Rapport de projet

Valentin Durand, Alexandre freret, pierre friboulet, jéremie le bas

Association Revivre

Table des matières

[La charte de projet 1](#_Toc388965064)

[Contexte 1](#_Toc388965065)

[Enjeux 1](#_Toc388965066)

[Objectifs 1](#_Toc388965067)

[Jalons du projet 1](#_Toc388965068)

[MCD 1](#_Toc388965069)

[Dictionnaire des données 1](#_Toc388965070)

[Diagramme de cas d’utilisation 2](#_Toc388965071)

[Répartition des tâches 2](#_Toc388965072)

[Découpage en lots 3](#_Toc388965073)

[Etude des langages 4](#_Toc388965074)

[Réalisation 4](#_Toc388965075)

[Schema Navigationnel d’interaction 4](#_Toc388965076)

[Quelques aperçus 6](#_Toc388965077)

[Exemple de code 8](#_Toc388965078)

[Avancement 1](#_Toc388965079)

# La charte de projet

## contexte

Le projet que nous devons réaliser est destiné à une association qui se nomme Revivre. C’est une association qui est chargée d’accueillir et d’orienter les personnes en difficultées sociales. L’association Revivre propose différentes activités pour réinsérer ses adhérents : chantiers en batiment, prestations diverses (nettoyage, conditionnement, assemblage). Elle propose aussi un système de location de véhicules et un restaurant.

Elle dispose actuellement d’une base de données sous Access, disponible via un Intranet qui est géré par une machine sous Windows Server R2. Cette base de données a été réalisée par un autodidacte (M. Gilles Leduc), l’organisation de la base de données n’est pas optimisée.

## enjeuX

Le principal enjeu de ce projet est d’accroitre la facilité d’utilisation de la base en permettant à d’autres membres de l’association de consulter et d’ajouter des informations très simplement. Il faudra aussi créer des relations cohérentes entre les tables ce qui entraine une totale restructuration logique de la base.

Nous devons aussi permettre à différents utilisateurs de se connecter en meme temps sur la base de données, il faut donc gérer le multisessions. Un autre enjeu est de convetir la base vers une version libre et gratuite.

## objectifs

Nous avons des objectifs précis à réaliser comme adapter la base et l’interface graphique pour l’utilisation d’une association. Pour détailler nous devons : gérer les chantiers avec la visualisation et l’impression des devis, calculer le coût des achats chez les fournisseurs, calculer les heures de travail. Nous devons gérer la facturation des locations des véhicules, facturer les repas du restaurant, suivre l’état des outillages et de leurs répartitions, l’attribution des clés des logements, visualiser toutes les données (encadrants, clients, salariés …).

## jalonS du projet

13/01/2014 : Choix du projet

06/02/2014 : Première prise de contact

24/02/2014 : Réalisation de la structure papier de la base de données

27/03/2014 : Présentation de la structure de gestion des chantiers

24/04/2014 : Présentation de la base au complet

05/06/2014 : Livraison de la partie chantier

26/02/2015 : Livraison au complet

05/03/2015 : Configuration et intégration de la base sur l’intranet

# C:\Users\Alexandre\Documents\GitHub\Revivre\MCD-Revivre.pngMCD

# Dictionnaire des données

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CHANTIER | | | | | | | | | |
| # | Données | 1 | 5 | 17 | 27 | 28 | 32 | 35 |  |
| 1 | NumEncardrant | \* | 1 |  |  |  |  |  |  |
| 2 | NomEncadrant | 1 |  |  |  |  |  |  |  |
| 3 | PrenomEncadrant | 1 |  |  |  |  |  |  |  |
| 4 | CategorieEncadrant | 1 |  |  |  |  |  |  |  |
| 5 | NumDevis |  | \* |  |  |  |  |  |  |
| 6 | Priorité |  | 1 |  |  |  |  |  |  |
| 7 | DateDebut |  | 1 |  |  |  |  |  |  |
| 8 | DateFinPrevue |  | 1 |  |  |  |  |  |  |
| 9 | Intitulé |  | 1 |  |  |  |  |  |  |
| 10 | Echeance |  | 1 |  |  |  |  |  |  |
| 11 | DateFinReel |  | 1 |  |  |  |  |  |  |
| 12 | (calculable) Montant |  | 1 |  |  |  |  |  |  |
| 13 | AchatsPrevus |  | 1 |  |  |  |  |  |  |
| 14 | HeuresPrevues |  | 1 |  |  |  |  |  |  |
| 15 | DateHeuresEffectuées |  | 1 |  |  |  |  |  |  |
| 16 | NbHeure |  | 1 |  |  |  |  |  |  |
| 17 | NumClient |  | 1 | \* |  |  |  |  |  |
| 18 | NomCLient |  |  | 1 |  |  |  |  |  |
| 19 | PrenomClient |  |  | 1 |  |  |  |  |  |
| 20 | TelFixeClient |  |  | 1 |  |  |  |  |  |
| 21 | TelPortClient |  |  | 1 |  |  |  |  |  |
| 22 | FaxClient |  |  | 1 |  |  |  |  |  |
| 23 | AdresseClient |  |  | 1 |  |  |  |  |  |
| 24 | EmailClient |  |  | 1 |  |  |  |  |  |
| 25 | CpClient |  |  | 1 |  |  |  |  |  |
| 26 | VilleClient |  |  | 1 |  |  |  |  |  |
| 27 | EtatDevis |  |  |  | \* |  |  |  |  |
| 28 | NumAchat |  |  |  |  | \* |  |  |  |
| 29 | DateAchat |  |  |  |  | 1 |  |  |  |
| 30 | TypeAchat |  |  |  |  | 1 |  |  |  |
| 31 | MontantAchat |  |  |  |  | 1 |  |  |  |
| 32 | NumControle |  |  |  |  |  | \* |  |  |
| 33 | DateControle |  |  |  |  |  | 1 |  |  |
| 34 | NatureControle |  |  |  |  |  | 1 |  |  |
| 35 | NumFournisseur |  |  |  |  |  |  | \* | 1 |
| 36 | Structure |  |  |  |  |  |  | 1 |  |
| 37 | NomFournisseur |  |  |  |  |  |  | 1 |  |
| 38 | PrenomFournisseur |  |  |  |  |  |  | 1 |  |
| 39 | NumProduit |  |  |  |  |  |  |  | \* |
| 40 | Conditionnement |  |  |  |  |  |  |  | 1 |
| 41 | Tarif |  |  |  |  |  |  |  | 1 |
| 42 | NomProduit |  |  |  |  |  |  |  | 1 |

# Diagramme de cas d’utilisation

Voir en annexe n°1.

Le diagramme de cas d’utilisation représente toutes les actions que pourront faire les différents utilisateurs de l’application que nous allons développer. Cela permet ainsi de définir des schémas d’utilisations et quels seront les différents profils utilisateurs à mettre en place.

# Répartition des tâches

La répartition des tâches s’est faîte de manière naturelle, chaque membre du groupe ayant des préférences et des compétences différentes. Cependant les tâches fondamentales comme la structure de la nouvelle base de données et l’établissement des cas d’utilisations ont été faites collectivement dans le souci que chaque membre du groupe ait toutes les informations essentielles à la réussite du projet.

Une fois ces tâches fondamentales effectuées, les tâches plus théoriques du projet comme réfléchir à l’apparence des interfaces utilisateurs (fenêtres IHM), la mise en page CSS du site, la mise en page de tous les documents (Diagramme de cas d’utilisations, dictionnaire de données, …) ainsi que la rédaction du rapport ont été effectué par Jérémie L. et Alexandre F. .

Les tâches de gestion de la base de données, de création des pages en HTML, d’implémentation du PHP et du JavaScript dans l’application ainsi que de la visualisation des données et de l’insertion de données ont été effectuées par Valentin D. et Pierre F. .

Pour pouvoir être à jour dans ce que chacun apportait au projet sans avoir besoin de faire de multiple transfert via clé USB nous avons décidé d’utiliser une plateforme de partage de fichiers, Git Hub, permettant à ce que chacun des membres du projet ait les mêmes fichiers continuellement actualisés grâce à des commandes « Commit ».

Nous avons aussi pris de multiples rendez-vous avec notre commanditaire ou avec nos tuteurs de projet pour pouvoir étudier conjointement les nouvelles tâches apportés au projet. En effet notre commanditaire a fait beaucoup évoluer les besoins de son application au cours du temps et nous avons dû régulièrement nous adapter à ces nouvelles demandes. Cela a aussi permis au projet d’avancer avec un aval de notre commanditaire, assurant que ce nous produisions lui convenait et que nous n’aurions pas à faire de gros changements de dernières minutes.

# Découpage en lots

|  |  |  |  |
| --- | --- | --- | --- |
| **N°** | **Story** | **Priorité** | **Itération** |
| 1 | L'utilisateur se connecte sur l'adresse de l'intranet | 1 | 1 |
| 2 | Au lancement l’utilisateur peut entrer ses identifiants | 2 | 1 |
| 3 | L'utilisateur peut visualiser les chantiers existants | 4 | 1 |
| 4 | L'utilisateur peut créer un nouveau chantier | 3 | 1 |
| 5 | L'utilisateur peut imprimer un devis | 7 | 1 |
| 6 | L'utilisateur peut attribuer des salariés au chantier | 5 | 1 |
| 7 | L'utilisateur peut visualiser les temps de travail | 8 | 1 |
| 8 | L'utilisateur ajoute des contrôles du chantier | 9 | 1 |
| 9 | L'utilisateur peut attribuer des encadrants au chantier | 6 | 1 |
| 10 | L'administrateur peut changer les droits des utilisateurs | 14 | 2 |
| 11 | L'administrateur peut créer des utilisateurs | 15 | 2 |
| 12 | L'utilisateur peut gérer un outil | 12 | 3 |
| 13 | L'utilisateur peut visualiser le niveau de carburant d'un véhicule | 13 | 3 |
| 14 | L'utilisateur peut gérer la location des véhicules | 10 | 4 |
| 15 | L'utilisateur peut attribuer un logement pour les personnes en réinsertion | 11 | 4 |
| 16 | L'utilisateur peut gérer les repas | 16 | 4 |

# 

# Etude des langages

Nous avons dû nous renseigner sur les différents langages disponibles pour réaliser une base de données et un site en intranet.

Concernant les choix du gestionnaire de base de données, nous avions le choix parmi un grand nombre de SGBD mais nous avons finalement choisi MySQL pour notre base de données.

Parmi les autres SGBD que nous aurions pu choisir il y a par exemple : Post Gre, Oracle et Access.

- Oracle et Access ont la particularité d’être payants or notre commanditaire nous a spécifié qu’il voulait une application faîte dans un langage gratuit pour ne pas avoir à payer de licenses, ce qui serait une grosse contrainte pour l’association qu’il représente.

- MySQL a été préféré à Post Gre pour sa meilleur intégration Web. De plus MySQL est un SGBD que nous avons étudié en cours donc plus intuitif pour nous et nous avons trouvé son panneau d’admnistration (PHPMyAdmin) très simple d’utlisiation et qu’il nous serait simple de faire un guide d’utilisation pour une personne non initée aux langagaes de base de données.

Pour la conception de l’interface, puisque notre application s’affichage dans un navigateur internet, nous avons choisis l’HTML avec une mise en page dynamique CSS. Ces choix sont presques obligatoires car ce sont les langages les plus répandus, les plus utilisés et les plus efficaces dans la conception de pages Web. L’HTML permet de définir la structure de la page tandis que le CSS permet de créer une mise en page dynamique, ce qui veut dire que chaque élément d’une même catégorie se voit attribuer la même mise en page sans avoir besoin de les éditer individuellement et ceci même après une modification du code HTML.

Nous avons aussi intégré le JavaScript dans notre application pour permettre des affichages dynamiques tels que des menus déroulants ou pour la réorganisation en temps réels des tables de données affichées (trie alphabétique, par type de donnée, …). De même, le JavaScript est à peu près le seul code à fournir ce type de fonctionnalités, donc comme pour le HTML et le CSS, notre choix a été très fortement limité.

De même pour la gestion de l’envoi et de la réception de données entre la page internet et le serveur de la base de données, peu de codes sont disponibles pour effectuer cette tâche. Nous avons donc dû utiliser le PHP pour permettre la récupération des données par la page internet et pour l’envoi des formulaires et des requêtes SQL au serveur.

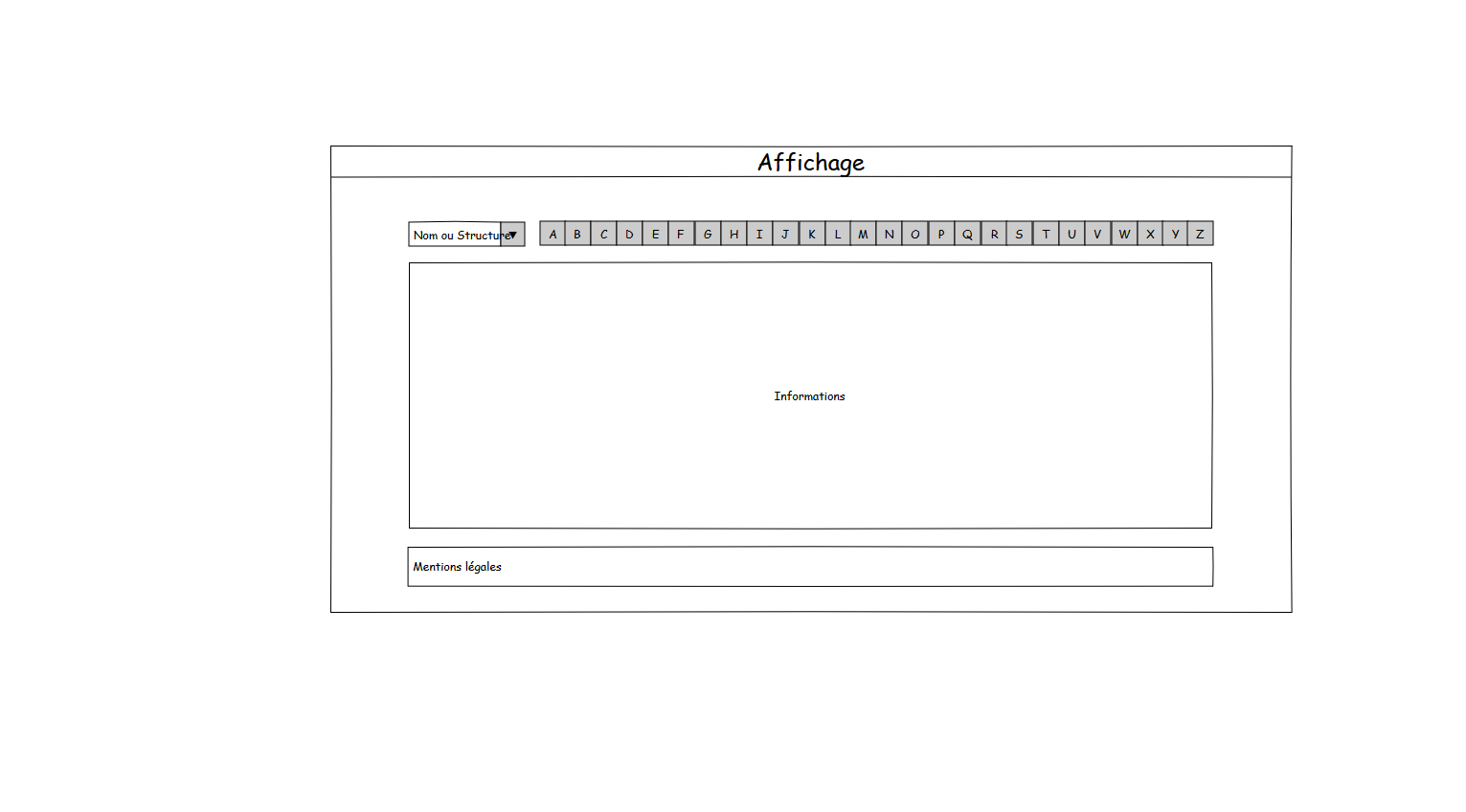
# Réalisation

## C:\Users\Alexandre\Documents\GitHub\Revivre\IHM\Accueil.pngSchema Navigationnel d’interaction

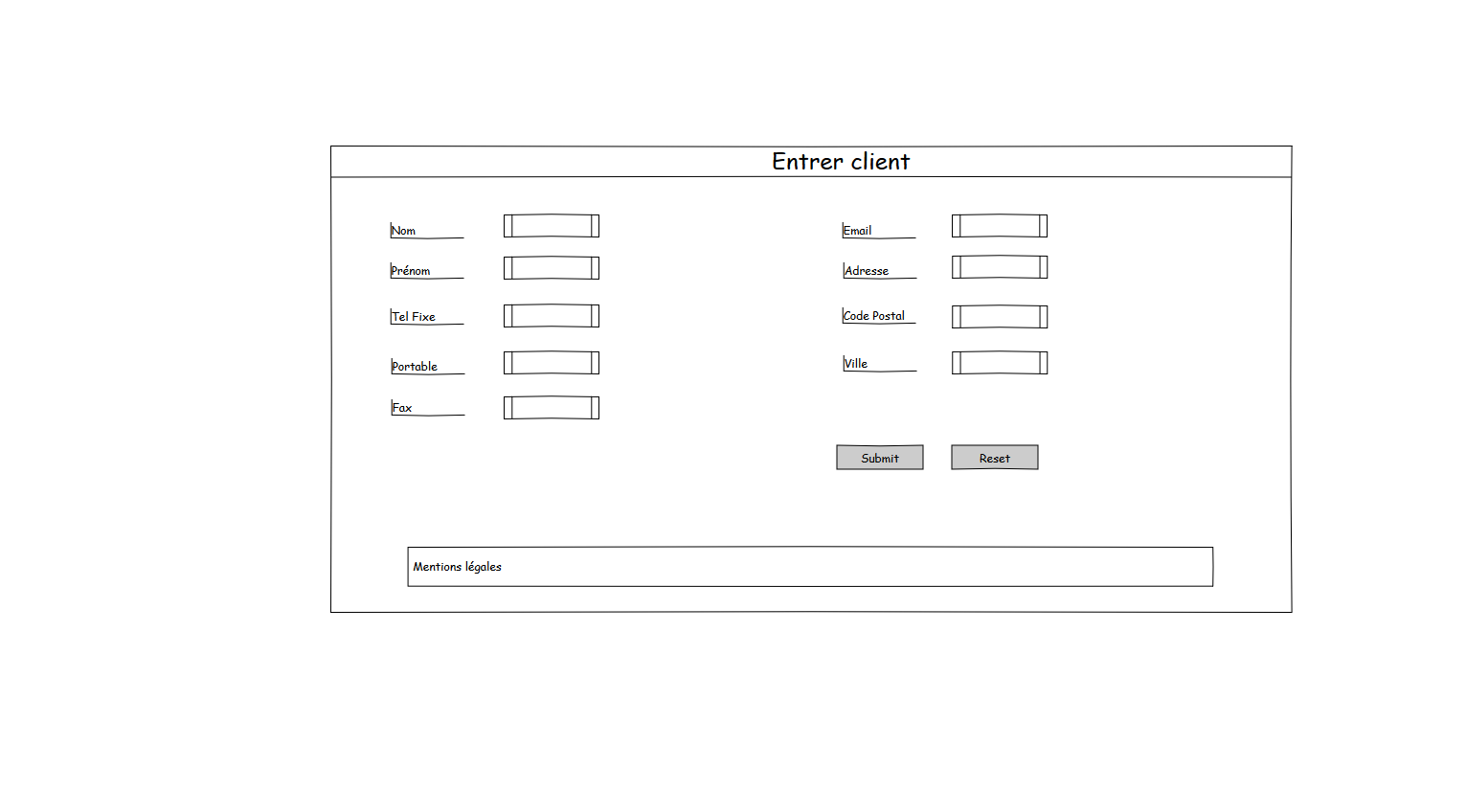
Ce schéma représente la page d’accueil avec :

* le logo de l’association « Revivre » en haut à gauche de format 225x139
* Un texte central présentant les fonctionnalités de l’intranet de manière brève
* Les différents menus déroulants en haut à gauche permettant de naviguer rapidement entre les différentes pages de l’intranet.

Notons que la disposition des menus déroulants et du logo de l’association sera commune à toutes les pages de l’intranet.

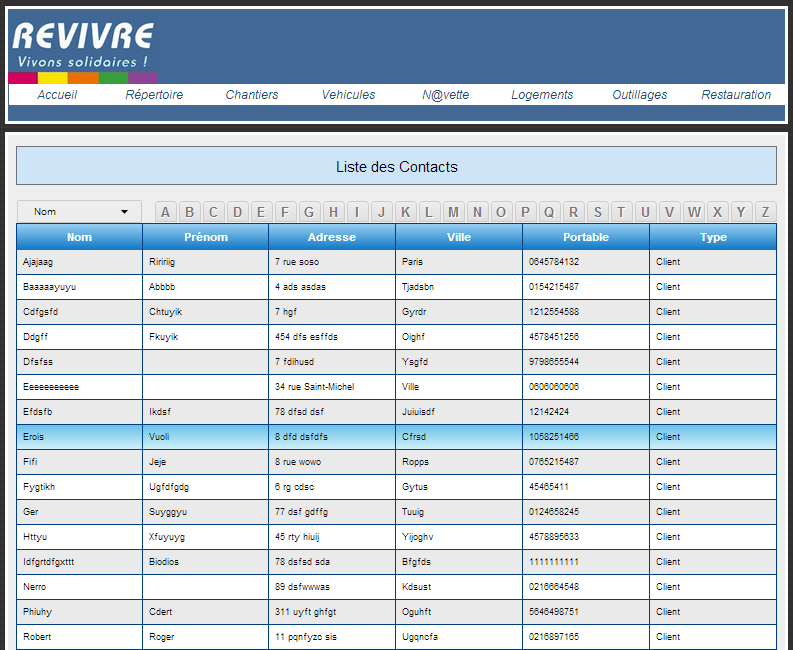
 Ce schéma représente l’affichage des données. Le menu déroulant en haut à gauche permet de trier les données affichées selon des critères précis tandis que les boutons de A à Z permettent de trier alphabétiquement.

L’application permet principalement de consulter les données, de ce fait toutes les pages d’affichages ressembleront à ce schéma quelles que soient les données affichées (chantiers, personnes, voitures, …).

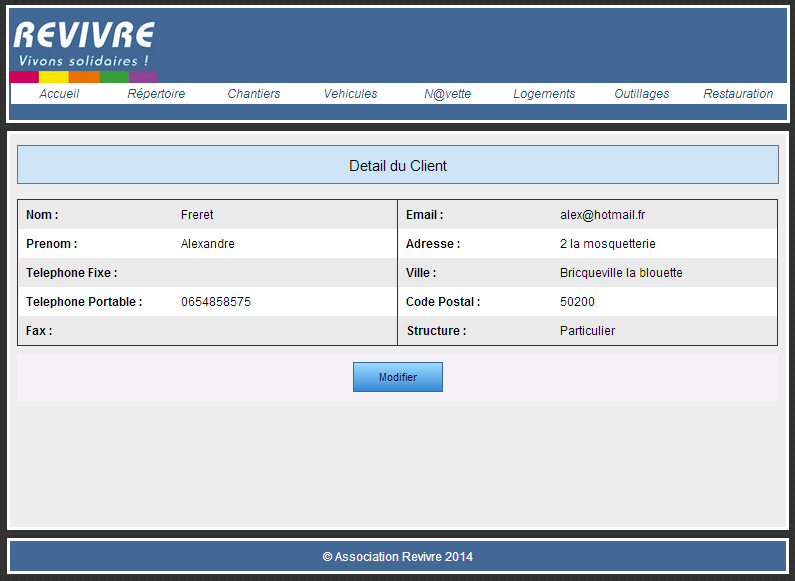


Ce dernier schéma représente l’insertion d’un client dans la base de données. On peut y voir tous les champs à compléter ainsi que le bouton d’envoi du formulaire (« Submit ») et le bouton de remise à zéro du formulaire (« Reset »).

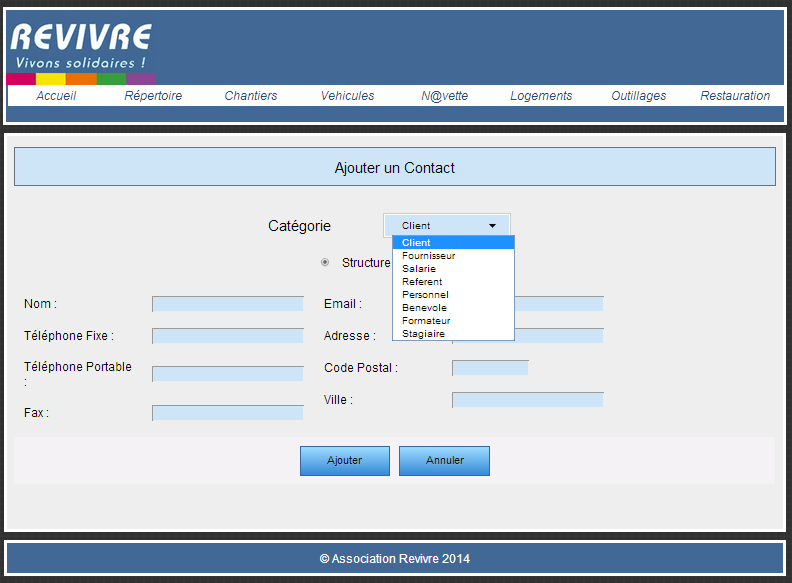
## quelques aperçus



Voici ce que l’on obtient si l’on veut consulter la liste des contacts. On peut remarquer le menu déroulants et les boutons de tris comme nous les avions vu dans le schéma d’affichage ainsi que de la barre de navigation tout en haut.



Un simple clique sur un contact de la liste précédente permet d’afficher les détails de la personne sélectionnées de manière suivante.

 La page d’ajout de contact a évolué mais reste la même dans sa conception. Elle a été optimisée par rapport à la conception originale pour ne pas être capable d’insérer uniquement des clients mais de pouvoir insérer n’importe quel type de personne.

## exemple de code

# Avancement

Nous nous devons de préciser que les demandes ont beaucoup évoluées depuis la première rencontre avec notre commanditaire, ainsi nous ne pouvons garantir une interface jusqu’à ce que celle-ci soit rendue car nous devons intégrer les nouvelles attentes à cette interface au fur et à mesure qu’elles se présentent.

Pour ne pas nous engager dans un développement de demandes inconnues, nous avons prévu de rendre le 12 juin une partie de l’application qui ne sera pas modifiée par le commanditaire et qui a été bien définie : la consultation des chantiers, la gestion des chantiers et l’impression des devis liés à ces chantiers. Ceci représente une grande partie de l’application finale et nous nous sommes engagés à l’installer le 12 juin sur le serveur de l’association. De plus la gestion des chantiers nécessite que la gestion des personnes soit opérationnelle car ceux-ci sont les acteurs des chantiers.

Pour 2014/2015, nous devrons finir l’application, ce qui veut dire finir la gestion des outils, des repas, des logements, de la location de voitures, du système « N@vette » et des autres ajouts potentiels que notre commanditaire pourrait ajouter.