

OpenLogement Analyse

Date : 23 Août 2006

Indice de révision du document : 0.04

Auteurs : Annie Gay & Florian Signoret

Table des mises à jour du document

Indice révision du document	Date	Objet de la mise à jour
0.00	4/05/2006	Création du document
0.01	31/05/2006	Correction
0.02	28/07/2006	Document complété
0.03	18/08/2006	Document complété
0.04	23/08/2006	Ajout des Annexes

Sommaire

I OBJET.....	5
1 . Objectifs.....	5
2 . Champ couvert.....	5
3 . Contexte organisationnel - acteurs concernés.....	5
II ANALYSE DU METIER.....	6
1 . Cas d'utilisation.....	6
2 . Procédures.....	7
III EXPRESSION DES BESOINS.....	10
1. Besoins non fonctionnels.....	10
2. Besoins fonctionnels.....	10
2.1 Codification des adresses.....	10
2.2 Plafonds.....	10
2.3 Identification des bâtiments et des groupes dans le SIG.....	10
2.4 Propositions.....	10
2.5 Suivi.....	10
2.6 Statistiques.....	10
3. Moyens mis en oeuvre.....	11
IV CONCEPTION DE LA BASE DE DONNEE.....	12
1. Diagramme de classes global.....	12
2. Parc locatif.....	13
3. Demande.....	14
4. Demandeur.....	15
5. Suivi.....	16
6. Urbanisme.....	16
7. Autres tables.....	16
V REALISATION.....	17
1. Gestion de la demande.....	17
1.1 Fonctionnement général.....	17
1.2 Onglets.....	17
1.3 Récupération de demandeurs existants.....	17

1.4 Lieu de naissance.....	17
1.5 Liste de ressources, liste de charges, liste des revenus imposables.....	17
1.6 Sélection de la ville.....	18
1.7 Codification de la voie.....	18
1.8 Validation de la demande.....	18
2. Suivi.....	18
3. Recherche de demande.....	18
4. Gestion du parc locatif.....	18
5. Gestion des propositions.....	18
6. Gestion des relances.....	18
7. Génération des fichiers PDF.....	18
8. Migration des données de Logem vers mySQL.....	19
VI SUITE DU PROJET.....	20
1. Codification des adresses.....	20
2. Calcul du type de ménage.....	20
3. Automatisation de l'état des logements.....	20
4. Edition de la lettre de conseil de demandeurs aux sociétés HLM.....	20
5. Affichage du code postal.....	20
6. Aide.....	20
7. Export de données.....	20
8. Archivage.....	20
9. Indication du plafond correspondant à la demande.....	20
10. Facilité de navigation.....	20
11. Import des revenus imposables.....	21
12. Résiliation spontanée.....	21
ANNEXES.....	22

I OBJET

1. Objectifs

OpenLogement a pour but d'aider le service Habitat de la ville d'Arles à suivre la demande et l'attribution des logements sociaux. OpenLogement doit également assister le service dans son conseil de demandeurs aux sociétés HLM.

Le service Habitat utilise actuellement le logiciel propriétaire « Logem » (Société SIGEC). Les reproches faits à ce produit sont :

- son coût de maintenance de 2000 €/an environ.
- son manque de portabilité. (Microsoft Windows et Microsoft Word requis)
- l'absence d'interface géographique. (SIG)
- la présence de fonctionnalités non utilisées.
- l'absence de fonctionnalités souhaitables.
- l'inexactitude de certains calculs et traitements.
- l'aspect fastidieux des outils statistiques mis à disposition.

2. Champ couvert

Les grandes fonctions de l'application sont :

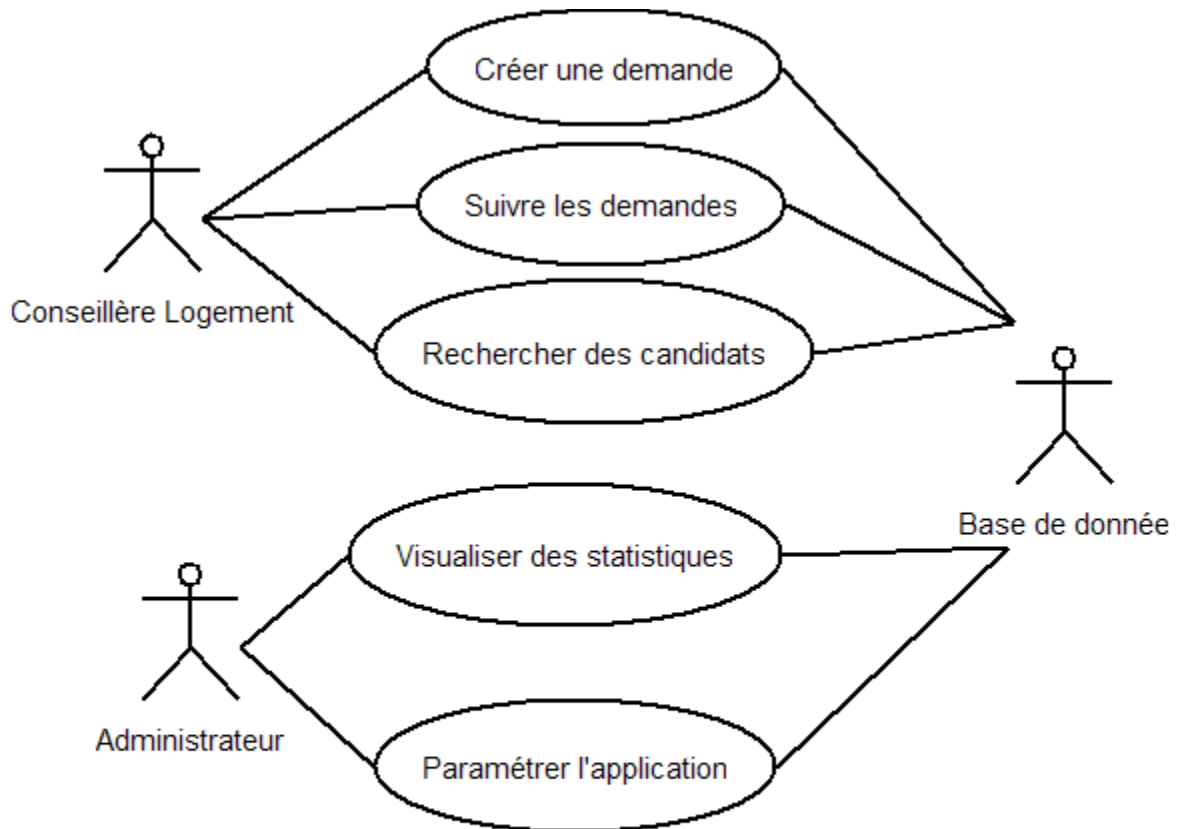
- La création des demandes.
- La recherche de demandes selon différents critères.
- La gestion du parc locatif.
- La gestion des propositions.
- La gestion des relances de dossier.
- L'archivage des demandes anciennes.
- La visualisation de statistiques.
- Le paramétrage de l'application, la gestion des données support.

3. Contexte organisationnel - acteurs concernés

Trois personnes au rez-de-chaussée (conseillères logement) s'occupent des 5 premières fonctions. Une personne à l'étage (administratrice) est chargée des 3 autres fonctions.

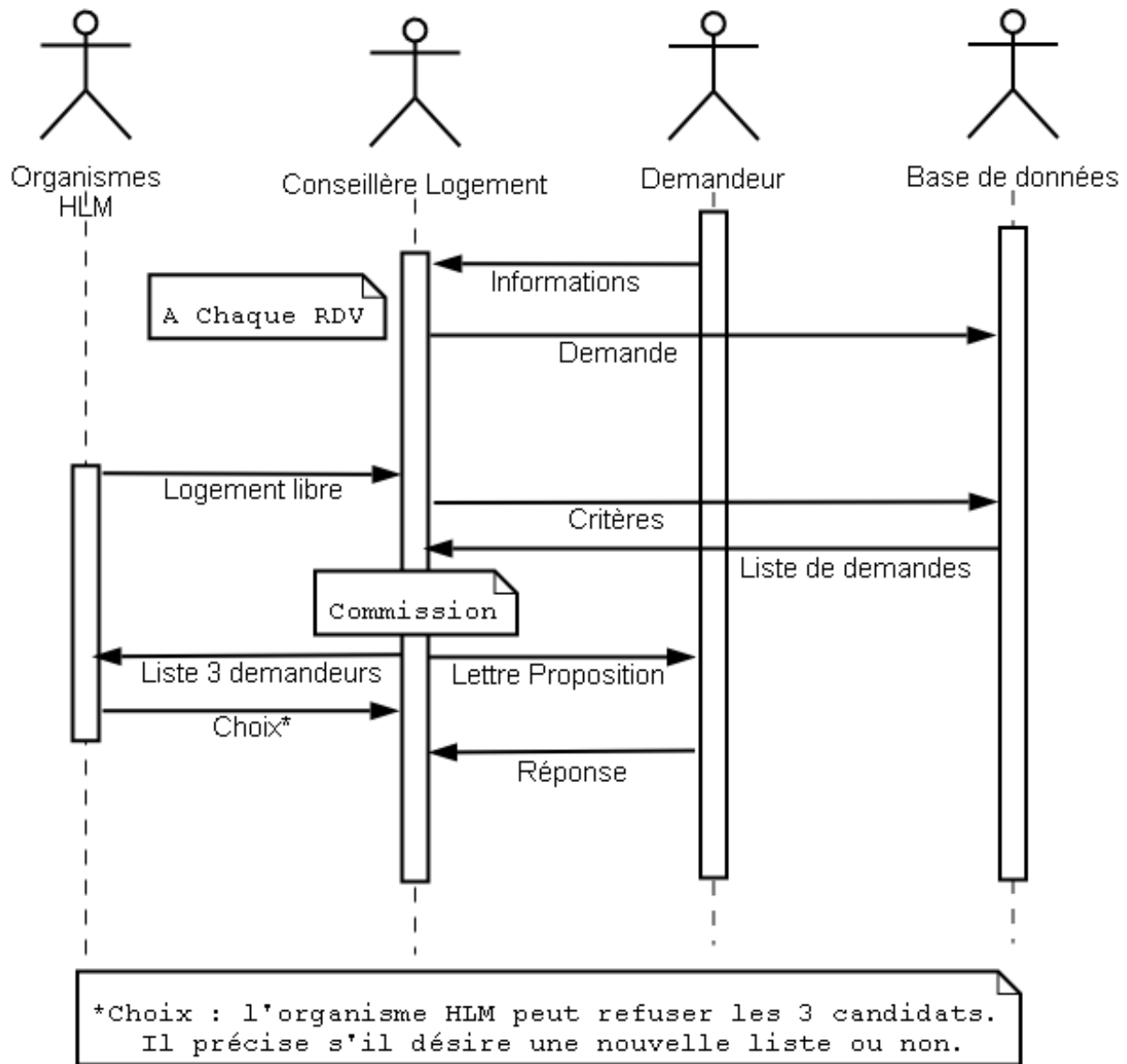
II ANALYSE DU METIER

1. Cas d'utilisation



Projet : OpenLogement
Titre : Cas d'utilisations
Date : 9/05/2006
Auteurs : Annie Gay, Florian Signoret

2. Procédures



Projet : OpenLogement
 Titre : Diagramme de séquence
 Date : 9/05/2006
 Auteurs : Annie Gay, Florian Signoret

Les personnes souhaitant bénéficier d'un logement HLM doivent tout d'abord se présenter chez les sociétés HLM de leur choix et y constituer un dossier de demande.

Ce n'est qu'après cette démarche que ces personnes peuvent, si elles le souhaitent, prendre rendez-vous avec le service Habitat. Elles amènent un certain nombre de pièces justificatives pour renseigner leur dossier. La conseillère logement peut alors enregistrer la demande.

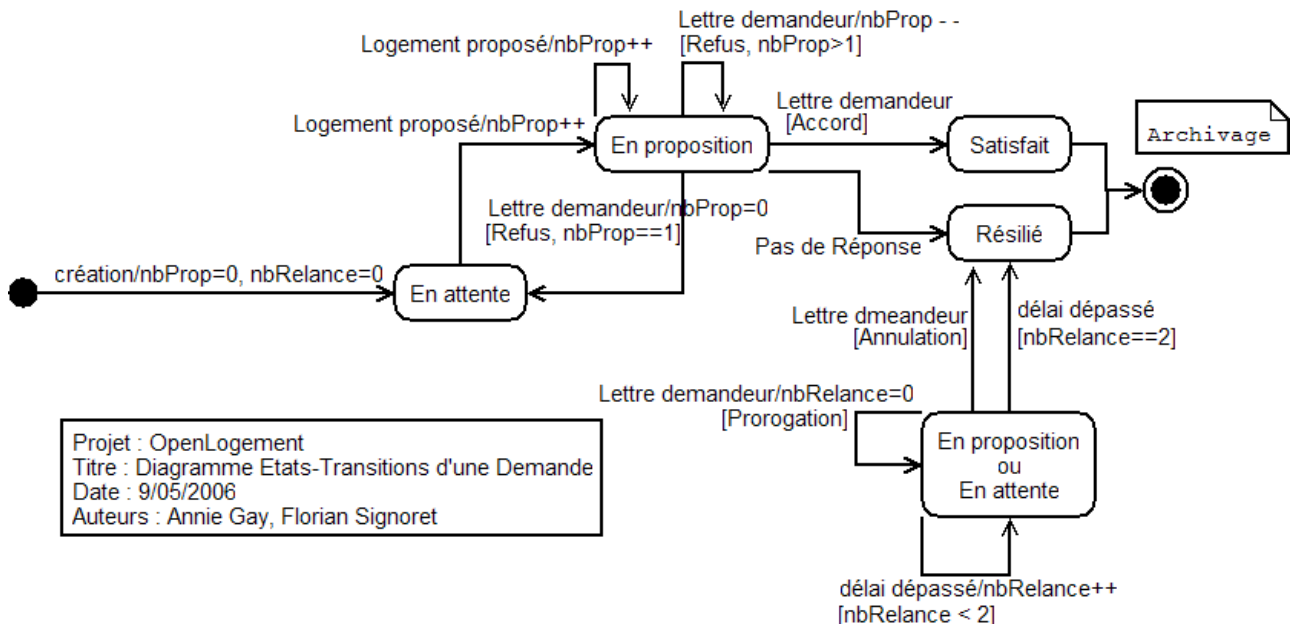
Les demandeurs peuvent ensuite prendre rendez-vous avec une assistante sociale ou avec un élu.

Lorsqu'un logement se libère, la société HLM envoie un courrier pour le signaler au service Habitat. Cette lettre contient un « numéro de bailleur » qui n'est autre que le numéro de logement attribué par la société.

Les conseillères de logement doivent alors, à l'aide de la recherche multicritères, et après un travail d'analyse des demandes jugées les plus urgentes, constituer une certaine liste de candidats potentiels. Cette liste est ramenée au nombre de 3 à 4 candidats lors d'une commission réunissant l'ensemble des décideurs du service. Un ordre de priorité est également défini.

Cette liste est envoyée à la société HLM. Les demandeurs concernés reçoivent aussi un courrier les invitant à se rapprocher de la société HLM en question.

La société HLM peut choisir un des candidats proposés, n'en choisir aucun, ou demander une autre liste. Elle fait parfois connaître son choix au service Habitat, mais il arrive que le service soit contraint de se renseigner auprès de la société ou du demandeur.



A sa création, une demande est tout d'abord en état d'Attente. Par la suite, elle peut se voir attribuer une ou plusieurs propositions. Lorsqu'on apprend le résultat d'une proposition, l'issue est enregistrée et l'état de la demande est mis à jour.

Si un demandeur a accepté une proposition, la demande est satisfaite. A tout moment, le demandeur peut résilier sa demande.

Chaque année, le service Habitat envoie une lettre (appelée relance) aux personnes dont la demande est encore active (état attente ou proposition) leur demandant si elles souhaitent maintenir la demande ou pas. Si, au bout d'un mois, aucune réponse n'a été donnée, une deuxième relance est envoyée. Si, au bout d'un mois, aucune réponse n'a été donnée, la demande est automatiquement résiliée.

Le service Habitat songe à effectuer ces procédures tous les mois pour répartir la charge de travail sur l'année entière. Les relances seraient donc envoyées aux demandes n'ayant pas été relancées depuis au moins 1 an.

III EXPRESSION DES BESOINS

1. Besoins non fonctionnels

- Stabilité : l'application doit être assez stable car son utilisation est indispensable durant les entretiens avec les demandeurs de logement.
- Evolutivité : l'application pouvant être reprise par d'autres mairies, la conception devra prévoir d'éventuelles adaptations et évolutions.
- Facilité d'emploi.
- Sécurité : les informations nominatives sont des informations à protéger, conformément à la loi.

2. Besoins fonctionnels

Les fonctionnalités nécessaires sont celles de l'existant, Logem, avec des ajouts, des modifications et des suppressions précisées par le service Habitat dans le document intitulé « Logiciel métier – Besoins du service Habitat ».

2.1 Codification des adresses

Actuellement, les adresses sont de simples champs texte. Afin d'être capable de géolocaliser les demandes, il est demandé d'identifier chaque voie arlésienne et chaque ville, à l'aide du fichier du service de la BDU.

2.2 Plafonds

Les revenus imposables des demandeurs ne doivent pas dépasser certains plafonds. Chacun des plafonds est propre à un mode de financement des projets HLM. Ces plafonds sont définis chaque année et pour chaque type de ménage par l'Etat et sont :

- le PLA-I (anciennement appelé PLAT)
- le PLUS (anciennement appelé PLAPLF)
- le PLI
- le PLS

Il est demandé, pour chaque demande, qu'il soit affiché, dès que les informations nécessaires sont disponibles :

- le pourcentage du dernier revenu imposable des demandeurs par rapport au PLUS de l'année N+2, N étant l'année d'imposition.
- Le nom du plus bas plafond qui n'a pas été dépassé par ce revenu imposable.

Le revenu imposable du demandeur principal est pris en compte. Si des revenus ont été déclarés par le conjoint la même année, ils sont ajoutés.

2.3 Identification des bâtiments et des groupes dans le SIG

Il est nécessaire de prévoir de récupérer les identifiants attribués par le SIG de la BDU aux bâtiments et aux groupes.

2.4 Propositions

Il doit être possible de proposer plusieurs logements à la fois à une demande donnée.

Il doit être possible de proposer un logement donné à plusieurs demandes.

2.5 Suivi

Les différents événements ayant affecté une demande doivent pouvoir être résumés chronologiquement à tout moment.

2.6 Statistiques

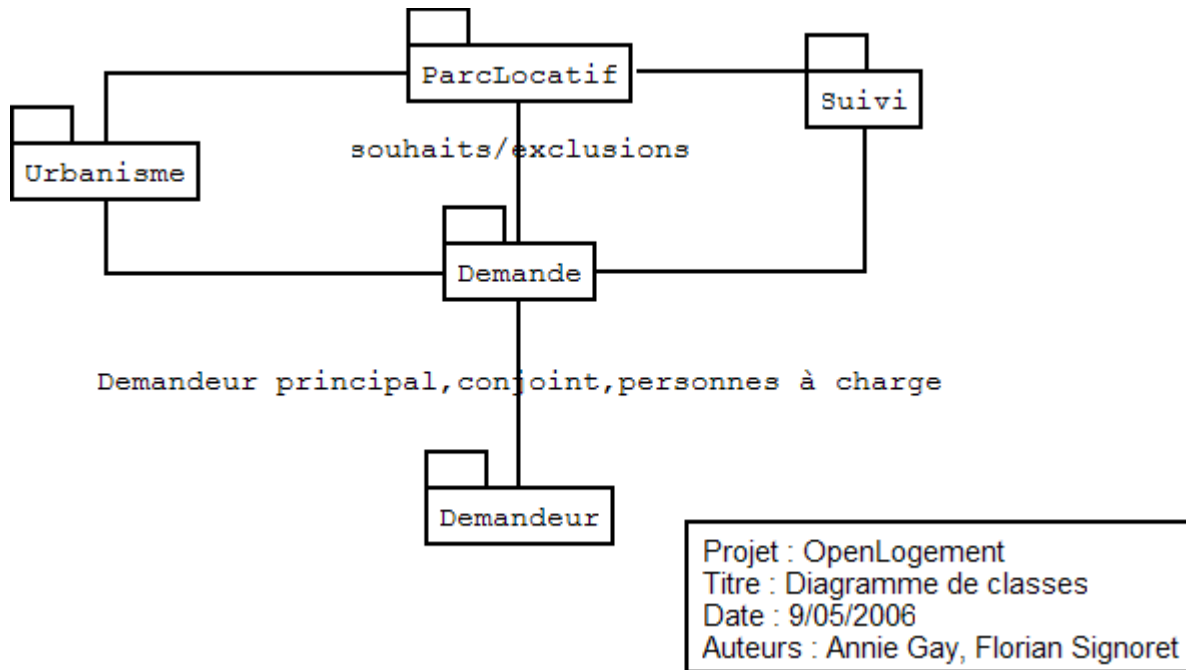
L'utilisateur souhaite être capable d'étudier tous les critères statistiques possibles fournis par la base de données.

3. Moyens mis en oeuvre

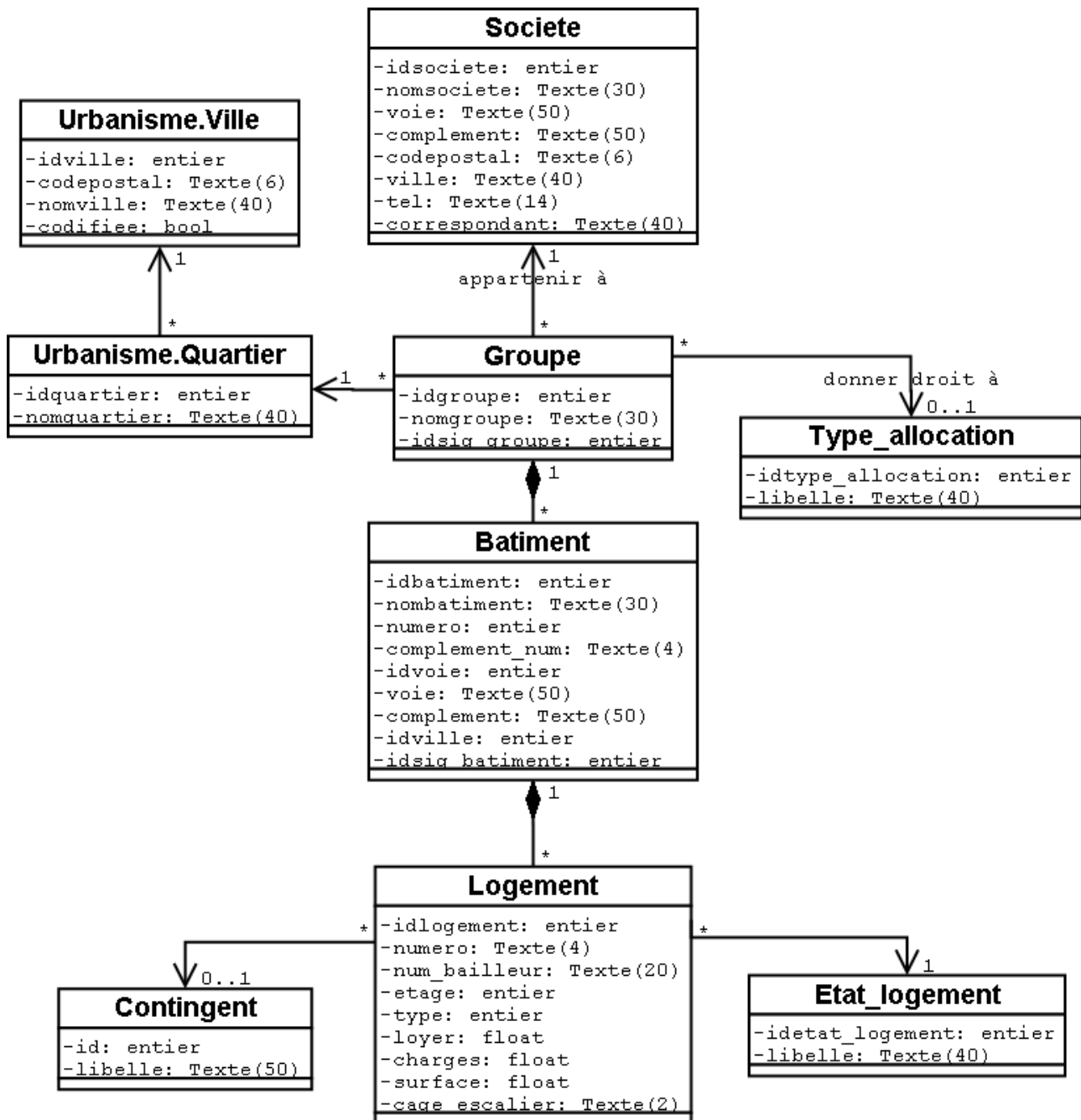
L'application est développée en PHP/MySQL à partir de la base OpenMairie.

IV CONCEPTION DE LA BASE DE DONNEE

1. Diagramme de classes global

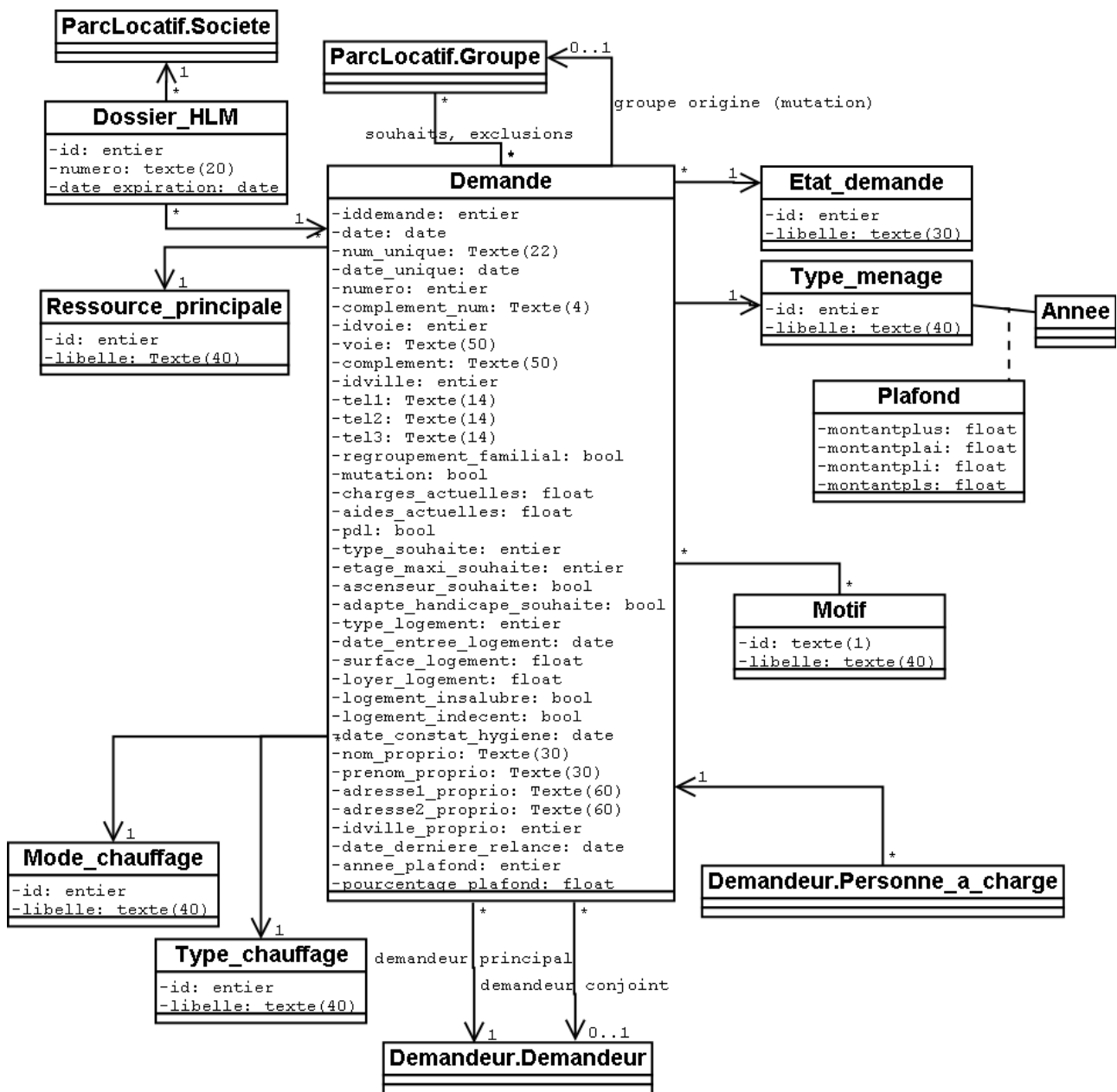


2. Parc locatif



- numero : numéro dans l'adresse
- complement_num : bis, ter etc...
- idvoie : identifiant de la voie dans l'adresse, si la société se trouve dans une ville codifiée
- voie : nom de la voie dans l'adresse, si la société se trouve dans une ville non codifiée
- complement : complément d'adresse
- type : nombre de pièces du logement
- cage_escalier : numéro ou lettre attribuée à la cage d'escalier du logement

3. Demande



Classe Demande :

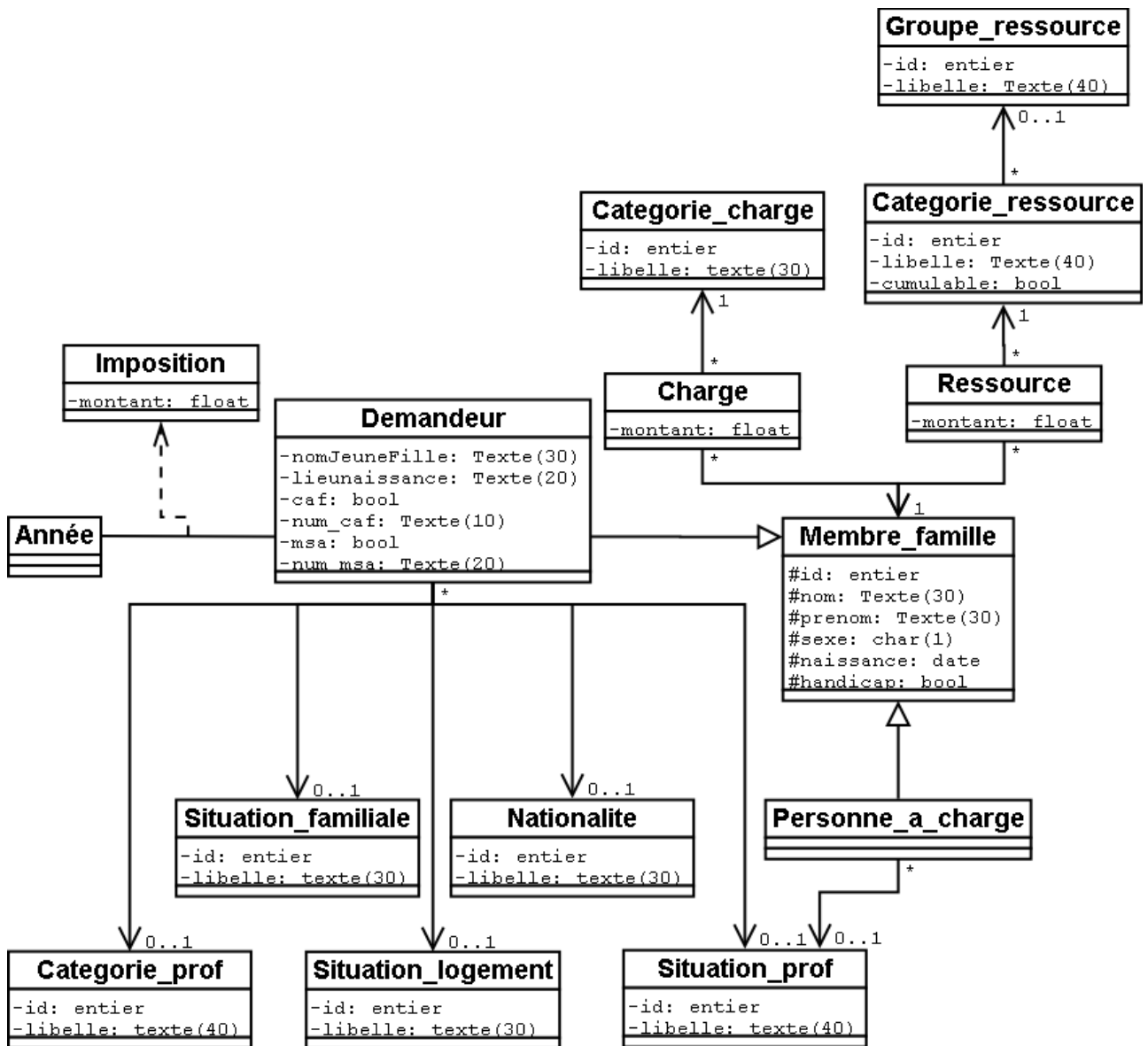
- date : date à laquelle la demande a été enregistrée
- num_unique : numéro unique attribué à la demande par la région
- date_unique : date d'attribution du numéro unique
- numero : numéro dans l'adresse des demandeurs
- complement_num : bis, ter etc...
- idvoie : identifiant de la voie dans l'adresse, si les demandeurs habitent dans une ville codifiée
- voie : nom de la voie dans l'adresse, si les demandeurs habitent dans une ville non codifiée
- complement : complément d'adresse
- regroupement_familial : vrai si la demande se fait dans le cadre d'un regroupement familial
- charges_actuelles : montant actuel des charges du foyer
- aides_actuelles : montant actuel des aides du foyer
- etage_maxi_souhaite : numéro de l'étage maximal accepté par les demandeurs
- ascenseur_possible : vrai si les demandeurs peuvent utiliser l'ascenseur
- adapte_handicape_souhaite : vrai si les demandeurs souhaitent des aménagements pour handicapés.
- type_logement : nombre de pièces du logement actuel des demandeurs
- date_entree_logement : date d'entrée dans le logement actuel des demandeurs

- logement_insalubre : vrai si le logement est insalubre
- logement_indecent : vrai si le logement est indécent

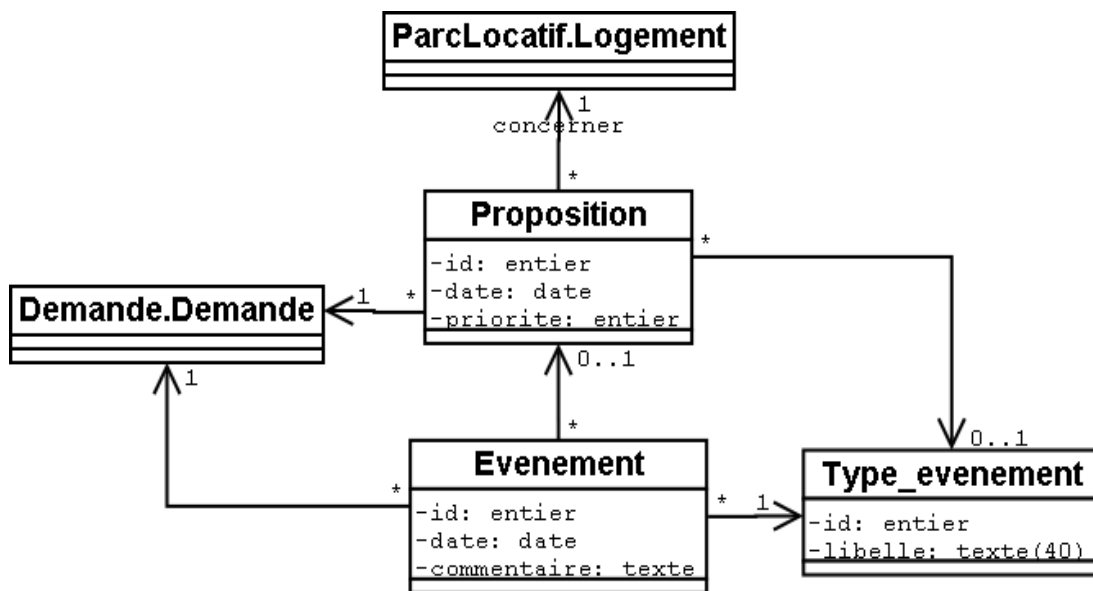
Classe Dossier_hlm :

- num : numéro attribué par la société HLM

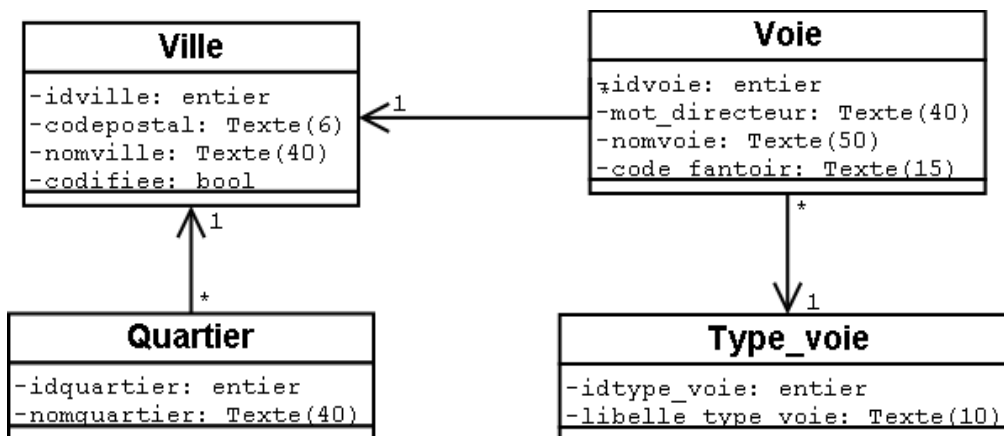
4. Demandeur



5. Suivi



6. Urbanisme



7. Autres tables

Les tables « nom de l'objet »_seq servent à simuler une numérotation automatique compatible avec à la fois MySQL et PostgreSQL.

Les tables collectivite, droit, profil et utilisateur sont propres à OpenMairie.

La table relance_a_faire liste les relances proposées à l'utilisateur, la table relance liste les relances en cours, la table suite recense les libellés des actions possibles : Relance, Résiliation.

V REALISATION

1. Gestion de la demande

1.1 Fonctionnement général

La demande est l'objet principal et donc le plus complexe de l'application. Les possibilités offertes par openMairie pour la gestion d'un objet sont très insuffisantes pour pouvoir répondre correctement aux besoins. En effet, la facilité de programmation par le paramétrage amène l'inconvénient du manque de liberté pour une personnalisation marquée.

Afin de garder l'utilisation d'openMairie, la solution que j'ai choisie est de redéfinir complètement la méthode « formulaire() ». Au lieu d'utiliser un objet de la classe « form », elle affiche directement les balises HTML du formulaire. Ainsi, j'ai pu introduire des mécanismes complexes, impossibles à mettre en oeuvre par le paramétrage de l'objet form.

1.2 Onglets

Le nombre de champs étant important, j'ai utilisé un système d'onglets supplémentaires. Il utilise la propriété CSS « display » et des fonctions Javascript définies dans /obj/onglets.js.

1.3 Récupération de demandeurs existants

Dans Logem, chaque demande possède son propre demandeur. (ses attributs sont même placés dans la table demande) Même s'il s'agit d'une deuxième demande, toutes les informations du demandeur sont redemandées.

L'analyse a préconisé la séparation du demandeur et de la demande, avec possibilité d'utiliser un demandeur pour plusieurs demandes.

Selon le service Habitat, il n'est pas rare que cette situation se produise, et j'ai donc mis en place un bouton de recherche de demandeur par le nom et le prénom. Si le demandeur est trouvé, son identifiant et toutes ses informations connues sont préremplies. Les boutons radio permettent de choisir la création ou la réutilisation. Les données peuvent aussi être mises à jour lors d'une réutilisation.

Ce système a, selon moi, des limites : il suppose que l'utilisateur va effectuer systématiquement la recherche, ce qui n'est pas du tout une certitude ; il n'est peut être pas souhaitable que « l'état du demandeur à la création de l'ancienne demande » soit changé, pour des raisons statistiques ou administratives ; l'archivage tel qu'il sera imposé par la CNIL pourra limiter ou faire disparaître la présence de plusieurs demandes pour un seul demandeur.

De toutes manières, le fonctionnement en mode doublon n'est pas rendu impossible, et les boutons cités précédemment peuvent aisément être retirés.

1.4 Lieu de naissance

Il s'agit d'un champ texte, sans aucune codification.

1.5 Liste de ressources, liste de charges, liste des revenus imposables

Cet outil, non disponible dans openMairie, permet de gérer une liste d'éléments se trouvant dans une table différente de l'objet traité, avec une relation de un à plusieurs. Dans l'utilisation classique d'openMairie, ils auraient donné lieu à la création de nombreux sous-formulaires.

L'outil utilise les fonctions javascript des fichiers /obj/items.js et /obj/impots.js.

L'envoi d'un tableau par formulaire nécessite la sélection de toutes les lignes lors de la validation. Ceci est fait dans le fichier /obj/verifForms.js.

1.6 Sélection de la ville

OpenMairie fournit l'élément « combog » qui permet de remplir un champ à partir d'un autre. Une utilisation classique dicterait que l'on affiche deux champs (l'identifiant de la ville et son nom), avec une flèche allant du nom vers l'identifiant.

Cependant, jugeant ce fonctionnement peu intuitif et pouvant induire en erreur si le nom de la ville est modifié avant l'utilisation de la flèche, j'ai choisi de le personnaliser. J'ai ajouté un champ de recherche, caché l'identifiant de la ville et grisé le nom de la ville. Ainsi, l'utilisateur peut entrer un texte de recherche dans le champ non grisé, lancer la recherche en cliquant sur la flèche, et savoir à tout instant quelle valeur est réellement prête à être envoyée par le formulaire.

1.7 Codification de la voie

Pour certaines villes « codifiées », on souhaite repérer géographiquement la voie exacte utilisée dans l'adresse, au lieu d'avoir une chaîne de caractères inexploitable par un SIG.

Ce champ n'est rendu disponible que si une ville codifiée a été sélectionnée au préalable.

1.8 Validation de la demande

Lors de l'ajout ou de la modification de la demande, on vérifie par javascript que les champs obligatoires ont bien été remplis. La fonction se trouve dans /obj/verifForms.js.

2. Suivi

Il s'agit de l'historique des évènements ayant affecté la demande. Cette table est remplie par les traitements.

3. Recherche de demande

Les onglets sont identiques à ceux de la demande. (fichier /obj/onglets_recherche.js)

Le formulaire de recherche est codé dans /scr/recherchedemande.php.

Il appelle /scr/tab.recherchedemande.php qui utilise la classe /obj/tabrecherchedemande.class.php.

Lors de la validation d'une recherche, tous les paramètres sont enregistrés dans des variables de session. Ainsi, si l'on veut modifier la recherche, le formulaire récupère les anciennes valeurs.

4. Gestion du parc locatif

L'organisation du parc locatif est « en entonnoir ». En effet, des sociétés contiennent des groupes qui contiennent des bâtiment qui contiennent des logements.

OpenMairie ne permet pas de créer des sous-formulaires dans un sous-formulaire. Pour remédier à ce problème, j'ai ajouté un lien hypertexte dans chaque sous formulaire. Ainsi, on peut atteindre l'enregistrement du sous-formulaire en cours, mais au niveau inférieur. La navigation n'est pas idéale, mais est tout de même améliorée.

5. Gestion des propositions

Il s'agit d'un sous formulaire de l'objet logement.

A la création ou la modification d'une proposition, des traitements sont effectués pour mettre à jour les différentes données liées à la proposition. Pour cela, j'ai redéfini les méthodes supprimer, modifier et triggerajouter.

6. Gestion des relances

La page « générer relances » permet de proposer une liste de relance, à partir des relances existantes, de l'état des demandes de la base de données, et de la date de leur dernière relance. Ce n'est qu'une fois le bouton « Effectuer » utilisé que la relance sera effective et accessible dans le menu « relance en cours ». Les lettres de relance sont disponibles dans le menu « Lettres relances ».

7. Génération des fichiers PDF

La génération d'état d'OpenMairie ne permet pas de mettre en forme un mot donné ni de faire des présentations recherchées. Cependant, il est intéressant de laisser la possibilité à l'utilisateur de modifier certains contenus.

OpenLogement : Analyse

C'est pour cela que j'ai à la fois utilisé l'outil « Paramétrage=>Etats » et à la fois personnalisé la génération des fichiers PDF en codant directement avec la classe FPDF. Les fichiers créés se trouvent dans /pdf/.

8. Migration des données de Logem vers MySQL

Un outil de traitement a été programmé. (dossier « migration ») Des messages permettent de résoudre les éventuels problèmes nécessitant l'intervention humaine, notamment lors de l'attribution d'une ville à un couple code postal – nom de ville ambigu.

Suivre les instructions de la page index.

VI SUITE DU PROJET

Cette partie présente les fonctionnalités qui n'ont pas eu le temps d'être programmées. Certaines amélioreraient l'application, d'autres sont indispensables pour une mise en exploitation.

1. Codification des adresses

Après la migration des données de Logem, les adresses ne sont pas codifiées. Elles ont toutes été placées dans le champ « voie » de type texte. Deux solutions sont possibles : codifier à la main ces adresses, au fur et à mesure, et travailler avec les deux systèmes ; programmer un script qui détecte le mot directeur et essaye de trouver l'adresse adéquate pour chaque demande, semblable à l'outil de migration des villes.

2. Calcul du type de ménage

Pour l'instant, le type de ménage est modifiable manuellement et doit être renseigné à la création de la demande. Il est nécessaire que ce champ ne soit plus modifiable et se mette à jour tout seul, à partir du nombre de personnes à charge et des dates de naissance des demandeurs. (en effet, un ménage est jeune si la somme des âges des demandeurs ne dépasse pas 55ans)

3. Automatisation de l'état des logements

Pour l'instant, tout changement d'état d'un logement doit être fait manuellement. Il est nécessaire que cela se fasse automatiquement lors, par exemple, de la création d'une proposition ou de la modification du résultat d'une proposition.

4. Edition de la lettre de conseil de demandeurs aux sociétés HLM

Cette lettre n'a pas encore été programmée. Lors de sa conception, il sera nécessaire de prévoir à quel endroit elle sera disponible. En effet, à l'heure actuelle, l'utilisateur crée les différentes propositions séparément. Deux solutions sont possibles : modifier l'application afin que les propositions soient créées toutes en même temps que la lettre de conseil aux sociétés ; récupérer toutes les propositions existantes adéquates lors de l'édition de la lettre. La lettre ne devra pas comporter le logo d'Arles.

5. Affichage du code postal

Le système actuel de sélection d'une ville ne fait pas apparaître le code postal. Il serait judicieux de l'afficher dans le libellé, car un nom de ville n'est pas unique.

6. Aide

L'aide à l'utilisateur n'a pas été créée.

7. Export de données

Le but est de permettre la génération de statistiques. Logem utilise actuellement QBE de WinDesign pour effectuer cette tâche. L'utilisateur peut alors effectuer des statistiques dans un tableur. Ce système ne fonctionne pas bien lors de relations de plusieurs à plusieurs. Une réflexion sur les solutions possibles s'impose.

8. Archivage

La CNIL impose, selon la déclaration simplifiée n°20 à laquelle adhère actuellement le service Habitat, que les informations nominatives des demandes résiliées, satisfaites, ou ayant été créées/renouvelées il y a plus de 5 ans soient supprimées. Ceci empêcherait la création des études statistiques faites régulièrement. C'est pourquoi le service Habitat a effectué une déclaration en ligne normale à la CNIL le 7 Juillet 2006. Il reste à connaître la suite donnée à cette demande, définir et programmer la fonction d'archivage.

9. Indication du plafond correspondant à la demande

L'affichage du plafond doit être programmé, comme défini dans la partie Besoins.

10. Facilité de navigation

Le nombre important de champs dans les formulaires de la demande oblige l'utilisateur à monter et descendre sans cesse dans la page web, à l'aide de l'ascenseur, afin de cliquer sur le menu, sur les onglets

ou sur les boutons. Ceci est fastidieux, surtout pour des utilisateurs non habitués à l'utilisation de la roulette de la souris. Il est nécessaire de trouver une solution à ce problème.

11. Import des revenus imposables

Les années d'un certain nombre de revenus imposables de la base de données Logem ne comportent que 2 chiffres au lieu des 4 habituels. Un « 20 » a été ajouté au début de ces dates, mais cela ne donne pas toujours quelque chose de vraisemblable (imposition en 2008). Il est nécessaire de déterminer, avec le service Habitat, quelle marche à suivre adopter pour le traitement de ces anomalies.

12. Résiliation spontanée

Il est nécessaire de rendre possible la résiliation spontanée d'une demande. En effet, à l'heure actuelle, une demande n'est résiliée que lors de sa satisfaction ou de la non réponse à une deuxième relance.

13. Transactions

Beaucoup de fonctionnalités nécessitent d'effectuer plusieurs requêtes de mise à jour à la fois lors d'un seul clic de souris. Une erreur entre deux requêtes de ce type peut amener des erreurs d'incohérence dans la base de données.

La solution serait d'utiliser le système des « transactions ». Une série de requêtes n'est validée que lorsque les requêtes se sont toutes terminées et si aucune erreur n'est survenue.

La version de mySQL utilisée actuellement par le serveur n'offre pas cette fonction. Il serait donc nécessaire de mettre à jour la version ou d'utiliser des tables de type InnoDB ou BDB après avoir activé le module nécessaire à leur création.

14. Prise en compte des exclusions dans le moteur de recherche

Il serait utile d'ajouter une case à cocher dans l'onglet « souhaits » du moteur de recherche. Cette case serait intitulée « Exiger que le groupe ne soit pas exclu » et permettrait à l'utilisateur d'éviter les demandes qui excluent le groupe faisant l'objet de la recherche.

ANNEXES

- 1. Logiciel métier – Besoins du service Habitat : document rédigé par le service Habitat.**
- 2. Schéma de la base de donnée de Logem.**
- 3. Documents types fournis par le service Habitat.**
- 4. E-mails échangés dans le cadre du projet**
- 5. Documents relatifs à la CNIL.**
- 6. Plans de tests.**