2. La description détaillée des Cas d'Utilisations par les fiches descriptives:

1. Calculer et afficher les horaires de prière :

Cas d'utilisation	Calculer et afficher les horaires de prière
Objectif	Calculer les horaires de prière d'une région spécifique en

	utilisant les données calculée par le cas d'utilisation « calculer les mesures astronomiques et les fonctions mathématiques »
Acteur	L'utilisateur
Pré conditions	• Région saisie • Longitude et latitude Récupérées
Post condition	Horaires de prière calculés et affichés
Scénario nominal	• L'utilisateur demande au système de calculer les Horaires de prière d'une région spécifique. • Le système récupère les données du cas d'utilisation « calculer les mesures astronomiques ». • Le système effectue le calcul l'horaire d'Dhuhr et enregistré le résultat dans un tableau. • Le système effectue le calcul de l'horaire du Lever, du Asr, du Coucher et du Isha à l'aide de résultat de Dhuhr. • Le système calcule l'horaire du Meghreb à l'aide du résultat de Coucher et calcule l'horaire du Imsak à l'aide du résultat du Fajr et en registre les résultats dans un tableau. • Le système convertit le résultat de chaque horaire de prière de sa forme réelle à une forme « heures : minutes » et l'affiche à l'utilisateur.
Scénario alternatif	/
Scénario d'exception	/

Tableau 4 : « Description textuelle du cas d'utilisation

Calculer et afficher les horaires de prière »

2. Déterminer et afficher les prévisions météo :

Cas d'utilisation	Déterminer et afficher les prévisions météo
Objectif	Le système affiche les prévisions météo de la région saisie.
Acteur	Acteur principal : l'utilisateur. Acteur secondaire : Google Weather.
Pré conditions	Connexion avec Google Weather.
Post condition	La prévision météo de la vile récupérée. Les prévisions météo affichées à l'utilisateur.
Scénario nominal	• l'utilisateur demande au système la météo d'une ville saisie. • Le système envoie une requête à Google Weather pour demander des prévisions météo de la vile saisie. • Le service Google Weather retourne le résultat au système. • Le système affiche les prévisions météo récupérées à l'utilisateur.
Scénario alternatif	/
Scénario d'exception	• Le système ne connecte pas avec Google Weather. • Terminer le cas et retourne au menu principal de l'application.

Tableau 5 : « Description textuelle du cas d'utilisation

Déterminer et afficher les prévisions météo »

3. Calculer les mesures astronomiques et les fonctions mathématiques :

Cas d'utilisation	Calculer les mesures astronomiques et les fonctions mathématiques.
Objectif	Le système précise les paramètres de calcul (l'équation du

	temps, la déclinaison solaire, latitude, longitude, …)
Acteur	/
Pré conditions	La région est saisie. Longitude et Latitude de région fournies par google maps.
Post condition	• Les mesures astronomiques sont récupérées dans le table de conversion ainsi que les formules des horaires de prière. • Les résultats sont envoyés au cas d'utilisation « calculer et afficher les horaires de prière ».
Scénario nominal	• L'utilisateur demande le calcul des horaires de prière pour sa région. • L'utilisateur saisir la région. • Le système procède aux calculs des mesures astronomiques et des fonctions mathématiques (équation du temps, ….etc). • Le système sauvegarde les résultats dans un tableau. • Le système envoie le résultat au cas d'utilisation « calculer et afficher les horaires de prière ».
Scénario alternatif	/
Scénario d'exception	/

Tableau 6 : « Description textuelle du cas d'utilisation Calculer les mesures astronomiques et les fonctions mathématiques »

4. Obtenir carte :

Cas d'utilisation	Obtenir carte
Objectif	l'utilisateur saisit le nom de la ville pour calculer les

	horaires de prière ou déterminer les prévisions météo.
Acteur	Acteur principal : l'utilisateur. Acteur secondaire : Google Maps/Google weather.
Pré conditions	Connexion avec le service API Google Maps. Connexion avec le service API Google weather.
Post condition	afficher une carte de la région.
Scénario nominal	• Le système affiche une fenêtre pour saisir la ville. • l'utilisateur saisit le non de la ville. • Le système établit une connexion au service API Google Maps. • Le système récupère une carte de la vile et l'affiche.
Scénario alternatif	• Le système ne répond pas s'il y'a une erreur dans le nom de la ville. • l'utilisateur retourne au menu principal.
Scénario d'exception	• Le système ne peut pas se connecte au service API. • Le système affiche une notification à l'utilisateur. • Le cas d'utilisation se termine et ou retourne au menu principal de l'application.

Tableau 7« Description textuelle du cas d'utilisation Obtenir carte »

2.3. Description de Cas d'Utilisation par les diagrammes d'activités :

Le diagramme d'activité permet de représenter le déroulement des activités en fonction des événements du système et de modéliser des comportements parallélisables.

1. Calculer et afficher les horaires de prière :

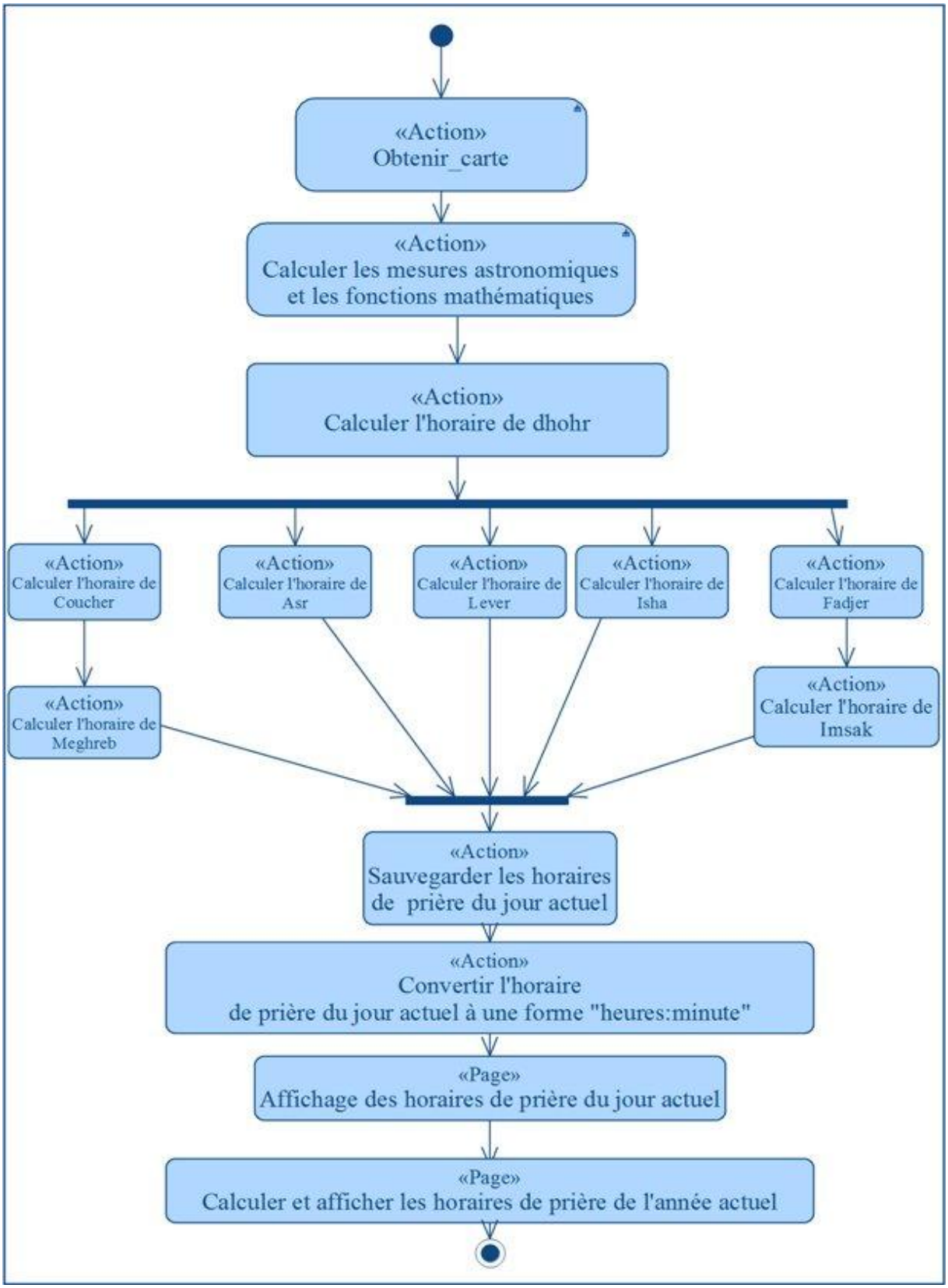

Figure 21:: « Diagramme d'activité du cas d'utilisation «Calculer et afficher les horaires de prière »

2. Déterminer et afficher les prévisions météo :

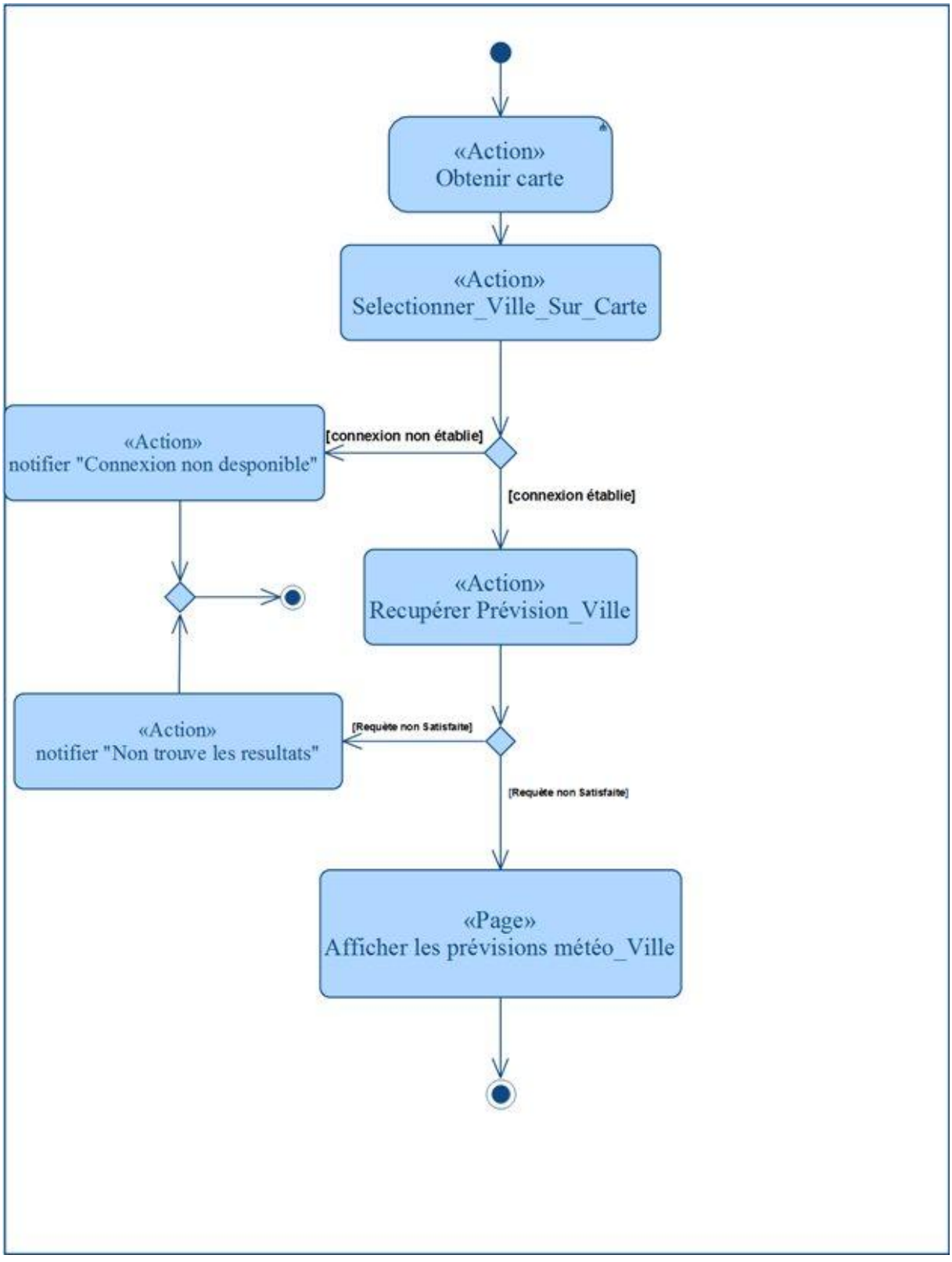

Figure 22: Diagramme d'activité du cas d'utilisation «Déterminer et afficher les prévisions météo »

3. Calculer les mesures astronomiques et les fonctions mathématiques :

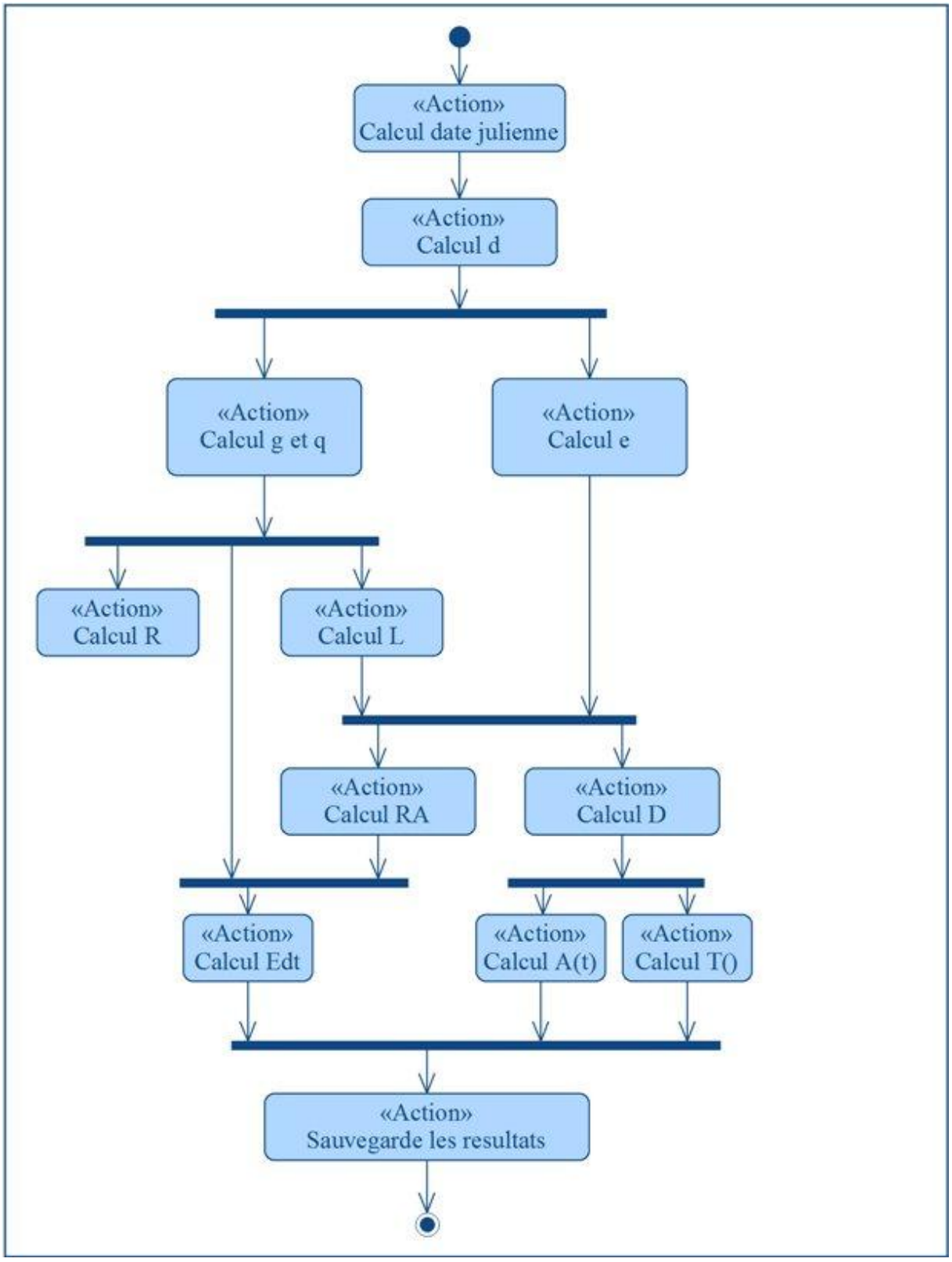

Figure 23: «Diagramme d'activité du cas d'utilisation « Calculer les mesures astronomiques et les fonctions mathématiques »

4. Obtenir carte :

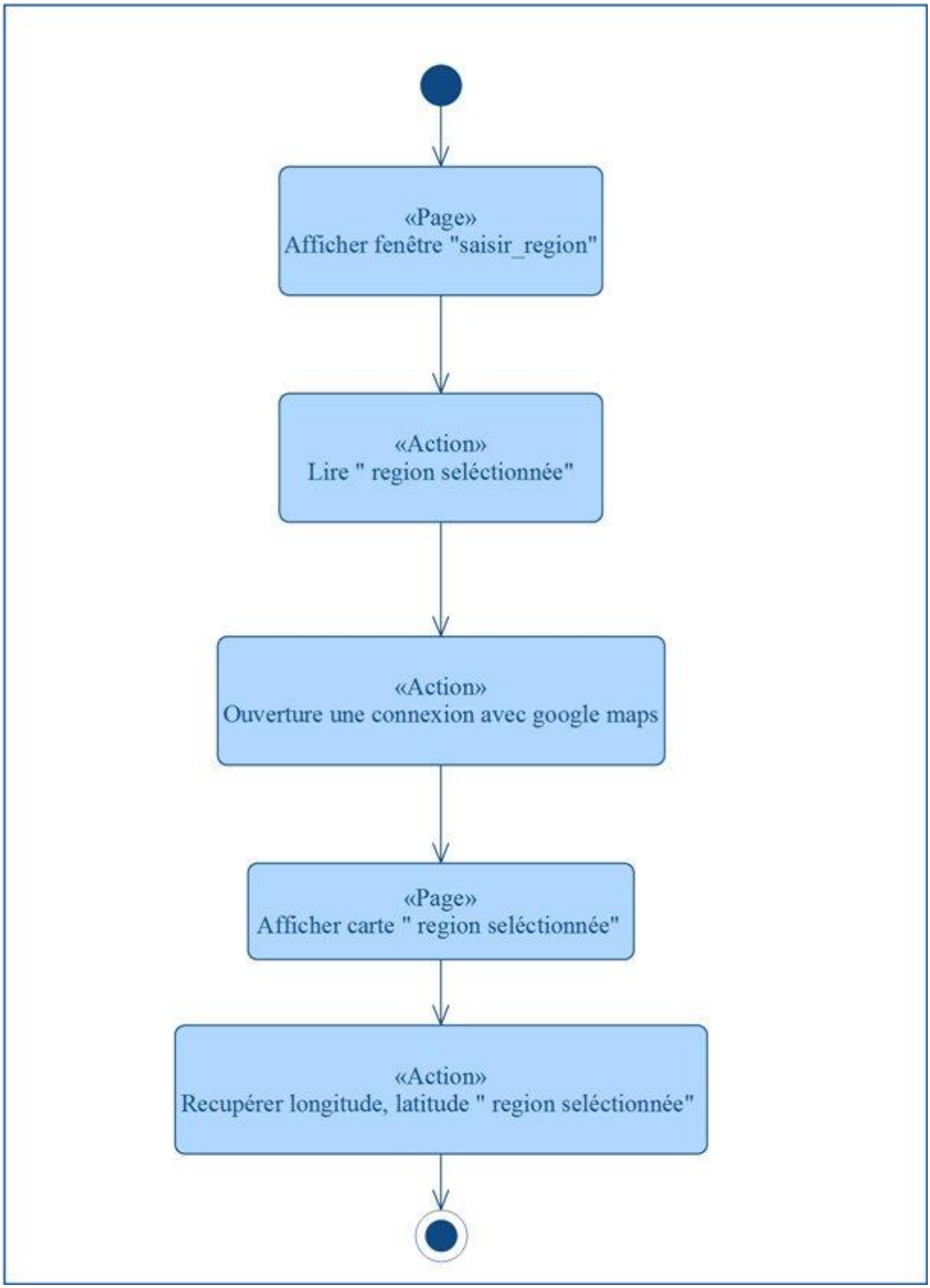

Figure 24: « Diagramme d'activité du cas d'utilisation Obtenir carte »

3. La Spécification détaillée des besoins :

3.1. Description des Cas d'Utilisation par les diagrammes de séquence :

Les diagrammes de séquence sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation UML.

1. Calculer et afficher les horaires de prière :

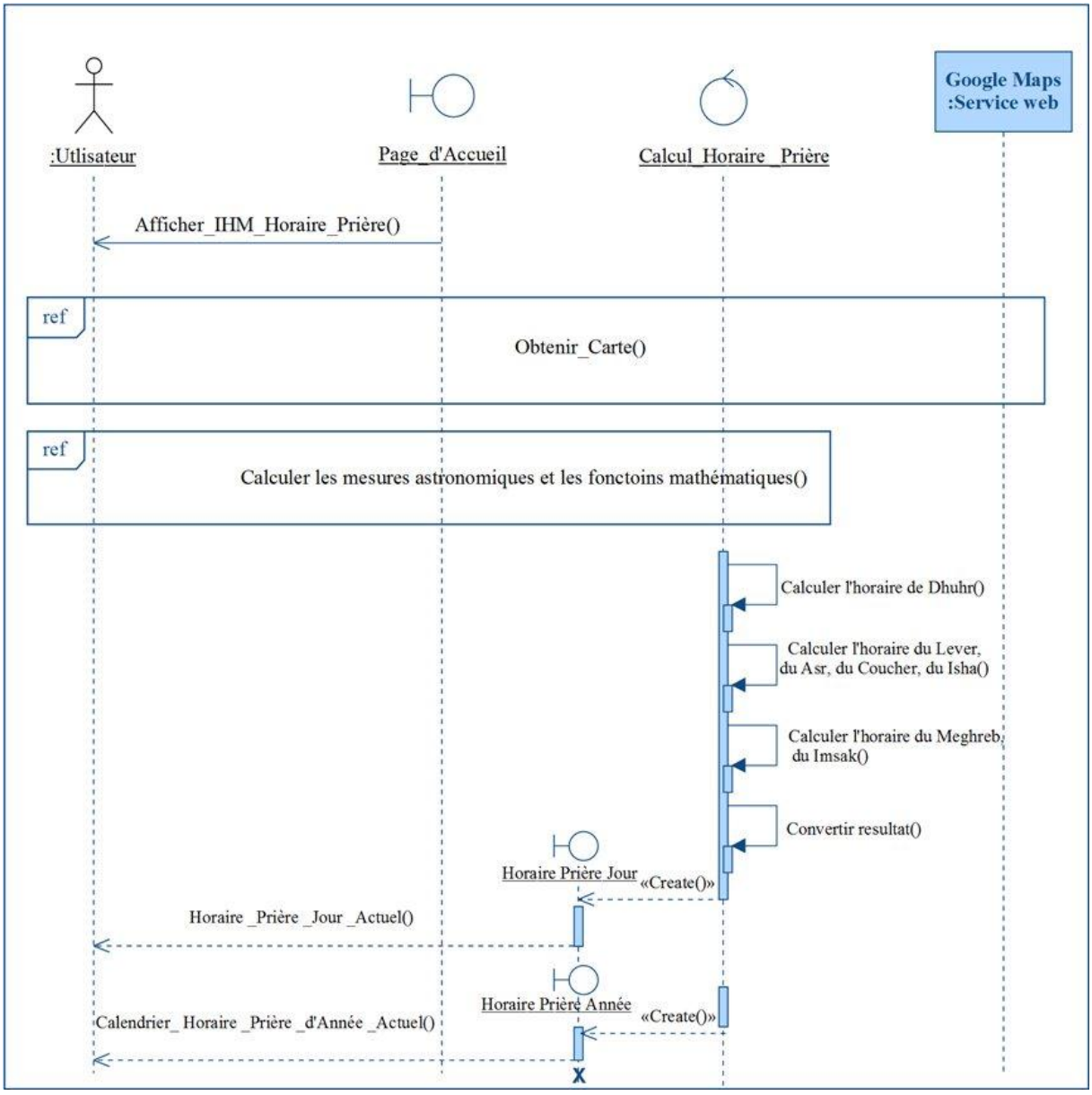

Figure 25 : « Diagramme de séquence du cas d'utilisation Calculer et afficher les horaires de prière »

2. Déterminer et afficher les prévisions météo :

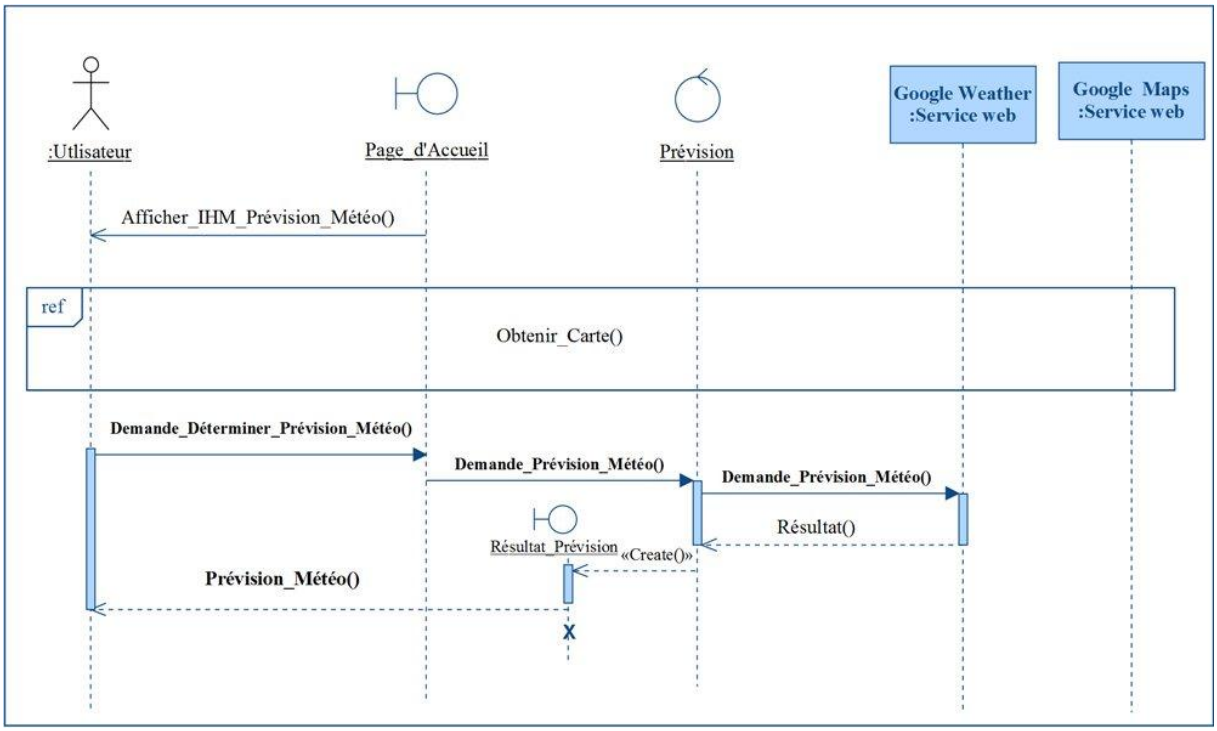

Figure 26:« Diagramme de système du cas d'utilisation Déterminer et afficher les prévisions météo »

3.Calculer les mesures astronomiques et les fonctions mathématiques :

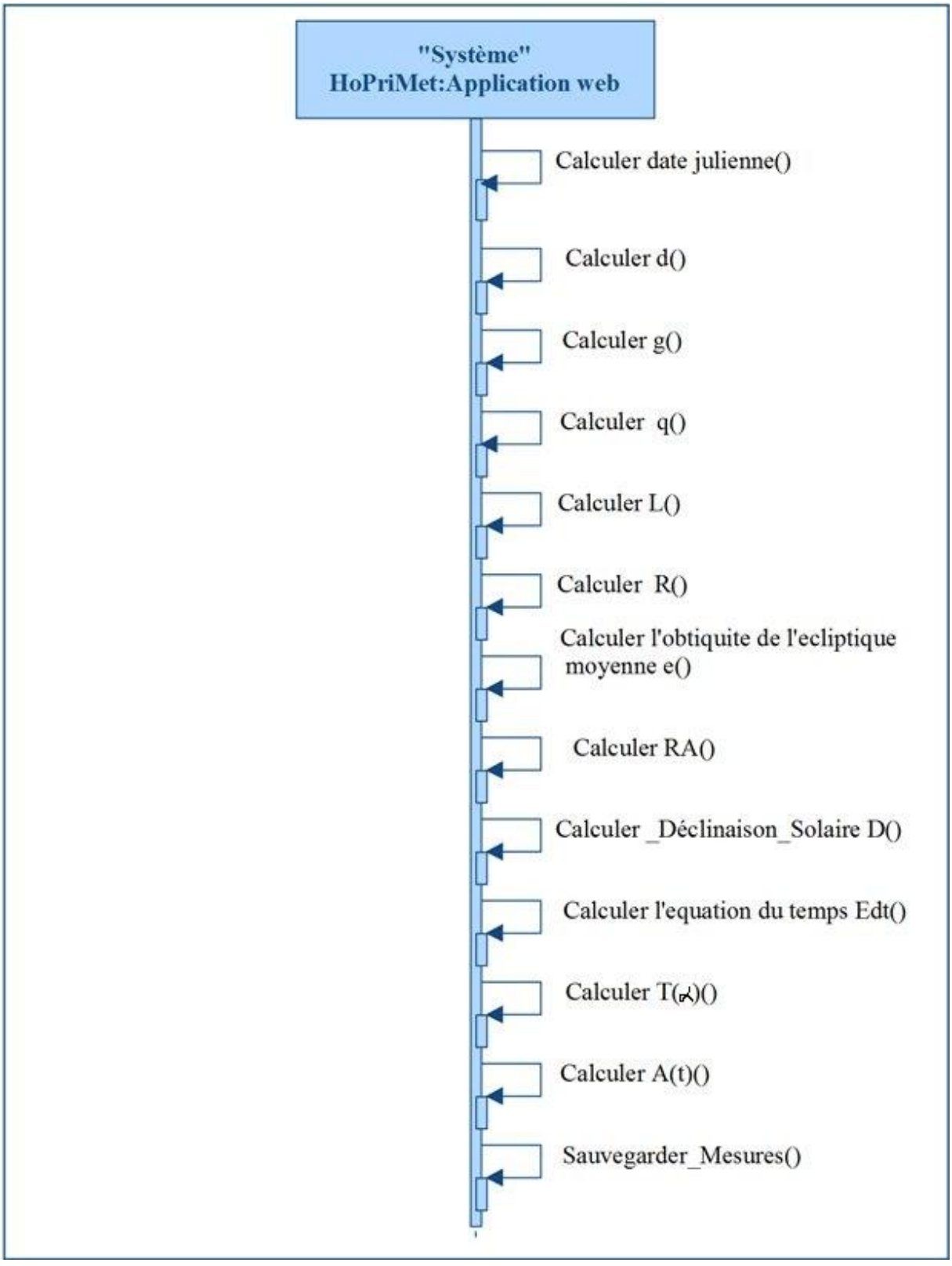

Figure 27:« Diagramme de séquence système du cas d'utilisation »

4. Obtenir carte :

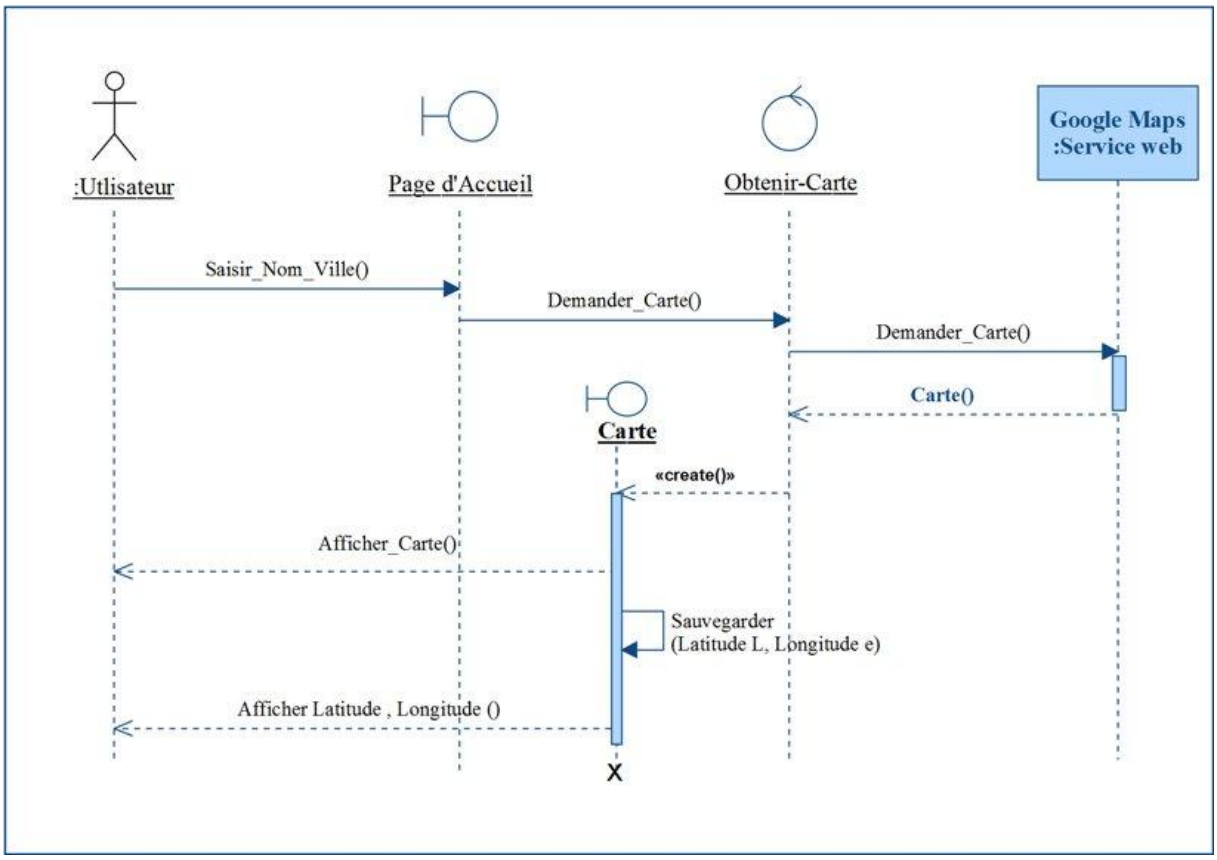

Figure 28: « Diagramme de séquence système du cas d'utilisation »

Remarque : nos objets n'étant pas persistants, nous n'avons pas élaboré le modèle de domaine ; en effet, nos objets sont tous des objets systèmes. Par conséquent nous passons directement à la conception détaillée des classes de conception.

4. Conception détaillée du système:

Pages «interface» :

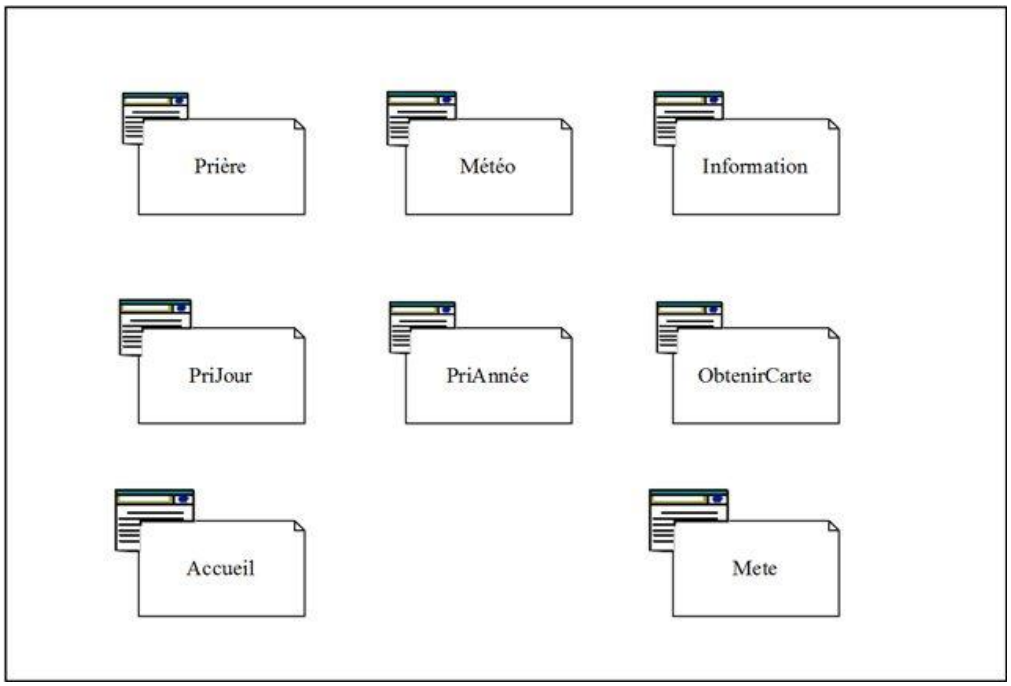

Figure 29: Pages «contrôle»

Légende de la figure :

Page	Role
Prière	interface du service calcul prière
PriJour	page résultat contenant les horaires de prière du jour
PriAnnée	Calendrier des horaires de prière de l'année
Mete	page résultat de météo

Pages «contrôle» :

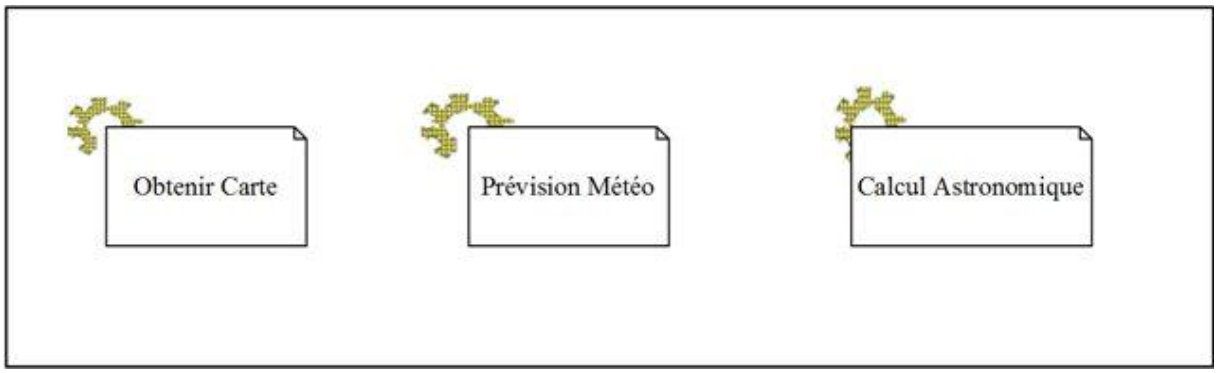

Figure 30: Pages «interface»

Formulaires «form» :

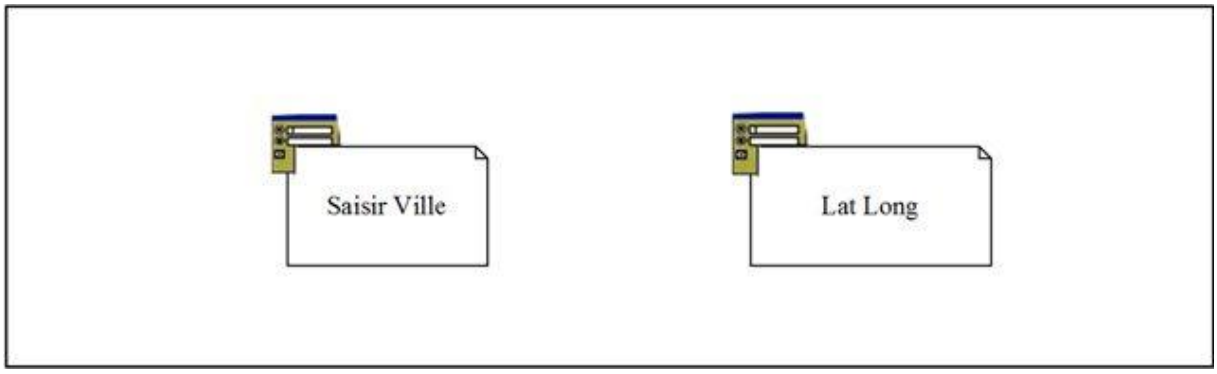

Figure 31: Formulaires «form»

4.1. Diagramme des classes de conception :

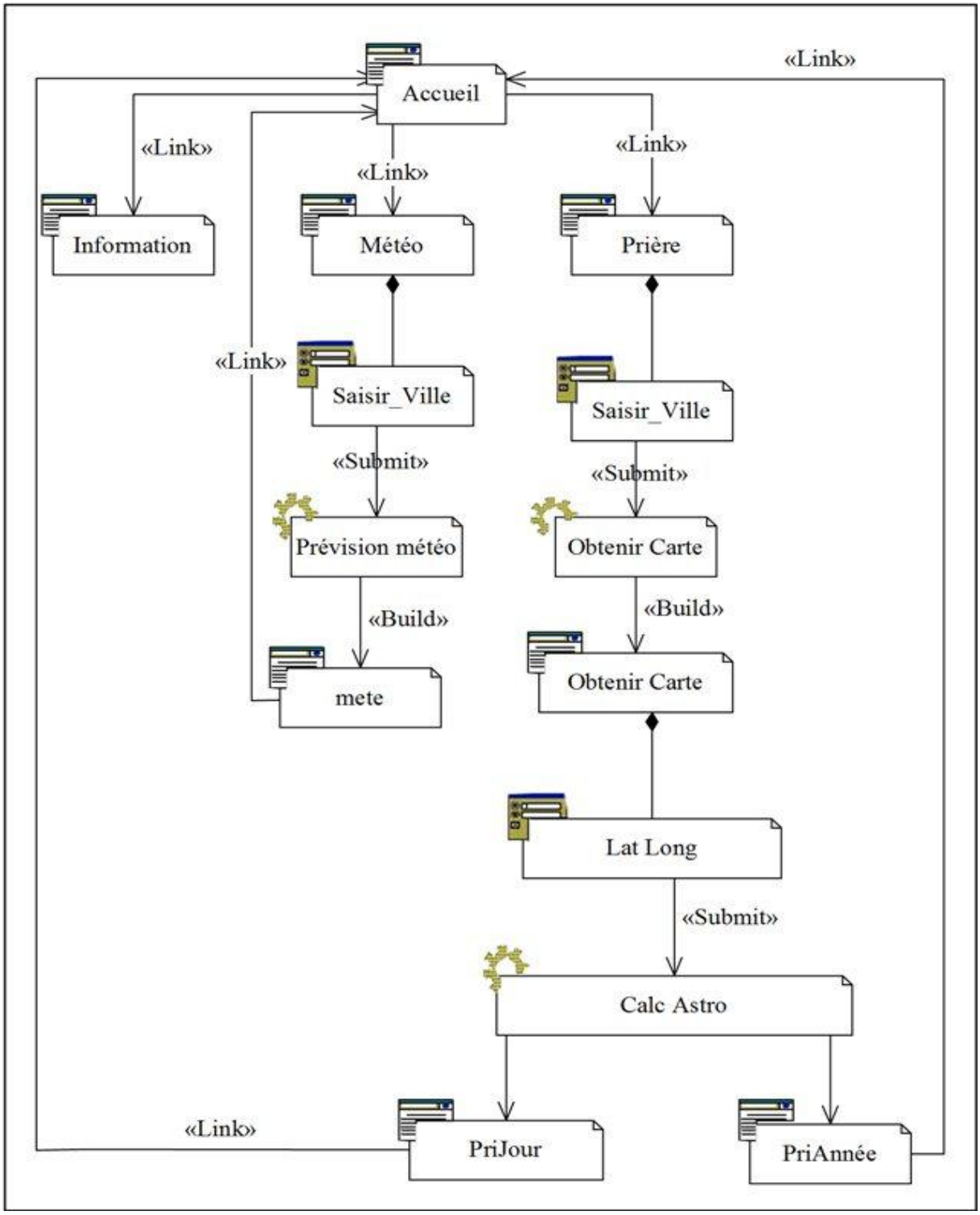

Figure 32: « Diagramme des classes de conception »

4.2. Diagramme de navigation:

UML nous offre la possibilité de représenter formellement la navigation, au moyen d'un diagramme d'états (ce que nous avons choisi) ou éventuellement d'un diagramme d'activité. L'utilisation de la modélisation UML pour la navigation nous permet en outre de la relier efficacement aux classes dialogues que nous avons identifiées.

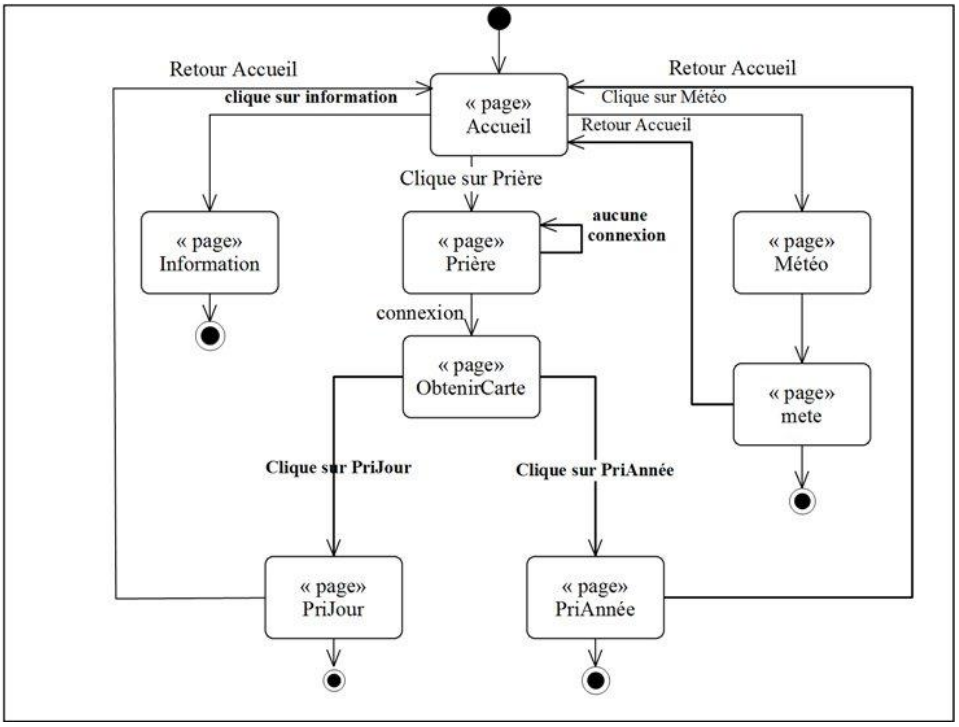

Figure 33: « Diagramme de navigation»

5. Architecture d'application :

Nous avons choisi de représenter l'architecture de l'application par un diagramme de déploiement montrant la distribution de tous les composants de l'application sur les différents postes. L'application peut très bien être hébergée sur un serveur d'application et accessible via internet ou implantée sur le poste client.

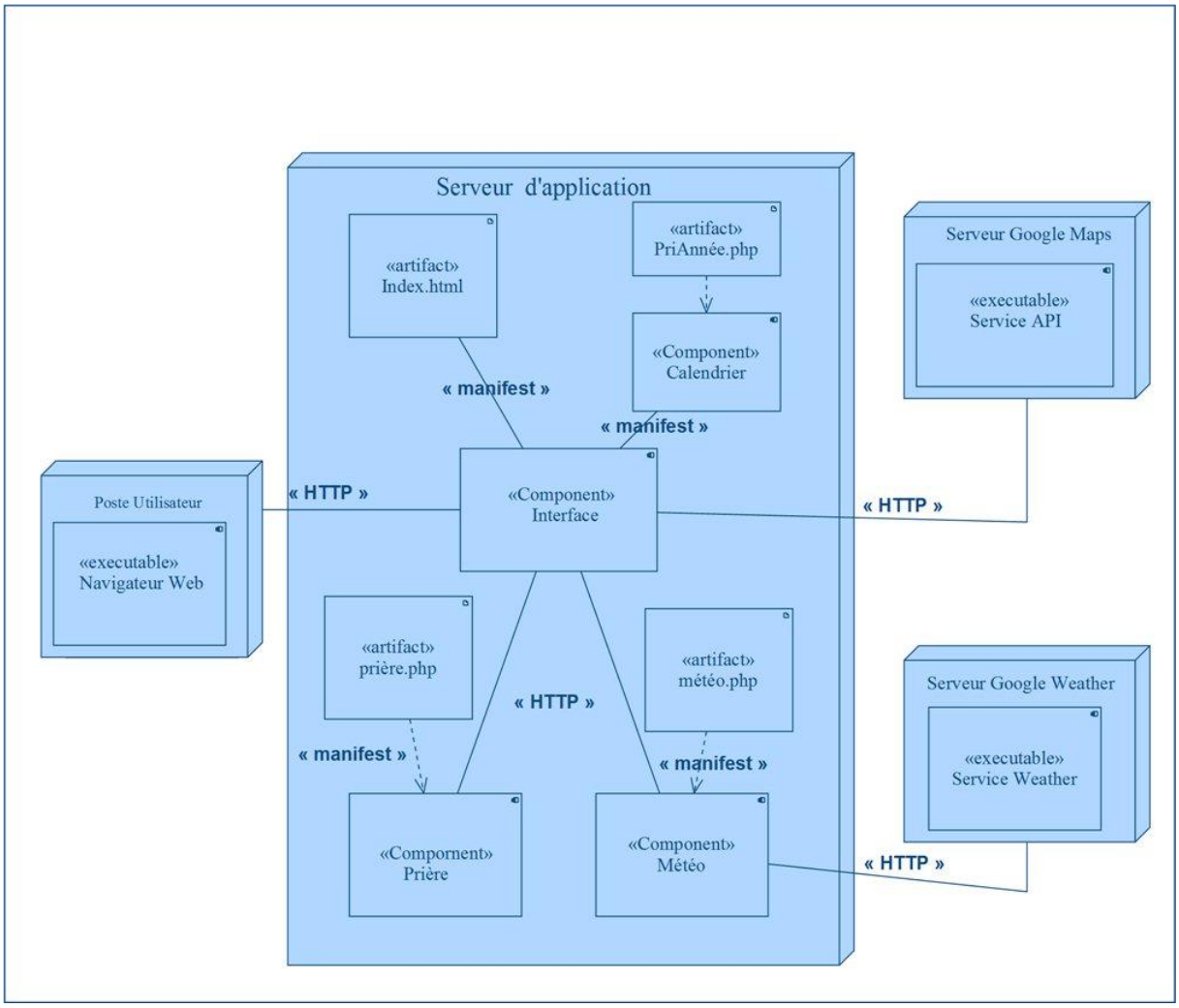

Figure 34: « Diagramme de déploiement »

Conclusion :

Pour la réalisation de ce chapitre, nous avons utilisé un langage de modélisation UML pour les applications Web, en suivant une démarche simplifiée et adaptée aux applications web inspirée du processus unifié (UP).

L'analyse et la conception détaillée nous permettent de bien représenter le système à réaliser. Aussi, elles nous facilitent la tâche de programmation.

Chapitre 5 :

Implémentation

Introduction:

Dans ce chapitre, nous abordons en premier lieu les aspects de programmation et les outils utilisés dans notre application. Nous expliquons ensuite une partie du code source de notre application que nous jugeons pertinentes (les fonctions les plus importantes).En fin, nous présentons les interfaces de l'application développée.

1. Langages de programmation:

Les langages de programmation qu'on a utilisés pour développer notre application ''HoPriMet'' sont :

- ✓ Le Langage HTML.
- ✓ Le Langage PHP.
- ✓ Le Langage JavaScript.

Nous présentons brièvement les caractéristiques de chacun d'eux.

1.1. Le Langage HTML :

HTML (Hyper Text Markup Langage) est langage de description de document. C'est un langage particulier issu de la norme SGML (Stadardized Generalized Markup Langage)qui définit des langages de balisage[8].

C'est le langage qui a été construit pour décrire les documents hypertextes sur Internet.HTML est normalisé par le W3C (consortium qui regroupe nombreuses organisations).

La structure générale d'un document HTML :

```
<HTML >
<HEAD><TITLE> titre de la page </ TITLE ></ HEAD>
<BODY><Contenu de la page></ BODY >
</ HTML >
```

1.2. Le Langage PHP :

Le langage PHP a été créé en 1994 par Rasmus Lerdorf pour les besoins des pages web personnelles (livre d'or, compteur, ect.). A l'époque, PHP signifiait Personnel Home Page. C'est un langage incrusté au HTML et interprété (PHP3) ou compilé (PHP4) coté serveur. Il est extensible grâce à de nombreux modules et son code source est ouvert. Comme il supporte tous les standards du web et qu'il est gratuit, il s'est rapidement répandu sur la toile. En 1997,

PHP devient un projet collectif et son interpréteur est réécrit par Zeev Suraski et Andi Gutmans pour donner la version 3 qui s'appelle désormais PHP: Hyper Text Preprocessor.

PHP est un langage de script qui s'incruste dans le langage HTML. Prenons l'exemple du traditionnel Hello World :

```
< HTML >
< HEAD >
< TITLE >Example</ TITLE >
</ HEAD>
<BODY>
<? PHP echo " Hello, World !! " ; ? >
</ BODY ></ HTML >
```

Résultat affiché:

```
Hello, World !!
```

1.3. Le Langage JavaScript:[9]

JavaScript est un langage de scripts incorporé aux balises Html qui permet d'améliorer la présentation et l'interactivité des pages Web.

JavaScript est donc une extension du code Html des pages Web. Les scripts, qui s'ajoutent ici aux balises Html, peuvent en quelque sorte être comparés aux macros d'un traitement de texte. Ces scripts vont être gérés et exécutés par le browser lui-même sans devoir faire appel aux ressources du serveur.

Ces instructions seront donc traitées en direct et surtout sans retard par le navigateur.

2. Outils de développement:

2.1. Wamp Server [10]

Wamp Server (anciennement **WAMP5**) est une plateforme de développement Web de type WAMP, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi que php MyAdmin pour l'administration Web des bases MySQL.

Il dispose d'une interface d'administration permettant de gérer et d'administrer ses serveurs au travers d'un icône de la barre (icône près de l'horloge de Windows).

La grande nouveauté de WampServer 2 réside dans la possibilité d'y installer et d'utiliser n'importe quelle version de PHP, Apache ou MySQL en un clic. Ainsi, chaque développeur peut reproduire fidèlement son serveur de production sur sa machine locale.

WAMP est un acronyme informatique signifiant :

Windows »,« **A**pache »,« **M**ySQL »,« **P**HP »

Les rôles de ces quatre composants sont les suivants :

Apache est le serveur web « frontal » : il est « devant » tous les autres et répond directement aux requêtes du client web (navigateur) ; le langage de script PHP sert la logique ;**M**ySQL stocke toutes les données de l'application ;et enfin **W**indows assure l'attribution des ressources à ces trois composants.

2.2. Un navigateur Web :

Dans notre application nous utilisons le navigateur '**Mozilla Firefox** '

Mozilla Firefox est un navigateur Web libre et gratuit, développé et distribué par la Mozilla fondation avec l'aide de milliers de bénévoles grâce aux méthodes de développement du logiciel libre/open source et à la liberté du code source.

Firefox est à l'origine un programme dérivé du logiciel Mozilla (actuellement connu sous le nom de SeaMonkey), mais reprenant uniquement les fonctions de navigation de celui-ci. Ce logiciel multiplate-forme est compatible avec diverses versions deWindows, Mac OS X et GNU/Linux (incluant Android). Il a été porté sur d'autres systèmes d'exploitation, ce qui est rendu possible par la mise à disposition de son code source sous trois licences libres différentes en même temps. Ce logiciel a connu un succès croissant depuis sa sortie, dépassant 1,2 milliard de téléchargements en janvier 2010. Même si ce nombre ne reflète pas le nombre réel d'utilisateurs du logiciel, Firefox est rapidement devenu le principal concurrent d'Internet Explorer, le navigateur Web de Microsoft. En décembre 2010, Firefox devient temporairement le navigateur le plus utilisé en Europe devant Internet Explorer et Google Chrome. Il se situe actuellement, selon une majorité des études réalisées en Europe et dans le monde, derrière le navigateur Google Chrome.

3. Implémentation:

3.1. Le service « Calcul des Horaires de prière » :

Pour mettre en œuvre ce service, nous avons utilisé le JavaScript pour le chargement de la carte géographique Google Maps afin d'assurer l'interaction dynamique et immédiate de l'utilisateur sans retour au serveur de notre application. Nous l'avons également utilisé pour récupérer les informations dont nous avons besoin, à savoir les coordonnées géographiques de la région concernée. Pour le calcul des horaires de prière et l'affichage des résultats nous avons utilisé le PHP. Pour les différentes interfaces graphiques que l'utilisateur peut voir et manipuler nous avons utilisé le HTML. Voici quelques précisions concernant ces différentes manipulations :

- **Création de la carte géographique :**

Pour la création de la carte géographique, nous avons utilisé des APIs fournies parGoogle Maps (en JavaScript).

Au début, il faut effectuer une connexion avec le serveur REST APIs Google Maps qui est situé à l'url « http://maps.google.com/maps/api » en précisant le type de réponse qui est dans notre cas « js » qui signifie JavaScript. Pour cela nous avons déclaré un bloc JavaScript vide qui pointe sur cette url comme c'est illustré dans le code suivant :

```
10
11  <script language="javascript" type="text/javascript"
12  src="http://maps.google.com/maps/api/js?sensor=false">
13  </script>
14
```

Puis définir la fonction qui va configurer la carte en définissant le niveau de zoom,le point où nous allons centrer la carte, nous donnons les coordonnées et nous précisons le type de la carte. Finalement cette fonction crée l'objet de la carte « *map*» (**map = new google.maps.Map (document.getElementById ('map_canvas'), myOptions);**) par l'API Map(identificateur , Options). Cette API prend comme paramètres : l'identificateur d'un dans la page HTML où nous allons afficher la carte, l'ensemble des paramètres de configuration de la carte. La séquence de code JavaScript qui réalise cette manipulation est comme suit :

```
66
67  function initialize() {
68        var formu = document.getElementById("formselect");
69        var Lat, Long;
70        if (formu.latitudeinput)    Lat= parseFloat(formu.latitudeinput.value);
71        if (formu.longitudeinput)   Long= parseFloat(formu.longitudeinput.value);
72        var latlng = new google.maps.LatLng(0,0);
73         var myOptions = { zoom: 12, center: latlng,  mapTypeId: google.maps.MapTypeId.ROADMAP  };
74        geocoder = new google.maps.Geocoder();
75        map = new google.maps.Map(document.getElementById('map_canvas'), myOptions);
76         }
77
```

Puis dans la partie BODY de la même page HTML nous faisons appel à la fonction créatrice de la carte : initialize()

```
18
19  <body onLoad="initialize()">
20
```

Une fois la carte affichée, nous procédons à la récupération des coordonnées géographiques de la région saisie.

```
53                  map.setCenter(results[0].geometry.location);
54
```

- **Le calcul des horaires :**

Pour le calcul des horaires de prière nous avons utilisé principalement le PHP.

Ce calcul se base principalement sur le calcul de l'équation du temps et la déclinaison solaire, selon l'algorithme présenté dans le chapitre 4. Mais à l'implémentation de cet algorithme nous avons été confrontés à un problème pour les angles g, q, L qui sont en degrés et pour le RA qui est en heures, où les valeurs peuvent dépasser la borne 360° pour les angles et 24 pour RA.

Pour résoudre ce problème nous avons proposé deux fonctions : fixAngle(α) e tfixHeur(h) qui permettent de réduire, respectivement, la valeur de α à l'intervalle [-360°, 360°] et la valeur de h (heures) à l'intervalle [0h, 23h]. Le code PHP suivant présente les deux fonctions :

```php
function fixangle($a)
{
    $a = $a - 360.0 * floor($a / 360.0);
    $a = $a < 0 ? $a + 360.0 : $a;
    return $a;
}
```

```php
function fixheur($a)
{
    $a = $a - 24.0 * floor($a / 24.0);
    $a = $a < 0 ? $a + 24.0 : $a;
    return $a;
}
```

Nous utilisons les deux fonctions précédentes (fixheur(), fixangle()) pour le calcul de l'équation du temps et la déclinaison solaire grâce au code PHP suivant :

```php
function PositionSolaire($jd)
{
    $D = $jd - 2451545.0;
    $g = $this->fixangle(357.529 + 0.98560028* $D);
    $q = $this->fixangle(280.459 + 0.98564736* $D);
    $L = $this->fixangle($q + 1.915* $this->dsin($g) + 0.020* $this->dsin(2*$g));

    $R = 1.00014 - 0.01671* $this->dcos($g) - 0.00014* $this->dcos(2*$g);
    $e = 23.439 - 0.00000036* $D;

    $d = $this->darcsin($this->dsin($e)* $this->dsin($L));
    $RA = $this->darctan2($this->dcos($e)* $this->dsin($L), $this->dcos($L))/ 15;
    $RA = $this->fixheur($RA);
    $EdT = $q/15 - $RA;

    return array($d, $EdT);
}
```

Pour afficher les Horaires de prière d'un jour actuel nous utilisons le code php suivant :

```php
<?php

    $CalculAstro = new CalculAstro($method);
    $date = strtotime($year. '-1-1');
    $finDate = strtotime(($year+ 1). '-1-1');
    $curentdate=date('M d');
    while ($date < $finDate)
    {
        $times = $CalculAstro->getHorairePriere($date, $latitudeinput, $longitudeinput, $fuseauHoraire);
        $day = date('M d', $date);
        if($curentdate==$day)
        print $day. "\t". implode("\t", $times). "\n";
        $date += 24* 60* 60;
        }
?>
```

4. Les interfaces de l'application:

Cette section présente quelques interfaces de l'application.

4.1.Page d'Accueil :

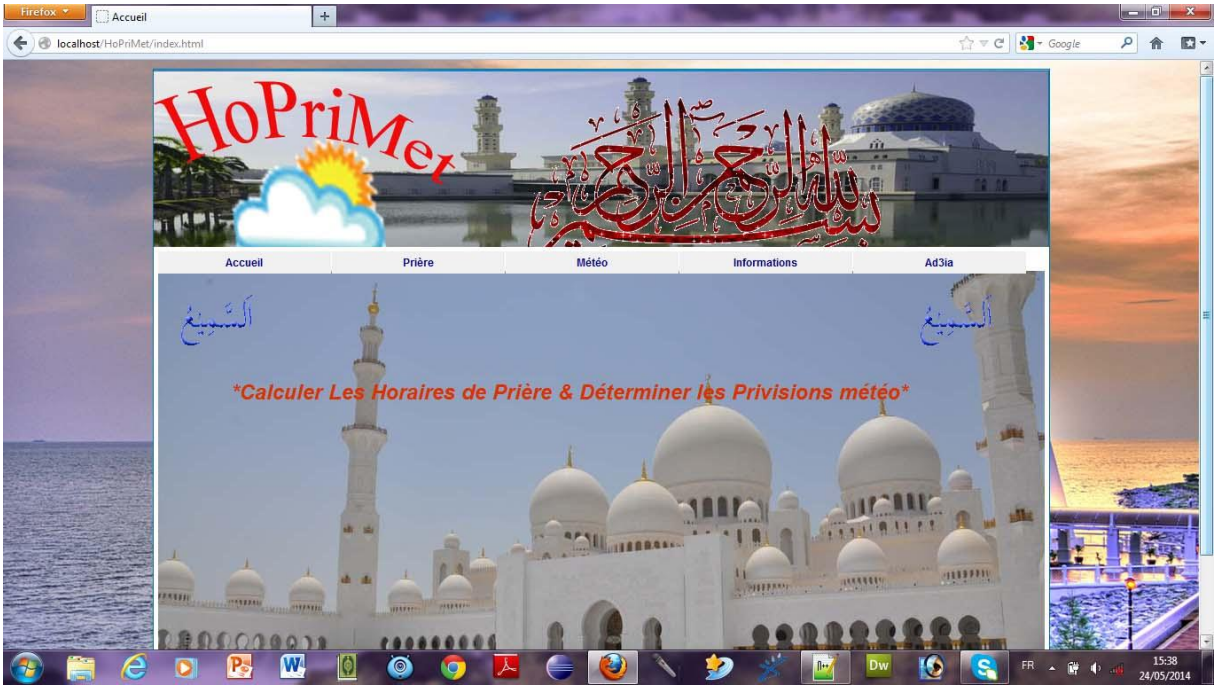

Figure 35: « Page d'Accueil »

4.2. Obtenir Carte :

Pour obtenir la carte, on saisit le pays ou la ville ; puis on invoque le service google maps pour obtenir carte. Une fois la carte avec les coordonnées affichées on choisit soit de calculer les horaires de prière du jour actuel ou de calculer le calendrier de prière de l'année actuelle.

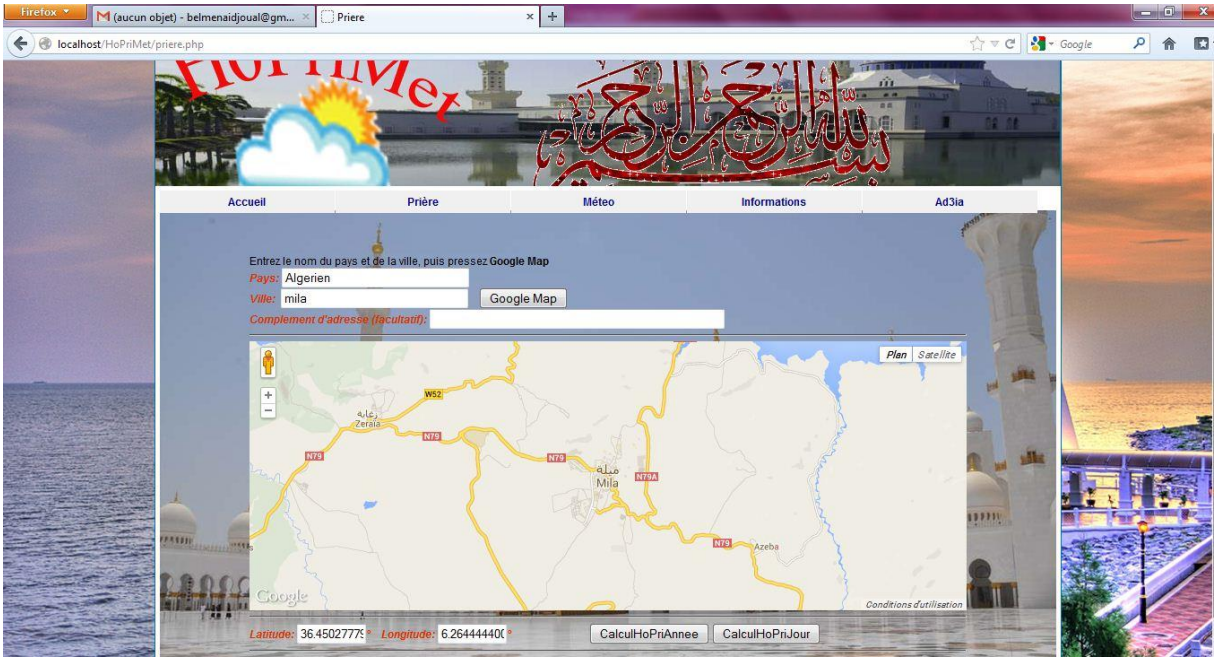

Figure 36: « Obtenir Carte »

4.3. Résultat de calcul du jour actuel :

Figure 37: «Résultat de calcul du jour actuel »

4.4. Résultat de calcul de l'année actuelle :

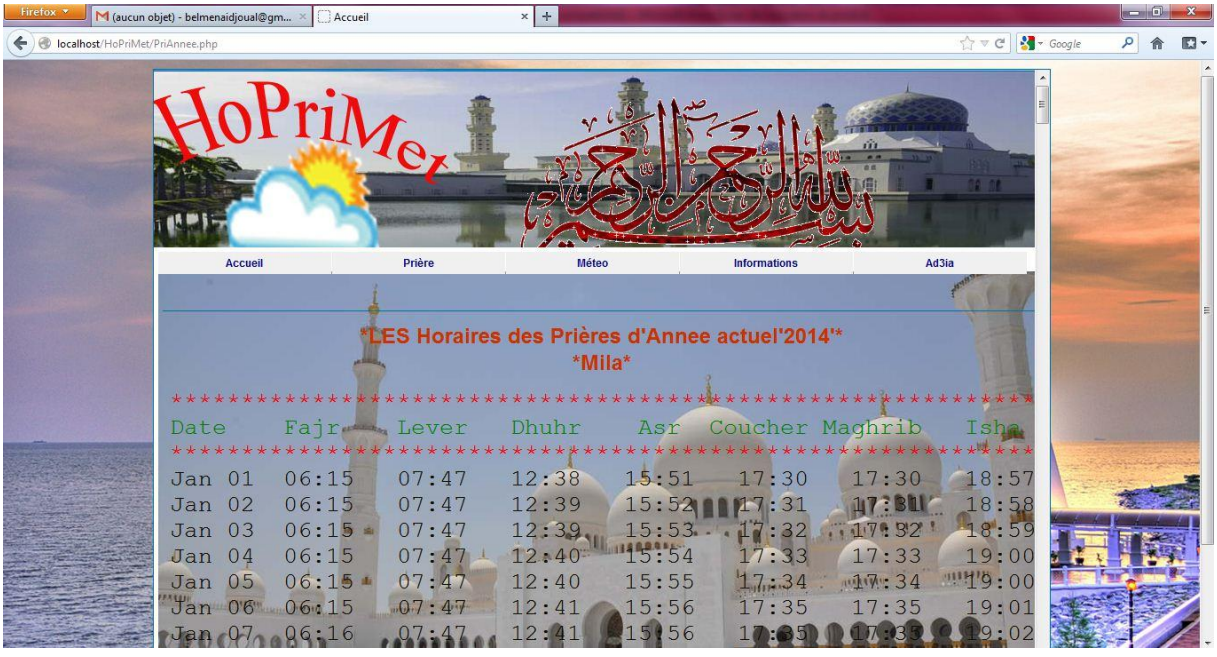

Figure 38: «Résultat de calcul de l'année actuelle »

4.5. Déterminer les prévisions météo :

L'utilisateur doit spécifier le nom de la ville pour avoir les prévisions actuelles de météo et celles descinq jours suivants.

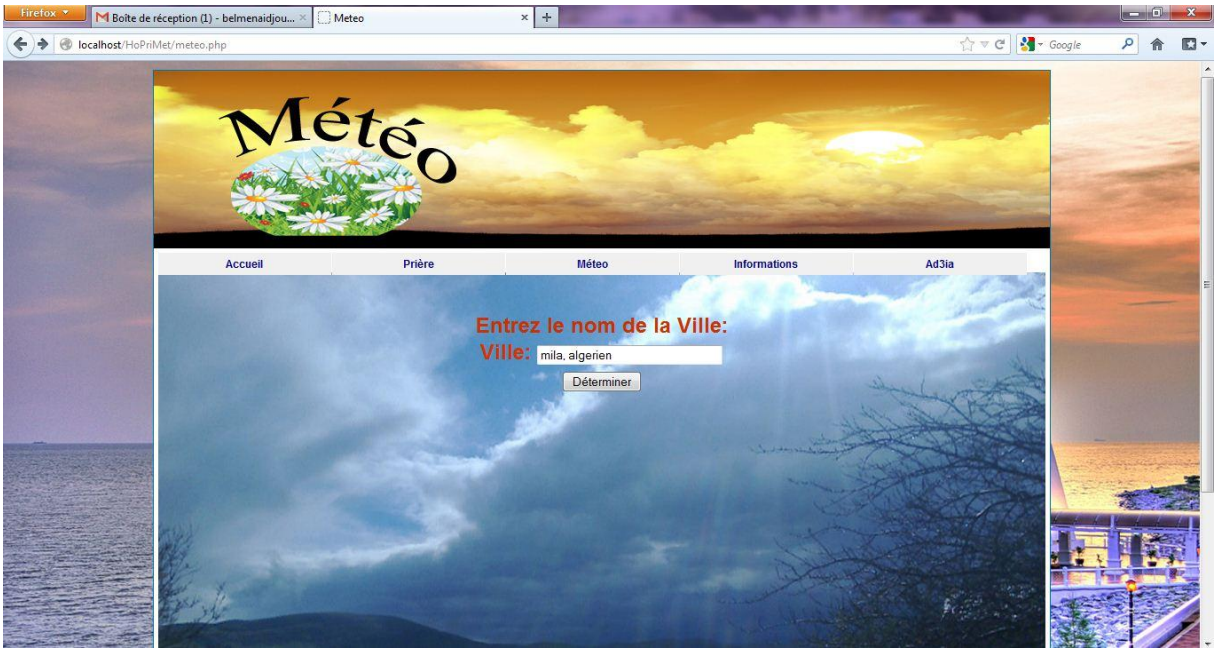

Figure 39: « Déterminer les prévisions météo »

4.6. Résultat des prévisions météo :

Figure 40: « Résultat des prévisions météo »

Conclusion :

Dans ce chapitre nous avons présenté la réalisation de notre application en précisant les outils de développement utilisés. Les principales interfaces de notre application "HoPriMet" ont été aussi présentés avec des exemples d'exécutions.

Conclusion Générale

Conclusion générale

Le travail que nous avons effectué consiste à concevoir et réaliser une application web HoPriMet qui offre à son utilisateur un des besoins nécessaires pour tous les musulmans à savoir, l'horaire de prière d'une position quelconque sur une carte géographique.

HoPriMet est accessible via le web et exploite les ressources, la performance et le coût raisonnable offerts par des services Google à savoir : Google Maps et Google Weather.

Pour nous aider à mener à terme notre projet, nous avons suivi une méthode simplifiée basée UML et adaptée au développement des applications web, conduite par les cas d'utilisation et composée de trois étapes : analyse, conception et implémentation.

Nous rappelons que l'application développée offre principalement les fonctionnalités suivantes : **HoPriMet** garantit ainsi la disponibilité de l'information pour le musulman pratiquant son culte en lui offrant la possibilité de pointer sur une carte n'importe quelle région du monde, d'en calculer les horaires de prière à la date de navigation, d'en afficher le calendrier annuel des horaires de prière et enfin d'afficher les prévisions météo de la région pointée sur la carte.

Au cours du développement de notre système nous avons acquis de nouvelles connaissances dans le domaine des réseaux et du web, notamment sur les techniques utilisées pour la publication des données et des ressources sur le web telles que l'UDDI et SOAP, ainsi que certaines notions du domaine de l'astronomie et leur utilisation dans la pratique de nos pratiques cultuelles.

L'application développée est entièrement opérationnelle concernant les fonctionnalités citées dans ce mémoire.

Références bibliographiques :

[Justin, 2002]: O'Sullivan Justin, Edmond David, Ter Hofstede Arthur, What's in a service Distrib. Parallel Databases ?, 12(2-3): 117–133, 2002.

[Colan, 2003]: IBM.BPEL4WS (version1.1), http://www.ibm.com /developerworks/library/ws-bpel, 2003.

[1]:https://aresu.dsi.cnrs.fr/IMG/pdf/failles_de_securite_v1-3.pdf

[2]: Mémoire de fin d'études : Evaluation des techniques de codage d'ontologies sur les performances de la composition de services Web

[3]:http://excerpts.numilog.com/books/9782100554034.pdf

[4]:http://fr.wikipedia.org/wiki/Google_Maps

[5]: http://maps.googleapis.com/maps/api/service/output?parameters

[6] :https://support.google.com/earth/answer/181050?hl=fr

[7] : UML2 Modéliser une application web

[8] : Mémoire de fin d'études :Conception et réalisation d'un site web dynamique de partage de photos. Promotion 2008-2009.

[9] : www.lehtml.com/download/js_doc.pdf

[10] : http://fr.wikipedia.org/wiki/WampServer