

SOMMAIRE

INTRODUCTION.....	5
I- PRESENTATION DE L'ORGANISME D'ACCUEIL.....	5
1- Présentation du Groupe Al Omrane.....	6
2- Présentation d'Al Omrane Oujda	6
3- Organigramme et Structure de la société Al Omrane Oujda	7
4- Présentation de la division du système d'information.....	8
II- ETUDE PRELIMINAIRE.....	8
1- Problématique	9
2- Besoins fonctionnels.....	10
2.1- La gestion des demandes	10
2.2- La gestion des réponses	10
2.3- La gestion des commentaires.....	11
2.4- La gestion des utilisateurs	11
3- Besoins techniques.....	11
4- Règles de gestion.....	11
5- Identification des acteurs.....	12
6- Identification des messages	12
7- Choix techniques	12
7.1- Présentation du MVC	13
7.2- Avantages de l'architecture en MVC.....	13
III- MODELISATION DE L'APPLICATION	13
1- Diagramme des cas d'utilisation	13
2- Modélisation de la navigation	15
3- Diagrammes de séquences.....	16
4- Modèle conceptuel de données.....	17
5- Modèle logique de données.....	18
IV- REALISATION DE L'APPLICATION	18
1- Choix de l'outil.....	19
2- Etapes de réalisation de l'application.....	19
2.1- La prise en main de cakePHP	19

2.2-	La création et configuration de la base de données	20
2.3-	La création des tables, modèles, vues et contrôleurs	20
2.4-	Définition des relations entre les modèles.....	21
2.5-	Tester les relations entre les modèles via le maquettage rapide	22
2.6-	Création du système d'authentification.....	22
2.7-	Création du système de gestion des demandes.....	22
2.8-	Intégrer un système d'upload des fichiers	22
2.9-	Création du système de gestion des utilisateurs.....	23
3-	Présentation de l'application	23
CONCLUSION.....		26
BIBLIOGRAPHIE		27

INTRODUCTION

Dans le cadre du cursus de la filière informatique de la faculté des sciences et Techniques de Fès, un projet de fin d'études en vue de l'obtention du diplôme de licence est prévu à la fin de la 3^{ème} année et qui doit être réalisé dans le cadre d'un stage de deux mois à passer au sein d'un organisme public ou privé.

Mon stage de fin d'études fut passé au sein de la société Al Omrane d'Oujda et plus précisément au service Informatique qui gère le système d'information de ladite société et veille sur la bonne exploitation de ce système.

L'informatisation des procédures de travail est d'une importance capitale pour la société au vue des nombreux avantages qu'elle offre en termes de temps, de coût et de simplification du travail. C'est pourquoi le service informatique est en perpétuelle recherche de développement de nouvelles applications qu'il met en réseau pour faciliter le travail au sein de la société.

Mon projet de fin d'études rentre donc dans cette dynamique et le travail qui m'a été confié consiste en le développement d'une application informatique permettant un contact permanent avec le service informatique en vue de signaler tel ou tel problème rencontré par les utilisateurs du système ou demander des informations. Cette application va permettre une réactivité rapide et de ce fait, gagner beaucoup de temps. Elle permettra également de garder une traçabilité de l'ensemble des demandes exprimées par les différents services de la société et les réponses faites ou solutions apportées aux problèmes posés, d'où une capitalisation du savoir et du savoir faire au sein de la société.

I- PRESENTATION DE L'ORGANISME D'ACCUEIL

1- Présentation du Groupe Al Omrane

Le Holding d'Aménagement Al Omrane a été créé dans un contexte particulier marqué par l'intérêt grandissant porté d'une manière générale à la question de l'habitat depuis quelques années.

L'émergence du Holding d'Aménagement Al Omrane s'est également opérée dans le cadre de la mise en place de la nouvelle politique gouvernementale en matière d'habitat qui s'est fixée l'objectif d'accroître le rythme de production de logements sociaux, lequel objectif traduit une triple préoccupation :

- Augmenter l'offre réglementaire en logements pour répondre aux besoins nouveaux
- Rendre adéquate l'offre avec la demande
- Atténuer progressivement le déficit cumulé.

Le Holding Al Omrane développe et met en œuvre aujourd'hui un ensemble de programmes :

- Création de villes nouvelles,
- Aménagement de pôles urbains,
- Éradication des bidonvilles dans le cadre du programme « villes sans bidonvilles »,
- Requalification des quartiers sous équipés,
- Mise à niveau urbaine des agglomérations,
- Réhabilitation du patrimoine dans les médinas et les ksour,
- Confortement des constructions menaçant ruine,

Ainsi, le Groupe Al Omrane, grâce à ses ressources humaines, à sa culture d'entreprise, à la cohérence de ses démarches en matière de montage et de mise en œuvre de programmes, au partenariat engagé avec les promoteurs immobiliers privés et à la coopération développée avec les institutions internationales, est en mesure de mobiliser, dans les meilleures conditions, des financements à la hauteur des investissements nécessaires à la réalisation de projets d'envergure susceptibles de répondre aux besoins de la croissance urbaine et du développement humain.

2- Présentation d'Al Omrane Oujda

La Société Al Omrane Oujda est une filiale du Holding d'aménagement Al Omrane dont la vocation est d'occuper une place prépondérante dans la production de l'habitat dans la région de l'Oriental.

Placée sous la tutelle du Ministère de l'Habitat, de l'Urbanisme et de l'Aménagement de l'espace, Al Omrane Oujda s'attache à la satisfaction des besoins dans un cadre concurrentiel et dans le respect des politiques publiques d'aménagement décidées par le gouvernement.

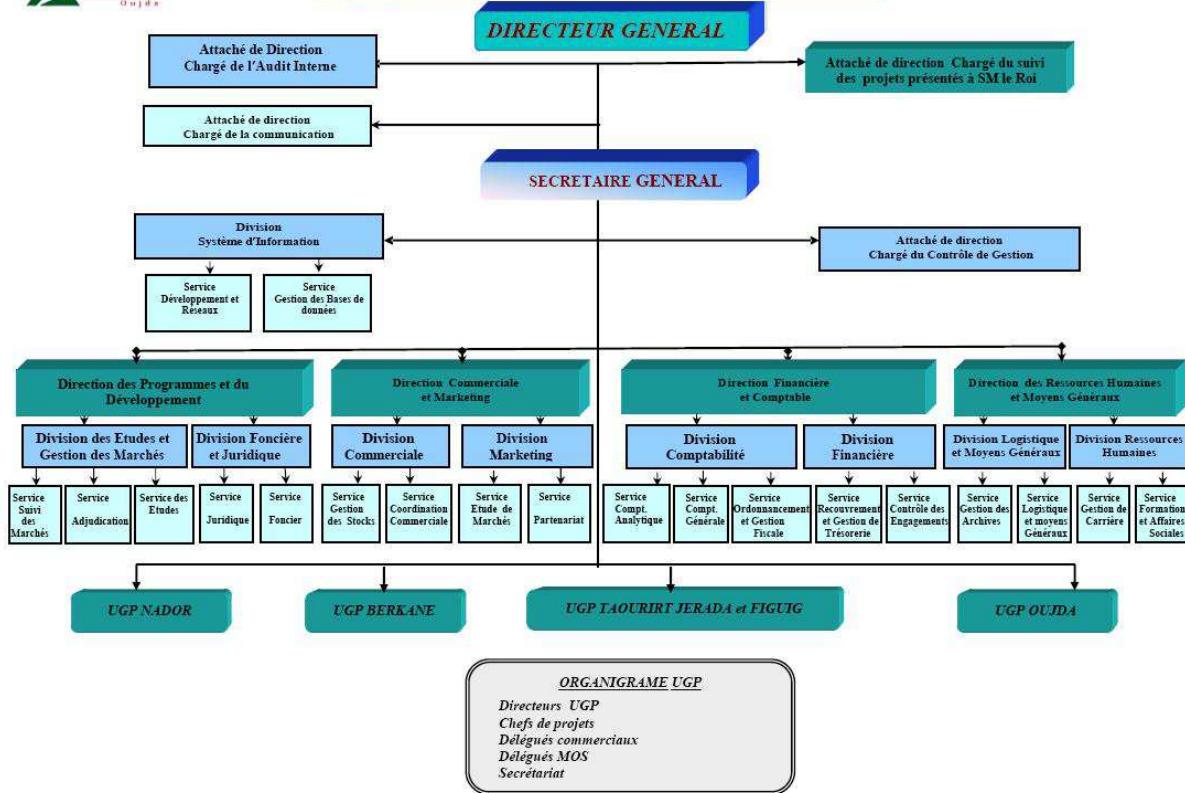
Ainsi, Al Omrane Oujda a pu se consacrer au développement d'un certain nombre de pôles urbains nouveaux, entités intégrées où l'habitat (lots de terrain équipés, ensembles immobiliers de logements pour les différents créneaux de marché) voisine avec les unités d'activité (commerces, lots industriels) et les équipements sociaux, éducatifs et culturels.

Par ailleurs, Al Omrane Oujda a vu se diversifier ses milieux d'intervention alors même que la géographie régionale créait déjà de grandes variétés (espace méditerranéen, zones pré sahariennes, massifs montagneux ou plaines irriguées). Par cette approche, Al Omrane Oujda n'a plus une compétence globale, mais une addition de savoir-faire de haut niveau spécialisé par cible d'intervention et coordonné par la Direction.

Ainsi, tous sont associés à «l'option qualité » définie et approfondie pour chaque créneau dans le but de satisfaire les destinataires finaux : c'est une fonction stratégique qui s'impose également à tous les fournisseurs et intervenants aux côtés de la société. Al Omrane Oujda reflète aujourd'hui la complexité de ses milieux d'intervention. Il est bien armé pour affronter la concurrence et pour offrir un partenariat de qualité à tous les organismes, à toutes les institutions qui sollicitent son savoir-faire.

3- Organigramme et Structure de la société Al Omrane Oujda

Organigramme de la Société AL OMRANE -OUJDA-



4- Présentation de la division du système d'information

Parmi les structures de cette organisation figure la Division du système d'information où j'ai été accueilli pendant deux mois dans le cadre de mon stage de fin d'études:

La division SI d'Al Omrane Oujda est responsable de gérer le système d'information au sein de la société, il a comme mission de :

- Assurer le bon fonctionnement du système d'information
- Proposer et développer de nouvelles applications selon les besoins de la société
- Piloter les évolutions du système d'information
- Suivi des projets externalisé à un prestataire
- Support aux utilisateurs dans leurs quotidiens
- Développement et mise à jour du site officiel : www.alomrane-oujda.ma

Le travail qui m'a été confié au sein de ce service consiste en le développement d'une application informatique faisant l'objet de ce qui suit dans ce rapport.

II- ETUDE PRELIMINAIRE

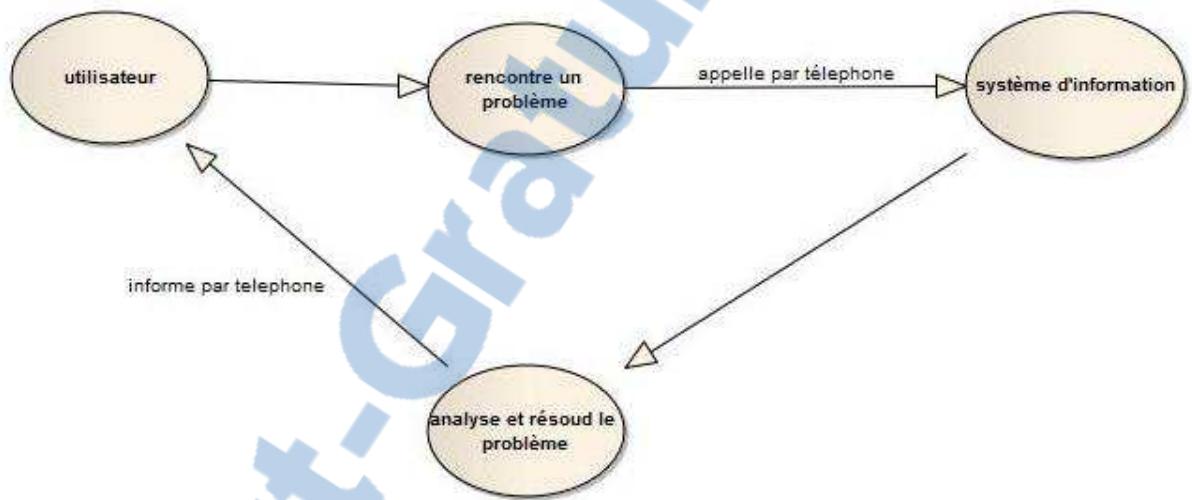
1- Problématique

Les différents services d'Al Omrane (administratif, commercial, technique, comptabilité) utilisent des applications Oracle Forms pour le traitement de leurs données.

Dans leur quotidien, les employés demandent du support auprès du service informatique afin de réaliser les tâches suivantes :

- résoudre une anomalie engendrée par une application
- migrer les données Excel vers la base de données oracle
- Etat de reporting

Actuellement le processus est le suivant : (par exemple cas d'une anomalie)



Ce processus présente plusieurs inconvénients :

- pas de traçabilité sur les demandes ni sur les solutions trouvées
- date non illustrée
- durée d'intervention non connue
- refaire le traitement pour des demandes redondantes
- l'échange de fichier se faisant via Outlook, il est difficile de garder le lien entre une demande et les fichiers la concernant

La solution proposée est l'informatisation de ce processus afin d'écartier l'appel téléphonique et le remplacer par une application intranet qui va gérer les échanges des messages (demandes et réponses) et fichiers entre les différents services et le service informatique.

Cette application sera composée des modules d'authentification, gestion des utilisateurs, gestion des demandes, gestion des réponses et des commentaires.

2- Besoins fonctionnels

L'objectif de l'application est de gérer les demandes exprimées par les différents services d'Al Omrane au service informatique. Pour cela, elle doit permettre de réaliser les tâches suivantes :

La gestion des demandes

Une demande est caractérisée par :

- Id
- Auteur et service auquel il appartient
- Type
 - Résolution d'une anomalie
 - Migration des données d'un fichier Excel vers la base de données
 - Etat de reporting
- Etat
 - Traitée
 - En cours
 - Abandonnée
- Priorité
 - Critique
 - Urgente
 - Normale
- Objet
- Description
- Date de création
- Pièces jointes

La gestion des demandes se base sur les tâches suivantes :

- Postuler une demande
- Editer une demande
- Afficher une demande
- Abandonner une demande
 - si l'employé juge que le traitement de sa demande est inutile (problème résolu par exemple) il serait en mesure de l'abandonner
- Supprimer une demande
- Afficher la liste des demandes avec possibilité de les filtrer par type, priorité, état, service, auteur ou date
- Rechercher une demande par mot clé

La gestion des réponses

Une réponse est caractérisée par :

- Auteur
- Pièces jointes
- Date
- Relation avec d'autres demandes
 - L'utilisateur peut mentionner une demande existante s'il la juge en relation avec celle en cours de traitement (par exemple cas des doublants)

L'utilisateur doit pouvoir répondre à une demande ainsi que la modification de sa réponse.

La gestion des commentaires

Les commentaires serviront à clarifier au mieux les besoins exprimés et les réponses faites, exprimer la satisfaction, décrire les problèmes rencontrés, etc ...

Un commentaire est caractérisé par :

- Auteur
- Date

La gestion des utilisateurs

Un utilisateur est caractérisé par :

- Nom et prénom
- Service
- Téléphone
- Identifiant
- Mot de passe

L'administrateur doit pouvoir consulter la liste des utilisateurs, ajouter, éditer ou supprimer un utilisateur.

Un utilisateur doit pouvoir modifier son compte

3- Besoins techniques

L'application doit être disponible à tout moment et permettre un accès multiutilisateur, il est donc nécessaire que l'environnement d'exploitation dispose d'une architecture réseaux client/serveur.

L'application doit offrir une interface simple et claire d'utilisation, ainsi qu'une gestion de tout scénario possible afin de maintenir son bon fonctionnement.

4- Règles de gestion

- Un utilisateur non Authentifié ne peut pas accéder à l'application



- A l'exception de l'administrateur, un utilisateur ne peut pas effectuer les tâches suivantes :
 - Suppression d'une demande
 - Edition ou abandon d'une demande dont il n'est pas l'auteur
 - Edition ou abandon d'une demande déjà traitée ou abandonnée
 - Afficher une demande ou la liste des demandes postulées par un utilisateur appartenant à un service différent que le sien
 - Lors de la saisie d'une demande, le champ relatif à l'objet doit être obligatoirement instruit

5- Identification des acteurs

Les acteurs qui peuvent manipuler l'application sont :

- Tout employé autorisé à utiliser l'application et peut de ce fait ajouter, consulter, éditer, commenter, abandonner, rechercher une demande ou modifier son compte
- L'administrateur (employés au sein du système d'information) peut en plus des fonctionnalités des autres employés répondre ou supprimer une demande, il peut aussi gérer l'ensemble utilisateurs

6- Identification des messages

- L'application doit être en mesure de fournir :
 - La liste complète des demandes
 - La liste des demandes répondant à certains critères (recherche, filtre)
 - Les informations détaillées d'une demande précise (réponse et commentaires)
- L'application doit recevoir :
 - Nouvelle demande
 - Edition d'une demande
 - Réponse à une demande
 - Abandonner ou supprimer une demande
 - Critères d'affichages de la liste des demandes
 - Relation entre les demandes
 - Commentaire sur une demande

7- Choix techniques

L'application doit être accessible depuis tous les ordinateurs du réseau intranet de la société Il s'agit d'une application web :

- L'architecture utilisée : Modèle-Vue-Contrôleur MVC
- Langage de programmation : PHP
- SGBD : MySQL

7.1- Présentation du MVC

Le Modèle-Vue-Contrôleur (MVC) est une architecture et une méthode de conception qui organise l'interface homme-machine (IHM) d'une application logicielle. Ce paradigme divise l'IHM en un modèle (modèle de données), une vue (présentation, interface utilisateur) et un contrôleur (logique de contrôle, gestion des événements, synchronisation), chacun ayant un rôle précis dans l'interface.

➤ Le modèle :

Le modèle représente le comportement de l'application : traitements des données, interactions avec la base de données, etc. Il décrit ou contient les données manipulées par l'application. Il assure la gestion de ces données et garantit leur intégrité. Dans le cas typique d'une base de données, c'est le modèle qui la contient. Le modèle offre des méthodes pour mettre à jour ces données (insertion, suppression, changement de valeur). Il offre aussi des méthodes pour récupérer ces données.

➤ La vue :

La vue correspond à l'interface avec laquelle l'utilisateur interagit. Sa première tâche est de présenter les résultats renvoyés par le modèle. Sa seconde tâche est de recevoir toutes les actions de l'utilisateur (clic de souris, sélection d'une entrée, boutons, etc). Ces différents événements sont envoyés au contrôleur.

➤ Le contrôleur :

Le contrôleur est l'élément qui va utiliser les données pour les envoyer à la vue. Son rôle est donc de récupérer les informations, de les traiter en fonction des paramètres demandés par la vue (par l'utilisateur, exemple: afficher les derniers articles), puis de renvoyer à la vue les données afin d'être affichées.

Si une action nécessite un changement des données, le contrôleur demande leurs modifications au modèle.

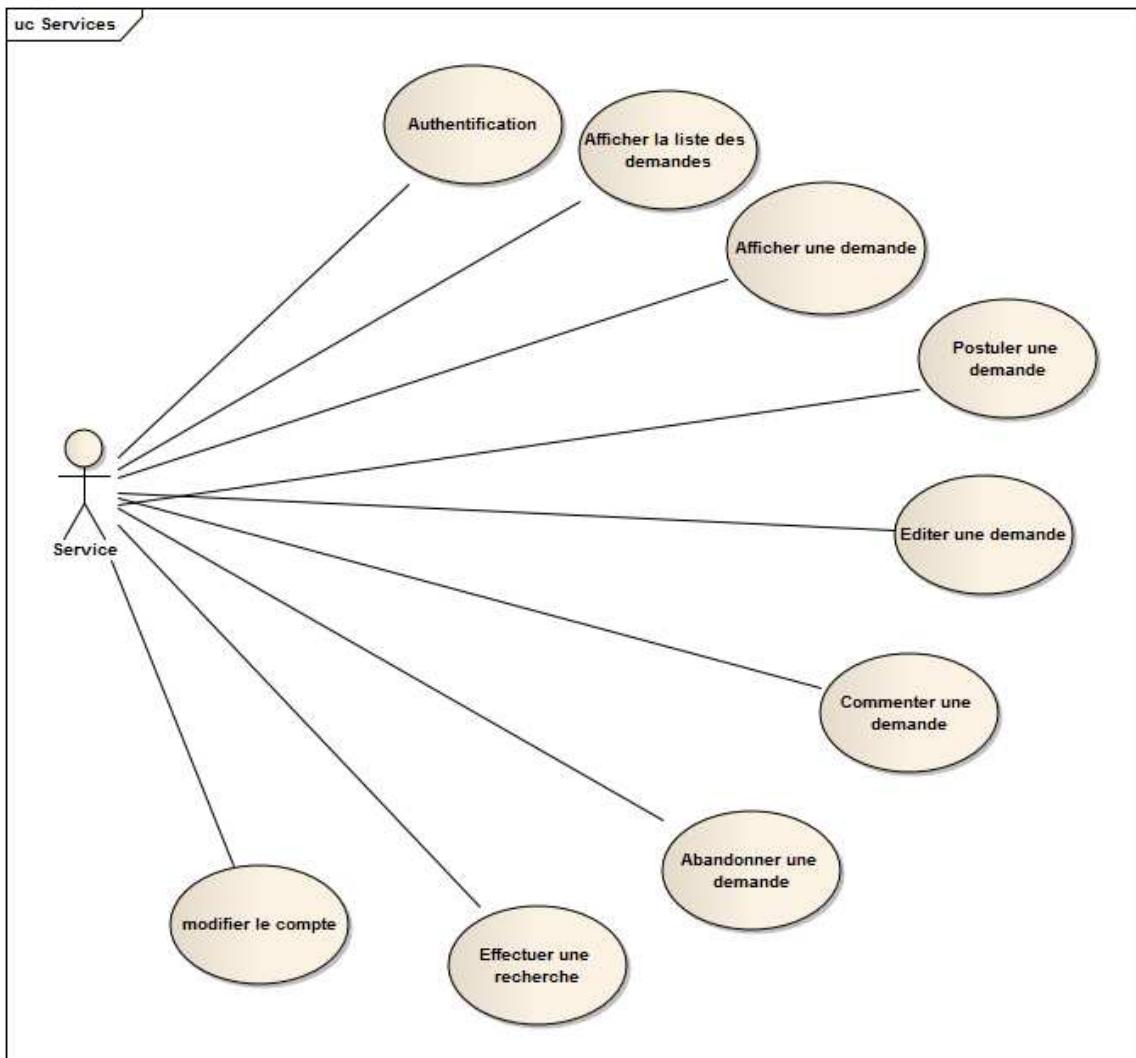
7.2- Avantages de l'architecture en MVC

L'approche MVC permet la transformation d'une application en un ensemble maintenable, modulaire et rapidement développé grâce à une conception claire et efficace assurée par la séparation des tâches dans des modèles, vues et contrôleurs simplifiant ainsi l'ajout et la modification des différentes fonctionnalités.

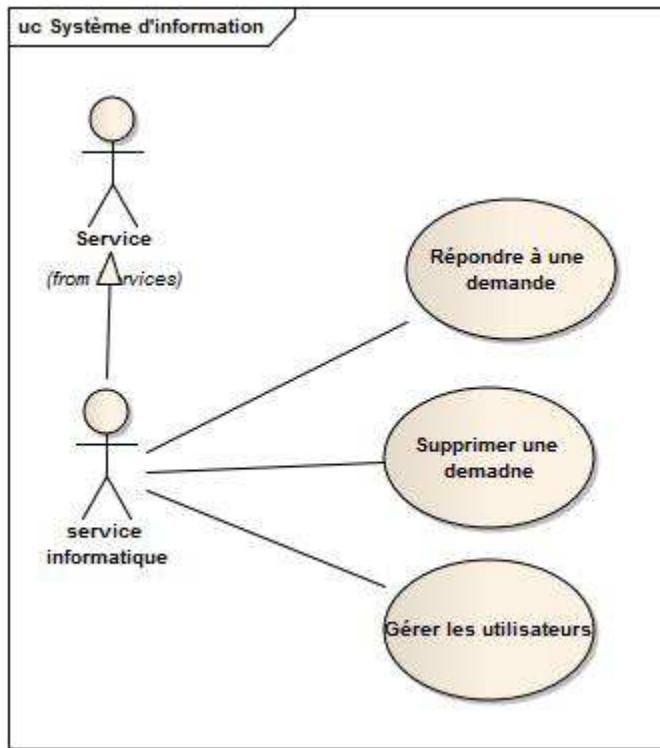
III- MODELISATION DE L'APPLICATION

1- Diagramme des cas d'utilisation

- Acteur : Employé d'un service autre que celui d'informatique



- Acteur : Employé du système d'information



2- Modélisation de la navigation

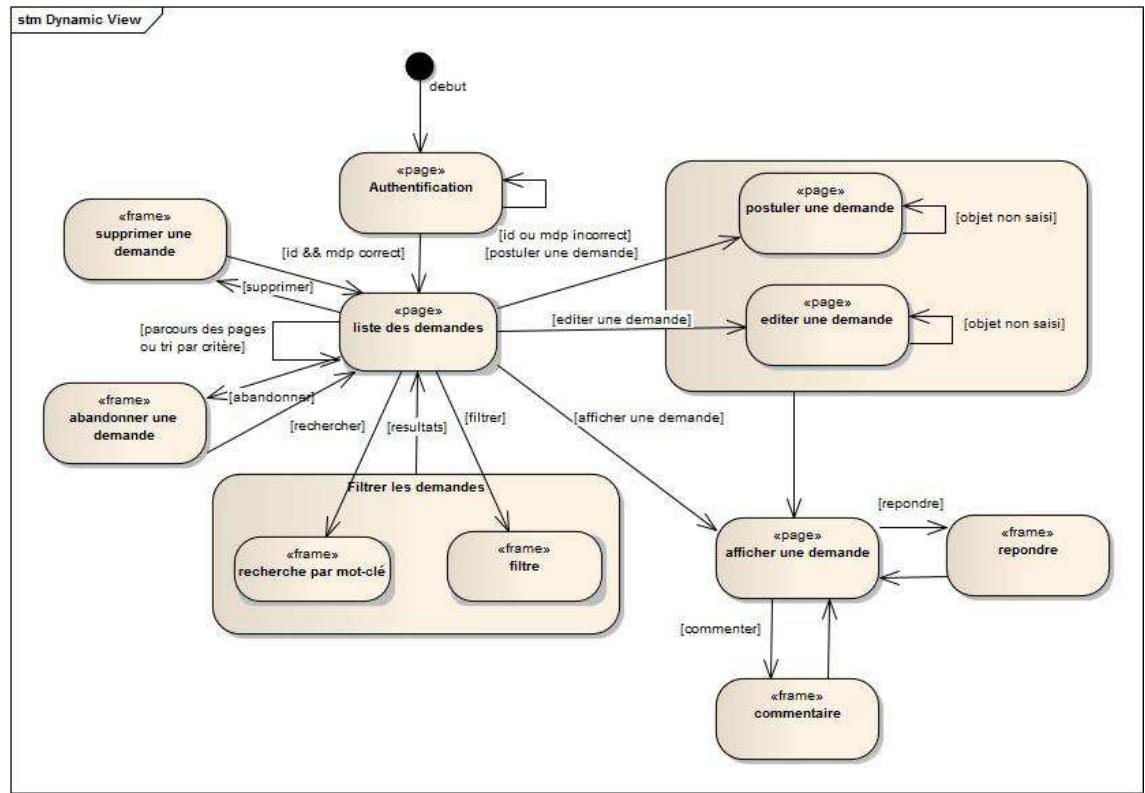
Pour représenter l'ensemble des chemins possibles entre les principaux écrans proposés à l'utilisateur, j'ai utilisé un diagramme d'états-transitions avec un nombre restreint d'éléments standards, à savoir :

- des états pour représenter les classes dialogues
- des transitions entre états déclenchées par des événements et pouvant porter des conditions, pour représenter les actions IHM

Les conventions graphiques suivantes sont utilisées :

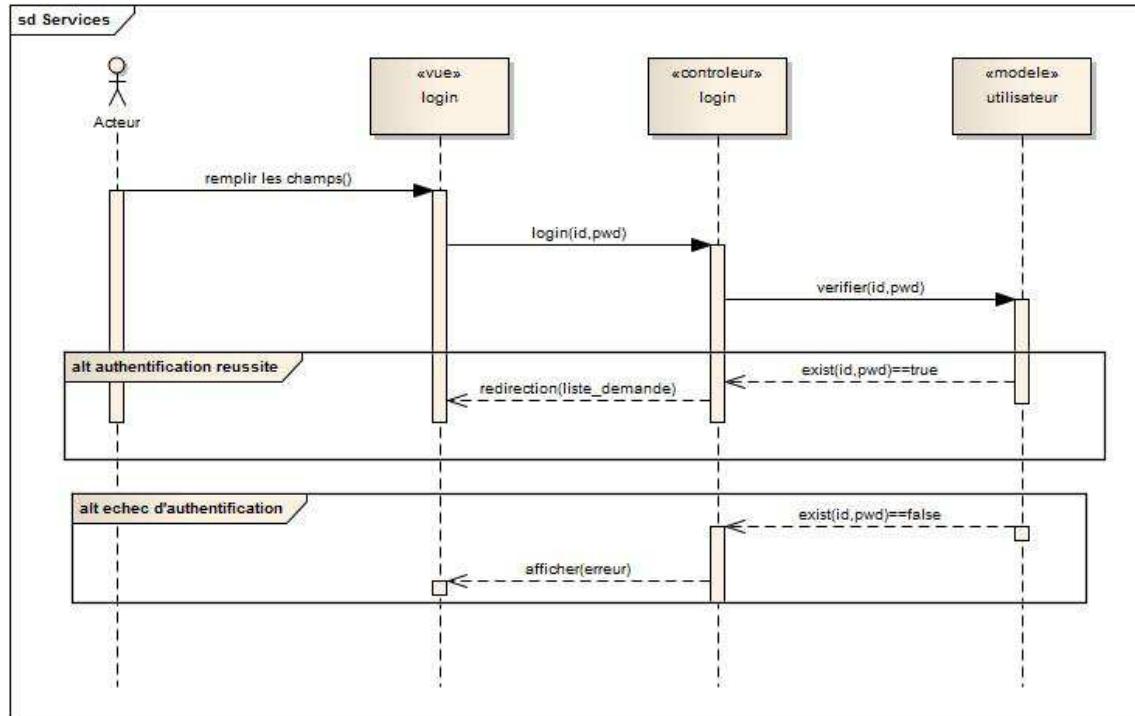
- << page >> pour une page complète
- << frame >> pour un frame particulier à l'intérieur d'une page

- Diagramme de navigation

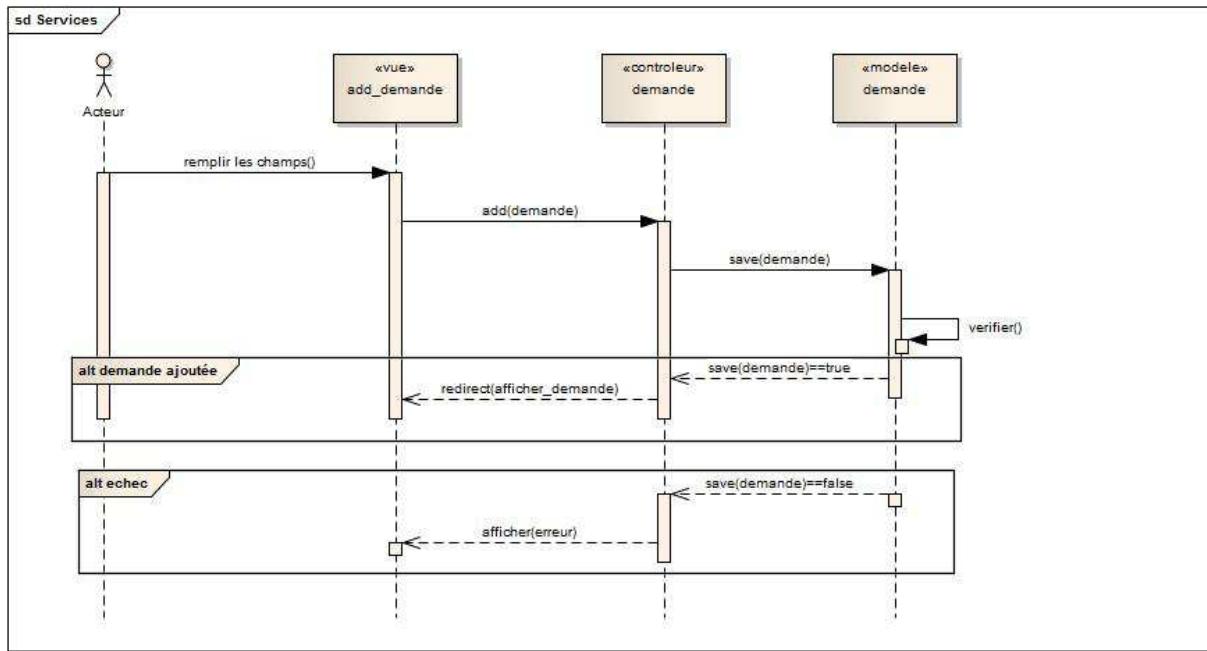


3- Diagrammes de séquences

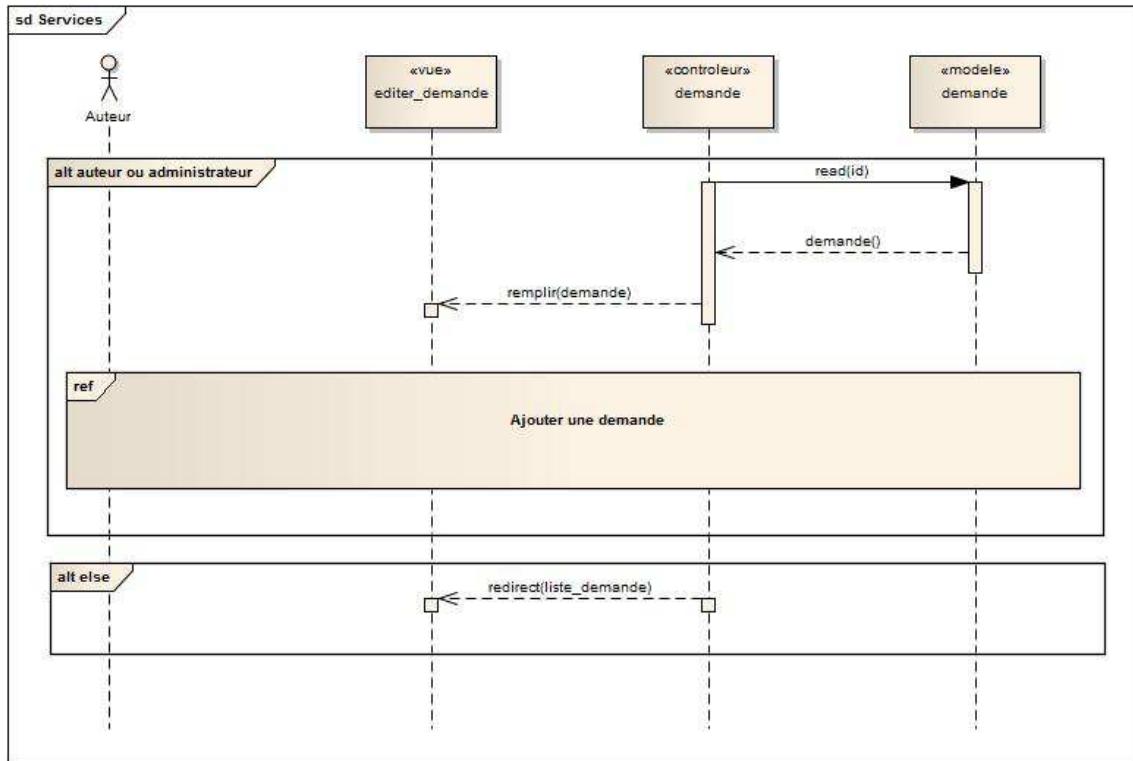
➤ Authentication



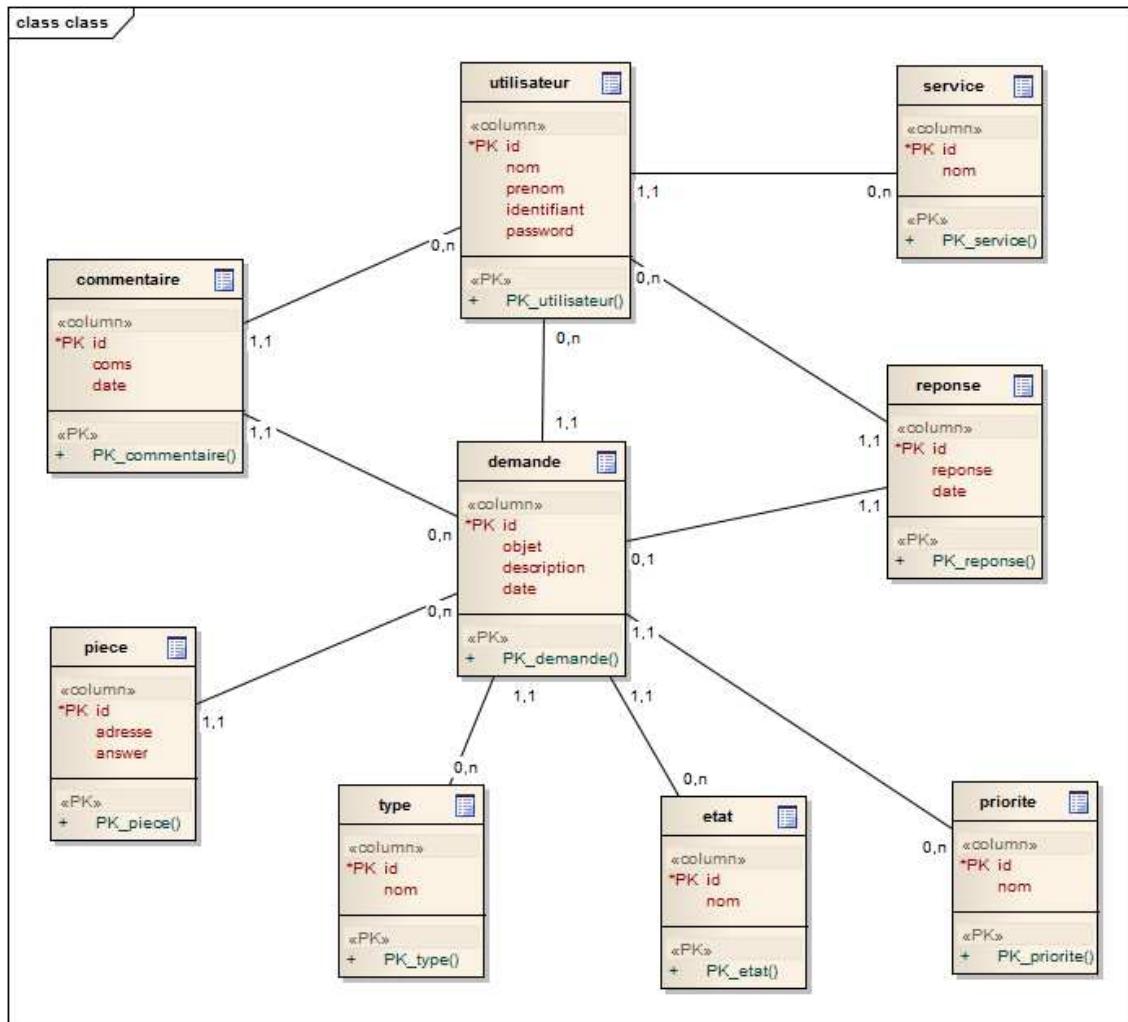
➤ Ajout d'une demande



➤ Edition d'une demande



4- Modèle conceptuel de données



5- Modèle logique de données

Les tables à implémenter sont :

- **Utilisateur** ([id](#),[service_id](#), nom, prenom, tel, identifiant, password)
- **Service** ([id](#), nom)
- **Demande** ([id](#),[utilisateur_id](#),[type_id](#),[etat_id](#),[priorite_id](#), objet, description, date)
- **Relation** ([demande_id](#),[demande_reference_id](#))
- **Réponse** ([id](#),[demande_id](#),[utilisateur_id](#), reponse, date)
- **Commentaire** ([id](#),[demande_id](#),[utilisateur_id](#),#coms, date)
- **Piece** ([id](#), #[demande_id](#), answer)
- **Type** ([id](#), nom)
- **Etat** ([id](#), nom)
- **Priorité** ([id](#), nom)

IV- REALISATION DE L'APPLICATION

1- Choix de l'outil

Dans le but de réaliser une application performante, sécurisée, bien organisée, structurée et aisément maintenable et évolutive, il est préférable d'utiliser un Framework PHP.

Un Framework est un ensemble d'outils et de composants logiciels organisés conformément à un plan d'architecture et des design patterns. L'ensemble forme un squelette de programme. Il est souvent fourni sous la forme d'une bibliothèque logicielle, et accompagné du plan de l'architecture cible du Framework.

Qu'apportera t-il de plus dans le développement de l'application ?

Un Framework permet de bien démarrer le développement de l'application en apportant :

- Une organisation dans la manière de structurer proprement une application,
- un gain de temps important
- de la sécurité
- une maintenabilité et évolutivité plus aisée
- un outillage important
-

Mon choix a été porté sur cakePHP qui est un Framework de développement rapide pour PHP, gratuit et open-source. Il est caractérisé par :

- Une architecture MVC mise en place et prête à être utilisée
- Une documentation assez complète ainsi qu'une communauté active
- Installation facile et configuration minimale
- Un ensemble d'outils et de fonctionnalités permettant un gain de temps remarquable et une assurance de bien faire les choses puisque ces outils sont bien testés et améliorés par une large communauté.

La liste d'outils et de fonctionnalités que j'ai utilisées lors du développement de mon application est la suivante :

- Fonctions CRUD (create, read, update, delete) intégrées pour les interactions avec la base de données
- Scaffolding (maquettage rapide) d'application : permet de réaliser une application basique afin de tester le schéma de base de données.
- Validation intégrée des données
- Assistant intégré de manipulation de formulaire
- Composants intégrés : Authentification, Session, Pagination

2- Etapes de réalisation de l'application

La prise en main de cakePHP

Pour me familiariser avec cakePHP, j'ai suivi un petit tutoriel de création d'une simple application d'ajout, suppression, édition et affichage.

Je me suis basé principalement sur la documentation de cakePHP tout au long de la réalisation de l'application.

La création et configuration de la base de données

La création de la base de données est effectuée via l'outil phpMyAdmin.
La connexion à la base de données est faite en configurant cakePHP par la modification du code figurant dans le fichier de configuration de la base de données

```
class DATABASE_CONFIG {  
  
    var $default = array(  
        'driver' => 'mysql',  
        'persistent' => false,  
        'host' => 'localhost',  
        'login' => 'root',  
        'password' => "",  
        'database' => 'si',  
        'prefix' => "",  
    );
```

La création des tables, modèles, vues et contrôleurs

Pour la réalisation cette tâche, il est préférable de respecter les conventions de noms imposées par cakePHP afin d'éviter les tracas et les affres de la configuration.

Les conventions de noms de cakePHP sont les suivantes :

a. Conventions pour les modèles et les tables :

- Les noms de classe de modèle sont au singulier (Personne, GrossePersonne)
- Un modèle est créé dans un fichier de même nom et se trouvant dans le dossier « models » (/modèles/personne.php)
- Les noms de tables correspondantes aux modèles CakePHP sont au pluriel et utilisent le caractère souligné (**underscore**). Les tables correspondantes aux modèles mentionnés ci-dessus seront donc respectivement : "personnes", "grosse_personnes"
- Les clés étrangères sont reconnues par défaut grâce au nom (singulier) de la table associée, suivi de "_id".

b. Convention pour les contrôleurs :

- Les noms des classes de contrôleurs sont au pluriel, et se terminent par 'Controller' (PersonnesController, GrossePersonnesController).
- Un contrôleur est créé dans un fichier nom_controller.php situé dans le dossier « controllers » (personnes_controller.php)

c. Convention pour les vues :

Les fichiers de gabarits de vue (template) sont nommés d'après les fonctions du contrôleur qu'elles affichent, sous une forme "soulignée" (underscored). La fonction `soyezPret()` de la classe `PersonnesController` cherchera un gabarit de vue dans : `/app/views/personnes/soyez_pret.ctp`

En respectant ces conventions on aura les fichiers et classes suivants :

- Nom de la table : demandes
 - Classe du Modèle : "Demande", trouvée dans `/app/models/demande.php`
 - Classe du Contrôleur: "DemandesController", trouvée dans `/app/controllers/demandes_controller.php`
-
- Nom de la table : users
 - Classe du Modèle : "User", trouvée dans `/app/models/users.php`
 - Classe du Contrôleur: "UsersController", trouvée dans `/app/controllers/users_controller.php`

La même procédure est suivie pour les autres tables.

Définition des relations entre les modèles

Une des caractéristiques les plus puissantes de CakePHP est sa capacité d'établir les liens nécessaires entre les modèles d'après les informations fournies, et effectuer automatiquement les jointures lors de la lecture depuis la base de données. Dans CakePHP, les liens entre modèles sont gérés par des associations.

a. Les types d'association :

- `hasOne` pour une relation 1 : 1
- `hasMany` pour une relation 1 : n
- `hasAndBelongsToMany` pour une relation m : n
- `belongsTo` est le complément naturel des associations `hasOne` et `hasMany`

b. Relier les modèles :

- Une demande a une réponse, plusieurs commentaires et pièces jointes, elle appartient à un utilisateur, un type, un état et une priorité, on définira donc les associations dans la classe `Demande` comme suit :

```
var $belongsTo = array ('Type', 'User', 'Etat', 'Priorite');  
var $hasOne='Reponse';  
var $hasMany = array('Commentaire','Piece');
```

De même pour les autres modèles on aura :

- Modèle User :

```
var $belongsTo = 'Service';
```

LE NUMERO 1 MONDIAL DU MÉMOIRES



- ```

 var $hasMany = array('Demande', 'Commentaire','Reponse');
 • Modèle Reponse et modèle Commentaire
 var $belongsTo = array ('Demande','User');
 • Modèle Piece:
 var $belongsTo = 'Demande'
 • Modèle Service:
 var $hasMany = 'User'
 • les modèles Type, Etat, Priorite:
 var $hasMany = 'Demande';

```

### **Tester les relations entre les modèles via le maquettage rapide**

Le maquettage rapide permet de réaliser une application d'ajout, suppression, édition et affichage afin de tester le bon fonctionnement des modèles (insertion et lecture depuis la base de données, jointures prises en compte ...)

Pour effectuer un maquettage rapide il suffit de spécifier « var \$scaffold » dans les contrôleurs.

### **Création du système d'authentification**

La création d'un système d'authentification est effectuée par l'intermédiaire de l'assistant « Auth » qui gère la création des variables de session concernant l'utilisateur connecté et la redirection de celui-ci vers la page d'accueil, ou l'affichage d'un message d'erreur en cas de données inexistantes. Il gère aussi la déconnexion en détruisant la session et redirigeant l'utilisateur vers la page d'authentification.

### **Création du système de gestion des demandes**

Pour chaque fonctionnalité (ajout, édition, suppression ...) on procède comme suit :

- Création de la fonction qui gère le processus dans le contrôleur de demandes.
- Création de la vue associée à chaque fonction
- Ajouter, si nécessaire, des règles de validation au modèle
- Tester la fonction séparément du reste de l'application
- Intégrer la fonction et tester son fonctionnement au sein de l'application

### **Intégrer un système d'upload des fichiers**

Afin de gérer les Uploads multiples, il est possible d'intégrer un plugin flash/javascript.

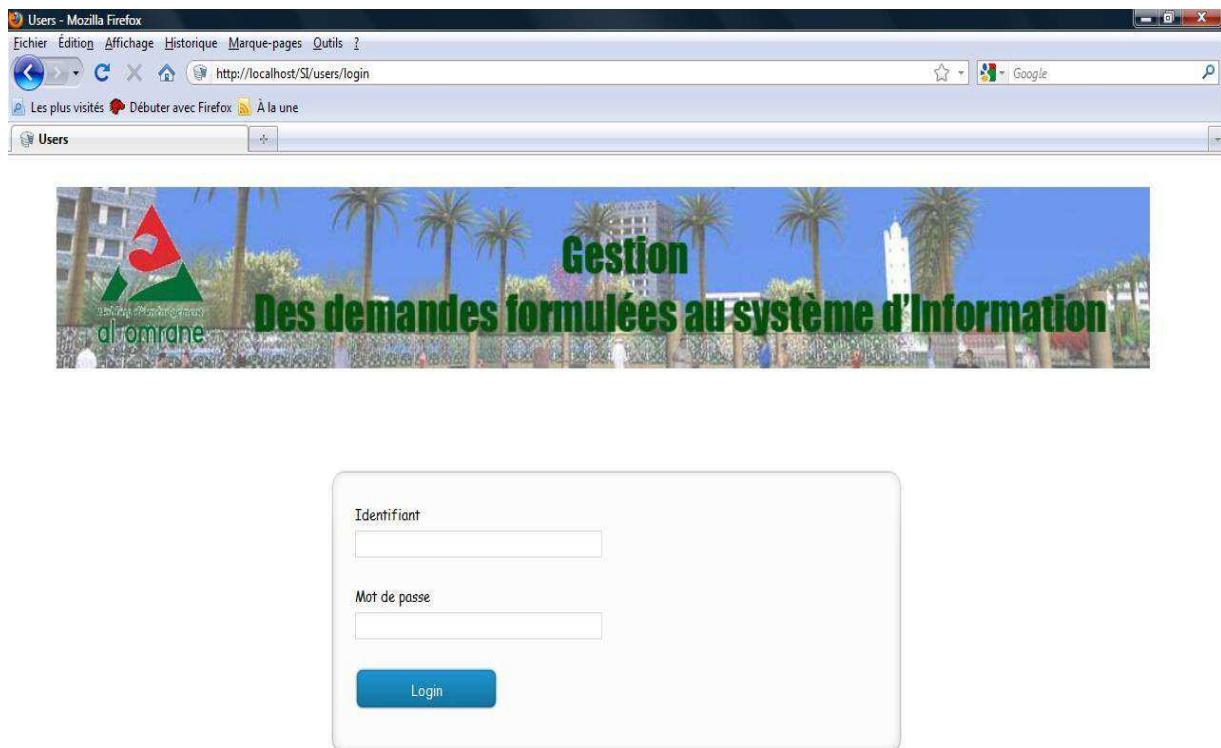
J'ai choisi le plugin Uploadify basé sur la bibliothèque javascript « JQuery », simple d'installation et de configuration. Il offre un design attrayant avec une barre de progression, donne également des possibilités d'annuler un upload, restreindre les types de fichiers que l'utilisateur peut sélectionner et bien d'autres fonctionnalités.

## Création du système de gestion des utilisateurs

En suivant la même procédure que celle adoptée pour la création du système de gestion de demandes, on arrive à créer un système de gestion d'utilisateurs (ajout, suppression, édition, affichage) accessible aux administrateurs.

### 3- Présentation de l'application

La connexion à l'application se fait via un navigateur WEB, la figure ci-dessous présente l'interface d'accès à l'application.



➤ Formulaire d'ajout d'une demande

**Postuler une demande**

Objet

Description

Type  
 migration

Priorité  
 normal

Pièces jointes  
  
 Système d'information... (16.89KB)

**Actions**

- [Retour à la liste des demandes](#)
- [Déconnexion](#)

**Recherche**

Le formulaire ci-dessus permet à un utilisateur de postuler une demande.

Pour cela, il doit remplir les champs objet et description, sélectionner le type et la priorité de sa demande, joindre des fichiers si nécessaire et valider via le bouton ‘Postuler’.

A noter que si le champ objet est vide, la demande ne sera pas ajoutée avec affichage d'un message d'erreur.

#### ➤ Formulaire d'édition d'une demande

**Editer la demande**

Objet  
 demande 2

Description  
 Description demande 2

Type  
 migration

Priorité  
 normal

Pièces jointes  
  
 Excel.xlsx

**Actions**

- [Postuler une demande](#)
- [Abandonner](#)
- [Retour à la liste des demandes](#)
- [Déconnexion](#)

**Recherche**

Le formulaire d'édition est semblable à celui d'ajout, il offre de plus la possibilité de supprimer les pièces jointes à la demande.

➤ Formulaire de réponse

**Réponse**

Cette demande n'est pas encore traitée

**Répondre**

Reponse  
réponse

Relier :  
5 7

Pièces jointes  
Joindre

**Repondre**

L'employé du système d'information répond à une demande via le formulaire ci-dessus.

Pour cela, il remplit le champ « réponse », il peut aussi joindre des fichiers ou mentionner les demandes en relation avec celle courante en saisissant leurs ids dans le champ « relier ».

## CONCLUSION

Les tests effectués tout au long du développement de cette application ont fait preuve de son fonctionnement correct répondant ainsi aux besoins exprimés et l'atteinte des objectifs fixés.

Une fois installée, cette application contribuera à la modernisation du système d'information d'Al Omrane d'Oujda.

Toute fois, cette application peut être améliorée en ajoutant les fonctionnalités suivantes :

- Page de statistiques
- Possibilité d'export sous plusieurs formats
- Interfaçage avec l'intranet afin d'avoir une authentification unique pour accéder à toutes les fonctionnalités exposées sur l'intranet, en ajoutant un web service pour l'authentification.

Aussi, le développement de cette application m'a permis de mettre en pratique mes connaissances théoriques acquises au cours de ma formation à la FST de Fès. Ce travail était également l'occasion d'effectuer des recherches en vue d'améliorer mes compétences en matière d'informatique et principalement le développement Web.

A signaler également que la période de ce stage, au sein de la société Al Omrane d'Oujda, m'a permis de prendre connaissance des pratiques de l'entreprise, des relations humaines et de l'esprit de travail en groupe.

## BIBLIOGRAPHIE

### Livre

Pascal Roques; UML 2 Modéliser une application web ; EYROLLES

### Sites Web

<http://www.alomrane-oujda.ma>

<http://www.cakephp.org>

<http://www.uploadify.com>

<http://www.wikipedia.org>