

République Algérienne Démocratique et Populaire

Université Abou Bakr Belkaid– Tlemcen

Faculté de Technologie

Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Licence en Informatique

Thème

Gestion de la prise d'un rendez-vous

Réalisé par :

- KAZI AOUEL Wafaa.
- BELBACHIR Leila.

Encadré par :

- HALFAOUI Amel.

Présenté le 10 Juin 2014 devant la commission d'examination composée de MM.

- Benamar.A (examineur)
- Belhabi.A (examinatrice)

Résumé

A travers ce projet, nous avons réalisé un site web dynamique facile à s'intégrer dans l'environnement de travail de multiples secteurs médicaux, commerciaux et administratifs. Cette application se traduit par la mise en œuvre d'un système qui facilite la prise d'un rendez-vous baptisé MyRDV. Notre système vise essentiellement à laisser libre accès au client de prendre le rendez-vous à l'heure qui l'arrange et d'éliminer le temps d'attente des clients entre les rendez-vous. La modélisation du système MyRDV a été faite par UML en s'appuyant sur le processus UP. La base de données et l'application ont été respectivement créées par le SGBD Oracle et le langage PHP, réputés par leur puissance et efficacité.

Mots clés : Rendez-vous, UML, PHP.

Abstract

Through this project, we wanted to make an easy dynamic web site to integrate into the work environment of multiple medical sectors, commercial and administrative. This application results in implementing a system that MyRDV essentially aims to reduce the daily care of people by reducing customer waiting time in making an appointment. Modeling MyRDV system was made by UML based on the UP process. The data base and application were respectively created by the Oracle DBMS and PHP, known for their power and effectiveness.

Tags: Appointment, UML, PHP.

Remerciements

En témoignage de notre gratitude et de notre profond respect nous adressons nos vifs et sincères remerciements à :

Dieu miséricordieux, pour son aide et sa bénédiction.

A notre encadreur Madame Halfaoui pour ses conseils, sa disponibilité et son encouragement qui nous ont permis de réaliser ce travail dans les meilleures conditions.

Les jurys pour leurs efforts et leur soin apporté à notre travail.

Aux enseignants de notre université et département informatique.

Nous ne pouvons nommer ici toutes les personnes qui de près ou de loin nous ont aidé et encouragé mais nous les en remercions vivement.

Enfin, nous tenons à dire combien le soutien quotidien de notre famille a été important tout au long de ces quelques années, nous leur devons beaucoup.

Table des matières

Table des illustrations

Introduction générale	1
Chapitre1 : Description du système MyRDV et du langage UML.	
I. Introduction :.....	2
II. Définition du système:	2
III.Les fonctionnalités du système:	2
IV. Le langage de modélisation UML:	5
IV.1. Une notion sur la programmation orientée objet :	4
IV.2.Intérêt d'une méthode objet:	4
IV.3. Notion sur l'UML :	5
V. Les diagrammes UML	6
VI. Processus Unifié(UP) :.....	7
VII.CONCLUSION :	8
Chapitre2 : Conception du Système MyRDV	
I.Introduction :.....	11
II.Description détaillée du système :	171
III. Modèle logique de données MyRDV.....	18
IV. Conclusion :	19
Chapitre3 : Implémentation du Système MyRDV	
I.Introduction.....	Erreur ! Signet non défini.20
II. Définition d'un site web :	280
III.Choix du langage de programmation :	280
IV. Choix de l'outil de développement :	22
V.Choix du SGBD :	23
VI. Description de l'application :.....	24
VII. Conclusion :	28
Conclusion générale	29
Références Webographies	30

Table des illustrations

Figure 1 : diagramme de cas d'utilisation de la gestion d'un RDV.....	10
Figure2 :Diagramme de séquence de la partie réservation/authentification.....	13
Figure 3 .Diagramme de séquence partie réserver/modifier.....	14
Figure 4: Diagramme de séquence réserver/annuler.....	15
Figure 5: Diagramme de classe de la gestion des RDV.....	16
Figure 6: code déclencheur attribuer statut RDV.....	17
Figure 7: Code trigger incrémentation automatique du code client.....	18
Figure 8: Capture d'écran d'accueil du site MyRDV.....	24
Figure 9: Capture d'écran de la prise d'un rendez-vous sur le site MyRDV.....	25
Figure 10: Capture d'écran d'inscription sur le site MyRDV.....	25
Figure 10: Capture d'écran de la page d'à propos du site MyRDV.....	26
Figure 11: Capture d'écran de la position de contact du site MyRDV.....	27
Figure 12: Capture d'écran du formulaire de contact du site MyRDV.....	27

Introduction générale

Il ne fait désormais plus aucun doute que l'informatique est la révolution la plus importante et la plus innovante qui a marqué la vie de l'humanité moderne. En effet, les logiciels informatiques proposent maintenant des solutions à tous les problèmes de la vie, aussi bien dans des domaines professionnels que pour des applications personnelles. Leurs méthodes de conception et de développement ont vu l'évènement d'autant de technologies qui facilitent leur mise en place et leurs donnent des possibilités et des fonctionnalités de plus en plus étendues, et surtout avec l'interaction d'internet qui est l'outil par excellence pour se faire connaître de tous .C'est aussi le meilleur moyen pour les organismes ou les entreprises de vendre leurs produits et services. Les particuliers, les grandes structures commerciales, les sociétés aussi bien privées que publics bref tous les secteurs pour se faire connaître ou mieux servir leur clientèle utilisent les médias (radio, télévision, ...). Mais tous ces différents moyens précédemment cités ont une portée faible (rayon d'action très limités) et ne donnent pas la possibilité aux usagers de pouvoir interagir avec l'organisation. De plus Internet aujourd'hui occupe une place de choix dans la vie de tous les jours à travers ses multiples services et plus particulièrement son gain de temps. On a tenu compte de cette innovation et de tous ces opportunités qu'elles présentent tel que la création d'un site web dynamique qu'on l'en a opté pour notre projet de fin d'étude.

Ainsi, l'objectif de notre projet est de réaliser une application informatique (un site web dynamique) interactive, fiable, conviviale et facile à s'intégrer dans l'environnement de travail de multiples secteurs médicaux, commerciaux et administratifs. Cette application vise essentiellement à diminuer la charge quotidienne des gens en diminuant le temps d'attente des clients dans la prise d'un rendez-vous.

Plan du document

Le document est organisé en chapitres. Le découpage a été fait de manière à couvrir les domaines impliqués dans notre travail.

Le **chapitre 1** est consacré à la description de notre application ainsi la présentation UML (les définitions, les descriptions, et l'utilité des différents digrammes).

Le **chapitre 2** est consacré à l'étude de notre système en présentant les digrammes montrant la conception de notre projet.

Le **chapitre 3** constitue la partie réalisation et implémentation du site web, nous expliquerons le choix du langage de programmation, en décrivant d'une manière générale l'application réalisée ainsi qu'une illustration graphique de l'application du point de vue profil d'un utilisateur donné, avec une description des choix ergonomiques adoptés.

**Chapitre 1 : Description du système
MyRDV et du langage UML.**

I. Introduction :

Dans la vie quotidienne, chaque individu se retrouve face à un emploi du temps chargé à savoir régler des factures, faire des courses...etc. En plus de cela, il doit attendre des heures pour prendre un rendez-vous, il y a des jours où l'on redoute de devoir prendre un rendez-vous chez un médecin spécialiste à l'hôpital par exemple, non par peur des résultats des examens, mais plutôt parce qu'on craint le temps d'attente pour qu'en fin de compte aucun des rendez-vous proposés ne nous convient ou bien qu'il est indisponible. Pour pallier ce manque de flexibilité et rapidité, on a pensé à créer une application web qui consiste à prendre un rendez-vous en ligne en quelques clics seulement dans le but de garantir une meilleure gestion du quotidien.

II. Définition du système:

Notre projet vise à mieux gérer la prise d'un rendez-vous, il représente un site web dynamique sous la forme d'une plate-forme, qui permet une planification efficace et évolutive au sein de l'organisation et fournit un confort supplémentaire au client lorsqu'il s'agit de prendre un rendez-vous, que ce soit pour un voyage, une place de concert ou pour un séminaire, nombreuses sont les personnes qui ont l'habitude de réserver en ligne. Avec ce site, le client dispose enfin d'un service comparable, disponible 24h/24 et 7 jours/7. Il peut désormais choisir une plage horaire à sa meilleure convenance et, le cas échéant, annuler un rendez-vous à temps.

Il existe d'autres sites concurrents tels que: www.Prendre-Mon-RDV.com, www.clicrdv.com mais l'inconvénient qu'ils présentent est: leurs accès non fonctionnel en Algérie.

III. Les fonctionnalités du système:

Notre application comprendra les fonctionnalités suivantes :

- Accès sécurisé par login et mot de passe crypté.
- Possibilité de sélectionner un rendez-vous par discipline, voire par médecin.

Chapitre 1 : Description du système MyRDV et du langage UML

- Possibilité de filtres complémentaires : choix d'un jour, d'une tranche horaire, ou d'une plage de jours ou d'heures particulier.
- Affichage de la liste des rendez-vous disponibles correspondant aux critères.
- Gestion des rendez-vous (modification, suppression)
- Gestion de fiches contact clients ou patients (recherche, création, modification, suppression).
- Gestion des horaires de travail (ouverture, fermeture de plages horaires).
- Statistiques (relevé global et détaillé des rendez-vous)
- Impression d'un relevé des rendez-vous (utile pour l'administrateur).
- Export des rendez-vous vers un fichier Excel (utile pour l'administrateur).
- Affichage des consignes particulières liées au type de rendez-vous.
- Dispose la fonction de rappel de rendez-vous par SMS ou par mail, le site web contribue à réduire le nombre de désistements de dernière minute.
- Etre au courant si le RDV a été confirmé à partir de votre site internet.

IV. Le langage de modélisation UML:

Pour faire face à la complexité croissante des systèmes d'information, de nouvelles méthodes, langages et outils ont été créés. La principale avancée réside dans la programmation orientée objet (P.O.O.) dont le langage de modélisation UML est inclus. Nous allons commencer par donner une notion sur ce langage ; ainsi le processus UP qui utilise UML. La réalisation d'un site web doit être impérativement précédée d'une méthodologie d'analyse et de conception qui a pour objectif de permettre de formaliser les étapes préliminaires du développement d'un site afin de rendre ce développement plus fidèle aux besoins du client.

IV.1. Une notion sur la programmation orientée objet :

La programmation orientée objet consiste à modéliser informatiquement un ensemble d'éléments d'une partie du monde réel (que l'on appelle *domaine*) en un ensemble d'entités informatiques. Ces entités informatiques sont appelées *objet*. Il s'agit de données informatiques regroupant les principales caractéristiques des éléments du monde réel (taille, la couleur, ...).

L'approche objet est une idée qui a désormais fait ses preuves. Simula a été le premier langage de programmation à implémenter le concept de classes en 1967 ! En 1976, Smalltalk implémente les concepts d'encapsulation, d'agrégation, et d'héritage (les principaux concepts de l'approche objet). D'autre part, de nombreux langages orientés objets ont été mis au point dans un but universitaire (Eiffel, Objective C, Loops...).

Un objet représente une entité du monde réel, ou de monde virtuel dans le cas d'objets immatériels, qui se caractérisent par une identité, des états significatifs et par un comportement. L'identité d'un objet permet de distinguer les objets les uns par rapport aux autres. Son état correspond aux valeurs de tous les attributs à un instant donné. Ces propriétés sont définies dans la classe d'appartenance de l'objet. Enfin, le comportement d'un objet se définit par l'ensemble des opérations qu'il peut exécuter en réaction aux messages envoyés (un message = demande d'exécution d'une opération) par les autres objets. Ces opérations sont définies dans la classe d'appartenance de l'objet. A partir de cette année, Rumbaugh et Booch (rejoints en 1995 par Jacobson) ont unis leurs efforts, pour mettre au point la méthode unifiée (Unified Method 0.8), incorporant les avantages de chacune des méthodes précédentes. La méthode unifiée à partir de la version 1.0 devient UML, soumis à l'OMG (Object Management Group) en Janvier 1997, et acceptée en novembre 1997 dans sa version 1.1, date à partir de laquelle UML devient un standard international. La version qui à vue nos jours est la version 2.0 d'UML.

IV.2. Intérêt d'une méthode objet:

Les langages orientés objet constituent chacun une manière spécifique d'implémenter le paradigme objet. Ainsi, une méthode objet permet de définir le problème à haut niveau sans rentrer dans les spécificités d'un langage. Il représente ainsi un outil permettant de

définir un problème de façon graphique, afin par exemple de le présenter à tous les acteurs d'un projet (n'étant pas forcément des experts en un langage de programmation). De plus, le fait de programmer à l'aide d'un langage orienté objet ne fait pas d'un programmeur un concepteur objet. En effet il est tout à fait possible de produire un code syntaxiquement juste sans pour autant adopter une approche objet. Ainsi la programmation orientée objet implique

- en premier lieu une conception abstraite d'un modèle objet (c'est le rôle de la méthode objet)
- en second plan l'implémentation à l'aide d'un langage orienté objet (tel que C++/Java/...)

IV.3. Notion sur l'UML :

IV.3.1. Définition :

«UnifiedModelingLanguage » est un langage unifié de modélisation objets. Ce n'est pas une méthode, il ne donne pas de solution pour la mise en œuvre d'un projet. C'est avant tout un **formalisme graphique** issu de notations employées dans différentes méthodes objets. [1]

IV.3.2. A quoi sert UML ?

UML utilise l'approche objet en présentant un langage de description universel. Il permet grâce à un ensemble de diagrammes très explicites, de représenter l'architecture et le fonctionnement des systèmes informatiques complexes en tenant compte des relations entre les concepts utilisés et l'implémentation qui en découle. [2]

UML est avant tout un support de communication performant, qui facilite la représentation et la compréhension de solutions objet :

- ✓ Sa notation graphique permet d'exprimer visuellement une solution objet, ce qui facilite la comparaison et l'évaluation de solutions.
- ✓ L'aspect formel de sa notation, limite les ambiguïtés et les incompréhensions.
- ✓ Son indépendance par rapport aux langages de programmation, aux domaines d'application et aux processus, en fait un langage universel.

UML est donc bien plus qu'un simple outil qui permet de "dessiner" des représentations mentales... Il permet de parler un langage commun, normalisé mais accessible, car visuel. Il représente un juste milieu entre langage mathématique et naturel, pas trop complexe mais suffisamment rigoureux, car basé sur un méta modèle. Une autre caractéristique importante d'UML, est qu'il cadre l'analyse. UML permet de représenter un système selon différentes vues complémentaires : les diagrammes.

V. Les diagrammes UML

Un diagramme UML est une représentation graphique, qui s'intéresse à un aspect précis du modèle ; c'est une perspective du modèle. Il existe deux types de diagramme :

V.1. Diagrammes structurels

On citera :

V.1.1. Diagramme de classes :

Il représente les classes intervenant dans le système. Comme dans l'exemple

V.1.2. Diagramme d'objets :

Il sert à représenter les instances de classes (objets) utilisées dans le système.

V.1.3. Diagramme des cas d'utilisation :

Il permet d'identifier les possibilités d'interaction entre le système et les acteurs (intervenants extérieurs au système), c'est-à-dire toutes les fonctionnalités que doit fournir le système.

V.1.4. Diagramme d'activité :

Il permet de décrire sous forme de flux ou d'enchaînement d'activités le comportement du système ou de ses composants.

Comme il y a d'autres diagrammes :

- Diagramme de composants.
- Diagramme de déploiement.
- Diagramme des paquetages.
- Diagramme de structure composite.
- Diagramme états-transitions.
- Diagrammes comportementaux.

V.2. Diagrammes d'interaction (dynamique)

- Diagramme de séquence :

Représentation séquentielle du déroulement des traitements et des interactions entre les éléments du système et/ou de ses acteurs.

- Diagramme de communication
- Diagramme global d'interaction
- Diagramme de temps

VI. Processus Unifié(UP) :

VI.1. Définition :[3]

Pour définir le processus unifié, nous allons simplement définir les deux termes qui le composent :

- **Processus** : Suite continue d'opérations constituant la manière de fabriquer. En d'autres termes, c'est une succession de tâches dans le but d'accomplir un travail, un projet.
- **Unifié** : Etre amené à l'unité, se fondre en un tout. En fait, les méthodes d'analyse et de conception orientées objet, étaient variées jusqu'à ce que Rumbaugh, Jacobson et Booch eut l'idée de les unifier.

VI.2. Caractéristiques :

D'après les auteurs d'UML, un processus de développement qui possède ces qualités devraient favoriser la réussite d'un projet. Cependant, dans le cadre de la modélisation d'une application informatique, les auteurs d'UML préconisent d'utiliser une démarche [4]:

- Centrée sur l'architecture logicielle.
- Itérative et incrémentale : s'y prendre en plusieurs fois, en affinant son analyse par étapes garantira une meilleure modélisation du système.
- Guidée par les besoins des utilisateurs du système :

Avec UML, ce sont les utilisateurs qui guident la définition des modèles :

Chapitre 1 : Description du système MyRDV et du langage UML

- Le périmètre du système à modéliser est défini par les besoins des utilisateurs (les utilisateurs définissent ce que doit être le système).
- -Le but du système à modéliser est de répondre aux besoins de ses utilisateurs (les utilisateurs sont les clients du système).
- Les besoins des utilisateurs servent aussi de fil rouge, tout au long du cycle de développement (itératif et incrémental) :
- -A chaque itération de la phase d'analyse, on clarifie, affine et valide les besoins des utilisateurs.
- A chaque itération de la phase de conception et de réalisation, on veille à la prise en compte des besoins des utilisateurs.
- A chaque itération de la phase de test, on vérifie que les besoins des utilisateurs sont satisfaits.

VII.CONCLUSION :

Comme UML n'impose pas de méthode de travail particulière, il peut être intégré à n'importe quel processus de développement logiciel de manière transparente. Parmi les processus cités précédemment, nous avons choisi le processus UP qu'on utilisera pour modéliser notre système. Cette modélisation sera détaillée dans le chapitre suivant.

Chapitre2 : Conception du Système

MyRDV

I. Introduction :

Dans cette partie, nous avons déterminé un premier découpage de l'application, et établit les différentes étapes de conception comme nous l'avons déjà cité. Les diagrammes appropriés pour notre projet sont :

- Diagramme de cas d'utilisation qui nous aide dans la spécification des besoins de notre système.
- Diagramme de séquence qui nous donne le scénario des actions et interactions entre les acteurs et notre système.
- Diagramme de classes qui définit l'architecture statique de notre système.

II. Description détaillée du système :

II.1. Diagramme de cas d'utilisation :

Comme son nom l'indique, un cas d'utilisation représente une fonctionnalité du système. Ce diagramme nous permet d'identifier toutes les possibilités d'interaction entre le système et les acteurs, ce qui veut dire toutes les fonctionnalités que le système doit fournir.

La figure suivante représente le diagramme de cas d'utilisation pour le système MyRDV :

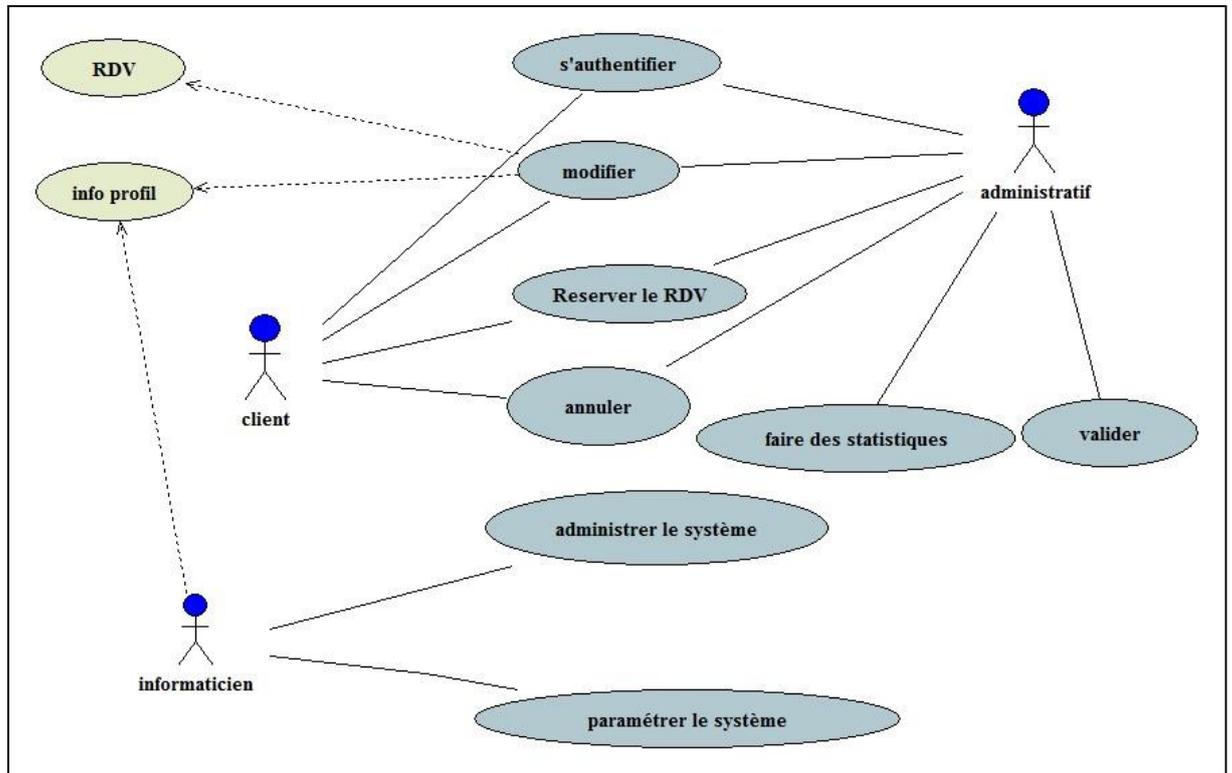


Figure 1: Diagramme de cas d'utilisation de la gestion des RDV

II.1.1. Identification des acteurs dans MyRDV :

Un acteur est une personne interagissant avec le système. Chaque acteur est identifié par un rôle. Ces rôles décrivent les capacités et les besoins de l'acteur. Les personnes interagissant avec le système MyRDV sont :

- le client,
- l'administratif,
- l'informaticien.

II.1.2.Description des interactions :

❖ Réserver un RDV :

L'objectif de notre site web est prendre un rendez-vous cela se fait en se connectant sur le site et en choisissant la date et le lieu voulu

Le rendez-vous sera automatiquement enregistré dans notre base de données avec toutes les informations nécessaires (comme le nom du client, date, heure) et aussi il sera mis en statut « En attente », il sera validé ou annulé par la suite.

❖ S'authentifier :

Après avoir visité notre site et choisit le rendez-vous adéquat, le client doit remplir un formulaire. Ce formulaire nous servira à l'enregistrer sur notre base de données et de le contacter en cas de besoin (modification, annulation, validation, ...).

L'authentification se fait qu'une seule fois. Le client recevra un identifiant et un mot de passe. La prochaine connexion sur le site sera prise en compte avec son identifiant.

❖ Annuler :

Le client a le droit d'annuler son rendez-vous 48 heures avant sa date prévue. Il recevra un message d'annulation pour confirmer son annulation.

En laissant le client annulait le rendez-vous, on aura la possibilité de recontacter les clients qui ont voulu avoir cette date maintenant libérée. Le premier avoir répondu à notre contact aura le rendez-vous validé directement.

❖ Modifier :

Le client a le droit également de faire des modifications :

- Il peut modifier l'heure, la date, le lieu de son rendez-vous, en parallèle le système vérifiera si les modifications sont possibles (date, heure, lieu libres) c'est juste après qu'il aura le choix de faire ou non sa modification.
- Il peut aussi modifier ses informations personnelles (info profils) qui seront automatiquement prises en compte, et notre base de données sera mise à jour.

✓ **Remarque :**

Toutes ces fonctionnalités précédentes sont occupées par le client comme elles sont occupées pas l'administratif de la même manière. Sauf que la différence s'effectue dans le lieu :

Le client reste chez lui et prend son rendez-vous qu'en se connectant au site contrairement à l'administratif qui est coté administration et exécute les commandes désirées du client.

❖ **Valider :**

Comme on l'a déjà dit dans la partie « Réserver le RDV », le statut du rendez-vous est mis par défaut « En attente ». Cette attente peut durer quelques jours avant que l'administratif système le valide. Le client recevra alors un message de confirmation du rendez-vous.

❖ **Faire des statistiques :**

Comme tout organisme et système, des statistiques sont faites pour évaluer leur travail mensuel ou annuel. Cela nous aidera à améliorer nos fonctionnalités et nos capacités. Ce travail est le rôle de l'administratif. Il aura juste besoin de l'application coté administration. Il a à sa disposition toutes les fonctions nécessaires dans les diverses statistiques.

❖ **Paramétrer le système :**

L'informaticien a pour rôle de faire un ensemble de paramétrage, qui suivront le système à administrer, et qui devront aussi être aussi souple que possible.

❖ **Administrer le système :**

Cela dépendra du cahier de charge de chaque administration (les conditions changent d'une administration à une autre)

II.2. Diagramme de séquence :

Dans notre projet, on a définis trois diagrammes de séquence. Ces diagrammes représentent toutes les actions et interactions entre acteurs et système selon un ordre chronologique.

- Réserver et s'authentifier :

Ce diagramme représente la partie où le client choisit le rendez-vous approprié tout en respectant la disponibilité des lieux et des dates, ainsi que l'authentification si le rendez-vous est réservé.

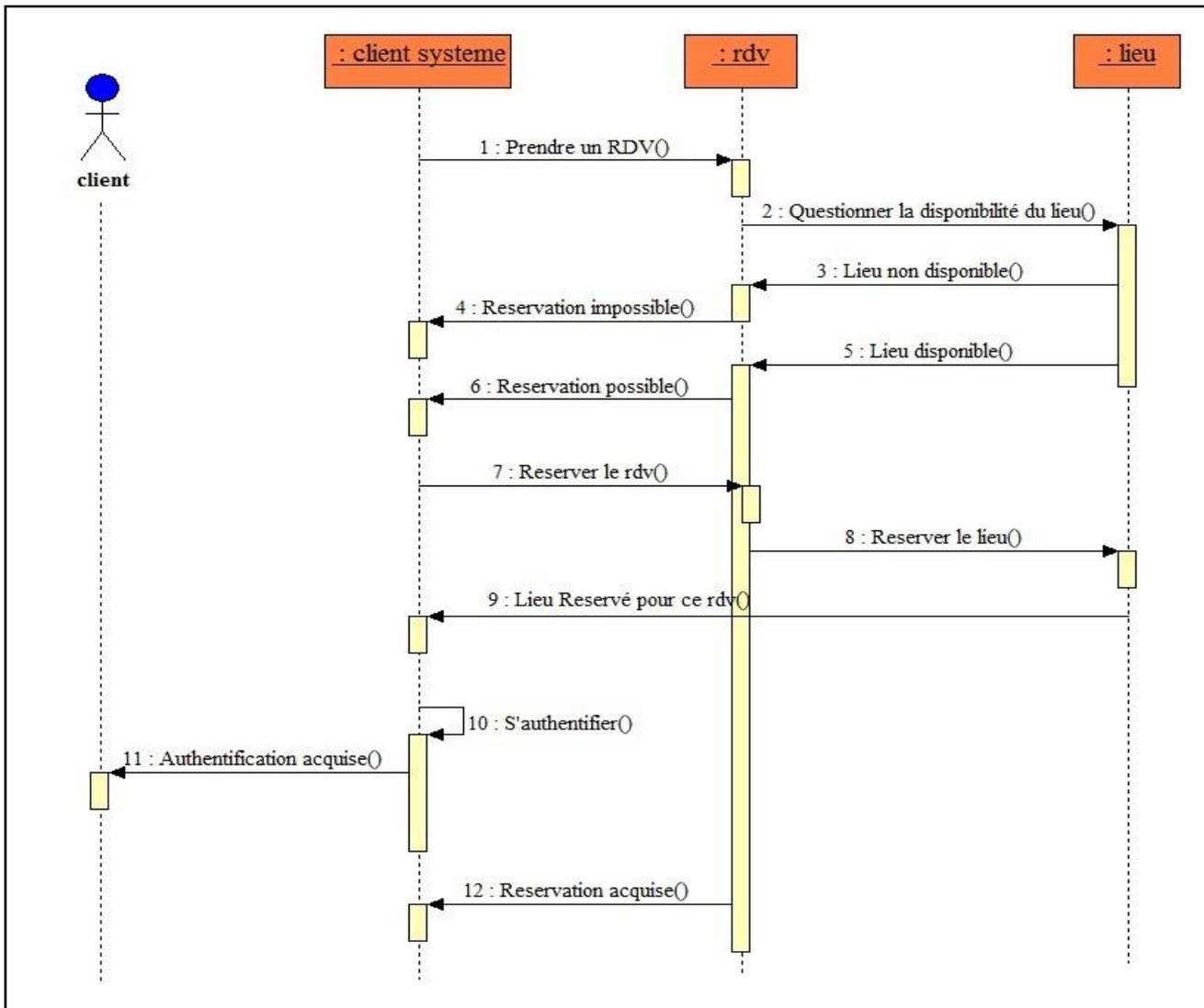


Figure 2 : Diagramme de séquence de la partie réservation/authentification

- **Réserver et modifier :**

Ce diagramme représente la partie de la modification après réservation évidemment.

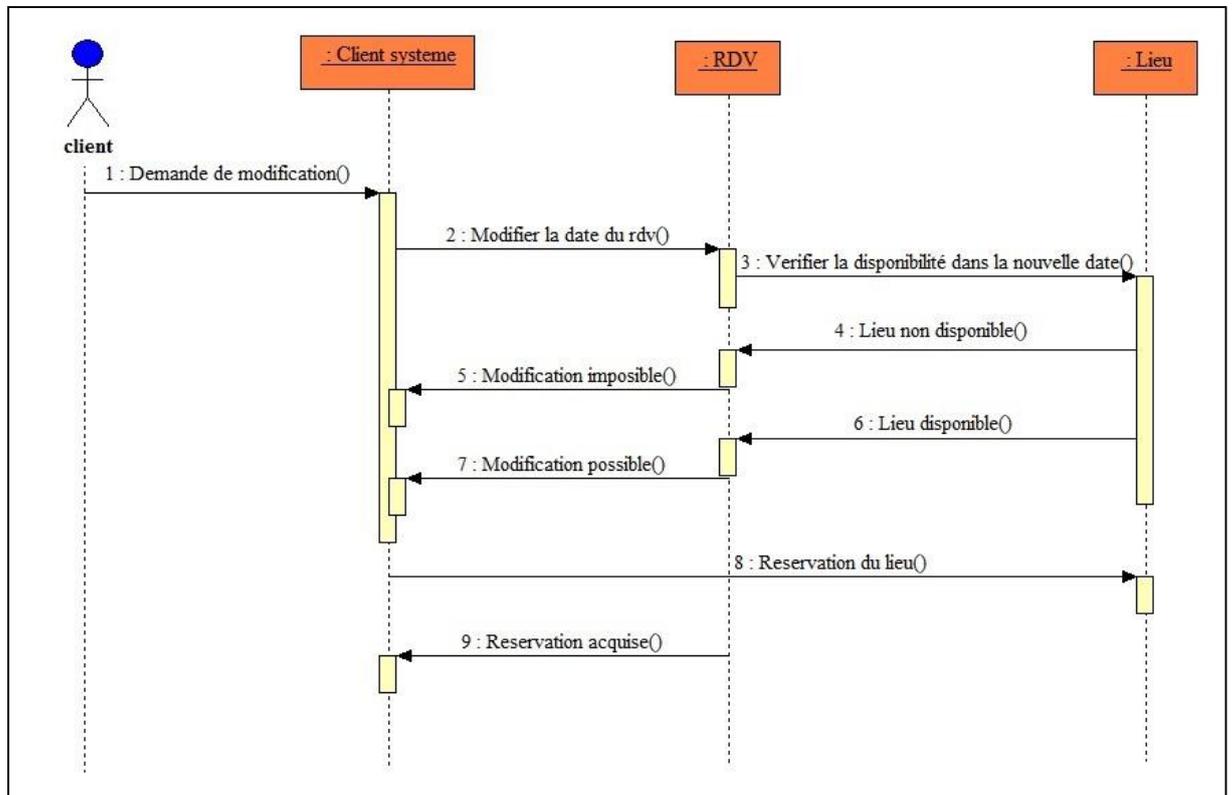


Figure 3: Diagramme de séquence partie réserver/modifier

- **Réserver/annuler :**

Ce diagramme représente la partie où le client annule le rendez-vous après l'avoir réserver.

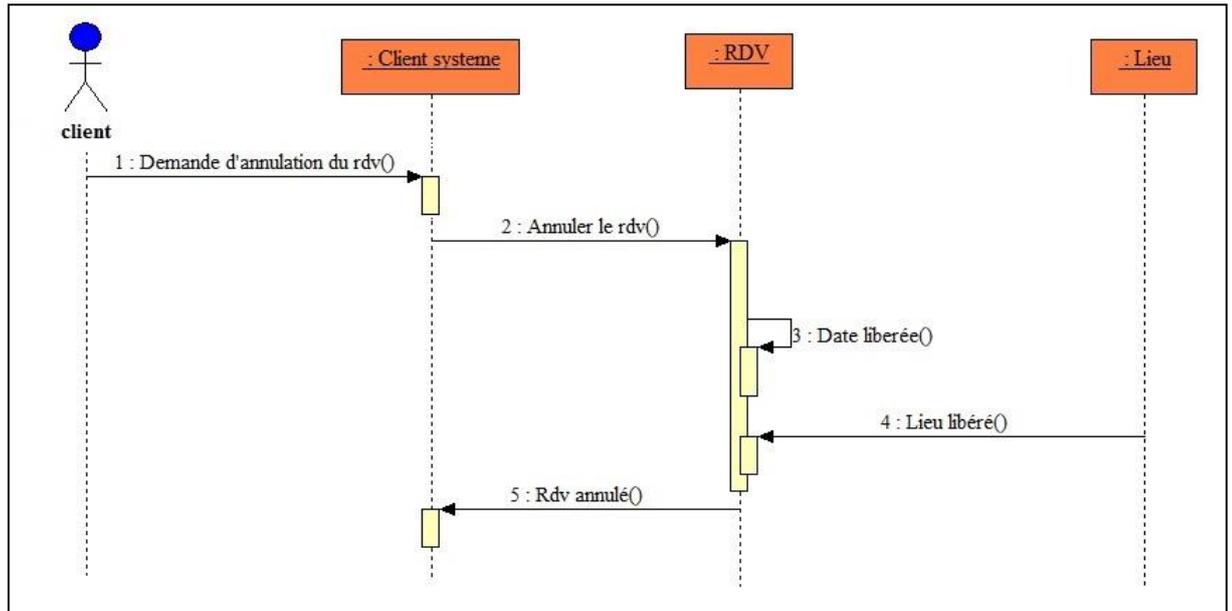


Figure 4: Diagramme de séquence réserver/annuler

II.3. Diagramme de classe :

Le diagramme de classes fait abstraction des aspects temporels ou dynamiques. Il identifie les classes et les associations de notre système. Le diagramme contient 11 classes.

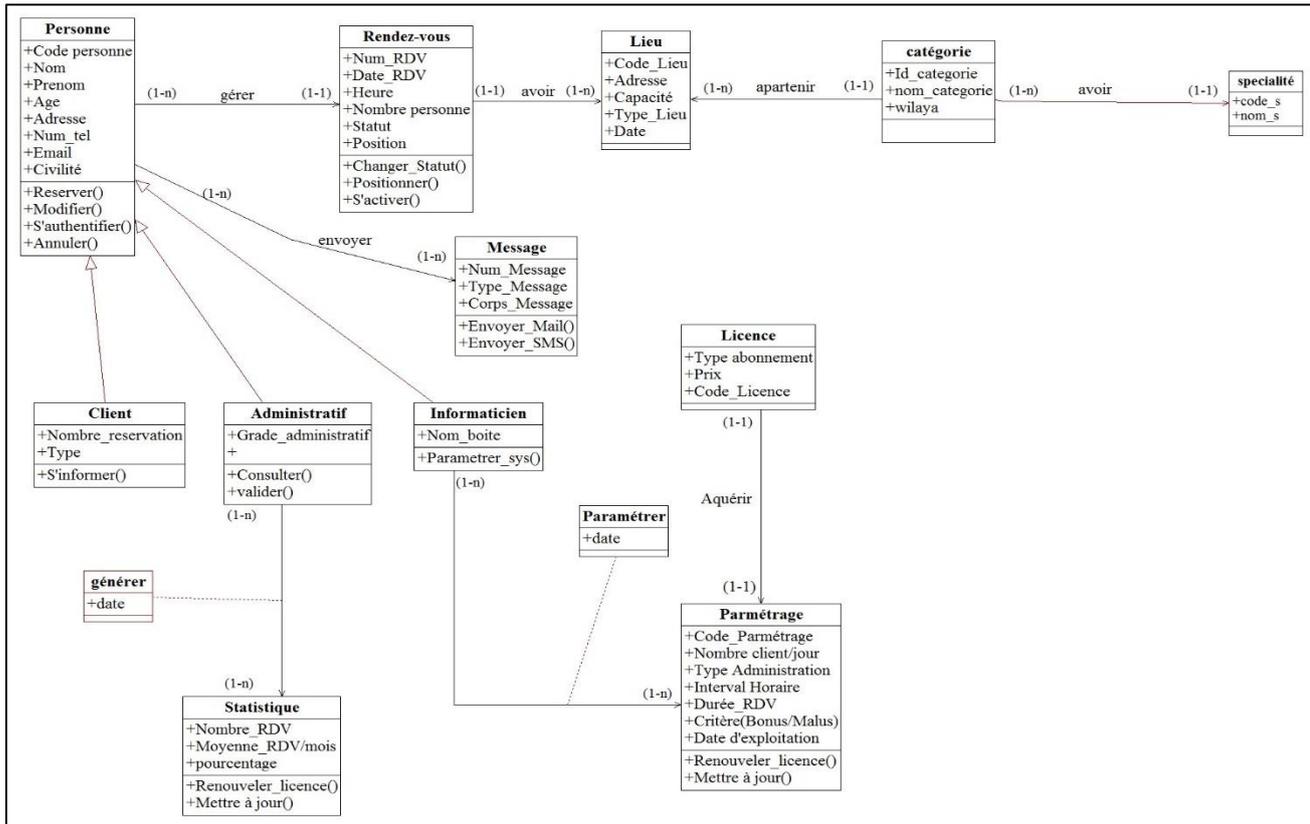


Figure 5: Diagramme de classe de la gestion des RDV

III. Modèle logique de données MyRDV

Le modèle logique présenté ci-dessous a été obtenu en appliquant les règles de passage sur le diagramme de classes (Figure).

Personne (code_personne, nom, prénom, âge, adresse, num_tel, e-mail, civilité) ;

Client (username, password, nombre_réservation, type) under personne (par héritage) ;

Administratif (grade_administratif) under personne (par héritage) ;

Informaticien (nom_boite) under personne (par héritage) ;

Rendez-vous(num_rdv, code_personne*, code_lieu*, date_rdv, heure, nombre_personne, statut, position)

Lieu (code_lieu, adresse, capacité, type_lieu, date)

Catégorie (id_categorie, code_lieu*, nom_categorie, spécialité, wilaya)

Chapitre2 : Conception du système MyRDV

Message (num_msg, type_msg, corps_msg)

Statistique (code_statistique, corps_statistique)

Paramétrage (code_paramétrage, code_licence*, nombre client_jour, type_administration, interval_horaire, durée_rdv, critère, date_expiration)

Licence (code_licence, type_abonnement, prix)

Paramétrer (code_paramétrage*, code_informaticien*, date)

Générer (code_statistique*,code_administratif*, date)

Envoyer (num_msg*, code_personne*)

Specialité(code_s,nom_s,categorie*)

❖ Implémentation de déclencheurs (Aspect dynamique) :

- Déclencheur « STATUS » :

Si le client choisit une date déjà confirmée le déclencheur indiquera un message que la date est déjà prise ; ci-dessus le code du déclencheur :

```
CREATE OR REPLACE TRIGGER "STATUS"
before insert on "RDV"
for eachrow
declare rdv number;
begin
rdv:=0;
select Code_rdv into rdv from RDV where heure= :new.heure and DATE_RDV=
:new.date_rdv and statut='C';
if rdv <> 0 then
raise_application_error (-20501,'ce rdv est pris par un autre, vous devez changer si
une annulation un message vous sera transmis');
end if;
exception
when no_data_found then rdv := 0;
if rdv = 0 and :new.statut <> 'C' then
```

```
select 'A' into:new.statutfrom dual;
end if;
end;
```

Figure 6: code déclencheur attribuer statut RDV

- déclencheur d'incrémentation automatique des clés:

Chaque fois qu'une personne est ajouté son code sera incrémenté automatiquement, ci-dessus le code source

```
CREATE OR REPLACE TRIGGER "TR_INC_CLIENT"
before insert on client
for eachrow
begin
select s_inc_client.nextval
into:new.CODE_CLIENT
from DUAL;
END;
```

Figure 7: Code trigger incrémentation automatique du code client

IV. Conclusion :

Nous avons réalisé dans ce chapitre la conception UML du système MyRDV. Cette conception est une étape nécessaire et très importante pour pouvoir créer notre base de données et réaliser l'application, le prochain chapitre détaillera les étapes de l'implémentation de l'application MyRDV.

Chapitre3 : Implémentation du Système

MyRDV

I. Introduction

Après avoir présenté une étude conceptuelle des données et de traitements dans le chapitre précédent, Nous allons, dans ce chapitre, justifier le choix du langage, donner un bref aperçu sur les outils utilisés, présenter les résultats de notre travail, et finir par une conclusion et quelques perspectives.

II. Définition d'un site web :

Un site web est un ensemble de page web qui sont écrites en des langages informatiques, parmi ses langages informatiques, il y a le HTML, le CSS et PHP qui seront utilisés pour la réalisation de notre projet [5].

III. Choix du langage de programmation :

Dans un esprit de défi, et de vouloir mettre en œuvre les connaissances qu'on a acquis durant notre formation, nous avons choisi HTML CSS et PHP des langages qui sont très utiliser, notamment par un grand nombre de programmeurs professionnels ce qui en fait des langages de haut niveau.

III.1.Définition du HTML :

Le HTML « HyperText Mark-Up Language »est un langage dit de « marquage » de « structuration » ou de « balisage » dont le rôle est de formaliser l'écriture d'un document avec des balises de formatage. Les balises permettent d'indiquer la façon dont doit être présenté le document et les liens qu'il établit avec d'autres documents [6].

III.2. Définition et caractéristiques du CSS :

Le CSS « *CascadingStyleSheets* » est un langage qui consiste à regrouper dans un même document des caractéristiques de mise en forme associées à des groupes d'éléments. Il suffit de définir par un nom un ensemble de définitions et de caractéristiques de mise en forme, et de l'appeler pour l'appliquer à un texte [7].



Les feuilles de style ont été mises au point afin de compenser les manques du langage HTML en ce qui concerne la mise en page et la présentation. En effet, le HTML offre un certain nombre de balises permettant de mettre en page et de définir le style d'un texte,

toutefois chaque élément possède son propre style, indépendamment des éléments qui l'entourent. Grâce aux feuilles de style, lorsque la charte graphique d'un site composé de plusieurs centaines de pages web doit être changée, il suffit de modifier la définition des feuilles de style en un seul endroit pour changer l'apparence du site tout entier.

III.3.Définition et principales caractéristiques du PHP

Le langage PHP est principalement un langage de **programmation web côté serveur** ce qui veut dire que c'est le serveur (la machine qui héberge le site web en question) qui va



interpréter le code PHP et générer du code qui pourra être interprété par un logiciel. Le plus souvent, le code généré est le HTML afin d'être lu par un navigateur mais il peut être utilisé pour d'autres langages ou formats tels que WML, GIF, PDF, SVG, etc. Il a été conçu pour permettre la création d'applications dynamiques, le plus souvent dédiées au web. PHP peut être installé sur les principaux **serveurs web** du marché, les plus répandus étant IIS et Apache. Ce couplage permet de récupérer des informations issues d'une base de données, d'un système de fichiers (contenu de fichiers et de l'arborescence) ou plus simplement des données envoyées par le navigateur afin d'être interprétées ou stockées pour une utilisation ultérieure. C'est un langage peu typé et souple et donc facile à apprendre par un débutant mais, de ce fait, des failles de sécurité peuvent rapidement apparaître dans les applications. Son utilisation commence avec le traitement des formulaires puis par l'accès aux bases de données. L'accès aux bases de données est aisé une fois l'installation des modules correspondant effectuée sur le serveur. La force la plus évidente de ce langage est qu'il est devenu au fil du temps un incontournable des **offres d'hébergement**. Libre, gratuit, simple d'utilisation et d'installation, ce langage nécessite comme tout langage de réseau une bonne compréhension des mécanismes sous-jacents ainsi qu'une connaissance des problèmes de sécurité [8].

III.4- jQuery :

jQuery est une bibliothèque JavaScript libre qui porte sur l'interaction entre JavaScript (comprenant Ajax) et HTML, et a pour but de simplifier des commandes communes de JavaScript. La première version date de janvier 2006 [9].



La bibliothèque contient notamment les fonctionnalités suivantes :

- Parcours et modification du DOM (y compris le support des sélecteurs CSS 1 à 3 et un support basique de XPath) ;
- Événements ;
- Effets visuels et animations ;
- Manipulations des feuilles de style en cascade (ajout/suppression des classes, d'attributs...);
- Ajax ;
- Plugins ;
- Utilitaires (version du navigateur web...).

IV. Choix de l'outil de développement :

Vu la multidisciplinarité et la domination croissante des sites web, plusieurs outils de développement de HTML et PHP ne cessent de voir le jour. On peut rencontrer pas mal d'Environnement de Développement Intégré (EDI) HTML et PHP. Certains sont en open source et d'autres commerciaux.

Nous avons utilisé Dreamweaver 8 qui est un éditeur de site web de type « tel affichage, tel résultat ». Dreamweaver 8 offre deux modes de conception par son menu affichage. L'utilisateur peut choisir entre un mode création permettant d'effectuer la mise en page directement à l'aide d'outils simples, comparables à un logiciel de traitement de texte (insertion de tableau, d'image, etc.). Il est également possible d'afficher et de modifier directement le code (HTML ou autre) qui compose la page. On peut passer très facilement d'un mode d'affichage à l'autre, ou opter pour un affichage mixte. Il offre aujourd'hui la possibilité de concevoir des feuilles de style et les liaisons avec des bases de données.



Nous avons aussi utilisé Bootstrap qui est une collection d'outils utile à la création de sites web et applications web. C'est un ensemble qui contient des codes HTML et CSS, ainsi que des extensions JavaScript en option. C'est l'un des projets les plus populaires sur la plate-forme de gestion de développement GitHub.

❖ WampServer

On a ainsi utilisé comme outil le WampServer qui 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 phpMyAdmin 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 *trayicon* (icône près de l'horloge de Windows).

V. Choix du SGBD :

De nombreux SGBD sont disponibles sur le marché, partant des SGBD gratuits jusqu'aux SGBD destinés spécialement aux professionnels, comportant de plus nombreuses fonctionnalités, mais plus coûteux.

Citons quelques-uns : MsAccess, SQL Server Oracle & TimesTen & DBLibrary, DB2, Interbase (Delphi) Sybase, SQL Anywhere , Adabas , MySQL, PostgreSQL, OpenOfficeBase, Monet , SD-SQL-Server...

Vu que notre modélisation était basée sur l'approche objet, nous avons choisi Oracle comme SGBD car c'est un SGBDRO, mais aussi pour sa puissance, sa forte appréciation par les développeurs et pour ces fonctionnalités suivantes :



- La définition et la manipulation des données.
- La cohérence des données.
- La confidentialité des données.
- L'intégrité des données.
- La sauvegarde et la restauration des données.
- La gestion des accès concurrents.

VI. Description de l'application :

Chapitre3 : Implémentation du système MyRDV

Nous allons présenter nos principaux écrans de notre application web :

- **Ecran d'accueil « Accueil » :**

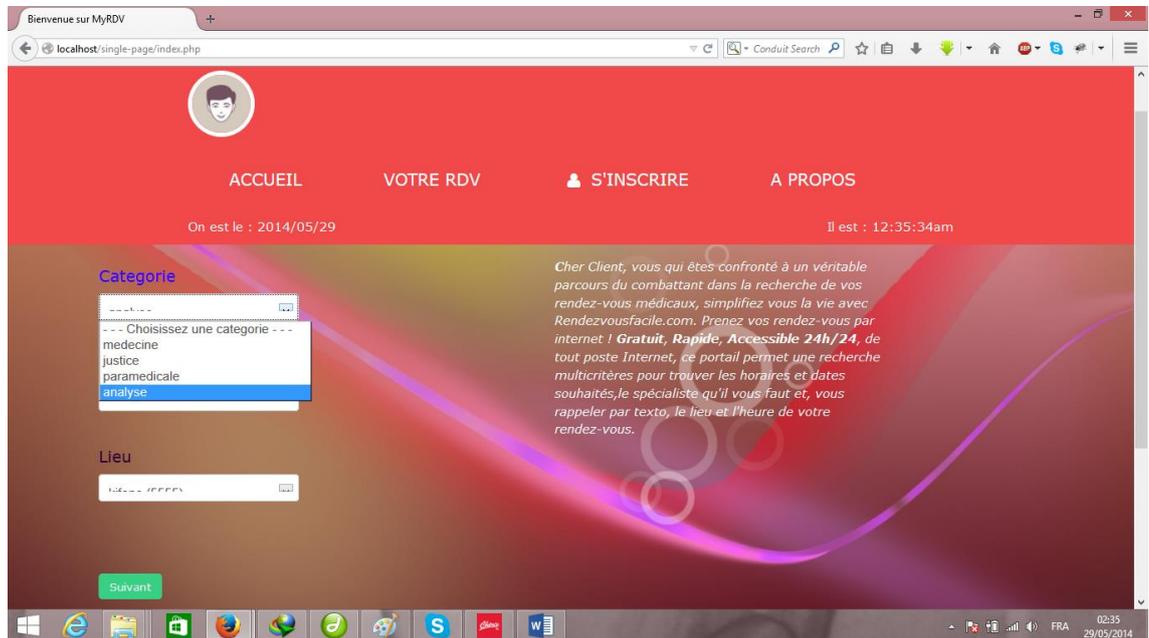


Figure 8: Capture d'écran d'accueil du site MyRDV

Dans cette page le client va se retrouver dans la page d'accueil du site MyRDV où il y a une barre de navigation incluant l'accueil, Prendre un RDV, Inscription, A propos_

Dans cette page le client choisit

1. La catégorie de l'organisme (Médecin, Paramédical, Avocat...).
2. Choix de la spécialité de la catégorie d'auparavant.
3. Choix du lieu du rendez-vous (adresse).

Après avoir choisi la catégorie, le lieu, et la spécialité on a qu'à cliquer sur le bouton « suivant » pour aller vers la page du calendrier qui est la suivante

- **Ecran du calendrier « Votre RDV »:**

Cette page permet au client de choisir la date et l'heure qui lui convient selon la disponibilité.

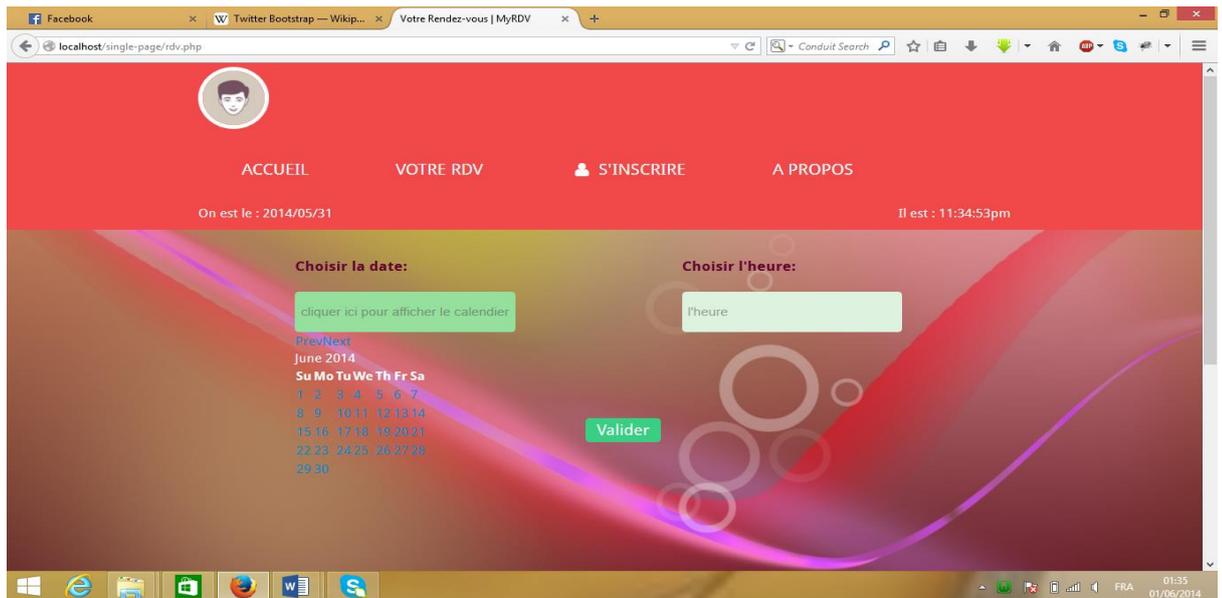


Figure 9: Capture d'écran de la prise d'un rendez-vous sur le site MyRDV

- *Écran d'inscription « S'inscrire »:*

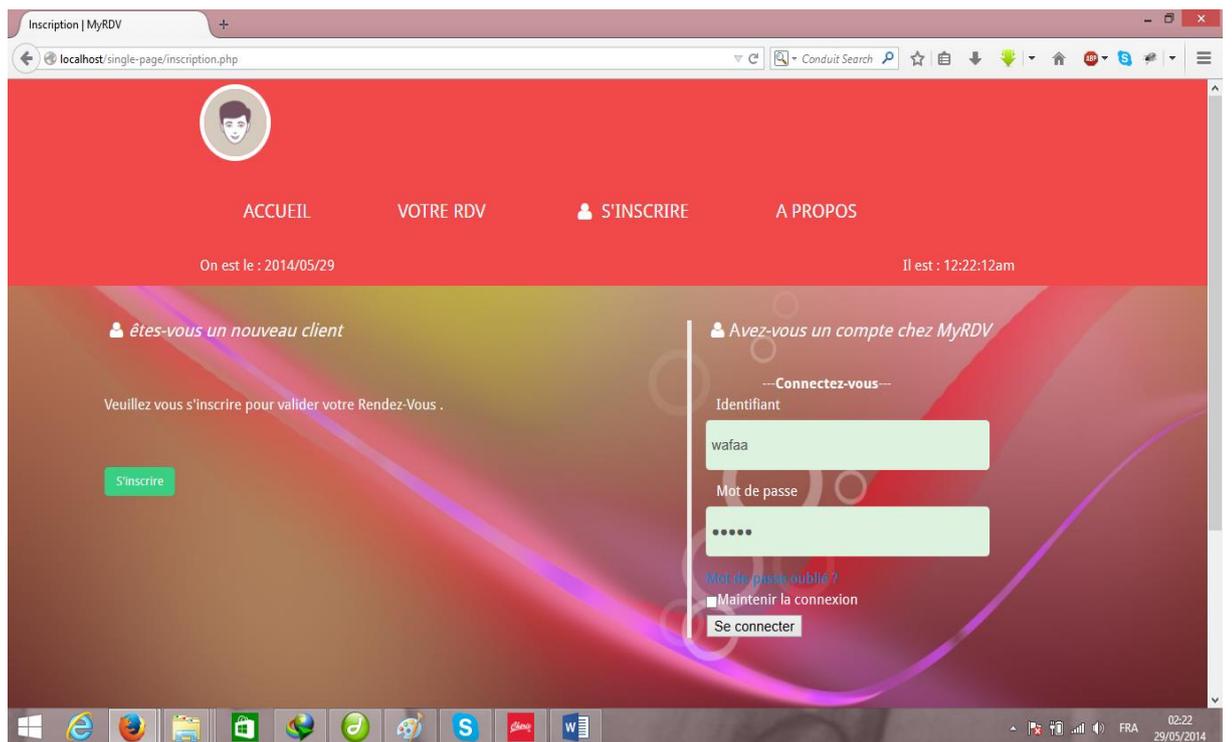


Figure 10: Capture d'écran d'inscription sur le site MyRDV

Après avoir choisi la date du rendez-vous on bascule vers la page d'inscription qui contient deux parties :

Chapitre3 : Implémentation du système MyRDV

1. Partie où le client est déjà inscrit dans notre base de données « Avez-vous un compte chez MyRDV » il se connectera avec son identifiant et son mot de passe directement et puis il prend son rendez-vous.
2. Partie où le client est nouveau dans ce cas le client doit remplir un formulaire par suite il réserve.

- **Ecran d'à propos « A propos » :**

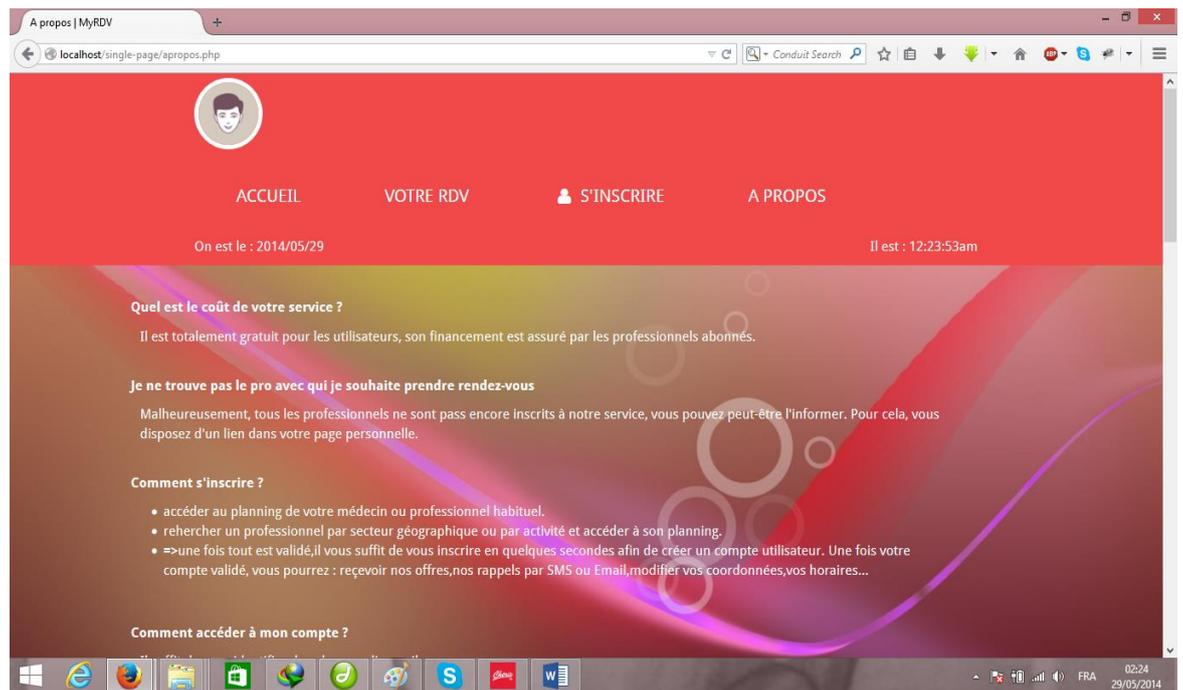


Figure 10: Capture d'écran de la page d'à propos du site MyRDV

Cet écran répond aux questions fréquentes des clients.

- **Ecran de contact « Contact » :**

Cet écran est permet aux personnes intéressés par le produit de contacter la boite informatique ou bien l'administration.

Chapitre3 : Implémentation du système MyRDV

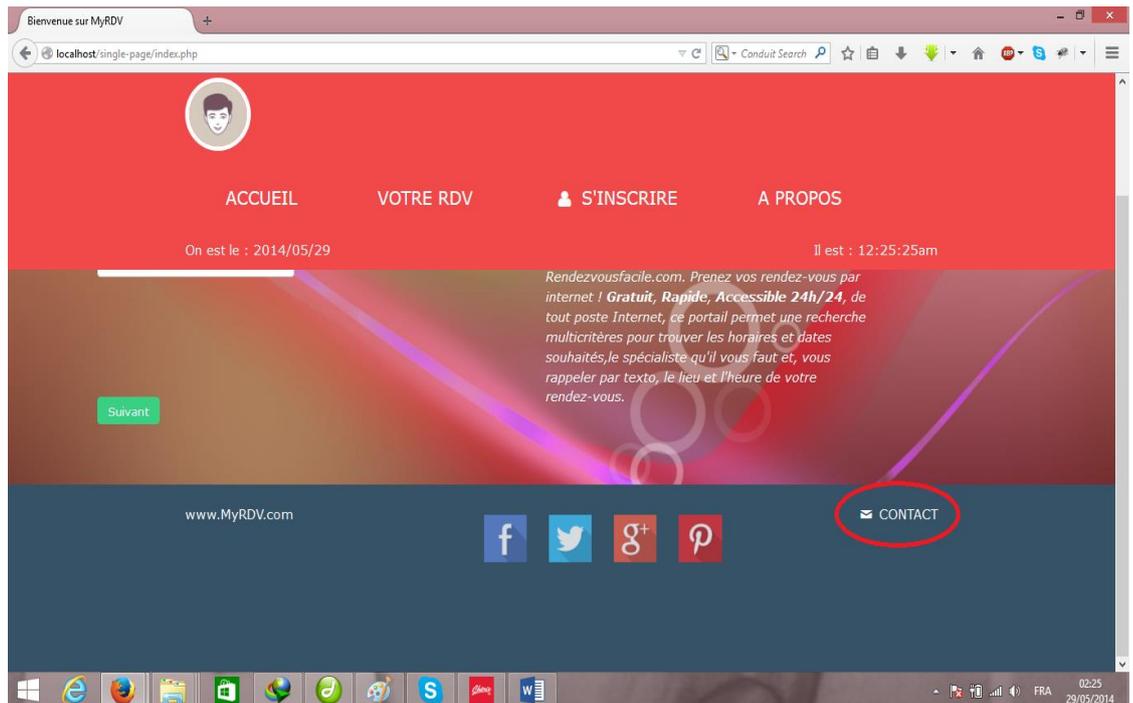


Figure 11: Capture d'écran de la position de contact du site MyRDV

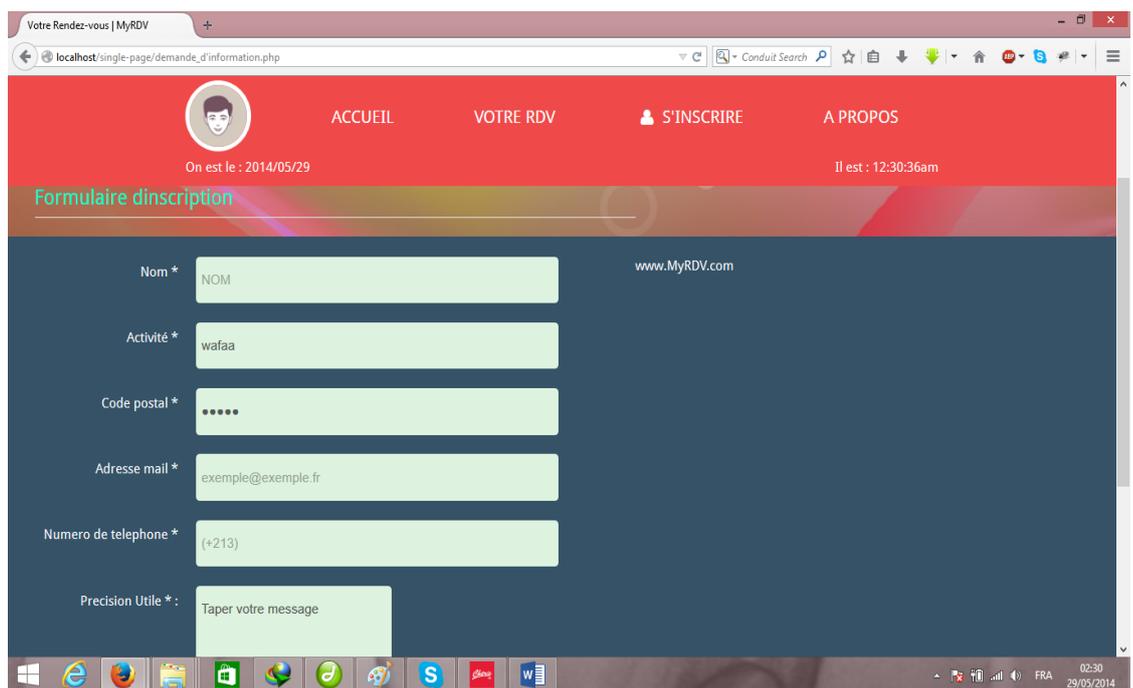


Figure 12: Capture d'écran du formulaire de contact du site MyRDV

VII. Conclusion :

Dans ce chapitre, nous avons fait la réalisation de notre projet MyRDV. Nous avons justifié notre choix de langage et d'outils de développement. Nous avons aussi donné tous les écrans qui constituent notre site web. Nous avons essayé d'optimiser notre application pour répondre aux besoins de l'utilisateur.

Conclusion générale

Notre Projet porte sur l'organisation et l'automatisation de la gestion d'une prise d'un rendez-vous, afin d'augmenter la fiabilité, l'efficacité de l'effort humain et de faciliter les tâches pénibles au sein d'un organisme ; en le déroulant sur une plateforme d'un site web dynamique.

Le but principal de ce projet se base sur l'optimisation du temps ainsi vos clients n'ont plus besoin d'attendre l'ouverture de votre structure pour prendre un rendez-vous. Vous les libérez de leurs contraintes professionnelles et familiales, d'où une simplicité de la charge quotidienne

Ce projet nous a permis d'avoir une approche complète du développement d'un site web et une bonne initiation au cycle complet du développement d'un site web, de la conception à la validation en passant par les différentes étapes incrémentales de codage et de tests et nous a appris aussi à concevoir une base de données complète.

Nous avons appliqué au maximum possible les règles de bases permettant d'avoir une application performante. Nous avons utilisé UML pour modéliser le système et le langage PHP pour implémenter notre application.

Le site web MyRDV ainsi construit fonctionne correctement et répond aux besoins énoncés, mais peut être amélioré, nous prévoyons de :

- rajouter d'autres déclencheurs pour automatiser encore plus les actions de réaffectation de rendez-vous en cas d'annulation.
- héberger le site et profiter des retours de test du système
- rajouter un tableau de bord dynamique et des notifications qui seront automatiquement envoyé au client

Références webographies

www.prendre-mon-rdv.com/

www.orsys.fr/formations-php.asp

[1] http://fr.wikipedia.org/wiki/UML_%28informatique%29

[2] <http://www.commentcamarche.net/contents/1141-introduction-a-uml>

[3] <http://sabricole.developpez.com/uml/tutoriel/unifiedProcess/>

[4] <http://www.commentcamarche.net/contents/1139-caracteristiques-d-uml>

[5] http://fr.wikipedia.org/wiki/Page_web_dynamique

[6] http://fr.wikipedia.org/wiki/Hypertext_Markup_Language

[7] www.futura-sciences.com/magazines/high-tech/infos/dico/d/internet-css-4050/

[8] <http://fr.wikipedia.org/wiki/PHP>

[9] <http://fr.wikipedia.org/wiki/JQuery>

Résumé

A travers ce projet, nous avons réalisé un site web dynamique facile à s'intégrer dans l'environnement de travail de multiples secteurs médicaux, commerciaux et administratifs. Cette application se traduit par la mise en œuvre d'un système qui facilite la prise d'un rendez-vous baptisé MyRDV. Notre système vise essentiellement à laisser libre accès au client de prendre le rendez-vous à l'heure qui l'arrange et d'éliminer le temps d'attente des clients entre les rendez-vous.

La modélisation du système MyRDV a été faite par UML en s'appuyant sur le processus UP. La base de données et l'application ont été respectivement créées par le SGBD Oracle et le langage PHP, réputés par leur puissance et efficacité.

Mots clés : Rendez-vous, UML, PHP.

Abstract

Through this project, we wanted to make an easy dynamic web site to integrate into the work environment of multiple medical sectors, commercial and administrative. This application results in implementing the system MyRDV. Our system essentially aims to reduce the daily care of people by reducing customer waiting time in making an appointment.

Modeling MyRDV system was made by UML based on the UPprocess. The database and application were respectively created by the Oracle DBMS and PHP, known for their power and effectiveness.

Keys words : Appointment, UML, PHP.

تلخيص

من خلال هذا المشروع رغبتنا في جعل موقع ويب ديناميكي سهل الاندماج في بيئة العمل من قطاعات طبية متعددة وتجارية وإدارية. نتائج هذا التطبيق في تنفيذ النظام MyRDV الذي يهدف أساسا لتقليل وقت الانتظار للعملاء في جعل الموعد.

قدم نظام MyRDV النمذجة UML من قبل على أساس عملية UP. تم إنشاء قاعدة البيانات وتطبيق على التوالي من قبل نظم إدارة قواعد البيانات أوراكل و PHP، والمعروف عن قوتها وفعاليتها.

به: موعد، UML، PHP.

