

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE  
UNIVERSITÉ DU QUÉBEC

RAPPORT DE PROJET PRÉSENTÉ À  
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE  
À L'OBTENTION DE LA  
MAITRISE EN GÉNIE DES TECHNOLOGIES DE L'INFORMATION

PAR  
Hicham TAZOUTI

GÉNÉRATEUR DE SONDAGES EN LIGNE POUR LE SUIVI DES PATIENTS QUI  
ONT DES MALADIES CHRONIQUES

MONTRÉAL, LE 12 DÉCEMBRE 2016



Hicham TAZOUTI, 2016



Cette licence [Creative Commons](https://creativecommons.org/licenses/by-nc-nd/4.0/) signifie qu'il est permis de diffuser, d'imprimer ou de sauvegarder sur un autre support une partie ou la totalité de cette œuvre à condition de mentionner l'auteur, que ces utilisations soient faites à des fins non commerciales et que le contenu de l'œuvre n'ait pas été modifié.

## **PRÉSENTATION DU JURY**

CE RAPPORT DE PROJET A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

Professeur Alain April, directeur de projet,  
Département de Génie Logiciel et TI à l'École de technologie supérieure

Professeur Abdelaoued Gherbi, membre du jury  
Département de Génie Logiciel et TI à l'École de technologie supérieure







## **REMERCIEMENTS**

Je tiens tout d'abord à offrir mes remerciements les plus sincères à mon directeur de projet, M. Alain April pour ses conseils, son temps, sa patience et son support tout au long de ce projet. Ses orientations et ses corrections lors de l'écriture de ce rapport m'ont permis de réaliser un travail de qualité. Encore une fois merci.

Bien entendu, je voudrais aussi remercier le Docteur Benoit Deslaurier qui a activement participé à ce projet, et qui a fourni les informations nécessaires pour arriver aux résultats attendus. J'espère que le projet lui sera utile dans le future.

Finalement j'aimerais dédier ce travail à ma femme kaoutar, pour son soutien tout au long des études de la maîtrise, pour mes enfants Owais et Ranya et pour mes parents.



# **GÉNÉRATEUR DE SONDAGES EN LIGNE POUR LE SUIVI DES PATIENTS QUI ONT DES MALADIES CHRONIQUES**

Hicham TAZOUTI

## **RÉSUMÉ**

Ce rapport présente le travail en recherche appliquée effectuée afin de réaliser un générateur de sondage de ligne pour le suivi des patients qui ont des maladies chroniques. Ce projet s'inscrit dans les nouvelles voies explorées par les chercheurs et qui se concentrent sur la prévention et l'autogestion de la maladie par le patient. Le souhait du Docteur Benoit Deslaurier est d'implémenter dans un premier lieu le questionnaire qui fait l'évaluation de l'état de la maladie pulmonaire obstructive chronique chez le patient à l'aide du logiciel libre LimeSurvey, dans un second temps l'étude et le développement d'une application a été entamé afin de permettre aux gestionnaires du cas du patient de consulter les résultats des questionnaires et de les alerter le dans le cas où le score obtenu nécessite une intervention.

Outre l'amélioration de la santé des patients atteints de genre de maladie, ce genre de projet aura un impact économique sur les dépenses hospitalières, et il participera certainement à la diminution de l'engorgement des urgences.

Mots-clés : questionnaire, MPOC, Limesurvey, JAVA, infonuagique



# **ONLINE QUESTIONNAIRE FOR PATIENTS WITH CHRONIC DISEASES TO BETTER FOLLOW-UP ON THEIR CONDITION**

Hicham TAZOUTI

## **ABSTRACT**

This report presents the applied research work that has been done to develop an on line questionnaire for patients with chronic diseases to better follow-up on their condition. This project is in line with the new paths explored by researchers that focus on the prevention and self-management of the disease. The wish of the Doctor Benoit Deslaurier is to implement in the first place the questionnaire which evaluates the condition of the chronic obstructive pulmonary disease by using the free software LimeSurvey, secondly the study and the development of an application has been initiated in order to allow the patient's case managers to consult the results of the questionnaires and to alert them in case the score obtained requires intervention.

In addition to improving the patient's health, this project will have an economic impact on hospital expenditures, and it will certainly contribute to reduce the congestion of emergencies.

Key-words: questionnaire, MPOC, Limesurvey, JAVA, cloud



## TABLES DES MATIÈRES

INTRODUCTION .....	1
Chapitre 1 Revue de littérature et problématique .....	3
1.1 Problématiques.....	3
1.2 Les différentes approches de sondage.....	4
1.3 Autres considérations.....	6
1.4 Conclusion .....	6
Chapitre 2 Présentation de la solution pour l’outil de sondage en ligne.....	7
2.1 Contexte .....	7
2.2 Méthodologies de travail.....	7
2.3 Exigences du système .....	8
2.3.1 Survol du modèle du cas d’utilisation.....	8
2.3.2 Les acteurs.....	10
2.2.3 Exigence fonctionnelles et non fonctionnelles.....	10
2.3.4 Contraintes de conception .....	11
2.3.5 Interfaces utilisateurs .....	11
2.4 Architecture de la solution .....	13
2.4.1 Vue Composants et connecteurs .....	13
2.4.2 Vue modules.....	15
2.5 Réalisation de l’application SDSSR .....	18
2.5.1 Liste des principaux libraires et cadriciels .....	19
2.5.2 Détails sur l’implémentation .....	22
2.5.3 Tests et qualité.....	25
2.6 Processus de compilation et de déploiement .....	27
2.6.1 Déploiement sur le serveur local.....	27
2.6.2 Déploiement en production sur le serveur Cloud.....	28
2.6.3 Installation des logiciels nécessaire pour procéder au déploiement.....	29
2.7 Conclusion .....	32
Chapitre 3 Résultats et analyses.....	33
3.1 Résultats.....	33
3.2 Expérimentation et validation.....	35
3.3 Recommandations.....	36
CONCLUSION 38	
ANNEXE I Exemple du questionnaire MPOC.....	39
ANNEXE II Procédure pour l’ajout et la modification du questionnaire dans LimeSurvey...41	
ANNEXE III Consultation des résultats avec l’application SDSSR .....	45

BIBLIOGRAPHIE .....49

## LISTE DES TABLEAUX

	Page
Tableau 1.1 Méthodes de sondages (Valerie M. Sue Lois A. Ritter [3]).....	4
Tableau 2.1 Descriptions des cas d'utilisations.....	9
Tableau 2.2 Acteurs du système .....	10
Tableau 2.3 Exigences fonctionnelles.....	10
Tableau 2.4 Exigences non fonctionnelles.....	11
Tableau 2.5 Contraintes de conceptions .....	11
Tableau 2.6 Composants et leurs responsabilités .....	14
Tableau 2.7 Connecteur et leurs responsabilités.....	15
Tableau 2.8 Modules et leurs responsabilités .....	16
Tableau 2.9 Caractéristiques du serveur Cloud d'OVH .....	28
Tableau 3.2 Accès aux applications installées.....	35



## LISTE DES FIGURES

	Page
Figure 2.1 Diagramme des Cas d'utilisation .....	9
Figure 2.2 Prototype de la page de connexion.....	12
Figure 2.3 - Prototype de la page des participants .....	12
Figure 2.4 Vue Composants et connecteurs.....	14
Figure 2.5 Architecture en couches .....	15
Figure 2.6 Vue en modules .....	16
Figure 2.7 diagramme de séquence illustrant les appels entre.....	17
Figure 2.8 dépendances entre modules sdssr générés par maven .....	19
Figure 2.9 Capture d'écran présentant la hiérarchie des modules .....	21
Figure 2.10 Capture d'écran montrant le code dans Sourceforge.....	22
Figure 2.11 Diagramme des classes qui permettent d'envoyer des notifications.....	25
Figure 2.12 Console de sonarqube.....	26
Figure 2.13 Capture d'écran du build avec mvn build package .....	27
Figure 2.14 Capture d'écran de l'installation de java sur le serveur could.....	29
Figure 2.15 Capture d'écran illustrant la vérification de l'installation .....	30
Figure 2.16 Sortie d'écran du fichier info.php.....	30
Figure 2.17 Capture d'ecran montrant la vérification.....	31
Figure 2.18 Sortie d'écran du fichier d'arrêt et de démarrage du serveur wildFly .....	31
Figure 2.19 Capture d'écran montrant la réussite .....	32
Figure 3.1 Invitation reçue par un patient.....	33
Figure 3.2 Les scores des patients dans l'application SDSSR .....	34
Figure 3.3 Notification reçue par le GC.....	34



## **LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES**

MPOC	Maladie pulmonaire obstructive chronique
BPCO	Broncho-pneumopathie chronique obstructive
ÉTS	École de technologie supérieure
ASPC	Agence de santé publique du Canada
JSF	Java Server Face
JAR	Java Archive
WAR	Web Archive
SVN	Subversion
JEE	Java entreprise edition
SDSSR	System de suivi des symptômes respiratoires
SLA	Service Level agrument (accord de niveau de service)
SSH	Secure Shell
SDSSR	System de suivi des symptômes respiratoires
FTP	File transfert protocol
VPS	Virtual private server (serveur dédié virtuel)
IP	Internet protocol
AC	Acteur
GC	Gestionnaire de cas
AJAX	Asynchronous JAvascript and Xml
JSON	JavaScript Object Notation



## **LISTE DES SYMBOLES ET UNITÉS DE MESURE**

%	Pourcentage
GHz	Gigahertz
GO	Gigaoctet
Mbps	Mégabit par secondes



## INTRODUCTION

Les maladies respiratoires chroniques sont des maladies qui affectent les voies respiratoires et d'autres parties du poumon. Les plus connues sont l'asthme, la maladie pulmonaire obstructive chronique (MPOC) et le cancer du poumon. Ces maladies sont prises très au sérieux par les gouvernements et les instituts de recherches médicales, car si elles ne sont pas soignées correctement, elles peuvent avoir des conséquences graves sur la vie du patient. C'est pourquoi les chercheurs ne cessent de multiplier les efforts afin de trouver des traitements définitifs ou du moins améliorer les conditions générales des patients en les questionnant, de façon régulière, sur leurs formes physiques ou sur leurs activités quotidiennes.

C'est dans cette optique que ce projet de recherche appliquée de 15 crédits s'inscrit. Il vise à créer un logiciel de questionnaires, en ligne, pour des patients atteints de maladies chroniques. Il vise à permettre au personnel de la santé de faire le suivi des patients. Le premier questionnaire, qui sera implémenté dans la preuve de concept, est celui qui consiste à connaître l'état du MPOC chez le patient de manière à prendre des actions préventives.

Le premier chapitre de ce rapport présentera une revue de littérature afin d'établir ce qui se fait dans le domaine des sondages en ligne et plus particulièrement ceux qui concernent le domaine médical. Il traite aussi de la problématique liée aux maladies chroniques et la raison d'être de ce projet.

Le deuxième chapitre présentera le travail qui a été fait pour mettre en place le prototype logiciel de sondage en ligne. Il présente le cœur du travail réalisé tout en abordant les activités importantes du domaine du génie logiciel dans le processus de développement du prototype logiciel (c.-à-d. les exigences, la conception et l'implémentation) ainsi que les décisions techniques qui ont été prises.

Le dernier chapitre présentera les résultats des expérimentations réalisées conjointement avec le docteur Benoit Deslaurier ainsi que les tests et les revues du travail qui ont permis d'effectuer des améliorations successives. À la fin de ce chapitre, les travaux futurs ainsi que les recommandations sont présentés.

## **Chapitre 1**

### **Revue de littérature et problématique**

Le but de ce chapitre est de faire une courte introduction de ce qui se fait dans le domaine du suivi de patients atteints de maladies chroniques, et plus précisément à l'aide de sondages en ligne. La problématique et la raison d'être de ces sondages y sont présentées. Par la suite différentes approches sont présentées telles qu'exécutées dans le domaine de la recherche clinique.

#### **1.1 Problématiques**

Les patients atteints de maladies respiratoires chroniques, telles que la BPCO (bronchopneumopathie chronique obstructive) ou l'asthme, ont besoin d'un suivi presque quotidien de la part de leurs médecins ou de leurs gestionnaires de cas (GC) (par exemple, les infirmières). L'agence de la santé publique du Canada (ASPC) estime que cette maladie est en nette progression et le nombre de personnes qui en souffrent avoisine quatre-millions en 2007 [1].

Avec ce type de maladie, l'état du patient peut se dégrader du jour au lendemain et ce dernier devra se rendre à l'hôpital afin de recevoir des soins coûteux tels que le démontre une étude réalisée par Wouters, E. F. M. (2003) [2]. Cette étude a été réalisée dans sept pays, dont le Canada. Cette étude décrit le grand impact causé par les maladies respiratoires chroniques sur le système de santé. Les dépenses avoisinent 4,000\$ par patient dont 54% à 82% de ces coûts sont associés aux frais d'hospitalisation. Cette situation est confirmée aussi dans les publications et statistiques de l'ASPC. Le rapport décrivant le fardeau économique de cette maladie au Canada (FEMC) [3] précise que les coûts des soins hospitaliers ont atteint deux-millions de dollars en 2008. À titre de comparaison, les maladies cardiovasculaires sont en premières places avec des coûts annuels estimés à cinq-millions de dollars.

Conséquemment, les médecins se tournent vers la prévention des crises par un suivi plus rigoureux du patient (c.-à-d. suivi des prises de médicaments, des activités quotidiennes, des symptômes et autres paramètres déterminés par les médecins spécialistes). Pour effectuer ce suivi, les patients doivent se rendre à l'hôpital chaque semaine pour répondre à une série de questions, généralement par le biais de questionnaires papier. Ce questionnaire aborde l'alimentation, le tabagisme, les activités physiques et interroge l'évolution de leurs symptômes ressentis lors des derniers jours. Les réponses recueillies permettent aux médecins de faire des diagnostics et ajuster, le cas échéant, la médication. Une étude réalisée dans sept hôpitaux par Jean Bourbeau; Marcel Julien; François Maltais; et al [4] démontre qu'avec la prévention et l'éducation des patients, les visites dans les urgences ont été réduites de 40% et l'aggravation des symptômes a diminué de 39.8%.

Parallèlement, ce processus de prévention et d'éducation, qui se fait typiquement à l'aide de questionnaires papier, est long et demande beaucoup d'effort de la part du personnel soignant. C'est pourquoi il est intéressant d'évaluer d'autres moyens de collecte d'information de suivi médical et de sondages. La section qui suit discute du choix de la méthode de sondage, qui joue un rôle important dans l'amélioration du processus d'études cliniques et de suivi des patients à l'aide de questionnaires.

## 1.2 Les différentes approches de sondage

Pour réaliser des sondages en ligne, de façon générale et plus particulièrement dans le domaine médical, il y a plusieurs méthodes utilisées. Chaque méthode comporte des avantages et des inconvénients. Le tableau suivant présente une comparaison entre les différentes méthodes publiées par Valérie M. Sue Lois A. Ritter [5].

Tableau 1.1 Méthodes de sondages (Valerie M. Sue Lois A. Ritter [5])

Questionnaires	Avantage	Désavantage
Par poste	<ul style="list-style-type: none"> <li>• Cout bas</li> <li>• Large portée géographique</li> </ul>	<ul style="list-style-type: none"> <li>• Faible taux de réponse</li> <li>• Longue période de</li> </ul>

	<ul style="list-style-type: none"> <li>• Pas d'intervieweur</li> <li>• L'anonymat pour les questions sensibles</li> </ul>	<ul style="list-style-type: none"> <li>• réponse</li> <li>• Questions d'urgence inefficace</li> </ul>
Par téléphone	<ul style="list-style-type: none"> <li>• réponses rapides</li> <li>• On peut poser des questions complexes</li> <li>• Large portée géographique</li> </ul>	<ul style="list-style-type: none"> <li>• Moins de lignes téléphoniques terrestres</li> <li>• Confusion avec les appels de vente</li> <li>• Intrusif</li> </ul>
Face à face	<ul style="list-style-type: none"> <li>• Bons taux de réponse</li> <li>• On peut poser des questions complexes</li> <li>• Possibilité pour entrevues plus longues.</li> </ul>	<ul style="list-style-type: none"> <li>• Portée géographique limitée</li> <li>• Long et coûteux</li> <li>• Sensible à la partialité l'intervieweur</li> <li>• Sujets délicats sont difficiles à explorer</li> </ul>
En ligne	<ul style="list-style-type: none"> <li>• Peuvent être peu coûteux</li> <li>• Efficace</li> <li>• Entrée directe de données</li> <li>• Large portée géographique</li> </ul>	<ul style="list-style-type: none"> <li>• Recours au logiciel</li> <li>• Trop d'enquêtes numériques</li> </ul>

Tel que présenté dans ce rapport, le choix d'une méthode de sondage en ligne n'est pas simple. Ce choix implique plusieurs facteurs et il peut aussi impliquer des questionnements d'ordre éthiques et sécuritaires (c.-à-d. principalement d'accès et de protection des données). En ce qui concerne ce projet de recherche, l'option de faire des sondages en ligne a été choisie dès le départ. Ce choix repose sur les facteurs suivants :

- Besoin d'un grand échantillon;
- Informations sensibles;
- Une grande répartition géographique de l'échantillon.

### **1.3 Autres considérations**

Une fois le choix de la méthode effectué, il faut aussi prendre en considération d'autres facteurs tels que :

- Le choix des questions;
- L'ordre des questions;
- L'apparence des questions;
- Nombre de questions par page;
- La possibilité d'avoir des filtres sur les questions.

Pour conclure, les Technologies de l'information (TI) peuvent apporter des améliorations notables à ce processus :

- Les patients ne se déplacent plus et peuvent avoir accès aux questionnaires des médecins plusieurs fois par semaine. Ceci permet un suivi efficace des symptômes et ainsi permet d'intervenir préventivement afin de prévenir des crises;
- À long terme, les frais relatifs au traitement de ces maladies vont diminuer.

### **1.4 Conclusion**

Le choix de mettre à la disposition des patients atteints de maladies respiratoires chroniques des questionnaires en ligne est un choix judicieux. Cette approche va permettre aux médecins de faire un bon suivi des symptômes, ce qui va permettre d'anticiper les crises par toutes sortes de moyens tels que l'ajustement des médicaments, les conseils sur les activités physiques et sur la respiration. Si ce processus permet d'améliorer la condition générale des patients, il aura aussi un impact à la fois économique et sur l'engorgement des urgences. Le chapitre suivant présente le système de sondages en ligne.

## **Chapitre 2**

### **Présentation de la solution pour l'outil de sondage en ligne**

#### **2.1 Contexte**

Afin de concevoir un prototype logiciel de sondage en ligne, le choix technologique a été arrêté sur le logiciel libre LimeSurvey [6]. Ce logiciel libre permet la création et la gestion de sondages en ligne. Ce logiciel permet principalement de :

- Créer des sondages;
- Ajouter des utilisateurs et les inviter à participer aux sondages par l'envoi de courriels;
- Visualiser les résultats des participations.

Cette décision a été prise lors d'une itération précédente de ce projet et elle est bien documentée dans le rapport de projet de l'étudiant Sébastien Dion [7]. Le logiciel libre LimeSurvey traite bien les questionnaires simples. C'est-à-dire ceux qui ne demandent aucun calcul ou personnalisation. Le questionnaire du cas d'étude sur les maladies pulmonaires obstructives chroniques (MPOC), qui est effectué dans le cadre de ce projet de recherche appliquée, nécessite un calcul personnalisé des pointages obtenus par le patient et qui déterminent la gravité de son état de santé. Pour cette raison, il est nécessaire de développer une application qui permet de pallier aux manques de LimeSurvey. Cette application a été nommée le système de suivi des symptômes respiratoire (SDSSR). Elle inclut initialement un seul questionnaire, en l'occurrence celui sur le MPOC, mais elle est conçue de manière à supporter d'autres questionnaires pour effectuer le suivi d'autres maladies.

#### **2.2 Méthodologies de travail**

La méthodologie utilisée pour développer ce prototype logiciel (c.-à-d. le SDSSR) est une méthode que l'on peut qualifier d'agile et itérative. Cela implique que l'acceptation des changements, la collaboration et l'application sont au centre des préoccupations de l'équipe

du projet. Chaque itération est caractérisée par un cycle qui commence par l'analyse, la conception, le développement, les tests et le déploiement sur le serveur. Une itération est clôturée par une revue afin de valider la valeur ajoutée de l'itération sur le logiciel final (c.-à-d. le logiciel déployé) avec le client, le docteur Benoît Deslaurier.

## **2.3 Exigences du système**

L'objectif de cette section est de fournir une description du comportement souhaité du système de sondage en ligne SDSSR pour les patients atteints de maladies chroniques. Il décrira:

- Un survol du modèle du cas d'utilisation;
- Les acteurs qui interagissent avec le logiciel;
- Les exigences fonctionnelles rédigées sous forme de cas d'utilisation;
- Les exigences non fonctionnelles;
- Les contraintes de conception.

### **2.3.1 Survol du modèle du cas d'utilisation**

La figure suivante (voir la Figure 2.1) présente le diagramme de cas d'utilisations pour le SDSSR et son interaction avec Limesurvey. Ce modèle est relativement facile à décrire et apporte plusieurs avantages pour la compréhension des exigences et la manière avec laquelle le système futur devra les satisfaire. Par ailleurs ce modèle sera aussi utile pendant les phases de conception, d'implémentation et de tests.

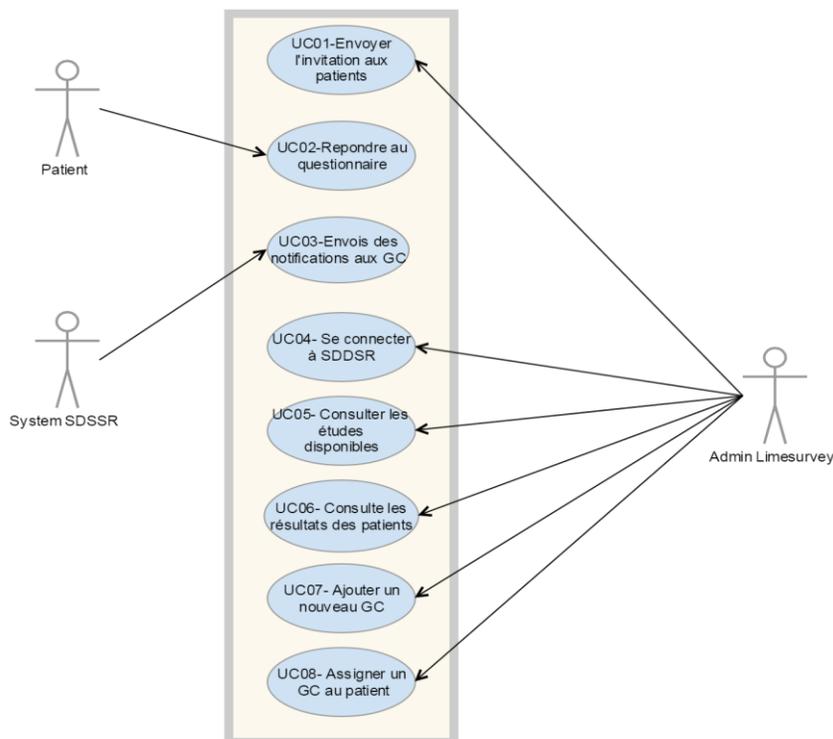


Figure 2.1 Diagramme des Cas d'utilisation

Le tableau qui suit décrit chaque cas d'utilisation présenté dans la figure 2.1.

Tableau 2.1 Descriptions des cas d'utilisations

Cas d'utilisation	Description
UC-01	L'administrateur de LimeSurvey envoie des invitations par courriel aux patients qui participent au questionnaire.
UC-02	Le patient lorsqu'il reçoit l'invitation, répond au questionnaire. Les résultats des questionnaires devraient être disponibles pour l'application SDSSR
UC-03	Des notifications, sous forme de courriel, sont envoyées aux gestionnaires de cas (médecins, infirmières). Le but est d'informer le GC que le score atteint par le score du patient a dépassé le seuil toléré et que son cas requière plus d'attention.

UC-04	Seuls les administrateurs de limeSurvey ont le droit d'accès à l'application SDSSR.
UC-05	Toutes les études (questionnaires) créées dans limesurvey, doivent être disponibles dans SDSSR.
UC-06	L'administrateur en se connectant à l'application, peut consulter les scores obtenus par les patients.
UC-07	L'administrateur en se connectant à l'application, peut ajouter un nouveau GC à l'étude.
UC-08	L'administrateur assigne un ou plusieurs GC au patient sélectionné, ceci devrait permettre au GC de recevoir les notifications qui concernent ce patient

### 2.3.2 Les acteurs

Pour le système de gestion de sondage en ligne, trois types d'acteurs ont été identifiés :

Tableau 2.2 Acteurs du système

Acteur	Description
AC-01	Le patient qui devrait répondre aux questionnaires.
AC-02	Administrateur de limeSurvey, il est le principal utilisateur de l'application SDSSR
AC-03	Le system SDDSR a pour rôle d'envoyer les notifications quand le score d'un patient dépasse le seuil toléré

### 2.2.3 Exigence fonctionnelles et non fonctionnelles

Tableau 2.3 Exigences fonctionnelles

Exigences fonctionnelles	Description
EF-01	Tous les questionnaires(ou études) dans limesurvey doivent être affichés dans l'application SDSSR.
EF-02	Seuls les administrateurs de limesurvey ont accès à l'application

	SDSSR.
EF-03	SDSSR doit envoyer un courriel de notification lorsque le résultat du patient dans le questionnaire MPOC dépasse le seuil toléré
EF-04	SDSSR doit être bilingue (Français et Anglais) Les exigences non fonctionnelles

Tableau 2.4 Exigences non fonctionnelles

Exigences Non fonctionnelles	Description
ENF-01	L'application doit être disponible à un minimum de 95% du temps.
ENF-02	L'application doit être protégée contre les accès non autorisés.
ENF-03	Préserver l'intégrité des données de l'application.

### 2.3.4 Contraintes de conception

Voici les contraintes de conceptions relatives à la conception du système SDSSR

Tableau 2.5 Contraintes de conceptions

Contrainte	Description
C01	L'application doit être en Français et en Anglais
C02	L'application doit être une application WEB

### 2.3.5 Interfaces utilisateurs

Voici le prototype de l'interface utilisateur qui a été à la base de tous les développements et évolutions itératives du projet. Cela représente le point de départ.

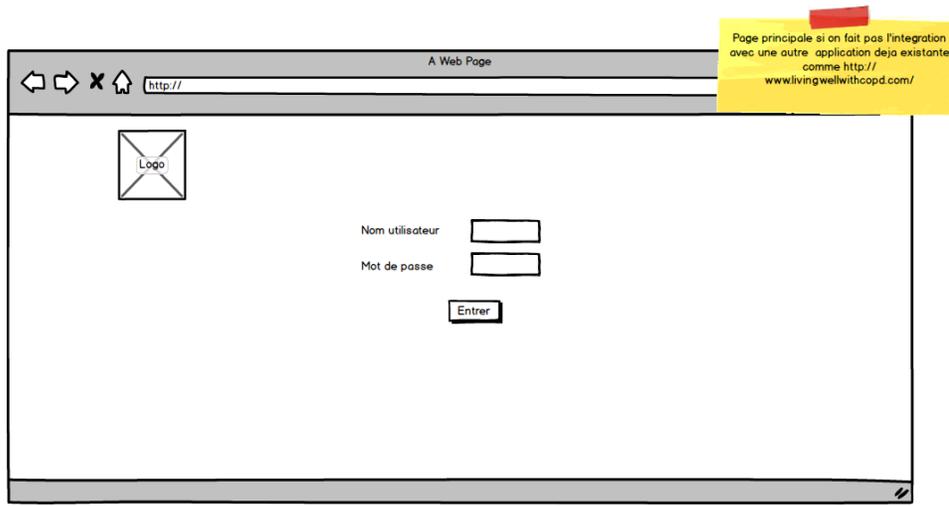


Figure 2.2 Prototype de la page de connexion

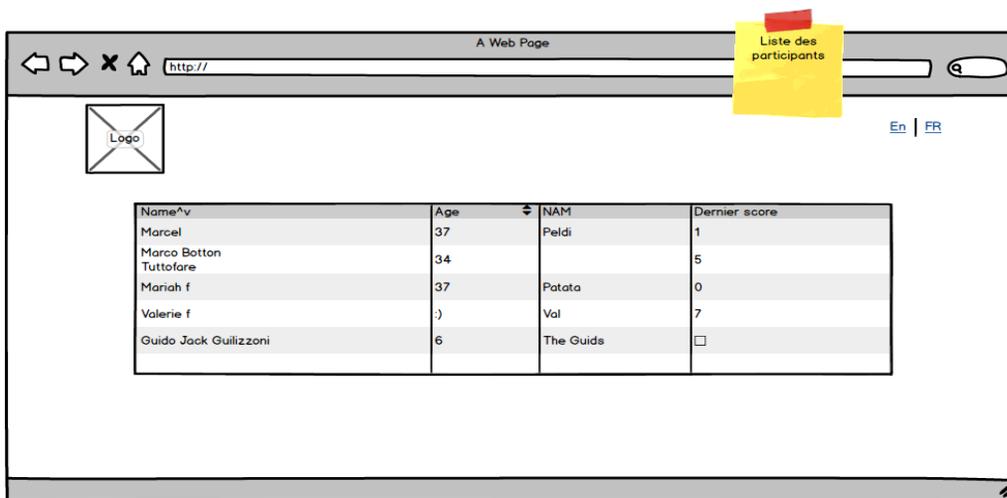


Figure 2.3 Prototype de la page des participants

Ces prototypes d'interfaces utilisateurs ont subi des modifications et des ajustements lors du processus itératif de développement.

## 2.4 Architecture de la solution

L'objectif principal de cette section est de présenter les décisions concernant l'architecture logicielle du prototype du système de suivi des symptômes respiratoires (SDSSR) selon deux points de vue principaux:

1. Composants et connecteurs : montre les éléments qui ont un comportement en mode exécution (composant) ainsi que leurs interactions (connecteurs);
2. Module : concerne la structure du système du point de vue d'organisation du code ou de données (unités d'implémentation).

Cette architecture a comme objectifs:

- Montrer les décisions de conception;
- Définir les contraintes de l'implémentation;
- Établir la base de départ qui facilite la communication entre les parties prenantes;
- Fournir un document de formation pour de nouveaux membres de l'équipe de développement, qui auront à travailler sur le projet à l'avenir, et;
- Produire un document qui va servir lors de l'évolution et la maintenance du système.

Il faut noter que cette section ne traite pas du logiciel libre LimeSurvey mais seulement de l'interaction qu'il peut y avoir avec l'application SDSSR.

### 2.4.1 Vue Composants et connecteurs

La vue composant et connecteur (C&C) est une représentation à haut niveau du prototype de l'application SDSSR et de son interaction avec Limesurvey et l'endroit où les informations seront stockées. Cette vue décrit comment les éléments du système interagissent entre eux à l'exécution (c.-à-d. le « Runtime »). La figure 2.4 présente la vue C&C pour le système SDSSR.

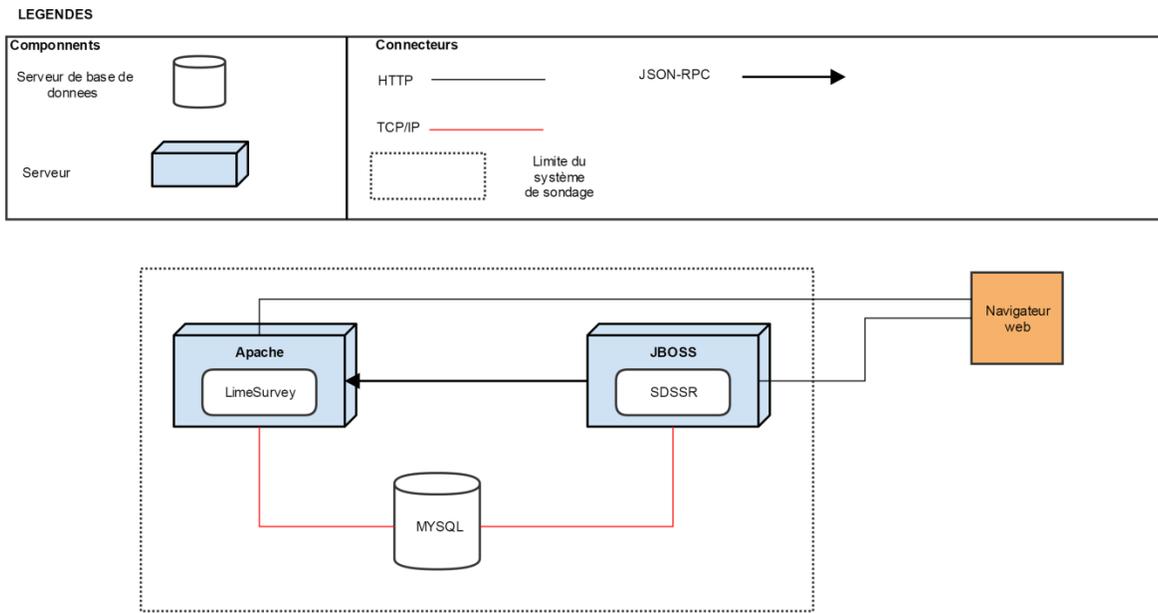


Figure 2.4 Vue Composants et connecteurs

Les tableaux 2.6 et 2.7 décrivent respectivement les composants et les connecteurs et leurs responsabilités.

Tableau 2.6 Composants et leurs responsabilités

Composant	Responsabilité
Serveur Apache	Serveur libre qui contient l'application LimeSurvey.
Serveur JBoss	Renommé WildFly, c'est le serveur JEE libre d'application qui contient SDSSR.
MySQL	Serveur de base de données. C'est l'endroit où les données seront stockées. À l'origine, elle contiendra le schéma nécessaire pour le fonctionnement de LimeSurvey. Cette proposition de solution préconise le partage avec l'application SDSSR.
Navigateur WEB	Permet à l'utilisateur de se connecter aux deux applications, qui sont compatibles avec la plupart des navigateurs Web (c.-à-d. IE, Chrome, Firefox et bien d'autres)

Tableau 2.7 Connecteur et leurs responsabilités

Connecteur	Responsabilité
HTTP	Les requêtes HTTP proviennent du client, qui est un navigateur Web. Le serveur fournira les réponses adéquates.
TCP/IP	Le protocole de transfert de données. Dans le contexte de ce projet, entre l'application (c.-à-d. dans son conteneur) et le serveur de base de données MySQL.
JSON-RPC	Le protocole d'appel de procédure à distance, encodé en format JSON. Il est utilisé pour l'authentification des utilisateurs dans le système SDSSR.

## 2.4.2 Vue modules

Le prototype d'application SDSSR sera une application Web construite avec les technologies JAVA/JEE qui possèdera trois couches logiques (voir figure 2.5):

- a) Couche de présentation;
- b) Couches de services;
- c) Couche d'accès aux données.

L'architecture proposée, en couches, offre plusieurs avantages. En effet les différentes couches peuvent évoluer et être modifiées séparément ce qui garantit au logiciel plus de portabilité, de maintenabilité ainsi que de facilité à comprendre.

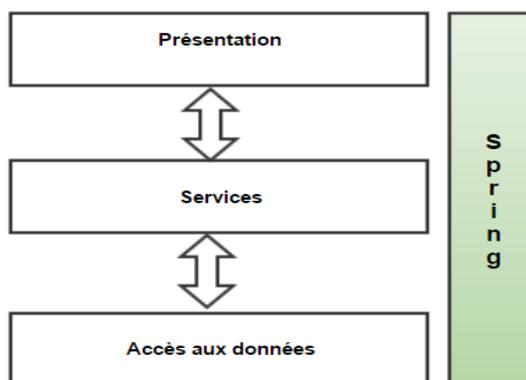


Figure 2.5 Architecture en couches

Cependant lors de la mise en place de cette architecture, il faut que chaque couche interagisse seulement avec la couche qui la suit. Pour illustrer ce concept, dans le contexte spécifique de ce projet, il est interdit d'appeler les éléments de la couche « Accès aux données » par la couche « Présentation ». La figure suivante présente une représentation des différents modules de l'application proposée (voir Figure 2.6).

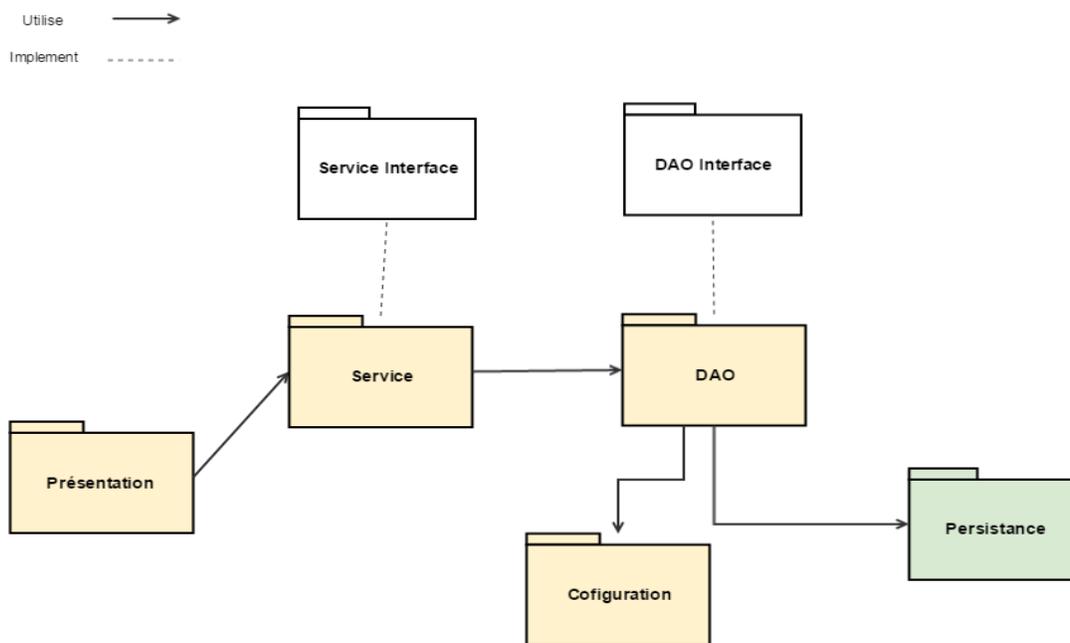


Figure 2.6 Vue en modules

Le Tableau suivant décrit chaque module présenté à la Figure 2.6 ainsi que leur responsabilité.

Tableau 2.8 Modules et leurs responsabilités

Module	Responsabilité	Autres informations
Présentation	Le module qui permet de faire l'interaction entre l'utilisateur et le système SDSSR.	La couche de présentation est construite à l'aide de technologies JSF et la librairie « primesfaces » qui contient des composants.

Service	Couche fonctionnelle de l'application, qui contient la logique métier	
Accès aux données	Le module responsable d'interagir avec la base de données	Utilise la librairie « Hibernate » pour la persistance des données, il faut souligner que dans certains cas on utilise le SQL directement.
Spring	Spring fait le lien entre les couches. Il est aussi responsable de l'injection des dépendances dans les trois couches	

Voici un diagramme de séquence qui illustre les appels planifiés entre les différents modules du système proposé:

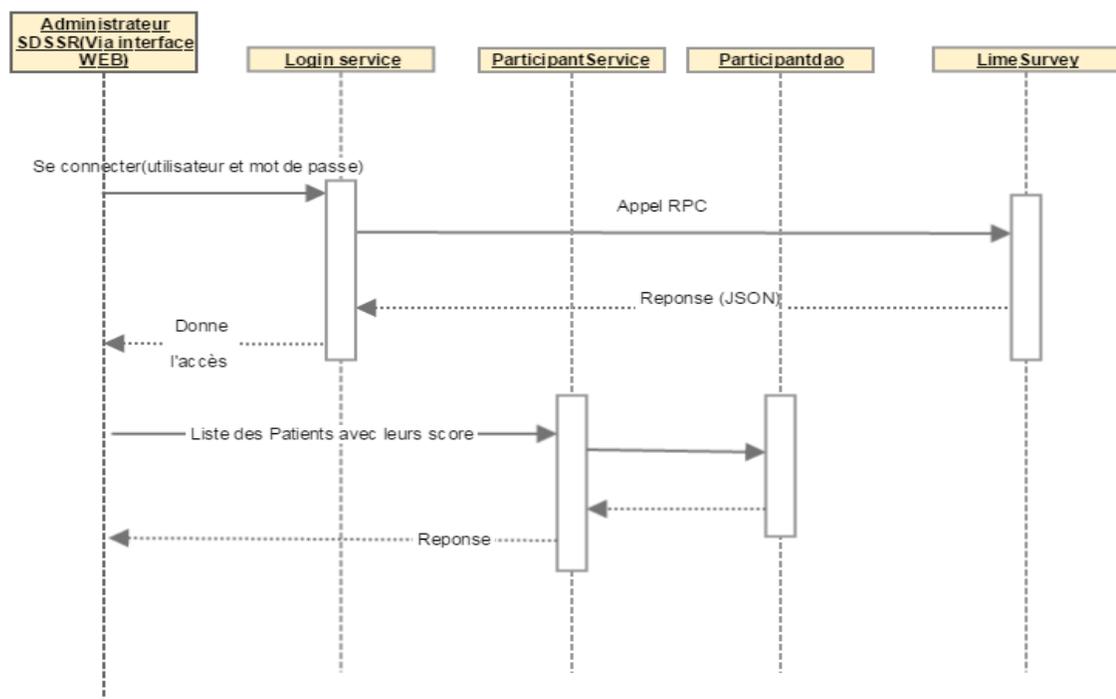


Figure 2.7 Diagramme de séquence illustrant les appels entre les modules du système

Ce diagramme illustre principalement, la manière de fonctionnement de l'application SDSSR. En effet la partie présentation devrait toujours passer par les services, ces derniers appellent à leurs tours la couche d'accès aux données, respectant ainsi le principe des appels des méthodes dans une architecture multicouches.

## 2.5 Réalisation de l'application SDSSR

Afin de réaliser ce prototype d'application, une décision a été prise pour choisir le langage de développement (c.-à-d. le langage de programmation JAVA) et d'utiliser divers cadriciels et libraires libres qui sont, pour la plupart, largement connus, utilisés et populaires auprès de la grande communauté des développeurs. Cette décision est influencée par une étude réalisée par L'IEEE qui place le langage Java parmi les trois langages les plus populaires au monde [8]. Ceci est aussi confirmé par « Google Trends » (voir Figure 2.8) qui présente la fréquence avec laquelle les termes des langages de programmation les plus connus, ont été recherchés au cours des cinq dernières années.



Figure 2.8 Fréquences de recherches des langages de programmation

La section suivante précise et introduit les cadriciels et libraires utilisées pour le projet.

### 2.5.1 Liste des principaux libraires et cadriciels

Cette section décrit les principales librairies et cadriciels proposés dans le cadre de ce projet. Ce sont tous des logiciels libres qui possèdent une grande communauté d'utilisateurs. Ils disposent aussi de documentations abondantes.

**Spring** [9]: est un cadriciel libre qui sera utilisé pour développer le prototype de l'application. Il servira pour injecter les dépendances entre les différentes classes de l'application. La version prévue est la 4.0.6.

**Hibernate** [10]: ce logiciel libre gère la persistance des objets dans les bases de données relationnelle. Pour ce projet, le choix s'est arrêté sur MySQL dans notre contexte d'application. La version prévue est la 4.3.6.

**Maven (version 3)** [11]: est un logiciel libre de la fondation Apache qui sert structurer le projet, gérer le cycle de vie (c.-à-d. compiler, tester, installer et déployer le logiciel) ainsi qu'à l'automatisation et à la production du logiciel. Dans le cadre du projet SDSSR une structure modulaire du projet sera créée, telle que le démontre la figure 2.8.

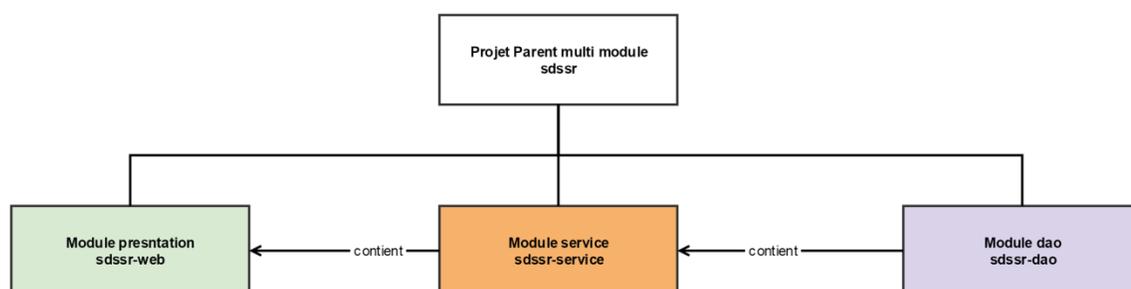


Figure 2.9 Dépendances entre modules SDSSR générés par Maven

Le module « sdssr-dao » est compilé et empaqueté à l'aide du logiciel libre Maven, sous forme de « JAR », et sera inclus dans le « sdssr-service » qui sera à son tour inclus dans l'artéfact final « sdssr web » qui lui sera finalement déployé sur le serveur JEE.

**Serveur Apache** [12]: Serveur HTTP, sur lequel sera déployée l'application LimeSurvey.

**WildFly** [13] : Anciennement Jboss. C'est un serveur libre d'application Java EE. Il va contenir le logiciel SDSSR. La version utilisée sera la 8.2.1. final.

**JSF** [14]: est un cadre Java pour le développement d'applications WEB. Dans le cadre de ce projet de recherche appliquée, la version 2.2 sera utilisée. Son but est de permettre une grande productivité en fournissant les fonctionnalités récurrentes telles que la validation, la conversion et AJAX. JSF est basé sur la notion de composants, contrairement à ceux basés sur les actions (c.-à-d. requête/réponse). Un composant est un élément d'interface graphique configurable et réutilisable. Il peut être une simple zone texte ou une table avec pagination.

**Primefaces** [15]: est une librairie de composants graphiques pour le développement d'interfaces utilisateur à l'aide de JSF. Cette librairie se caractérise par sa simplicité, sa stabilité, son support d'AJAX, ainsi que la richesse de ses composants (c.-à-d. plus de 100 composants disponibles) qui peuvent grandement accélérer le développement.

**Éclipse** [16]: est un environnement de développement libre. La figure 2.9 présente une capture d'écran qui démontre le projet SDSSR dans Éclipse. On peut y voir les modules créés par Maven ainsi que la configuration du serveur « WildFly » qui sert à déployer l'application

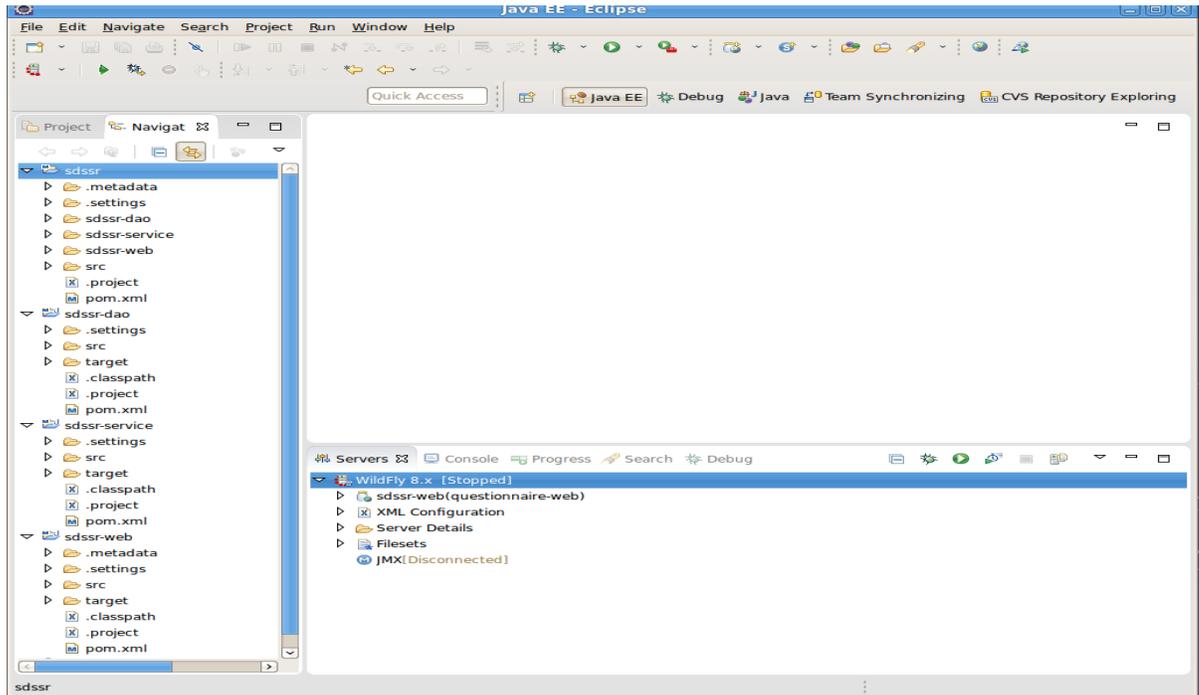


Figure 2.10 Capture d'écran présentant la hiérarchie des modules dans l'EDI eclipse

**SVN [17]** : ce logiciel libre sert pour la gestion du code source. À défaut d'avoir un serveur SVN local, la solution *Sourceforge* a été choisie. En effet, cette approche offre plusieurs avantages, tels que : la simplicité et l'utilisation d'une interface Web conviviale pour gérer le code et les droits d'utilisateurs. *Sourceforge* offre aussi la possibilité de mettre en œuvre un Wiki pour y entreposer toutes sortes d'information qui permet une meilleure collaboration entre les membres de l'équipe de développement.

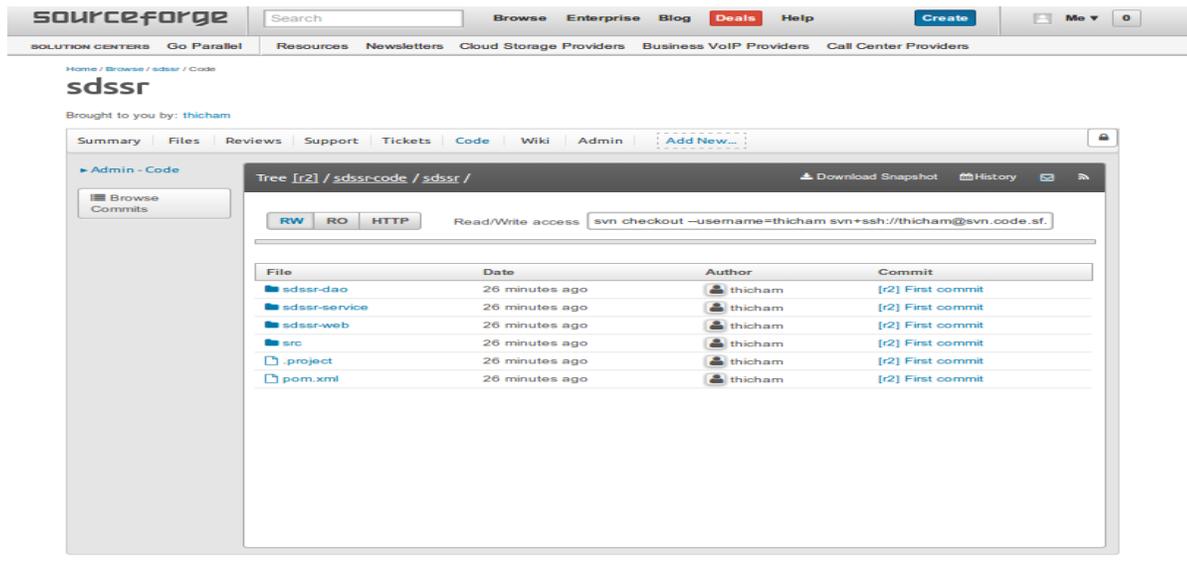


Figure 2.11 Capture d'écran montrant le code dans Sourceforge

## 2.5.2 Détails sur l'implémentation

Cette section décrit les parties importantes du code source. Elle a comme objectif de décrire comment il sera facile d'effectuer la maintenance et l'implémentation de changements futurs et aussi de permettre d'intégrer facilement de nouveaux membres de l'équipe de projet.

➤ **HibernateConfiguration** : Est la classe responsable de la configuration du logiciel Hibernate. Elle utilise les annotations suivantes :

- @Configuration
- @EnableTransactionManagement
- @ComponentScan({ "org.sdssr.configuration" })
- @PropertySource(value = { "classpath:application.properties" })

Elle lit le fichier de configuration *application.properties*, qui contient les informations nécessaires pour se connecter à la base de données.

Extrait du fichier *application.properties* :

```
#DB locale
ora.jdbc.driverClassName = com.mysql.jdbc.Driver
```

```

ora.jdbc.url = jdbc:mysql://localhost:3306/limedb?useSSL=false
ora.jdbc.password = root
ora.jdbc.username=root
#DB PROD OVH
#ora.jdbc.url = jdbc:mysql://127.0.0.1:3306/questionnaires?useSSL=false
#ora.jdbc.username = root
#ora.jdbc.password = *****
ora.hibernate.dialect = org.hibernate.dialect.MySQLDialect
ora.hibernate.show_sql = true

```

Lors de la mise en production, il faut mettre en commentaires les parties qui concernent la base de données locale et la remplacer par celle de la production, aussi il faut commenter l'option « ora.hibernate.show\_sql », car elle peut avoir des impacts sur la performance.

➤ AppConfig

Class java de configuration globale de Spring, elle utilise trois annotations :

- @Configuration
- @EnableScheduling
- @ComponentScan(basePackages = "org.sdssr")

➤ Utilisation de Spring avec les JSF

- JSF ont besoin de configuration afin d'utiliser les services de la couche « sdssr-service » qui sont développés avec Spring (contrairement aux composants JEE tels que les EJB), en effet pour utiliser ce dernier avec JSF il faut créer un fichier applicationContext.xml dans le répertoire WEB-INF, voici son contenu:

```

<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:context="http://www.springframework.org/schema/context"
xmlns:mvc="http://www.springframework.org/schema/mvc"
xmlns:tx="http://www.springframework.org/schema/tx"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

```

```

xsi:schemaLocation="
http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-4.1.xsd
http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc-4.1.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-4.1.xsd">
    <!-- For Scanning the packages and registering beans with the applicationContext-
->
    <context:component-scan base-package="sdsr" />
    </beans>

```

Par la suite il faut ajouter une nouvelle entrée dans le fichier « faces-config.xml » :

```
<el-resolver>org.springframework.web.jsf.el.SpringBeanFacesELResolver</el-resolver>
```

- Pour appeler les services de Spring à partir des *Managed Bean* de JSF, il faut annoter ces derniers par « l'annotation *component* » puis le service sera injecté avec « l'annotation *Autowired* ». Voici un extrait de la class «UserBean » qui montre bien l'injection des ressources :

```

@ManagedBean(name="userBean")
@Scope("request")
@Component
public class UserBean {
private List<User> usersList =new ArrayList<User>();
@Autowired
UserService service;

```

#### ➤ Envois des notifications

Le système doit envoyer une notification aux médecins (ou aux gestionnaires de cas) quand le score obtenu par un patient dépasse le seuil toléré. C'est pour cette raison qu'il a été nécessaire de créer une classe « ScheduledTasks » qui contient la méthode «

perofomBPCOAlert » qui devrait se déclencher chaque heure pour vérifier les nouvelles réponses et envoyer un courriel le cas échéant. Pour effectuer cette fonction avec le langage Spring, une solution simple, avec l'aide de l'annotation « Scheduled, » est proposée tel qu'il est démontré dans l'extrait de code source suivant :

```
@Scheduled(cron="0 0 * * * *")
public void perofomBPCOAlert() {
    ....
    ....
}
```

Tel que discuté, cette méthode doit connaître les messages déjà envoyés (c.-à-d. pour ne pas les envoyer plusieurs fois). C'est pour cette raison qu'il a été nécessaire de modifier le schéma de la base de données en créant trois nouvelles tables (voir Figure 2.12):

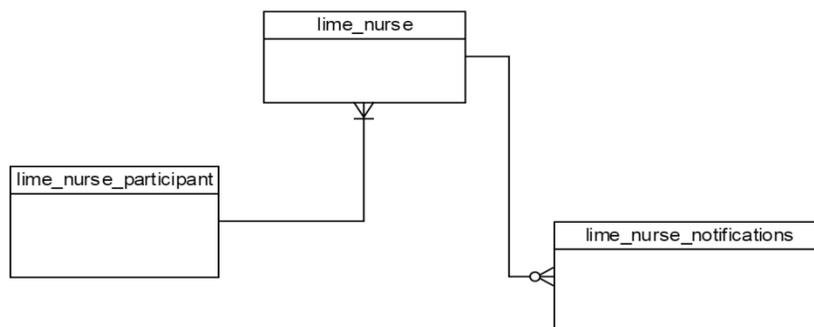


Figure 2.12 Diagramme des classes qui permettent d'envoyer des notifications

### 2.5.3 Tests et qualité

Afin de vérifier le bon fonctionnement du code, il est primordial d'effectuer des tests unitaires. Pour y arriver le cadriciel « JUnit » [18] version 4 a été utilisée. En plus d'être bien intégré avec L'environnement de développement d'Éclipse, les dernières versions ont rendu l'écriture des cas de test beaucoup plus simple grâce à l'introduction d'annotations.

Dans la même optique et afin de s'assurer de la qualité du code source, le logiciel libre Sonarqube [19] a été installé. Il offre plusieurs mesures et indicateurs tels que le nombre de

ligne de codes, les défauts potentiels, le respect des règles de codage, etc. La figure suivante montre la console de Sonarqube :

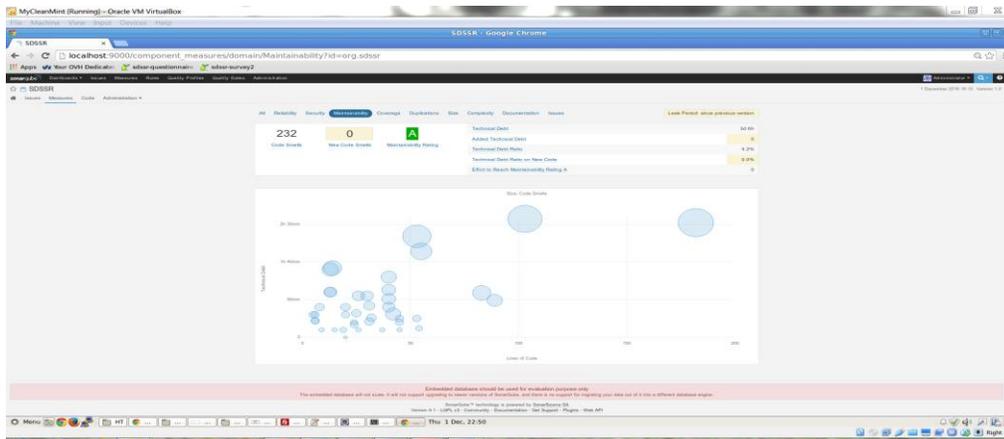


Figure 2.13 Console de sonarqube

Comme le projet SDSSR est un projet modulaire, Sonarqube a donc besoin d'un fichier de configuration. Voici son contenu :

```
# Root project information
sonar.projectKey=org.sdssr
sonar.projectName=SDSSR
sonar.projectVersion=1.0

# Some properties that will be inherited by the modules
sonar.sources=src

# List of the module identifiers
sonar.modules=sdssr-web,sdssr-service,sdssr-dao
sonar.exclusions=**/*.js

# Properties can obviously be overridden for
# each module - just prefix them with the module ID
sdssr-web.sonar.projectName=sdssr-web
sdssr-service.sonar.projectName=sdssr-service
sdssr-dao.sonar.projectName=sdssr-dao
```

## 2.6 Processus de compilation et de déploiement

Pour compiler le code, la commande « Maven » suivante est utilisée. Elle peut être exécutée à partir de ligne de commande: « *mvn build package* » ou à l'aide d'Éclipse, tel qu'il est décrit à la figure 2.14 (c.-à-d. en plus de la compilation on choisi d'exécuter les tests unitaires):

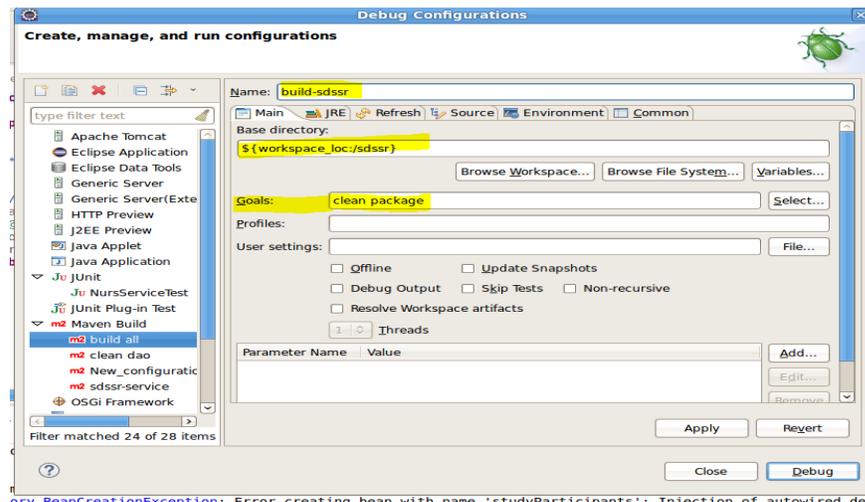


Figure 2.14 Capture d'écran du « build » avec « mvn build package»

Une fois la compilation terminée et la création du « WAR » complétée avec succès, à l'aide du logiciel « Maven », il est alors possible de procéder au déploiement d'abord sur le serveur local puis sur le serveur infonuagique de production.

### 2.6.1 Déploiement sur le serveur local

Cette opération est relativement simple. Elle peut se faire à l'aide d'Éclipse qui déploie automatiquement sur le serveur local. Éclipse permet aussi de démarrer le serveur (ou bien l'arrêter) en mode normal ou en mode de débogage. Pour se faire, il faut télécharger le serveur « WildFly », et le décompresser dans un répertoire du disque et faire pointer Éclipse vers ce dernier.

## 2.6.2 Déploiement en production sur le serveur Cloud

Pour le déploiement en production, deux options sont possibles. La première étant le déploiement sur les serveurs de l'ÉTS mais des difficultés d'ordre logistique nous amène à utiliser la seconde option qui est le déploiement sur le nuage. En effet cette option offre une multitude d'avantages, tels que : la garantie d'avoir une haute disponibilité de l'application, la sauvegarde à intervalles réguliers des données et du statut du serveur. Ceci permet de se concentrer seulement sur les tâches de déploiement et de configuration des applications.

Pour le choix du fournisseur du service infonuagique, plusieurs fournisseurs ont été étudiés tels que : Amzon, Godaddy, Google, hostpapa et OVH. Le choix final a été OVH car il correspond aux critères recherchés pour cette étape du projet en termes de services offerts et de coûts, en plus d'offrir l'avantage d'avoir l'hébergement des serveurs à Montréal.

Tableau 2.9 Caractéristiques du serveur Cloud d'OVH

Caractéristiques	KVM openstak 1 vCore 2,4 GHz 2 Go mémoire vive SSD 10 Go
System d'exploitation	Ubuntu 16.04 LTS (GNU/Linux 4.4.0.45-generic x86_64)
Nom du serveur VPS	vps91586.vps.ovh.ca
IP VPS	158.69.208.55
SLA	99,95% (à peu près 5h d'arrêt de service toléré pendant une année).
Bande passante	100Mbps- Traffic illimité.
Gestion	Espace client disponible qui permet l'arrêt, le redémarrage ou la surveillance de la machine. Accès root a la machine.

Surveillance	Depuis l'espace client.
--------------	-------------------------

### 2.6.3 Installation des logiciels nécessaire pour procéder au déploiement

Cette section décrit les étapes importantes pour l'installation et la configuration des logiciels nécessaires au bon fonctionnement des applications (pour plus de détails concernant leurs installations, veuillez consulter les sites officiels). Ces étapes ont pour but de faire comprendre et reproduire l'environnement de production sur le serveur actuel ou sur un autre serveur au besoin :

- Installation et configuration sur la machine locale d'un client SSH (à l'aide de Putty) et d'un client FTP(à l'aide de FileZilla). Ces logiciels sont importants, car ils vont permettre de configurer et de transfert des fichiers sur le serveur distant;
- Installation de java : se connecter sur le serveur distant en tant que « Root » et exécuter la commande suivante : « *sudo apt-get install openjdk-8-jdk* »

```

root@vps91586:~# sudo apt-get install openjdk-8-jdk
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
ca-certificates-java dbus-x11 fontconfig fontconfig-config fonts-dejavu-core font
libatk1.0-0 libatk1.0-data libavahi-client3 libavahi-common-data libavahi-common3
libflac8 libfontconfig1 libgconf-2-4 libgdk-pixbuf2.0-0 libgdk-pixbuf2.0-common
libgtk2.0-bin libgtk2.0-common libharfbuzz0b libice-dev libice6 libjpeg8 libjpeg
libpciaccess0 libpcsclite1 libpixmap-1-0 libpthread-stubs0-dev libpulse0 libsm-de
libxau-dev libxcb-dri2-0 libxcb-dri3-0 libxcb-glx0 libxcb-present0 libxcb-render
libxshmfence1 libxt-dev libxt6 libxtst6 libxxf86vm1 openjdk-8-jdk-headless openj
Suggested packages:
gconf-defaults-service default-jre libasound2-plugins alsa-utils libbonobo2-bin
libice-doc liblcms2-utils pcscd pulseaudio libsm-doc libxcb-doc libxt-doc openjdk
| ttf-wqy-zenhei fonts-indic
The following NEW packages will be installed:
ca-certificates-java dbus-x11 fontconfig fontconfig-config fonts-dejavu-core font
libatk1.0-0 libatk1.0-data libavahi-client3 libavahi-common-data libavahi-common3
libflac8 libfontconfig1 libgconf-2-4 libgdk-pixbuf2.0-0 libgdk-pixbuf2.0-common

```

Figure 2.15 Capture d'écran de l'installation de java sur le serveur Cloud

Suite à l'installation d'openjdk version8, il faut installer Apache, Mysql et PHP qui sont nécessaires au fonctionnement du logiciel libre LimeSurvey :

- Installer le serveur apache sur le serveur et entrer les commandes suivantes :
  - o `sudo apt-get update`
  - o `sudo apt-get install apache2`

Pour s'assurer que le serveur a été bien installé, vous pouvez accéder avec l'URL suivant `vps91586.vps.ovh.ca` (ou `158.69.208.55`) à partir d'un navigateur Web.

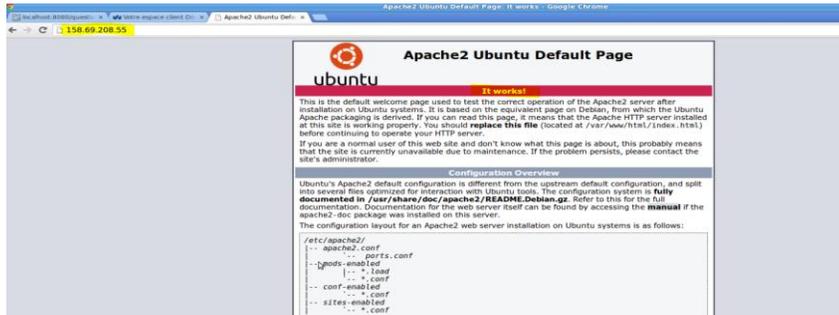


Figure 2.16 Capture d'écran illustrant la vérification de l'installation du serveur Apache

- Installer le serveur de base de données Mysql :
  - o `sudo apt-get install mysql-server php5-mysql`
- Installer PHP :
  - o `apt-get install php libapache2-mod-php php-mcrypt php-mysql`
  - o `sudo systemctl restart apache2`
  - o Il est recommandé d'installer php myadmin: `apt-get install phpmyadmin`

Pour vérifier que PHP a bien été installé on peut créer un fichier `info.php` dans le répertoire du serveur apache `/var/www/html`

```

root@vps91586: /var/www/html
root@vps91586:/var/www/html# ll
total 36
drwxr-xr-x 3 root root 4096 Sep  2 23:42 ./
drwxr-xr-x 3 root root 4096 Aug 21 15:53 ../
-rw-r--r-- 1 root root 11321 Aug 21 15:53 index.html
-rw-r--r-- 1 root root 20 Aug 21 16:23 info.php
drwxr-xr-x 19 root root 4096 Aug 21 18:07 subss-questionnaires/
-rw-r--r-- 1 root root 206 Aug 27 12:15 testmail.php
-rw-r--r-- 1 root root 1024 Aug 27 13:33 .testmail.php.swp
root@vps91586:/var/www/html# cat info.php
<?php
phpinfo();
?>
root@vps91586:/var/www/html#

```

Figure 2.17 Sortie d'écran du fichier `info.php`

Puis, accédez à L'URL « `vps91586.vps.ovh.ca/info.php` » pour vérifier le bon fonctionnement. Si tout se passe correctement, le résultat suivant apparaîtra:



- Pour vérifier que le serveur est bien installé exécuter la commande pour démarrer le serveur : `./standalone.sh`



Figure 2.20 Capture d'écran montrant la réussite de l'installation du serveur wildFly

## 2.7 Conclusion

Ce chapitre a été consacré à la conception et au développement de l'application de sondage en ligne pour les malades qui souffrent de maladies respiratoires chroniques. Ce processus a commencé par la définition des besoins et s'est achevé par le déploiement de l'application dans un serveur infonuagique de la firme OVH tout en suivant une méthode de développement agile et itérative. Le chapitre suivant qui présente l'expérimentation du prototype logiciel de sondage ainsi que les résultats obtenus.

## Chapitre 3

### Résultats et analyses

Ce chapitre présente la synthèse du travail qui a été réalisé ainsi que les résultats des expérimentations qui ont été validées avec le docteur Benoît Deslauriers. Il est suivi d'une revue critique afin de décrire ce qui reste à faire et les actions à prendre pour améliorer ce premier prototype logiciel.

#### 3.1 Résultats

Voici un sommaire de ce qui a été réalisé dans ce projet de recherche appliquée:

- OVH a été choisi à titre de fournisseur de serveur infonuagique;
- Suite à une étude approfondie, suivi d'essais en mode local, l'application LimeSurvey a été installée et configurée avec succès sur le serveur infonuagique;
- Le questionnaire MBOC a été créé à l'aide du logiciel LimeSurvey;
- Des utilisateurs « tests » ont été créés et incorporés dans le questionnaire. Des invitations leur ont été envoyées, via Limesurvey, tel que le démontre la figure suivante :

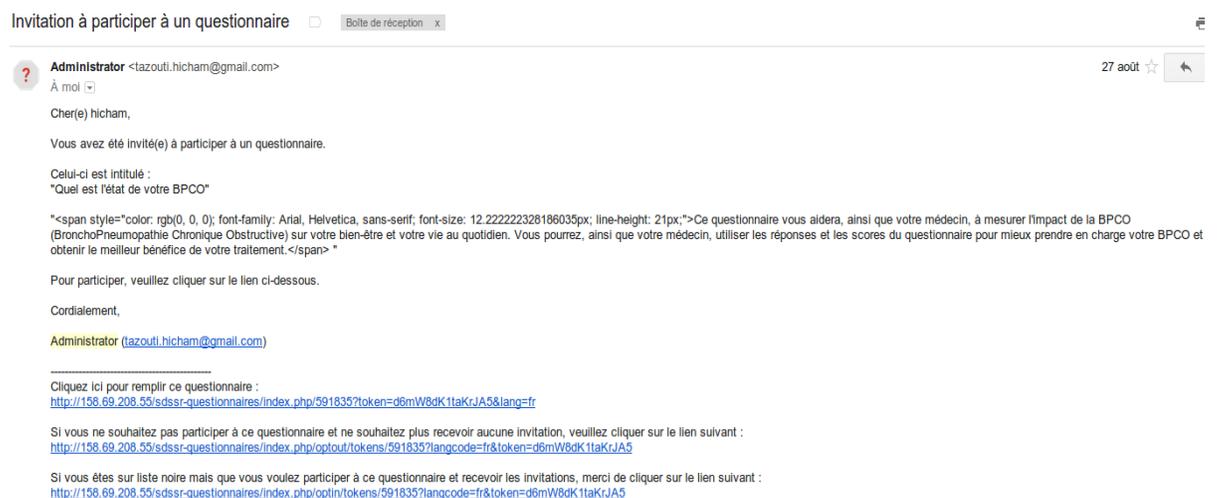
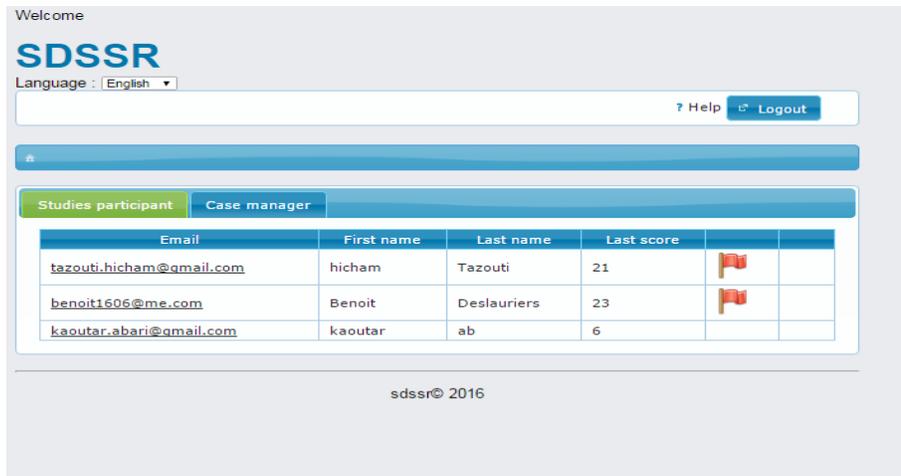


Figure 3.1 Invitation reçue par un patient test

- L'application SDSSR a été développée et déployée sur le serveur JEE dans l'environnement infonuagique;
- Les utilisateurs d'essais ont répondu aux questionnaires et leurs scores sont disponibles dans SDSSR tel que le démontre la figure suivante :



Welcome

**SDSSR**

Language : English

Help Logout

Studies participant Case manager

Email	First name	Last name	Last score		
<a href="mailto:tazouti.hicham@gmail.com">tazouti.hicham@gmail.com</a>	hicham	Tazouti	21		
<a href="mailto:benoit1606@me.com">benoit1606@me.com</a>	Benoit	Deslauriers	23		
<a href="mailto:kaoutar.abari@gmail.com">kaoutar.abari@gmail.com</a>	kaoutar	ab	6		

sdssr© 2016

Figure 3.2 Les scores des patients dans l'application SDSSR

Lorsque le score dépasse le seuil toléré, une notification est alors envoyée au GC tel que le démontre la capture d'écran suivante:

Alert: Patient with high average / Alert: Patient avec une moyenne élevée

Corbeille x

 [tazouti.hicham@gmail.com](mailto:tazouti.hicham@gmail.com)  
À moi

 Ce message n'a peut-être pas été envoyé par : [tazouti.hicham@gmail.com](mailto:tazouti.hicham@gmail.com) [En savoir plus](#) [Signaler comme hameçonnage](#)

The final score of this patient is very high / Le dernier score de ce patient est très élevée:  
Name/Nom: hicham Tazouti  
Last Score/Dernier score:21

Figure 3.3 Notification reçue par le GC

Le tableau suivant présente un sommaire des applications nécessaires au fonctionnement du prototype logiciel.

Tableau 3.1 Accès aux applications installées

Application	Accès
SDSSR	<a href="http://vps91586.vps.ovh.ca:8080/sdssr/">http://vps91586.vps.ovh.ca:8080/sdssr/</a>
LimeSurvey	<a href="http://vps91586.vps.ovh.ca/sdssr-questionnaires/index.php/admin/authentication/sa/login">http://vps91586.vps.ovh.ca/sdssr-questionnaires/index.php/admin/authentication/sa/login</a>
Console OVH	<a href="https://ca.ovh.com/">https://ca.ovh.com/</a>
Phpmyadmin	<a href="http://vps91586.vps.ovh.ca/phpmyadmin/">http://vps91586.vps.ovh.ca/phpmyadmin/</a>
Sourceforge	<a href="https://sourceforge.net/p/sdssr/">https://sourceforge.net/p/sdssr/</a>

Pour obtenir ce résultat, il a fallu faire des expérimentations successives. Chaque itération a du être validée par le docteur Benoit Deslauriers. À chaque itération des ajustements ont été nécessaires. La section qui suit décrit ces ajustements.

### 3.2 Expérimentation et validation

Au début du projet, une version de Limesurvey a été installée sur le serveur local et un questionnaire MPOC a été configuré puis envoyé au Docteur Benoit Deslaurier afin d'effectuer des tests. Des ajustements ont été faits principalement en ce qui concerne la langue. En effet le questionnaire devait être en français et en anglais. De plus, pendant la durée du projet, une nouvelle version (c.-à-d. la version 2.5) de Limesurvey a vu le jour. Ceci a impliqué une mise à jour sans grandes conséquences sur la version locale.

La deuxième itération a visé la création de l'application qui calcule les scores de chaque participant au questionnaire. C'est ce qui a été fait dans l'application SDSSR. La troisième itération concerne le déploiement des applications Limesurvey et SDSSR avec la prise de décision de déployer dans le serveur infonuagique. Cette itération a nécessité plusieurs activités :

- Investigation sur les fournisseurs infonuagiques;

- Expérimentation en testant quelques fournisseurs pendant leurs périodes d'essai;
- Contrat de trois mois puis d'une année avec le fournisseur OVH;
- Configuration du serveur infonuagique;
- Déploiement de Limesurvey et SDDR.

Lors de la dernière itération, le concept de gestion de la configuration et des notifications ont été introduits. Une version finale du prototype de l'application SDSSR a été déployée sur le serveur infonuagique de OVH. Tel qu'il a été mentionné auparavant, le système de sondage en ligne a été déployé, mais comme tout prototype logiciel, il contient des imperfections qui peuvent être améliorées. La section suivante discute des améliorations possibles.

### 3.3 Recommandations

La solution développée est loin d'être une version finale de production. Il y a des améliorations qui devraient être effectuées lors d'une prochaine itération :

- 1- Amélioration du code existant :
  - a. Inclure une charte graphique du projet qui inclut un logo, et qui devrait garantir une cohérence;
  - b. Améliorer les logs de l'application, afin de faire un suivi efficace des erreurs;
  - c. L'application SDSSR peut être consultée via les appareils mobiles, mais pas d'une façon optimale. Il est recommandé donc soit d'utiliser la librairie bootstrap [20] ou de se procurer une licence des thèmes de la librairie primefaces.
- 2- Amélioration de la sécurité de l'application SDSSR :

Il serait important d'ajouter des fonctionnalités pour la gestion des mots de passe et de noms d'utilisateur oubliés.
- 3- Procédure de déploiement et environnement de test :

À l'avenir, il serait souhaitable d'automatiser la procédure de déploiement. Ceci pourrait se faire à l'aide du logiciel « Maven » ou à l'aide d'un simple script.

Dans le cas où le projet prendrait de l'ampleur, il serait préférable d'utiliser un outil d'automatisation et d'intégration continue tel que « Jenkins » [21]. Il serait aussi important de créer un environnement de test utile et essentiel avant de déployer en production (c.-à-d. sur le serveur infonuagique).

4- Procédure de maintenance :

Étant donné que le prototype de génération de sondage est en production, il est important d'avoir une procédure de maintenance qui devrait fournir des réponses lorsque des problèmes ou des incidents surviennent. Ceci est important dans le but de les corriger. Des logiciels libres peuvent être installés, tels que Mantis [22] ou Trac [23].

5- Mise à jour de Limesurvey :

Pour mettre à jour Limesurvey, le moyen le plus efficace est d'utiliser l'utilitaire « confortUpdate ». Pour pallier aux problèmes qui peuvent survenir lors de la mise à jour, il est nécessaire de créer un script qui fait une copie de sécurité de la version qui va être mise à jour ainsi que de sa base de données MySQL.

## CONCLUSION

L'objectif de ce projet était de réaliser un outil de sondage en ligne pour les personnes atteintes de maladies chroniques afin de permettre aux médecins d'améliorer la santé de ces malades qui souffrent quotidiennement de ces maladies. Par ailleurs ce projet pourrait avoir un impact non négligeable sur le plan économique au niveau du système de la santé et sur le désengorgement des urgences.

Pour y arriver, il a fallait résoudre plusieurs problématiques, telles que : la compréhension et l'interconnexion à Limesurvey, la définition des exigences du projet, la conception et la réalisation de prototypes successifs et enfin tout ce qui concerne les essais et le déploiement de l'application sur une plateforme infonuagique.

Sur le plan personnel, ce projet m'a permis de mettre en pratique mes connaissances en ingénierie des TIs acquises tout au long de mon cursus de maîtrise à l'ÉTS. Il m'a aussi permis d'acquérir d'autres connaissances en conception et en programmation, par exemple, le JSF, le Spring, l'utilisation de Maven et de Hibernate qui sont considérés parmi les technologies les plus utilisées dans le développement d'applications Web. Mais je suis heureux surtout d'avoir participé à un projet qui a comme objectif de faire progresser la recherche médicale et aider les patients dans la vie quotidienne. Finalement, ce projet n'en est qu'à ses débuts avec cette première version fonctionnelle d'une première version de son prototype logiciel..

## ANNEXE I

### Exemple du questionnaire MPOC

Ce questionnaire vous aidera, ainsi que votre professionnel de la santé, à évaluer l'effet de la MPOC (Maladie pulmonaire obstructive chronique) sur votre bien-être et votre vie quotidienne. Vos réponses et le score du test peuvent être utilisés par vous et votre professionnel de la santé afin d'aider à améliorer la prise en charge de votre MPOC et à bénéficier au maximum du traitement.

Si vous souhaitez remplir le questionnaire à la main sur papier, [veuillez cliquer ici](#), puis imprimer le questionnaire.

Si vous remplissez le questionnaire en ligne, pour chaque question ci-dessous, cliquez à l'aide de votre souris pour cocher la case (X) qui correspond le mieux à votre état actuel.

Exemple: Je suis très heureux (heureuse)  0  1  2  3  4  5 Je suis très triste

			SCORE
Je ne tousse jamais	<input type="radio"/> 0 <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5	Je tousse tout le temps	<input type="text"/>
Je n'ai pas du tout de sécrétions bronchiques dans mes poumons	<input type="radio"/> 0 <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5	J'ai les poumons complètement encombrés de sécrétions bronchiques	<input type="text"/>
Je ne ressens aucune oppression respiratoire	<input type="radio"/> 0 <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5	Je ressens une forte oppression respiratoire	<input type="text"/>
Je ne suis pas essoufflé(e) quand je monte une côte ou un étage	<input type="radio"/> 0 <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5	Je suis très essoufflé(e) quand je monte une côte ou un étage	<input type="text"/>
Je ne suis pas limité(e) dans l'accomplissement de mes activités domestiques	<input type="radio"/> 0 <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5	Je suis très limité(e) dans l'accomplissement de mes activités domestiques	<input type="text"/>
Je ne suis pas inquiet(ète) quand je quitte la maison, en dépit de mes problèmes pulmonaires	<input type="radio"/> 0 <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5	Je suis très inquiet(ète) quand je quitte la maison, à cause de mes problèmes pulmonaires	<input type="text"/>
Je dors bien	<input type="radio"/> 0 <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5	Je ne dors pas bien à cause de mes problèmes pulmonaires	<input type="text"/>
J'ai beaucoup d'énergie	<input type="radio"/> 0 <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5	Je n'ai pas d'énergie du tout	<input type="text"/>
<input type="button" value="Cliquez pour obtenir le total de votre score"/>			<input type="text"/>

Figure-A I-1 Questionnaire MPOC



## ANNEXE II

### Procédure pour l'ajout et la modification du questionnaire dans LimeSurvey

Voici les étapes pour modifier un questionnaire ou créer un nouveau :

- 1- Se connecter sur l'application LimeSurvey (<http://158.69.208.55/sdssr-questionnaires/index.php/admin/authentication/sa/login>)

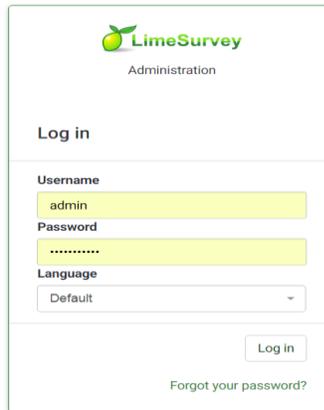


Figure-0-1 II-1 Écran d'accueil limesurvey

Si l'authentification réussit, on peut alors accéder à la page principale de l'application LimeSurvey, qui présente plusieurs options comme la création ou modification des questionnaires, la configuration globale ou la modification et la personnalisation des paramètres d'affichage.



Figure-A II-2 Page de configuration limesurvey

- 2- Pour modifier un questionnaire existant, cliquer sur Liste des questionnaires puis sélectionner le questionnaire que vous voulez modifier (ici MBCO) comme le démontre les deux captures suivantes :

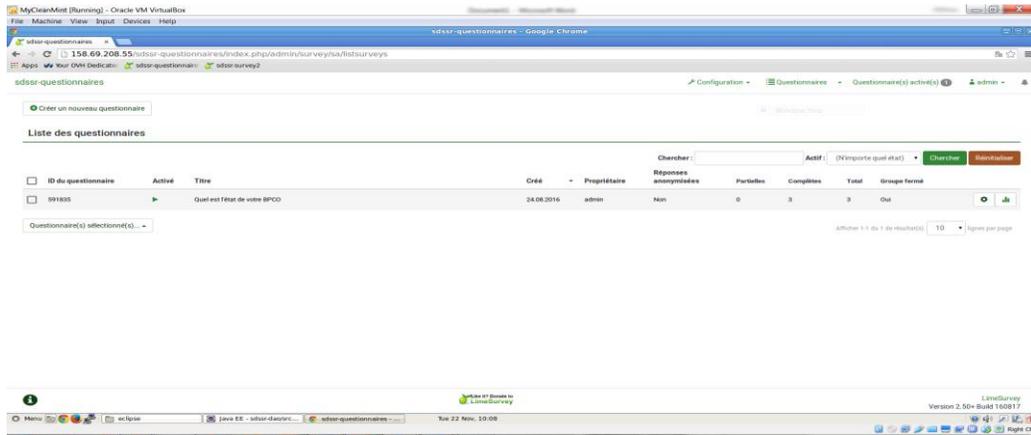


Figure-A II-3 Ensemble des questionnaires

Il faut choisir l'option « Liste des questions », pour accéder à l'ensemble des questions qui forme le questionnaire.

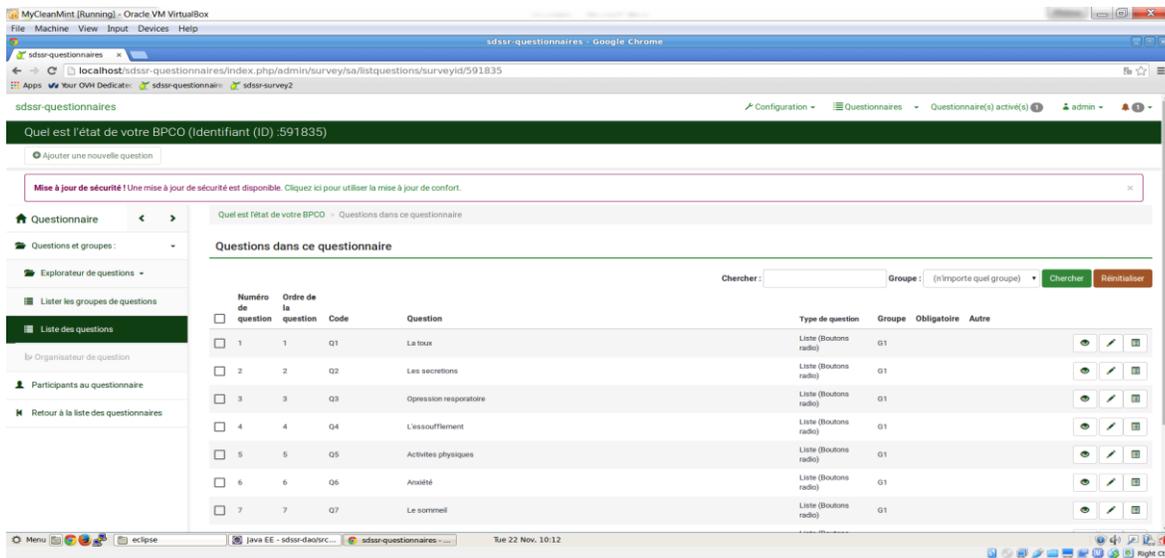


Figure-A II-4 Écran des questions

Trois options se présentent pour chaque question :

- Modifier l'entête de la question;
- Visualiser la question;
- Voir le résumé de la question.

Pour pouvoir modifier le contenu il faut avant tout désactiver le questionnaire et accéder à la question un menu se présente (capture suivante) qui permet de réaliser cette opération :



Figure A II.4 Option pour modifier les questions

Important :

- avant de désactiver le questionnaire, il faut s'assurer d'en faire une sauvegarde.
- après avoir modifié le questionnaire, il faut l'activer pour qu'il soit accessible pour les utilisateurs.

### 3- Pour ajouter un nouveau questionnaire

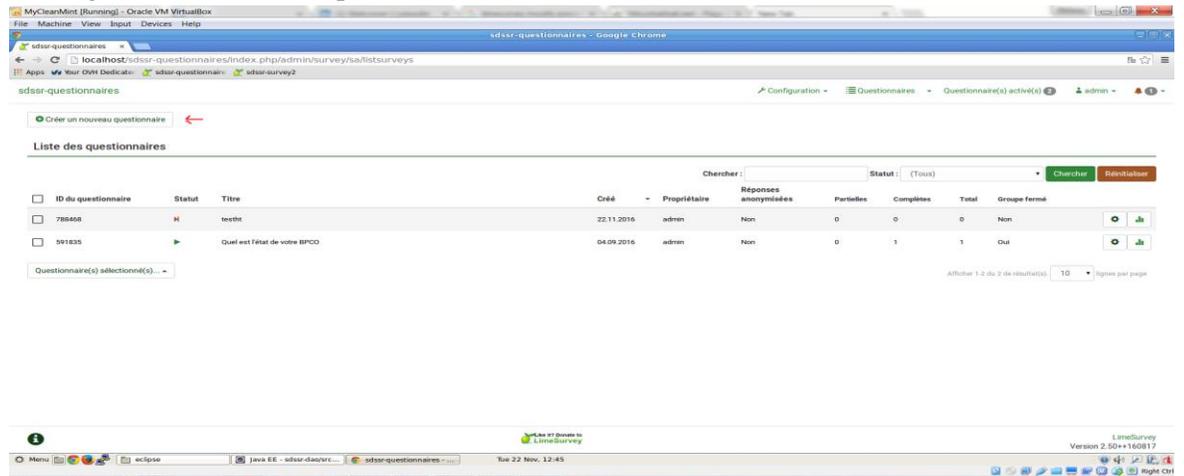


Figure-A II-5 Ajout d'un nouveau questionnaire

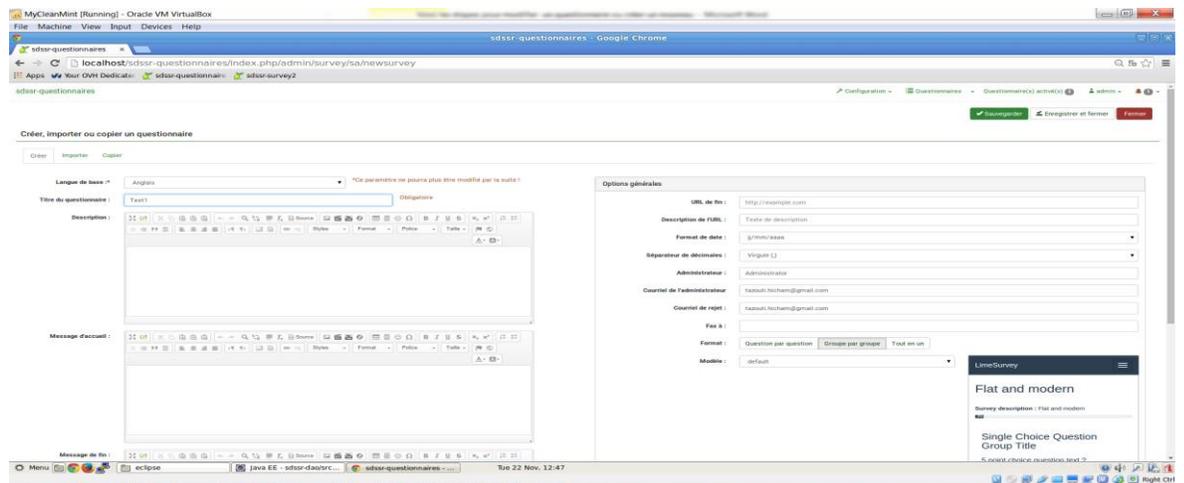
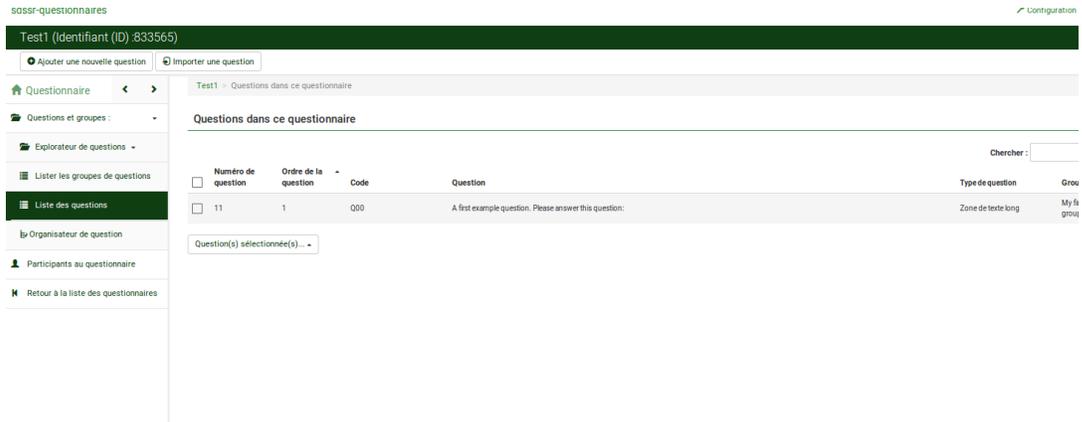


Figure-A II-6 Détails du questionnaire

L'étape suivante est d'accéder au questionnaire afin de créer les questions, il est aussi possible de les importer depuis un autre :



The screenshot shows a web interface for managing a questionnaire titled 'Test1 (Identifiant (D) 833565)'. The interface includes a sidebar with navigation options and a main content area displaying a table of questions.

Navigation options in the sidebar:

- Questionnaire
- Questions et groupes
- Explorateur de questions
- Lister les groupes de questions
- Liste des questions (highlighted)
- Organisateur de question
- Participants au questionnaire
- Retour à la liste des questionnaires

Main content area: Questions dans ce questionnaire

Search bar: Chercher :

<input type="checkbox"/>	Numéro de question	Ordre de la question	Code	Question	Type de question	Groupe
<input type="checkbox"/>	11	1	Q00	A first example question. Please answer this question:	Zone de texte long	My file group

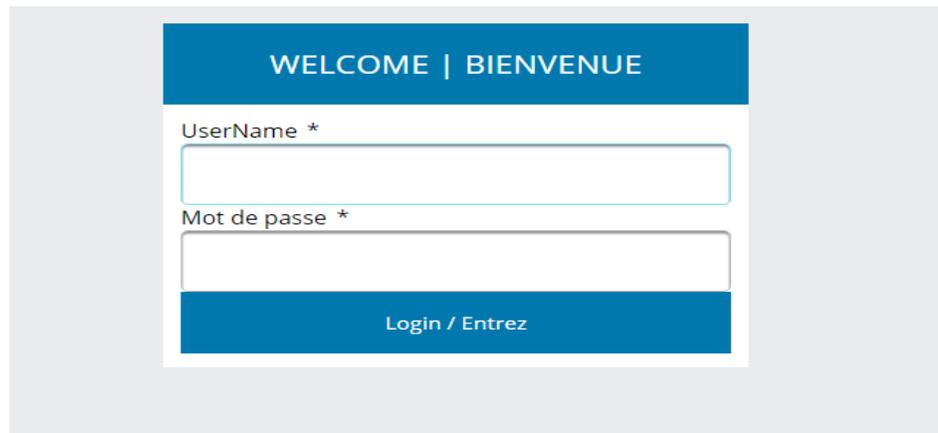
Question(s) sélectionnée(s) : -

Figure-A II-7 Ajout de nouvelles questions

## ANNEXE III

### Consultation des résultats avec l'application SDSSR

1. Se connecter SDSSR (le nom d'utilisateur et le mot de passe est celui des administrateurs limesurvey) <http://158.69.208.55:8080/sdssr/>



WELCOME | BIENVENUE

UserName \*

Mot de passe \*

Login / Entrez

Figure-A III-1 Écran de connexion SDDS

2. L'écran d'accueil de l'application SDSSR



Welcome admin

**SDSSR**

Language : English

Se déconnecter

Liste des études

Id	Titre	Langue
591835	Quel est l'état de votre BPCO	fr

sdssr© 2016

Figure-A III-2 Écran principal

Pour l'instant il n'y a un seul questionnaire, celui qui concerne l'état du BPCO.

3. En cliquant sur le numéro du questionnaire (généré par LimeSurvey) on accède à la liste des patients recrutés pour cette étude

Courriel	Prénom	Nom	Dernier score		
<a href="mailto:tazouti.hicham@gmail.com">tazouti.hicham@gmail.com</a>	hicham	Tazouti	21		
<a href="mailto:benoit1606@me.com">benoit1606@me.com</a>	Benoit	Deslauriers	23		
<a href="mailto:kaoutar.abari@gmail.com">kaoutar.abari@gmail.com</a>	kaoutar	ab	6		

sdssr© 2016

Figure-A III-3 Écran montrant les participants

La capture d'écran montre le score obtenu par chaque patient, dans le cas où le score dépasse le seuil toléré un drapeau rouge est alors affiché afin d'avertir l'utilisateur.

L'onglet gestionnaire de cas permet d'ajouter des personnes (médecins, infirmiers..) qui ont comme tâche de faire le suivi avec les patients.

**Add Case manager**

Nom Complet

Courriel

Nom Complet	Courriel
benoit	benoit1606@me.com
CM1	tazouti.hicham@gmail.com

Figure-A III-4 Écran d'ajout d'un nouveau CM

4. On choisit un patient pour consulter l'historique de ses scores obtenus

Score	Date
21	2016-10-25 00:23:13.0

sdssr© 2016

Figure-A III-5 Historique des scores

Il est aussi possible d'assigner des gestionnaires des cas au patient, ces derniers devraient recevoir une notification quand le score du patient dépasse le seuil toléré

Disponibles	Aassignés
CM1	benoit

Figure-A III.6 Assigner un CM au patient



## BIBLIOGRAPHIE

- [1] «Agpc la vie et le souffle,» [En ligne]. Available: <http://www.phac-aspc.gc.ca/publicat/2007/lbrdc-vsmrc/index-fra.php>.
- [2] E. Wouters, «Economic analysis of the confronting COPD survey: an overview of results,» 2003. [En ligne]. Available: [http://dx.doi.org/10.1016/S0954-6111\(03\)80020-3](http://dx.doi.org/10.1016/S0954-6111(03)80020-3).
- [3] «Le fardeau économique de la maladie au Canada(FEMC),» [En ligne]. Available: <http://www.phac-aspc.gc.ca/publicat/ebic-femc/2005-2008/assets/pdf/ebic-femc-2005-2008-fra.pdf>.
- [4] J. v. d. P. J. Bourbeau, «Promoting effective self-management programmes to,» [En ligne]. Available: <http://erj.ersjournals.com/content/erj/33/3/461.full.pdf>.
- [5] V. M. Sue, «Planning the Online Survey,» [En ligne]. Available: [https://us.sagepub.com/sites/default/files/upm-binaries/44130\\_2.pdf](https://us.sagepub.com/sites/default/files/upm-binaries/44130_2.pdf).
- [6] «Limesurvey,» [En ligne]. Available: <https://www.limesurvey.org/>.
- [7] S. Dion, «SondaHGJ - Générateur de sondages en ligne, Rapport d'étape,» [En ligne]. Available: [http://publicationslist.org/data/a.april/ref-410/GTI\\_LOG\\_792\\_HI2013\\_RapportEtape\\_DIOS24088800.pdf](http://publicationslist.org/data/a.april/ref-410/GTI_LOG_792_HI2013_RapportEtape_DIOS24088800.pdf).
- [8] «IEEE programming-languages,» [En ligne]. Available: <http://spectrum.ieee.org/computing/software/the-2016-top-programming-languages>.
- [9] S. Framework, «Spring Framework,» [En ligne]. Available: <https://spring.io/>.
- [10] «Hibernate,» [En ligne]. Available: <http://hibernate.org/>.
- [11] «Maven,» [En ligne]. Available: <https://maven.apache.org/>.
- [12] «Serveru Apache,» [En ligne]. Available: <https://httpd.apache.org/>.
- [13] «Serveur WildFly,» [En ligne]. Available: <http://wildfly.org/>.
- [14] «JSF,» [En ligne]. Available: <http://www.oracle.com/technetwork/java/javase/javaserverfaces-139869.html>.

- [15] «Primefaces,» [En ligne]. Available: <http://www.primefaces.org/>.
- [16] «L'EDI eclipse,» [En ligne]. Available: <https://eclipse.org/>.
- [17] «Subversion(svn),» [En ligne]. Available: <https://subversion.apache.org/>.
- [18] «JUnit,» [En ligne]. Available: <http://junit.org/junit4/>.
- [19] «sonarqube,» [En ligne]. Available: [www.sonarqube.org](http://www.sonarqube.org).
- [20] «Bootstrap,» [En ligne]. Available: <http://getbootstrap.com/>.
- [21] «Jenkins,» [En ligne]. Available: <https://jenkins.io/>.
- [22] «Mantis,» [En ligne]. Available: <https://www.mantisbt.org/>.
- [23] «Trac,» [En ligne]. Available: <https://trac.edgewall.org/>.

