

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

UNIVERSITÉ DU QUÉBEC

RAPPORT TECHNIQUE
PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
DANS LE CADRE DU COURS RÉALISATION ET MAINTENANCE DE
LOGICIELS

RAPPORT TECHNIQUE DU PROJET DE MAINTENANCE DE L'APPLICATION
FIXMYSHOULDER

PAR
MOHAMAD AWADA
SIHAM KADI
MALAK KHALIL
MOHAMMED MAMMAR
THIERRY WARNANT

MONTRÉAL, LE 23 AVRIL 2014



Object Zero Inc., 2014

REMERCIEMENTS

Nous remercions M. *Alain April*, professeur au département de génie logiciel à l'ÉTS, pour le support et l'encadrement fournis durant le mandat de ce projet. Nous tenons aussi à remercier M. *George Demirakos*, physiothérapeute, directeur de la clinique Club Sportif MAA et client du projet *FixMyShoulder*, pour le support et la contribution apportée durant ce projet.

RAPPORT TECHNIQUE DU PROJET DE MAINTENANCE DE L'APPLICATION FIXMYSHOULDER

MOHAMAD AWADA
SIHAM KADI
MALAK KHALIL
MOHAMMED MAMMAR
THIERRY WARNANT

RÉSUMÉ

Le rapport technique qui suit fournit les détails du projet de maintenance réalisé durant le cadre du cours Réalisation et maintenance de logiciels (MGL804). Le rapport présente les objectifs du projet, l'équipe de maintenance, les activités de maintenance, les outils utilisés durant le projet et les livrables produits durant le mandat. En outre, ce document fournit tous les détails en ce qui concerne les obstacles et difficultés rencontrés durant notre mandat et des recommandations pour des travaux subséquents.

Des livrables, autres que ce document, fournis en annexe, traitent des détails en ce qui concerne le plan de projet utilisé, les analyses d'impacts et les guides utilisateurs produits pour la formation des membres de l'équipe.

TABLE DES MATIÈRES

	Page
ÉCOLE DE TECHNOLOGIE SUPÉRIEURE	1
INTRODUCTION	15
CHAPITRE 1 CONTEXTE ET IDENTIFICATION DU PROJET	17
1.1 Contexte du projet.....	17
1.2 Objectifs.....	18
1.3 Porté	19
CHAPITRE 2 PLAN DE PROJET	21
2.1 Contrat d'équipe	21
2.2 Plan de projet	22
2.3 Méthodologie du travail.....	23
CHAPITRE 3 PRÉSENTATION DE L'APPLICATION.....	24
3.1 Application mobile Androïde	24
3.2 Application Web	26
CHAPITRE 4 INFRASTRUCTURE ET OUTILS DE DÉVELOPPEMENT	28
4.1 Outils de gestion de requêtes	28
4.2 Outils de gestion de configuration SVN	28
4.3 Outils de développement.....	29
4.3.1 Outils de développement Androïd	29
4.3.2 Outils de développement Web	30
4.4 Outils de documentation et des livrables	31
CHAPITRE 5 SLA	32
CHAPITRE 6 PROCESSUS DE MAINTENANCE	34
6.1 Standard	34
6.2 Flux du travail	34
6.3 Intrants et Extrants	35
CHAPITRE 7 REQUÊTE DE MAINTENANCE	39
7.1 Rencontre avec le client	39
7.2 Présentation des requêtes	39
7.3 Critère d'assignation des requêtes	41
7.4 Analyse d'impact des requêtes	41
CHAPITRE 8 AMÉLIORATIONS APPORTÉES.....	42
8.1 Amélioration <i>Androïd</i>	42

8.1.1	MGLFIX-6 Les vidéos dans l'application mobile ne démarrent pas.....	42
8.1.2	MGLFIX-13 Changement du contexte de la première page (Introduction).	42
8.1.3	MGLFIX-15 Push Notification.....	43
8.1.4	MGLFIX-18 Changer les couleurs de l'application.	45
8.1.5	MGLFIX-19 Changer les styles des boutons du menu	46
8.1.6	MGLFIX-21 Géopositionnement.....	46
8.1.7	MGLFIX-20 Changer le style des boutons Infos.....	47
8.2	Amélioration CMS.....	48
8.2.1	MGLFIX-23 Gestion des vidéos dynamiquement à partir du CMS	49
CHAPITRE 9 GESTION DU CLIENT		51
CHAPITRE 10 GUIDE UTILISATEUR.....		52
CONCLUSION		
RECOMMANDATIONS		55
LISTE DES RÉFÉRENCES BIBLIOGRAPHIQUES		62

LISTE DES TABLEAUX

	Page
Table 1	Étapes du processus de maintenance 35

LISTE DES FIGURES

	Page
Figure 1: Diagramme organisationnel	22
Figure 2: Version initiale de l'application mobile.....	25
Figure 3: Gestion des chapitres CMS	26
Figure 4: Gestion des vidéos CMS	27
Figure 5: Processus de maintenance	35
Figure 6: Nombres et types des requêtes levés	41
Figure 7: Introduction Vidéo	43
Figure 8: Push Notification.....	44
Figure 9: Amélioration des couleurs.....	45
Figure 10: Nouveau style de bouton	46
Figure 11: Géo Positionnement	47
Figure 12: Bouton information	48
Figure 13: Gestion 1 des vidéos.....	49
Figure 14: Gestion 2 des vidéos.....	49

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

ÉTS	École de Technologie Supérieure
ITIL	Information Technology Infrastructure Library
ISO	International Standard Organization
IEEE	Institute of Electrical and Electronics Engineer
UI	Interface Utilisateur
Workflow	Flux du travail
MGLFIX-xx	Identificateur unique d'une requête

INTRODUCTION

Ce document se veut le récapitulatif de l'ensemble du processus de maintenance du projet *FixMyShoulder* proposé par le client M. *George Demirakos* (représenté par son délégué au suivi de projet M. *Alain April*) et l'équipe de maintenance logicielle (nommée *Object Zero Maintenance*, comme le département de maintenance d'une pseudo-entreprise nommée *Object Zero Inc*). Pour ce faire, les processus, procédures et différents outils utilisés et documents produits au cours du processus de maintenance logicielle feront l'objet de chapitres spécifiques dans le but d'expliquer leur nécessité et leur utilisation. L'application *FixMyShoulder* étant composée d'un volet mobile *Android* et d'un second Web, nous allons nous attacher à segmenter les opérations de maintenance. Par cela, nous entendons que nous différencierons dans les chapitres suivant : 1) les différents billets de changement et d'amélioration requis par le client ou proposé par le gestionnaire de projet de maintenance, 2) le détail des analyses de risque et des améliorations apportées et 3) les guides d'utilisateurs des deux systèmes.

Ce rapport technique se structure comme suit : dans le premier chapitre, nous décrivons le projet, ses objectifs et la portée de celui-ci, soit son envergure et sa criticité. Dans le deuxième chapitre, nous présenterons le plan de projet, soit la composition de l'équipe projet, du mode opératoire et de communication entre les membres de l'équipe, les différents artefacts et échéanciers prévus du projet. Dans le troisième chapitre, nous présenterons les différents outils de gestion de billets et de versions du projet, de développement, de bureautique utilisés et d'enregistrement et de partage des documents. Le quatrième chapitre porte sur l'entente de service convenue entre le client et le département de maintenance logicielle du fournisseur. Le cinquième chapitre porte sur le processus de maintenance en tant que tel, soit pour les requêtes préventives et perfectives, proposées par l'équipe de maintenance logicielle, que celles correctives, provenant du client. Ce processus se base sur le workflow du traitement de chaque requête, à savoir l'ensemble des étapes d'évaluations, de priorisations, de répliquations, de modifications et d'acceptation de conformité des changements apportés. Le chapitre suivant porte sur la relation client, les requêtes présentées par celui-ci et les analyses d'impact de ces

demandes de changement. Le chapitre huit (8) présente et explique les améliorations apportées aux volets mobile et web de l'application *FixMyShoulder* en rapport aux requêtes retenues. En dernier lieu, nous présenterons les différents guides d'utilisateurs du produit, en tenant compte des changements effectués aux interfaces et à l'ajout d'options, dans le but de guider les utilisateurs dans leur expérience avec l'interface mobile, pour les clients externes à l'entreprise de M. *Demirakos*, et l'interface web, soit M. *Demirakos* et ses associés.

CHAPITRE 1

CONTEXTE ET IDENTIFICATION DU PROJET

1.1 Contexte du projet

Dans le cadre du cours MGL804 « Réalisation et maintenance de logiciels », il a été demandé à l'équipe de maintenance logicielle d'étudier un système conçu dans le cadre d'un projet de maîtrise par M. *Mathieu Crochet* [3] et de déployer un environnement de gestion de maintenance et de procéder au processus d'évolution et de correction du système *FixMyShoulder*. Ce système comprend deux volets, un premier, mobile, porté par l'environnement *Android* et présentant de l'information statique aux clients de M. *Demirakos* (clients dont nous ne connaissons pas les caractéristiques et les besoins), et un second, web, accessible uniquement par M. *Demirakos* pour faciliter la mise à jour des informations de la partie mobile de l'application. Au fur et à mesure de la mise en place de l'environnement de gestion et de développement, l'équipe a été suivie et conseillée par le responsable du cours MGL804 et le délégué au suivi de projet de Messieurs *Demikaros* et *April*.

Dans un premier temps, des améliorations correctives ont été identifiées, planifiées, négociées (avec le délégué au suivi de projet du client) et implantées; en second lieu, le client a été rencontré pour être informé des demandes-requêtes de changement qu'il voulait proposer à l'équipe de maintenance. En aucun temps il a été question de gérer des rapports de problèmes, soit des demandes « nécessitant des corrections de l'application ... ayant ... la priorité sur n'importe quel autre travail en cours » [1]. Suite à une étape d'investigation, des analyses de risque et des rencontres d'équipe, le gestionnaire de projet a proposé au client une priorisation d'implantation des requêtes de changement pour tenir compte de la date de fin de projet (25 avril 2014). Il a été convenu de donner deux semaines (soit à partir du 11 avril 2014) à l'équipe pour effectuer le transfert de l'application de l'environnement de développement et maintenance au

système du client, afin de tester le système et de tenir compte des particularités de l'environnement de production.

Dans le contexte de création et de gestion d'une équipe de maintenance logicielle, il a été primordial de statuer rapidement sur les modalités de gestion de la composition et du modus operandi de l'équipe. Pour ce faire, la gestion de l'équipe de développement est développée dans le chapitre 2 Plan de projet et dans le document en annexe Plan de projet version 1.0.

Pour la gestion des relations avec le client, nous détaillerons cette question dans le chapitre 6 Requête de maintenance. Dans le document Entente de service (voir l'annexe 4), plusieurs sections et mentions sont faites à ce propos. Selon les degrés d'autorisation nécessaires, le gestionnaire de projet de maintenance a rencontré, discuté et obtenu l'accord du client ou de son délégué au suivi de projet de maintenance. Ces relations sont entretenues dans le cadre d'une communication et d'une implication serrée du client et de son délégué dans le but de développer une relation de confiance et un suivi de projet tout au long du projet de maintenance logicielle

1.2 Objectifs

L'objectif de notre mandat était de mettre en place un processus de maintenance selon les règles de l'art. Le processus de maintenance se basé sur le modèle S^{3m}, ISO12207 et ISO14674. De plus, l'équipe de maintenance devait élaborer une entente de service (SLA) afin de définir la portée du projet. En outre, un système de billetterie était nécessaire afin d'implanter le processus de maintenance et de gérer les requêtes levées tout au long du projet. Un autre objectif important de ce mandat était d'appliquer les étapes du processus de maintenance par les membres de l'équipe et de gérer une équipe de maintenance. En outre, la gestion du client était un autre objectif que l'équipe de maintenance devait atteindre.

1.3 Porté

Ce rapport technique porte sur la maintenance des volets mobile Android et web de l'application *FixMyShoulder*. À cet effet, l'effort de programmation de maintenance s'est porté sur l'ensemble des types de maintenance, soit les maintenances corrective, perfective, préventive et corrective. Pour chaque requête de changement (levée par l'équipe de maintenance ou par le client), un rigoureux processus de maintenance logicielle est appliqué et géré par l'entremise du système de gestion et suivi des incidents JIRA. Au terme de la période de trois (3) mois et demi de la durée du projet, l'équipe de maintenance remettra au client l'ensemble des livrables générés par le projet, ainsi que le code produit lors des activités de maintenances réalisées, en plus de présenter le système et de donner une formation sur l'utilisation des nouvelles fonctionnalités. Les livrables fournis au client sont disponibles en annexe et sont utilisés dans ce rapport à titre de références dans le rapport pour appuyer la démarche de l'équipe de maintenance. La taille de l'application, que ce soit le volet mobile ou web, est très raisonnable, dans le sens où nous entendons qu'il a été aisé, par les développeurs de l'équipe de maintenance, de se faire une représentation de l'ensemble de l'application. La partie mobile *Android* comprenait en tout des classes Java et XML, alors que la partie web comprenait des classes HTML, PHP et des fichiers vidéo. L'application maintenue a été entièrement dupliquée par l'équipe de maintenance. L'environnement de développement des requêtes et demandes de changement était indépendant de l'environnement de production de l'application, côté client, telle qu'elle l'ait actuellement, au moment de la rédaction du rapport et que le système modifié n'a pas encore été transféré et implanté chez le client. L'application maintenue n'est pas de type critique, il s'agit d'une application d'information à l'endroit des visiteurs et des clients de l'entreprise de M. *Demirakos*, soit des affichages d'informations statiques et des vidéos instructifs

CHAPITRE 2

PLAN DE PROJET

2.1 Contrat d'équipe

Dans ce chapitre nous allons présenter les membres de l'équipe, la planification du projet et la méthodologie appliquée afin d'atteindre les objectifs.

Tout d'abord, les membres de l'équipe ont signé un contrat d'équipe qui dicte les règles auxquelles les membres de l'équipe adhèrent afin de travailler dans un environnement professionnel (voir l'annexe A). Le contrat d'équipe présente les exigences personnelles de chaque membre afin de bien travailler en équipe. Les exigences personnelles dictent clairement les attentes envers les membres de l'équipe afin de travailler dans un environnement collaboratif. En outre, ce document fournit les objectifs de l'équipe à atteindre durant notre mandat. Les règles de conduite et éthique qui doivent être respectées par tous les membres de l'équipe afin de travailler en harmonie se trouvent dans ce document. Enfin, des sections spécifiques ont été dédiées à propos du rôle de chaque membre. Ce contrat nous a permis d'identifier les qualités de chaque membre de l'équipe. L'identification de ces qualités fournit les éléments de force que chaque membre pourrait apporter à l'équipe pour atteindre les objectifs précédemment identifiés. Le contrat d'équipe est présenté à l'annexe A pour référence. L'avantage du contrat d'équipe est qu'il nous a permis de mettre en place un diagramme organisationnel. La figure 1 présente l'organigramme de l'équipe de projet, ainsi que le rôle respectif des membres de chacun durant le projet.

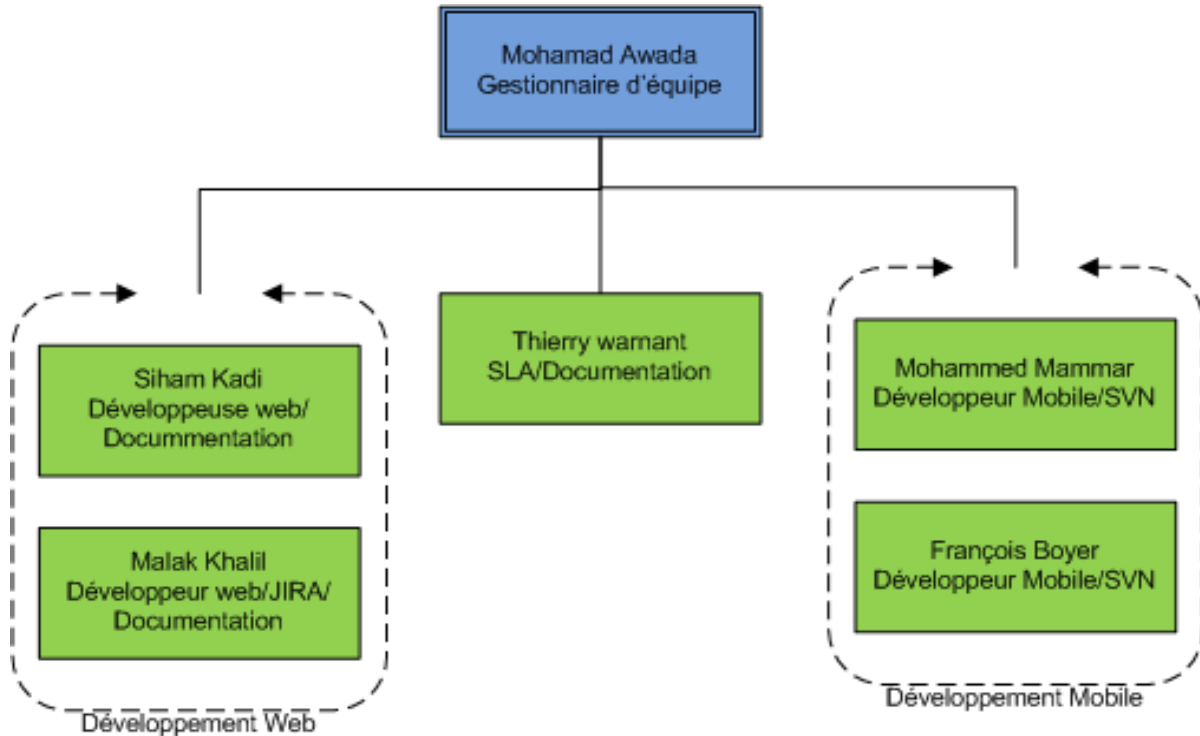


Figure 1: Diagramme organisationnel

2.2 Plan de projet

Afin de mieux gérer les membres de l'équipe, un plan de projet, qui permet de définir le rôle de chaque membre, les échéanciers et les livrables à produire durant le projet de maintenance, a été mis en place. Le plan de projet nous a servi de guide afin de mieux déterminer les objectifs du projet de maintenance et de mettre en place l'infrastructure utilisée durant l'implémentation des changements. Le document du plan de projet (voir l'annexe B) avait comme but de fournir les détails des livrables produits, des risques gérés, des outils utilisés et enfin des standards et processus appliqué durant le mandat. La section des livrables fournit les documents et les artefacts à produire durant le projet, accompagnés d'une description afin de mieux comprendre leur utilité et leur objectif. La section des risques avait pour but d'identifier les obstacles que l'équipe pouvait rencontrer durant le mandat, dont, entre autres, les risques de gestion, humains et technologiques. De plus, ce document énumère les outils utilisés durant le mandat, soit les outils de maintenance, d'implémentation, de gestion et de documentation. Les outils

de maintenance seront plus détaillés dans le prochain chapitre afin de justifier leur utilisation. Enfin, une section sur le processus de maintenance était fournie. La section des standards fournit la liste exhaustive des dix étapes du processus standard de maintenance issue du modèle S³M. Le processus de maintenance et les détails des étapes seront explicités dans le chapitre 6.

2.3 Méthodologie du travail

Pour atteindre nos objectifs et s'assurer que le processus de maintenance était implémenté correctement, l'équipe a tenu des rencontres hebdomadaires. Ces rencontres avaient comme but de coordonner les activités entre les membres de l'équipe et de s'assurer que les tâches étaient livrées dans les délais déterminés. De plus, ces rencontres nous ont permis de faire un suivi sur l'état de la progression des tâches et de mettre à jour la documentation nécessaire. En outre, ces rencontres ont permis d'aider des membres dans l'équipe afin d'installer les applications et l'infrastructure nécessaire pour entamer leurs tâches. La communication par courriels était une autre méthode utilisée afin de soumettre les tâches à suivre, les statuts des activités et les directions en ce qui concerne les priorités des requêtes et des documents à produire. Des sommaires de suivi bihebdomadaire étaient envoyés au client afin de fournir l'état de l'avancement des requêtes et des améliorations apportées à l'application.

CHAPITRE 3

PRÉSENTATION DE L'APPLICATION

3.1 Application mobile Androïde

Dans ce chapitre, nous allons présenter un résumé de l'application *FixMyShoulder*. L'application se base sur un livre de physiothérapie des épaules et est disponible sous la forme d'une application mobile disponible sur les plateformes *Androïd* et *IOS Apple*. L'application mobile fournit les sections suivantes:

- Les chapitres du livre *FixMyShoulder*
- Des vidéos (Exercices de physiothérapie tirés du livre)
- Et une section info (Nous contacter/E-mail/Adresse)

Les captures d'écran suivantes ont pour but de présenter l'application initiale, avant toute modification apportée par l'équipe de maintenance. D'autres captures d'écrans seront produites pour présenter l'évolution de l'application suite aux modifications apportées. Ces figures seront fournies dans les sections ultérieures du document.

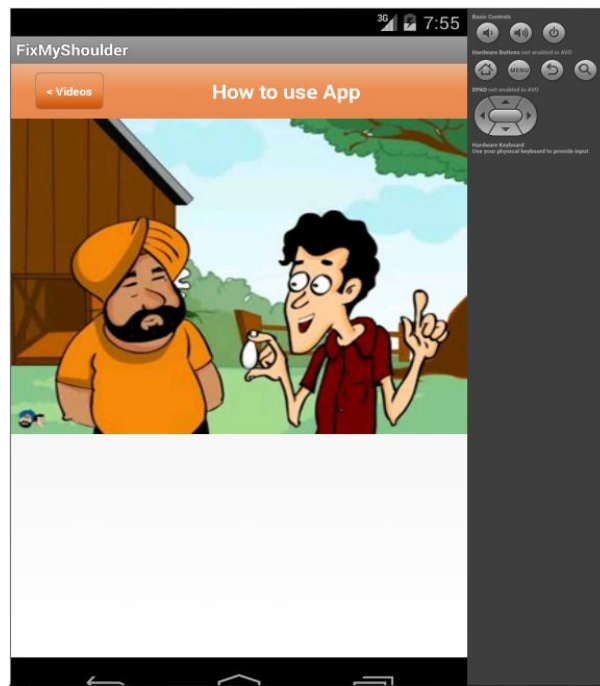
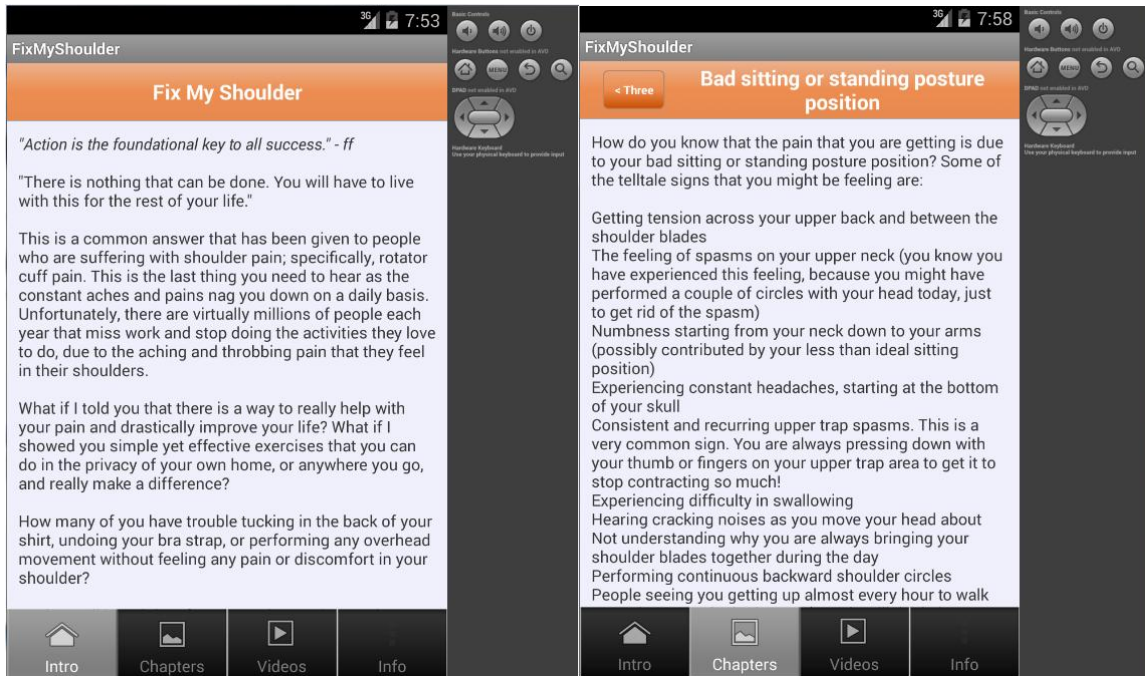


Figure 2: Version initiale de l'application mobile

3.2 Application Web

La partie CMS (Content Management System) permet la mise à jour de l'application mobile. C'est-à-dire qu'à chaque fois que le client désire effectuer une opération d'ajout de retrait ou de modification, d'un chapitre ou d'une vidéo, il doit procéder au changement à partir du CMS. Ces changements seront répliqués automatiquement dans la partie mobile de l'application. Cette façon de procéder a pour but de faciliter les mises à jour sans avoir à modifier l'application mobile.

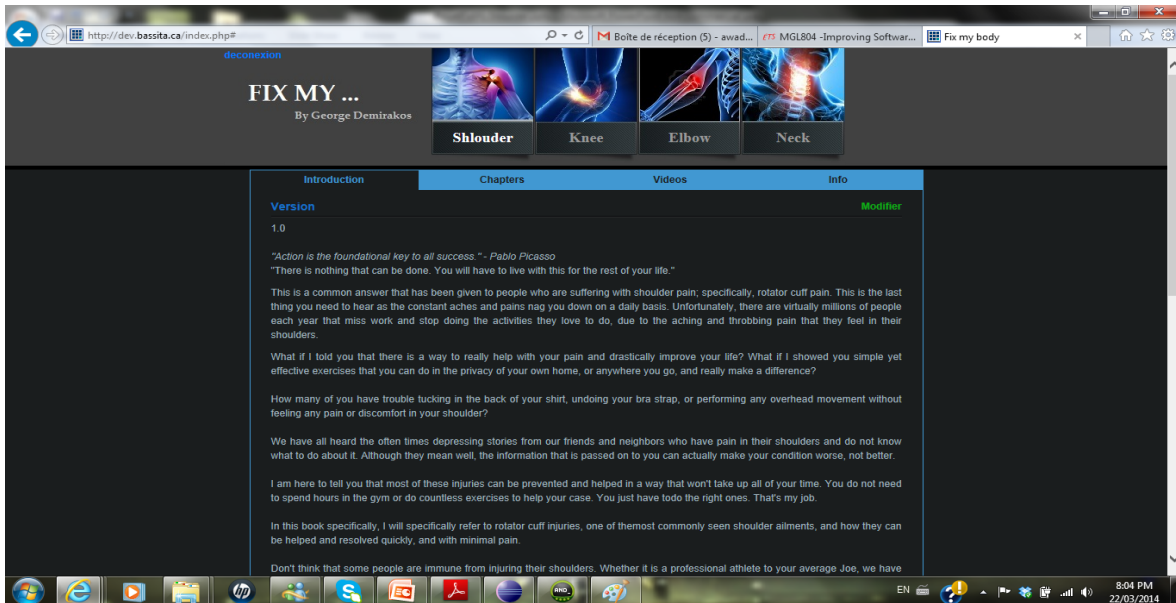


Figure 3: Gestion des chapitres CMS

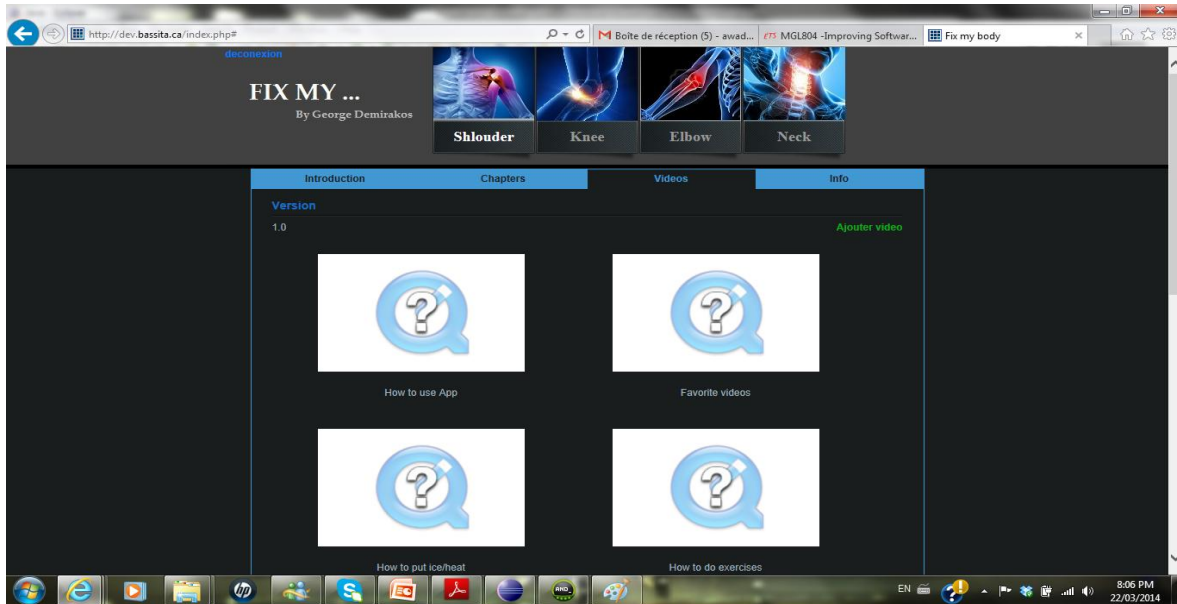


Figure 4: Gestion des vidéos CMS

CHAPITRE 4

INFRASTRUCTURE ET OUTILS DE DÉVELOPPEMENT

Dans ce chapitre nous allons présenter les outils utilisés durant le mandat. Ces outils ont été mis en place afin de gérer, développer et documenter les activités de maintenance. Le choix et la mise en place de ces outils ont constitué une étape très importante pour le bon déroulement de ce projet. Nous avons classé ces outils en quatre catégories :

- Outils de développements
- Outils de gestion des requêtes
- Outils de gestion des configurations
- Outil de documentation et de la production des livrables

4.1 Outils de gestion de requêtes

L'outil JIRA avait pour but de permettre la gestion des requêtes de maintenance. Correctement configuré, cet outil nous a permis de définir le type de chaque requête, la priorité et l'assignation de la requête à un membre de l'équipe. Le *workflow* du processus de maintenance a été implanté dans l'outil et les membres pouvaient consulter l'état de progression de chaque requête, en tout temps. Les détails de ce processus sont fournis dans le chapitre 6.

4.2 Outils de gestion de configuration SVN

Le SVN ou *Apache Subversion* est parmi les outils les plus populaires dans le monde de la gestion et l'archivage du code source. La plupart des outils dans ce domaine répondent aux besoins classiques des développeurs et SVN est parmi ceux qui se distinguent par sa gratuité, sa simplicité d'installation et de configuration, et sa facilité d'utilisation. Pour faciliter le déploiement et la préparation de notre environnement de maintenance, nous avons opté pour un des serveurs de gestion de configuration compatible avec SVN, en l'occurrence <http://riouxsvn.com/>. Une fois les comptes des usagers créés, nous avons

eu qu'à procéder à l'installation du client *Subversion* et à télécharger le code source. Chaque développeur devait créer une branche pour chaque billet qui lui est associé. Cette façon de faire a permis à chaque membre de l'équipe de travailler sur une version stable qui ne contenait pas les changements des autres membres. Une fois toutes les tâches implémentées et validées, nous avons procédé au fusionnement du nouveau code avec la branche racine avant de transférer le tout sur notre serveur SVN.

4.3 Outils de développement

Comme nous l'avons mentionné au point 2.1 du présent document, le projet qui fait l'objet de la maintenance dans le cadre de ce mandat est un projet qui se compose de deux parties, un volet *Android*, qui traduit un livre de physiothérapie de l'épaule, et un volet Web CMS, qui permet la mise à jour de cette application.

Pour apporter des améliorations ou des ajouts de fonctionnalités selon le processus de maintenance et les requêtes levées, nous avons besoin d'outils de développement Web et *Android*.

4.3.1 Outils de développement Android

La maintenance de l'application *Android* du système *FixMyShoulder* a été effectuée avec *Eclipse* en utilisant le plugin ADT (*Android Development Toolkit*). Le langage de développement est Java, un langage moderne, simple et orienté objet. ADT permet de créer des projets *Android* et de générer rapidement des UIs riches en contenu. Le SDK *tools* d'ADT facilite énormément le débogage et la maintenance. En fin, cette extension permet de générer les fichiers .apk qui sont déployés sur les téléphones intelligents en mode production. L'un des outils les plus intéressants d'ADT est l'émulateur (machine virtuelle) qui permet la simulation l'expérience utilisateur des téléphones intelligents fonctionnant avec *Android*. Le SDK permet de créer plusieurs images du système *Android* et de varier les plateformes de tests sans avoir besoin de disposer de différents téléphones avec différentes versions du même système.

Pour améliorer la performance de l'émulateur, installé par défaut, qui est connu pour sa lenteur, nous avons téléchargé et configuré sur ce dernier l'accélérateur physique fourni par [Intel® Hardware AcceleratedExecution Manager](#). Cet accélérateur a donné des résultats impressionnants en terme de temps de réponse (Ex : démarrage de l'émulateur sans accélérateur est de 3 à 4 minutes, avec l'émulateur ça prend moins de 30 seconds). Le déploiement, l'exécution et le débogage ont été rendus plus faciles.

Enfin, *Eclipse* est un environnement de développement ouvert qui permet l'intégration des outils de gestion de code source tel que SVN.

4.3.2 Outils de développement Web

Pour la partie web du projet, nous avons utilisé l'environnement de développement *NetBeans*¹, qui est un outil open source, permettant le développement des modules Web en utilisant le PHP qui est un langage qui permet de produire des pages web dynamiques via le serveur *HTTP*, *HTML* qui est un langage de balisage qui permet de mettre en place le contenu d'une page avec la possibilité d'y intégrer des ressources multimédias², et le CSS pour la mise en page du site web.

Pour l'hébergement du CMS, nous avons utilisé le serveur Linux (CentOs), Apache 2.2.26, PHP 5.4.6 et Mysql 5.5.36-log

¹<https://netbeans.org/>

²<http://fr.wikipedia.org/wiki/HTML>

4.4 Outils de documentation et des livrables

Tout au court du mandat, pour la production les livrables cités au point 3.4 du présent document, nous avons utilisé les outils Word et Visio respectivement, pour la rédaction des documents et pour la génération des figures illustratives.

Afin de réaliser un travail collaboratif, avec un partage équilibré les tâches, nous avons utilisé Google Drive. Cet outil a permis de stocker et centraliser tous les documents nécessaires pour le projet pour qu'ils soient accessibles à tous les membres de l'équipe, et pour conserver une traçabilité des versions : c'est-à-dire qui a fait quoi et quand, afin de produire des livrables de bonne qualité.

CHAPITRE 5

SLA

Pour le document d'entente de service (voir l'annexe 4) (livrable produit durant le processus de projet de maintenance), l'équipe projet s'est servie du modèle *Service Level Agreement – 2000 Work Program* [2] comme référence de développement des éléments à inclure dans le document d'entente présenté à notre client. Comprenant que ce document représentait la formalisation d'un accord de service « négocié » entre les deux parties, il a été primordial dans la composition et le développement de ce document que la notion d'entente mutuelle bénéfique, représentant clairement les deux parties, soit présente et transparaisse dans l'entente finale.

Tenant compte que la courte période sur laquelle s'est réalisé le mandat de maintenance du projet pour l'application *FixMyShoulder*, soit trois (3) mois et demi, le nombre relativement restreint de requêtes de maintenance (14 billets enregistrés dans le système *JIRA*), le degré de complexité des requêtes de changement contrôlé et tenant compte des moyens relativement limités pour la mise en place d'une architecture système de maintenance, l'entente de service proposé comprenait des clauses simples, des objectifs aisément réalisables, mais respectueux des besoins et des attentes du client en termes de délais de livraison, de qualité et de complétude de l'évolution du système. L'application *FixMyShoulder* étant déjà fonctionnelle, l'équipe projet de maintenance, sur la base de ce système, se devait de respecter les décisions de développement précédemment prises. Toutefois, dans le respect du mandat de l'équipe *Object Zero Maintenance*, des opérations de maintenance préventives, perfectives et adaptatives ont été implantées dans le strict respect d'un processus de consultation, de proposition et d'obtention de l'accord du client.

Bien que les *key performance indicators* (KPI) définies dans l'entente aient portées sur la relève des systèmes en cas de panne ou de problèmes d'accès, l'entente de service a été formulée dans le but de centrer les interventions de maintenance sur les résultats

d'implantation des requêtes plutôt que sur l'effort consenti. Par la définition d'un workflow, le respect scrupuleux des étapes constituant le processus de gestion de la maintenance et le suivi des interventions et des vérifications (rapports d'analyse de risques et d'assurance qualité) a permis au client (ou son représentant) de suivre, maîtriser et valider l'atteinte des objectifs et la qualité des actions de maintenance sur l'application *FixMyShoulder*. Le cadre de développement des modifications apportées très bien maîtrisé et stable, l'expertise de l'équipe de maintenance, la qualité et la célérité du travail effectué, la relativement courte période du mandat de maintenance, nous n'avons pas eu à mettre en application de nombreuses clauses de l'entente de service. Toutefois, ce document a clairement guidé la prise en charge de la relation client par le gestionnaire de projet de maintenance et tout du long du projet, de façon hebdomadaire et à l'atteinte des étapes importantes, les deux parties ont été en contact par l'entremise de courriels ou de rencontres, ponctuées de remises de documents.

CHAPITRE 6

PROCESSUS DE MAINTENANCE

6.1 Standard

Le flux de travail du projet de maintenance fut une partie critique de travail de l'équipe. Il a été défini en se basant sur un exemple de flux fourni par M. Alain April et sur le modèle S^{3M} et les normes **ISO12207** et **ISO14764**.

6.2 Flux du travail

Le flux de travail est représenté en deux catégories :

- Statut : Représente l'état actuel d'une requête de maintenance
- Transition : Représente le transfert d'un statut à un autre

La prochaine figure fournit le processus de maintenance implanté durant le projet. Les étapes seront détaillées dans les sections subséquentes.

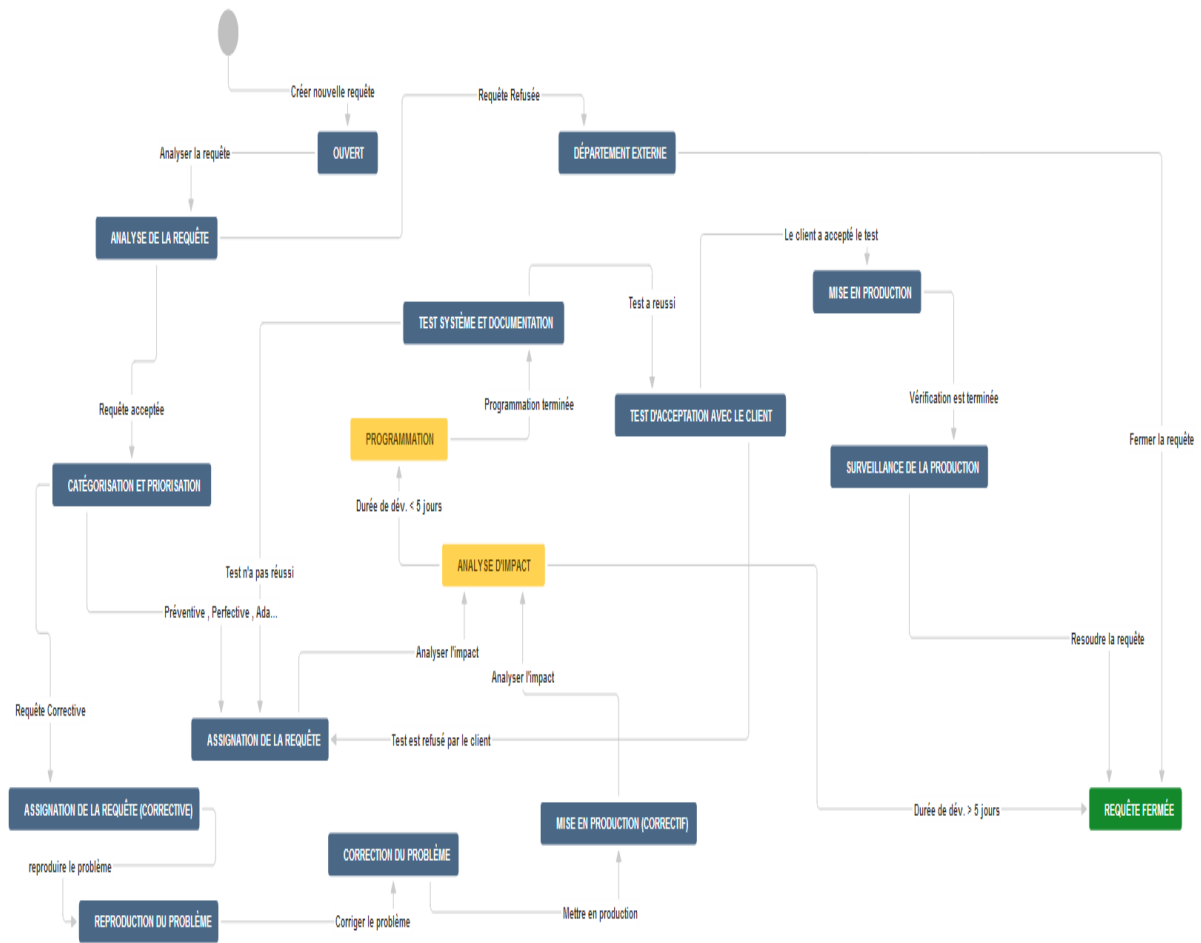


Figure 5: Processus de maintenance

6.3 Intrants et Extrants

Pour mieux expliquer les statuts et transitions de flux de travail, le tableau 6.1 décrit en détail tous les intrants et extrants du processus de maintenance d'une nouvelle requête, de sa création jusqu'à sa fermeture.

ID	Nom d'étape	Statut	Transition(s)
(1)	Créer un ticket	Création	Création (2) >> OUVERT
(2)	Ouvert	OUVERT	Analyser la requête (3) >> ANALYSE DE LA REQUÊTE
(3)	Analyse de la requête	ANALYSE DE LA REQUÊTE	Requête acceptée (4) >> CATÉGORISATION ET PRIORISATION Requête Refusée (5) >> DÉPARTEMENT EXTERNE
(4)	Catégorisation et Prioritisation	CATÉGORISATION ET PRIORISATION	Préventive, Perfective, Adaptive (7) >> ASSIGNATION DE LA REQUÊTE Requête Corrective (6) >> ASSIGNATION DE LA REQUÊTE (CORRECTIVE)
(5)	Département Externe	DÉPARTEMENT EXTERNE	Fermer la requête (17) >> REQUÊTE FERMÉE
(6)	Assignment de la requête (Corrective)	ASSIGNATION DE LA REQUÊTE (CORRECTIVE)	reproduire le problème (8) >> REPRODUCTION DU PROBLÈME
(7)	Assignment de La requête (Préventive, Perfective, Adaptive)	ASSIGNATION DE LA REQUÊTE	Analyser l'impact (11) >> ANALYSE D'IMPACT
(8)	Reproduction du problème	REPRODUCTION DU PROBLÈME	Corriger le problème (9) >> CORRECTION DU PROBLÈME
(9)	Correction du problème	CORRECTION DU PROBLÈME	Mettre en production (10) >> MISE EN PRODUCTION (CORRECTIF)

(10)	Mise en production (Correctif)	MISE EN PRODUCTION (CORRECTIF)	Analyser l'impact (11) >> ANALYSE D'IMPACT
(11)	Analyse d'impact	ANALYSE D'IMPACT	Durée de dév. > 5 jours (17) >> REQUÊTE FERMÉE Durée de dév. < 5 jours (12) >> PROGRAMMATION
(12)	Programmation	PROGRAMMATION	Programmation terminée (13) >> TEST SYSTÈME ET DOCUMENTATION
(13)	Test système et documentation	TEST SYSTÈME ET DOCUMENTATION	Test n'a pas réussi (7) >> ASSIGNATION DE LA REQUÊTE Test a réussi (14) >> TEST D'ACCEPTATION AVEC LE CLIENT
(14)	Test d'acceptation avec le client	TEST D'ACCEPTATION AVEC LE CLIENT	Le client a accepté le test (15) >> MISE EN PRODUCTION Test est refusé par le client (7) >> ASSIGNATION DE LA REQUÊTE
(15)	Mise en production	MISE EN PRODUCTION	Vérification est terminée (16) >> SURVEILLANCE DE LA PRODUCTION
(16)	Surveillance de la production	SURVEILLANCE DE LA PRODUCTION	Resoudre la requête (17) >> REQUÊTE FERMÉE
(17)	Requête Fermée	REQUÊTE FERMÉE	

Table 1: Étapes du processus de maintenance

Type des requêtes :

- Corrective : Sers à corriger les erreurs après la mise en production
- Adaptative : Sers à modifier le logiciel après sa mise en production à cause d'un changement de l'environnement technique
- Préventive : Sers à corriger les défauts latents avant qu'ils deviennent des défaillances
- Perfective : Sers à modifier le logiciel pour améliorer son exécution

Durée de développement :

La durée de développement d'une requête de maintenance ne devrait pas dépasser cinq (5) jours, dans le cas échéant la requête se ferme, car elle ne devait pas faire partie du cycle de maintenance. En dépassant la limite de cinq (5) d'efforts, une requête est considérée comme un nouveau module et fait alors partie du cycle de développement.

Création d'une requête :

Une requête se crée par le client (Adaptative), développeur ou le chef du projet (préventive, corrective, perfective).

Fermeture d'une requête :

Toutes les requêtes sont fermées par le chef du projet.

CHAPITRE 7

REQUÊTE DE MAINTENANCE

7.1 Rencontre avec le client

Durant notre mandat, nous avons eu une seule rencontre avec le client. La rencontre avait pour but de comprendre les besoins du client et d'épauler le client en ce qui concerne les priorités des améliorations désirées. Afin d'avoir une rencontre constructive et d'atteindre l'objectif de se rencontrer, un courriel a été envoyé au client avant la première rencontre. Ce courriel décrivait le but de notre projet et la portée de notre mandat. De plus, nous avons demandé au client de préparer une liste d'améliorations et de changements qu'il désirait apporter au système dans son état initial. De son côté, l'équipe de maintenance a effectué une analyse des fonctionnalités existantes afin de noter les améliorations à suggérer au client. Durant la rencontre, le client a présenté la nature de son travail et comment l'application pourra l'aider dans son travail. Le gestionnaire de projet a aussi fait part des possibilités de changements-modifications pouvant être considérées.

Le client n'avait pas l'intention de changer les fonctionnalités de l'application mais plutôt d'apporter des améliorations aux interfaces utilisateurs (UI). En outre, le client n'avait pas une idée claire des changements désirés et des améliorations à apporter sur l'UI. Afin d'aider le client à mieux définir ses besoins, les suggestions de l'équipe ont été présentées au client en ce qui concerne les améliorations possibles durant notre mandat. Les suggestions avaient pour but d'améliorer les couleurs de l'application, changer la page de l'introduction de l'application *Android*, améliorer l'utilisabilité du CMS et des changements qui avaient pour but de mettre à jour l'UI.

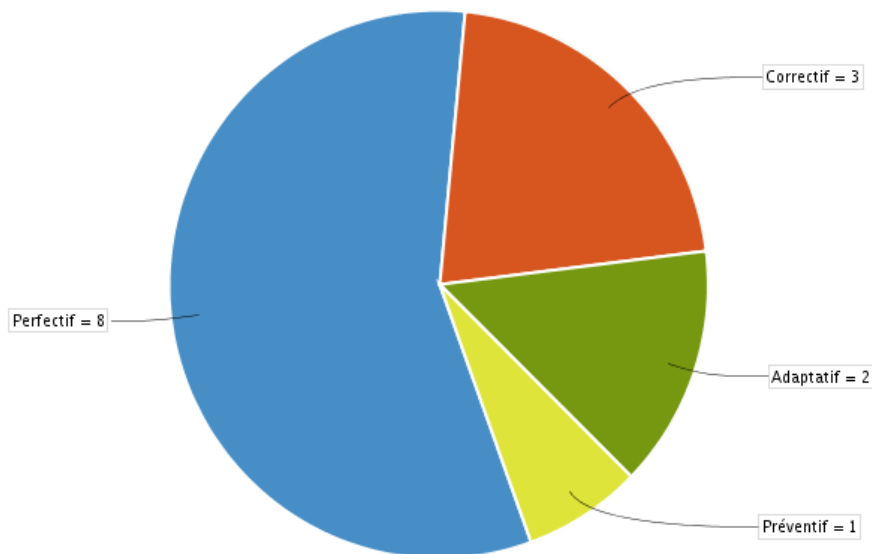
7.2 Présentation des requêtes

Suite à la rencontre du 14 février 2014 avec le client, 14 requêtes ont été levées.

Six d'entre elles ont été levées par notre équipe de maintenance. Celles-ci concernent essentiellement des améliorations à apporter au projet; à titre d'exemples, le changement du logo de l'application ainsi que ses couleurs pour concorder avec celles du site web de la mise à jour, et aussi des ajouts de fonctionnalité notamment la gestion dynamique des vidéos.

Huit requêtes ont été levées par le client et ont été acceptées par notre équipe. Ces requêtes ont aussi pour but d'ajouter des nouvelles fonctionnalités et d'améliorer l'expérience utilisateur (tant du gestionnaire que des clients) du système. Les requêtes couvrent tous les types de maintenance : 8 requêtes perfectives, 2 requêtes adaptatives, 1 requête préventive et 3 requêtes correctives. Dans l'ensemble, ces requêtes ont une priorité moyenne. Nous n'avons pas eu à gérer de requêtes de priorité urgente ou critique. Les requêtes sont identifiées comme suit : MGLFIX-xx, le « xx » représente le numéro de référence de chaque requête.

La figure suivante fournit le nombre de requêtes levées et le type de chaque requête (informations extraites du système *JIRA*).



Type	Nb de requête	%
Perfectif	8	57%
Correctif	3	21%
Adaptatif	2	14%
Préventif	1	7%

Figure 6: Nombres et types des requêtes levés

7.3 Critère d'assignation des requêtes

L'assignation des requêtes était basée sur deux critères importants. Le premier critère était l'assignation de la requête en fonction de l'équipe de développement. C'est-à-dire que si la requête était associée au module *Android*, l'assignation était affectée à un membre de l'équipe développement *Android*. Si la requête était associée au module *CMS*, l'assignation était affectée à un membre de l'équipe développement Web. Le deuxième critère était basé sur l'habileté technique de chaque membre. Dès le début du projet nous avons identifié les habiletés techniques de chaque membre afin de mieux gérer l'assignation. Si la requête avait rapport avec les fonctionnalités de l'application, l'assignation était affectée à un développeur 'Backend'. Si la requête avait rapport avec l'UI de l'application, l'assignation était affectée à un développeur 'UI'. Cette méthode d'assignation avait pour but de minimiser les impacts en cas de problème d'implantation d'une fonctionnalité.

7.4 Analyse d'impact des requêtes

Pour chaque requête levée durant l'activité de maintenance une analyse d'impact était effectuée. L'analyse était faite par le développeur assigné à la requête. De plus, l'équipe de maintenance a mis en place un document d'analyse d'impact qui fournit les détails de toutes les analyses d'impacts effectuées.

CHAPITRE 8

AMÉLIORATIONS APPORTÉES

Les améliorations et changements apportés à ce projet se divisent en deux parties :

- Partie web
- Partie *Android*

8.1 Amélioration *Android*

8.1.1 MGLFIX-6 Les vidéos dans l'application mobile ne démarrent pas

Après une analyse des logs, il s'est avéré que les vidéos de l'application ne supportaient pas la lecture en mode streaming. Le problème a été résolu suite à l'application des recommandations indiquées sur le site *d'Android* relativement au format des vidéos qui supportent le streaming. (Voir : <http://developer.android.com/guide/appendix/media-formats.html>)

Pour résoudre le problème, nous avons opté pour un des outils qui permettent la conversion des vidéos en mode streaming, en l'occurrence *MP4Box*.

8.1.2 MGLFIX-13 Changement du contexte de la première page (Introduction).

Le remplacement du texte d'introduction par une vidéo introductive a été un bon choix pour exposer plus d'informations à l'utilisateur à propos de l'application. Au démarrage de l'application, une vidéo est démarrée automatiquement. La lecture de cette dernière est suspendue temporairement si l'utilisateur quitte l'application ou navigue sur une autre page. La lecture est toutefois reprise au même endroit où elle s'est arrêtée lorsque l'utilisateur retourne à la page principale.

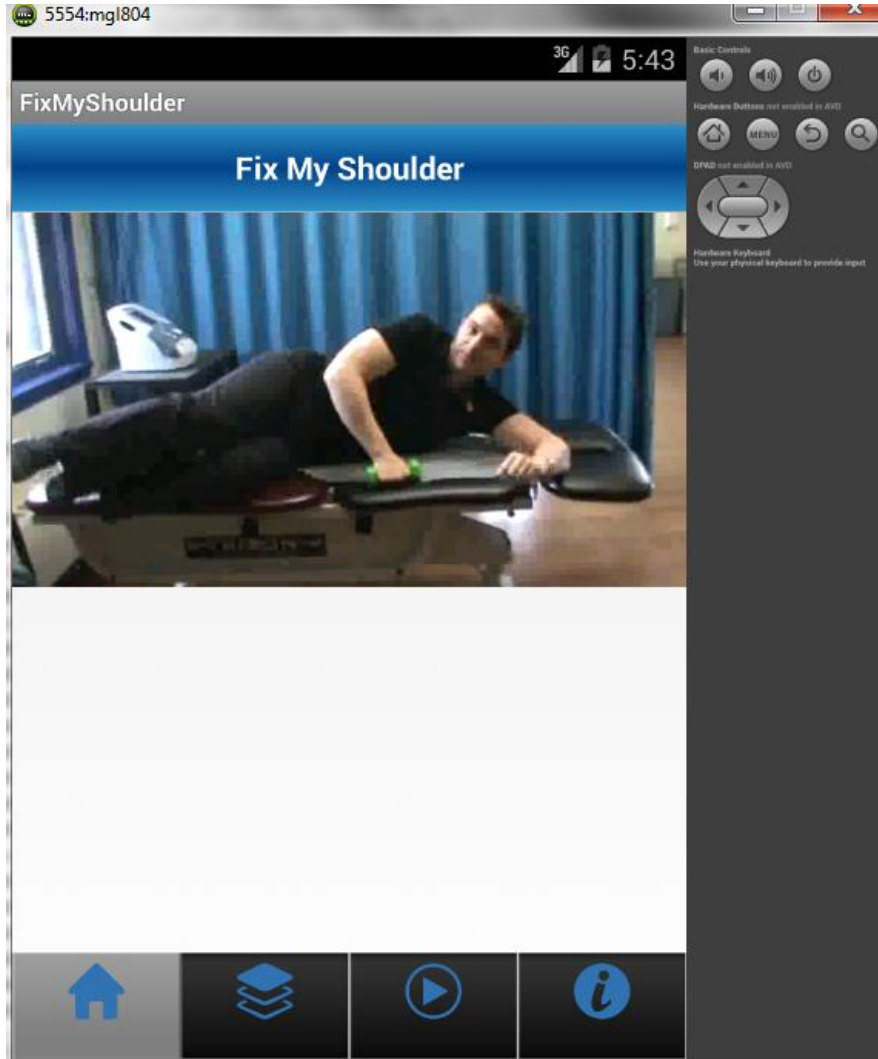


Figure 7: Introduction Vidéo

8.1.3 MGLFIX-15 Push Notification

C'est l'un des changements qui a eu plus d'impact sur l'architecture de l'application *Android* et le *CMS*. L'idée est d'informer, à l'aide d'une notification, l'utilisateur dès qu'il y a un changement au niveau du contenu de l'application, que ce soit au niveau des vidéos ou du texte des chapitres. Pour implémenter cette fonctionnalité, il a fallu concevoir un service au niveau de la plateforme *Android* sous forme d'une routine qui compare en continu la version du contenu sur *Android* avec celui du serveur. La version du serveur est fournie par un fichier *xml* (version.xml) qui est incrémenté automatiquement à chaque

modification au niveau du *CMS*. Une fois l'utilisateur informé du changement, il peut cliquer sur la notification pour que l'application se connecte au serveur et télécharge la dernière version du contenu. Le service qui vérifie la version sur les *CMS* a été configuré pour s'activer automatiquement au démarrage du téléphone. À noter que cette fonctionnalité a été réalisée en collaboration avec la personne qui s'occupe de la maintenance de la partie web. Un billet a été créé en conséquence, après l'analyse d'impact, pour expliquer les besoins fonctionnels nécessaires (MGLFIX-35).

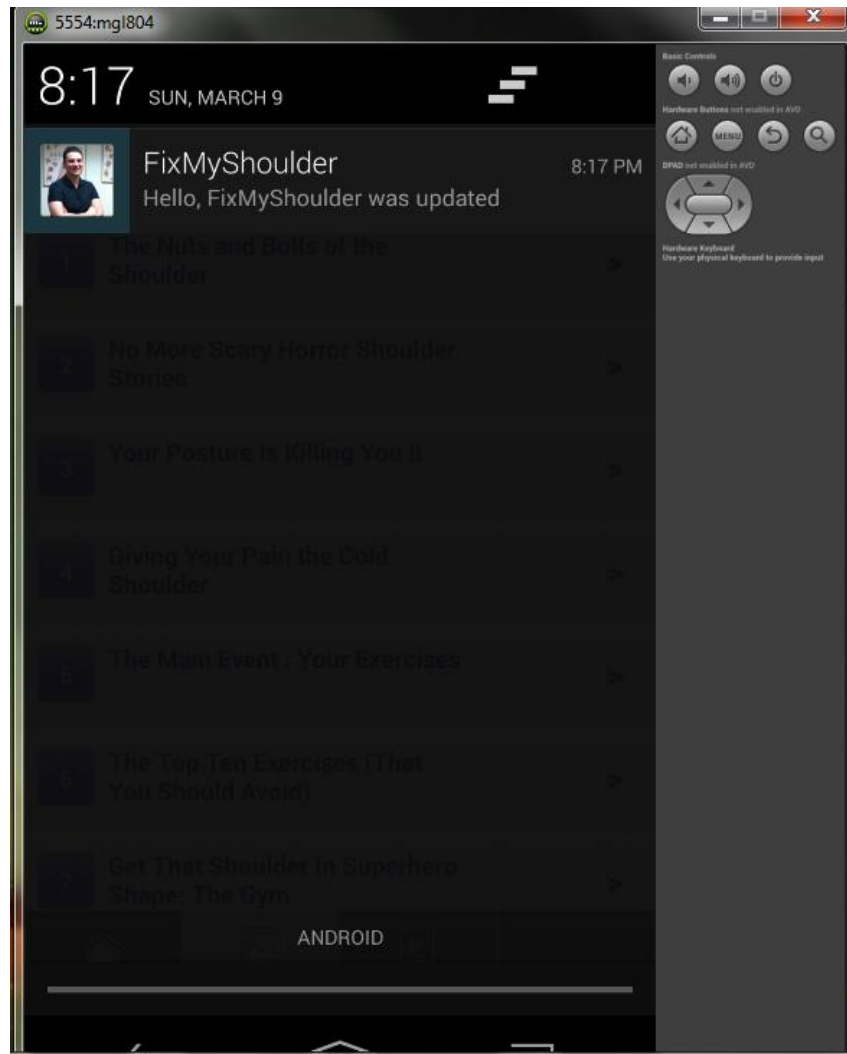


Figure 8: Push Notification

8.1.4 MGLFIX-18 Changer les couleurs de l'application.

Pour harmoniser l'interface usager de l'application *Android* avec le *CMS*, un changement a été nécessaire au niveau des couleurs de l'arrière-plan des différentes fenêtres.

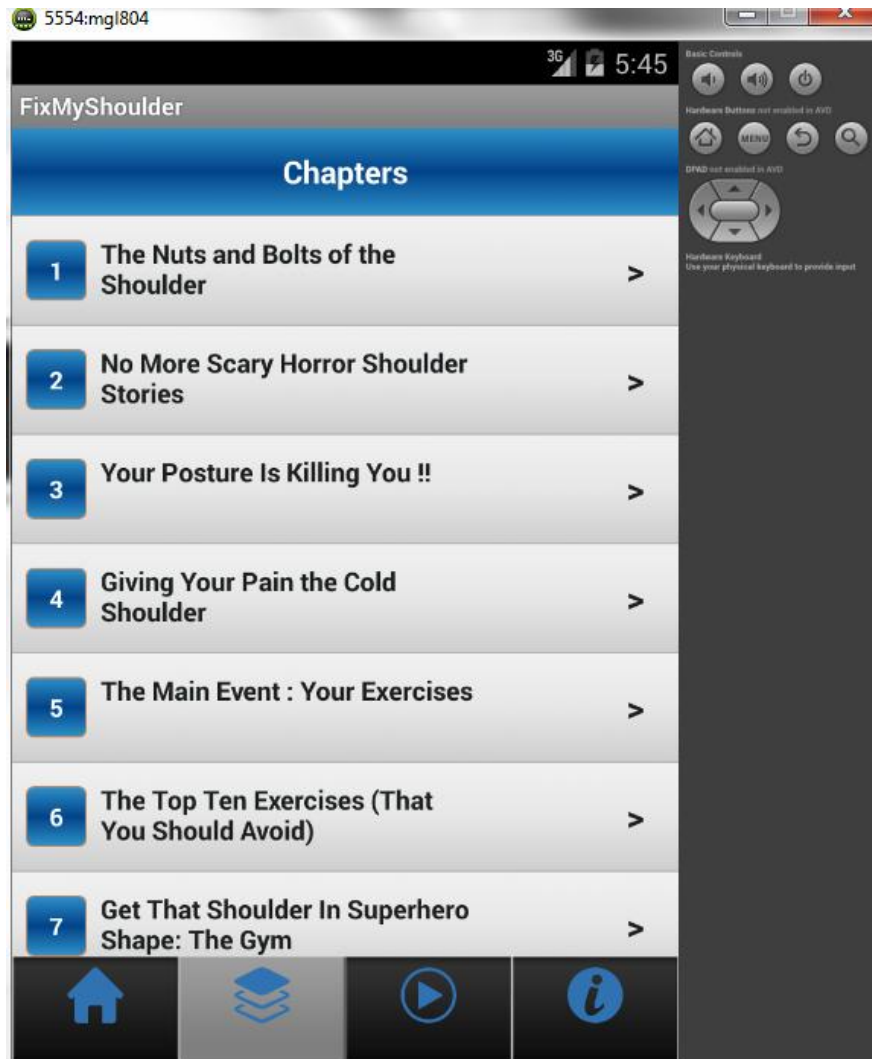


Figure 9: Amélioration des couleurs

8.1.5 MGLFIX-19 Changer les styles des boutons du menu

Les boutons du menu de l'application *Androïd* ont été remplacés pour mieux les adapter aux nouvelles couleurs apportées suite à l'implémentation de MGLFIX-20.



Figure 10: Nouveau style de bouton

8.1.6 MGLFIX-21 Géopositionnement

Suite à l'analyse de l'application, les membres de l'équipe ont proposé au client une fonctionnalité de géopositionnement de l'adresse de la clinique pour faciliter la localisation de cette dernière à travers une carte interactive de *Google Maps*.

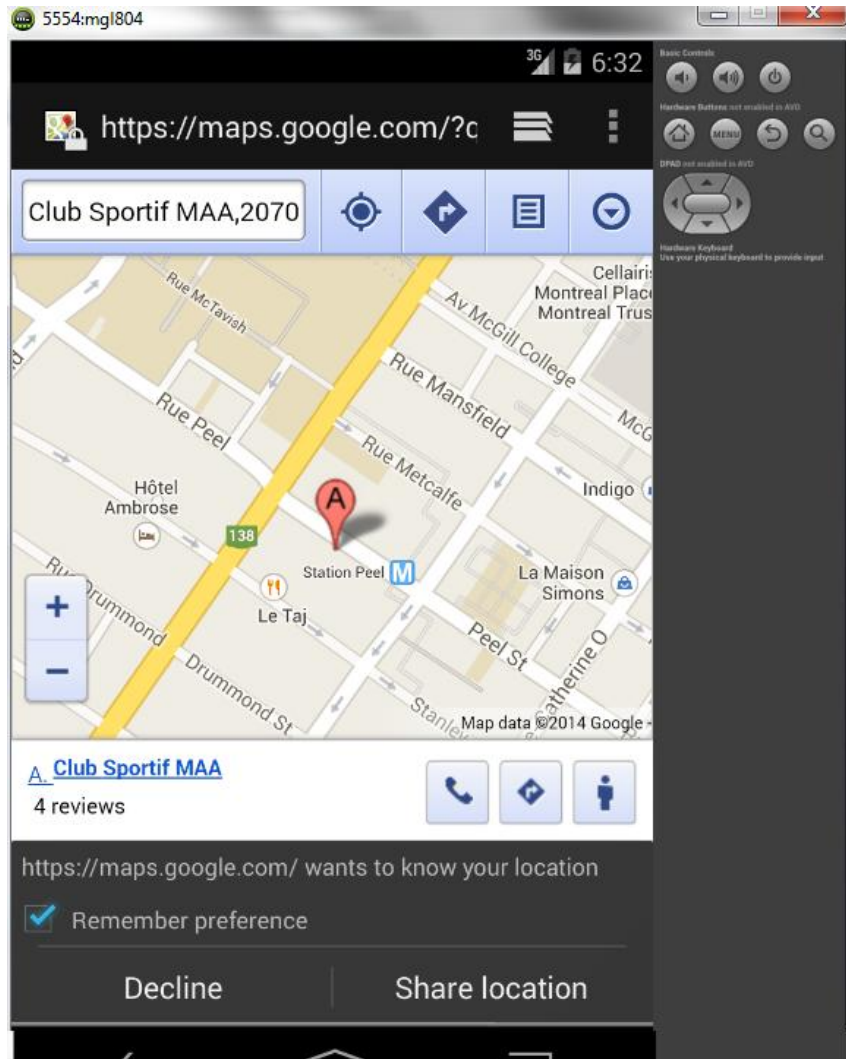


Figure 11: Géo Positionnement

8.1.7 MGLFIX-20 Changer le style des boutons Infos

Dans la section Infos, l'utilisateur a accès aux coordonnées du client comme son adresse Email, Facebook et Twitter. Il fallait ajouter des images pour chacun des liens des différents comptes de réseaux sociaux.

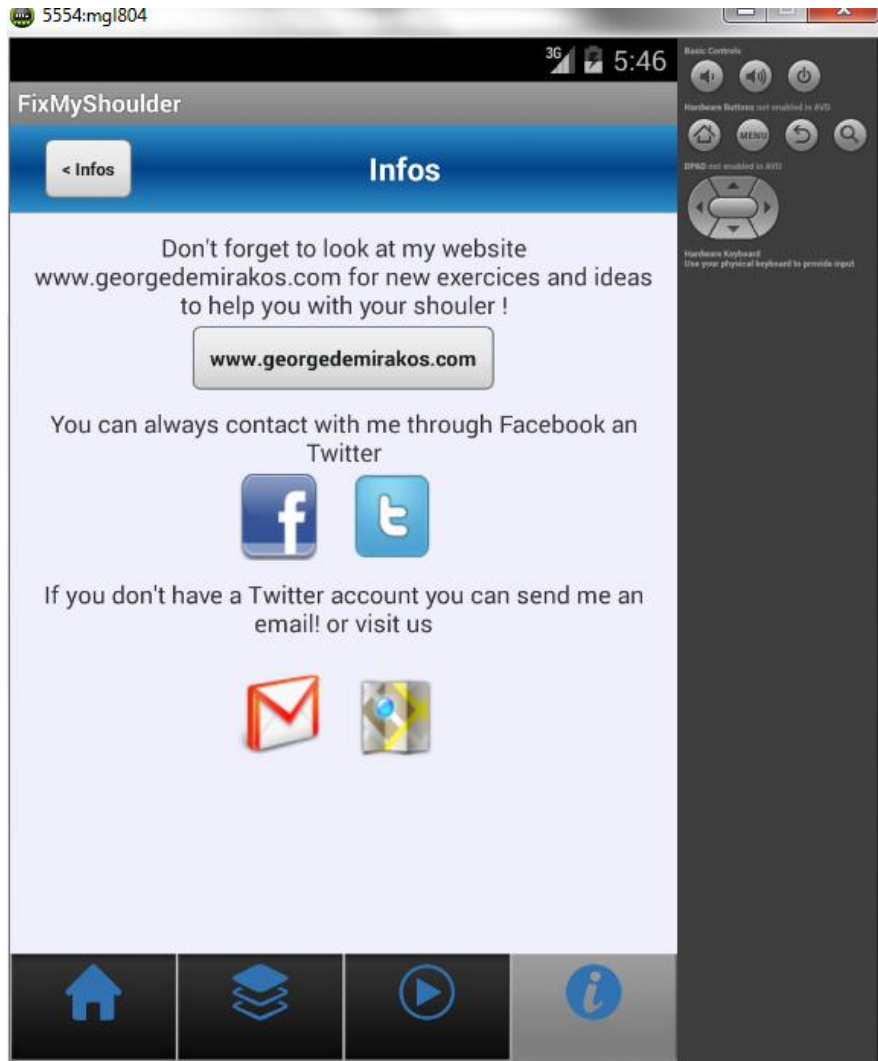


Figure 12: Bouton information

8.2 Amélioration CMS

Avant ce mandat, le client n'avait pas la possibilité de gérer à partir du CMS les vidéos de son application, la liste des vidéos était une liste statique.

Suite à notre intervention, le client peut gérer ces vidéos. Il peut télécharger les vidéos en utilisant le bouton « gérer les vidéos », il peut aussi les supprimer et choisir la vidéo qu'il souhaite mettre à la page d'accueil de son application *Androïd*.

8.2.1 MGLFIX-23 Gestion des vidéos dynamiquement à partir du CMS

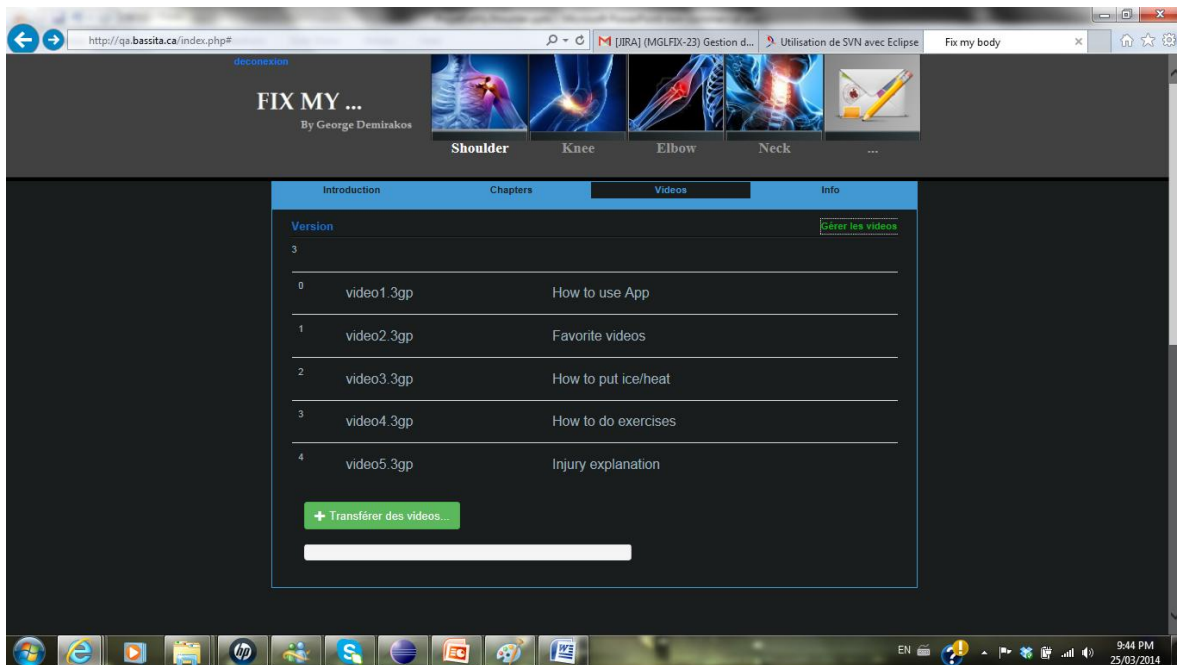


Figure 13: Gestion 1 des vidéos

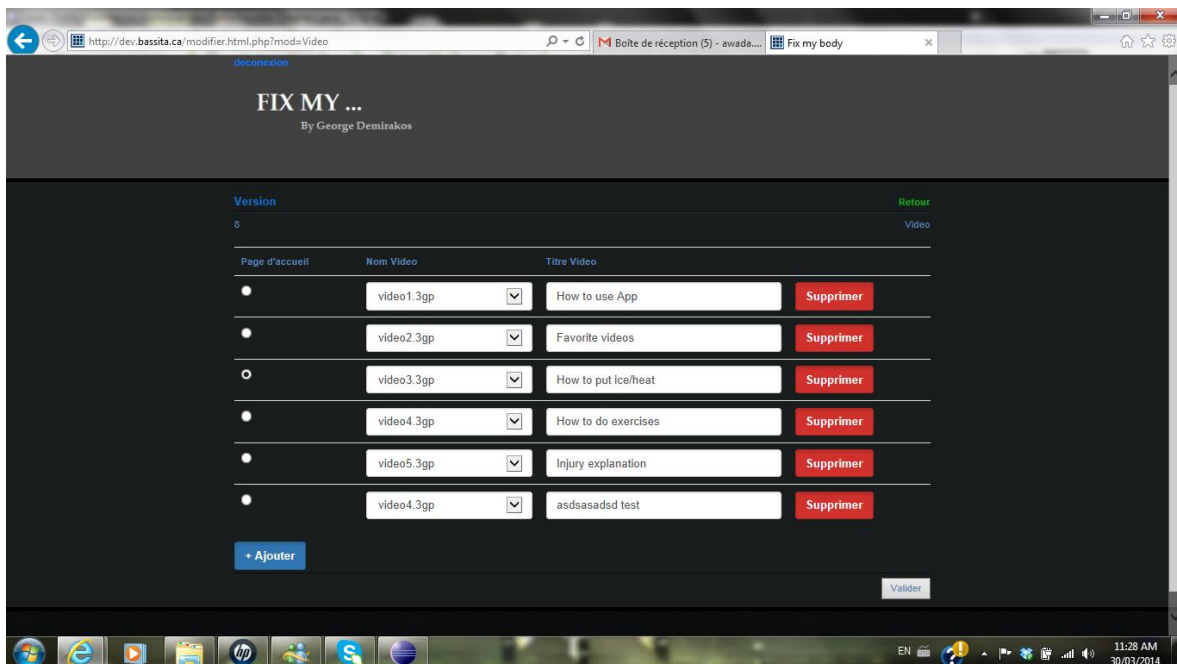


Figure 14: Gestion 2 des vidéos

CHAPITRE 9

GESTION DU CLIENT

Afin d'atteindre tous les objectifs de ce projet, il était très important de maintenir une excellente relation client-fournisseur. La gestion du client était la tâche du gestionnaire de l'équipe durant notre mandat. La gestion du client avait pour but de communiquer au client les mises à jour sur l'état d'avancement du projet. À toutes les deux semaines un communiqué était envoyé au client via courriel pour l'informer du statut des requêtes (à savoir Ouverte, À 50% ou Fermé), le progrès des requêtes en cours et les obstacles rencontrés. Les communiqués étaient aussi accompagnés de captures d'écran pour permettre au client de visualiser les changements. En utilisant cette méthodologie, nous avons pu obtenir le consentement du client en ce qui concerne les améliorations et les modifications apportées sur l'application mobile et web.

L'utilisation du courriel pour établir et maintenir une communication régulière avec le client a été retenue pour les raisons suivantes. Premièrement, le client n'était pas toujours disponible pour des rencontres ou des discussions par téléphone, considérant l'horaire de travail chargé de celui-ci. Deuxièmement, il existait un conflit entre l'horaire du gestionnaire de l'équipe et l'horaire du client. Pour ces raisons, il a été essentiel de trouver une autre méthode pour gérer le client.

La communication par courriel nous a permis de gérer le client sans avoir à gérer un conflit d'horaire et d'interrompre le client durant son travail. L'avantage de cette méthode est qu'elle permet de conserver un historique des communications entre l'équipe et le client afin de les consulter au besoin.

CHAPITRE 10

GUIDE UTILISATEUR

Afin de mieux préparer l'équipe et de mettre à jour tous les membres de l'équipe en ce qui concerne les outils, des guides utilisateurs ont été produits.

Le guide JIRA (voir l'annexe 5) avait pour but de faciliter la tâche du développeur pour l'installation et l'utilisation du système de gestion de requête. Ce document fournit le processus de maintenance logicielle 'Workflow' implémenté et fournit des explications en ce qui concerne chaque étape du processus. En outre, ce document fournit les étapes à suivre en ce qui concerne:

- La création d'une requête
- Catégorisation de la requête
- Prioritisation de la requête
- Assignment de la requête
- Gestion de la requête

Le guide Installation-Utilisation-SVN (voir l'annexe 6) avait pour but de fournir les étapes à suivre en ce qui concerne l'installation du système du contrôle de version et l'utilisation de ce système. Le document fournit les étapes à suivre en ce qui concerne:

- L'installation du système Tortoise/SVN
- Les étapes à suivre afin d'accéder au code source du projet
- Les étapes à suivre afin de mettre à jour le code source local
- Les étapes à suivre afin de commettre les changements dans le répertoire du projet

Les guides utilisateurs fournissent des captures d'écran avec des explications afin de mieux assister le développeur et d'améliorer la compréhension de l'utilisation du système.

CONCLUSION

En conclusion, nous terminons ce rapport de projet de maintenance logicielle en répondant à quelques questions essentielles. À savoir, quelle a été la capacité de l'équipe de maintenance à gérer le nombre de requêtes levées dans le cadre du projet ? Quel a été l'apport de l'équipe au « projet », dans le sens où l'application ayant fait l'objet des interventions de l'équipe avait un passé et aura nécessairement un futur ? Quelles ont été les difficultés rencontrées tout au long de son mandat de maintenance pour l'application *FixMyShoulder* ? Et les risques liés à la réalisation du projet, puisqu'il y en a toujours ? De même, en cette fin de mandat pour l'équipe de maintenance, nous mentionnerons les leçons apprises dans le cadre du projet et ce qui pourrait éventuellement être réalisé différemment.

L'équipe maintenance du projet *FixMyShoulder* a rencontré quatre (4) difficultés majeures. Chacune d'elles s'est révélé un défi et a nécessité des prises de décision vigoureuses du gestionnaire de projet. Premièrement, l'équipe a dû déplorer la perte d'un membre pour des raisons personnelles. Cette personne était impliquée tant au niveau de la gestion des systèmes de gestion de projet et de la plateforme *CVS Tortoise*, que d'une partie de la rédaction des livrables. Ne pouvant recruter une nouvelle ressource, le chef d'équipe a réparti au mieux des capacités et des compétences des autres membres la charge de travail et les responsabilités de cette personne. Deuxièmement, le processus de définition d'une version préliminaire et par incrémentations du processus de maintenance (défini dans le système de gestion des billets *JIRA*). Dans ce cas, le chef d'équipe, accompagné de différents membres de l'équipe, a eu de nombreuses discussions afin de raffiner la compréhension et la formalisation du processus avec le professeur, M. *April*, et des membres d'autres équipes. Les problématiques proposées aux différentes équipes d'étudiants étant complexes et différentes ces discussions et séances de partages de réflexions ont pu avoir lieu en toute collégialité et se sont révélées très bénéfiques. Troisièmement, la difficulté de la maîtrise et l'adaptation du modèle d'entente de service ont été initialement sous-estimées. Dans ce cas, l'équipe de maintenance logicielle dirigée par son chef de projet a su reconnaître la nécessité de cheminer dans un processus

progressif de compréhension et de maîtrise du projet et de la documentation guidant devant guider à l'élaboration de ce livrable essentiel. Au fur et à mesure, il nous a été possible de déterminer les sections qui ne s'avéraient pas nécessaires ou applicables au projet *FixMyShoulder*, quels objectifs réalistes sur lesquels pouvaient s'engager l'équipe auprès du client dans les temps impartis, quelles métriques devaient servir de références dans le suivi et l'audit de complétude et de qualité du service de maintenance, le contexte réel de pratique et la capacité de l'équipe à se proposer de compléter son mandat. Cet exercice mené à terme nous a permis de présenter un document d'entente de service particulièrement bien adapté au contexte de maintenance logiciel du système *FixMyShoulder*, à l'entière satisfaction du client. Et dernièrement, l'ensemble de l'équipe a dû surmonter la difficulté d'utiliser une nouvelle technologie, soit le système d'exploitation *Android* et la librairie *API Android SDK*. À cette occasion, l'équipe a pu compter sur l'expérience et la qualité de ces membres pour s'assurer que le langage de programmation soit rapidement appris et maîtrisé ; sont venus en complément la lecture de documentations et du code du projet dans sa version initiale, le visionnement de tutoriels et la pratique.

RECOMMANDATIONS

Dans le but de laisser la prochaine équipe de maintenance libre de déterminer les tâches prioritaires et essentielles au support du système, nous ne proposerons que deux (2) recommandations dans les évolutions futures. La première, pour ne pas discriminer les utilisateurs d'un type de système d'exploitation mobile à ceux d'un autre, nous suggérons d'intégrer les changements survenus dans la version finale du système sur *Android* dans la version *IOS* d'*Apple* (système ayant fait l'objet de développement par une équipe précédente). Et dans un deuxième temps, nous suggérons l'ajout de sections dynamiques dans le CMS (de mises à jour du contenu) spécifiques aux sujets traités par le livre du client ayant servi d'inspiration au développement du système *FixMyShoulder*, soit les éléments *Elbow*, *Knee* et *Neck* (faisant l'objet de chapitres distinctifs dans le livre).

Propositions de modifications à apporter dans une version future de la partie *Android* :

- Les vidéos de l'application sont téléchargées à chaque lecture. Nous recommandons de télécharger les vidéos dans la zone de stockage du téléphone et de faire la lecture en local. Cela permettrait d'économiser sur la consommation du data et de baisser la facture du fournisseur en conséquence.
- Les coordonnées du client (email, twitter, facebook et l'adresse) sont directement codés dans le code de l'application. Il faudrait permettre au client de les changer sur le CMS comme les chapitres et les vidéos par exemple.
- L'affichage d'images illustratives à côté du texte des chapitres est vraiment essentiel dû à la nature médico-pédagogique de l'application.
- Changer les boutons du menu principal suivant le contexte de l'application. Par exemple, lorsqu'on navigue d'un chapitre à un sous-chapitre on voit plus haut un bouton 'Retour'. Ce dernier devrait être affiché sur le menu principal au lieu de prendre toute une bande dans le haut de la fenêtre. L'idée est de gagner de l'espace pour afficher plus de contenu et d'éviter ou réduire le défilement (effet de *scrolling*).

ANNEXE I

PLAN DE PROJET

ANNEXE II

CONTRAT D'ÉQUIPE

ANNEXE III

ANALYSE D'IMPACT

ANNEXE IV

SLA

ANNEXE V

GUIDE UTILISATEUR JIRA

ANNEXE VI

GUIDE UTILISATEUR SVN

LISTE DES RÉFÉRENCES BIBLIOGRAPHIQUES

École de technologie supérieure, cours de MGL804. Hiver 2014.

- [1] April A. et Abran A. 2011. *Améliorer la maintenance logiciel*. Longueuil : Loze-Dion Éditeur, 337 p.
- [2] April A. 2014. *Maintenance & support service level agreement*. Service Level Agreement – 2000 Work Program.
- [3] Crochet M. 2013. *Application mobile pour Android concernant la physiothérapie de l'épaule (FixMyShoulder) et serveur de mise à jour pour application Apple et Android de FixMyShoulder*.