



Institut Supérieur
d'Informatique de
Modélisation et de
leurs Applications

Complexe des Cézeaux
BP 125
63173 Aubière Cedex



Institut National de
Recherche Agronomique

58 rue Montalembert
63009 Clermont-Ferrand Cedex 1

Rapport de stage de 3^{ème} année
Filière Génie Logiciel et Systèmes Informatiques

Conception et réalisation d'un système de gestion centralisée des données issues de projets de recherche clinique en nutrition humaine

TOME I

Présenté par : **Romain Guidoux**
Responsable ISIMA : **Philippe Lacomme**
Responsable INRA : **Sylvie Rousset**

4 mois
01 septembre 2011

Remerciements

Je remercie Sylvie Rousset, ma maître de stage, pour son aide et sa disponibilité tout au long du stage. J'ai ainsi pu lui poser de nombreuses questions, et ses réponses m'ont beaucoup aidé dans la compréhension de ce milieu qu'est le laboratoire de nutrition humaine.

Je tiens aussi à remercier Philippe Lacomme, mon responsable ISIMA, qui a suivi mon travail tout au long du stage et a été lui aussi très disponible. Les points réguliers que nous avons faits m'ont permis d'améliorer ma façon de travailler et de voir les choses.

Enfin, je remercie Libo Ren, doctorant à l'ISIMA, pour l'aide qu'il m'a apportée, notamment pour PHP et jQuery.

Résumé

L'unité de nutrition humaine mène des projets de **recherche clinique** afin d'étudier l'impact des comportements alimentaires sur le vieillissement et les maladies chroniques, et de proposer des stratégies nutritionnelles pour retarder l'apparition des maladies.

Mon travail a consisté à réaliser une **application Internet** qui permette aux chercheurs de centraliser les **données** des différents projets, et qui offre une forte interactivité avec les volontaires qui participent aux protocoles. Elle doit notamment permettre la création et l'envoi de **questionnaires** aux volontaires, et leur proposer un agenda pour leurs futurs rendez-vous avec les chercheurs.

Une première version du cahier des charges avait été rédigée par deux étudiants en projet à l'ISIMA. J'ai donc commencé par le compléter et le modifier en allant rencontrer plusieurs personnes susceptibles d'être intéressées par ce projet. J'ai ensuite étudié les différents choix technologiques qui s'offraient à moi. J'ai finalement choisi le langage **PHP** et les frameworks CodeIgniter (PHP) et jQuery (**JavaScript**).

Au moment où je rédige ce rapport, une première version de la partie gestion des questionnaires est opérationnelle et est en cours de test auprès des utilisateurs.

Mots clés : recherche clinique, application Internet, données, questionnaires, PHP, JavaScript.

Abstract

The Human Nutrition Unit leads **clinical research** projects in order to study the impact of dietary behaviour on ageing and chronic diseases, and to propose nutritional strategies to delay the diseases occurrence.

My work consisted in developing a **web application** which aims to centralize **data** from many projects. It also allows a strong interactivity between with the protocols' volunteers and researchers. It has, among others, to enable **questionnaires'** creation and sending to volunteers, and to make a schedule available for their future appointments with researchers.

A first version of the requirements had already been written by two students from ISIMA, so I started off with completing and modifying it by meeting several persons who could be interested in this project. Then I made technological choices, and finally chose the **PHP** programming language and two frameworks: CodeIgniter (PHP) and jQuery (**JavaScript**).

At the moment I have implemented a first version of the questionnaires which is currently being evaluated by the future users.

Key words: clinical research, web application, data, questionnaires, PHP, JavaScript.

Table des illustrations

Figure 1 : Logos de l'UNH et du CRNH d'Auvergne	2
Figure 2 : Composition du CRNH Auvergne	3
Figure 3 : Chambre calorimétrique du LNH	4
Figure 4 : Capteurs ActiHeart et Armband	4
Figure 5 : Schéma du système ActivCollector	5
Figure 6 : Parcours d'un volontaire	6
Figure 7 : Fonctionnalités du site Internet d'ActivCollector.....	7
Figure 8 : Modules du site Internet	10
Figure 9 : Exemple de planning pour les chambres calorimétriques	12
Figure 10 : Capteur [Actiheart]	14
Figure 11 : Exemple de questionnaire	15
Figure 12 : Exemple de journal de bord	16
Figure 13 : Exemple de questions différentes selon le sexe du volontaire, extraites d'un questionnaire papier	18
Figure 14 : Questions liées, extraites d'un questionnaire papier	18
Figure 15 : Représentation des questionnaires	20
Figure 16 : Format du fichier Excel de dépouillement.....	21
Figure 17 : Question portant sur les loisirs	21
Figure 18 : Questions liées à la question sur les loisirs.....	22
Figure 19 : Exemple de question liée à la fréquence de consommation d'un aliment.....	23
Figure 20 : Exemple de question liée à la quantité d'absorption d'un aliment	23
Figure 21 : Premier onglet de la feuille d'analyses.....	24
Figure 22 : Deuxième onglet de la feuille d'analyses	24
Figure 23 : Exemple de somme des rangs	25
Figure 24 : Principe de demande d'une page sur Internet	26
Figure 25 : Exemple de code HTML	26

Figure 26 : Comparaison des langages côté serveur	27
Figure 27 : Exemple de requête utilisant DQL	29
Figure 28 : Fonctionnement de Doctrine.....	29
Figure 29 : Architecture MVC	31
Figure 30 : Avantages et inconvénients des frameworks PHP.....	32
Figure 31 : Fonctionnement de [CodeIgniter 2].....	33
Figure 32 : Inclusion de fichiers dans le constructeur de la bibliothèque Doctrine	34
Figure 33 : Diagramme de classes de l'exemple	35
Figure 34 : Scénario d'utilisation du mode lazy-loading.....	35
Figure 35 : Scénario d'utilisation du mode eager-loading.....	36
Figure 36 : Correspondance entre éléments HTML et DOM.....	37
Figure 37 : Inclusion conditionnelle d'un script JavaScript	37
Figure 38 : Compatibilités des frameworks entre navigateurs	38
Figure 39 : Avantages et inconvénients des différents frameworks JavaScript.....	38
Figure 40 : Exemple d'utilisation de Datepicker, un des plugins de jQuery UI.....	39
Figure 41 : Syntaxe de la sélection d'éléments avec jQuery	39
Figure 42 : Code HTML d'exemple	40
Figure 43 : Exemples d'utilisation de jQuery	40
Figure 44 : Exemple de 2 tâches synchrones	41
Figure 45 : Exemple de 2 tâches asynchrones.....	41
Figure 46 : Exemple de données XML	42
Figure 47 : Fonctionnement d'AJAX	42
Figure 48 : Vue récapitulative des technologies utilisées	43
Figure 49 : Structure de base des pages du site.....	47
Figure 50 : Génération d'un vue	48
Figure 51 : Définition des constantes pour les menus.....	48
Figure 52 : Génération de la page	48

Figure 53 : Question ouverte extraite d'un questionnaire papier.....	49
Figure 54 : Eléments utilisés pour la saisie de texte	49
Figure 55 : Echelle à libellés extrémaux extraite d'un questionnaire papier.....	50
Figure 56 : Echelle avec un libellé par choix extraite d'un questionnaire papier.....	50
Figure 57 : Echelle de valeurs extraite d'un questionnaire papier.....	50
Figure 58 : Représentation d'un choix unique parmi N possibilités.....	51
Figure 59 : Exemple de question à P choix parmi N.....	51
Figure 60 : Représentation d'une question à P réponses parmi N possibilités.....	51
Figure 61 : Echelle continue extraite d'un questionnaire papier	52
Figure 62 : Représentation d'une échelle continue.....	52
Figure 63 : Question de tri extraite d'un questionnaire papier	53
Figure 64 : Représentation d'une question de classement, dont l'élément "Poulet" est en cours de déplacement.....	53
Figure 65 : Cas particulier d'un test de classement, extrait d'un questionnaire papier.....	54
Figure 66 : Première question du cas particulier : le choix.....	54
Figure 67 : Seconde question du cas particulier : le classement	54
Figure 68 : Exemple de question tabulaire, extraite d'un questionnaire papier.....	55
Figure 69 : Question tabulaire, question 1	56
Figure 70 : Question tabulaire, question 2	56
Figure 71 : Question tabulaire, question 3	56
Figure 72 : Question tabulaire, question 4	56
Figure 73 : Question tabulaire, question 5	57
Figure 74 : Question tabulaire, question 6	57
Figure 75 : Question tabulaire, question 7	57
Figure 76 : Question tabulaire sans proposition d'éléments, extrait d'un questionnaire papier	58
Figure 77 : Question associée à une image	59
Figure 78 : Diagramme de classes UML des questionnaires	63

Figure 79 : MLD de la partie de la base de données qui concerne les questionnaires	64
Figure 80 : Exemples d'internationalisation de textes	65
Figure 81 : Mapping d'une table.....	66
Figure 82 : Mapping d'une colonne.....	66
Figure 83 : Différentes sortes de mapping One-To-One.....	67
Figure 84 : Différentes sortes de mapping One-To-Many	67
Figure 85 : Mapping Many-To-One.....	68
Figure 86 : Différentes sortes de mapping Many-To-Many	68
Figure 87 : Annotation pour une relation One-To-One.....	68
Figure 88 : Interface de gestion des questionnaires par le chef de projet	69
Figure 89 : Génération d'un questionnaire au format PDF.....	70
Figure 90 : Interface d'envoi d'un questionnaire	71
Figure 91 : Interface de création d'un questionnaire, partie renseignements généraux.....	72
Figure 92 : Interface de création d'une nouvelle question.....	73
Figure 93 : Gestion des questions	74
Figure 94 : Affichage d'une question à un volontaire	75
Figure 95 : Formulaire de saisie d'une nouvelle remarque.....	76
Figure 96 : Gestion des comptes dans la version de démonstration.....	76
Figure 97 : Echanges avec la juriste de Toulouse	77
Figure 98 : Flux de données sur le site d'ActivCollector.....	78
Figure 99 : Cryptage de l'adresse email	79
Figure 100 : Décryptage d'une donnée cryptée	79
Figure 101 : Position de la clé sur le serveur	80
Figure 102 : Chiffrement d'un mot de passe	80
Figure 103 : Financements des organismes publics de recherche.....	82
Figure 104 : Projet à 5 ans avec Almerys.....	84
Figure 105 : Objectifs du projet à 5 ans	85

Figure 106 : Tâches du projet à 1 an et répartition entre les partenaires.....	87
Figure 107 : Planification des tâches durant la première année	88
Figure 108 : Communications dans le cadre de la rédaction du dossier	89
Figure 109 : Avancement de mon travail	90
Figure 110 : Chronologie de mon travail	91

Table des matières du tome I

Remerciements

Résumé

Table des illustrations

Table des matières du tome I

Table des matières du tome II

Lexique

Introduction	1
1. Contexte général et présentation d'ActivCollector	2
1.1. Présentation de l'entreprise	2
1.2. Contexte.....	3
1.3. Le site Internet d'ActivCollector	6
1.3.1. Objectifs et utilisateurs	6
1.3.2. Les modules.....	9
1.3.3. Etude du module « questionnaires »	16
2. Choix techniques et implémentation.....	26
2.1. Principes de base	26
2.2. Choix d'un langage de programmation	27
2.3. Choix d'un outil de mapping objet-relationnel.....	28
2.4. Etude des frameworks PHP	30
2.5. Présentation de CodeIgniter.....	33
2.5.1. Principe général.....	33
2.5.2. Intégration de Doctrine 2 dans CodeIgniter	34
2.6. Etude des frameworks JavaScript.....	36
2.6.1. Présentation de JavaScript.....	36
2.6.2. Choix d'un framework	37
2.6.3. Présentation de jQuery	39
2.7. Récapitulatif des technologies utilisées	43

2.8.	Outils utilisés	44
2.8.1.	Wamp Server.....	44
2.8.2.	Eclipse et le plugin Aptana.....	44
2.8.3.	BitKinex	45
2.8.4.	MySQL Workbench	46
3.	Implémentation et aspects juridiques	47
3.1.	Création d'une bibliothèque dans CodeIgniter	47
3.2.	Informatisation des questionnaires	49
3.2.1.	Les différents types de questions rencontrés	49
3.2.2.	Un outil déjà existant : LimeSurvey.....	59
3.2.3.	Modélisation de la base de données	60
3.2.4.	Mapping objet-relationnel avec Doctrine.....	65
3.3.	Interfaces de gestion des questionnaires.....	69
3.4.	Déploiement d'une version 1 de démonstration	75
3.5.	Aspects juridiques.....	77
4.	Le projet mobilité.....	81
4.1.	Naissance du projet.....	81
4.2.	Rencontre avec Almerys.....	83
4.3.	Le dossier de demande de bourse	86
4.4.	Discussion.....	89
	Conclusion.....	93

Webographie

Références bibliographiques

Table des matières du tome II

Table des matières du tome I

Table des matières du tome II

1. Annexe I : Transparents pour l’animation scientifique du 14 juin	1
2. Annexe II : Dossier CNIL destiné à la juriste INRA de Toulouse	9
3. Annexe III : Transparents pour la réunion chez Almerys	22
4. Annexe IV : Fiche PDF envoyée par mail aux testeurs du site.....	29

Lexique

Lexique juridique et licences de logiciels

- Apache :** Lorsque le logiciel est modifié, le copyright et le texte de la licence doivent être conservés, mais un nouveau copyright peut néanmoins être ajouté. Chaque fichier modifié doit clairement faire apparaître qu'il a été modifié, et les modifications peuvent hériter de la licence Apache ou être régies par une nouvelle licence. Si aucune licence n'est précisée, c'est la licence Apache qui est adoptée.
- BSD :** Permet de réutiliser tout ou une partie du logiciel sans restriction, qu'il soit intégré dans un logiciel libre ou propriétaire. Le texte de la licence peut lui-même être modifié.
- GPL :** Cette licence met en œuvre la notion de copyleft, ce qui oblige les développeurs du logiciel à le rendre libre d'utilisation, de modification et de diffusion. D'autre part, toutes les versions ultérieures du logiciel seront soumises à cette licence. C'est ce qu'on appelle le caractère héréditaire de la GPL.
- LGPL :** Un programme sous licence LGPL peut être lié à du code non LGPL, sans pour autant révoquer la licence. La différence par rapport à la licence GPL est que la LGPL permet s'affranchir de l'hérédité de la licence GPL.
- MIT :** Donne droit à toute personne d'utiliser, de copier, de modifier, de publier, de distribuer, de vendre et de changer la licence du logiciel. La seule obligation est de garder le nom des auteurs ainsi que la notice de copyright.

Lexique informatique

- API :** Signifie « **A**pplication **P**rogramming **I**nterface », ou « Interface de programmation » en français. Une API est un ensemble de classes qui permet l'interopérabilité de plusieurs composants logiciels.
- Cache :** Le cache est un mécanisme qui consiste à enregistrer temporairement sur le disque dur des données provenant d'un site Internet de manière à pouvoir y accéder plus rapidement les fois suivantes.
- Cookie :** Un cookie est un petit fichier créé par un site Internet sur l'ordinateur du client. Il peut par exemple contenir des informations telles que la date à laquelle s'est connecté le visiteur sur le site pour la dernière fois.
- Framework :** Un framework (ou cadriceiel en français) est un ensemble de composants logiciels servant à créer les bases structurelles de nouvelles applications.
- Mapping :** Le mapping est une opération qui consiste à mettre en correspondance des données issues de plusieurs modèles de données différents.
- MLD :** Le MLD (pour « **M**odèle **L**ogique des **D**onnées »), est un modèle utilisé dans le cadre de la méthode d'analyse Merise. Il permet de faire apparaître la structure et l'organisation des données telles qu'elles seront implémentées.
- Plugin :** Un plugin est un module d'extension qui permet d'ajouter des fonctionnalités à un logiciel.
- UML :** UML (pour « **U**nified **M**odeling **L**anguage », ou « langage de modélisation unifié » en français), est un langage qui permet de modéliser graphiquement des applications dans le cadre de la programmation orientée objets.
- URL :** Une URL (pour « **U**niform **R**esource **L**ocator », ou « localisateur uniforme de ressources » en français) désigne une adresse Internet, sous la forme d'une chaîne de caractères. Exemple : <http://www.isima.fr>
- Tuple :** Un tuple est une liste ordonnée d'attributs relatifs à un même élément. Une table d'une base de données contient un ensemble de tuples.

Introduction

J'ai effectué mon stage de troisième année à l'ISIMA au Laboratoire de Nutrition Humaine (LNH) dans une équipe de l'Unité de Nutrition Humaine de l'INRA de Clermont-Ferrand, pour une durée de quatre mois. Ma maître de stage à l'INRA était Sylvie Rousset et mon tuteur ISIMA était Philippe Lacomme.

Le but de mon stage était de créer un site Internet qui aide les chercheurs en leur permettant d'informatiser certaines tâches, mais aussi et surtout de centraliser les données de leurs différents projets. En effet pour l'instant tout est au format papier, ce qui pose des problèmes quant à l'archivage de toutes ces données. De plus, le site doit permettre une certaine interaction avec les volontaires qui participent aux protocoles, afin de les intéresser et de les motiver du mieux possible.

J'ai tout d'abord commencé par faire une analyse des besoins en allant rencontrer les potentiels futurs utilisateurs et en me basant sur ce que deux étudiants de l'ISIMA avaient commencé à faire durant leur projet.

J'ai ensuite commencé à modéliser la base de données en essayant de trouver des solutions aux problèmes qui se sont posés à moi. En parallèle, j'ai aussi étudié les différentes technologies disponibles afin de déterminer quel langage de programmation et quels frameworks utiliser.

Enfin, j'ai commencé l'implémentation d'une des parties les plus longues à réaliser techniquement : les questionnaires. Je n'ai pas eu le temps de traiter les autres parties car je me suis concentré sur celle-ci afin de pouvoir proposer une première version de test aux utilisateurs. J'ai pour cela dû faire une demande auprès des informaticiens du centre de Theix, afin d'obtenir un accès à un de leurs serveurs.

Je commencerai tout d'abord par présenter le contexte général et les besoins des utilisateurs, puis je justifierai les choix techniques que j'ai faits et j'expliquerai succinctement leur fonctionnement. Enfin, j'aborderai la façon dont j'ai implémenté le site et les questions juridiques qui se sont posées à moi lors de la phase d'analyse des besoins.

1. Contexte général et présentation d'ActivCollector

1.1. Présentation de l'entreprise

J'ai effectué mon stage de troisième année à l'ISIMA, d'une durée de quatre mois, dans l'équipe Contrôle de l'Homéostasie Lipido-Energétique et Obésité (CHLEO) de l'INRA-UNH, au Laboratoire de Nutrition Humaine (LNH) à Clermont-Ferrand.

L'équipe CHLEO est une des équipes de l'Unité de Nutrition Humaine (UNH) de l'Institut National de la Recherche Agronomique (INRA). D'autres équipes travaillant sur la nutrition humaine sont installées à Theix (figure 1).

D'autre part, l'INRA est l'un des six organismes que regroupe le Centre de Recherche en Nutrition Humaine (CRNH) d'Auvergne (figure 2). Ce dernier est en effet un groupement d'intérêt public. Créé en 1992 par le ministère de la Recherche, il rassemble :

- Le Centre Hospitalier Universitaire de Clermont-Ferrand (CHU).
- L'Institut National de la Santé Et de la Recherche Médicale (INSERM).
- Le centre Jean Perrin.
- L'INRA.
- L'université d'Auvergne Clermont 1.
- L'université Blaise Pascal Clermont 2.



Figure 1 : Logos de l'UNH et du CRNH d'Auvergne

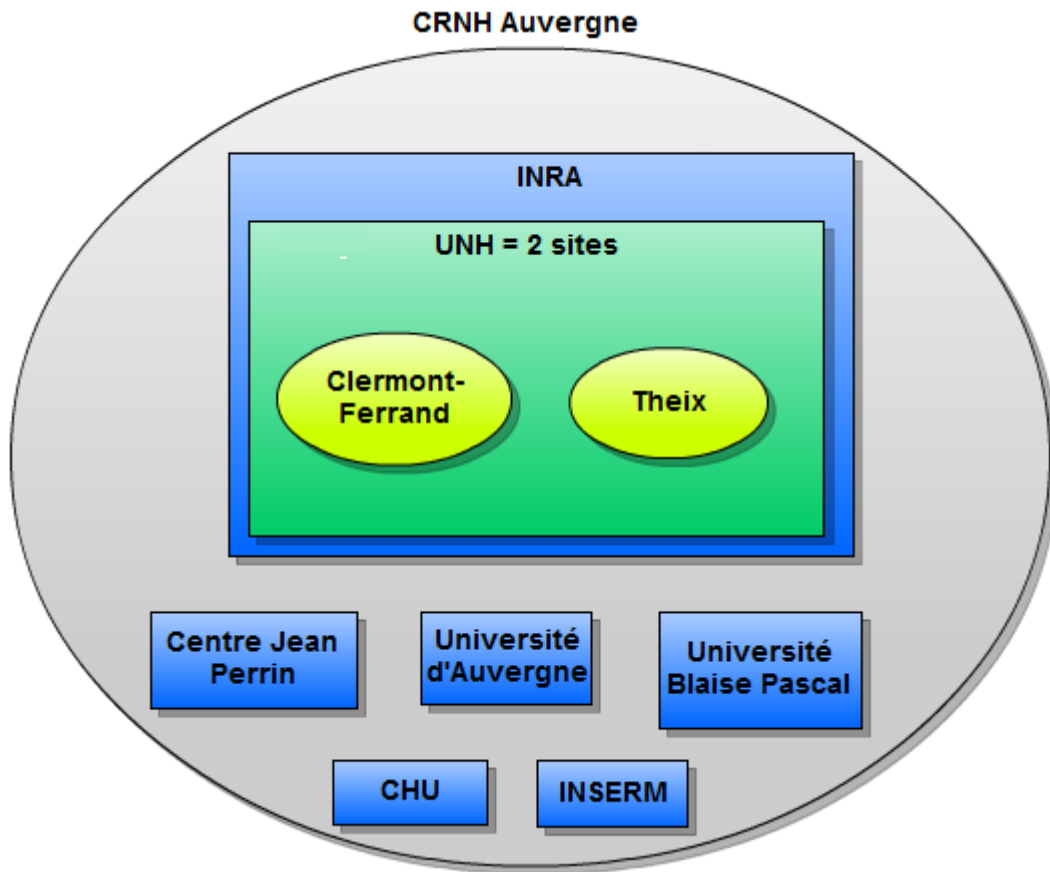


Figure 2 : Composition du CRNH Auvergne

1.2. Contexte

L'équipe CHLEO a travaillé en 2008 sur un projet nommé Modelheart qui consistait à prédire la dépense énergétique de volontaires humains à partir de leur fréquence cardiaque, l'intensité de leurs mouvements, leur température corporelle, etc. Deux groupes distincts de volontaires ont participé au projet. Ceux du premier groupe venaient au laboratoire pour passer 24 heures en chambre calorimétrique, ce qui permet de mesurer la dépense énergétique à partir des échanges respiratoires des volontaires (O_2 consommé et CO_2 produit, figure 3).

C'est l'estimation faite par cette chambre qui a servi de référence aux chercheurs. D'autre part, les volontaires étaient aussi équipés de capteurs Actiheart et Armband (figure 4) pour déterminer si le sujet est endormi ou éveillé, debout ou allongé, et mesurer l'intensité de ses mouvements et de sa fréquence cardiaque. Ils étaient aussi équipés d'un électrocardiogramme relié à une alarme au cas où le volontaire ferait un malaise pendant son séjour en chambre calorimétrique.



Figure 3 : Chambre calorimétrique du LNH

Le second groupe portait lui aussi les capteurs Actiheart et Armband et évoluait dans les conditions normales de vie, c'est-à-dire à domicile. Les capteurs servent entre autres à mesurer la dépense énergétique, la fréquence cardiaque et l'accélérométrie. D'autre part, les volontaires de ce groupe devaient ingérer de l'eau doublement marquée afin d'évaluer leur dépense énergétique globale.

Les capteurs de chaque volontaire ont enregistré des données minute par minute. Ce projet a donc permis de recueillir 32 millions de données, soit une quantité d'informations difficilement traitable sans outil informatique.



Figure 4 : Capteurs ActiHeart et Armband

L'INRA-UNH et l'ISIMA se sont alors associés pour créer un logiciel nommé Finder2E (Finder Of Energy Expenditure), capable de traiter les informations récoltées et de tester des fonctions mathématiques de prédiction de la dépense énergétique. Il a été initié en 2009 par des étudiants de l'ISIMA, puis d'autres étudiants de l'université Blaise Pascal se sont succédés pour déboguer le logiciel et lui ajouter de nouvelles fonctionnalités.

J'ai été amené à travailler sur ce logiciel durant les deux premières semaines de stage afin de résoudre des problèmes de mémoire. J'ai ainsi modifié quelques algorithmes afin de libérer de la mémoire plus régulièrement, et j'ai modifié la façon de générer des graphes dans le logiciel. Le temps de génération d'un graphe est ainsi passé de plus de 10 minutes à une dizaine de

secondes. Un étudiant de licence L3, Andrianaina Ramilijaona, a ensuite pris le relais sur ce logiciel, de fin mai à début août.

D'autre part, la plupart des documents liés à des projets sont conservés au format papier, que ce soient des résultats de composition corporelle déterminés par DEXA (« **D**ual **E**nergy **X**-ray **A**bsorptiometry », ou « absorption bi-photonique à rayons X »), qui permet entre autres d'évaluer la quantité de masse grasse d'un individu, des questionnaires envoyés aux volontaires ou des cahiers d'observations tenus par les **A**ssistants de **R**echerche **C**linique (ARC).

Sylvie Rousset et Philippe Lacomme ont alors eu l'idée de créer un site Internet qui servirait à dématérialiser un maximum de données et surtout à les conserver et à les consulter des années plus tard. En effet, au bout d'un certain temps, les chercheurs sont obligés de faire le tri dans leurs archives, faute de place. Ce sont donc les données de projets entiers qui sont perdues. Le site Internet et le logiciel Finder2E forment un système que nous avons appelé **ActivCollector** (figure 5).

Durant mon stage, j'ai manifesté mon intérêt pour continuer à travailler à l'INRA. Les événements se sont alors ensuite succédés de sorte que j'ai pu assister à la naissance d'un projet de collaboration avec une entreprise. Ce projet consiste à réaliser une brique « mobilité » dans le système **ActivCollector**, qui permettra aux volontaires d'y accéder via leurs smartphones.

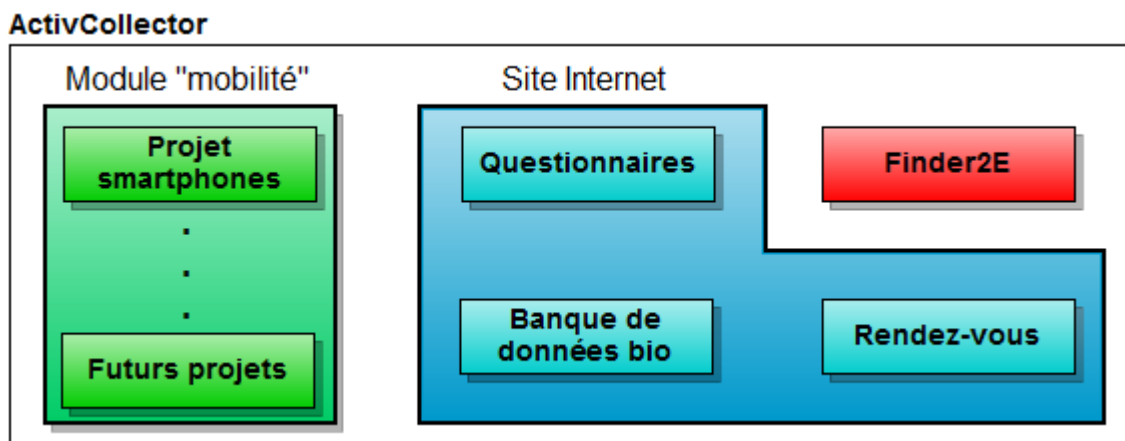


Figure 5 : Schéma du système **ActivCollector**

De futurs projets viendront sûrement se greffer à la partie mobilité dans les années à venir.

1.3. Le site Internet d'ActivCollector

1.3.1. Objectifs et utilisateurs

Quelques mois avant que je ne commence mon stage, deux étudiants de l'ISIMA ont commencé à rédiger le cahier des charges du site Internet durant leur projet de troisième année [Kamèche & Roux, 2011].

Les objectifs de ce site sont de regrouper et d'informatiser les données liées aux projets de recherche clinique (c'est-à-dire sur volontaires humains) afin qu'elles puissent être consultées ultérieurement, sans limite de temps. De plus, le site doit permettre une plus grande interactivité entre chercheurs d'une part, et entre chercheurs et volontaires d'autre part. Cela se fera notamment par des rappels envoyés par email, comme nous le verrons plus loin. Enfin, nous avons aussi voulu constituer une banque de données biologiques issues des examens médicaux que passent les volontaires au cours des protocoles de recherche clinique.

Lorsqu'un volontaire se présente pour participer à une étude, il passe une visite médicale de pré inclusion. Elle a pour but de déterminer si le volontaire répond à tous les critères biologiques et sociaux de participation au protocole. S'il passe la visite avec succès, le volontaire est dit « inclus » (figure 6).

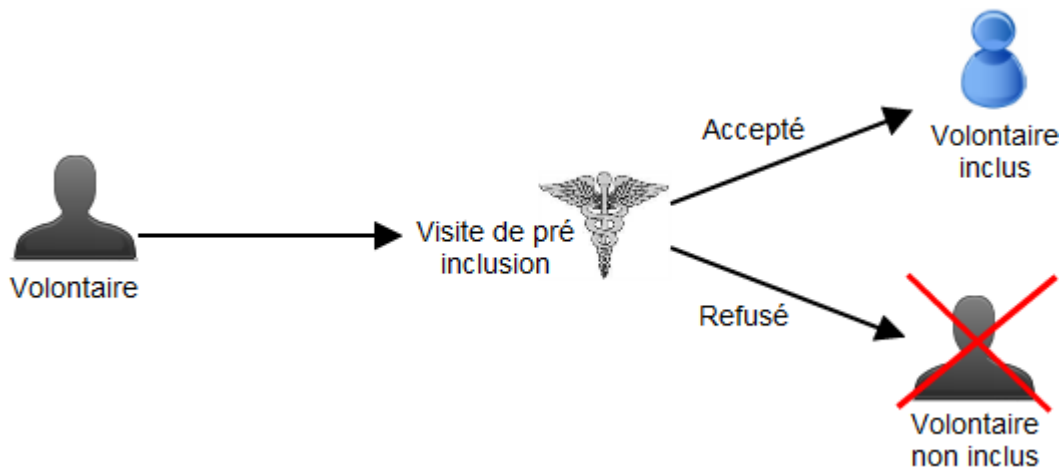


Figure 6 : Parcours d'un volontaire

Afin de bien cerner les besoins et de prendre en compte leur évolution j'ai donc, avec l'aide de Sylvie, rencontré différentes personnes susceptibles d'être intéressées par le projet.

J'ai ainsi pu cerner plusieurs profils d'utilisateurs et les actions qu'ils pourront réaliser. Tous, sauf les visiteurs, auront accès à un espace personnel sur le site.

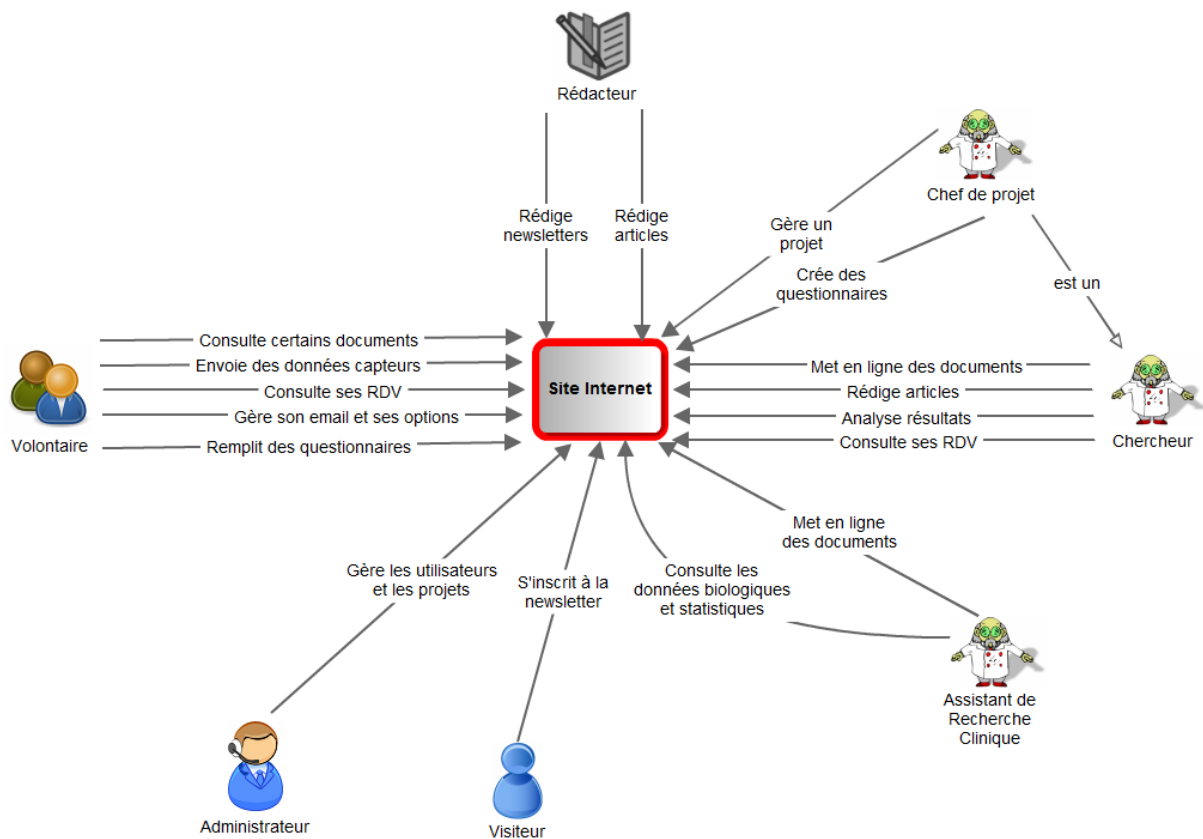


Figure 7 : Fonctionnalités du site Internet d'ActivCollector

Les types d'utilisateurs sont :

- Les visiteurs : nous avons convenu de mettre sur le site quelques pages accessibles par tous qui présentent le Centre de Recherche en Nutrition Humaine (CRNH). Lors de notre entretien avec Martine Duclos, nous avons décidé de mettre en avant les annonces de recherche de volontaires sur la page d'accueil. En effet, il faut que les volontaires potentiels aient accès aux informations les plus importantes dès qu'ils arrivent sur le site (figure 7).

D'autre part, chaque internaute pourra s'inscrire à une newsletter afin de suivre les actualités du CRNH et de l'INRA et/ou les annonces de recherche de volontaires en cours.

- Les volontaires : nous avons estimé que seuls les volontaires inclus seront pris en compte dans le site Internet. En effet, il n'est pas intéressant de garder les volontaires non inclus dans la base de données car pour chaque nouveau protocole, les volontaires doivent passer une nouvelle visite de pré inclusion. D'autre part, les assistants de recherche clinique (ARC) tiennent à jour des listes et prennent en compte d'autres critères pour contacter les gens, tels que le montant total des indemnités perçues par le

volontaire durant l'année en cours, ou encore la durée d'exclusion à respecter à la fin d'un protocole.

- Les rédacteurs : ce sont des employés de l'INRA qui seront chargés de mettre à jour la page d'accueil, par exemple en publiant des news ou en gérant les annonces de recherche de volontaires.
- Les chercheurs : ils devront pouvoir accéder aux données récoltées dans le cadre du projet auquel ils sont affectés, gérer leurs rendez-vous avec les volontaires, etc.
- Les assistants de recherche clinique : ils pourront envoyer sur le site des documents relatifs à chaque projet, tels que le dossier soumis à l'avis du **Comité de Protection des Personnes (CPP)** ou de l'**AFSSAPS (Agence Française de Sécurité Sanitaire des Produits de Santé)**.
- Les chefs de projet : en plus des actions réalisables par les chercheurs, ils pourront accéder à certaines fonctionnalités supplémentaires, telles que la gestion des questionnaires à destination des volontaires par exemple. Etant donné que ce travail peut être un peu lourd pour une seule personne, nous autorisons l'existence de plusieurs chefs de projets si nécessaire.
- L'administrateur : cette personne sera chargée de gérer le site. Elle devra par exemple créer les espaces projets et les profils des utilisateurs, mais aussi affecter chaque utilisateur à un projet. Là encore il sera possible de créer plusieurs administrateurs si le besoin s'en fait sentir.

Les chercheurs disposent actuellement d'un fichier sur lequel figurent des renseignements concernant les volontaires. Ce fichier contient des informations personnelles telles que leur nom, leur prénom, leur adresse et numéro de téléphone, mais aussi un code volontaire. Ce code est pour le moment formé comme ceci : trois premières lettres du nom + deux premières lettres du prénom. Cependant le format de ce code est amené à évoluer au gré des recommandations de la CNIL.

Afin d'anonymiser au maximum les données présentes dans la base de donnée du site, seul le code du volontaire et son adresse email seront conservés.

Enfin, sur les conseils de Sébastien Cat, informaticien sur le site de Theix, l'identification des personnels de l'INRA se fera via leurs nom d'utilisateur et mot de passe LDAP, dans le but de simplifier leur accès au site et de ne pas les rebuter à l'utiliser.

Concernant des chercheurs de laboratoires extérieurs qui pourraient être amenés à participer à un projet, comme des industriels par exemple, il sera possible soit d'ajouter leur nom dans l'annuaire LDAP, soit de leur donner un couple nom d'utilisateur / mot de passe provisoire géré par le site lui-même, comme ce sera le cas pour les volontaires.

1.3.2. Les modules

A partir des informations recueillies, j'ai pu identifier les différents modules du site Internet (figure 8).

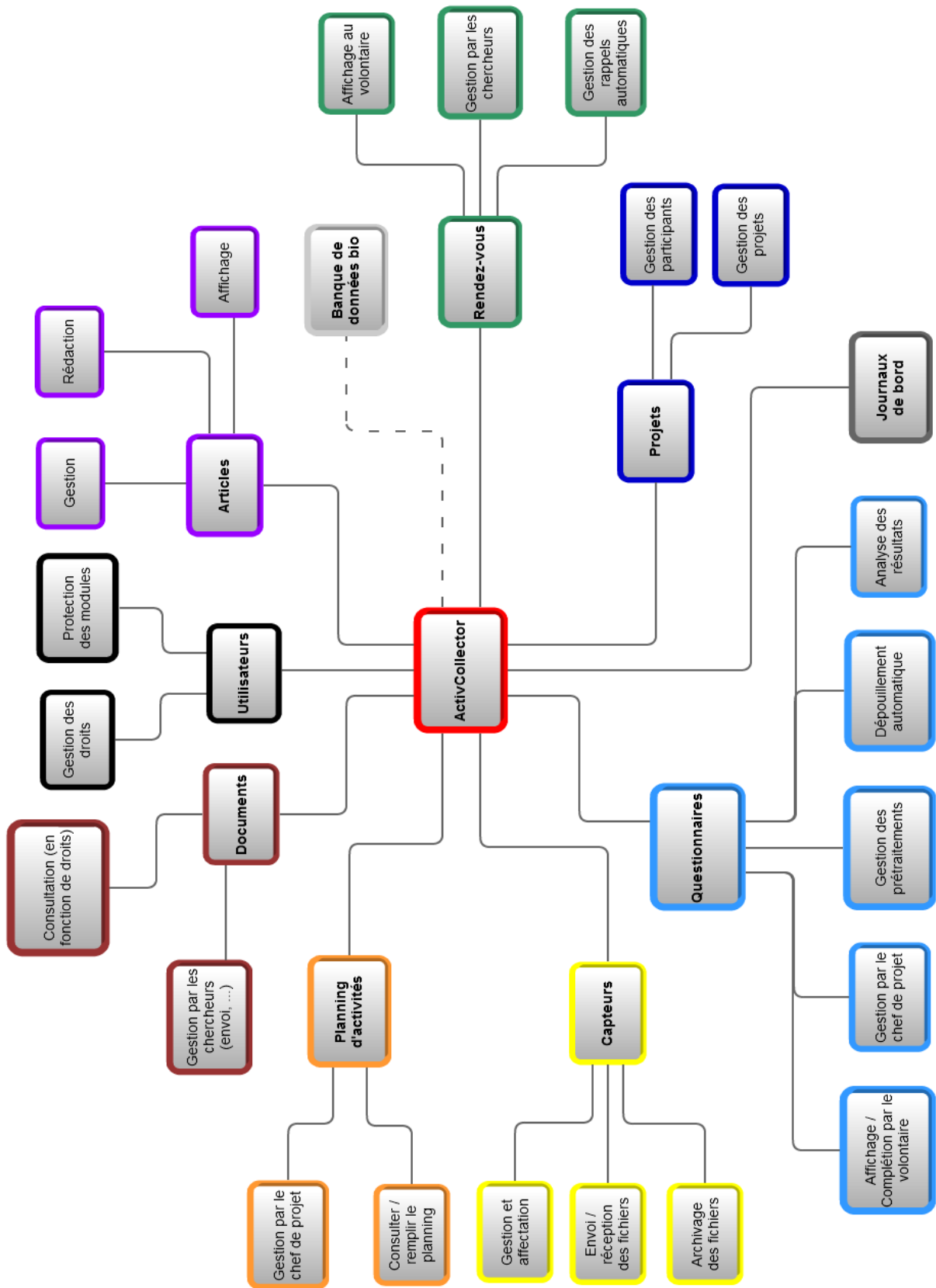


Figure 8 : Modules du site Internet

1.3.2.1. Les projets

Chaque projet que l'on souhaitera gérer via le site Internet fera l'objet de la création d'un espace projet par l'administrateur. Cet espace permettra de rassembler les données, les documents et les utilisateurs liés à un projet donné.

Les volontaires, de même que les chercheurs et les chefs de projets, auront accès à un espace personnel par projet. La durée d'accès à cet espace sera néanmoins limitée pour les volontaires. Nous avons en effet décidé d'autoriser l'accès à partir de la date de début du projet jusqu'à x jours après la fin du projet, cette durée étant paramétrable par l'administrateur. Il relèvera de la responsabilité du chef de projet de décider du statut de son projet : en cours ou terminé.

1.3.2.2. Les rendez-vous

La gestion des rendez-vous sur le site Internet est capitale. C'est en effet un outil qui permettra aux chercheurs de communiquer plus facilement avec les volontaires.

Lorsque le volontaire passe avec succès la visite de pré inclusion, une série de rendez-vous est prévue pendant la durée du protocole. Le chercheur doit donc pouvoir renseigner ces dates via son espace personnel, et le volontaire doit pouvoir les consulter de son côté.

Cependant, il se peut que le chercheur ait besoin de définir de nouveaux rendez-vous au cours de l'étude. Deux possibilités lui seront proposées :

- Soit il proposera de nouvelles dates à un volontaire via le site. Dans ce cas une fois les dates proposées, elles apparaîtront sur l'espace personnel du volontaire concerné. Celui-ci pourra alors choisir la date qui lui convient ou n'en choisir aucune.
- Soit il imposera de nouvelles dates qui apparaîtront directement dans l'agenda du volontaire, sans demande de confirmation ni de choix. On suppose que les deux parties se sont entendues en-dehors du cadre d'ActivCollector.

Par ailleurs, nous avons prévu d'envoyer des courriels de rappel aux volontaires quelques heures avant un rendez-vous. Le délai sera paramétrable par l'administrateur.

Néanmoins, chaque volontaire aura la possibilité de désactiver ces rappels automatiques. On peut en effet imaginer que certains volontaires ne veulent pas recevoir de mails venant de l'INRA, ou encore qu'ils ne souhaitent pas communiquer leur adresse email.

1.3.2.3. Les plannings d'activités

Certains projets nécessitent la mise en place de plannings d'activités pour les volontaires. Ces plannings définissent les activités que doivent réaliser les volontaires tout au long du projet, comme par exemple « marcher à 3km/h », « remplir des questionnaires », « prendre un complément alimentaire ». A chaque action est associée une heure précise.

Les plannings seront créés par les chercheurs et pourront être utilisés de deux façons légèrement différentes.

Les volontaires qui participent à des tests en conditions contrôlées (par exemple en chambre calorimétrique) ont un planning comme celui de la figure 9 :

Volontaire : ADABO Planning de chambre calorimétrique

Heure de début	Durée	Activité
8h		Réveil
8h45	00:05	Marche à 3km/h
8h53	00:10	Marche à 4km/h
9h50	00:05	Questionnaires
10h20	00:30	Vélo d'appartement à 10km/h
11h55	00:20	Repas
...

Figure 9 : Exemple de planning pour les chambres calorimétriques

Si l'activité ne débute pas exactement à l'heure indiquée, le chercheur note l'heure réelle de début. De même, la durée de l'activité est consignée dans le planning.

Pour les volontaires qui font des tests en conditions habituelles de vie (c'est-à-dire non contrôlées), le chercheur ne pourra pas corriger les heures. Il faudra donc que les volontaires puissent faire les corrections eux-mêmes via leur espace personnel.

Enfin, comme pour les plannings de rendez-vous, des rappels par email pourront être envoyés aux volontaires à domicile pour leur montrer le planning du lendemain par exemple.

1.3.2.4. Les capteurs

Il sera possible d'effectuer un suivi des capteurs qui ont été prêtés aux volontaires. Il sera ainsi possible pour les chercheurs de savoir où se trouve tel capteur à un instant t, et de consulter les dates de retour prévues.

Des alertes seront aussi envoyées par email aux personnes qui tarderont à rendre les capteurs qui leurs seront prêtés.

Les capteurs auxquels nous avons pensé sont les suivants :

- **Armband** : appareil qui se porte autour du bras et qui mesure notamment l'accélérométrie et la température corporelle, et estime la dépense énergétique à partir de ces mesures. Il permet aussi de savoir si le sujet est allongé ou debout, éveillé ou endormi.
- **Actiheart (figure 10)** : ce capteur estime lui-aussi la dépense énergétique et l'accélérométrie, en plus de la fréquence cardiaque.
- **Holter** : c'est un électrocardiogramme portable qui permet de mesurer la variabilité cardiaque.
- **Balance** : les chercheurs peuvent prêter des balances aux volontaires si ceux-ci sont amenés à se peser régulièrement au cours du protocole. Nous l'avons ainsi considérée comme un capteur à part entière.
- **Smartphones** : ces téléphones portables sont aujourd'hui de véritables ordinateurs et sont de plus en plus répandus dans la population, notamment chez les jeunes. Trois groupes d'étudiants en licence ont mené cette année un projet pour l'INRA, sous la responsabilité de Philippe Lacomme. Leur but était de réaliser une application pour collecter des données GPS et d'accélérométrie puis les stocker sur un serveur. Le but final serait de proposer aux chercheurs des analyses de ces données pour savoir par exemple à quels moments et pendant combien de temps l'utilisateur a couru, afin de calculer la dépense énergétique et dispenser des conseils aux utilisateurs. Les applications ont été réalisées sur 3 systèmes d'exploitation : Android, iOS (iPhone) et Windows Phone 7.

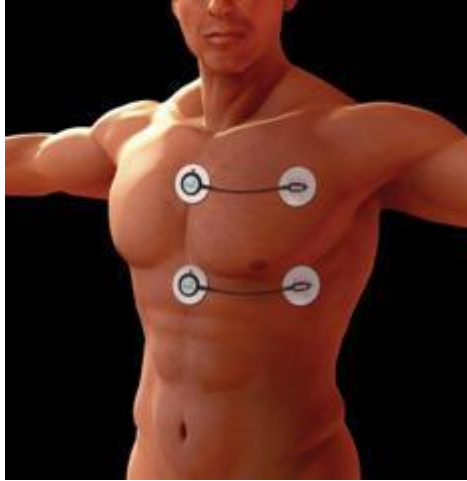


Figure 10 : Capteur [Actiheart]

D'autre part, il sera possible pour les volontaires d'extraire eux-mêmes les données de certains capteurs qui leurs seront prêtés et de les envoyer sur le site d'ActivCollector pour y être stockées dans la base de données. Ces données pourront ensuite être utilisées par des logiciels tels que Finder2E.

Bien entendu, les chercheurs devront eux-aussi pouvoir extraire les données des capteurs et les entrer dans la base.

1.3.2.5. Les questionnaires

Certains projets nécessitent l'envoi de questionnaires aux volontaires. Ceux-ci doivent alors les compléter le plus précisément possible et les renvoyer au laboratoire.

Pour l'instant les chercheurs doivent donc gérer la rédaction des questionnaires sur Word, leur envoi aux volontaires, s'assurer que ces derniers les ont bien renvoyés à temps, puis les dépouiller afin de faire des analyses statistiques.

La phase de dépouillement s'avère être fastidieuse car l'opération est répétitive et coûteuse en temps.

De plus, les questionnaires étant au format papier, leur stockage est difficile et des tris sont régulièrement faits par les chercheurs, faute de place pour continuer à stocker ceux qui sont anciens.

La figure 11 est un exemple de questionnaire au format papier qui a été utilisé dans un projet de recherche clinique.

Nom de code : _____ Sexe : Masculin

Féminin

Date de naissance : _____

Imaginez-vous en train de manger l'aliment ci-dessous. Quelles émotions ressentiriez-vous et avec quelle intensité ?

0 : je ne ressens pas cette émotion
 1 : je la ressens un peu
 2 : je la ressens moyennement
 3 : je la ressens beaucoup
 4 : je la ressens extrêmement

Chocolat

	0	1	2	3	4
aimer	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
doute	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
gêne	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
frustration	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
amusé	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
culpabilité	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 11 : Exemple de questionnaire

1.3.2.6. Les journaux de bord

En étudiant les questionnaires que les chercheurs ont utilisés pour leurs études précédentes, j'ai remarqué qu'il était parfois demandé aux volontaires de remplir des journaux de bord dans lesquels ils indiquent les repas qu'ils prennent ainsi que tout autre apport alimentaire.

Les informations demandées dans ces journaux ainsi que leur structure peuvent être différentes selon les protocoles. Il faut donc que, comme les questionnaires, les chefs de projet puissent créer les journaux de toute pièce.

La figure 12 montre un journal de bord sur lequel le volontaire peut inscrire tous les moments où il a eu envie de grignoter pendant le protocole.

Le journal de bord

Merci de détacher cette feuille et de la remplir si vous avez ressenti le besoin de grignoter.

Quand ?	A quel stade du protocole en êtes-vous ? (début, milieu, fin)	A quelle heure ?	Où ?	Emotions ressenties	Pensées automatiques accompagnant les émotions
<i>Exemple 1 :</i> le 12/06/08	milieu	A 10 h	Au domicile	anxiété	« je ne réussirai jamais »

Figure 12 : Exemple de journal de bord

1.3.2.7. La banque de données biologiques

Le site devra stocker les données biologiques issues d'analyses réalisées au cours de visites de pré inclusion et de toutes les visites ultérieures programmées dans le protocole.

Les données biologiques peuvent être par exemple le poids, le tour de taille, les taux de masses maigre et grasse, le taux de cholestérol, etc. Une liste de caractéristiques biologiques a été définie en concertation avec les chercheurs.

Si besoin, de nouvelles caractéristiques pourront être ajoutées à cette liste, donc il faut veiller à ne pas limiter les types de données stockés.

Les chercheurs pourront ainsi suivre l'évolution de certaines caractéristiques biologiques au cours du protocole, ou faire des comparaisons entre projets.

1.3.3. Etude du module « questionnaires »

1.3.3.1. Caractéristiques principales des questionnaires

Les questionnaires classiques

J'ai distingué 5 grandes étapes dans la réalisation de cette partie :

1. La création et la gestion des questionnaires par les chefs de projets. Il est très important de créer une interface conviviale et pratique afin que les utilisateurs ne soient pas rebutés par l'utilisation du site. Après discussion, nous avons décidé que les questionnaires ne seront gérés que par les chefs de projet.

D'autre part, il faut prévoir une interface afin que les chercheurs puissent modifier les réponses données par les volontaires, par exemple s'ils se sont trompés et/ou n'ont pas été assez précis.

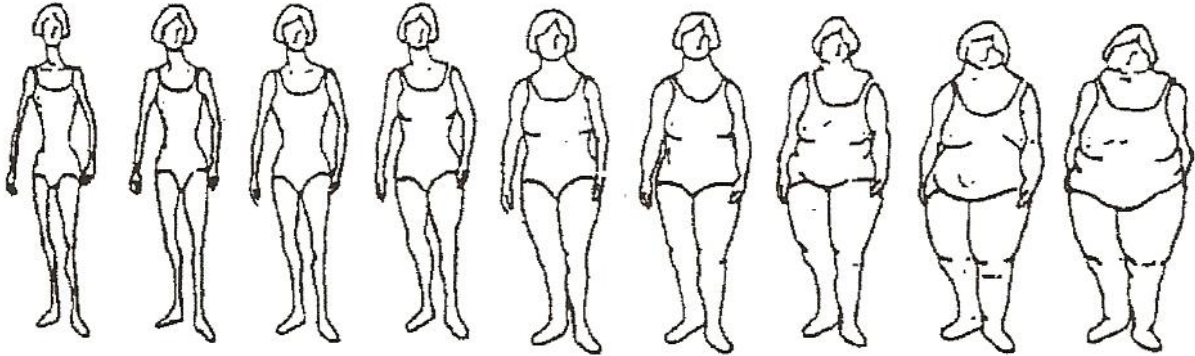
2. L'affichage des questions aux volontaires et l'enregistrement de leurs réponses. Les questions doivent être affichées une par une au volontaire afin qu'il ne soit pas influencé par les questions suivantes ni par ses réponses précédentes. Néanmoins, le volontaire pourra modifier la réponse à la dernière question à laquelle il a répondu. Chaque question peut posséder une condition d'affichage, basée soit sur les caractéristiques du volontaire (son sexe, son âge, son indice de masse corporelle (IMC)) soit sur la réponse donnée à une question précédente.
3. La définition de pré traitements par le chef de projet. Les pré traitements consistent à effectuer des calculs sur les réponses des volontaires aux questions. Par exemple, il sera possible de définir un pré traitement comme étant la somme des valeurs des réponses à la question 2 multiplié par la réponse à la question 25, divisé par la réponse à la question 11. Les résultats du dépouillement des questionnaires de chaque volontaire seront exportables au format Excel.
4. Le dépouillement automatique des questionnaires, qui sera lancé dès qu'un volontaire renverra ses réponses.
5. L'analyse statistique des questionnaires, qui se fera à la demande des chercheurs sur des familles de volontaires, en fonction par exemple de l'âge, du sexe ou de l'indice de masse corporelle.
Là encore les résultats seront exportables au format Excel afin de pouvoir être analysés par le logiciel SAS ou un autre logiciel statistique.

Etant donné que tous les volontaires n'auront pas Internet et que ceux qui ont un accès peuvent le perdre en cours de protocole pour diverses raisons, nous avons prévu de pouvoir imprimer et d'envoyer les questionnaires par courrier. D'autre part, les chercheurs auront un accès sur le site pour enregistrer les réponses fournies par les volontaires ayant reçu une version papier.

J'ai aussi remarqué, durant mon étude des questionnaires, que certaines questions ne sont destinées qu'à certaines personnes. Par exemple, les deux questions de la figure 13 sont similaires mais l'une est destinée aux hommes et l'autre aux femmes :

*** Si vous êtes une femme**

Parmi les silhouettes ci-dessous, choisissez celle qui, selon vous, correspond le plus à votre propre silhouette.



*** Si vous êtes un homme**

Parmi les silhouettes ci-dessous, choisissez celle qui, selon vous, correspond le plus à votre propre silhouette.

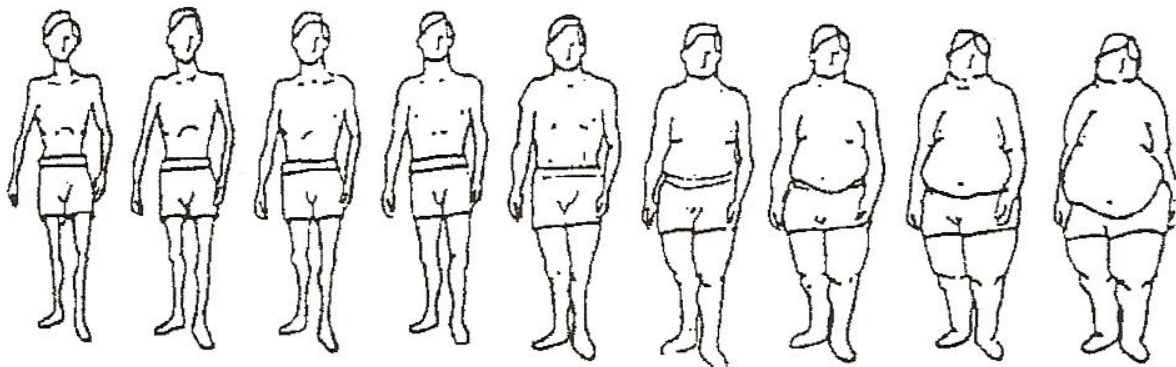


Figure 13 : Exemple de questions différentes selon le sexe du volontaire, extraites d'un questionnaire papier

De la même manière, on peut souhaiter n'afficher une question au volontaire que si une certaine réponse a été donnée précédemment (figure 14) :

En ce moment, suivez-vous un régime alimentaire particulier ?

oui non

Si oui, avez-vous l'impression de ne pas manger à votre faim ?

oui non

Figure 14 : Questions liées, extraites d'un questionnaire papier

Il faut donc introduire la notion de condition d'affichage. Chaque question doit pouvoir être associée à une condition.

Après discussion avec Sylvie Rousset, nous avons décidé qu'une condition sur le volontaire ne peut porter que sur son sexe, son âge (plus précisément sur une tranche d'âges continue) ou son IMC (sur une tranche continue).

De la même manière, nous avons convenu que la condition d'affichage d'une question ne peut être fonction que de la réponse à **une seule** question précédente.

Après avoir étudié les questionnaires existants, j'ai déterminé différents types de questions qu'il sera possible de choisir :

- Question ouverte :
 - Texte court.
 - Texte long.
 - Valeur numérique.
- Echelle de catégories : 1 choix possible parmi N propositions.
- P choix parmi N propositions.
- Echelle continue : le volontaire doit donner une valeur comprise dans un intervalle (par exemple : « Quel est votre degré de douleur sur une échelle de 1 à 10 ? »).

Concernant les questions ouvertes, le chercheur peut choisir si une réponse est obligatoire ou non. Les questions de type « P choix parmi N » ne peuvent, quant à elles, pas être obligatoires car elles laissent au volontaire la liberté de ne rien répondre. Tous les autres types de questions nécessitent de donner obligatoirement des réponses.

Concernant l'ordre d'affichage des questions aux volontaires, il a été décidé que :

- Si le questionnaire ne comporte qu'une partie, les questions peuvent être affichées selon un ordre prédéfini ou aléatoire.
- Sinon :
 - Les parties peuvent être affichées dans un ordre aléatoire.
 - Les questions de toutes les parties (indépendamment les unes des autres) peuvent être affichées selon un ordre prédéfini ou aléatoire.

Les questionnaires alimentaires

Enfin, nous aimerions pouvoir créer un type particulier de questionnaires afin de pouvoir faire des relevés alimentaires ou des questionnaires de fréquence alimentaire qui recensent la nature et la quantité des aliments consommés.

Ce type de questionnaire sera relié à une base de données externe (figure 15). Cette base permet de connaître la composition en nutriments des aliments. Plusieurs bases existent, mais celle qui sera utilisée n'a pas encore été définie au moment où je rédige ce rapport :

- [Ciqual] est une base créée conjointement par le ministère de l'agriculture, le ministère de la recherche, l'INRA, l'Institut de recherche pour le développement (Orstom/IRD), l'Institut national agronomique Paris-Grignon (INA PG) et l'Institut français pour la nutrition (IFN), en 1985.
- [Dietsys] est une base créée par le National Cancer Institute (NCI) américain.
- [Epibase] est une base créée par l'INRA à partir des données fournies par Ciqual et Dietsys.

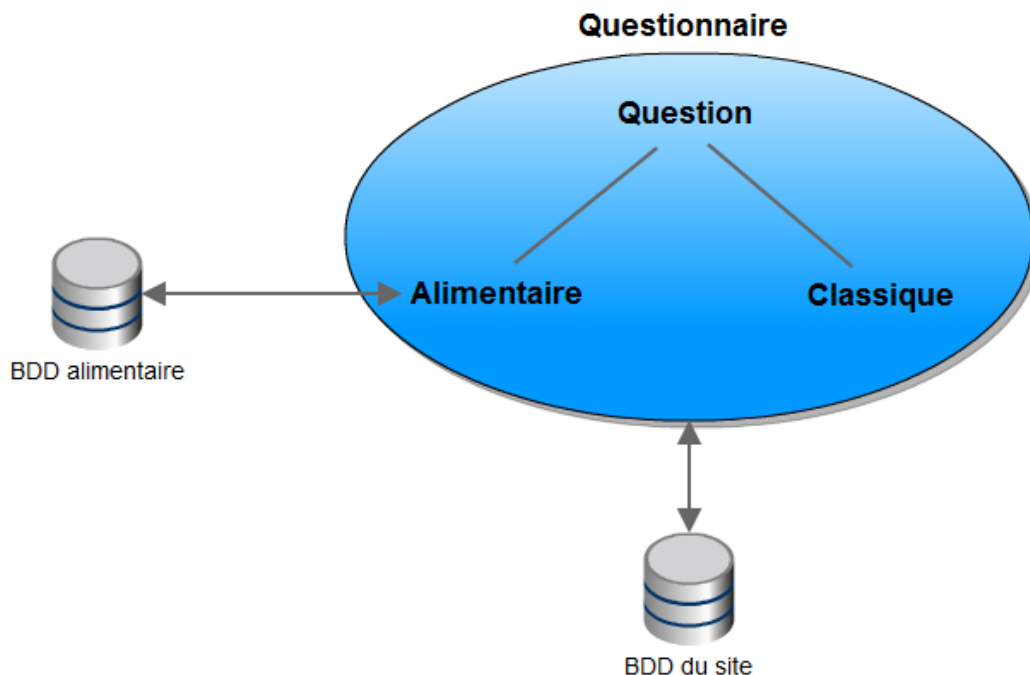


Figure 15 : Représentation des questionnaires

La liaison d'une question à un aliment sera faite par le chercheur lors de la création du questionnaire. Ensuite, au cours du dépouillement, on recherchera les quantités en nutriments apportées par les aliments.

D'autre part, dans certains cas les chercheurs et les volontaires se rencontrent une fois le questionnaire rempli afin de corriger et/ou compléter les réponses. Par exemple, les diététiciens passent actuellement du temps avec les volontaires qui omettent involontairement des détails qui peuvent se révéler importants pour l'étude, comme la quantité de matière grasse du lait (demi-écrémé, écrémé ou entier).

1.3.3.2. Le dépouillement

Cas général

Le chef de projet doit pouvoir définir les pré traitements qui seront exécutés de manière automatique lorsque le volontaire termine un questionnaire.

Tous les types de questions pourront être dépouillés automatiquement, excepté les questions ouvertes.

Les résultats de chaque questionnaire pourront être enregistrés dans un fichier Excel selon le format présenté sur la figure 16, à la demande du chef de projet ou d'un chercheur, afin de pouvoir être analysés par un logiciel de statistique (par exemple SAS) :

Code volontaire	Réponses					Pré traitements		
	Q1	Q2	Q3	...	Qn	Prétraitement 1	...	Prétraitement m
ADABO	5	2	J'ai bien aimé ce...	...	1	3,5	...	7,1

Figure 16 : Format du fichier Excel de dépouillement

Un pré traitement concerne un seul questionnaire et est associé à un libellé tel que « Activités de loisir » par exemple.

Un prétraitement est un ensemble d'opérations à effectuer sur les réponses numériques à certaines questions. Prenons pour exemple le questionnaire sur les activités physiques des volontaires, section loisirs. La question 1 (Q1) porte sur les sports pratiqués (figure 17) :

Parmi les activités listées ci-dessous, choisissez celles que vous avez pratiquées au moins 10 fois au cours des 12 derniers mois. Ne comptez pas le temps passé à marcher ou à faire du vélo pour vous rendre à votre travail, il sera comptabilisé dans la section "activité professionnelle et domestique".

- Aérobie Cheval/Equitation Hockey Natation plaisir Roller-skate Tennis
 Badminton Course cross Jardinage Natation compét. Rugby Vélo plaisir
 Basket-ball Danse Jogging Patinage plaisir Skateboard Vélo vitesse

Figure 17 : Question portant sur les loisirs

Chacun de ces sports est associé à une valeur équivalente à X fois le métabolisme de repos, évaluée en METs.

Puis, pour chaque activité sélectionnée, on pose les questions suivantes (figure 18) :

Question 2. Pendant combien de mois avez-vous pratiqué cette activité au cours des 12 derniers mois ?
Question 3. Combien de fois par mois en moyenne avez-vous pratiqué cette activité ?
Question 4. Quel est le temps moyen que vous avez passé à chaque fois sur cette activité (en minutes) ?

Figure 18 : Questions liées à la question sur les loisirs

Le pré traitement nommé « Activités de loisirs de l'année passée (en MET-h / semaine) », est composé des opérations suivantes :

$$R = \text{résultat Q2} * \text{résultat Q3} * \text{résultat Q4} * \text{résultat Q1} / 60 / 52$$

Ici 60 désigne le nombre de minutes par heure et 52 le nombre de semaines dans une année.

Ces opérations nous retournent un résultat R (exprimé ici en nombre d'heures par semaine) pour une activité. Cependant, si plusieurs activités ont été sélectionnées, alors il y a autant de valeurs de R que de sports. Le résultat final est la somme de tous les résultats intermédiaires.

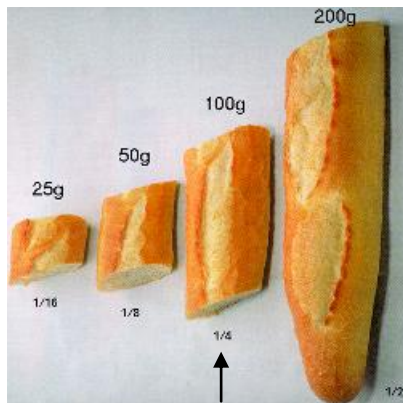
Cas particulier : les tables nutritionnelles

Certaines questions peuvent être posées dans le but de connaître les quantités de nutriments ingérés sous forme calorique, glucidique, lipidique et protéique à partir des aliments.

On distingue deux types de questions à ce propos :

- Le premier consiste à demander au volontaire la fréquence à laquelle il mange une certaine quantité d'un aliment donné (figure 19).

Habituellement au cours d'une semaine, combien de fois mangez-vous un quart de baguette de pain blanc ?



Votre réponse :

Figure 19 : Exemple de question liée à la fréquence de consommation d'un aliment

- Le second consiste à lui demander, pour un jour donné, la quantité consommée d'un aliment (figure 20).



Figure 20 : Exemple de question liée à la quantité d'absorption d'un aliment

Le chercheur doit associer un aliment à la question, ainsi que des informations sur la période et les quantités énoncées dans le libellé de la question, afin que lors du dépouillement le calcul de la conversion d'une quantité d'aliments en une quantité de nutriments soit fait automatiquement.

L'aliment est associé à sa composition nutritionnelle via une base de données externe. Le choix de cette base n'est à l'heure actuelle pas encore défini. Ce pourront être les tables Ciqual (Régal), Dietsys ou Epibase (basé sur Régal et Dietsys).

1.3.3.3. Analyse statistique des résultats dépouillés

Une fois les dépouillements réalisés, les chefs de projets et les chercheurs pourront analyser les résultats obtenus par famille de volontaires. Une famille est définie selon les mêmes critères que dans le logiciel Finder 2E (sexe, âge, IMC, VO2, indice sportif).

L'analyse créera un fichier Excel avec 2 onglets :

- Le premier contiendra les résultats aux questions et aux prétraitements effectués pour chaque volontaire de la famille (figure 21) :

Volontaires	Réponses			Pré traitements		
	Q1	...	Qn	PT1	...	PTm
ADABO	2		6	2,5		4,12
CHANA	4		3	3,42		3,68
...						

Figure 21 : Premier onglet de la feuille d'analyses

- Le second contiendra les indices statistiques sur la famille complète (figure 22) :

	MoyQ1	ECQ1	MinQ1	MaxQ1	Quartile1Q1	Quartile3Q1	MédQ1	...
Famille	2,45	1	0	5	0,8	3,2	2,2	

Figure 22 : Deuxième onglet de la feuille d'analyses

Les colonnes qui doivent figurer dans ce dernier tableau sont les suivantes :

- MoyQi est la moyenne des réponses à la question i.
- ECQi est l'écart type des réponses à la question i.
- MinQi est la valeur minimale des réponses à la question i.
- MaxQi est la valeur maximale des réponses à la question i.
- Quartile1Qi est la valeur du premier quart de la série de réponses à la question i.
- Quartile3Qi est la valeur du troisième quart de la série de réponses à la question i.
- MédQi est la valeur médiane des réponses à la question i.

Pour les questions qui demandent un classement et dont la réponse est exprimée sous forme de rang, le but est de déterminer :

- La somme des rangs de chaque élément, résultant des réponses de tous les volontaires (figure 23).

Volontaire	Bœuf	Agneau	Veau
ADABO	1	2	3
CHANA	2	3	1
Somme des rangs	3	5	4

Figure 23 : Exemple de somme des rangs

- La fréquence de classement de chaque élément. Exemple : boeuf classé 1er par 75% des volontaires ; etc.

2. Choix techniques et implémentation

2.1. Principes de base

Lorsqu'un internaute demande à consulter une page d'un site Internet, il envoie une requête HTTP (pour « **H**yper**T**ext **T**ransfer **P**rotocol », ou « protocole de transfert hypertexte » en français) au serveur qui héberge le site, sous la forme d'une URL telle que <https://www3.clermont.inra.fr/ActivCollector> (figure 24).

Le serveur génère alors la page demandée et la retourne en réponse au navigateur du client, qui se charge de l'afficher.

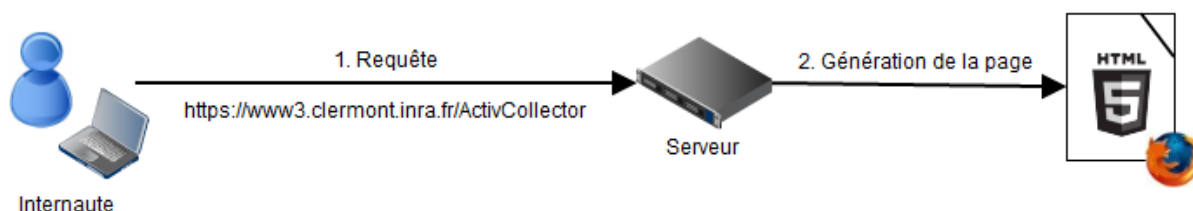


Figure 24 : Principe de demande d'une page sur Internet

Le langage utilisé pour représenter une page est le HTML (pour « **H**yper**T**ext **M**arkup **L**anguage », ou « langage de balisage hypertexte » en français). Il permet de structurer les informations d'une façon logique. Dans l'exemple ci-dessous, deux paragraphes font partie d'un même bloc logique (figure 25) :

```
<div>
  <p>Premier paragraphe</p>
  <p>Second paragraphe</p>
</div>
```

Figure 25 : Exemple de code HTML

La page est ensuite mise en forme par des feuilles de style CSS qui associent à des éléments des mises en forme particulières, pour colorer du texte par exemple. Ces feuilles déterminent le design du site.

Si le site est statique, le serveur ne fait que délivrer des pages HTML dont le contenu ne change jamais. Si le site est dynamique, c'est-à-dire que le contenu des pages change en fonction de certains paramètres (comme le profil de la personne connectée), alors il est

nécessaire d'utiliser un langage de programmation pour déterminer les informations à renvoyer au client.

2.2. Choix d'un langage de programmation

Il existe plusieurs langages de programmation utilisés pour créer des sites Internet dynamiques. Voici un tableau décrivant les avantages et inconvénients des principaux langages (figure 26) :

	PHP	JEE	ASP.Net	Ruby On Rails
Facilité de déploiement	+++	+	++	++
Portabilité	+++	+++	+	+++
Documentation	+++	+++	++	++
Maîtrise / Temps d'apprentissage	+++	++	+	+
API disponibles	++	+++	+++	+++

Figure 26 : Comparaison des langages côté serveur

Le PHP (pour « **PHP: Hypertext Preprocessor** », ou « **PHP : Préprocesseur hypertexte** ») est un langage très répandu dans le monde amateur, mais moins dans le monde professionnel, où il a parfois mauvaise réputation. En effet, ce langage permet beaucoup de libertés, ce qui peut conduire à un code qui devient difficilement gérable s'il est mal utilisé. Certaines entreprises attachent à PHP une image d'amateurisme et de bidouille. Il y a en revanche une très grande communauté d'utilisateurs, ce qui induit de nombreuses sources d'informations disponibles sur Internet. D'autre part, un site PHP est très facilement déployable et fonctionne avec tous les systèmes d'exploitation, que ce soit Mac, Linux ou Windows.

Le développement d'applications web JEE (Java Enterprise Edition) se fait souvent avec le framework Struts. Cette infrastructure m'a semblé plutôt lourde à mettre en place dans le cadre de ce projet. Il fallait en effet que le développement commence plutôt rapidement après les phases d'analyse et de choix techniques. Il y a en revanche beaucoup de documentation sur Internet du fait de la grande utilisation de ce langage par les entreprises, et nombre de bibliothèques et d'outils sont disponibles. Pour utiliser cette infrastructure, un serveur d'application doit être mis en place, tel que GlassFish ou JBoss.

ASP.Net, qui fait partie de la plateforme Microsoft .Net, possède de nombreuses API mais aussi l'inconvénient de ne fonctionner que sur un serveur web IIS (pour « **Internet Information Services** »). Il faut cependant noter l'existence du projet [Mono] qui a pour but de porter les applications .Net sous Linux.

Ruby On Rails est un framework web très jeune, car sa première version est sortie en 2004. D'après les témoignages que j'ai lus sur Internet, ce framework est assez long à prendre en main mais peut être très efficace. D'autre part, les API changent rapidement et certaines deviennent assez vite obsolètes car non maintenues.

Etant donné tous ces éléments et les contraintes relatives à ce projet, j'ai choisi d'utiliser le langage PHP, qui présente un bon compromis entre la rapidité de mise en place et les performances. J'ai ensuite dû faire le choix de construire le site entièrement moi-même ou d'utiliser un framework. J'ai donc fait des recherches sur ces outils ainsi sur des outils d'accès aux bases de données.

2.3. Choix d'un outil de mapping objet-relationnel

Plusieurs outils de mapping objet-relationnel (ORM, pour « **Object-Relational Mapping** ») sont disponibles pour PHP, tels que Propel, [Doctrine], PHPDataMapper ou encore PHPMyObject. Les deux premiers sont néanmoins les plus réputés et les plus utilisés. Ils sont respectivement soumis aux licences MIT et LGPL.

Propel et les versions 1.x de Doctrine se basent sur le patron Active Record pour récupérer les données en base : un objet correspond à un tuple de la base de données. Ainsi, lorsque l'objet est modifié, le tuple l'est aussi. D'autre part, il faut que chaque classe redéfinisse certaines méthodes et / ou étende une classe de base. Ainsi, dans Doctrine 1.x chaque classe doit hériter de `Doctrine_Record` et implémenter une méthode `setTableDefinition()` qui décrit la structure de l'objet dans la base de données.

La version 2 de Doctrine utilise un autre patron de conception : Data Mapping. Cela consiste à faire correspondre deux modèles de données : le modèle objet et le modèle de la base de données. L'avantage est que l'on peut ainsi accéder aux données sans connaître la structure de la base. Cette dernière version de Doctrine s'inspire beaucoup du framework [Hibernate], très réputé en Java. En effet, Doctrine possède son propre langage de requêtage, DQL (pour « **Doctrine Query Language** »), à l'instar d'Hibernate qui propose le langage HQL (pour « **Hibernate Query Language** »). Il permet de gérer la persistance des données d'une manière SQL orientée objet.

```
$query = $em->createQuery("SELECT u
                             FROM User u
                             JOIN u.address a
                             WHERE a.city = 'Berlin'");
$users = $query->getResult();
```

Figure 27 : Exemple de requête utilisant DQL

Dans l'exemple précédent (figure 27), on sélectionne tous les utilisateurs qui habitent à Berlin. Les champs figurant dans la requête, comme `a.city`, sont les propriétés des objets et non les colonnes des tables SQL. Les résultats sont « hydratés », c'est-à-dire transformés en objets tels que définis dans les classes PHP (figure 28).

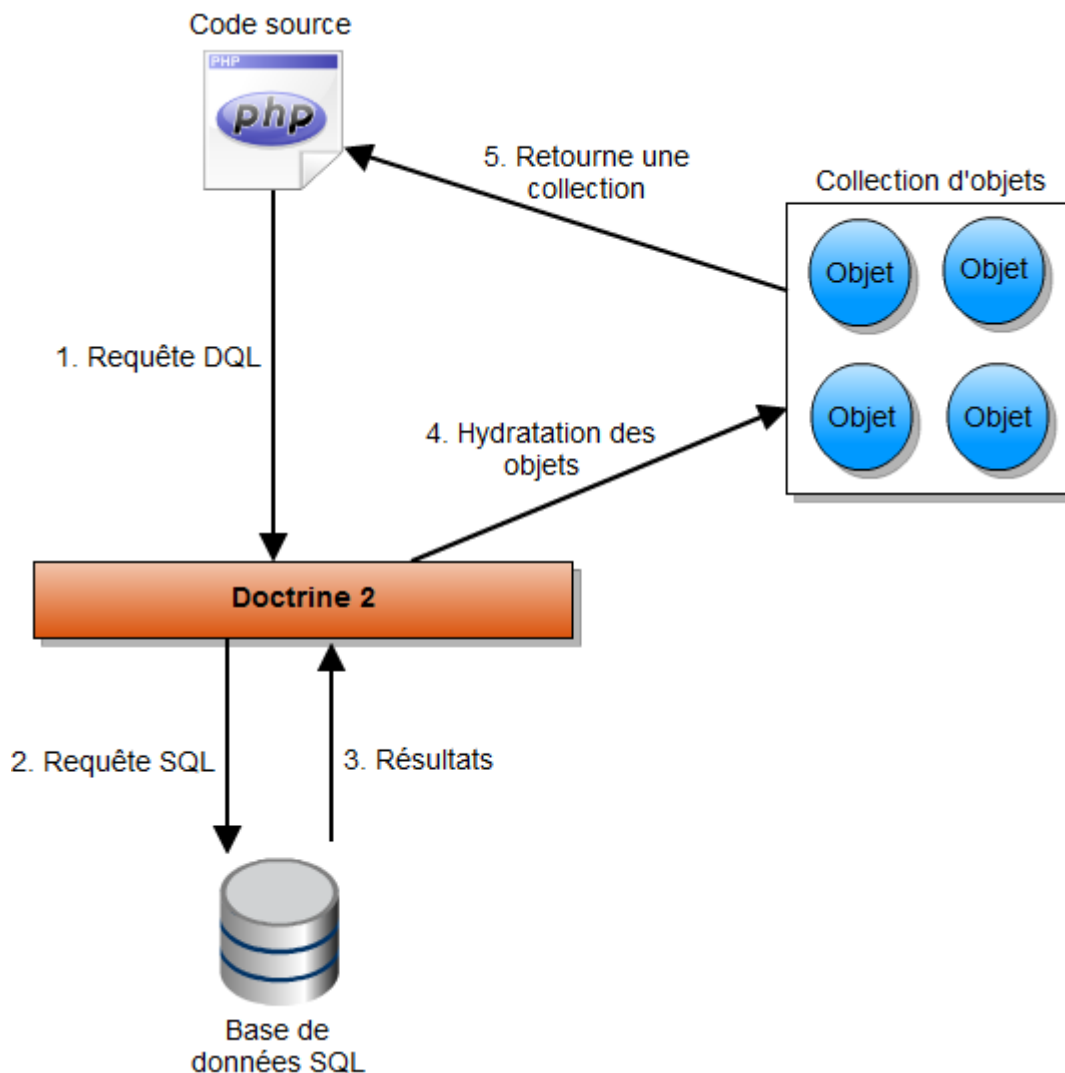


Figure 28 : Fonctionnement de Doctrine

Après de nombreuses recherches sur Internet, il m'a paru évident que Doctrine, dans sa version 2, est le plus complet des ORM. De plus, il possède une communauté active et une

documentation très bien rédigée. J'ai néanmoins passé du temps au début du stage pour trouver comment le configurer.

Comme toute bibliothèque, elle possède aussi quelques défauts :

- Une colonne ne peut pas être auto incrémentée lorsqu'elle fait partie d'une clé composite (clé primaire composée de plusieurs colonnes) [Bug Doctrine].
- Les clés étrangères ne peuvent pas faire partie d'une clé primaire. Cette limitation sera résolue dans la prochaine version, la v2.1, prévue pour fin juin 2011 [Feuille de route Doctrine 2].

Par rapport à sa dernière version 1, Doctrine 2 apporte :

- 30% de gain mémoire.
- 20% de gain de rapidité d'exécution.
- L'utilisation d'annotations ou de fichiers XML pour décrire le mapping objet / base de données. En comparaison, Hibernate utilise des fichiers XML et JPA (pour « **Java Persistence API** ») utilise les annotations.

2.4. Etude des frameworks PHP

Après avoir choisi le langage PHP pour le développement, je me suis demandé s'il serait mieux de construire le site entièrement moi-même ou si un framework pourrait simplifier une partie du développement. Il y a aujourd'hui un vaste choix de frameworks PHP.

Les frameworks apportent une architecture de base ainsi que des fonctionnalités déjà implémentées. Tous utilisent une architecture Modèle Vue Contrôleur (MVC, figure 29).

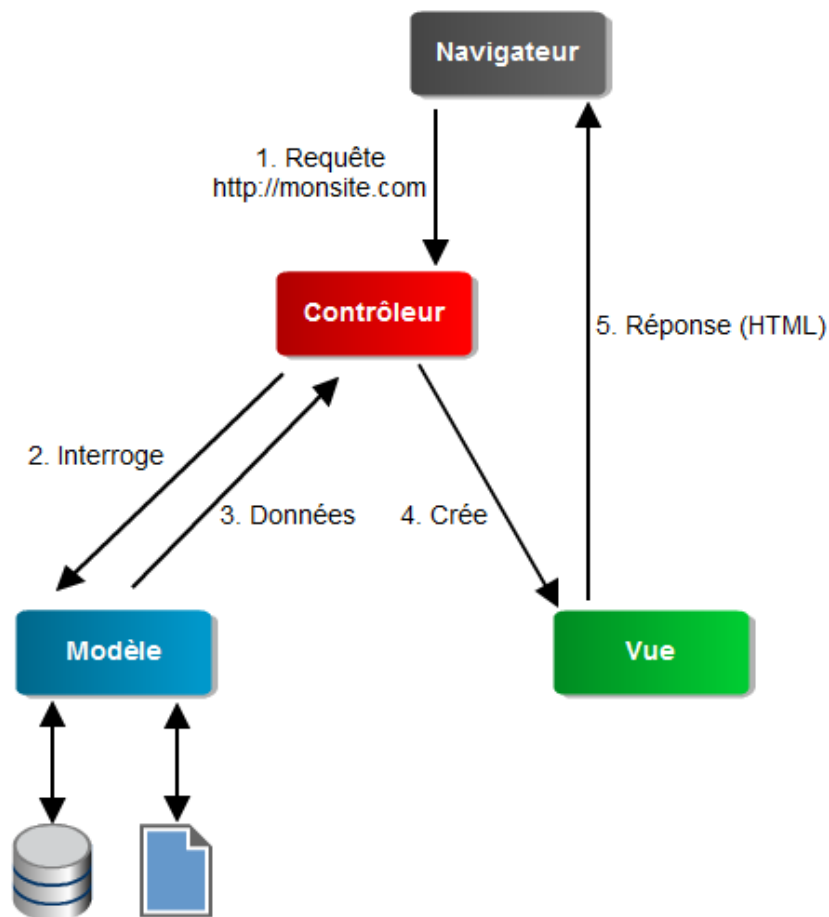


Figure 29 : Architecture MVC

Dans le cas d'une application web, le contrôleur est appelé lorsque le navigateur envoie une requête au serveur pour connaître le contenu d'une page. Il interroge alors éventuellement la couche Modèle afin d'obtenir des données venant de diverses sources comme une base de données par exemple. Il crée ensuite une vue avec ces données, comme une page HTML, qu'il renvoie au navigateur.

La vue ne communique jamais directement avec la couche Modèle, et ni la vue ni le contrôleur n'accèdent directement aux sources de données.

J'ai concentré mes recherches sur les quatre frameworks les plus connus. La figure 30 présente leurs avantages et inconvénients :

Framework	Licence	Avantages	Inconvénients
Symfony	MIT	Framework complet	Pas de plugin pour utiliser Doctrine 2 dans Symfony 1,x
		La v2 intégrera Doctrine 2 par défaut	La v2 est repoussée (elle était initialement prévue fin mars 2011)
Zend Framework	BSD	Couplable à Doctrine 2	N'a pas l'air si facile à prendre en main
			Propose beaucoup plus de fonctionnalités que ce dont nous aurions besoin
CodeIgniter	Apache	Extrêmement léger et rapide	Pas de système d'authentification livré avec le framework, mais il existe de très bons plugins
		Importantes bibliothèques de fonctions d'aide	
		Couplable à Doctrine 2	
		Très peu de configuration	
		Pas d'utilisation de ligne de commande	
CakePHP	MIT	Léger, rapide	Possède son propre ORM
		Facile et rapide à mettre en place	Couplage pas très simple à Doctrine 2

Figure 30 : Avantages et inconvénients des frameworks PHP

De plus, il faut noter que tous proposent :

- Un mécanisme d'internationalisation.
- Un système de cache, ce qui allège la charge du serveur et rend le chargement des pages plus rapide du côté utilisateur.
- Des fonctionnalités très pratiques pour vérifier les données transmises via des formulaires.

La version actuelle de Symfony intègre l'ORM Propel par défaut, mais la prochaine version le remplacera par Doctrine 2.

Mes critères principaux pour le choix d'un framework ont été la simplicité d'utilisation, l'efficacité et la possibilité d'intégrer Doctrine 2.

CodeIgniter est celui qui m'a le plus séduit. En plus des éléments précédents, il ne nécessite en effet que très peu de configuration et permet de personnaliser tous les éléments du framework.

2.5. Présentation de CodeIgniter

2.5.1. Principe général

Voici un schéma qui présente le fonctionnement de CodeIgniter (figure 31):

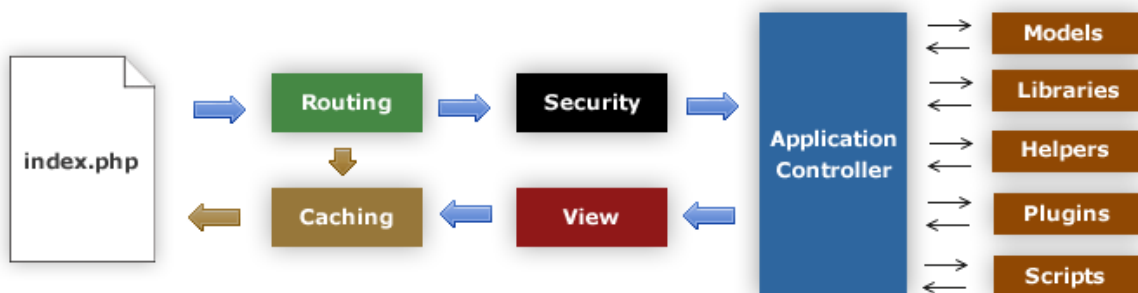


Figure 31 : Fonctionnement de [CodeIgniter 2]

On accède à toutes les pages de l'application via la page `index.php`. C'est elle qui nous redirige ensuite vers le contrôleur demandé. Les URL ont la forme suivante : <https://www3.clermont.inra.fr/ActivCollector/index.php/contrôleur/methode/parametres>.

Après `index.php`, on trouve respectivement :

- Le nom du contrôleur à exécuter.
- Le nom de la méthode du contrôleur. Elle correspond à une action à réaliser.
- Les paramètres GET à passer en paramètre à la méthode.

Par exemple, l'URL pour accéder à l'interface de création des questionnaires, que nous verrons plus tard, est `index.php/managerAccount/questionnaire/form/1`, ce qui signifie qu'on exécute la méthode `form()` du contrôleur questionnaire qui se trouve dans le répertoire `managerAccount`, en lui passant en paramètre l'identifiant du questionnaire : 1.

Sur le schéma ci-dessus, c'est le routeur qui est chargé de décomposer l'URL en actions. Il est possible de modifier son comportement pour avoir des URL qui suivent une autre syntaxe.

Une fois le contrôleur trouvé, le système regarde si un fichier cache existe pour la page demandée, ce qui permet d'accélérer le chargement des pages lorsque leur contenu n'a pas changé.

Si la page n'est pas en cache, la requête HTTP est sécurisée, de même que les données transmises par l'utilisateur (cf boîte « Sécurité »).

Ensuite seulement le contrôleur demandé est exécuté. Dans son code il est notamment possible de faire appel :

- A des modèles, pour accéder à des données.
- A des bibliothèques fournies par CodeIgniter, comme par exemple un système de sessions basé sur des cookies ou une bibliothèque d'upload de fichiers. Il est aussi possible de créer ses propres bibliothèques.
- A des « helpers », qui sont des fonctions génériques pouvant être utilisées n'importe où dans l'application, par exemple pour faciliter la gestion des URL dans le code ou pour faciliter la gestion des dates. Là encore il est possible de créer ses propres helpers, et même de modifier ceux existants pour leur donner un comportement différent.

2.5.2. Intégration de Doctrine 2 dans CodeIgniter

Doctrine 2 s'installe dans CodeIgniter comme une bibliothèque. Il faut donc créer une classe Doctrine dans un répertoire spécifique. Son constructeur doit inclure le fichier de configuration de la base de données ainsi que le chargeur de classe (ClassLoader) de CodeIgniter (figure 32) :

```
require APPPATH.'config/database.php';
require_once APPPATH.'libraries/Doctrine/Common/ClassLoader.php';
[...]
$connectionOptions = array(
    'driver' => 'pdo_mysql',
    'user' => $db['default']['username'],
    'password' => $db['default']['password'],
    'host' => $db['default']['hostname'],
    'dbname' => $db['default']['database']
);
```

Figure 32 : Inclusion de fichiers dans le constructeur de la bibliothèque Doctrine

Le reste de la configuration se fait ensuite comme décrit dans la documentation de Doctrine. Il faut notamment renseigner le répertoire où sont stockées les classes utilisées dans l'application ainsi qu'un répertoire où seront générées des classes proxy. Ces classes ajoutent des comportements aux classes de base (les modèles) et sont notamment utilisées par Doctrine pour faire du lazy-loading, c'est-à-dire extraire de la base de données le moins d'informations possible. Lorsqu'on tente d'accéder à des informations non extraites, Doctrine exécute alors une nouvelle requête sur la base de données pour extraire les nouvelles données.

Considérons l'exemple de la figure 33, où un questionnaire est composé de 1 à n sections :

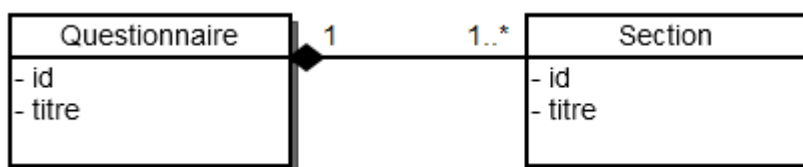


Figure 33 : Diagramme de classes de l'exemple

Voici un schéma qui montre le fonctionnement de Doctrine lorsqu'on utilise le mode lazy-loading :

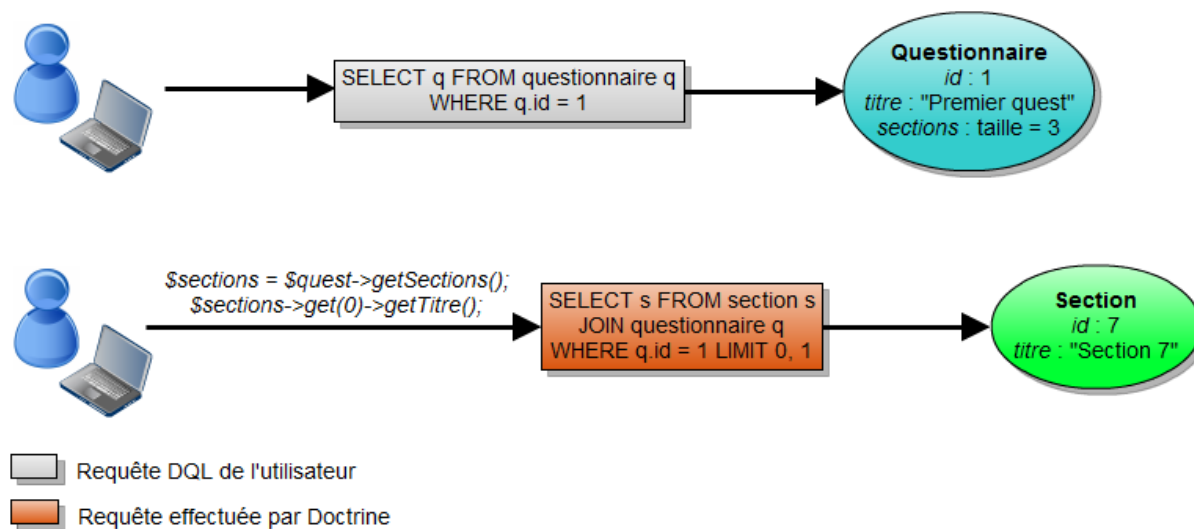


Figure 34 : Scénario d'utilisation du mode lazy-loading

La figure 34 présente deux actions successives :

1. Tout d'abord le développeur exécute une requête DQL pour accéder au questionnaire d'identifiant 1. Tous les attributs de l'objet créé sont renseignés, excepté `sections`. La seule information disponible à cet instant est la taille de la collection, c'est-à-dire le nombre de sections associées au questionnaire.
2. Le développeur cherche ensuite à accéder à une méthode (`getTitre()`) sur le premier objet de la collection de sections. Doctrine exécute donc automatiquement une requête pour extraire de la base de données la section voulue. On remarque que seul l'objet auquel on tente d'accéder est récupéré, et non pas la collection entière. Ainsi, si le développeur tente d'accéder, plus tard, à un autre élément de la collection, une nouvelle requête SQL sera exécutée.

Le lazy-loading peut ainsi être très consommateur de mémoire et ralentir considérablement l'application, c'est pourquoi il existe un autre mode de chargement des données, l'eager-

loading. Cet autre mode permet de charger un objet et tous ses objets associés en une seule requête (figure 35).

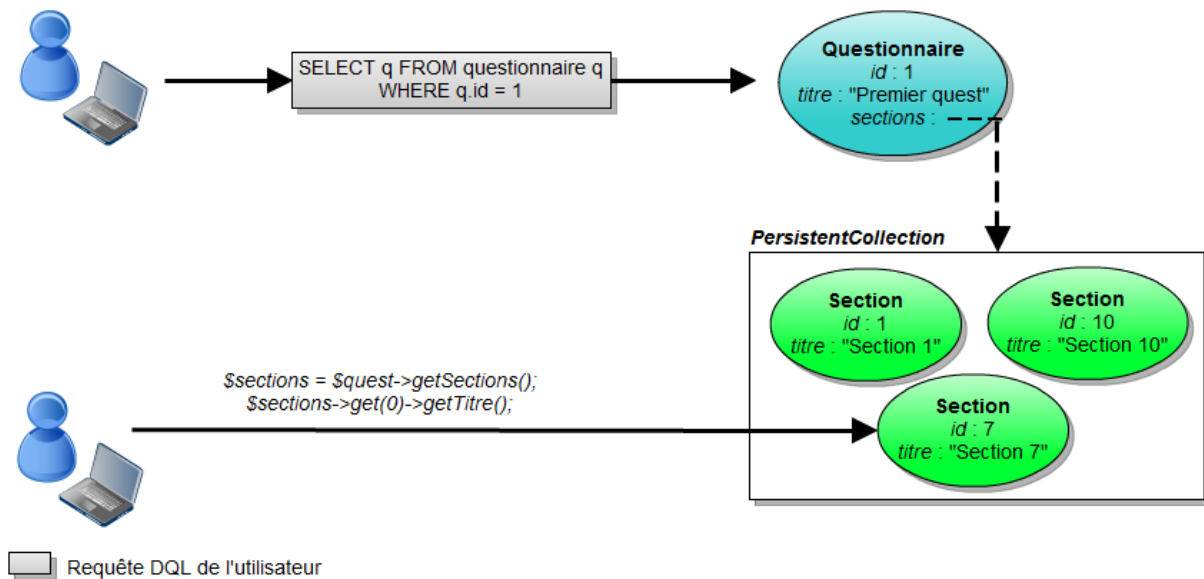


Figure 35 : Scénario d'utilisation du mode eager-loading

On remarque facilement la différence avec le lazy-loading : la première requête DQL exécutée par le développeur récupère non seulement le questionnaire mais aussi toutes les sections qui lui sont liées.

Ainsi, lorsqu'il tente d'accéder à la méthode `getTitre()` sur le premier objet de la collection, aucune autre requête SQL n'est effectuée car les sections sont déjà chargées en mémoire.

2.6. Etude des frameworks JavaScript

2.6.1. Présentation de JavaScript

JavaScript est un langage de programmation qui est exécuté par le navigateur, du côté client. Il n'entraîne donc aucune charge pour le serveur. Il permet, en se basant sur le DOM (pour « **D**ocument **O**bject **M**odel », ou « modèle objet de document » en français), de modifier la page HTML après qu'elle ait été envoyée par le serveur au navigateur.

JavaScript modifie le DOM, et ces modifications sont ensuite répercutées sur la page affichée à l'utilisateur.

Le DOM est une représentation objet de la page web. Ainsi, à chaque balise HTML correspond un objet en mémoire, et le document est représentable sous forme d'arbre. Un

élément peut donc avoir un parent, des ascendants, des descendants et des frères. La figure 36 illustre la logique du DOM :

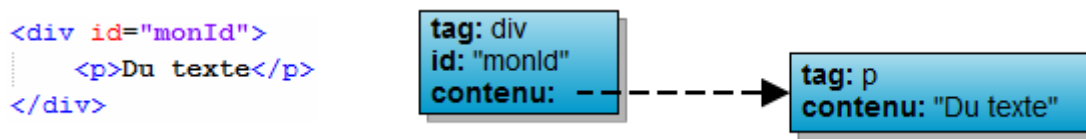


Figure 36 : Correspondance entre éléments HTML et DOM

Sur cet exemple, l'élément `p` a comme père le `div` d'identifiant `monId`. Le paragraphe `p` est donc le fils direct de ce bloc `div`.

JavaScript est un langage très puissant mais qui peut être difficile à maîtriser lorsque l'on cherche à ce que l'application soit compatible avec le plus de navigateurs possibles. En effet, chaque navigateur a des particularités et certains ne respectent pas tous les standards, comme Internet Explorer par exemple.

Ainsi, si l'on veut n'exécuter un script que si le navigateur est Internet Explorer 6 ou inférieur, on peut entourer l'inclusion du script par un commentaire conditionnel (figure 37) :

```
<!--[if lte IE 6]>
  <script src="./js/functions-ie6.js"></script>
<![endif]-->
```

Figure 37 : Inclusion conditionnelle d'un script JavaScript

Ceci devient vite fastidieux lorsqu'on a beaucoup de code, dont la lisibilité et la facilité de maintenance se dégradent.

2.6.2. Choix d'un framework

Il existe un vaste choix de frameworks JavaScripts qui ont pour but de simplifier la programmation. J'ai retenu les trois plus connus que sont jQuery, MooTools et Prototype.js.

La figure 38 est un tableau qui montre la compatibilité de ces frameworks avec les navigateurs Internet les plus connus. La colonne « Version actuelle » indique la version la plus récente du navigateur à l'heure où je rédige ce rapport. J'ai indiqué, pour chaque framework, les versions des navigateurs supportées.

	Version actuelle	jQuery	MooTools	Prototype js
Internet Explorer	9	6+	6+	6+
Mozilla Firefox	5.0	2+	2+	1.5+
Safari	5	3+	3+	2.0.4+
Opera	11.11	9+	9+	9.25+
Google Chrome	12.0	1+	4+	1+

Figure 38 : Compatibilités des frameworks entre navigateurs

Il n'y a pas ici de différence significative : tous les frameworks sont compatibles avec d'anciennes versions de tous les navigateurs.

Voici maintenant les différents avantages et inconvénients que j'ai trouvés pour chacun d'eux (figure 39) :

Framework	Licence	Taille	Avantages	Inconvénients
jQuery + jQuery UI	MIT	84 ko + 204 ko = 288 ko	Bonne documentation, beaucoup d'exemples et de démonstrations	Trop de choix de plugins ?
			Vaste choix de plugins	
			Apprentissage rapide	
			Syntaxe courte	
			Personnalisable	
Prototype js + script.aculo.us	MIT	160 ko + 156 ko = 316 ko	Syntaxe courte	Structure de la documentation
				Lourde
				Quelques problèmes de compatibilité entre navigateurs (qui tendent à être résolus)
MooTools	MIT	86 ko + plugins	Légère	Documentation
			Syntaxe courte	
			Plugins	

Figure 39 : Avantages et inconvénients des différents frameworks JavaScript

Tout d'abord, on peut remarquer que Mootools et jQuery sont les frameworks les plus légers. En effet, leur poids est d'environ 85 kilo-octets, contre 160 pour Prototype js. Le poids influe sur le temps de chargement des pages web. Néanmoins, on ajoute souvent certains plugins, ce qui alourdi considérablement la taille globale des scripts à charger.

Ensuite, les documentations de MooTools et Prototype js m'ont semblé bien moins complètes et faciles d'utilisation que celle de jQuery. Or la documentation est essentielle lorsqu'il s'agit d'apprendre à utiliser un framework.

J'ai finalement choisi d'utiliser jQuery car, en plus des avantages que je viens d'énumérer, de nombreuses autres ressources sont disponibles sur Internet. De plus, Libo Ren, un doctorant

de Philippe Lacomme, connaît bien ce framework pour l'avoir déjà utilisé. J'ai ainsi pu lui demander quelques conseils lorsque j'avais des difficultés.

2.6.3. Présentation de jQuery

jQuery dispose d'un ensemble de plugins très pratiques et fortement paramétrables appelé jQuery UI. Ces plugins permettent aux développeurs d'utiliser des outils configurables tels que des boîtes de dialogue, des calendriers, des listes triables, etc. Ces éléments pré conçus m'ont beaucoup aidé à concevoir des interfaces plus conviviales.

La figure 40 montre l'utilisation de Datepicker [jQuery UI Datepicker], un des plugins regroupés dans jQuery UI qui permet de créer des calendriers :

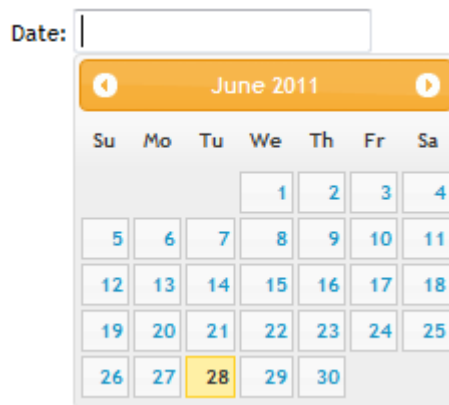


Figure 40 : Exemple d'utilisation de Datepicker, un des plugins de jQuery UI

Avec jQuery, l'accès aux éléments de la page se fait de la façon suivante (figure 41) :

```
$('selecteur', 'contexte').fonction();  
jquery('selecteur', 'contexte').fonction();
```

Figure 41 : Syntaxe de la sélection d'éléments avec jQuery

Les deux instructions précédentes sont équivalentes. Le symbole \$ est juste un raccourci pour désigner l'objet `jquery`.

Le sélecteur peut désigner tous les éléments HTML (comme `p` ou `div` par exemple) d'un type donné ou un groupe d'éléments correspondant à des critères précis, et le contexte est un autre sélecteur qui indique que le sélecteur principal ne doit faire sa recherche que parmi les éléments retournés par le contexte.

Le code HTML de la figure 42 montre divers éléments (div, p, table, etc). Certains possèdent des attributs couramment utilisés en HTML (id, class) ou personnalisés (mon-attribut-perso):

```
<div id="info">
  <table mon-attribut-perso="maValeur">
    <tr>
      <td class="impair">1</td>
      <td class="pair">2</td>
    </tr>
    <tr>
      <td class="impair">3</td>
      <td class="pair">4</td>
    </tr>
    <tr>
      <td class="impair">5</td>
      <td class="pair">6</td>
    </tr>
  </table>
</div>
```

Figure 42 : Code HTML d'exemple

Selon les standards du W3C (« World Wide Web Consortium »), un identifiant doit être unique dans une page, tandis que les classes peuvent être utilisées sur plusieurs éléments.

Voici quelques instructions qui montrent la concision et la simplicité de jQuery (figure 43) :

```
$('#info > table[mon-attribut-perso]').size();

$('td.impair', 'table').html('999');

$('tr > td').not(':last').each(function (index, valeur) {
  $(this).removeClass();
});
```

Figure 43 : Exemples d'utilisation de jQuery

Dans la première instruction, le sélecteur retourne la table descendante directe (fille) d'un élément ayant un identifiant info, et qui possède un attribut nommé mon-attribut-perso, quelque soit sa valeur. La fonction size() retourne le nombre d'éléments retournés par le sélecteur.

La seconde instruction retourne tous les éléments td qui possèdent la classe impair et qui sont des descendants (directs ou indirects) d'une table. La valeur de chaque élément retourné est ensuite modifiée par 999 via la fonction html().

Enfin, la troisième instruction commence par sélectionner tous les éléments td descendants direct d'un élément tr. La fonction not permet ici de retirer le dernier td des éléments

sélectionnés. On applique ensuite une fonction personnalisée à chaque élément sélectionné. Elle consiste ici à supprimer la classe de l'élément.

D'autre part, jQuery propose une fonction très pratique pour utiliser AJAX. AJAX (pour « **A**synchronous **J**avaScript **A**nd **X**ML », ou « JavaScript et XML asynchrones » en français) est la combinaison des technologies JavaScript, XML et XMLHttpRequest qui permet de communiquer avec le serveur sans recharger la page affichée dans le navigateur. C'est l'objet XMLHttpRequest, qui est un objet JavaScript, qui permet cette communication asynchrone. Le terme asynchrone désigne l'exécution parallèle de plusieurs tâches indépendantes, c'est-à-dire qu'aucune de ces tâches n'attend de résultat provenant des autres tâches.

Voici un exemple de deux tâches synchrones. L'une d'elles, ici la tâche 2, a besoin du résultat de la tâche 1 pour commencer à s'exécuter (figure 44).

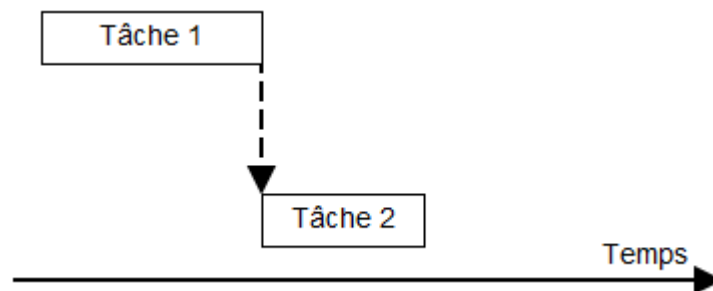


Figure 44 : Exemple de 2 tâches synchrones

Voici maintenant un exemple de deux tâches asynchrones, totalement indépendantes (figure 45) :

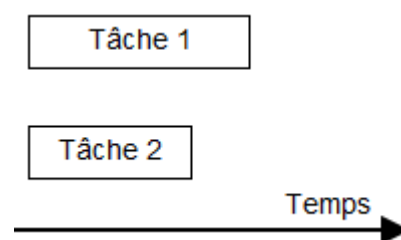


Figure 45 : Exemple de 2 tâches asynchrones

Quant au XML (pour « **eX**tensible **M**arkup **L**anguage », ou « langage de balisage extensible » en français), c'est un langage de balisage qui permet de définir ses propres balises afin de structurer des informations. Le XHTML, successeur de l'HTML, se fonde notamment sur la syntaxe définie par XML.

Voici un exemple de fichier XML (figure 46) :

```
<infos>
  <questionnaires>
    <questionnaire identifiant="1">
      <titre>Perception de la faim</titre>
      <date>12/05/2011</date>
    </questionnaire>
    <questionnaire identifiant="15" fini="oui">
      <titre>Questionnaire sur l'état de santé des patients</titre>
      <date>07/04/2011</date>
    </questionnaire>
    [...]
  </questionnaires>
</infos>
```

Figure 46 : Exemple de données XML

Le XML est utilisé dans AJAX pour la transmission des données entre le serveur et le client (c'est-à-dire le navigateur). Cependant, il est aussi possible que le serveur renvoie un résultat au format HTML, texte ou JSON. JSON (pour « **J**ava**S**cript **O**bject **N**otation », ou « notation objet JavaScript » en français) est un format de données utilisé en JavaScript pour décrire des objets.

La figure 47 montre la façon dont fonctionne AJAX :

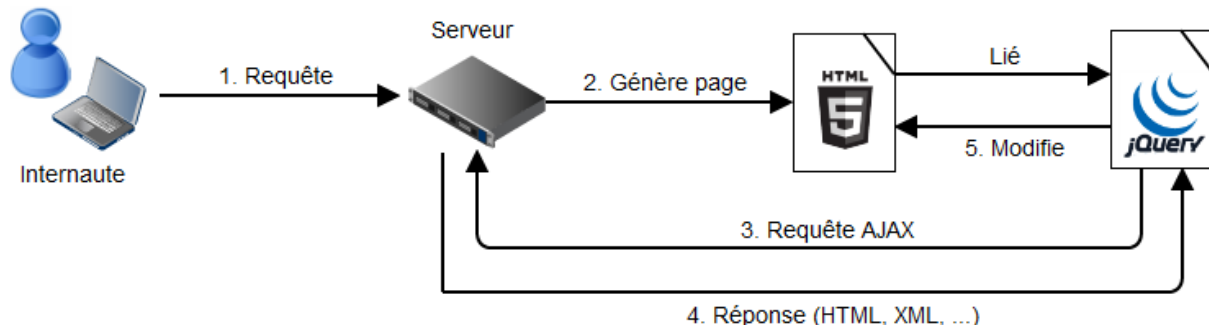


Figure 47 : Fonctionnement d'AJAX

Tout d'abord, l'utilisateur fait une requête pour obtenir le contenu d'une page de manière classique. Le serveur lui renvoie alors la page générée, qui est affichée dans le navigateur. Ensuite, en fonction de certaines actions (un clic, un déclenchement automatique, etc), le code JavaScript réalise un appel AJAX vers le serveur via une URL. Cet appel est asynchrone, ce qui permet à l'utilisateur de pouvoir continuer à interagir avec la page affichée dans son navigateur. Lorsque le serveur renvoie les données demandées via la requête AJAX, le code JavaScript se charge de les traiter. On peut ainsi ajouter des informations sur une page sans avoir à la recharger. Tout est fait de manière invisible pour l'utilisateur.

Je me suis beaucoup servi de ces appels AJAX afin de rendre la navigation plus souple pour l'internaute, notamment dans la page de création des questionnaires.

2.7. Récapitulatif des technologies utilisées

La figure 48 fait apparaître les composants et les technologies utilisés pour la création du site d'ActivCollector, ainsi que leurs interactions.

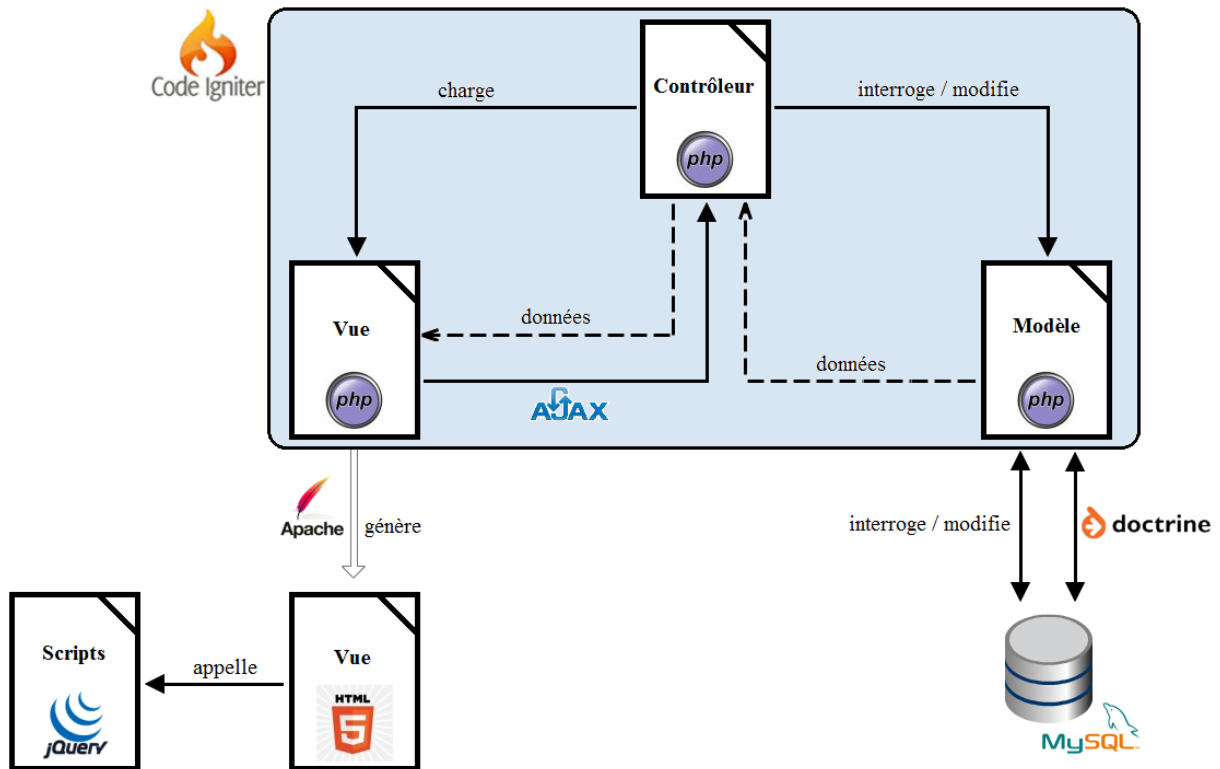


Figure 48 : Vue récapitulative des technologies utilisées

CodeIgniter, qui fonctionne selon une architecture MVC, pose la structure de base du site. Les modèles communiquent via la base de données MySQL soit directement via la bibliothèque de CodeIgniter, soit en passant par l'ORM Doctrine, qui permet de récupérer les tuples de la base de données sous forme d'objets dans le code PHP. Ceci est très pratique lorsqu'il s'agit de gérer les questionnaires, car les associations entre les différentes tables sont très nombreuses. Les requêtes SQL qu'il aurait fallu écrire si je n'avais pas utilisé Doctrine auraient été très compliquées et sûrement moins optimisées.

Les vues HTML sont générées par un serveur Apache, et l'utilisation de JavaScript dans les pages est assurée par le framework jQuery, qui permet de réaliser de belles présentations sans

trop de lignes de code. D'autre part, les vues et les contrôleurs communiquent lorsque des requêtes AJAX sont exécutées.

2.8. Outils utilisés

Cette partie est dédiée aux outils que j'ai utilisés durant mon stage.

2.8.1. Wamp Server

Afin de pouvoir tester le site Internet sur mon ordinateur j'ai installé le logiciel « Wamp Server » [Wamp], qui est une plateforme de développement de type WAMP (pour **W**indows **A**pache **M**ySQL **P**HP). Il existe un autre logiciel, Easy PHP, qui installe ces environnements de la même façon que Wamp Server.

Apache est un serveur HTTP qui génère des pages web qui sont envoyées au navigateur. MySQL est un **S**ystème de **G**estion de **B**ases de **D**onnées **R**elationnelles (SGBDR) qui permet de stocker des données et d'exécuter des requêtes pour en extraire des tuples qui correspondent à un ou plusieurs critères.

Wamp Server installe ainsi tous ces logiciels sur l'ordinateur, ce qui permet de développer et tester des applications web sans être connecté à Internet.

Cette architecture est aussi possible sur les systèmes d'exploitation Linux et Mac. On parle alors respectivement de LAMP et MAMP. Cependant, les logiciels sont à installer séparément les uns des autres : il n'existe pas, à ma connaissance, de logiciel similaire à WAMP Server.

2.8.2. Eclipse et le plugin Aptana

Plusieurs environnements de développement intégrés (IDE) sont disponibles pour PHP, mais peu regroupent toutes les fonctionnalités indispensables, telles que :

- La complétion de code PHP, Javascript / JQuery, CSS, HTML.
- Une coloration syntaxique qui soit un minimum agréable.
- Un accès facile aux documentations PHP et Javascript / JQuery.
- Des indications concernant les erreurs et avertissements dans le code.
- La génération automatique de code (pour les accesseurs et les mutateurs).

Parmi les IDE que j'ai trouvés on peut notamment citer [NetBeans], [PHP Designer], [Zend Studio] et [Komodo]. Je n'en ai cependant trouvé qu'un, après de longues recherches, qui réunisse toutes ces caractéristiques fonctionnelles indispensables.

Il s'agit de « Eclipse for PHP Developers » [Eclipse PHP]. Ce dernier permet de gérer des projets HTML / CSS / PHP facilement, le tout de façon configurable. J'ai ainsi pu déterminer mes propres conventions de codage, comme par exemple placer les accolades à la ligne lors de la déclaration d'une fonction. De plus, Eclipse propose la documentation PHP de la même manière qu'il propose la Javadoc dans sa version Java.

Eclipse ne propose néanmoins pas d'auto complétion pour Javascript ni pour le HTML. J'ai donc dû installer un plugin nommé « Aptana Studio » [Aptana]. Afin d'avoir accès à la documentation et à l'auto complétion pour JQuery, j'ai dû installer un autre plugin, interne à Aptana : « jQuery Support ». La version actuelle du plugin se base sur JQuery 1.4 alors que la version la plus récente est JQuery 1.6, mais elle permet au moins d'avoir une bonne base de travail.

La seule fonctionnalité manquante dans cet IDE est la génération automatique des accesseurs et mutateurs. Il est cependant possible de définir des templates de code qui rendent leur écriture moins répétitive.

2.8.3. BitKinex

BitKinex est, entre autres, un client WebDAV pour Windows. WebDAV (pour « Web-based Distributed Authoring and Versioning ») est un protocole qui permet la gestion de fichiers sur un serveur distant.

J'ai choisi ce logiciel car j'ai mis en place une version de démonstration du site d'ActivCollector sur le serveur afin d'avoir des retours de la part des utilisateurs. Le serveur a été mis en place par Sébastien Cat, qui est responsable informatique à l'INRA, sur le site de Theix.

Avant d'installer ce BitKinex j'ai d'abord essayé une fonctionnalité de Windows, mais les opérations réalisées sur le serveur, comme la création ou la suppression de fichiers, était vraiment très lente. Avec BitKinex ce problème de lenteur est inexistant.

2.8.4. MySQL Workbench

Ce logiciel, développé par Oracle, permet de modéliser facilement une base de données. En effet, la création des tables et des colonnes se fait via une interface graphique.

Il permet aussi de gérer les clés appliquées aux colonnes, comme par exemple les clés primaires et étrangères. D'autre part, il est possible d'exporter une modélisation sous forme d'images ou de script SQL, ce qui est très pratique et évite au développeur un travail pénible.

3. Implémentation et aspects juridiques

3.1. Création d'une bibliothèque dans CodeIgniter

Afin de gérer facilement l'uniformisation des différentes vues du site, j'ai créé une bibliothèque dans CodeIgniter.

L'idée est d'avoir une structure HTML de base identique pour chaque page du site, ce qui évite de la copier-coller dans toutes les vues. Ainsi, si une modification doit être faite il n'y aura qu'une seule page à modifier.

Cette structure peut être représentée comme sur la figure 49 :

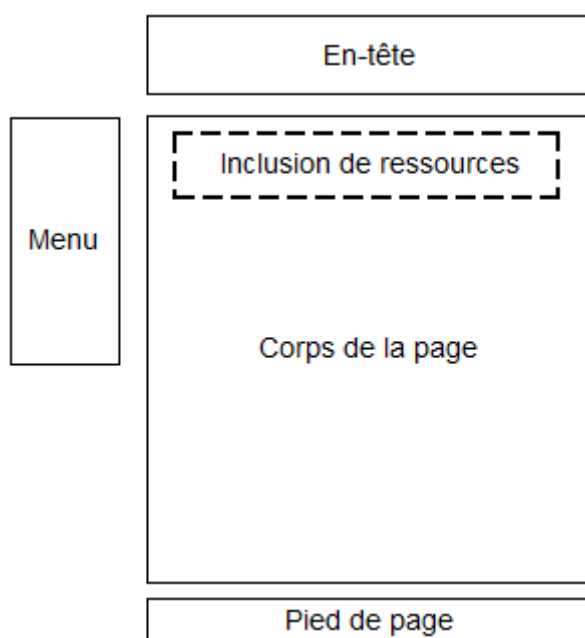


Figure 49 : Structure de base des pages du site

L'en-tête et le pied de page peuvent être considérés comme statiques, mais le menu change selon le profil de l'utilisateur (visiteur, volontaire, chercheur, etc). De même, le corps de page et les ressources à inclure sont spécifiques à chaque page.

Parmi les différentes ressources que l'on peut avoir à inclure, il y a :

- Les feuilles de style CSS, qui permettent de donner un style visuel aux éléments de la page. Certaines feuilles peuvent être destinées à être affichées dans un navigateur Internet alors que d'autres permettent de donner un style différent lorsque l'utilisateur souhaite imprimer la page.

- Les scripts JavaScript, qui permettent de changer dynamiquement le contenu de la page sans la recharger.

Ces deux types de ressources se trouvent dans des fichiers externes ou bien directement dans la page.

J'ai donc créé une bibliothèque qui peut gérer tous les types de ressources et qui est la plus pratique possible à utiliser. La figure 50 présente un exemple de génération d'une vue à partir d'un contrôleur, avec ma bibliothèque :

```
$this->layout->loadBasicSettings('Questionnaires', Layout::MENU_VISITOR);

$this->layout->addJsFile('jquery.alerts');
$this->layout->addCssFile('questionnaires');
$this->layout->addCssScript(".questions {min-height: 20px;}");

$this->layout->loadView('questionnaires/createQuestionnaire', $data);
$this->layout->generate();
```

Figure 50 : Génération d'un vue

La première méthode initialise les informations élémentaires de la page, à savoir son titre et le menu à afficher. Le menu est ici défini sous forme de constante (figure 51). Cette dernière correspond à une page PHP à inclure :

```
const MENU_VISITOR = 'menus/visitor.php';
const MENU_VOLUNTEER = 'menus/volunteer.php';
const MENU_RESEARCHER = 'menus/researcher.php';
const MENU_PROJECT_MANAGER = 'menus/manager.php';
const NO_MENU = -1;
```

Figure 51 : Définition des constantes pour les menus

J'appelle ensuite les méthodes qui permettent d'inclure des fichiers et des scripts CSS et JavaScript, puis enfin je définis le fichier PHP qui contient la vue qui constituera le corps de la page.

La méthode `generate()` fait appel à une méthode implémentée par CodeIgniter. C'est elle qui génère véritablement la page. Toutes les autres méthodes ne font que remplir la variable `data` (figure 52) :

```
$this->CI->load->view('layout.php', $this->data);
```

Figure 52 : Génération de la page

Le fichier `layout.php` a lui pour but de délimiter les grandes structures de la page (menu, entête, etc) et d'ajouter le contenu de la variable `data` aux bons endroits.

3.2. Informatisation des questionnaires

J'ai commencé la création du site par la partie qui concerne les questionnaires car ils constituent une des grandes briques du site Internet. Les chercheurs devront pouvoir créer des questionnaires et les envoyer aux volontaires participant à un projet.

3.2.1. Les différents types de questions rencontrés

3.2.1.1. Les questions ouvertes

Une question ouverte permet à l'utilisateur de saisir le texte de son choix pour répondre à la question.

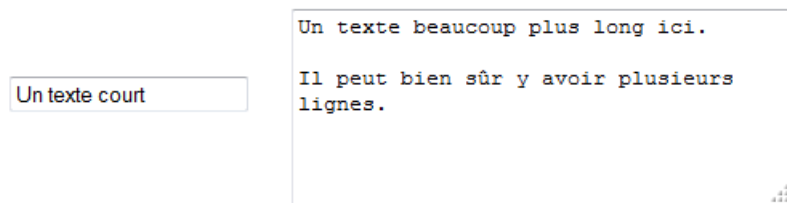
On peut distinguer plusieurs utilisations de ce type de questions. La première consiste à demander une information courte (figure 53), telle que la pratique d'un régime particulier (végétarien, végétalien, etc). On peut cependant vouloir permettre au volontaire de saisir un texte plus long, en lui demandant par exemple pourquoi il a participé au protocole en cours.

Est-ce que vous avez adopté un régime alimentaire particulier ?

Exemple de réponse : En principe je ne mange pas de viande rouge.

Figure 53 : Question ouverte extraite d'un questionnaire papier

Ces utilisations sont représentées différemment en informatique. Les textes courts et les textes plus longs sont saisis respectivement dans les champs suivants représentés sur la figure 54 :



Un texte court

Un texte beaucoup plus long ici.
Il peut bien sûr y avoir plusieurs lignes.

Figure 54 : Éléments utilisés pour la saisie de texte

Dans certains cas il est souhaitable de n'autoriser l'utilisateur qu'à entrer une valeur numérique dans un champ de texte court. Ceci est faisable grâce à JavaScript. J'ai donc ajouté un troisième type de réponse pour prendre en compte ce cas.

3.2.1.2. Un choix parmi N

Ce type de questions permet de choisir une réponse parmi N proposées. Dans les questionnaires que j'ai étudiés, ce type de question est très souvent représenté par des échelles de catégories. Parmi ces échelles on distingue trois types de représentation.

La première consiste à n'attribuer des libellés qu'aux bornes (figure 55) :

Avec quel degré percevez-vous la faim ?						
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
pas du tout faim						extrêmement faim

Figure 55 : Echelle à libellés extrêmes extraite d'un questionnaire papier

La seconde fait apparaître un libellé pour chaque choix (figure 56) :

En général, diriez-vous que votre santé est :					
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Excellente	Très bonne	Bonne	Satisfaisante	Mauvaise	

Figure 56 : Echelle avec un libellé par choix extraite d'un questionnaire papier

La dernière est identique dans le sens où il y a un label par choix. Cependant, la légende de ces libellés n'est pas indiquée pour chaque question mais pour un groupe de questions (figure 57) :

Si vous ne ressentez pas du tout l'état mentionné, cochez la case 0.							
Si vous ressentez un peu cet état, cochez la case 1.....							
Si vous ressentez tout à fait cet état, cochez la case 6.							
	0	1	2	3	4	5	6
1 - Dynamique	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2 - Heureux	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3 - Triste	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
...							

Figure 57 : Echelle de valeurs extraite d'un questionnaire papier

Si on prend comme exemple la dernière question, on aurait plusieurs questions telles que celle de la figure 58 :

Si vous ne ressentez pas du tout de tristesse, cochez la case 0.
Si vous ressentez un peu de tristesse, cochez la case 1.....
Si vous ressentez tout à fait cette tristesse, cochez la case 6.

0 1 2 3 4 5 6

Figure 58 : Représentation d'un choix unique parmi N possibilités

Seul un état à la fois peut être sélectionné, et on peut attribuer à chaque bouton un libellé (visible par l'utilisateur) ainsi qu'une valeur réelle (qui sera prise en compte lors du dépouillement). Cette valeur réelle ne peut être qu'un nombre.

3.2.1.3. P choix parmi N

Je n'ai pas vu ce type de questions dans les questionnaires qui m'ont été fournis, mais j'ai pensé qu'il serait utile de l'ajouter aux autres types car il pourra servir un jour. De plus, comme nous le verrons plus loin je m'en servirai pour simplifier la représentation de certaines questions.

On pourrait imaginer une question telle que celle de la figure 59 :

Parmi les sports suivants, lesquels pratiquez-vous ?			
<input type="checkbox"/> Aérobie	<input type="checkbox"/> Cheval/Equitation	<input type="checkbox"/> Hockey	<input type="checkbox"/> Natation plaisir
<input type="checkbox"/> Roller-skate	<input type="checkbox"/> Tennis	<input type="checkbox"/> Badminton	...

Figure 59 : Exemple de question à P choix parmi N

Le volontaire peut cocher autant de cases qu'il le souhaite, ou ne pas en cocher du tout. La représentation informatique est très similaire à la représentation papier (figure 60) :

Parmi les sports suivants, lesquels pratiquez-vous ?

- Aérobie
- Cheval/Equitation
- Hockey
- Natation plaisir
- Roller-skate
- Tennis
- Badminton
- ...

Figure 60 : Représentation d'une question à P réponses parmi N possibilités

3.2.1.4. Les échelles continues

Il arrive parfois que le chercheur demande au volontaire de donner une évaluation sur un critère au moyen d'une échelle continue. Le volontaire doit alors tracer un trait sur l'échelle pour représenter sa réponse. Les chercheurs calculent ensuite où se situe de trait du volontaire par rapport aux extrémités (figure 61).

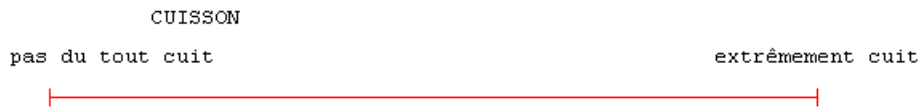


Figure 61 : Echelle continue extraite d'un questionnaire papier

Il est possible de représenter ce type de question à l'écran de la façon suivante, grâce au plugin [jQuery UI Slider] (figure 62).



Figure 62 : Représentation d'une échelle continue

Il faut pour cela définir 2 choix de réponse qui permettront de définir les libellés affichés aux extrémités de la barre ainsi que les valeurs qui leurs sont associées.

Sur l'exemple ci-dessus, les deux libellés seraient donc « pas du tout cuit » et « extrêmement cuit », associés respectivement à des valeurs telles que 0 et 100.

Par contre, toutes les échelles de ce type auront la même taille de pas. Il a ainsi été décidé avec Sylvie Rousset que le la taille de pas sera de 1%. Ainsi, si on a une échelle qui va de 0 à 10, la barre se déplacera par pas de 0,1.

3.2.1.5. Classement de réponses

Cas général

Certaines questions consistent à demander au volontaire de trier plusieurs réponses. Ce type de question est différent des autres car on ne doit pas seulement retenir la réponse donnée (ici le rang), mais aussi la réponse qui lui est associée. De plus, cela nous oblige à enregistrer autant de réponses qu'il y a d'éléments à trier.

Voici un exemple d'une question de ce type (figure 63) :

Classez ces viandes de la plus appréciée (1) à la moins appréciée (4) :	
- porc	<input type="checkbox"/>
- bœuf	<input type="checkbox"/>
- poulet	<input type="checkbox"/>
- agneau	<input type="checkbox"/>

Figure 63 : Question de tri extraite d'un questionnaire papier

J'ai pensé à une représentation basée sur le plugin de tri de jQuery UI : Sortable [jQuery UI Sortable]. Les éléments sont triables à la souris : lorsqu'on bouge la souris tout en gardant le bouton gauche enfoncé, on peut le faire bouger et donc modifier l'ordre des éléments (figure 64).

Classez ces viandes de la plus appréciée (1) à la moins appréciée (4) :

- Porc
- Boeuf
- Poulet
- Agneau

Figure 64 : Représentation d'une question de classement, dont l'élément "Poulet" est en cours de déplacement

Néanmoins, ce type de représentation impose que lors du dépouillement, l'élément le plus haut dans la liste ait toujours la valeur 1 et que le plus bas ait la valeur N (avec N le nombre d'éléments).

Cas particulier

Il existe un cas particulier pour ce type de questions. Il peut en effet y avoir deux questions en une dans les questionnaires papiers (figure 65) :

- Dans la première il est demandé de cocher des choix parmi ceux proposés.
- Dans la seconde il doit recopier ceux sélectionnés en les triant.

Quel type de viande mangez-vous ?

- | | |
|---------------------------------|--|
| <input type="checkbox"/> Bœuf | <input type="checkbox"/> Veau |
| <input type="checkbox"/> Porc | <input type="checkbox"/> Agneau |
| <input type="checkbox"/> Mouton | <input type="checkbox"/> Volaille (dinde/poulet) |
| <input type="checkbox"/> Cheval | <input type="checkbox"/> Lapin |

Et parmi ces viandes, quelles sont les 4 viandes que vous consommez le plus souvent ? Veuillez les citer dans l'ordre de fréquence de consommation (du plus fréquent au moins fréquent).

1. _____
2. _____
3. _____
4. _____

Figure 65 : Cas particulier d'un test de classement, extrait d'un questionnaire papier

Pour saisir cette question sur le site, il faudrait la décomposer en deux questions. La première consiste à proposer P choix parmi N (figure 66). Le volontaire peut bien sûr ne cocher aucun choix (s'il est végétarien par exemple).

Quel type de viande mangez-vous ?

- | | |
|---|--|
| <input checked="" type="checkbox"/> Boeuf | <input type="checkbox"/> Veau |
| <input checked="" type="checkbox"/> Porc | <input checked="" type="checkbox"/> Agneau |
| <input type="checkbox"/> Mouton | <input type="checkbox"/> Volaille (dinde/poulet) |
| <input type="checkbox"/> Cheval | <input type="checkbox"/> Lapin |

Figure 66 : Première question du cas particulier : le choix

La seconde question consiste à classer tous les choix cochés dans la première question par ordre décroissant. Le volontaire peut alors ordonner ses choix par glisser-déposer à la souris (figure 67) :

Veuillez trier ces viandes dans l'ordre de fréquence de consommation (du plus fréquent au moins fréquent).

- Agneau
- Porc
- Boeuf

Figure 67 : Seconde question du cas particulier : le classement

Bien sûr, si aucun élément n'a été sélectionné dans la première question, la deuxième ne doit pas être affichée.

Etant donné qu'on ne peut pas limiter le nombre d'élément à classer (ici 4 dans la question originale du questionnaire papier), nous avons décidé de laisser le volontaire trier tous les éléments qu'il aura sélectionnés.

3.2.1.6. Les questions tabulaires

Les questions organisées sous forme de tableau dans les questionnaires papier sont les plus difficiles à traiter. En effet, elles regroupent une multitude de questions en une.

Cas où une partie des données est présentée

Le premier cas est celui où le volontaire doit remplir un tableau en fonction d'éléments qu'il choisit dans un tableau.

La figure 68 est représentative de ce type de questions :

Question 1 - Parmi les activités listées ci-dessous, écrivez, en clair dans le tableau en bas de page celles que vous avez pratiquées au moins 10 fois au cours des 12 derniers mois. Si vous avez pratiqué une activité absente de la liste, au moins 10 fois au cours des 12 derniers mois, notez également le nom de cette activité dans le tableau. Ne comptez pas le temps passé à marcher ou à faire du vélo pour vous rendre à votre travail, il sera comptabilisé dans la section " activité professionnelle et domestique ".

Précisez la fréquence de pratique de chacune des activités reportées dans le tableau.

Aérobic	Cheval/Equitation	Hockey	Natation plaisir	Roller-skate	Tennis
Badminton	Course cross	Jardinage	Natation compét.	Rugby	Vélo plaisir
Basket-ball	Danse	Jogging	Patinage plaisir	Skateboard	Vélo vitesse
Bowling	Football	Karaté judo	Patinage compétit.	Ski alpin	Planche à voile
Boxe	Frisbee	Kayak	Pêche	Ski de fond	Voile
Canoë plaisir	Golf	Marche plaisir	Ping pong/Tennis de table	Ski nautique	VTT
Canoë compétition	Gymnastique	Marche rapide	Plongée sous-marine	Squash	...autre
Chasse	Hand-ball	Moto cross	Randonnée montagne	Surf	...autre

Exemple : vous avez fait 30 à 40 mn de jogging pendant 6 mois, de 9 à 10 fois par mois

Nom de l'activité	Nombre de mois	Nombre moyen de fois/mois	Temps moyen passé à chaque fois (en minutes)
Ex. : Jogging	6	9,5	35

Figure 68 : Exemple de question tabulaire, extraite d'un questionnaire papier

Ici, pour chaque activité choisie par le volontaire, il doit répondre à 3 questions :

- Le nombre de mois pendant lesquels il a pratiqué l'activité.
- Le nombre moyen de fois par mois qu'il pratique l'activité.
- Le temps moyen passé à chaque fois, en minutes.

Etant donné qu'on ne pourra pas présenter les informations sous cette forme à l'écran, j'ai proposé qu'on définisse 4 questions :

- La première est une question à P choix parmi N possibilités, où le volontaire choisit toutes les activités qu'il pratique.
- Les 3 autres sont les questions qui correspondent aux autres colonnes du tableau, qui se répètent chacune autant de fois qu'il y a de réponses choisies dans la première question.

Par contre, l'informatisation du questionnaire ne permet pas d'ajouter un sport qui serait absent de la liste. Une solution consisterait à proposer une liste d'éléments (ici des sports) qui soit assez complète, de sorte que le volontaire puisse choisir ses activités le mieux possible.

On aurait donc un enchaînement de questions qui ressemblerait aux figures ci-dessous, si on choisit 2 activités (Jogging et Tennis) en réponse à la première question :

Parmi les activités listées ci-dessous, choisissez celles que vous avez pratiquées au moins 10 fois au cours des 12 derniers mois. Ne comptez pas le temps passé à marcher ou à faire du vélo pour vous rendre à votre travail, il sera comptabilisé dans la section "activité professionnelle et domestique".

- Aérobie Cheval/Equitation Hockey Natation plaisir Roller-skate Tennis
 Badminton Course cross Jardinage Natation compét. Rugby Vélo plaisir
 Basket-ball Danse Jogging Patinage plaisir Skateboard Vélo vitesse

.....
Figure 69 : Question tabulaire, question 1

Pendant combien de mois avez-vous pratiqué cette activité au cours des 12 derniers mois ? Jogging
Votre réponse : <input type="text" value="6"/>

Figure 70 : Question tabulaire, question 2

Pendant combien de mois avez-vous participé à cette activité au cours des 12 derniers mois ? Tennis
Votre réponse : <input type="text" value="2"/>

Figure 71 : Question tabulaire, question 3

Combien de fois par mois en moyenne avez-vous pratiqué cette activité ? Jogging
Votre réponse : <input type="text" value="4"/>

Figure 72 : Question tabulaire, question 4

Combien de fois par mois en moyenne avez-vous pratiqué cette activité ? Tennis

Votre réponse :

Figure 73 : Question tabulaire, question 5

Quel est le temps moyen que vous avez passé à chaque fois sur cette activité (en minutes) ? Jogging

Votre réponse :

Figure 74 : Question tabulaire, question 6

Quel est le temps moyen que vous avez passé à chaque fois sur cette activité (en minutes) ? Tennis

Votre réponse :

Figure 75 : Question tabulaire, question 7

Cas où toutes les données sont à fournir

Il existe aussi une variante de ce premier cas. En effet, certains tableaux sont à remplir intégralement par le candidat, il n'y a pas de liste d'éléments.

Voici un exemple d'un tel tableau (figure 76) :

Question 8 – Activité(s) Professionnelle(s)

Si, pendant les 12 derniers mois, vous étiez retraité, au foyer, au chômage, en formation ou étudiant, en invalidité ou en arrêt de travail, ne remplissez que les colonnes “ profession ” et “ code”, et passez à la question 9.

Listez dans le tableau ci-dessous **les activités professionnelles** que vous avez **exercées pendant plus d'un mois durant les 12 derniers mois**. Ce tableau ne concerne que la(les) période(s) pendant la(les)quelle(s) vous avez eu une activité professionnelle.

Reportez-vous aux tableaux en bas de page pour remplir les colonnes “ Code ” et “ Travail pratiqué non assis ”.

Exemple : vous avez été secrétaire pendant 6 mois, puis vous avez eu 2 mois de chômage, et depuis 3 mois et demi vous êtes représentante.

ACTIVITE PROFESSIONNELLE			TEMPS DE TRAJET	TEMPS MOYEN DE TRAVAIL		TRAVAIL PRATIQUE ASSIS	TRAVAIL PRATIQUE NON ASSIS		
PROFESSION	CODE voir “ tableau des codes ” N°	DUREE au cours des 12 derniers mois Nombre de mois	MARCHE OU VELO (aller et retour) Minutes/jour	Jours/sem	Heures/jour	Heures/jour	Cochez la catégorie qui décrit le mieux votre travail (une seule réponse)		
							A	B	C
Ex. : Secrétaire	1	6	5	5	7	6,5	X		
Ex. : Représentante	3	3,5	0	4	8,75	2		X	

Tableau des codes	
1. Travail dans un bureau	4. Etudiant
2. Travail à l'extérieur d'un bureau	5. Au foyer
3. Travail dans un bureau et à l'extérieur	6. Retraité
	7. Arrêt de travail, invalidité,
	8. Chômage

Tableau des catégories de travail		
Catégorie A	Catégorie B	Catégorie C
Debout sans charges lourdes Ménage léger (repasser, cuisiner, laver, dépoussiérer) Marche lente avec des arrêts	Port de charges légères Marche régulière Ménage intensif (passer la serpillière, balayer, récupérer, gratter) Jardinage (planter, désherber) Peinture, plâtrage, plomberie, soudure, électricité	Port de charges moyennes à lourdes Travail en plein air (chantier, agriculture, construction, bûcheron)

Figure 76 : Question tabulaire sans proposition d'éléments, extrait d'un questionnaire papier

Là encore, la solution consiste à proposer une liste complète d'éléments pour revenir au premier cas. Ici, il faut donc fournir une liste aussi complète que possible des métiers.

Ainsi, on peut ensuite définir 8 questions pour représenter ce tableau :

- La première est une question à P choix parmi N possibilités, où le volontaire choisit tous les métiers qu'il a pratiqués.

- Les 7 autres sont les questions qui correspondent aux autres colonnes du tableau, qui se répètent chacune autant de fois qu'il y a de réponse choisies dans la première question.

3.2.1.7. Les questions associées à une image

J'ai remarqué, parmi les questionnaires qui m'ont été prêtés, que certaines questions nécessitent la visualisation d'une image pour pouvoir répondre. Cette image contient différents éléments identifiés par une lettre ou un chiffre, et le volontaire doit choisir un de ces éléments (figure 77).

Regardez la photo ci-dessous : quand vous mangez des légumes en général, quelle quantité moyenne en mangez-vous ? Cochez la lettre qui correspond à la portion de légumes que vous mangez en moyenne (cochez une seule case) : A B C D E F G

A **B** **C** **D** **E** **F** **G**

Figure 77 : Question associée à une image

Il faut donc que toute question puisse contenir, de façon facultative, une image.

3.2.2. Un outil déjà existant : LimeSurvey

Après avoir commencé à implémenter les questionnaires, Sébastien Cat (informaticien sur le site de Theix) m'a fait découvrir un outil disponible sur Internet (sous licence GPL) qui gère des questionnaires : [LimeSurvey]. Il se trouve que cet outil possède peu ou prou les mêmes possibilités que ce que mon cahier des charges prévoit, à savoir :

- Des questionnaires multilingues.
- Des types de questions assez complets.
- L'intégration d'images dans un questionnaire.

- La possibilité d'imprimer les questionnaires.
- La possibilité de définir des conditions sur les questions dépendant des réponses précédentes.
- La possibilité pour un volontaire d'enregistrer ses réponses pour continuer de remplir le questionnaire plus tard.

Il n'est cependant pas dit qu'utiliser cet outil comme base de travail m'aurait fait gagner du temps. En effet, les questions tabulaires, les échelles continues et les classifications d'éléments provenant des réponses à une autre question ne sont pas implémentés. De même, il semble que les questions peuvent être affichées dans un ordre prédéfini, mais pas dans un ordre aléatoire.

D'autre part, LimeSurvey permet de compiler les résultats des questions, mais pas de définir des pré traitements.

Enfin, la gestion du multilinguisme ne serait pas uniformisée à l'échelle du site si j'avais utilisé LimeSurvey.

J'en ai profité pour regarder la structure de leur base de données et il se trouve que leur architecture est quasiment la même que la mienne.

3.2.3. Modélisation de la base de données

Un questionnaire est une entité qui doit être réutilisable dans n'importe quel autre projet. Il est composé d'une ou plusieurs parties (table `section`) qui sont elles-mêmes composées d'une ou plusieurs questions. Les questions ont leur propre table, `question`, et sont liées à un type de question (table `question_type`) et éventuellement à des choix de réponse (`answer`). Ces éléments de réponse possèdent un libellé (facultatif) et une valeur qui sera utilisée lors du dépouillement.

Les questions devant pouvoir être utilisées dans n'importe quel questionnaire, le lien entre une question et une partie d'un questionnaire se fait via la table `question_selected`.

Comme une question doit être liée à une ou plusieurs conditions, cette table comporte plusieurs colonnes dédiées aux conditions :

- `condition_gender`, `condition_min_age`, `condition_min_bmi`, etc : ces colonnes permettent de fixer des conditions sur trois des caractéristiques d'un volontaire : son sexe, son âge et son indice de masse corporelle (IMC).
- `condition_on_question` et `condition_answer` : ces colonnes servent à lier la question avec la réponse à une question précédente. Ainsi, la question n'est posée au volontaire que s'il a répondu à la question ciblée par la réponse ciblée.

Comme il ne peut y avoir qu'une seule condition sur une question, je n'ai pas créé de table spécifique pour les conditions et j'ai préféré les mettre dans `question_selected`.

D'autre part, comme décrit dans l'analyse des questionnaires, il a fallu mettre en place des questions dont les différentes réponses possibles sont les réponses choisies par le volontaire à une question précédente. J'ai pour cela ajouté une colonne `link_to_question_answers` à la table `question_selected`. La question doit donc toujours être affichée après la question ciblée.

Enfin, une autre particularité des questionnaires est qu'une question doit pouvoir être posée autant de fois qu'il y a eu de réponses sélectionnées à une question précédente (pour les questions tabulaires). J'ai donc mis en place la colonne `loop_on_question_answers` qui cible la question sur laquelle « boucler ».

Ensuite, chaque volontaire doit posséder sa propre instance d'un questionnaire lorsqu'un chef de projet lui envoie un questionnaire à remplir. Ceci est matérialisé par la table `questionnaire_completed`. C'est elle qui fait le lien entre le questionnaire, le volontaire et les réponses de ce dernier (table `question_answered`). De plus, elle contient quatre dates :

- La date à laquelle le questionnaire sera envoyé au volontaire. Le chef de projet pourra donc prévoir l'envoi de questionnaires pendant une période où il ne sera pas présent.
- La date à laquelle le questionnaire se termine. Cette date est néanmoins seulement théorique car les chercheurs veulent recevoir les réponses des volontaires même si ceux-ci le renvoient leur questionnaire après la date limite.
- La date à laquelle le volontaire a commencé à répondre au questionnaire.

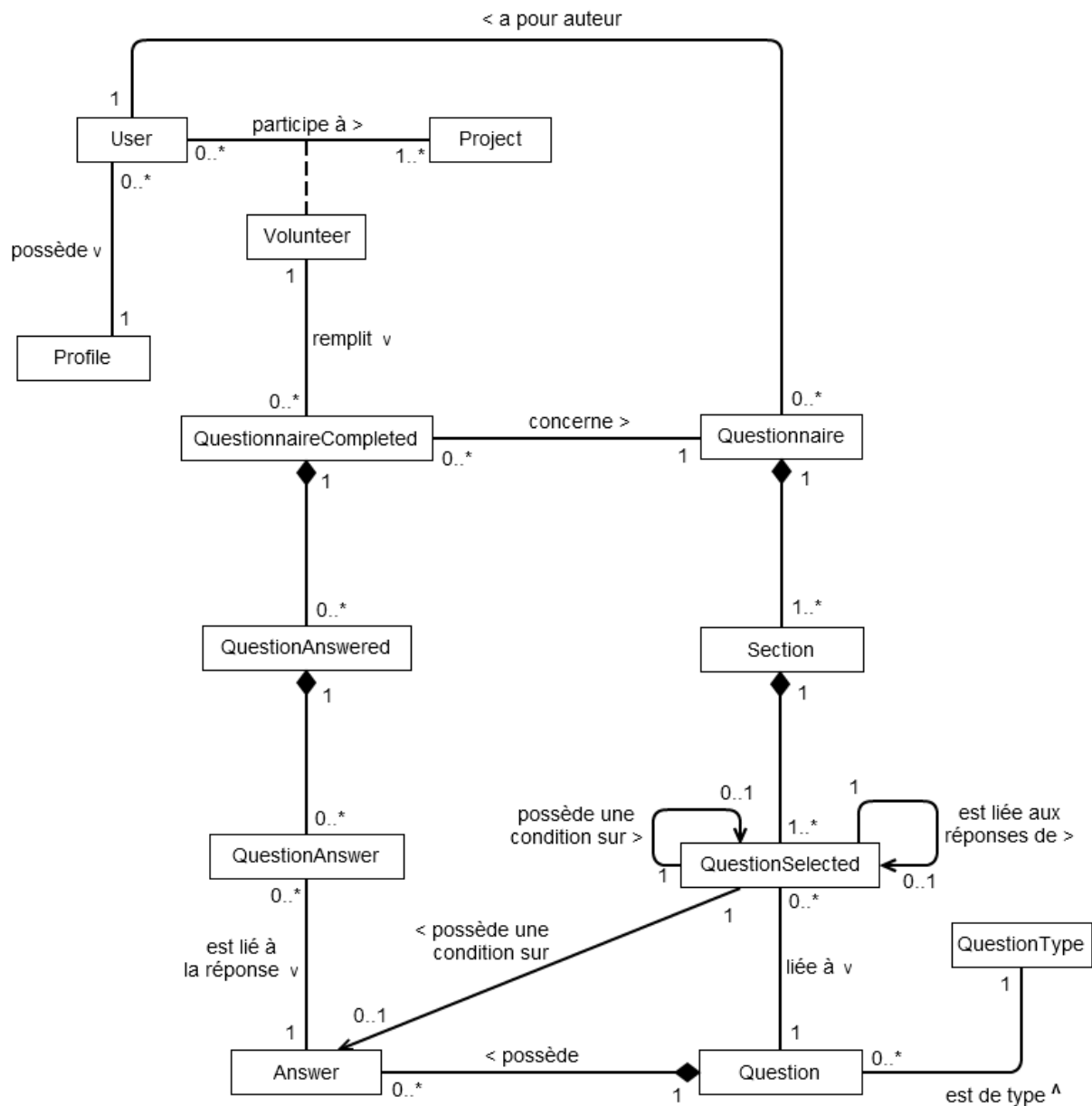
- La date à laquelle le volontaire a fini de répondre au questionnaire. C'est la présence de cette date qui me permettra, dans l'application, de savoir si le questionnaire est prêt à être dépouillé et s'il peut être analysé.

Chaque question à laquelle le volontaire répond fait l'objet de la création d'un tuple dans la table `question_answered`. On y stocke bien entendu la question ciblée (un tuple de la table `question_selected`), mais aussi la date de la réponse et le numéro d'ordre de la question. En effet, comme les questions sont affichées une par une aux volontaires et qu'elles peuvent être dans un ordre aléatoire, les chercheurs veulent savoir dans quel ordre les volontaires ont reçu les questions.

Les réponses en elles-mêmes sont stockées dans la table `question_answer`. Dans le cas d'une question ouverte, où il n'y a aucune cotation proposée, la réponse du volontaire est stockée dans la colonne `text_answer`. Pour les autres types de questions, un lien est fait avec une des réponses à la question via la colonne `id_chosen_answer`.

Dans le cas où le volontaire ne répond pas une question (si celle-ci ne nécessite pas une réponse obligatoire), j'enregistre une instance de `question_answered`, mais sans la lier à des `question_answer`. Cela me permet de savoir que la question a déjà été posée lorsque je recherche une nouvelle question à poser au volontaire.

La figure 78 est le diagramme de classes UML de la partie questionnaires de l'application, et la figure 79 est le MLD correspondant. J'ai aussi fait figurer les classes et les tables liées aux volontaires et aux projets car elles ont des liens forts avec les questionnaires :



Answer

Classe "Answer"

A \blacklozenge $\xrightarrow{0..*}$ B

A est composé de 0 à n B, et la durée de vie des objets de type B est liée à celle de l'objet A qui les contient

A — B

Association entre A et B

Figure 78 : Diagramme de classes UML des questionnaires

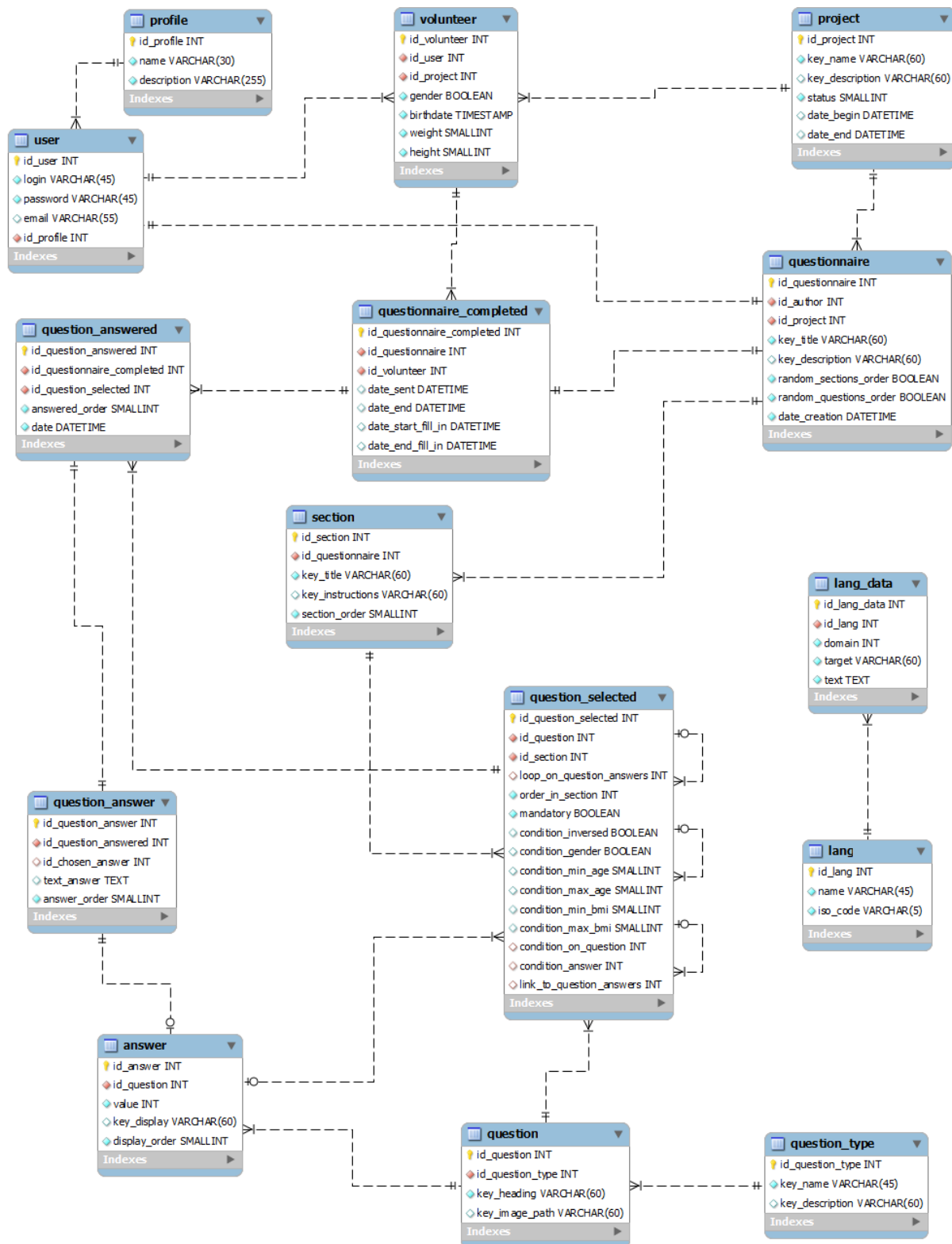


Figure 79 : MLD de la partie de la base de données qui concerne les questionnaires

Comme les questionnaires doivent être multilingues et que des langues peuvent être ajoutées ou supprimées à n'importe quel moment, il n'est pas possible de stocker les libellés des questions, les libellés des réponses et autres textes directement dans ces tables. J'ai donc opté pour une solution qui m'a été proposée par Libo Ren, qui consiste à créer deux tables lang et

`lang_data`, et à mettre dans les tables liées aux questionnaires des clés qui identifient de manière unique un tuple. Par exemple, le libellé de la question d'identifiant 34 a pour clé de titre la valeur `question_heading_34`.

La table `lang` contient le code ISO et le nom de toutes les langues (par exemple `fr_fr`, Français). La seconde table, `lang_data`, permet quant à elle de faire le lien entre une clé et une langue. Etant donné que les questionnaires ne seront pas les seules données dynamiques à être internationalisées, la table `lang_data` contient aussi une clé numérique `domain` qui identifie le domaine de la clé principale. Par exemple, la clé de domaine des questionnaires est 1.

Langues		
id	Code ISO	Nom
1	fr_fr	Français
2	en_uk	English

Données			
id_langue	Clé de domaine	Clé texte	Texte
1	1	une_clef	Un premier texte
2	1	une_clef	A first text
1	2	une_clef	Un second texte

Figure 80 : Exemples d'internationalisation de textes

Sur l'exemple de la figure 80, une clé texte est identique pour deux domaines, et la clé du domaine 1 est traduite dans deux langues.

3.2.4. Mapping objet-relationnel avec Doctrine

Afin de pouvoir utiliser Doctrine, il faut tout d'abord créer des objets correspondant aux tables de la base de données. Il faut ensuite faire le mapping, c'est-à-dire la correspondance, entre les attributs et les colonnes.

Trois solutions sont proposées par Doctrine :

- Créer un fichier XML qui regroupe les mappings des classes de l'application. Cette solution s'inspire de celle proposée par Hibernate.
- Créer des fichiers YAML (pour « **Y**AML Ain't Markup **L**anguage », ou « **Y**AML n'est pas un langage de balisage » en français). Seule la syntaxe diffère du XML.

- Ajouter des annotations dans les blocs de commentaires des classes. Cette solution s'inspire de JPA.

J'ai pour ma part choisi les annotations car je trouve ce système très propre, même s'il ne centralise pas les informations de mapping, contrairement à la solution XML.

Tout d'abord, il faut marquer la classe comme étant une entité (figure 81):

```
/**
 * @Table(name="question_selected")
 * @Entity
 */
class QuestionSelected
{
    [...]
}
```

Figure 81 : Mapping d'une table

Si la classe n'a pas le même nom que la table, comme c'est le cas ici (le nom de la table comporte un underscore), il faut alors ajouter une annotation `@Table` qui prend un attribut `name`.

La correspondance entre attributs et colonnes se fait ensuite via une annotation `@Column` (figure 82) :

```
/**
 * @Column(name="condition_inversed", type="boolean")
 */
private $conditionInversed;
```

Figure 82 : Mapping d'une colonne

Là encore un attribut `name` est disponible, ainsi qu'un attribut `type`. Ce dernier permet de préciser à Doctrine quel mode de conversion il doit appliquer lorsqu'il extrait les données de la base.

Lorsque l'on ne souhaite pas simplement mapper (faire correspondre) une colonne à un attribut mais une association entre deux objets, plusieurs types d'associations sont autorisées :

- One-To-One (« un vers un ») : représente un pointeur d'un objet vers un autre. Trois types de relations sont possibles (figure 83) :
 - Unidirectionnel : seul un des objets connaît l'autre.
 - Bidirectionnel : les deux objets se connaissent mutuellement.

- Autoréférencé : dans le cas où un objet pointe vers un objet du même type (de la même classe).

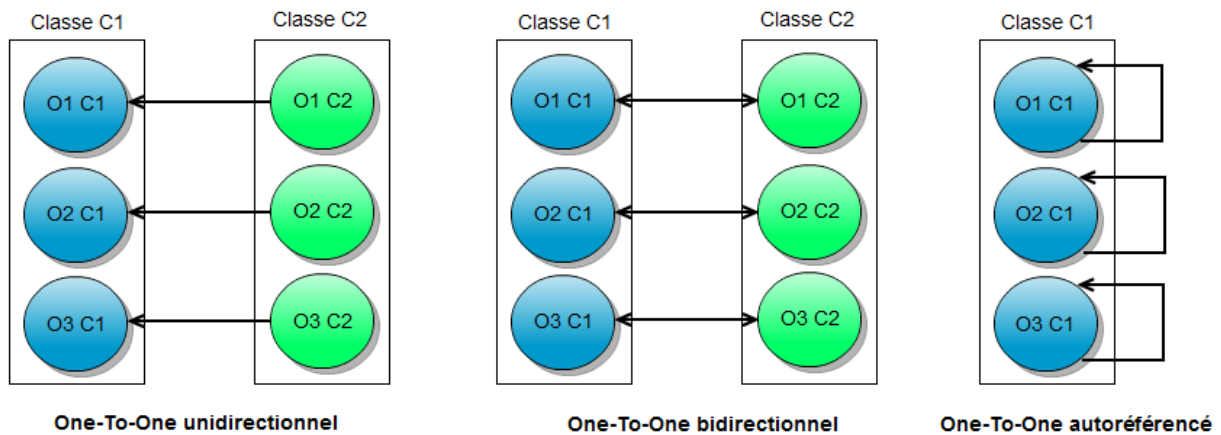


Figure 83 : Différentes sortes de mapping One-To-One

- One-To-Many (« un vers plusieurs », figure 84) : un objet pointe vers plusieurs objets. Exemples d'utilisation : une adresse peut pointer vers plusieurs personnes (si plusieurs d'entre elles vivent dans le même foyer par exemple).

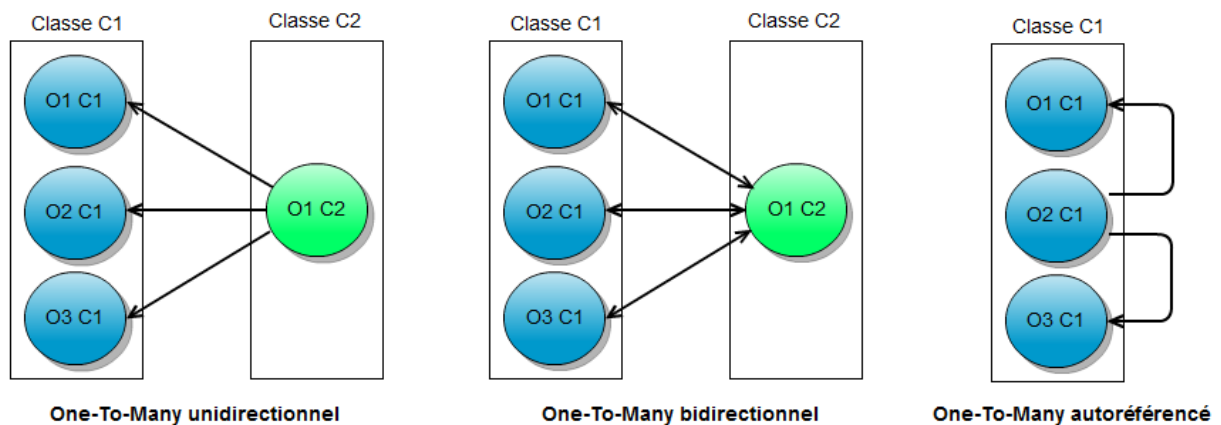


Figure 84 : Différentes sortes de mapping One-To-Many

- Many-To-One (« plusieurs vers un », figure 85) : plusieurs objets pointent sur un même objet. Exemple d'utilisation : plusieurs numéros de téléphone (fax, domicile, etc) pointent vers un même utilisateur.

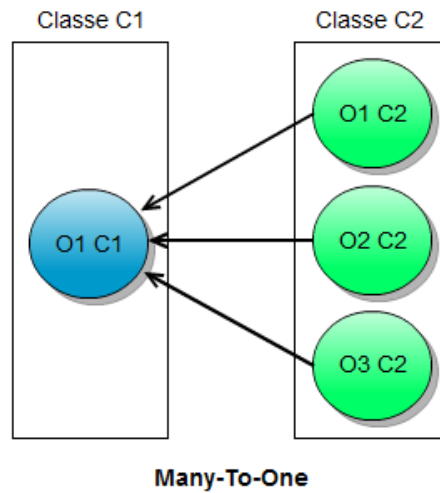


Figure 85 : Mapping Many-To-One

- Many-To-Many (« plusieurs vers plusieurs », figure 86) : un objet peut pointer vers plusieurs autres objets, et un objet peut avoir plusieurs pointeurs.

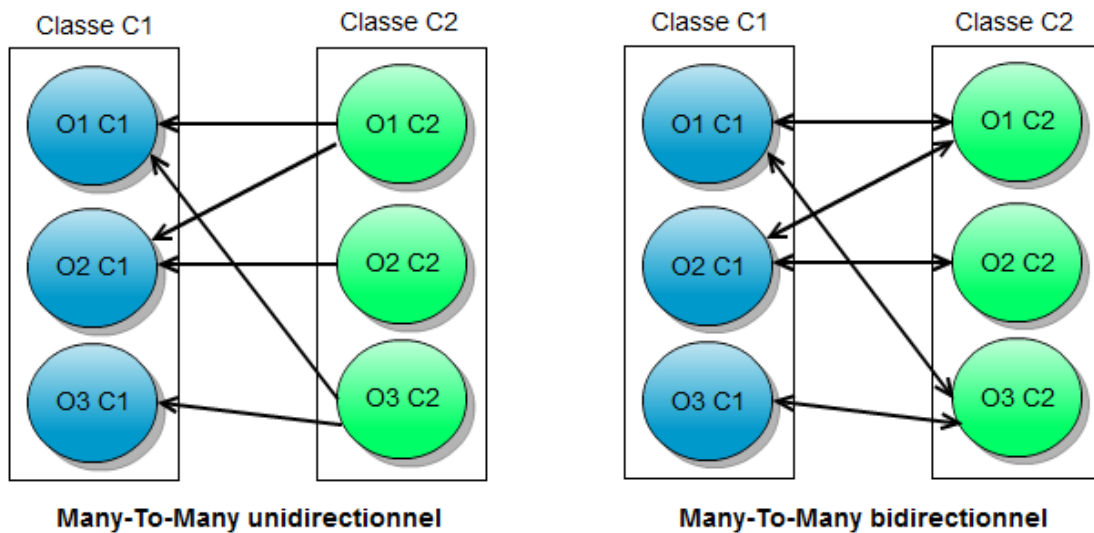


Figure 86 : Différentes sortes de mapping Many-To-Many

Voici un exemple d'annotation pour une relation One-To-One unidirectionnelle (figure 87) :

```
/**
 * @ManyToOne(targetEntity="Question")
 * @JoinColumn(name="id_question", referencedColumnName="id_question")
 */
private $question;
```

Figure 87 : Annotation pour une relation One-To-One

L'annotation `@ManyToOne` comporte un attribut `targetEntity` qui permet de préciser le type de la cible, ici une question. Une seconde annotation `@JoinColumn` permet ensuite de préciser

la clé étrangère (name) et sa colonne cible (referencedColumnName), qui ont ici le même nom dans les deux tables.

3.3. Interfaces de gestion des questionnaires

J'ai essayé de créer des IHM (Interfaces Homme Machine) intuitives, car il est fort probable que les utilisateurs n'utilisent pas le site Internet si celui-ci n'est pas convivial, qu'ils soient visiteurs, volontaires ou chercheurs.















La première page présentée est destinée aux chefs de projets. C'est grâce à elle qu'ils pourront gérer leurs questionnaires (figure 88).

Questionnaires envoyés

Titre	Date d'envoi	Date limite	Date de réception	Volontaire
✓ Appréciation de sa silhouette	08/07/2011 11:53	15/07/2011 11:53	08/07/2011 11:54	GUIRO











Envoyer un questionnaire de la liste ci-dessous

Questionnaires liés au projet

Titre	Date de création	
✗ Questionnaire d'état de santé sf-36	07/06/2011 11:33	  
✗ Perception de la faim et de l'envie	07/06/2011 12:04	  
✗ Appréciation de la fatigue musculaire	07/06/2011 12:12	  
✗ Appéciation d'aliments	07/06/2011 13:08	  
Appréciation de sa silhouette	07/06/2011 13:25	 

Créer un nouveau questionnaire

Questionnaires liés aux autres projets

Titre	Date de création	Projet	
Questionnaire d'état de santé sf-36	07/06/2011 11:33	Projet de BRAMAC	
Perception de la faim et de l'envie	07/06/2011 12:04	Projet de BRAMAC	
Appréciation de la fatigue musculaire	07/06/2011 12:12	Projet de BRAMAC	
Appréciation des matières grasses	07/06/2011 13:08	Projet de BRAMAC	
Appréciation de sa silhouette	07/06/2011 13:25	Projet de BRAMAC	
Questionnaire d'état de santé sf-36	07/06/2011 11:33	Projet de CAIAUC	
Perception de la faim et de l'envie	07/06/2011 12:04	Projet de CAIAUC	
Appréciation de la fatigue musculaire	07/06/2011 12:12	Projet de CAIAUC	
Appréciation des matières grasses	07/06/2011 13:08	Projet de CAIAUC	
Appréciation de sa silhouette	07/06/2011 13:25	Projet de CAIAUC	

1 2 3 4 5 6 ≥ Fin »

Figure 88 : Interface de gestion des questionnaires par le chef de projet

La première partie, en haut, liste les questionnaires qui ont été envoyés aux volontaires. Elle permet de voir rapidement ceux qui auraient déjà dû être renvoyés et ceux qui ont été renvoyés dans les temps. Les chefs de projet peuvent d'ailleurs modifier les réponses faites par les volontaires en cliquant sur une icône à l'extrémité droite de chaque ligne correspondant à un questionnaire reçu.

La seconde partie liste tous les questionnaires qui ont été créés. Ceux qui n'ont pas été envoyés à des volontaires peuvent encore être supprimés ou modifiés. Il est aussi possible, pour tous les questionnaires, de générer une version PDF (figure 89). Ceci peut être utile pour imprimer des questionnaires et les envoyer aux volontaires qui n'ont pas Internet.

The figure displays two pages of a questionnaire PDF. The left page, titled "Perception de contrôle", includes a header with the CRNH logo, a patient name field, and fields for date and time. It contains instructions for the respondent and a list of affirmations. The right page, titled "Partie 1 : Perception du contrôle du poids corporel", contains four questions with multiple-choice options. The questions are:

- Question 1 : Si je prends, perds, ou maintiens mon poids, cela ne dépend que de moi.
- Question 2 : Avoir le bon poids est en grande partie une question de chance.
- Question 3 : Peu m'importe ce qui doit arriver, que je prenne du poids, que j'en perde ou que mon poids se maintienne, advienne que pourra.
- Question 4 : Si je mange bien et fais assez d'exercice et me repose, je peux contrôler mon poids comme je veux.

Figure 89 : Génération d'un questionnaire au format PDF

Enfin, la troisième partie liste tous les questionnaires créés dans le cadre d'autres projets. Cela permet de copier dans son propre projet un questionnaire déjà existant dans un autre.

J'ai mis en place un système de pagination dans ces trois parties, car un trop grand nombre de questionnaires affichés peuvent nuire à la clarté de l'interface. Cette pagination est faite avec AJAX, ce qui permet de ne pas recharger complètement la fenêtre lorsqu'on clique sur une

autre page, mais seulement le tableau voulu. J'économise ainsi un nombre non négligeable de requêtes vers la base de données.

Deux boutons sont aussi disponibles en bas de chacune des deux premières parties. Le premier permet d'envoyer un questionnaire à des volontaires, et le second permet de créer un nouveau questionnaire.

L'interface permettant l'envoi de questionnaires prend la forme d'une fenêtre modale dans laquelle sont listés tous les questionnaires et les volontaires du projet (figure 90). Les dates d'envoi et de fin sont aussi à renseigner à l'aide de calendriers. Pour ces derniers, je me suis servi d'un plugin pour jQuery, dont une version simplifiée a été intégrée à jQuery UI.

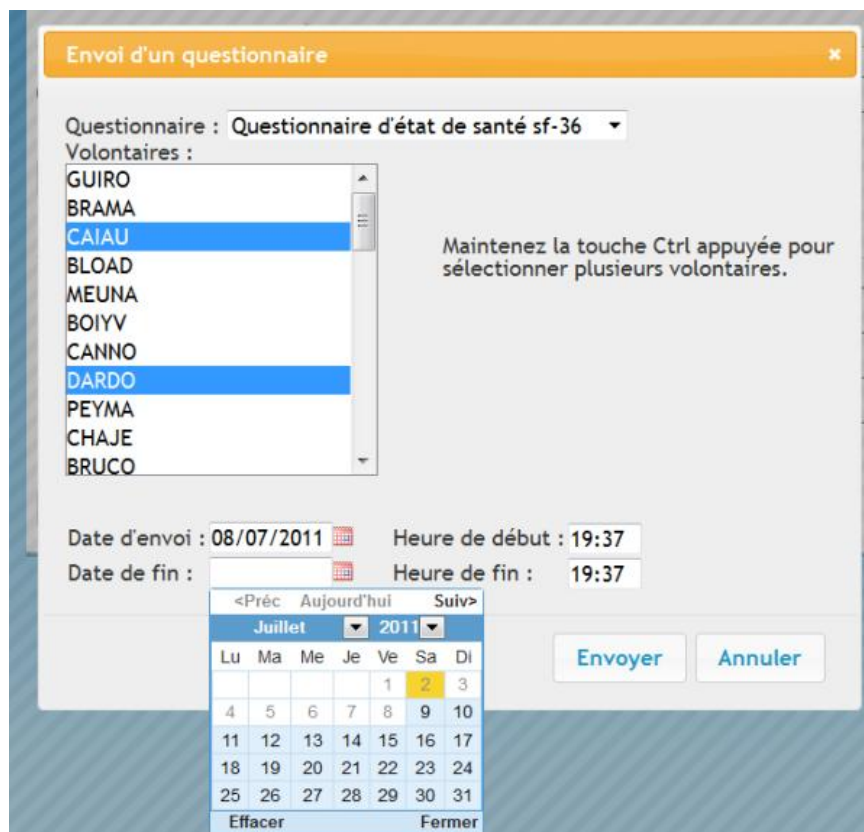


Figure 90 : Interface d'envoi d'un questionnaire

Quant à l'interface de création d'un questionnaire, c'est celle qui m'a pris le plus de temps à créer car il y a une multitude de fonctionnalités à mettre en place à l'aide de JavaScript, afin de rendre l'interface conviviale. Elle se présente comme ceci (figure 91) :

Langues activées : Français English Deutsch

▼ 1. Renseignements généraux concernant le questionnaire

Titre du questionnaire : FR UK DE
 Appéciation d'aliments

Description du questionnaire : FR UK DE
 Questionnaire destiné à mesurer l'appréciation des aliments.

Afficher les parties dans un ordre aléatoire
 Afficher les questions de chaque partie dans un ordre aléatoire

▶ 2. Contenu du questionnaire

▶ 3. Enregistrement du questionnaire

▶ Créer une nouvelle question

Questions existantes :
 Projet : Tous ▼
 Questionnaire :

Figure 91 : Interface de création d'un questionnaire, partie renseignements généraux

Il y a trois parties :

- Les renseignements généraux qui concernent le questionnaire, tels que son titre ou sa description. Les langues sont activables et désactivables par les boutons situés en haut de l'interface. Des drapeaux apparaissent alors à côté de chaque champ. En cliquant sur un drapeau, le champ de saisie change et on peut alors renseigner le texte dans la langue sélectionnée.
- Le contenu du questionnaire. Le questionnaire peut être composé de plusieurs parties. Dans chacune d'elles le chef de projet peut inclure des questions.
- L'enregistrement du questionnaire. J'ai placé cette fonctionnalité dans une partie en bas de l'écran afin que les utilisateurs n'oublient pas de valider leurs modifications.

Les questions que l'utilisateur peut insérer dans les différentes parties du questionnaire peuvent exister ou non. Dans ce dernier cas, l'utilisateur dispose d'une partie intitulée « Créer une nouvelle question ». Elle se présente comme ceci (figure 92) :

Créer une nouvelle question

Langues activées : Français English Deutsch

Ajouter à la fin de la partie : 4



Libellé de la question :  Quel est votre sport favori ?

Image : Parcourir Réinitialiser 


Type de réponse : 1 réponse parmi N

Réponses proposées :

Libellé : Jogging Valeur : 0 Supprimer

Libellé : Tennis Valeur : 2 Supprimer

Libellé : Karting Valeur : 10 Supprimer

 Ajouter un choix de réponse

Enregistrer

Figure 92 : Interface de création d'une nouvelle question

Il faut préciser le libellé de la question, le type de réponse et éventuellement un choix d'image et des éléments de réponse. Il est possible d'inclure la question nouvellement créée dans une des parties du questionnaire (ici la partie 4). Si aucune partie n'est précisée, la question est bien créée mais n'est attachée à aucun questionnaire.

La figure 93 montre la partie de gestion du contenu du questionnaire. Chaque question (qui provient du panneau de création ou de la recherche) peut être déplacée dans la partie ou supprimée. De plus, un panneau de paramétrage apparaît en-dessous de chacun afin de déterminer si elle est obligatoire ainsi que pour définir des conditions d'affichage.

Sur cette figure, la question 3 ne sera affichée que si la réponse à la question 2 est « Je la ressens moyennement ».

Parmi les difficultés rencontrées, il y a notamment la gestion de la numérotation des questions qui doit changer lorsqu'une question est ajoutée, supprimée ou déplacée, de même que pour la gestion des conditions.

▼ 2. Contenu du questionnaire

Ajouter une partie

▼ Partie 1

Titre : Enlever

Consigne :

▶ Question 1. Aimer

▶ Question 2. Doute

▼ Question 3. Gêne

Je ne ressens pas cette émotion Enlever
 Je la ressens un peu
 Je la ressens moyennement
 Je la ressens beaucoup
 Je la ressens extrêmement

Paramétrage

Question obligatoire

Définir une condition d'affichage

Sexe : H F H et F

Age : entre et ans

IMC : entre et

Question : Question 2 Réponse attendue : R1 Je ne ressens pas cette émotion

Inverser la / les condition(s) sélectionnée(s)

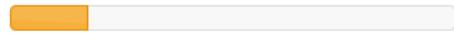
R1 Je ne ressens pas cette émotion
 R2 Je la ressens un peu
 R3 Je la ressens moyennement
 R4 Je la ressens beaucoup
 R5 Je la ressens extrêmement

▶ Question 4. Frustration

▶ Question 5. Amusé

Figure 93 : Gestion des questions

La figure 94 montre la façon dont sont posées les questions aux volontaires :



Partie 1 : Général

Consignes : *Aucune consigne.*



En général, diriez-vous que votre santé est :

- Excellente
- Très bonne
- Bonne
- Satisfaisante
- Mauvaise

Continuer

Figure 94 : Affichage d'une question à un volontaire

Deux barres de progression permettent au volontaire d'estimer le nombre de questions qu'il lui reste à répondre.

La première, en face du titre du questionnaire, indique le nombre de parties terminées. La seconde, en-dessous des consignes de la partie en cours, indique le nombre de questions terminées dans la partie.

3.4. Déploiement d'une version 1 de démonstration

J'ai pu proposer aux futurs utilisateurs, début juillet, de tester une première version en déployant l'application sur un serveur du site de Theix. Afin d'obtenir un accès à ce serveur j'ai fait ma demande auprès de Sébastien Cat. J'ai aussi été amené à le rencontrer pour discuter des différentes contraintes liées au serveur qui devait accueillir l'application. Il me fallait en effet des versions minimum de PHP et de MySQL afin de faire fonctionner Doctrine 2 et CodeIgniter.

La version mise en ligne pour les tests est légèrement différente de l'application finale. J'ai en effet ajouté une interface qui permet aux utilisateurs de me faire remonter facilement leurs remarques et idées, tout en leur laissant accès aux remarques des autres, afin de limiter le nombre de doublons (figure 95). Elles sont enregistrées dans la base de données, ce qui me permet d'aller les consulter à n'importe quel moment.

J'ai une remarque ! - Liste des remarques - Aide

[Voir la liste des remarques](#)

Nom :

Remarques :

Si votre remarque concerne une page particulière, merci d'indiquer son URL (ex. : <https://www3.clermont.inra.fr/ActivCollector/index.php/remarque>).

Si vous avez rencontré un problème, essayez de décrire quelles actions vous ont permis de rencontrer cette erreur.

Envoyer

Figure 95 : Formulaire de saisie d'une nouvelle remarque

D'autre part, comme la gestion des utilisateurs n'était pas encore implémentée, j'ai dû simuler son fonctionnement, ce qui s'est révélé assez complexe car je me fondais pour cela sur les identifiants des tables `user`, `volunteer` et `project` de la base de données. Il fallait donc que je fasse extrêmement attention à la façon dont je créais les comptes des utilisateurs.

De manière à ce que les tests de certains utilisateurs ne gênent pas ceux des autres, j'ai décidé de créer un espace projet pour chacun, ainsi qu'un compte « volontaire » et un autre « chef de projet » (figure 96). Ainsi, j'évite que certains suppriment un questionnaire pendant qu'un autre est en train de le modifier. Ce cas ne surviendrait quasiment jamais dans la version finale car le nombre de chefs de projets sera limité (une ou deux personnes en moyenne).

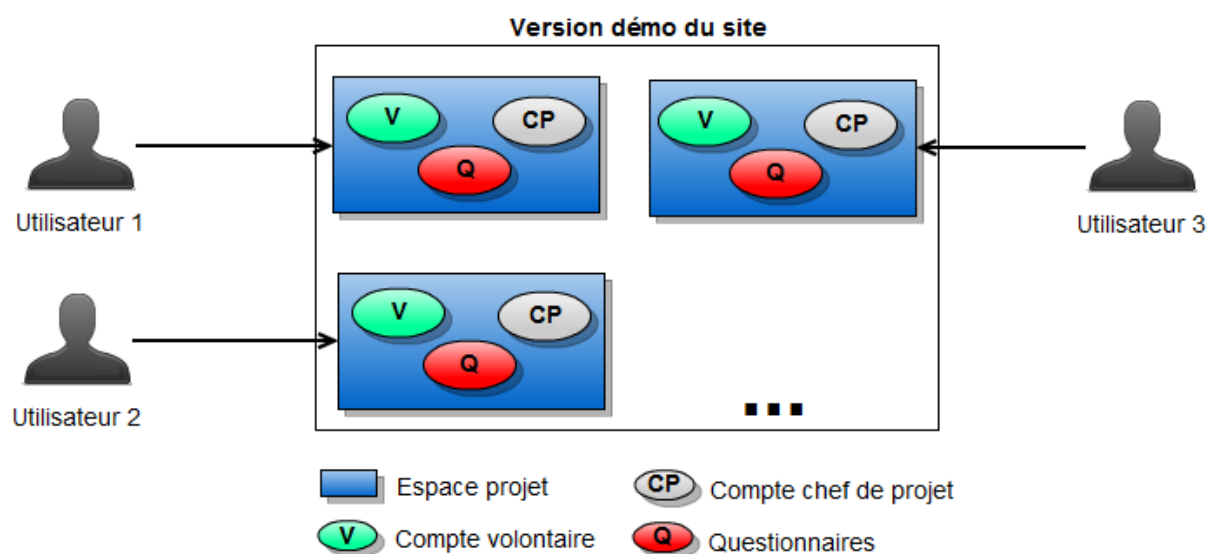


Figure 96 : Gestion des comptes dans la version de démonstration

3.5. Aspects juridiques

Etant donné que le site Internet stockera dans sa base de données plusieurs informations personnelles concernant les volontaires, Sylvie et moi avons monté un dossier à destination de Solange Prime, juriste de l'INRA de Toulouse (figure 97). Celle-ci fera ensuite la demande auprès de la CNIL (Commission Nationale Informatique et Libertés).

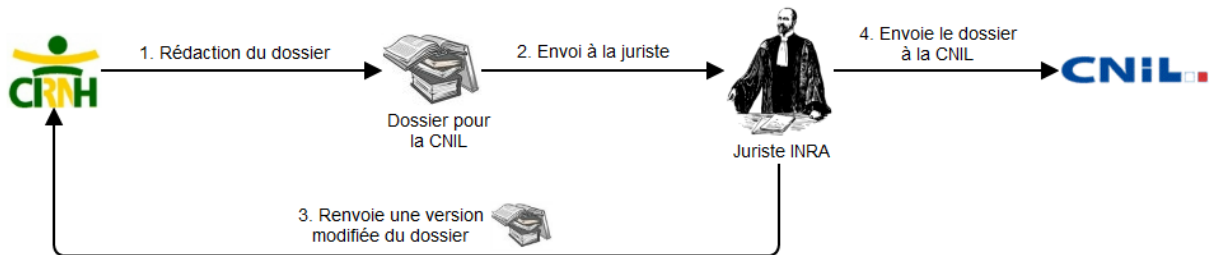


Figure 97 : Echanges avec la juriste de Toulouse

Les données stockées sont les suivantes :

- Les variables biologiques relevées lors des différents examens médicaux. Il est possible, à partir de certaines de ces données, de savoir si un volontaire est en bonne santé ou non.
- Les réponses aux questionnaires envoyés au cours du protocole. Là encore il est possible, selon le type du questionnaire, de déterminer par exemple si le volontaire est dépressif, s'il se sent mal, etc.
- Les données issues des capteurs.
- L'adresse email, s'ils souhaitent que le site puisse les contacter pour leur donner une quelconque information. Les visiteurs auront eux aussi la possibilité de laisser leur adresse email s'ils s'inscrivent à la newsletter.
- Le code volontaire.

Le code volontaire est un code unique pour chaque personne. Il est actuellement constitué des trois premières lettres du nom, des deux premières lettres du prénom et éventuellement d'une clé incrémentale pour éviter les doublons. Ce formalisme est néanmoins sujet à évolution. Il va être modifié dans quelques temps pour ne comporter que les deux premières lettres du nom, la première du prénom et une clé numérique.

Ce code est déjà stocké dans un fichier Excel, sur un des ordinateurs du LNH. Seules quelques personnes sont autorisées à y avoir accès. Néanmoins, la juriste INRA nous a fait savoir que le code volontaire du site Internet ne permet pas l'anonymat aux yeux de la CNIL. Les données sont en effet indirectement nominatives, car il est théoriquement possible de faire le lien entre les données du site et le fichier contenant les informations personnelles (nom, téléphone, adresse, etc).

La juriste nous a informés qu'une demande avait déjà été faite à la CNIL pour ce fichier. Elle considère donc qu'une simple demande d'extension devrait suffire.

La figure 98 montre les flux de données sur le site Internet :

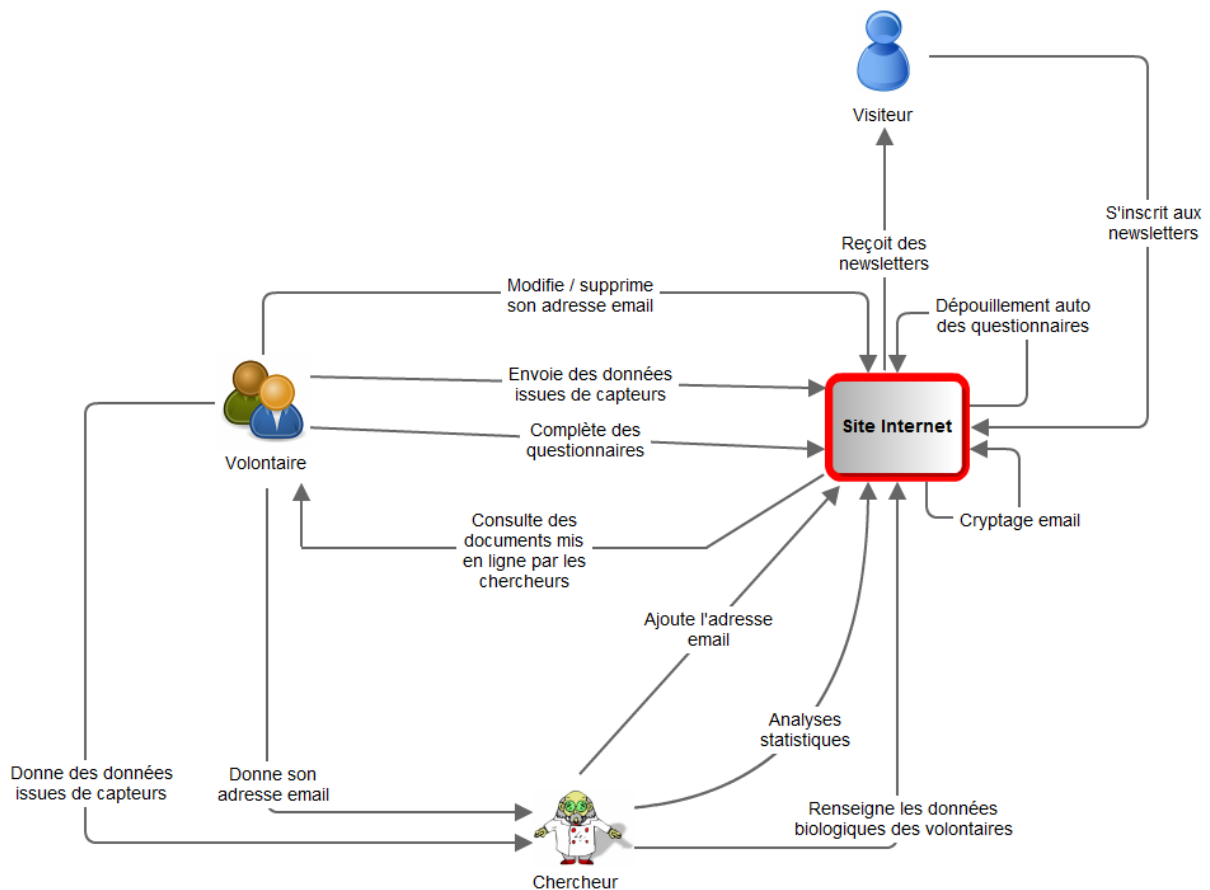


Figure 98 : Flux de données sur le site d'ActivCollector

Etant donné que les gens mettent souvent leur nom et/ou prénom dans leurs adresses email, nous avons réfléchi à un moyen de les sécuriser. Philippe Lacomme m'a alors conseillé de les crypter dans la base de données.

J'ai donc recherché un algorithme qui soit implémenté en PHP. Le cryptage étant une opération réversible, le cryptage et le décryptage ne sont possibles qu'avec une clé (une chaîne

de caractères) gérée par le système et qu'il faut garder secrète. Elle se situera dans le code source du site Internet, qui ne sera accessible que par les personnes qui auront un accès au système de fichiers du serveur (par exemple les personnes chargées de créer le site et les informaticiens qui gèrent le serveur).

L'algorithme que j'ai trouvé est l'algorithme de Rijndael, qui a été choisi en 2000 pour être le standard de chiffrement pour les organisations du gouvernement des États-Unis.

Voici comment sera utilisé cet algorithme (figure 99) :



Figure 99 : Cryptage de l'adresse email

L'adresse email donnée par l'utilisateur est cryptée par l'algorithme en utilisant une clé secrète. C'est le résultat de ce cryptage, une chaîne de caractères incompréhensible pour l'être humain, qui est stocké dans la base de données.

Le décryptage se fait alors de la façon suivante, lorsque le système souhaite joindre automatiquement un volontaire (figure 100) :

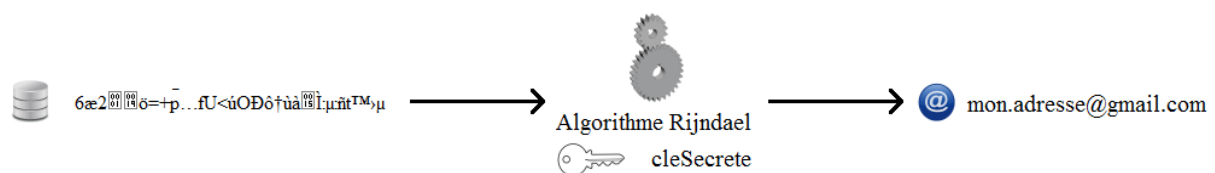


Figure 100 : Décryptage d'une donnée cryptée

Il faut noter que l'adresse email décryptée n'est utilisée que le temps de l'envoi d'un email. Elle est ensuite « détruite ».

La figure 101 montre où se trouve la clé sur le serveur :

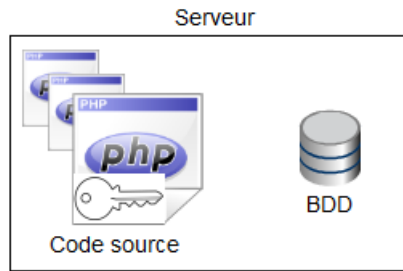


Figure 101 : Position de la clé sur le serveur

Le serveur se trouve dans une salle badgée de l'INRA de Theix (Saint-Genès Champanelle), sous la responsabilité de Sébastien Cat. De plus, l'accès (logiciel) au serveur nécessite de s'authentifier par un nom d'utilisateur et un mot de passe.

Une autre source de données sensibles stockées dans la base de données sera les mots de passe des utilisateurs. Les accès personnels des chercheurs et volontaires seront en effet protégés par un nom d'utilisateur et un mot de passe.

Le mot de passe sera stocké dans la base de données sous la forme de condensat. Un condensat est le résultat d'un algorithme de chiffrement qui retourne une chaîne de caractères pratiquement unique à partir d'une chaîne de caractères donnée (ici un mot de passe).

Contrairement au cryptage, il n'existe pas d'algorithme permettant de faire l'opération inverse, c'est-à-dire d'obtenir le mot de passe initial à partir du condensat.

L'algorithme que j'ai choisi est SHA-1, qui a été conçu par la NSA (« National Security Agency »), aux Etats-Unis. Le condensat produit a une taille de 160 bits, soit 40 chiffres hexadécimaux (caractères de 0 à 9 et de a à f).

Exemple de chiffrement SHA-1 (figure 102) :

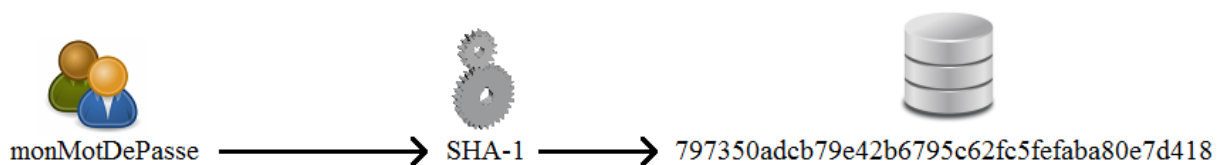


Figure 102 : Chiffrement d'un mot de passe

4. Le projet mobilité

4.1. Naissance du projet

Etant intéressé pour travailler à l'INRA, j'ai sollicité un entretien auprès du directeur de l'UNH (Jean-Michel Chardigny). Cet entretien en présence de monsieur Chardigny, Béatrice Morio (Directrice de Recherche de l'équipe CHLEO) et Sylvie Rousset (Ingénieur de recherche de l'équipe CHLEO et mon encadrante) a eu lieu fin mai.

Jean-Michel Chardigny m'a alors informé qu'un méta programme était en cours de création afin d'être présenté devant un conseil scientifique en fin d'année pour validation. Un tel projet procurerait à l'UNH les ressources financières nécessaires à la création d'un poste d'ingénieur en CDD. Néanmoins, monsieur Chardigny m'a avoué que ce projet aurait peu de chance d'être accepté. Sylvie nous a alors présenté une alternative : une demande de bourse innovation auprès du conseil régional d'Auvergne. Cette bourse permet de recruter un ingénieur en CDD pendant 1 an. Le projet doit montrer un transfert de technologie entre un laboratoire de recherche et une entreprise.

Les organismes publics de recherche peuvent trouver des financements de plusieurs façons, comme le montre le schéma suivant (figure 103) :

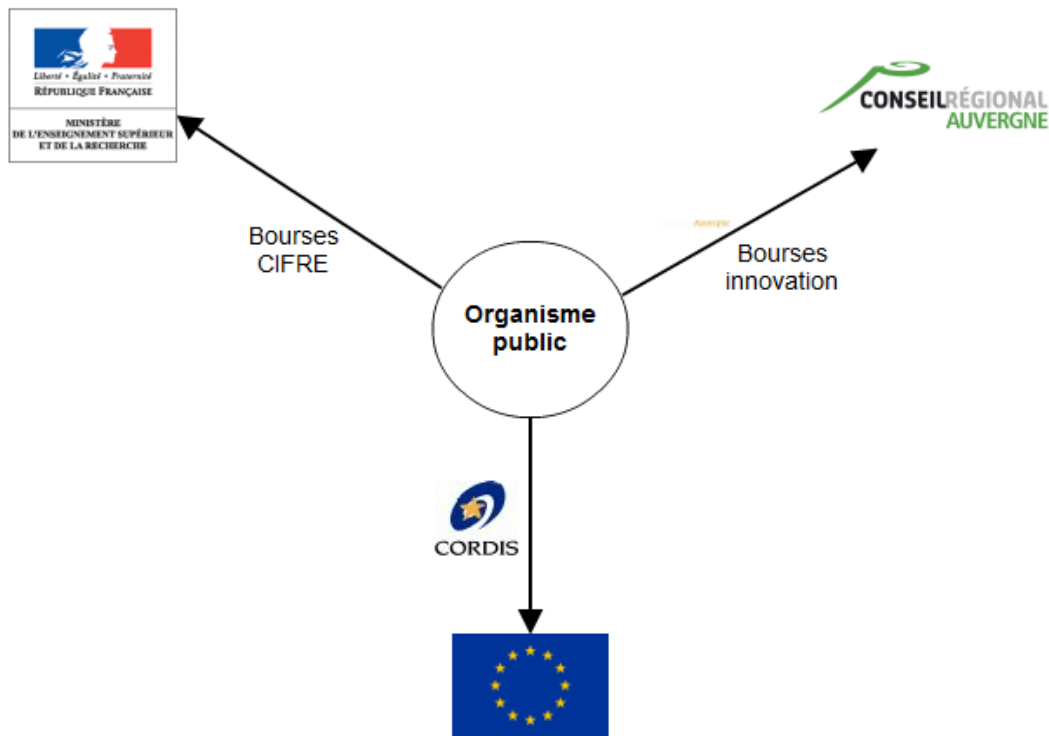


Figure 103 : Financements des organismes publics de recherche

Le ministère de l’enseignement supérieur et de la recherche propose des bourses CIFRE, l’Europe propose des financements via le service CORDIS, et le conseil régional d’Auvergne propose des bourses d’innovation. Nous avons choisi de monter un dossier « Bourse innovation » aux mois de juin et juillet.

Lors de notre réunion, nous avons pensé à l’entreprise Almerys grâce aux liens que Martine Duclos avait déjà noués avec leurs représentants quelques mois auparavant. Martine Duclos est professeur universitaire et praticien hospitalier (PUPH) au CHU de Clermont-Ferrand et mène des travaux de recherche au sein de l’équipe CHLEO de l’INRA-UNH.

Almerys est une filiale du groupe France Télécom et est active dans le domaine de la santé. Son cœur de métier historique est la gestion du tiers-payant pour les mutuelles. Martine Duclos a rencontré cette entreprise dans le cadre d’une demande de financement PHRC (**P**rogramme **H**ospitalier de **R**echerche **C**linique). Son projet consistait à utiliser les smartphones afin de comparer différentes stratégies éducatives dans le traitement de l’obésité en observant des mesures qui portent sur la diététique et l’activité physique. Cette demande de financement ayant finalement été refusée, nous avons pensé qu’il serait intéressant d’en reprendre les grands principes afin de collaborer avec Almerys.

4.2. Rencontre avec Almerys

Fin juin, Jean-Michel Chardigny, Martine Duclos, Sylvie Rousset, Robert Boualit (directeur du service santé d'Almerys) et moi-même nous sommes réunis afin de parler d'une éventuelle collaboration, sous accord de confidentialité. J'ai donc présenté mon travail de stage, la création d'un site Internet, ainsi qu'un projet réalisé par des étudiants de licence L3. Ce projet consistait à élaborer un programme informatique pour collecter des données GPS et d'accélérométrie acquises par des smartphones puis à les envoyer vers un serveur. Ce programme a été implémenté sur les 3 plus grands systèmes d'exploitation du moment (Android, iOS et Windows Phone 7) et a permis de montrer la faisabilité technique [Ramiljoana et al, 2011 ; Cournollet & Fobis, 2011 ; Michel & Besson, 2011].

Cette présentation a permis de montrer à l'entreprise que l'UNH a déjà pris des initiatives dans le domaine de l'informatique. Almerys est intéressé non seulement par le projet smartphones mais également par le site Internet du système ActivCollector.

Deux solutions ont été envisagées pour établir une collaboration :

- Un contrat de recherche.
- La demande de bourse innovation du Conseil Régional.

C'est cette deuxième solution qu'a retenue l'entreprise, en raison de l'aide financière apportée par le conseil régional. Ce dernier donne une bourse de 1500 € nets par mois pendant un an au porteur du projet. Après discussions, l'entreprise a accepté d'apporter un complément de salaire d'environ 300 € par mois pour couvrir le salaire d'un ingénieur.

Almerys et les partenaires du projet envisagent néanmoins une coopération de 5 ans sur ce projet. Après le CDD d'un an, l'entreprise serait prête à proposer 2 bourses CIFRE. Le projet à 1 an consistera à créer une application pour smartphones capable d'indiquer aux utilisateurs la durée et l'intensité de leur activité physique. De plus, l'application fournira des bilans / écarts entre l'objectif d'activité physique souhaitée et réalisée.

La première étape du projet consistera à construire un modèle de prédiction de l'activité physique à partir des données d'accélérométrie et GPS. Quand le modèle sera validé, l'application calculera l'écart entre les objectifs et l'activité physique de la personne. Les objectifs pourront être définis par l'utilisateur lui-même (par exemple faire 3h de marche par semaine) ou par le plan national nutrition santé (PNNS).

Le projet à 5 ans devra permettre de prendre en compte d'autres éléments tels que le sommeil ou le stress afin de faire un retour plus complet à l'utilisateur. L'application smartphone permettra aussi de poser / répondre à des questionnaires sur les conditions de vie. Les chercheurs pourront ainsi suivre l'évolution des comportements et des scores de qualité de vie.

La figure 104 montre le cadre du projet :

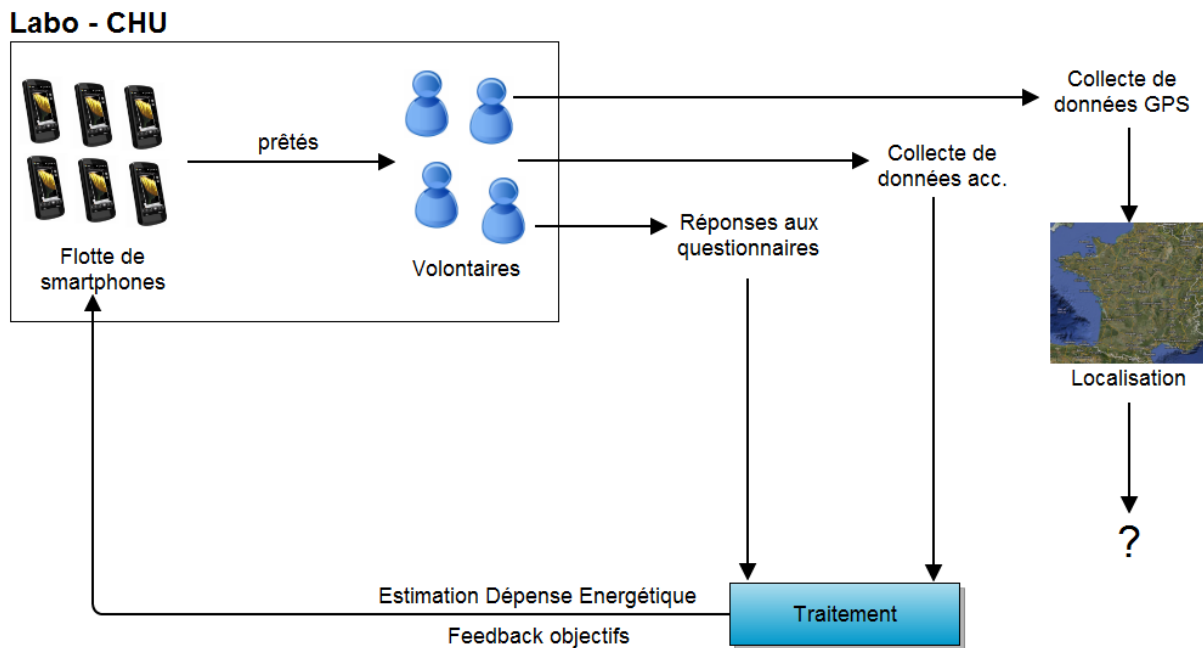


Figure 104 : Projet à 5 ans avec Almerys

Le projet comportera 4 partenaires :

- Almerys, par l'intermédiaire notamment de monsieur Boualit.
- L'INRA-UNH, avec Sylvie Rousset, ingénieur de recherche.
- Le CHU, avec Martine Duclos, chef du service de Médecine du Sport au CHU de Clermont-Ferrand.
- Le LIMOS (Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes), avec Philippe Lacomme, maître de conférences à l'université Blaise Pascal.

Chacun aura un rôle particulier. Etant donné qu'Almerys possède une grande compétence dans la gestion des données et dans les questionnaires en particulier, nous nous sommes mis d'accords pour qu'un de leurs ingénieurs passe environ 25% de son temps sur la partie

questionnaires de l'application. Le reste de la conception et du développement sera à la charge du porteur de projet, c'est-à-dire le bénéficiaire de la bourse.

L'entreprise a par ailleurs déjà développé des outils tels que des cahiers de vaccination ou des applications pour smartphones grand public qui ont pour but d'éduquer et d'aider les gens à mieux gérer leur santé en général.

L'INRA-UNH aura pour rôle de construire un modèle de prédiction sur les données acquises auprès de 25 volontaires, puis de tester et valider l'application en effectuant des tests sur une cinquantaine de volontaires normo-pondéraux ou obèses. C'est elle qui apportera la caution scientifique au projet.

Enfin, le LIMOS sera chargé de trouver des algorithmes qui permettent de déterminer le type d'activité physique effectué en fonction des données fournies par les smartphones.

Almerys sera responsable de la démarche des clients tels que des mutuelles ou des entreprises généralistes afin de leur vendre et d'installer l'application. Ces clients fourniront ensuite l'application à leurs propres clients (figure 105).

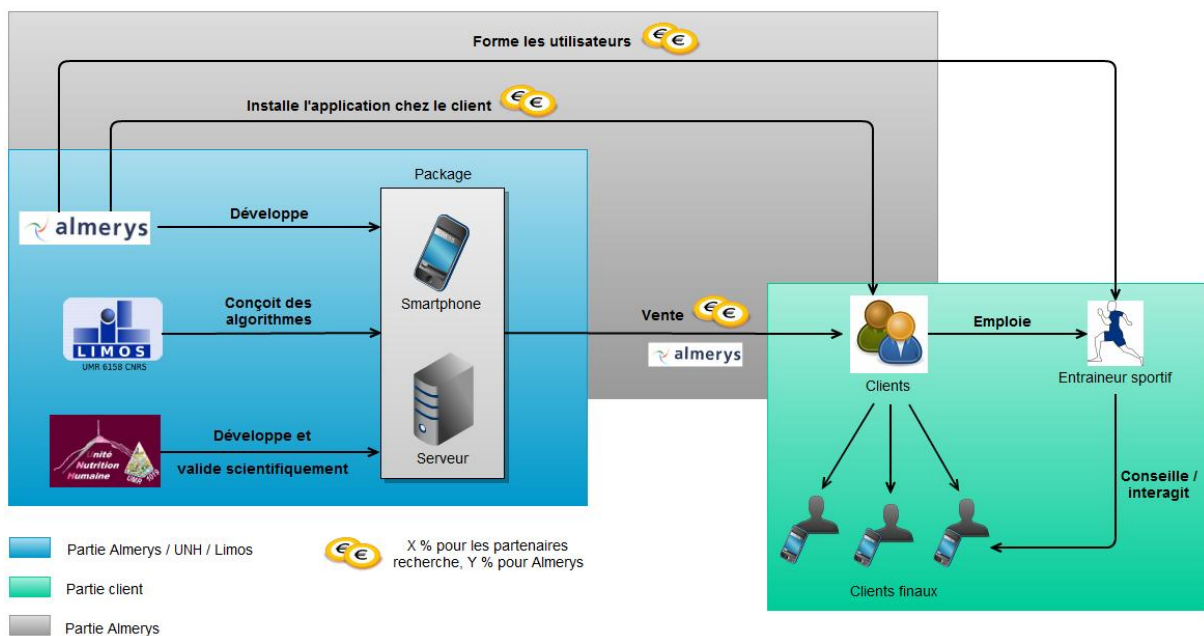


Figure 105 : Objectifs du projet à 5 ans

Les mutuelles s'associeront à des entraîneurs sportifs qui auront pour rôle de conseiller et d'interagir avec les utilisateurs finaux. Almerys pourra donc aussi vendre de la formation l'utilisation de l'application à ces entraîneurs.

4.3. Le dossier de demande de bourse

Lors de la préparation du dossier de demande de bourse, mon rôle a consisté à indiquer quelles seront les tâches à réaliser et à donner une estimation de la durée de chacune. J'ai pour cela bénéficié de l'aide de Sylvie Rousset et de Philippe Lacomme. La figure 106 présente les tâches et leurs affectations aux différents partenaires, qui a été décidée début juillet, lors d'une autre réunion avec Almerys :

	Boursier	UNH	CHU	LIMOS	Almerys
T1. Analyse					
T1.1. Analyse des besoins	X	X	X	X	X
T1.2. Analyse du travail de programmation déjà réalisé)	X			X	
T1.3. Analyse du marché					
=> Cibles, marché					X
=> Système(s) d'exploitation ciblé(s) ?					X
T2. Acquisition des données pour la construction du modèle					
T2.1. Spécification des activités		X	X		
T2.2. Rédaction d'un protocole soumis au Comité de Protection de personnes		X			
T2.3. Recrutement de 25 volontaires et mise en place de la procédure d'acquisition des données	X	X	X		
T3. Spécifications					
T3.1. Spécification des besoins (modélisation)	X			X	
T3.2. Spécifications techniques (développement informatique)	X				
T4. Réalisation d'une version v1					
T4.1. Gestion des capteurs et envoi des données sur le serveur	X	X			
T4.2. Traitement des données par le serveur	X	X			
T4.2.1. Calcul de la dépense énergétique	X	X			
T4.2.2. Reconnaissance de l'activité physique effectuée	X	X			
T4.3. Transmission des résultats sur le SmartPhone	X				
T4.4. Gestion des questionnaires					X
T4.5. Dépouillement des questionnaires					X
T4.6. Gestion du feedback affiché à l'utilisateur	X				X
T4.7. Interface de visualisation des données pour que le chercheur puisse valider les réalisations	X	X			
T5. Mise en production					
T5.1. Tests sur une personne		X	X		
T5.2. Tests sur un groupe de 50 volontaires		X	X		
T6. Validation des résultats obtenus par les tests sur les volontaires	X	X	X		
T7. Correction de bugs et mise en service de la version v2	X			X	X
T8. Rédaction du rapport final	X	X	X	X	X

Figure 106 : Tâches du projet à 1 an et répartition entre les partenaires

Les principales étapes seront l'analyse des besoins et du marché, la construction du modèle, les spécifications, le développement et la validation des résultats par l'UNH.

Voici maintenant les durées prévues pour chaque tâche (figure 107) :

Tâches / Mois	2011		2012										
	11	12	1	2	3	4	5	6	7	8	9	10	
T1: Analyse													
T1.1 : Analyse des besoins													
T1.2:Analyse du travail de programmation des étudiants de L3 (collecte des données d'accélérométrie et GPS)													
T1.3: Analyse du marché: système d'exploitation ciblé													
T2 : Acquisition des données pour la construction du modèle													
T2.1 :Spécification des activités													
T2.3 : Rédaction d'un protocole CPP													
T2.4: Recrutement de 25 volontaires et acquisition des données													
T3 : Spécifications													
T3.1 : Spécifications des besoins (modélisation)													
T3.2 Spécifications techniques (développement informatique)													
T4: Réalisation d'une version v1													
T4.1 : Gestion des capteurs et envoi des données sur le serveur													
T4.2: Traitement des données par le serveur													
Calcul de la dépense énergétique													
Reconnaissance de l'activité physique													
T4.3: Transmission des résultats sur le SmartPhone													
T4.4 : Gestion de l'envoi des questionnaires													
T4.5: Dépouillement des questionnaires													
T4.6: Gestion du feedback affiché à l'utilisateur													
T4.7: Interface de visualisation des données/résultats destinées aux chercheurs													
T5: Mise en production													
T5.1 Tests sur une personne													
T5.2 Tests sur 50 volontaires													
T6. Validation des résultats obtenus par les tests sur volontaires													
T7. Correction des bugs et mise en service de la version v2													
T8 : Rédaction du rapport final													

Figure 107 : Planification des tâches durant la première année

Afin de nous aider à monter ce dossier de demande de bourse, nous avons eu l'aide de Lucile Bruchet, du cluster d'excellence Alimentation-Nutrition-Santé [Nutravita], et de monsieur Perrin, du conseil régional.

Nutravita est l'équivalent d'Auvergne TIC dans le domaine de l'alimentation et de la santé. C'est un groupement qui permet la mise en réseau des acteurs de la filière, comme des entreprises ou des centres de recherche.

Le schéma suivant montre la façon dont nous avons communiqué durant la rédaction du dossier (figure 108) :

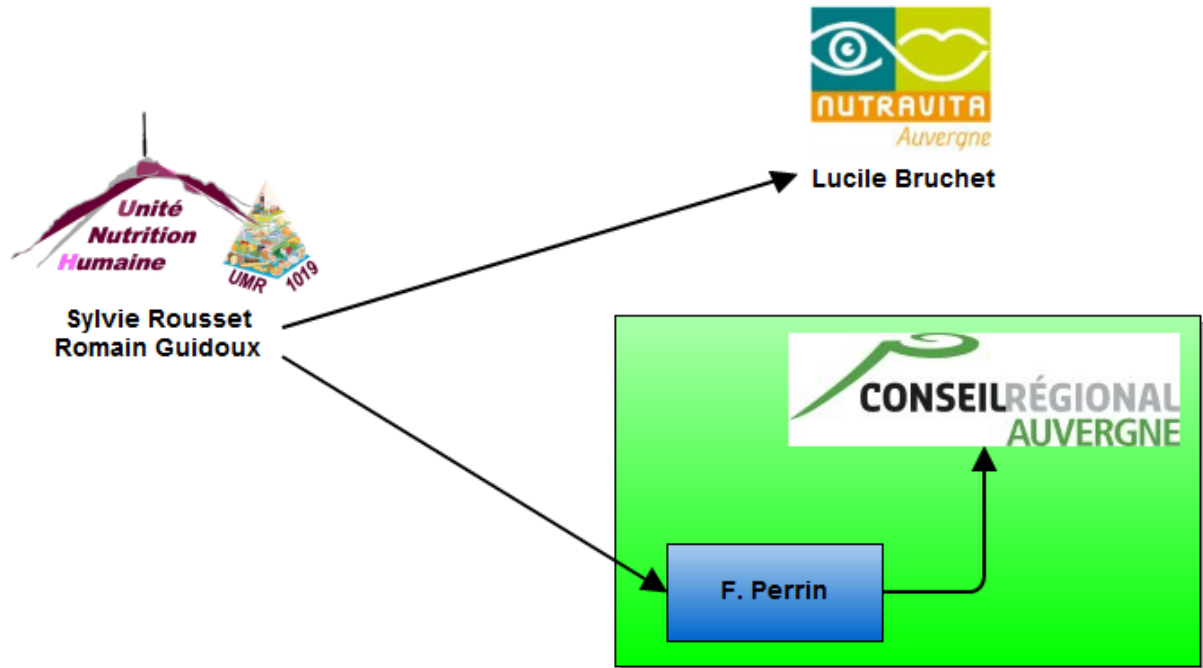


Figure 108 : Communications dans le cadre de la rédaction du dossier

4.4. Discussion

Ma mission principale a été de compléter le cahier des charges du site Internet d'ActivCollector commencé par deux étudiants de l'ISIMA, puis d'étudier les ressources technologiques afin de déterminer lesquelles étaient les plus adaptées dans le contexte de mon stage.

Après plusieurs mois d'utilisation, les frameworks CodeIgniter, Doctrine 2 et jQuery m'ont donné entière satisfaction. J'ai ainsi pu réaliser une première version des questionnaires qui est actuellement en cours de tests par les futurs utilisateurs. Leurs retours me permettront de savoir si les interfaces leur paraissent faciles à prendre en main et si les fonctionnalités apportées les satisfont.

La prochaine étape consistera à terminer la modélisation de la partie dépouillement des questionnaires et à commencer l'implémentation. Il faudra ensuite réfléchir à la façon dont

seront analysées les données. La figure 109 présente l'état d'avancement des modules des questionnaires au 15 juillet 2011.

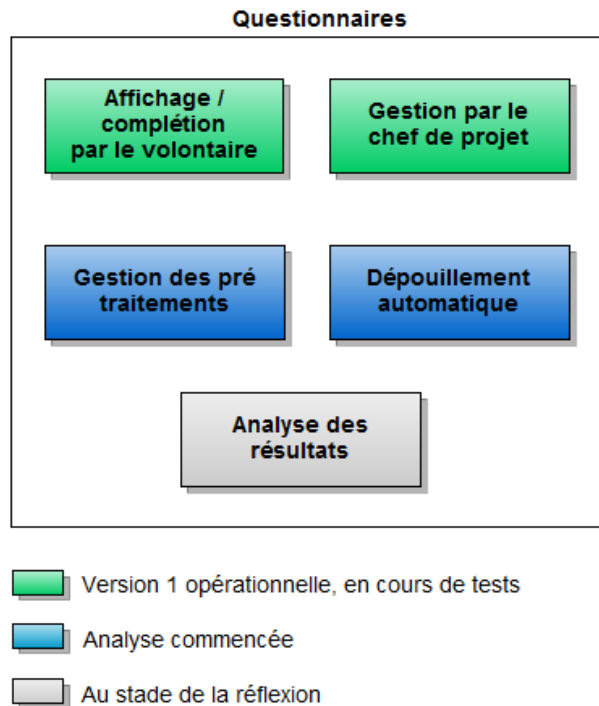


Figure 109 : Avancement de mon travail

Une deuxième version devrait ensuite être développée afin d'ajouter certaines fonctionnalités volontairement mises de côté dans un premier temps, comme la correspondance des réponses à une base de données pour faire des conversions d'aliments en nutriments par exemple (cas des questionnaires alimentaires). Ces questionnaires sont compliqués à réaliser car ils doivent s'interfacer avec ce qui a déjà été créé, tout en restant simples d'utilisation pour les chefs de projets.

Voici la chronologie de mon travail dans l'entreprise, de début avril jusqu'au mois de juillet (figure 110) :

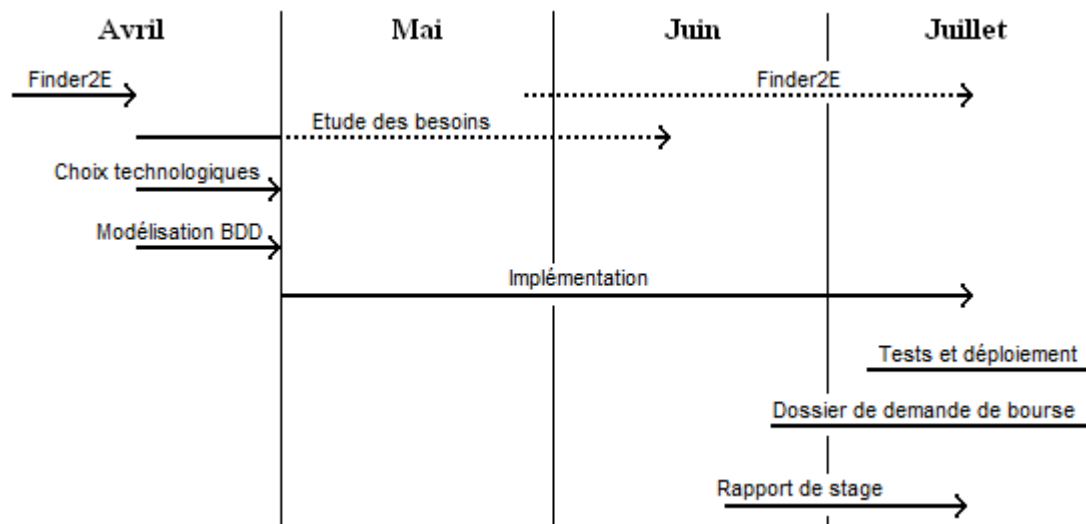


Figure 110 : Chronologie de mon travail

L'étude des besoins du site a surtout eu lieu au début du stage, au mois d'avril, mais j'ai aussi eu l'occasion de rencontrer des utilisateurs durant les mois de mai et juin afin de prendre en compte leurs idées et remarques. J'ai par exemple pu présenter le projet avec Sylvie Rousset à une animation scientifique courant juin sur le site de Theix. Une quinzaine de personnes était présente.

Le déploiement de l'application sur le serveur de Theix a quant à lui eu lieu fin juin, mais la première version de test a été mise en place et diffusée le 06 juillet 2011.

Ma mission secondaire a été de participer à l'encadrement d'un stagiaire de licence L3, qui a continué le travail sur Finder2E de mai à août, afin de l'aider à trouver l'origine de certains bugs par exemple. En effet, j'ai consacré mes deux premières semaines de stage au débogage et à la correction de problèmes mémoire sur ce logiciel, ce qui m'a permis de me familiariser avec son mode de fonctionnement.

Enfin, j'ai participé à la réponse de l'appel d'offre du conseil régional d'Auvergne qui propose des bourses d'innovation. Cette préparation a consisté à préparer un exposé sur le travail déjà effectué (élaboration d'ActivCollector, acquisition des données d'accélérométrie et GPS à partir des smartphones) ainsi que les tâches à réaliser dans le cadre du projet EMouve pour prédire l'activité physique et la dépense énergétique à partir des données collectées sur les smartphones.

Le dossier de demande de bourse et les réunions au LNH et chez Almerys m'ont permis de voir la création d'un projet du début jusqu'à la fin. J'ai donc dû gérer parallèlement mon

travail sur le site du système ActivCollector, le projet mobilité avec Almerys et les petites aides apportées à l'étudiant qui travaillait sur Finder2E.

Conclusion

L'objectif de mon stage est de concevoir un site Internet qui permette aux chercheurs d'être plus efficaces dans la gestion des données issues d'études cliniques.

J'ai rencontré beaucoup de personnes qui sont intéressées par cet outil, ce qui m'a permis de bien cerner leurs besoins. La partie gestion des questionnaires, par laquelle j'ai commencé, n'est pas encore terminée au moment où je rédige ce rapport. J'ai réalisé une première version qui permet de créer, modifier, supprimer des questionnaires, mais aussi d'en générer une version PDF. De plus, j'ai implémenté la partie qui permet aux volontaires de répondre aux questions.

Il reste néanmoins beaucoup de travail, comme par exemple le dépouillement des questionnaires que les volontaires renvoient ou les analyses statistiques effectuées sur des familles de volontaires. De plus, toutes les autres briques de l'application que sont la gestion des utilisateurs, des agendas pour les rendez-vous, des capteurs, etc n'ont pas encore été commencées.

Les principales difficultés techniques ont porté sur deux points. Le premier concerne l'utilisation de jQuery, car j'ai souvent dû utiliser plusieurs plug-ins sur un même élément HTML, ce qui peut causer certains problèmes peu documentés sur Internet, car rarement rencontrés. Le second point concerne la modélisation de la base de données. J'ai trouvé très difficile de pouvoir proposer une solution à certains besoins des utilisateurs concernant les questionnaires, notamment en devant créer des interfaces faciles d'utilisation. En effet, la modélisation que j'ai proposée devait être assez robuste pour pouvoir s'adapter à quelques évolutions des besoins des utilisateurs, ce qui est pour l'instant le cas. J'ai par exemple dû m'adapter lorsque nous avons décidé de permettre aux chercheurs de modifier les réponses données par les volontaires, et de stocker l'historique de ces modifications.

L'étude et l'implémentation du site Internet d'ActivCollector sont donc encore loin d'être terminées. J'ai donc accepté un contrat jusqu'à fin octobre pour pouvoir proposer une première version stable et utilisable aux utilisateurs.

Afin que cette transition soit aisée, j'ai documenté mon code et rédigé quelques documents qui listent les besoins tels qu'ils ont été définis à ce jour.

Webographie

[Actiheart] Bio-Lynx., *Introducing a New Version of Actiheart* [Site Internet], réf. du 04 juillet 2011. Canada.

» <http://www.bio-lynx.com/actiheart.htm>

[Aptana] *Aptana Studio 3* [Site Internet], réf. du 05 juillet 2011. Etats-Unis.

» <http://www.apтана.com/>

[Bug Doctrine] Nico Kaiser, *Doctrine 2 ORM - Allow @Id on @ManyToOne fields* [Site Internet], 09 janvier 2011 [réf. du 05 juillet 2011]. Etats-Unis.

» <http://www.doctrine-project.org/jira/browse/DDC-117>

[Ciqual] AFSSA., *Composition nutritionnelle des aliments - Table CIQUAL 2008* [Site Internet], réf. du 04 juillet 2011. France.

» <http://www.afssa.fr/TableCIQUAL>

[CodeIgniter 2] *Application Flow Chart* [Site Internet], réf. du 05 juillet 2011. Etats-Unis.

» http://codeigniter.com/user_guide/overview/appflow.html

[Dietsys] National Cancer Institute., *Dietsys* [Site Internet], 20 juin 2007, [réf. du 04 juillet 2011]. Etats-Unis.

» <http://appliedresearch.cancer.gov/DietSys/materials.html>

[Doctrine] *Welcome to the Doctrine Project* [Site Internet], réf. du 04 juillet 2011. Etats-Unis.

» <http://www.doctrine-project.org>

[Eclipse PHP] *PHP Development Tools Project* [Site Internet], réf. du 05 juillet 2011. Etats-Unis.

» <http://www.eclipse.org/pdt/>

[Feuille de route Doctrine 2] Benjamin Eberlei, *Whats next? Our Roadmap* [Site Internet], janvier 2011 [réf. du 05 juillet 2011]. Etats-Unis.

» <http://www.doctrine-project.org/blog/roadmap-doctrine2>

[Hibernate] *Relational Persistence for Java and .NET* [Site Internet], réf. du 05 juillet 2011. Etats-Unis.

» <http://www.hibernate.org/>

[jQuery UI Datepicker] *Datepicker* [Site Internet], réf. du 05 juillet 2011. Etats-Unis.

» <http://jqueryui.com/demos/datepicker/>

[jQuery UI Slider] *Slider* [Site Internet], réf. du 05 juillet 2011. Etats-Unis.

» <http://jqueryui.com/demos/slider/>

[jQuery UI Sortable] *Sortable* [Site Internet], réf. du 05 juillet 2011. Etats-Unis.

» <http://jqueryui.com/demos/sortable/>

[Komodo] *Komodo IDE* [Site Internet], réf. du 05 juillet 2011. Etats-Unis.

» <http://www.activestate.com/komodo-ide>

[LimeSurvey] *LimeSurvey* [Site Internet], réf. du 05 juillet 2011. Allemagne.

» <http://www.limesurvey.org/>

[Mono] *ASP.NET - Mono* [Site Internet], réf. du 05 juillet 2011. Etats-Unis.

» <http://www.mono-project.com/ASP.NET>

[MySQL Workbench] *MySQL Workbench* [Site Internet], réf. du 05 juillet 2011. Etats-Unis.

» <http://www.mysql.fr/products/workbench/>

[NetBeans] *NetBeans IDE* [Site Internet], réf. du 05 juillet 2011. Etats-Unis.

» <http://netbeans.org/>

[Nutravita] *Nutravita, le réseau de compétence en nutrition santé en Auvergne* [Site Internet],
réf. du 14 juillet 2011. France.

» <http://www.nutravita.fr/spip.php?article7>

[PHP Designer] *phpDesigner* [Site Internet], réf. du 05 juillet 2011. Danemark.

» <http://www.mpssoftware.dk/phpdesigner.php>

[Wamp] *WampServer* [Site Internet], réf. du 05 juillet 2011. France.

» <http://www.wampserver.com/>

[Zend Studio] *Zend Studio* [Site Internet], réf. du 05 juillet 2011. Etats-Unis.

» <http://www.zend.com/fr/products/studio/>

Références bibliographiques

[Cournollet & Fobis, 2011] Thibaud Cournollet et Samuel Fobis, *Réalisation d'un programme pour Windows Phone 7 pour le suivi de l'activité physique*, rapport de projet de licence, université Blaise Pascal, Clermont-Ferrand, 2011.

[Kamèche & Roux, 2011] Naïma Kamèche et Geoffrey Roux, *Développement du logiciel Finder2E pour tester des modèles de prédiction de la dépense énergétique et élaboration du cahier des charges du service Web « ActivCollector »*, rapport de stage de 3^{ème} année d'ingénieur de l'institut supérieur d'informatique, de modélisation et leurs applications, 2011.

[Michel & Besson, 2011] Jérôme Michel et Julien Besson, *Elaboration d'une application iPhone et d'un site Internet pour le suivi de l'activité physique*, rapport de projet de licence, université Blaise Pascal, Clermont-Ferrand, 2011.

[Ramilijoana et al, 2011] Andrianaina Ramilijaona, Sofiane Fatnassi et Andry Rajerison, *Développement Mobile sous Android 2.2*, rapport de projet de licence, université Blaise Pascal, Clermont-Ferrand, 2011.