ECOLE DE TECHNOLOGIE SUPERIEURE
UNIVERSITE DU QUEBEC

RAPPORT DE PROJET PRESENTE A
L'ECOLE DE TECHNOLOGIE SUPERIEURE

COMME EXIGENCE PARTIELLE
A L'OBTENTION DE LA
MAÎTRISE EN GÉNIE, CONCENTRATION
TECHNOLOGIES DE L'INFORMATION

PAR
MATHIEU CROCHET

APPLICATION MOBILE POUR ANDROID CONCERNANT LA PHYSIOTHERAPIE DE L'EPAULE (FIX MY SHOULDER) ET SERVEUR DE MISE A JOUR POUR APPLICATION APPLE ET ANDROID DE FIX MY SHOULDER

MONTREAL, LE 1 AVRIL 2013

REMERCIEMENTS

Je tiens à adresser mes remerciements à M. April, professeur au département Génie Logiciel de l'ETS, qui m'a permis de réaliser ce projet et à accepter de diriger ce travail.

Je tiens aussi à remercier M. Demirakos pour avoir proposé ce projet et qui a toujours été disponible lorsque j'avais besoin de le rencontrer.

Je remercie aussi Julie Vincent qui m'a permis de débuter le projet grâce à ses précédents travaux.

Je remercie aussi toutes les personnes qui m'ont soutenue lors de ma Maitrise.

APPLICATION MOBILE POUR ANDROID CONCERNANT LA PHYSIOTHERAPIE DE L'EPAULE (FIX MY SHOULDER) ET SERVEUR DE MISE A JOUR POUR APPLICATION APPLE ET ANDROID DE FIX MY SHOULDER

Mathieu CROCHET

RÉSUMÉ

Le rapport ci-dessous présente les travaux réalisés dans le cadre du développement d'une application pour Android ainsi que la mise en place d'un serveur permettant la mise à jour de celle-ci. Il se divise en deux principales parties : la réalisation de l'application suivie du développement du site et de la mise en place du serveur.

La première partie présente le développement d'application Android, les outils nécessaires, les langages utilisés et les standards de développement. Ensuite, les différentes étapes du développement de l'application Fix My Shoulder sont détaillées : l'analyse, la conception et l'implémentation finale.

La seconde partie concerne la section Web du projet avec la création d'un site Web permettant au client de mettre à jours les données de l'application. Ce serveur stockera aussi les vidéos disponibles.

Mots-clés : Android, site Web, serveur de mises à jour

APPLICATION MOBILE POUR ANDROID CONCERNANT LA PHYSIOTHERAPIE DE L'EPAULE (FIX MY SHOULDER) ET SERVEUR DE MISE A JOUR POUR APPLICATION APPLE ET ANDROID DE FIX MY SHOULDER

Mathieu CROCHET

<ABSTRACT>

The following report presents the work done in the context of developing an application for Android and the implementation of a server for the update. It is divided into two main parts: the realization of the application and the development of the site and setting up the server.

The first part presents the development of Android, the tools, languages used and standards development. Then, the different stages of the application Fix My Shoulder development are outlined: analysis, design and final implementation.

The second part presents the section Web project with the creation of a website that allows the customer to update the application data. This server will also store the videos.

Keywords: Android, website, server updates

TABLE DES MATIÈRES

Page

LISTE DES FIGURES

Page

LISTE DES ALGORITHMES

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

SDK : Software Development Kit

JDK : Java Development Kit

XML : Extensible Markup Language

PHP : Hypertext Preprocessor

HTML : Hypertext Markup Language

CSS : Cascading Style Sheets

INTRODUCTION

De nos jours, les téléphones intelligents sont de plus en plus populaires et utilisés pour effectuer de nombreuses choses : s'informer de l'actualité, se divertir, écouter de la musique ou encore regarder des vidéos. On trouve toutes sortes d'applications, sport, éducation, jeux, souvent orientées pour un public déjà connaisseur du sujet traité. Que ce soit sur l'App Store d'Apple ou sur le Play Store de Google, il y en a pour tout le monde.

Ce rapport présent les travaux effectués dans le cadre d'un projet de fin d'étude proposé par le physiothérapeute George Demirakos et le professeur Alain April. Ce projet est la prolongation du travail effectué par Julie Vincent, qui a développé une application mobile pour IOS. Pour ma part, je devais réaliser cette application pour Android, mettre en place un serveur permettant de stocker les vidéos ainsi qu'un site Web simplifiant la mise à jour de ces deux applications.

L'objectif était ici de mettre sous forme d'application un livre écrit par George Demirakos traitant de la réparation de l'épaule. Cette application visant à la fois un public néophyte et connaisseur de la physiothérapie de l'épaule.

Dans le premier chapitre, nous étudierons la revue de la littérature qui m'a permis de comprendre et de mettre en place le nécessaire pour le développement de l'application ainsi que le site Web.

Les second et troisième chapitres traiteront des développements et de l'organisation du travail. Nous verrons tout d'abord la méthodologie utilisée, l'analyse effectuée, la conception et enfin l'implémentation.

Pour finir, le dernier chapitre sera une discussion sur les limites de l'application et du site ainsi que les différentes possibilités d'évolution de ceux-ci.

CHAPITRE 1

Revue de la littérature

Pour la réalisation de ce projet, il m'a fallu étudier comment développer une application Android de A à Z car je ne n'avais pas de connaissance au sujet du développement d'application mobile. Je savais seulement que le langage utilisé était le Java. J'avais déjà eu l'occasion de développer des petites applications sous Windows avec ce langage durant mes études. La première étape du projet a donc été l'étude de toute la documentation concernant le sujet. Pour cela, j'ai sélectionné différents sites internet ou livres sur le développement d'application mobile Android.

Le site internet « Le Site du zéro » (http://www.siteduzero.com/) est un très bon site, pour les débutants, qui propose de nombreux tutoriels, que ce soit pour des langages de développement logiciel, internet ou encore mobile. Dans mon cas, je me suis servi de leurs tutoriels concernant le développement d'application mobile Android. Ce site est très intéressant car il nous apprend tout ce qui est nécessaire pour créer une simple application. Le site est organisé sous forme de chapitre comme un livre et donne petit à petit les informations nécessaires pour nous faire découvrir tout ce qui est réalisable grâce à de nombreux exemples et exercices.

Le site internet d'Android mis à notre disposition par Google (http://developer.android.com) est aussi un très bon site pour les débutants comme pour les plus expérimentés. Ce site est plus complet que le précédent car il contient tout ce qui est nécessaire à la réalisation d'une application Android mais aussi plein de widget que l'on peut réutiliser. Chaque outil ou fonction de base y est expliqué dans le détail. Ce qui est plus difficile va être de développer sa première application si on ne l'a jamais fait. En effet, même s'il est très complet, il n'y a aucun tutoriel pour nous guider pas à pas.

Le dernier site qui m'a été utile est http://android.developpez.com/cours/ qui lui va se situer entre les deux sites présentés ci-dessus. Celui-ci réunis de nombreux tutoriels sur de nombreux langages de développement comme le site du zéro. Cela m'a été très utile quand je voulais une autre approche de certains concepts ou tout simplement lorsque je cherchais des informations que je ne trouvais pas ailleurs.

Ces trois sites internet sont très complémentaires et m'ont permis d'acquérir toutes les connaissances nécessaires au bon déroulement du développement de l'application qu'il m'était demandé de réaliser. Pour compléter cette formation, je me suis aussi appuyé sur deux livres :

Le premier livre s'intitule « L'Art du développement Android », écrit par Mark Murphy en 2010. Ce livre regroupe toutes les bases de la programmation sous Android - de la création des interfaces graphiques à l'utilisation de GPS, en passant par l'accès aux services web et bien d'autres choses encore ! Il regorge d'astuces et de conseils pour aider à réaliser ses premières applications. A travers des dizaines de projets d'exemples, on arrive facilement à créer des applications convaincantes.

Le second livre est « Développement d'applications professionnelles avec Android 2 », écrit par Reto Meier en 2010. Cet ouvrage incontournable explique en détail comment tirer parti des caractéristiques d'Android pour créer des applications élaborées. Très complet, il est fondé sur les toutes dernières nouveautés du SDK, il passe en revue la plateforme de développement et apporte l'incomparable savoir-faire de son auteur au travers de nombreux exemples et de fonctionnalités au fil des chapitres. Ce livre nécessite tout de même une base de connaissances du développement Android.

Chaque livre ou site internet m'a aidé lors du développement de l'application et m'a apporté de nombreuses connaissances sur le domaine. Cela m'a permis de répondre au mieux aux demandes du client.

La seconde étude littérature m'a servi pour acquérir les nouvelles connaissances qui allaient m'être utile pour le développement du site Web qui a pour but de mettre à jour les applications Fix My Shoulder pour IOS et Android. Le milieu du Web est un domaine que je connais plutôt bien car je l'ai beaucoup étudié durant mes études ou mes stages. J'ai tout de même voulu me mettre à jour car c'est un domaine qui évolue rapidement. Pour cela, je me suis appuyé sur différents livres et sites internet.

Le premier livre s'intitule « PHP et MySQL » de Luke Welling, publié en 2009. Ce livre m'a beaucoup aidé quand j'ai débuté avec le PHP et les bases de données MySQL. Même si MySQL n'est pas utilisé dans ce projet, ce livre est un ouvrage complet la réalisation d'applications web, des fonctions les plus simples aux plus compliquées.

Le second livre est « JavaScript : Introduction et notions fondamentales » de Luc Van Lancker, sorti en 2008. Il s'adresse à un public de lecteurs initiés qui désirent d'ajouter des éléments dynamiques et de l'interactivité aux pages Web qu'ils développent. Cette connaissance du JavaScript est la base indispensable pour accéder aux techniques de création de sites plus évoluées que sont le Dhtml et l'Ajax.

En ce qui concerne les sites internet, il y a « Le Site du zéro » cité ci-dessus. Comme pour une application Android, ce site nous guide pas à pas dans l'élaboration d'un site mêlant HTML et PHP. On comprend facilement comment cela fonctionne grâce à de nombreux exemples ou exercices à faire chez soi.

Un autre site très intéressant est php.net (www.php.net). C'est le site officiel de PHP qui propose de nombreux éléments comme le téléchargement des sources et manuels de références en ligne. On y retrouve aussi toutes les fonctions ainsi que le détail de leur fonctionnement très facilement.

L'étude de ces deux littératures m'a permis d'acquérir toutes les connaissances nécessaires au bon déroulement du projet qui a pour objectif la réalisation d'une application Android

respectant certaines contraintes. Cette application a pour but de permettre au physiothérapeute George Demirakos de partager ses connaissances et offrir une documentation claire et simplifiée sur les différents traitements de l'épaule possibles. La seconde partie du projet consiste à développer un site Web permettant la mise à jour des textes présents dans les applications IOS et Android de Fix My Shoulder. Ce site sera stocké sur un serveur avec les vidéos que les utilisateurs pourront consulter en utilisant l'application.

CHAPITRE 2

Développements Android

La première partie de ce projet est le développement de l'application Android. Celle-ci a pour but de présenter les travaux du physiothérapeute George Demirakos. C'est une représentation numérique d'un livre sur les différents traitements de l'épaule avec des exercices simples adaptés à tous. Cette application existe déjà pour IOS d'Apple.

Le déroulement de ce développement a été effectué en plusieurs étapes : l'analyse de l'existant et identification des contraintes, le choix de la méthodologie adoptée, la définition des normes à respecter, la mise en place de la conception et la réalisation de celle-ci avec l'implémentation.

2.1 Analyse

Cette application étant déjà existante pour les téléphones Apple, l'étude des besoins du client et la création du cahier des charges a été réalisée par Julie Vincent, qui a développé la version IOS. C'est grâce à elle que j'ai pu débuter le développement très rapidement. Je me suis alors appuyé sur ses travaux pour commencer et j'ai aussi cherché les améliorations possibles sachant que j'apportais un œil extérieur au projet.

Après avoir étudié les travaux de Julie, j'ai compris que l'application était divisée en quatre parties.

- L'introduction : c'est la première page de l'application, avec l'introduction du livre écrit par G. Demirakos. Elle nous présente le reste de l'application et le contenu des autres chapitres.

- Les chapitres : on y retrouve la liste des chapitres disponibles. Chaque chapitre étant lié à des sous chapitres et leurs contenu.

- Les vidéos : nous présentent une illustration de certains exercices.

- Les informations : nous donnent des détails sur l'auteur du livre, les copyrights ou encore les différentes façons de contacter l'auteur ou de partager l'application.

Voici un diagramme représentant une vue d'ensemble de l'application basé sur celui réalisé par Julie Vincent :

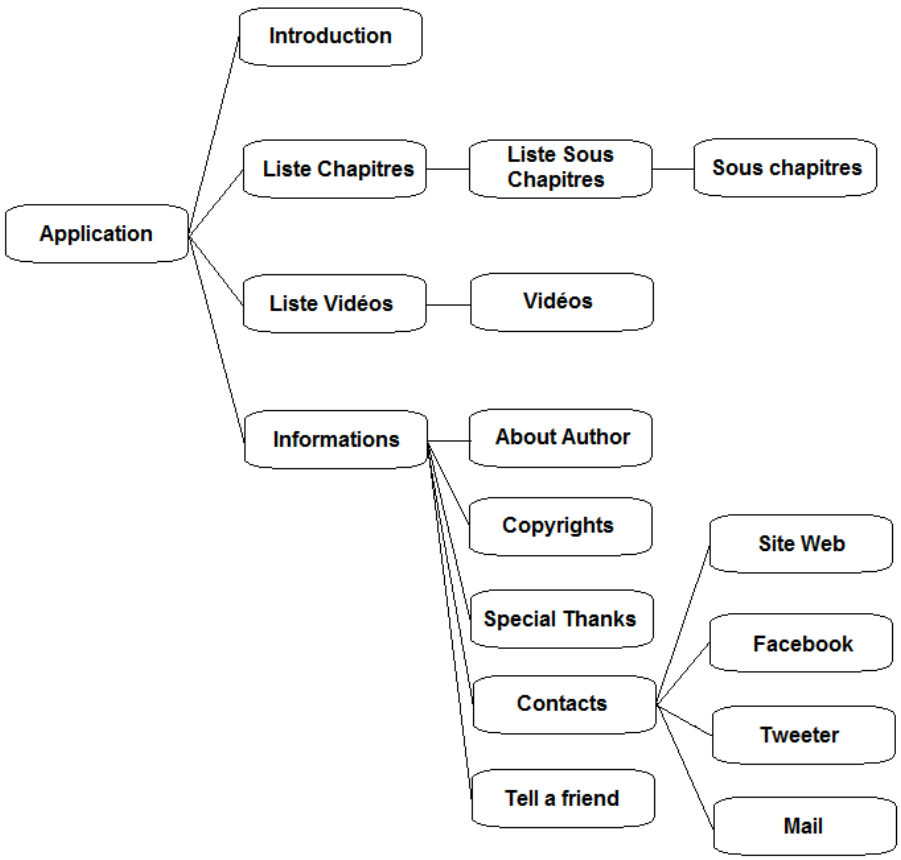


Figure 2.1 - Vue d'ensemble de l'application

Adapté de Julie Vincent

2.2 Exigences et contraintes

2.2.1 Exigences fonctionnelles

L'application doit respecter certaines exigences telles qu'une navigation intuitive, facile et rapide entre les différentes sections et entre le contenu de chacune d'entre elles. Elle doit aussi permettre de se connecter à internet pour effectuer les mises à jour et lire les vidéos car celles-ci ne sont pas stockées dans l'application. Enfin, elle doit assurer un affichage correct des informations en tout temps.

2.2.2 Exigences non fonctionnelles

Quant aux exigences non fonctionnelles, celles-ci concernent la convivialité, la maintenabilité, la performance et la compatibilité de l'application.

- Conviviale : l'application doit être compréhensible par toutes les personnes qui vont l'utiliser, néophytes ou expertes. Celle-ci doit permettre de trouver facilement et rapidement ce que l'on cherche.

- Maintenable : l'application doit pouvoir être mise à jour facilement en se connectant à internet à chaque fois qu'une modification de texte est effectuée.

- Performante : l'application doit se lancer rapidement et répondre aux besoins de l'utilisateur dans un délai respectable.

- Compatible : l'application doit être compatible avec les différentes versions Android existantes dans le milieu de la téléphonie mobile.

2.2.3 Contraintes

Les principales contraintes que je devais respecter étaient celles définies par Julie lors de la conception de la version IOS : La disposition des sections les unes par rapport aux autres, la présentation des informations et couleurs générales de l'application.

Les autres contraintes concernent les standards Android. L'application doit être supportée par tous les types d'écran disponibles. En effet, contrairement à IOS qui ne possède que deux tailles distinctes, Android est disponible sur de nombreux formats. Il faut donc en tester un maximum pour être sûr que tous les utilisateurs soient satisfaits. Cela dépend de la taille et de la qualité des images ou des différentes interfaces qui composent l'application. De plus, il existe d'autres pratiques à suivre pour que les utilisateurs apprécient au maximum l'application et partage celle-ci plutôt qu'ils la désinstallent. On peut citer le fait de ne pas bloquer le téléphone lorsque l'application est active, d'optimiser les algorithmes pour de meilleures performances, d'adapter les interfaces aux écrans mais aussi aux utilisateurs (éviter les petits boutons pour ceux qui ont des gros doigts, etc.) et d'effectuer un maximum de tests.

2.3 Méthodologie

Pour développer cette application, j'ai choisi le système d'exploitation Android car c'est, à l'heure actuelle, le plus utilisé pour téléphones intelligents et tablettes. D'autant plus que la version IOS a déjà été développée.

2.3.1 Présentation Android

Android respecte les principes de l'open source, ce qui signifie que n'importe qui peut avoir accès aux sources et les modifier. C'est un système extrêmement portable qui s'adapte à de nombreuses structures et est utilisé dans de divers domaines (comme par exemple dans

certains micro-ondes). De plus Android met tout en place pour que toutes les applications soient faciles à développer et à distribuer. Basé sur un noyau Linux, ce système d'exploitation comporte une interface développée principalement en Java pour une meilleure portabilité.

2.3.2 Procédure de travail

Le travail réalisé pour la première partie du projet, le développement de l'application Android, s'est déroulé en plusieurs étapes. La première consistait en l'étude de la technologie Android pour la réalisation de l'application, la deuxième concernait l'étude de l'existant et des travaux de Julie Vincent et la troisième était le développement de l'application au travers de la conception, l'implémentation et pour finir les tests pour s'assurer du bon fonctionnement de l'ensemble.

L'ensemble de cette partie du projet a été un apprentissage du développement mobile, de l'étude du langage à la publication de l'application. Cette publication ne fait pas partie du projet et se fera quand le client G. Demirakos l'aura décidé.

2.3.3 Conditions nécessaires au développement

De manière générale, n'importe quel matériel permet de développer des applications Android du moment que l'on utilise Windows, Mac OS X ou une distribution Linux. Il y a bien sûr certaines limites à ne pas franchir.

Pour un environnement Windows, sont tolérés XP (en version 32 bits), Vista (en version 32 et 64 bits) et 7 (aussi en 32 et 64 bits). Officieusement (en effet, Google n'a rien communiqué à ce sujet), Windows 8 est aussi supporté en 32 et 64 bits.

Sous Mac, il est nécessaire d'utiliser Mac OS 10.5.8 ou plus récent et un processeur x86.

Sous GNU/Linux, Google conseille d'utiliser une distribution Ubuntu plus récente que la 10.04. Enfin de manière générale, n'importe quelle distribution convient à partir du moment où la bibliothèque GNU C (glibc) est au moins à la version 2.7.

Ensuite, en plus de la JDK (Java Development Kit) nécessaire à l'environnement de développement Java, il est indispensable d'avoir le SDK d'Android. C'est un ensemble d'outils que met à disposition un éditeur afin de permettre de développer des applications pour un environnement précis. Le SDK Android permet donc de développer des applications uniquement pour Android.

2.4 Standards et normes

Lorsque l'on veut développer une application mobile, il y a certaines normes à respecter. Celles-ci sont nécessaires pour que le code soit interprété correctement et surtout pour être compris facilement lorsqu'on veut le mettre à jour ou le reprendre.

Google, qui est le distributeur d'Android, a mis en place des normes que tous les développeurs doivent suivre. Parmi les points les plus importants, on retrouve l'interdiction d'utiliser des noms de fichier ou d'icônes trop proches des applications système pour empêcher de mauvaises interprétations ou certains bugs. Il y a aussi l'interdiction d'autoriser à notre application d'afficher des informations personnelles, de créer des raccourcis, des favoris ou modifier les réglages par défaut du téléphone sans en informer l'utilisateur et lui en demander son autorisation au préalable.

D'autres normes vont concerner les formats vidéos ou images ainsi que leurs tailles pour que celles-ci s'adaptent facilement à toutes les tailles d'écran disponibles sur le marché.

2.5 Conception

2.5.1 Conception de l'application

Pour réaliser les demandes du client pour le développement de l'application Android, j'ai commencé par analyser les travaux de Julie Vincent. Ceux-ci m'ont permis d'avoir une vue d'ensemble de l'application, la composition de chaque page qui sera affichée à l'écran ainsi que les transitions entre celles-ci.

La vue d'ensemble, décrite précédemment, m'indiquait que l'application se divisait en quatre grandes catégories qui sont l'introduction, les chapitres, les vidéos et les informations (sur l'auteur ou comment le contacter). Pour respecter cela, j'ai décidé d'utiliser un système composé de quatre onglets pour que l'utilisateur puisse changer de catégorie de façon simple et rapide sans avoir à revenir au menu principale. Certaines catégories, comme les chapitres, permettent d'afficher de nouvelles pages telles que les sous chapitres. Il faut alors permettre à l'utilisateur de changer d'onglet sans pour autant perdre la page qu'il visionnait, s'il décide de revenir en arrière.

La composition de chaque page m'a donné de nombreuses indications sur le visuel de l'application : les couleurs utilisées ainsi que l'organisation des éléments composant une page. Chaque page possède un bandeau en haut de l'écran, indiquant le nom de la page affichée et la possibilité de revenir à la page précédente. En bas de chaque page on retrouve les quatre onglets qui permettent de changer de catégorie. Le corps d'une page, lui est composé principalement de textes ou de listes (chapitres, vidéos, etc.). La page d'introduction comprend seulement le texte de celle-ci. La page des chapitres est composée de différentes pages, une qui liste les chapitres, une autre qui liste les sous chapitre d'un chapitre affichant aussi une citation décrivant le chapitre et une dernière page pour le texte du sous chapitre. La section vidéo est composée d'une liste de vidéos ainsi qu'une page pour les visionner. Enfin, la section information comprend une liste des différentes informations

disponibles dans cette partie. Chaque lien ouvre sur une page avec davantage de détails sur l'information sélectionnée.

Pour finir, les transitions entre les pages se font grâce aux onglets, aux listes ou aux boutons retour en haut de l'écran si cela est possible.

2.5.2 Langages utilisés

2.5.2.1 Java

Le principal langage utilisé pour écrire des programmes Android est le Java, un langage orienté objet, utilisé pour développer de nombreux logiciels. Cependant, tous les programmes fonctionnant sur la machine virtuelle de Java (JVM : Java Virtual Machine), ne fonctionnera pas forcément sous Android. Celui-ci étant destiné à des appareils mobiles ayant peu de puissances (par rapport à un ordinateur classique), Google a développé sa propre machine virtuelle : Dalvik. Cette machine se base sur JVM mais certaines classes disponibles sous JVM ne le sont pas sous Dalvik.

Un programme Java est constitué de Classe d'objet, combinant à la fois les données utilisées (appelées Propriétés) et le code manipulant celles-ci (appelé Méthodes). Le code devient logiquement découpé en petites entités cohérentes et devient ainsi plus simple à maintenir et plus facilement réutilisable, étant intrinsèquement modulaire.

Algorithme 2.1 - Exemple classe Java

```
Public class Toto
{
      int age;
      char sexe;
      float taille;

      Infos(int age, char sexe, float taille)
      {
            System.out.println(" age : %d, sexe : %c et taille : %f",
            age,    sexe, taille);

      }
}
```

Dans cet exemple, la classe est « Toto », les propriétés sont l'âge, le sexe et la taille et la méthode est « Infos » qui va manipuler les propriétés en les affichant.

2.5.2.2 XML

Le second langage nécessaire à connaitre pour développer une application Android est le XML qui signifie eXtensible Markup Language (en français : langage extensible de balisage). XML n'est pas un langage de programmation, il n'y a pas de boucle for, de if, de while,..... . Il est presque exclusivement utilisé pour stocker (ou transférer d'un programme à un autre) des données (du texte) de façon structurée.

Ce langage utilise des balises (comme le HTML). Ces balises sont ouvrantes, <balise_ouvrante> ou fermantes, </balise_fermante>. Chaque balise ouvrante devra être fermée à un moment. Il est possible de mettre du texte entre la balise ouvrante et la balise fermante. Les balises peuvent être imbriquées : on peut insérer un ou plusieurs couples de balises (ouvrante et fermante) entre 2 balises (ouvrante + fermante) voici un exemple permettant de stocker des informations à propos des films d'un cinéma :

Algorithme 2.2 - Exemple XML

```
<cinema>
	<film>
		<nom>Les deux tours</nom>
		<realisateur>P Jackson</realisateur>
		<annee_sortie>2002</annee_sortie>
	</film>
	<film>
		<nom>Bladerunner</nom>
		<realisateur>R Scott</realisateur>
		<annee_sortie>1982</annee_sortie>
	</film>
</cinema>
```

La balise <cinema> est appelée "élément racine", la balise <film> est le "premier enfant", etc. Un fichier XML doit posséder un "élément racine". Lorsqu'il n'y a pas de texte entre la balise ouvrante et la balise fermante, on peut les remplacer par <balise/> (qui est à la fois la balise ouvrante et la balise fermante). Une balise peut contenir des attributs :

<balise nom="Toto" prénom="Jean Pierre"/>

Nom et prénom sont des attributs qui seront utilisés par un programme tiers. Un fichier XML doit toujours commencer par une ligne appelée prologue :

<?xml version="1.0"encoding="utf-8"?>

Cette ligne précise la version de XML utilisée (ici, c'est la version 1.0) et le type d'encodage pour le document. Voici un exemple de fichier XML utilisé lors du développement d'applications sous Android.

Algorithme 2.3 - Exemple XML Android

```
<?xml version="1.0"encoding="utf-8"?>
<LinearLayout
      xmlns:android=http://schemas.android.com/apk/res/android
      android:orientation="vertical"
      android:layout_width="fill_parent"
      android:layout_height="fill_parent">

      <TextView
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="@string/hello"/>

</LinearLayout>
```

Ces notions seront nécessaires pour comprendre les extraits de code dans la partie Implémentation. Mais connaitre ces deux langages ne suffit pas, il faut un ou plusieurs outils pour les utiliser et les mettre en place.

2.5.3 Outils utilisés

Pour le développement de l'application, je me suis servi le logiciel Eclipse, qui est très utilisé pour le développement d'application Java plus classique. De plus, ce logiciel est gratuit, puissant et fortement recommandé par Google dans la documentation officielle d'Android.

Néanmoins, Eclipse ne suffit pas pour les applications Android, il est nécessaire de télécharger deux autres composants :

- le SDK (software development kit) Android contient tous les "outils" utiles permettant de développer des applications sous Android, mais Eclipse n'est toujours pas prêt pour créer des applications Android.

- Le plugin "Android pour Eclipse" ADT (Android Development Tools), adapte Eclipse au développement d'applications sous Android. Il permet de créer des projets avec les fichiers de base nécessaires, de tester, de déboguer et d'exporter les projets pour publier les applications.

La dernière chose à faire est de créer un émulateur appelé ici AVD (Android Virtual Device) qui « fait croire » à l'ordinateur qu'il est un appareil sous Android. Cet émulateur permet de tester l'application sur toutes les plateformes Android disponibles sans même avoir l'appareil chez soi.

2.6 Implémentation

2.6.1 Implémentation Android

Pour réaliser et mettre en place la conception, j'ai commencé par développer toutes les pages que contient l'application. Chez Android, une application est composée d'un ensemble de fenêtres entre lesquelles l'utilisateur navigue. Chaque fenêtre affiche une page, chaque page est constitué de deux fichiers, une activité et une vue.

- L'activité : est une composante principale d'une application qui représente à la fois l'implémentation et les interactions des interfaces. Elle est entièrement développée en Java. Chaque activité possède sa propre interface et remplit

tout l'écran, il n'y a donc qu'une seule activité d'affichée à la fois. L'activité gère les interfaces et les éléments visuels qui fournissent des informations ou avec lesquels les utilisateurs peuvent interagir. Elle sert aussi de support pour ces interfaces. Cependant, ce n'est pas le rôle de l'activité que de créer et de disposer les éléments graphiques, elle n'est que l'échafaudage sur lequel vont s'insérer les objets graphiques. De plus, une activité contient des informations sur l'état actuel de l'application : ces informations s'appellent le « context ». Ce « context » constitue un lien avec le système Android ainsi que les autres activités de l'application, comme le montre la figure suivante.

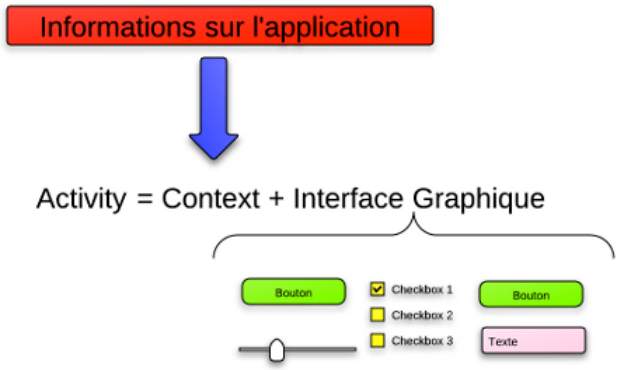


Figure 2.2 – Fonctionnement Android
Tiré du siteduzero.com

Une activité possède aussi différents états qui indiquent au système comment réagir en fonction de certaines actions. Par exemple, lorsqu'on reçoit un appel, il est plus important qu'on puisse y répondre que de continuer d'utiliser une application. Pour cela, chaque application possède une priorité plus ou moins élevée : quand une application se lance, elle se met en haut de ce qu'on appelle la pile d'activités. Dans ce cas, la pile est construite avec une structure de données de type « LIFO » (Last In First Out) qui signifie que le dernier élément ajouter sera le seul visible et sera le premier à sortir. En effet,

lorsqu'on ajoute une activité, celle-ci viendra se placer au sommet. Quand cette activité sera retirée, c'est la seconde activité, anciennement première qui prendra la place du sommet comme illustrée ci-dessous :

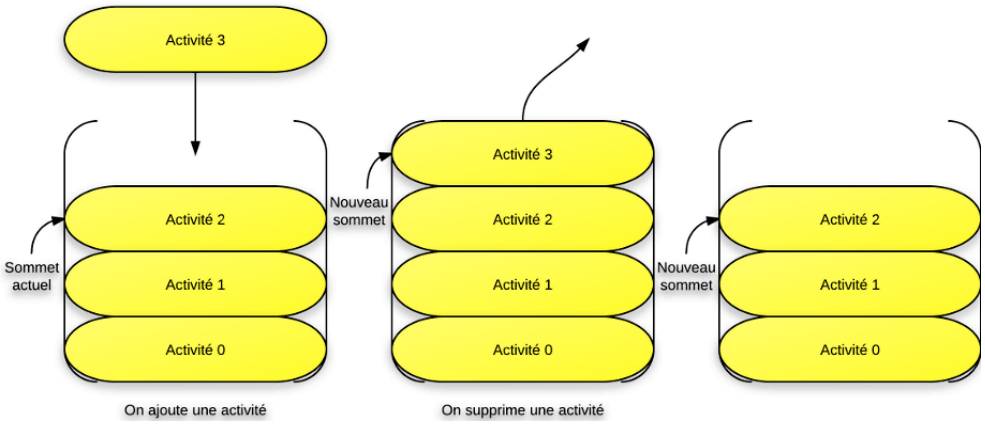


Figure 2.3 – Gestion FIFO des activités
Tiré du siteduzero.com

Une activité qui perd sa première place ne reviendra qu'une fois que toutes les activités qui se trouvent au-dessus d'elle seront finies. On retrouve ainsi le principe expliqué précédemment. Une seule application est visible sur le terminal dans laquelle on n'aperçoit seulement l'interface graphique de l'activité en cours.

Tableau 1 - Les états d'une activité

adapté du siteduzero.com

Etat	**Visibilité**	**Description**
Active	L'activité est visible en totalité.	Elle est sur le dessus de la pile, c'est ce que l'utilisateur consulte en ce moment même et il peut l'utiliser dans son intégralité.
Paused	L'activité est partiellement visible à l'écran. C'est le cas quand on reçoit un SMS et qu'une fenêtre semi-transparente se pose devant l'activité pour afficher le contenu du message et permet d'y répondre par exemple.	Ce n'est pas sur cette activité qu'agit l'utilisateur. L'application n'a plus le focus, c'est l'application sous-jacente qui l'a. Pour que notre application récupère le focus, l'utilisateur devra se débarrasser de l'application qui l'obstrue, puis l'utilisateur pourra à nouveau interagir avec.
Stopped	L'activité est tout simplement oblitérée par une autre activité, on ne peut plus la voir.	L'application n'a évidemment plus le focus, puisque l'utilisateur ne peut pas la voir, il ne peut pas agir dessus. Le système retient son

		état pour pouvoir reprendre, mais il peut arriver que le système tue l'application pour libérer de la mémoire système.

Pour résumer, voici une illustration du cycle de vie d'une activité :

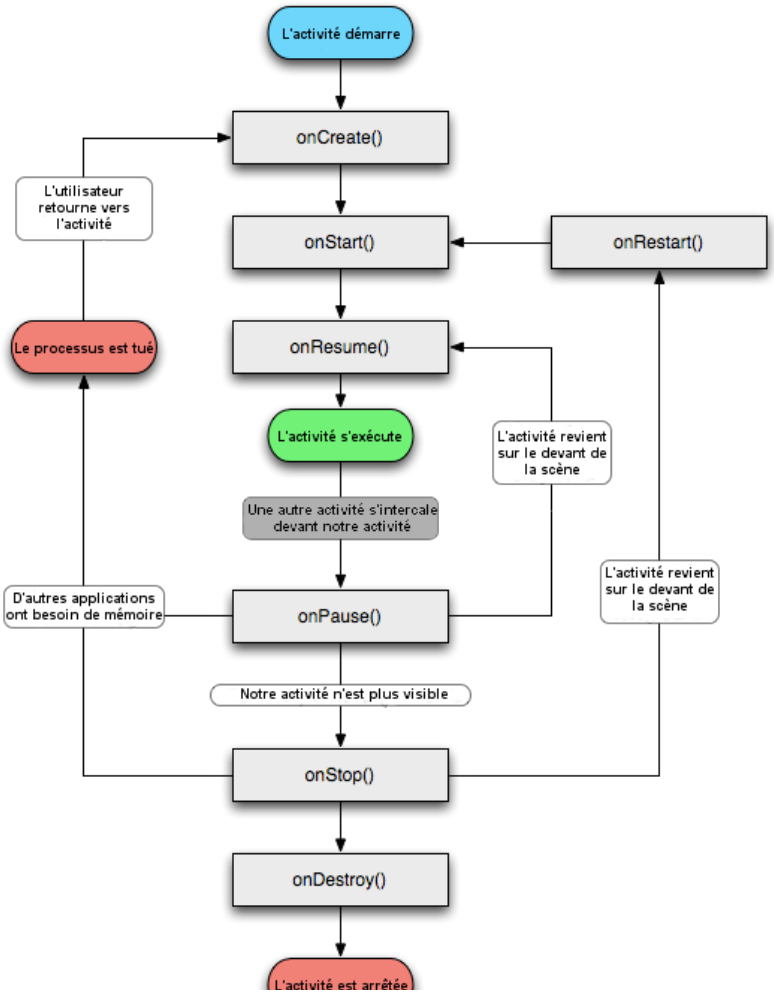


Figure 2.4 – Cycle de vie d'une activité

Tiré du siteduzero.com

- La vue : elle gère la mise en place de l'interface de l'activité. Son rôle est de fournir le contenu visuel sur lequel l'utilisateur pourra interagir. C'est elle qui va créer et disposer les éléments graphiques. Cette interface est codée en XML.

2.6.2 Développement de l'application

Dans cette partie, je vais détailler la création de chaque type de page de l'application en m'appuyant à chaque fois sur un élément de développement.

Chaque page possède en son haut un titre, centré de couleur blanche sur fond dégradé orangé et un menu d'onglet en bas pour pouvoir changer de catégorie rapidement et facilement comme on peut le voir sur l'image suivante :

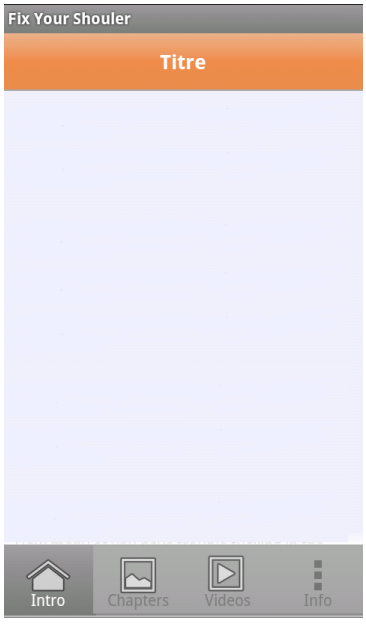


Figure 2.5 - Interface d'une page de l'application

Seul le corps d'une page change et nécessite alors une nouvelle activité et une nouvelle vue.

2.6.2.1 L'introduction

L'introduction est une partie assez simple de l'application car elle n'est constituée que d'une seule page présentant le reste de l'application et surtout les chapitres. Comme les autres pages, elle possède un titre en haut et des onglets en bas. Seul son corps est différent et est constitué uniquement d'un texte. Pour intégrer un texte dans un page Android, voici le code à insérer dans l'activité :

Algorithme 2.4 - Intégration du texte de l'introduction

```
@Override
public void onCreate(Bundle savedInstanceState)
{
	super.onCreate(savedInstanceState);
	setContentView(R.layout.activity_intro);

	// Texte de l'introduction
	TextView text = (TextView) findViewById(R.id.intro);
	text.setText(Html.fromHtml(getResources().getString(R.string.intro))
	);
}
```

Ici, @Override est facultatif mais permet d'indiquer au système que l'on va redéfinir la méthode onCreate(). « R » correspond aux ressources de l'application. R.layout fait référence aux vues, R.id fait référence aux éléments d'une vue ayant l'identifiant correspondant, etc. L'instruction « setContentView(R.layout.activity_intro) » signifie que l'on va lier l'activité à la vue « activity_intro ». L'instruction suivante « TextView text = (TextView) findViewById(R.id.intro) » indique que l'on va créer un objet de type TextView qui correspond à l'élément ayant pour identifiant « intro » dans la vue liée précédemment. La ligne qui suit donne une valeur à la variable « text » qui sera affichée à l'écran. Ici, nous voulons donner pour valeur le texte de l'introduction. Nous allons chercher le texte correspondant avec l'instruction « getRessources().getString(R.string.intro) » dans le fichier String.xml ayant pour identifiant « intro ». « Html.fromHtml » signifie que le texte à afficher

est en format Html et doit être interprété comme cela par le système pour un affichage correct.

Je détaillerais la gestion des textes présents dans l'application dans la partie « Gestion des textes ».

Cependant, cela n'affiche pas encore le texte à l'écran, la création d'une vue est obligatoire. Ici on utilise un « TextView ». Dans notre cas, le texte est plus long que la place disponible sur l'écran, il faut ajouter un « ScrollView » qui permet à l'utilisateur de faire défiler le texte et ainsi le voir dans son intégralité.

Algorithme 2.5 - Vue de l'introduction

```
<ScrollView
      android:id="@+id/scrollView1"
      android:layout_width="match_parent"
      android:layout_height="wrap_content" >

      <TextView
            android:id="@+id/intro"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/intro"
            style="@style/texte"
            />
</ScrollView>
```

On retrouve le « TextView » appelé dans l'activité avec l'identifiant « intro ». On remarque que chaque élément possède des caractéristiques précédées de « android: ». Id signifie l'identifiant, width et height respectivement la hauteur et la largeur, text le texte par défaut et style, le style que l'on veut donner à l'élément.

Ainsi, nous pouvons voir à l'écran :

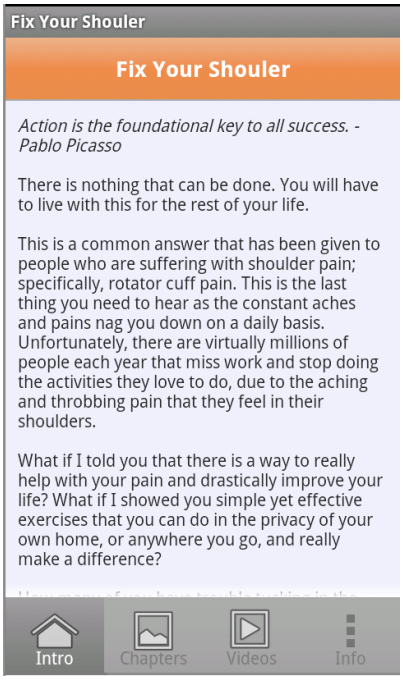


Figure 2.6 - Introduction de l'application

2.6.2.2 Les chapitres

Cette partie est celle qui contient le plus d'informations car elle présente les chapitres, sous chapitres ainsi que les textes de chacun. Pour cela, j'ai développé trois pages qui ont chacune une activité et une vue. Le corps de la première page contient une liste de différents chapitres contenue dans l'application. Voici le code de l'activité :

Algorithme 2.6 – Récupération des informations des chapitres

```
vue = (ListView) findViewById(R.id.chapters);

String[] chap = getResources().getStringArray(R.array.chap_list);

String[] chap_titre =
getResources().getStringArray(R.array.chap_titre_list);
String[] chap_num = getResources().getStringArray(R.array.chap_num);
```

Dans cette partie, comme pour l'introduction, j'ai récupèré un élément de la vue associé à l'activité grâce à son identifiant. Cet élément est une « ListView » et il permet de lister les items présents dans celui-ci dans la vue en fonction d'un adaptateur que je présenterais ensuite. Les trois lignes suivantes créés trois variables, « chap » qui contient la liste des chapitres, « chap_titre » contient le titre des chapitres et « chap_num », qui contient les numéros de chaque chapitre.

Algorithme 2.7 - Création de la liste des chapitres

```
ArrayList<HashMap<String, String>> liste = new ArrayList<HashMap<String,
String>>();

HashMap<String, String> element = null;

Integer nb_chap = chap.length;
```

Ensuite, j'ai créé une ArrayList « liste », une liste qui contint trois informations par ligne : le nom du chapitre, le titre et son numéro. Cette liste va contenir les éléments de la variable « element ». La dernière ligne récupère le nombre de chapitre qui correspond à la longueur du tableau « chap ».

Algorithme 2.8 - Création de chaque ligne de la liste

```
for(int i = 0 ; i < nb_chap ; i++)
{

      element = new HashMap<String, String>();

      element.put(KEY_TITLE, chap[i]); // titre
      element.put(KEY_SUBTITLE, chap_titre[i]); // sous titre
      element.put(KEY_THUMB_URL, chap_num[i]); // numero

      liste.add(element);
}

adapter=new LazyAdapter(this, liste);

vue.setAdapter(adapter);
```

Pour finir, j'ai rempli la variable « element » avec les informations des variables « chap », « chap_titre » et « chap_num ». Chaque « element » est ensuite ajouté dans la liste « liste ». L'itération « adapter = nem LazyAdapter(this, liste) » associe cette liste à l'adaptateur pour la mettre en forme, « vue.setAdapter(adapter) » va ensuite l'associer à la vue.

Voici comment se présente le code de la vue pour afficher cette liste à l'écran :

Algorithme 2.9 - Création d'une ListView dans une vue

```
<ListView
      android:id="@+id/chapters"
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:divider="#b5b5b5"
      android:dividerHeight="1dp"
      android:listSelector="@drawable/list_selector">
</ListView>
```

On retrouve des caractéristiques similaires à la vue "intro" mais cette fois-ci on affiche une ListView. L'important ici étant la caractéristique « listSelector », qui donne le style de la liste.

L'adaptateur dont je parlais dans l'activité plus haut est en fait une autre activité avec sa propre vue qui organise seulement une ligne de la liste. De ce fait, chaque ligne aura la même apparence et permet une meilleure gestion de l'interaction et d'organisation des éléments.

Figure 2.7 – Liste des chapitres

En ce qui concerne les sous chapitres, qui sont affichés lorsque l'utilisateur sélectionne un chapitre, il y a aussi une page unique avec la vue correspondante. La vue aurait pu être la même que celle des chapitres car le corps est essentiellement composé de la liste des sous chapitres mais deux choses changent. En effet, l'auteur a attribué une citation à chaque chapitre. Celle-ci s'affiche au-dessus de la liste des sous chapitres du chapitre correspondant. La liste ne présente pas le même nombre d'information, il n'y a que le titre et le numéro du sous chapitre et non plus le nom. Il y a donc un nouvel adaptateur avec sa vue correspondante et un nouveau « TextView » dans la vue de l'activité des sous chapitres.

Le code de l'activité est sensiblement le même que celui de l'activité des chapitres sauf qu'ici ce ne sont pas les chapitres qui sont récupérés dans des listes mais les sous chapitres. Il y a autant de listes créées que de chapitres. C'est en fonction du chapitre sélectionné

précédemment que la sélection de la liste de sous chapitres à afficher sera choisie et déterminera aussi le titre de la page.

En ce qui concerne la vue, celle-ci est similaire à celle des chapitres avec un TextView en plus au-dessus du « ListView ».

Figure 2.8 - Liste des sous chapitres

Enfin pour les textes des sous chapitres, le corps contient simplement le texte du sous chapitre sélectionné comme pour l'introduction sauf que là encore il y a une différence ce qui nécessite une vue spécifique. En effet, comme pour chaque page qui s'affiche après une action effectuée sur une autre page et à partir des onglets en bas, il y a un bouton de retour qui s'affiche à côté du titre. Je détaillerais plus précisément les transitions dans la partie « Transitions » ci-dessous.

Pour le code de l'activité, celui-ci ressemble au code de l'activité des sous chapitres sauf que ce sont les textes des chapitres qui sont récupérés et affichés en fonction du sous chapitre sélectionné.

Pour la vue correspondante, le code ressemble à celui de l'introduction sauf pour le bouton retour que j'expliquerai dans la partie « Transitions ».

Figure 2.9 - Texte du sous chapitre

Figure 2.10 - Pages de la partie Chapitre

2.6.2.3 Les vidéos

La partie « vidéos » se présente en deux parties, la liste des vidéos et le visionnage de la vidéo sélectionnée. La première partie se présente de la même manière que la première page des chapitres et présente la liste des vidéos disponibles.

L'activité de cette première page est simple car elle ressemble à celle des chapitres. En effet, la liste des vidéos est chargée dans un tableau et est associée à un adaptateur, le même que pour les sous chapitres.

La vue de cette page est simple et présente le même code que celle des chapitres, un titre ainsi qu'une « ListView » pour le corps. Il aurait été possible de prendre la même vue que celle des chapitres mais j'ai décidé d'en créer une autre pour rendre les liaisons entre les fichiers plus claires.

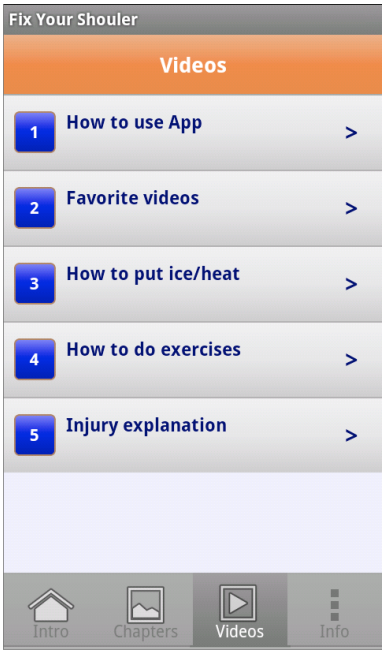


Figure 2.11 - Liste des vidéos

La seconde partie, la lecture de la vidéo sélectionnée, est très différente de ce que j'ai pu développer précédemment. En effet, celle-ci nécessite un lecteur vidéo. N'étant pas un expert du développement Android, je n'ai pas développé mon propre lecteur mais utilisé celui qu'Android propose. Voici le code important de l'activité :

Algorithme 2.10 - Intégration d'une vidéo

```
VideoView videoview;
Uri chemin;
videoview = (VideoView)findViewById(R.id.videoView);

MediaController mediaController = new MediaController(this);
mediaController.setAnchorView(videoview);
videoview.setMediaController(mediaController);

String[] videos = getResources().getStringArray(R.array.video_list);
String[] lien = getResources().getStringArray(R.array.video_lien);

chemin = Uri.parse(lien[video]);

videoview.setVideoURI(chemin);
```

Ici, j'ai créé une « VideoView » que j'associe à l'élément « VideoView » de la vue qui affichera la vidéo. Je l'associe ensuite à un « MediaController » qui permet de gérer un média, que ce soit un son ou une vidéo. L'instruction « chemin = Uri.parse(lien[video]) » permet de récupérer le lien de la vidéo qui est sur le serveur car les vidéos ne sont pas stockées dans l'application. On transmet ensuite le lien de la vidéo au VideoView.

Voici le code nécessaire :

Algorithme 2.11 - Vue affichant une vidéo

```
<VideoView
      android:id="@+id/videoView"
      android:layout_width="fill_parent"
      android:layout_height="fill_parent">
```

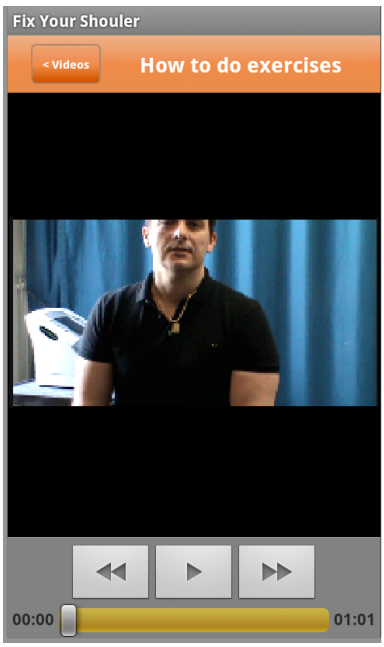


Figure 2.12 - Lecture d'une vidéo

2.6.2.4 Les informations

La section « informations » est un peu différente des autres sections car elle ne contient pas qu'un seul type d'information. En effet, on y retrouve à la fois des textes concernant l'auteur, les copyrights mais aussi comment contacter l'auteur ou partager l'application.

La première activité présente la liste des informations que l'utilisateur peut consulter. Son code est similaire aux autres activités présentant des listes.

Les trois premiers liens envoient l'utilisateur sur une consultation de texte, présentation de l'auteur, copyrights et remerciements. Le lien suivant nous affiche une page permettant de contacter l'auteur via son site internet, son Facebook, son Tweeter ou son adresse mail. Les deux derniers liens ouvrent la boite mail du téléphone pour contacter l'auteur ou pour partager l'application avec un ami. Je ne présenterais pas les trois premiers liens car le code a déjà été vu dans les précédentes activités et vues.

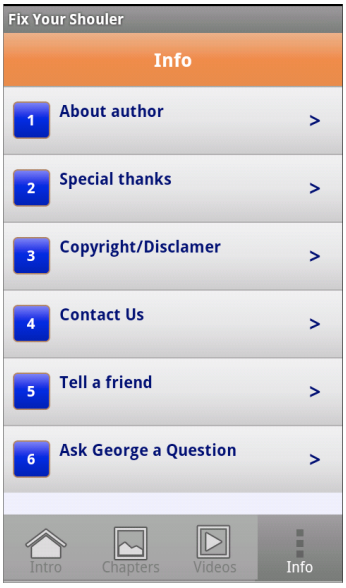


Figure 2.13 - Liste des informations

Figure 2.14 - Page présentant l'auteur

Pour la page contact, celle-ci se compose de textes et de quatre boutons, ouvrant soit le site internet, soit la page Facebook ou la page Tweeter de l'auteur, ou encore l'application mail du téléphone pour lui écrire. Voici le code de l'ouverture d'une page web après l'appui sur un bouton :

Algorithme 2.12 - Ouverture d'une page Web dans une nouvelle activité

```
String url = "http://www.site_web.com";

Intent i = new Intent(Intent.ACTION_VIEW);
Uri u = Uri.parse(url);
i.setData(u);
startActivity(i);
```

Un élément de type « Intent » est créé, auquel on passe en paramètre l'adresse url du site que l'on veut ouvrir. Ensuite, on lance une activité virtuelle, qui ouvre le site internet dans une nouvelle page.

Figure 2.15 - Affiche site internet

En ce qui concerne l'envoi d'un mail, le code est le même pour cette page comme pour les deux derniers liens de la liste des informations, seule l'adresse de la personne à contacter change.

Algorithme 2.13 - Envoie d'un mail avec Android

```
final Intent emailIntent = new
Intent(android.content.Intent.ACTION_SEND);
emailIntent.setType("plain/text");
emailIntent.putExtra(android.content.Intent.EXTRA_EMAIL, new String[]{
"adresse@mail.com"});
emailIntent.putExtra(android.content.Intent.EXTRA_SUBJECT,  "Quection :
Fix your shoulder app");
emailIntent.putExtra(android.content.Intent.EXTRA_TEXT,  "La question
..\n\n Sent from Android");
startActivity(Intent.createChooser(emailIntent, "Send  mail..."));
```

Ici, un nouvel « Intent » est créé, correspondant à l'Intent Android d'envoi de mail. Certaines informations lui sont transmises, comme l'adresse du correspondant, le sujet du mail ainsi que le texte de message. Pour finir on ouvre une activité virtuelle qui permet d'ajouter ou non des données avant d'envoyer le mail.

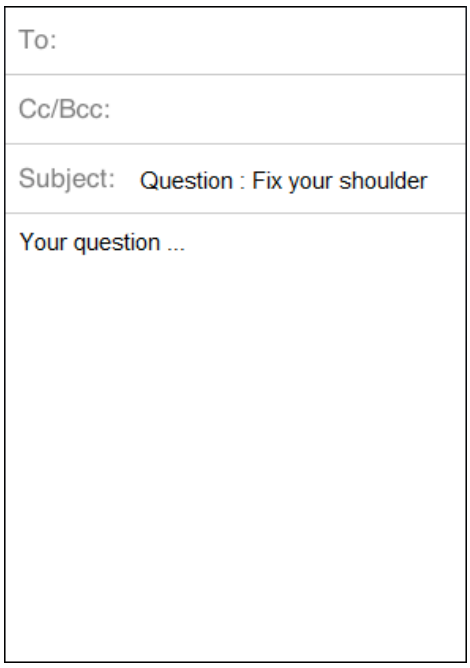


Figure 2.16 - Envoi d'un mail

2.6.3 Transitions

Dans l'application, il y a deux types de transitions : celles qui se font grâce aux onglets et permettent de changer de catégories et celles qui permettent de passer d'un page à une autre, comme d'un chapitre à ses sous chapitre et le retour à la liste des chapitres.

2.6.3.1 Onglets

L'activité qui va gérer les onglets est la base de la structure de l'application. C'est elle qui permet de passer d'une catégorie à une autre et afficher le contenu de celles-ci. Pour mieux comprendre comment cela fonctionne, voici le code de l'activité :

Algorithme 2.14 - Création d'un onglet

```
TabHost tabHost = getTabHost();
TabHost.TabSpec spec;
Intent intent;

//intro

intent = new Intent().setClass(this, Intro.class);
spec =
tabHost.newTabSpec("Intro").setIndicator("Intro",getResources().getDrawabl
e(R.drawable.ic_menu_home)).setContent(intent);
tabHost.addTab(spec);

//fin intro
```

Ici, le code concerne le premier onglet d'introduction, c'est le même pour les trois autres onglets, seuls le titre et l'image passée en paramètre change. Nous allons commencer par déclarer un « TabHost » et un « TabSpec ». Le « TabHost » servira de lien entre l'IHM et notre code. De plus, il sera le conteneur des différents onglets, quant au « TabSpec » il sera là pour spécifier les informations de l'onglet (l'activité vers laquelle il pointe, son nom et son image). L'Intent est là pour signaler l'activité de départ et celle qui doit être affichée si l'on sélectionne cet onglet. Le code de la vue liée à cette activité est celui-ci :

Algorithme 2.15 - Vue gérant les onglets

```
<FrameLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:id="@android:id/tabcontent"
```

```
        android:layout_weight="1" >
</FrameLayout>

<TabWidget
        android:id="@android:id/tabs"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_weight="0" >
</TabWidget>
```

Le « FrameLayout » va permettre d'afficher le contenu des pages telles que l'introduction, les chapitres, etc. Le « TabWidget » va quant à lui gérer les onglets et leurs états (en cours ou en arrière).

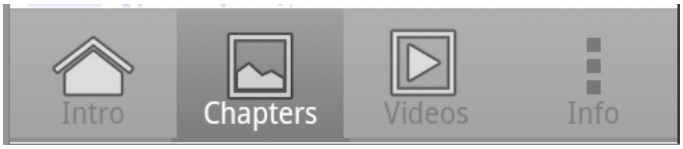


Figure 2.17 – Onglets

2.6.3.2 Page à page

Concernant les transitions page à page, cela se passe entièrement dans les activités. En effet, des données vont être transmises entre les activités et vont rendre possible la mise en place de bouton « retour » permettant de revenir facilement sur la page précédente sans revenir au début de chaque catégorie. Par exemple, lorsque je consulte le texte d'un sous chapitre et que je veux consulter le suivant, je n'ai pas besoin de revenir à la liste des chapitres, je peux tout de suite revenir sur la liste des sous chapitres que je consultais sans repréciser le sous chapitre.

Algorithme 2.16 - Envoie de données à une autre activité

```
Intent chap = null;

chap = new Intent(this, Destination.class);
chap.putExtra("chapitre",1);
chap.putExtra("onglet",1);
startActivity(chap);
```

Pour transmettre des données, on les ajoute à l'Intent qui va afficher la nouvelle activité. L'Intent va contenir l'activité présente et celle que l'on veut afficher. L'instruction « putExtra » va lier les données à l'Intent. Chaque donnée possède un nom et une valeur.

Algorithme 2.17 - Récupération des données transmises

```
Bundle extra = getIntent().getExtras();
if(extra != null)
{
      chap_numero = extra.getInt("chapitre ");
      onglet = extra.getInt("onglet ");
}
```

Le « Bundle » va récupérer les informations contenues dans l'Intent qui a permis d'afficher l'activité en cours. Si celui-ci n'est pas « null », on récupère les valeurs des données transmises pour les utiliser.

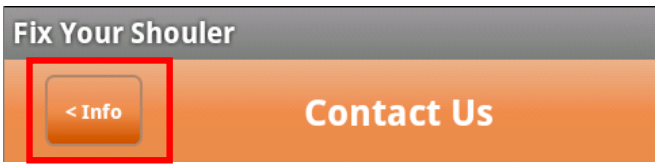


Figure 2.18 - Bouton retour

2.6.4 Gestion textes

Les textes présents dans l'application sont stockés dans des fichiers XML. Ici, j'ai décidé de tout mettre dans le fichier String.xml car l'accès aux ressources se fait facilement et cela permet de créer des tableaux directement dans le fichier XML, ce qui est un avantage pour générer les listes par la suite.

Les textes se présentent de cette manière, entre des balises string avec un identifiant, utilisé dans les activités ou les vues.

Algorithme 2.18 - Extrait du fichier String.xml

```
<string name="app_name">Shoulder</string>
<string name="title_activity_main">Fix Your Shouler</string>


<string-array name="chap_titre_list">
   <item name="titre_chap_1">"The Nutes And Bolts Of Shoulder"</item>
   <item name="titre_chap_2">"No More Scary Horror Shoulder Storie"</item>
   <item name="titre_chap_3">"Your Posture Is Killing You !!"</item>
   <item name="titre_chap_4">"Giving Your Pain The Cold Shoulder"</item>
   …
</string-array>
```

« String-array » permet ici de créer des tableaux de string. Dans cet exemple, on retrouve un extrait de la liste des titres des chapitres. En appelant ce « string-array », le tableau est tout de suite créé, ce qui simplifie le code de lecture de fichiers XML.

2.6.5 Mise à jour

La mise à jour des textes de l'application est l'élément liant le site Web à l'application. En effet, les textes seront mis à jours par le client via le site Web et l'application effectuera des tests pour contrôler que la version actuelle des textes est bien la dernière. Cette vérification se

déroule au lancement de l'application, avant que l'utilisateur puisse consulter l'introduction ou encore les chapitres.

L'ensemble du code se situe dans la première lancé car si une mise à jour doit être effectuée, elle doit se faire avant que l'utilisateur puisse consulter l'application.

Algorithme 2.19 - Vérification pour la mise à jour

```
XMLParser parser = new XMLParser();
String xml = parser.getXmlFromUrl(URL);
Document doc = parser.getDomElement(xml);
NodeList nl = doc.getElementsByTagName("textes");

Element e = (Element) nl.item(0);
Nouvelle_version = parser.getValue(e, "version");

version = getResources().getString(R.string.version);

if(Nouvelle_version.equals(version))
      "mise à jour pas nécessaire !";
else
      "mise à jour nécessaire !";
```

Pour commencer, je récupère le fichier XML présent sur le serveur que je parse pour récupérer les informations essentielles telles que la version des textes. Ensuite je compare cette version avec la version actuelle de l'application. Si elles sont différentes, j'effectue la mise à jour des textes.

Pour effectuer cette mise à jour, je convertis les XML récupéré du serveur en « InputStream » qui va me permettre de lire l'intégralité du fichier, y compris les balises XML.

Algorithme 2.20 - Récupération des données du serveur

```
InputStream in = uc.getInputStream();

if (in!=null)
{

	InputStreamReader tmp = new InputStreamReader(in);
	BufferedReader reader = new BufferedReader(tmp);
	String str;
	StringBuffer buf = new StringBuffer();

	while ((str = reader.readLine()) != null)
	{
		buf.append(str + "\n");
	}

	in.close();

}
```

Les nouvelles versions des textes sont dans la variable String str. Pour la modification du fichier actuel de l'application, je convertis celui-ci en « OutputStream » pour pouvoir l'éditer.

Algorithme 2.21 - Mise à jour des données

```
OutputStreamWriter out = new
OutputStreamWriter(openFileOutput("fichier.xml", 0));
out.write(str );
out.close();
```

2.6.6 Tests

Les tests concernant l'application ont tous été réalisés grâce à des téléphones Android et avec l'émulateur que propose Eclipse. Cela m'a permis de contrôler le fonctionnement de l'application sur différentes plateformes et d'être sûr que tous les utilisateurs qui allaient utiliser l'application soient satisfaits et non gênés par certains bugs dûs aux différentes tailles d'écran par exemple.

CHAPITRE 3

Développements Web

Le seconde partie du projet a pour but de développer un site Web qui permettra de mettre à jour l'application de façon rapide et facile sans avoir besoin de passer par le Play Store ou encore l'Apple Store pour la version IOS. Dans un premier temps, seuls les textes pourront être mis à jour mais des évolutions sont possibles.

Les différentes étapes qui ont menés au développement du site sont les même que pour l'application : l'analyse des besoins et identification des contraintes, le choix de la méthodologie adoptée, la définition des normes à respecter, la mise en place de la conception et la réalisation de celle-ci avec l'implémentation.

3.1 Analyse

Ce site doit répondre au principal besoin de pouvoir mettre à jour les textes de l'application de manière rapide et efficace. En effet, à chaque modification de l'application, celle-ci doit être transmise aux différentes plateformes de partage telles que le Play Store d'Android ou l'Apple Store d'Apple. L'application doit ensuite être validée car elle doit respecter un certains nombres de critères, ce qui peut prendre du temps. Sachant qu'ici nous ne modifions que les textes, ce qui représente une modification mineure, il est plus facile de mettre à jour les fichiers sur un serveur et laisser les applications se mettre à jours d'elles même grâce à une simple connexion au serveur dès que l'utilisateur lance l'application.

Pour commencer, ce site doit posséder une page de connexion pour que seules les personnes ayant le droit de modifier les textes puissent y avoir accès. Une fois connecté, l'utilisateur du site arrive sur une page présentant l'application et peut naviguer sur celle-ci comme s'il utilisait son téléphone. La différence est qu'ici il pourra modifier les informations présentes.

Voici une vue d'ensemble du site, qui est similaire à celle de l'application :

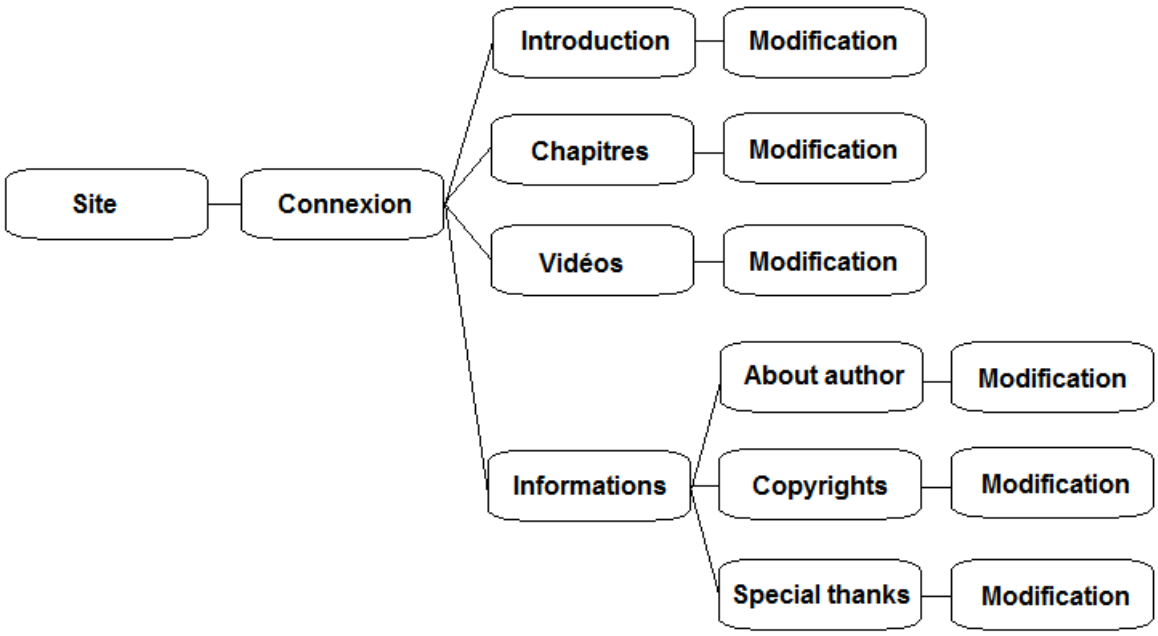


Figure 3.1 - Vue d'ensemble du site

3.2 Exigences et contraintes

3.2.1 Exigences fonctionnelles

Comme pour l'application, le site doit respecter certaines exigences telles qu'une navigation intuitive, facile et rapide entre les différentes sections et entre le contenu de chacune d'entre elle. Mais aussi la possibilité de modifier chaque texte disponible dans l'application en tout temps en étant connecté à internet. L'utilisateur du site devra forcément s'identifier pour accéder aux informations sur le site et ainsi les modifier.

3.2.2 Exigences non fonctionnelles

Quant aux exigences non fonctionnelles, celles-ci concernent la convivialité, la maintenabilité et la compatibilité du site.

- Conviviale : le site doit être compréhensible par les personnes qui vont l'utiliser. Celle-ci doit permettre de trouver facilement et rapidement ce que l'on cherche pour ne pas perdre de temps.

- Accessible : L'utilisateur doit pouvoir accéder au site à tout moment peu importe l'endroit où il se trouve tant qu'il est connecté à internet.

- Compatible : Le site va principalement être développé pour Firefox et Google Chrome.

3.2.3 Contraintes

Les principales contraintes que je devais respecter lors de la réalisation du site étaient celles concernant les modifications liées à l'application. En effet, les textes présents sur le site doivent être les mêmes que sur l'application et tous les textes doivent être modifiables. En aucun cas le design de l'application ou l'ajout de fonction ne doit être possible.

Les autres contraintes à respecter concernent les normes des langages utilisés pour le développement du site que je détaillerais plus tard dans le rapport.

3.3 Méthodologie

Pour développer ce site Web, j'ai décidé de le faire entièrement en ligne de code sans utiliser de CMS (Content Management System, logiciels destinés à la conception et à la mise à jour dynamique de sites Web) car ce n'est pas un site compliqué et il m'est plus facile de réaliser les objectifs de cette façon.

3.3.1 Procédure de travail

L'ensemble de cette seconde partie de projet, qu'est le développement du site Web permettant la mise à jour des textes composant l'application, s'est déroulé de la même façon que le développement de l'application Android. En effet, la première partie consistait en l'étude de la technologie et langages utilisés même si dans ce cas je connaissais ces langages. Les étapes suivantes étaient l'analyse des besoins du client, la conception, l'implémentation et pour finir les tests pour vérifier que tout fonctionne correctement.

L'ensemble de cette partie m'a permis de développer un site Web dans sa totalité de la création des différentes pages à la mise en ligne.

3.3.2 Conditions nécessaires au développement

Pour développer un site web, il suffit d'avoir un ordinateur pouvant se connecter à internet pour la mise en ligne et d'un éditeur de texte, peu importe le système d'exploitation utilisé (Windows, OS X, Linux, etc.).

Certains logiciels existent pour faciliter le développement du site en permettant au système de se comporter comme un serveur. Cela permet de rendre les tests plus rapides sans devoir envoyer les fichiers aux serveurs à chaque modification. On peut citer WAMP pour Windows ou LAMP pour Linux par exemple.

3.4 Standards et normes

Les normes à suivre lorsqu'on développe un site Web vont être le w3c (World Wide Web Consortium) qui concerne le HTML et le CSS au niveau de l'écriture du code, des syntaxes indispensables en début de page, aux écritures de balises ou valeurs d'attributs. Il existe aussi des normes concernant le PHP et le JavaScript qui ne concernent que les écritures des fichiers.

3.5 Conception

3.5.1 Conception du site

Pour répondre aux besoins du client de pouvoir mettre à jour les textes de l'application facilement, j'ai décidé de réaliser un site Web. Ce site se divise en trois parties, l'identification, la consultation de l'existant et la modification.

La page d'identification est la page d'accueil du site et propose simplement à l'utilisateur de saisir un identifiant et un mot de passe pour être sûr que celui qui accèdera aux informations en ait les droits et puisse les modifier. Cette page est un page de sécurité.

La seconde partie présente l'application à l'utilisateur du site. En effet, celui-ci pourra la consulter comme s'il utilisait son téléphone. En haut de l'écran se trouve quatre onglets correspondant aux différentes parties de l'application : l'introduction, les chapitres, les vidéos et les informations. Chaque onglet amène vers un contenu bien précis comme pour l'application mais avec quelques changements. Il n'y a plus de listes pour les chapitres, sous chapitres, vidéos et informations mais des sous onglets. Cela permet d'avoir tout sur le même écran et de pouvoir changer de page plus facilement et rapidement. Pour les vidéos, celles-ci sont toutes présentes sur la même page et consultables instantanément. Chaque page dont le

contenu est modifiable possède un bouton « modifier » nous envoyant sur une autre page ou l'utilisateur peut changer les textes.

La dernière partie, concernant la modification des textes se divise en trois. En haut, le rappel du nom de la page en cours de modification, qui est utile lors d'une modification de sous chapitre. Ensuite, il y a deux colonnes. Celle de gauche affiche le texte actuel avec possibilité de modification (ajout ou suppression de caractères). Chaque texte est écrit sous format HTML simplifié, certaines balises permettant de changer la taille du texte, de revenir à la ligne, de mettre le texte en italique ou en gras. La seconde colonne affiche le texte tel qu'il sera à l'écran du téléphone, avec interprétation des balises qui vont alors disparaitre.

3.5.2 Langages utilisés

3.5.2.1 HTML

L'Hypertext Markup Language (HTML), est le format de données conçu pour représenter les pages web. C'est un langage de balisage permettant d'écrire de l'hypertexte. Il permet également de structurer et de mettre en forme le contenu des pages, d'inclure des ressources multimédias, des formulaires de saisie, et des éléments programmables.

3.5.2.2 CSS

CSS (Cascading Style Sheets : feuilles de style en cascade) est un langage qui sert à décrire la présentation des documents HTML et XML. Les standards définissant CSS sont publiés par le World Wide Web Consortium (W3C).

3.5.2.3 PHP

PHP: Hypertext Preprocessor (PHP), est un langage de scripts libre principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale, en exécutant les programmes en ligne de commande.

3.5.2.4 JavaScript

JavaScript est un langage de programmation de scripts principalement utilisé dans les pages web interactives mais aussi côté serveur.

3.5.3 Outils utilisés

Lors du développement du site Web, j'ai utilisé différents outils tels qu'un éditeur de texte, un émulateur de serveur pour les tests et un autre pour transmettre les fichiers au serveur.

Notepad++ est un éditeur de texte générique codé en C++, qui intègre la coloration syntaxique de code source pour de nombreux langages informatiques, ce logiciel propose la possibilité de créer ses propres colorations syntaxiques pour un langage quelconque. Il est très efficace lorsque l'on veut tout coder manuellement et il propose différents outils simplifiant la réalisation de programmes.

WampServer est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Il permet d'émuler un serveur sur l'ordinateur et simplifie les phases de tests.

FileZilla est un client FTP libre et simple d'utilisation qui permettra aux débutants comme aux utilisateurs confirmés de se connecter à distance sur un serveur afin d'y télécharger des fichiers.

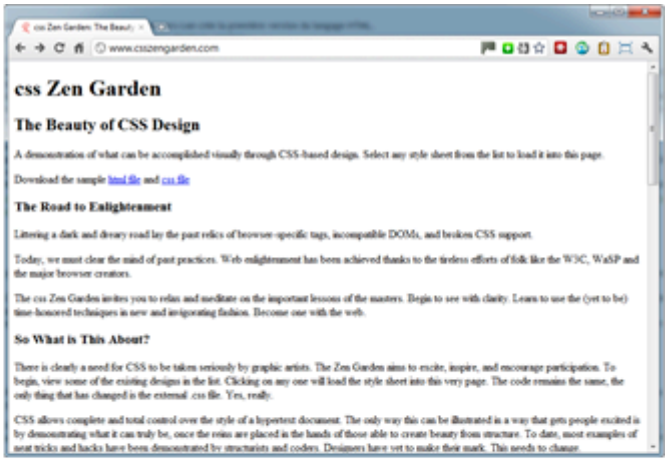
3.6 Implémentation

3.6.1 Implémentation Web

3.6.1.1 Fonctionnement site avec HTML et CSS

Un site Web est composé de pages entre lesquelles l'utilisateur peut naviguer. Chaque page est écrite en ligne de code. Il existe de nombreux langages permettant de réaliser des sites Web mais deux sont obligatoires, HTML et CSS. L'un ne va pas sans l'autre. En effet, HTML va créer tous les éléments composant une page : les textes, les images. CSS va disposer les éléments et créer le design de la page. Cela ressemble aux activités et vues d'Android dont je parlais précédemment.

HTML
(pas de CSS)



HTML + CSS

Figure 3.2 - Présentation HTML et CSS
Tiré du siteduzero.com

3.6.1.2 Apports de PHP et JavaScript

Lorsqu'on ajoute du PHP à un site Web, cela permet de proposer de nouvelles options aux utilisateurs. PHP permet d'utiliser des variables et des fonctions, de récupérer des données ou des actions réalisées par l'utilisateur et de transmettre des données aux pages suivantes. L'utilisation de PHP en tant que générateur de pages Web dynamiques est la plus répandue, mais il peut aussi être utilisé comme langage de programmation ou de script en ligne de commande sans utiliser de serveur HTTP ni de navigateur.

Lorsqu'un visiteur demande à consulter une page Web, son navigateur envoie une requête au serveur HTTP correspondant. Si la page est identifiée comme un script PHP (généralement grâce à l'extension .php), le serveur interprète le PHP et va le traiter pour générer le code final de la page (constitué généralement d'HTML et de CSS). Ce contenu est renvoyé au serveur HTTP, qui l'envoie finalement au client.

Figure 3.3 - Fonctionnement PHP

JavaScript, quant à lui va permettre d'effectuer des actions sans avoir à recharger une page. Cela permet de contrôler des formulaires au moment où l'utilisateur saisie les informations ou à modifier rapidement certains contenus sans changer de page.

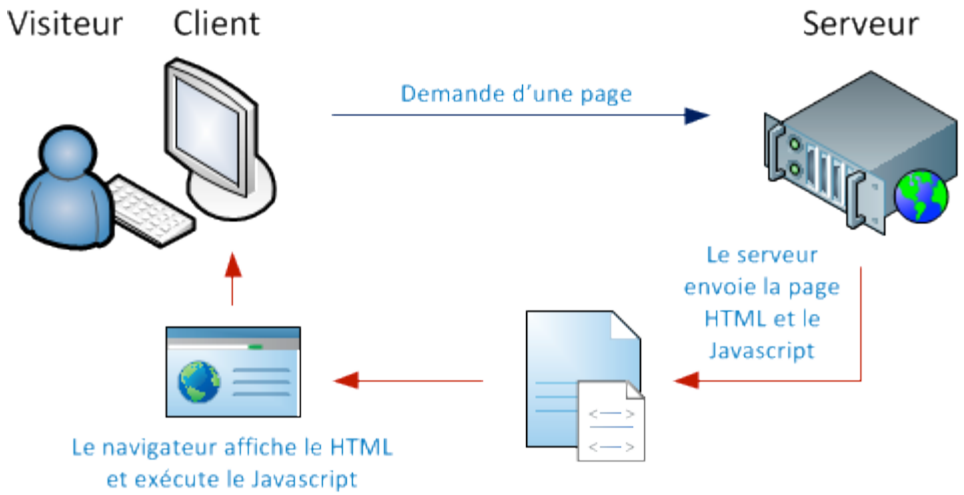


Figure 3.4 - Fonctionnement JavaScript

3.6.2 Développement du site

3.6.2.1 Composition de chaque page

Le site Web est divisé en trois parties : l'identification, la consultation de l'application et la modification des textes.

L'identification est très simple et se compose d'une seule page. Elle présente uniquement un formulaire demandant à l'utilisateur son login et son mot de passe. Cette page a été écrite en HTML, PHP et CSS. Une fois les informations saisies, elles sont transmises à la page suivante. Si elles sont correctes, on affiche le contenue du site (l'ensemble de l'application et la possibilité de modifier les textes) sinon on revient sur la page d'identification avec un message d'erreur.

Voici le code de vérification du login et mot de passe de l'utilisateur :

Algorithme 3.1 - Vérification des identifiants client

```
if(isset($_POST['login']))
{
	$login = htmlspecialchars($_POST['login']);
	$passwd = hash('ripemd160', htmlspecialchars($_POST['passwd']));

	if($login == "login" && $passwd == "mot de passe")
	{
		$_SESSION['id'] = "ok";
		$error = "";
	}
	else
		$error = "Login ou mot de passe incorrect !";
}
```

Ici, je vérifie d'abord si un login à bien été saisi. Si oui je le récupère et je code le mot de passe. Ensuite, je vérifie que les informations saisies sont correctes. Si oui, je créé la

variables de session ‘id’ pour ensuite afficher le reste du site. Sinon je créé un message d’erreur qui sera affiché sur la page d’identification.

La fonction « htmlspecialchars() » sert à contrôler le contenu saisi par l’utilisateur et empêcher que celui-ci saisisse du code HTML ou PHP qui serait interprété et modifierait mon code.

La fonction « hash » quant à elle va permettre de crypter le mot de passe saisi pour vérifier qu’il corresponde avec l’existant.

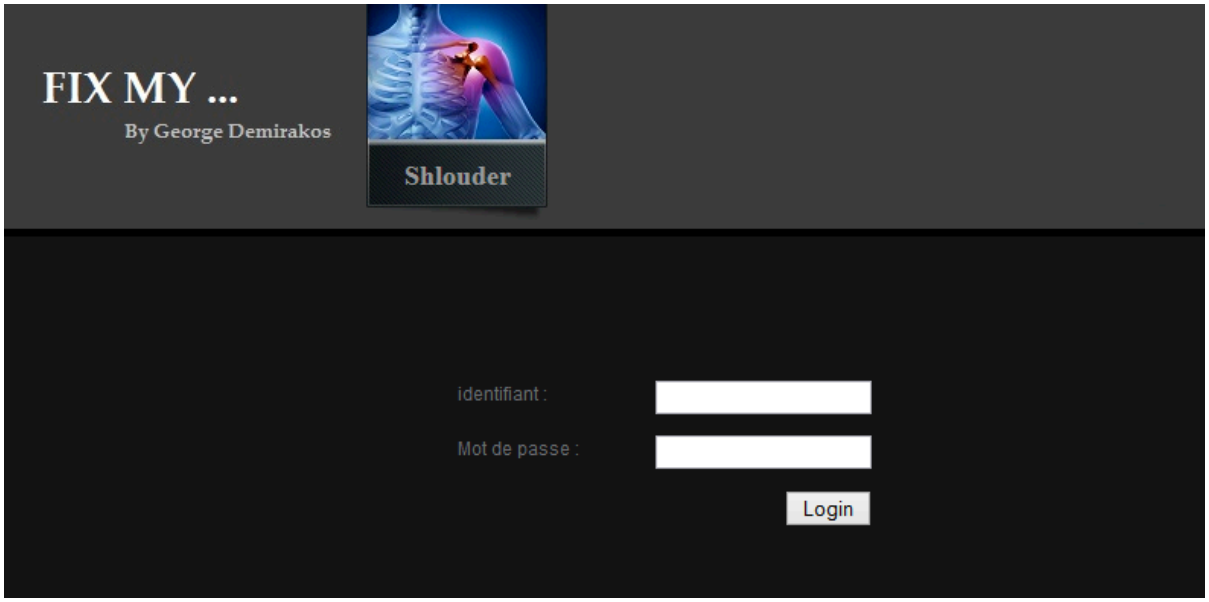


Figure 3.5 - Page d'identification du site Web

La seconde partie du site est aussi composée d’une seule page constituée d’onglets comme l’application mobile. C’est ici que le JavaScript va avoir son importance car c’est grâce à lui que l’on peut naviguer avec des onglets sur une page Web sans changer de page. Je vais détailler le fonctionnement des onglets dans cette partie et je placerais le code correspondant en annexe.

Du coté JavaScript, je vais utiliser une fonction qui prend en paramètre l'identifiant de l'onglet sur lequel l'utilisateur à cliquer. Je vais ensuite contrôler tous les onglets et lorsque je contrôlerai celui qui a été passé en paramètre, je le ferai afficher en changeant son CSS. Les autres onglets seront cachés.

Du coté HTML, je vais créer plusieurs blocs qui correspondront aux onglets avec pour chacun un titre. Chaque onglet est relié à son titre grâce à un identifiant. Le contenu des onglets est généré automatiquement avec les fichiers textes qui composent l'application.

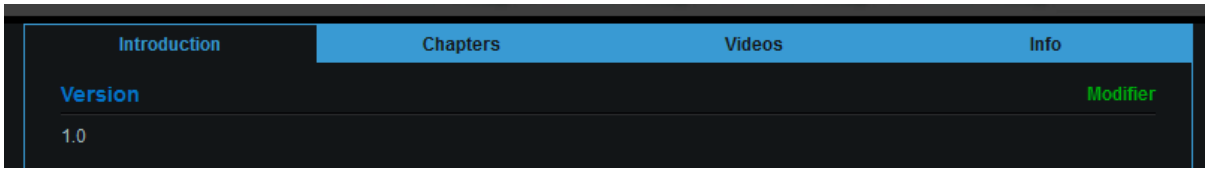


Figure 3.6 - Onglet du site Web

Le contenu de chaque onglet est le même que dans l'application, seul la présentation change. La différence se fait surtout pour les chapitres et les vidéos.

Pour les chapitres, j'ai réutilisé le système d'onglets pour ne pas changer de page et ainsi garder la même idée de départ. J'ai ensuite réutilisé ce principe pour les sous chapitres. Chaque chapitre est composé de sous chapitres, eux même composés de textes. Le problème ici était les identifiants des onglets, car deux onglets avec le même identifiant n'afficheront qu'un seul contenu. Pour corriger cela, j'ai décidé que les onglets principaux auront comme identifiant des numéros de 0 à 9, les chapitres de 10 à 99 et les sous chapitres de 100 à 999.

Voici l'affichage final de l'onglet des chapitres :

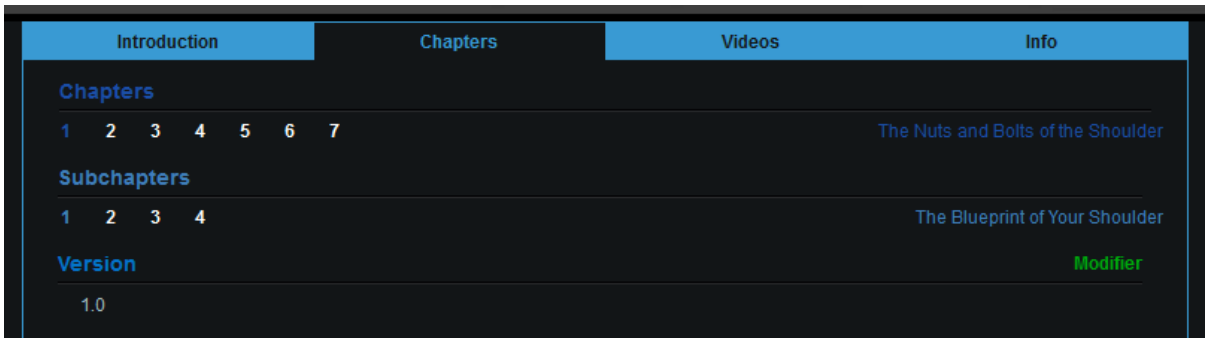


Figure 3.7 - Onglet des chapitres

Pour les vidéos, j'ai simplement utilisé une frame qui va afficher une vidéo passée en paramètre.

Voici le code utilisé :

Algorithme 3.2 - Object affichant une vidéo

```
<object classid="clsid:02BF25D5-8C17-4B23-BC80-D3488ABDDC6B" width="240"
height="150">
      <param name="src" value="video"></param>
      <param name="autoplay" value="false"></param>
      <param name="type" value="video/quicktime"></param>
      <param name="qtnext1" value="goto0"></param>
      <embed src="lien video" width="240" height="150" autoplay="false"
      type="video/quicktime"
      pluginspage="http://www.apple.com/quicktime/download/"
      qtnext1="goto0"></embed>
</object>
```

Ici, je détermine la taille du lecteur ainsi que certains paramètres de lecture et je spécifie qu'elle vidéo doit être lue.

Voici le résultat :

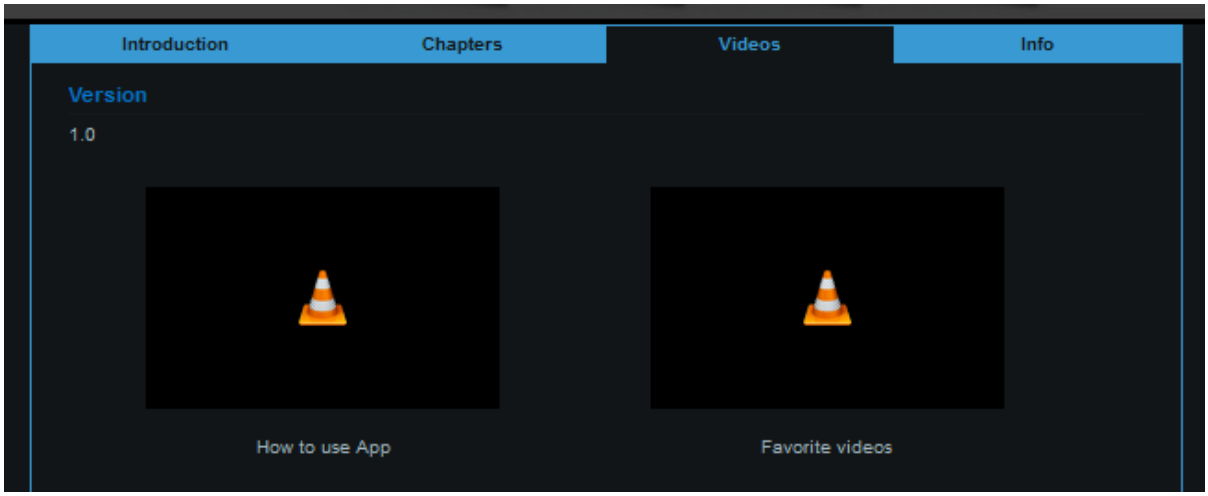


Figure 3.8 - Affiche des vidéos

Pour finir, la partie modification est elle aussi composée d'une seule page. Dans cette partie, je récupère le texte à modifier et je l'affiche dans un espace où il est possible de modifier un texte. A côté de cela, j'ai créé une zone qui est mise à jour à chaque modification du texte grâce à une fonction JavaScript. Dans cette fonction, je récupère le « textarea » qui contient le texte modifiable et lui attribue la fonction « OnChange() » qui prend en paramètre le texte en cours de modification. Ce texte est alors renvoyé dans l'espace prévu pour être afficher au bon format. Une fois les modification terminées, l'utilisateur valide et revient sur la page de consultation de l'application.

Lors d'une modification, le nouveau texte va toujours être contrôlé au cas où des caractères spéciaux soient saisis par l'utilisateur. Dans ce cas, il existe des fonctions PHP qui remplacent ces caractères pour que ceux-ci soient ensuite interprétés correctement par les téléphones mobiles.

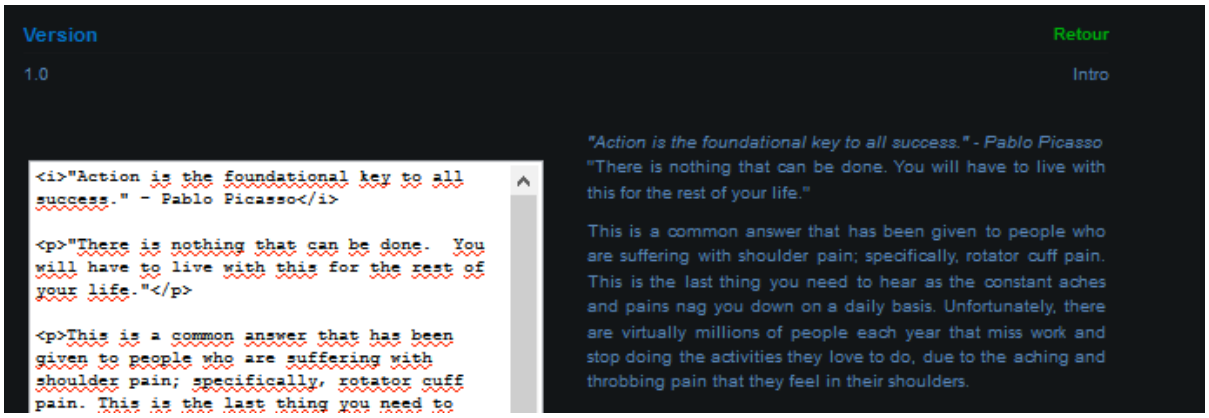


Figure 3.9 - Modification de l'introduction

3.6.2.2 Gestion des fichiers textes

Les fichiers textes qui composent les applications IOS et Android ont pour format le XML. Lors de la consultation des textes des applications, je me base sur les fichiers correspondant à l'application IOS. Pour interagir avec ces fichiers, j'utilise la méthode simpleXML() de PHP, quant au fichier pour Android, j'utilise le DOM.

SimpleXML est une API destinée à simplifier la manipulation des documents XML avec PHP. SimpleXML permet de manipuler un document XML via des fonctions avancées de programmation orientées objet pour atteindre une plus grande simplicité d'utilisation. En effet, SimpleXML convertit chaque élément balisé en objet. Chaque objet peut alors être affiché, modifié ou supprimé.

Voici un exemple du son utilisation :

Un document XML simple :

Algorithme 3.3 - Exemple XML pour SimpleXML()

```
<?xml version='1.0' standalone='yes'?>
<films>
        <film>
                <titre>Le nom de la rose</titre>
                 <duree>127 min</duree>
         </film>
         <film>
                 <titre>Sacré Graal</titre>
                <duree>91 min</duree>
         </film>
         <film>
          <titre>Le livre de la jungle</titre>
          <duree>75 min</duree>
         </film>
</films>
```

En PHP:

Algorithme 3.4 - Code PHP utilisant SimpleXML()

```
<?php

$simpleXml = new SimpleXMLElement($chaineXml);

// écrit "Le nom de la rose"
echo $simpleXml->film[0]->titre;
```

```
// supprime le 3eme film (la numérotation des éléments commence à 0, le troisième élément est donc numéro 2)
unset($simpleXml->film[2]);

// ajoute un film nommé "La liste de Schindler" (197 min)
$nouveauFilm   = $simpleXml->addChild('film');
$nouveauTitre  = $nouveauFilm->addChild('titre', 'La liste de Schindler');
$nouvelleDuree = $nouveauFilm->addChild('duree', '197 min');

// affiche le contenu de notre object simplexml
print_r($simpleXml);

?>
```

Cette méthode est efficace mais ne fonctionne pas avec le fichier Android, c'est pour cela que j'ai utilisé DOM. C'est une interface du standard XML, qui permet une gestion organisée de la structure d'un document XML.

CHAPITRE 4

Résultats

4.1 Présentation des résultats

L'application développée dans le cadre de ce projet est complétement réalisée et respecte tous les besoins du client. On y retrouve le contenu de son livre avec l'introduction et l'ensemble des chapitres, des vidéos permettent d'illustrer les textes et on peut facilement communiquer avec l'auteur. L'application est facile d'utilisation et intuitive. Celle-ci est terminée et validée par le client. Il ne reste plus qu'à la mettre à disposition des futurs utilisateurs grâce au Play Store de Google.

En ce qui concerne le site Web, celui-ci répond également aux besoins du client, qui peut désormais modifier les textes disponibles dans l'application de façon rapide, simple et en tout temps. Le site est en ligne et est opérationnel.

4.2 Discussion

4.2.1 Limites de l'application

On a vu que grâce à Android, beaucoup de choses sont possibles mais toutes ne sont pas compatibles avec l'application. En effet, celle-ci présente un livre illustré de vidéos et l'ajout de fonctionnalités est limité.

Pour le site, les limites concernant les liaisons entre celui-ci et l'application sont moins nombreuses car le Web évolue tous les jours et les possibilités de mises à jour augmentent à chaque fois. Peu de fonctionnalités peuvent être ajoutées mais les façons de faire peuvent apporter beaucoup.

4.2.2 Évolutions possibles

En ce qui concerne les évolutions, celles-ci sont plus nombreuses car même si les fonctionnalités ne changent pas, la façon de présenter les éléments ou de développer les fonctionnalités sont nombreuses et évoluent continuellement. De plus, l'application pourrait contenir des images permettant d'illustrer les textes en même temps qu'on les consulte grâce à des photos ou des schémas. Cela permet de comprendre les mots de l'auteur sans passer par une vidéo, qui nécessiterait de changer de page. Une autre évolution serait de conserver les questions posées à l'auteur et de les présenter dans une partie de l'application pour éviter les questions redondantes. Enfin, l'option de recherche pour trouver rapidement ce que l'on cherche sans parcourir tous les chapitres serait intéressante.

Du côté du site Web, des fonctionnalités pourraient être ajoutées même si le client n'en a pour le moment pas l'utilité telles que l'ajout ou la suppression de chapitres et sous chapitres ou encore l'ajout de photos ou schémas si ceux-ci apparaissent dans l'application. Les autres évolutions concernent le code et les façons de faire les choses. Avec le JavaScript, il est possible de réaliser énormément de choses et de rendre le site encore plus agréable d'utilisation.

CONCLUSION

Dans ce travail, l'objectif était de développer l'application Fix My Shoulder de George Demirakos sur un autre support qu'IOS d'Apple, puis de proposer et mettre en place une solution pour la mise à jour des textes de l'application sans faire intervenir les « marchés » des grandes marques. Les solutions trouvées ont été de développer l'application sur le système Android proposé par Google. Ce système est l'un des principaux concurrents d'Apple et permet donc de diffuser au maximum l'application. Il se base sur du Java qui est un langage de programmation orienté objet qui est très rependu dans de nombreux domaines. Cette application est maintenant développée et prête à être distribuée. Tous les besoins du client ont pu être réalisés. En ce qui concerne la mise à jour des textes, la solution a été de mettre en place un site Web et un serveur pour stocker les textes. Le site Web permettrait à l'utilisateur de modifier les textes d'où il le souhaite, à tout moment. Le site est désormais en ligne et fonctionnel. Le client pourra réaliser toutes les mises à jour nécessaires une fois l'application sur le marché.

Les difficultés rencontrées pendant le travail ont été diverses. Pour commencer, je ne connaissais pas du tout Android et comment réaliser des applications. J'ai du tout apprendre par moi-même et il m'a été difficile de bien maitriser tous les concepts. La seconde difficulté était de lier l'application au serveur pour effectuer les mises à jour. J'ai su faire face à ses soucis grâce à mes connaissances et une bonne gestion de mes acquis durant ma maitrise.

Ce projet m'a permis d'apprendre de nombreux concepts pour le développement mobile et de suivre la création d'une application du début à la fin. Il m'a permis de mettre en place des concepts vus précédemment, comme la gestion de la planification, la création d'application et la réalisation de documentation. Il m'a aussi aidé à voir mes forces et faiblesse et comment mieux les exploiter pour être efficace.

ANNEXE I

Fonction JavaScript pour les onglets

Fonction qui prend en paramètre l'identifiant de l'onglet à afficher (num), une valeur de départ et d'arriver (tabdebut et tabfin) pour les onglets qu'il faut cacher, le texte à mettre à jour (text) et l'id du paragraphe à mettre à jour (idclass).

```
function onglet(num, tabdebut, tabfin, text, idclass)
{
	var arrLinkId = new Array ();
	var debut = tabdebut;
	var fin = tabfin - tabdebut;
	for (var i=0;i<=fin;i++)
	{
		arrLinkId[i] = '_'+debut;
		debut++;
	}
	var arrClassLink = new Array('current','ghost');
	var strContent = new String();

	if(text!='')
		document.getElementById(idclass).innerHTML= text;

	for (i=0 ; i<=fin ; i++)
	{
		strContent = "onglet"+arrLinkId[i];

		if ( arrLinkId[i] == num )
		{
		    document.getElementById(arrLinkId[i]).className =
		    arrClassLink[0];
		    document.getElementById(strContent).className = 'on
		    content';
		}
		else
		{
		    document.getElementById(arrLinkId[i]).className =
		    arrClassLink[1];
		    document.getElementById(strContent).className = 'off
		    content';
		}
	  }


    }
```

ANNEXE II

Code PHP pour les onglets des chapitres

```
<ul id="onglet_chapter">
	<?php

	$num = 0;
	$i = 100;

	foreach ($xmlchapter->array->dict as $value)
	{
		$j = $num + 1;
		echo '<li class="onglet'.$i.'">';

		if($i==100)
			echo '<a href="#" id="_'.$i.'" class="current"
			onclick="onglet(this.id,100,110,\''.$xmlchapter->array-
			>dict[$num]->string[1].'\',\'titre_chap\')"
			>'.$j.'</a>';
		else
			echo '<a href="#" id="_'.$i.'" class="ghost"
			onclick="onglet(this.id,100,110,\''.$xmlchapter->array-
			>dict[$num]->string[1].'\',\'titre_chap\')"
			>'.$j.'</a>';

		echo '</li>';
		$i++;
		$num++;
	}
	?>


</ul>

<ul id="titre_chapter">
	<li>
		<p id='titre_chap'><?php echo $xmlchapter->array->dict[0]-
		>string[1]; ?></p>
	</li>
</ul>

<?php
	$num = 0;
	$i = 100;

	foreach ($xmlchapter->array->dict as $value)
	{
		if($i==100)
			echo '<div id="onglet_'.$i.'" class="on content">';
		else
			echo '<div id="onglet_'.$i.'" class="off content">';
```

```
            ?>
            <div class="header_04">Subchapters</div>
            <div id="container_subchapter">

            <ul id="onglet_subchapter">
                  <?php

                  $num2 = 0;
                  $x = 1000 + $num * 100;
                  $debut = $x;
                  $fin = $x + 30;

                  foreach ($xmlchapter->array->dict[$num]->array->dict as
                  $value)
                  {
                        $j = $num2 + 1;
                        echo '<li class="onglet'.$x.'">';

                        if($num2==0)
                              echo '<a href="#" id="_'.$x.'"
                              class="current"
                              onclick="onglet(this.id,'.$debut.','.$fin.',
                              \''.$xmlchapter->array->dict[$num]->array-
                              >dict[$num2]-
                              >string[0].'\',\'titre_souschap'.$num.'\')"
                              >'.$j.'</a>';
                        else
                              echo '<a href="#" id="_'.$x.'" class="ghost"
                        onclick="onglet(this.id,'.$debut.','.$fin.',\''.$x
                        mlchapter->array->dict[$num]->array->dict[$num2]-
                        >string[0].'\',\'titre_souschap'.$num.'\')"
                        >'.$j.'</a>';

                        echo '</li>';
                        $num2++;
                        $x++;
                  }
                  ?>

      </ul>

      <ul id="titre_souschapter">
            <li>
                  <p id='titre_souschap<?php echo $num; ?>'><?php echo
                  $xmlchapter->array->dict[$num]->array->dict[0]-
                  >string[0]; ?></p>
            </li>
      </ul>




<?php
      $num2 = 0;
```

```
	$x = 1000 + $num * 100;

	foreach ($xmlchapter->array->dict[$num]->array->dict as $value)
	{
		if($num2==0)
			echo '<div id="onglet_'.$x.'" class="on content">';
		else
			echo '<div id="onglet_'.$x.'" class="off content">';

		echo '<div class="header_05">Version <a
		href="modifier.html.php?mod=Chapter&chapter='.$num.'&subchapte
		r='.$num2.'" class="modifier">Modifier</a></div>';

		echo $xmlchapter->attributes()->version;
		echo "<br><br>";
		echo "<p class='sitation'>".$xmlchapter->array->dict[$num]-
		>string[2]."</p>";
		echo "<br>";
		echo $xmlchapter->array->dict[$num]->array->dict[$num2]-
		>string[1];


		echo '</div>';
		$num2++;
		$x++;
	}
?>

	</div>
	<?php

	echo '</div>';
	$num++;
	$i++;
}
?>
```

LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES

Developpez.com. Tutoriel Android. 2000. « Cours sur la programmation, les applications sous Android ». In *Articles, tutoriels et cours pour la programmation sur système Android, le développement d'applications mobiles.* En ligne. <http://android.developpez.com/cours/>. Consulté le 2 octobre 2012.

Google. Développement Android. 2009. « Android Developers». In *Le site de développement Android de Google.* En ligne. <http://developer.android.com/>. Consulté le 25 septembre 2012.

Le site du zéro. Tutoriel Android. 1999. « Créez des applications pour Android ». In *Le site du zéro.* En ligne. <http://www.siteduzero.com/informatique/tutoriels/creez-des-applications-pour-android/>. Consulté le 20 septembre 2012.

Le site du zéro. Tutoriel JavaScript. 1999. « Dynamisez vos sites web avec Javascript ». In *Le site du zéro.* En ligne. <http://www.siteduzero.com/informatique/tutoriels/dynamisez-vos-sites-web-avec-javascript/>. Consulté le 10 janvier 2013

Le site du zéro. Tutoriel PHP. 1999. « Concevez votre site web avec PHP et MySQL ». In *Le site du zéro.* En ligne. <http://www.siteduzero.com/informatique/tutoriels/concevez-votre-site-web-avec-php-et-mysql/>. Consulté le 20 Décembre 2012.

Meier, Reto. 2010. *Développement d'applications professionnelles avec Android 2,* Paris : Pearson Education France, 660 p.

Murphy, Mark. 2010. *L'Art du développement Android,* 2ème éd. Paris : Pearson Education France, 449 p.

.

PHP. PHP development. 2001. « PHP: Hypertext Preprocessor ». In *Le site du PHP*. En ligne. <www.php.net/>. Consulté le 15 janvier 2013.

Van Lancker Luc. 2008. *JavaScript : Introduction et notions fondamentales.* Eni Eds. 200 p.

Welling, Luke et Thomson Laura. 2009. *PHP et MySQL,* 4ème éd. Indianapolis : Sams Publishing, 893 p.