



Amélioration de l'accueil d'enfants hospitalisés

en collaboration avec le centre Marc Sautelet

par

Edmur LOPES
Robin CAVALIERI

Département Informatique - Microélectronique - Automatique

Encadrants internes : Alexandre BOE - Xavier REDON - Thomas VANTROYS

Encadrants externes : Dahim MOUCHOU - Margaux AVILA

Remerciements

Nous tenons à remercier toutes les personnes ayant participé de près ou de loin à l'élaboration de notre projet et plus particulièrement :

Monsieur MOUCHOU, éducateur spécialisé au centre Marc Sautelet, Madame AVILA, kinésithérapeute au centre Marc Sautelet, pour nous avoir reçu et guidé dans notre projet en formulant leurs besoins, et pour avoir mis leur temps à notre disposition.

Monsieur BOE, Monsieur REDON, Monsieur VANTROYS, encadrants de notre projet à Polytech Lille, pour leurs conseils et remarques concernant notre travail.

Sommaire

I) Contexte	5
II) Structure du de l'application	6
IV) Les fiches XML	7
III) Le code QR	9
V) Google Maps	10
VI) Mise à jour des fichiers	11
VIII) Conclusion	15
Annexes	16

I) Contexte

Le SSR Marc Sautelet est un centre de rééducation proposant différents types de soins comme l'orthophonie, l'ergothérapie, la kinésithérapie, l'orthoptie et bien d'autres. C'est par cette polyvalence que le centre accueille des enfants venus de toute la région pour suivre les différents programmes rééducatifs adaptés à leurs pathologies.

Ces enfants, âgés de 2 à 18 ans sont parfois amenés à rester au centre pour plusieurs jours. Afin qu'ils puissent découvrir au mieux l'espace qui les entoure ainsi que les professionnels qu'ils seront amenés à rencontrer au cours de leur période de convalescence, nous souhaitons mettre à leur disposition une tablette.

Cette tablette sera équipée d'une application via laquelle ils pourront partir à la rencontre de l'équipe éducative, découvrir les différentes professions avec des explications adaptées, avoir des informations plus générales sur le centre mais aussi jouer à des mini-jeux sélectionnés en fonction de leur âge.

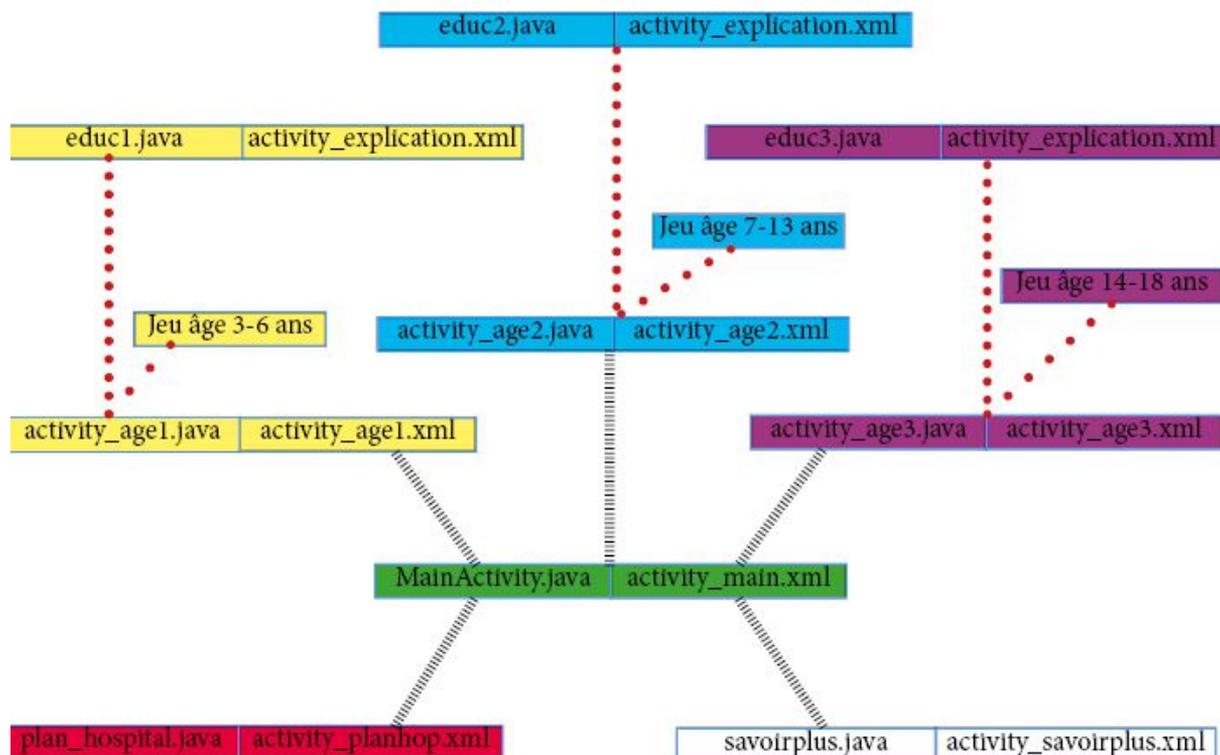
Le but de cette manoeuvre est qu'ils puissent appréhender leur séjour au centre dans une optique de découverte. De plus le format est attractif puisque les tablettes sont aujourd'hui omniprésentes dans la vie des enfants et adolescents.

II) Structure de de l'application

L'application est composée d'un menu à partir duquel sont accessibles :

- Un mode pour les 3 - 6 ans
- Un mode pour les 7 - 13 ans
- Un mode pour les 14 - 18 ans
- Un mode en savoir plus
- Un plan de l'hôpital avec organigramme

Voici l'arborescence de notre application :



Pour chaque couple, vous trouverez à gauche les “.java” et à droite les fichiers “.xml” correspondant à l’affichage lié à son fichier java. Un fichier java est lié à un fichier xml par le biais d’un `setContentView` qui va prendre en argument l’identifiant de la page à mettre en relation.

“`activity_explication.xml`” est commun à plusieurs fichiers java. Ce XML contient un `WebView` qui permet d’afficher les différentes pages HTML créées pour expliquer les différents métiers mais aussi pour l’organigramme du centre avec le plan.

Cela nous a permis d’économiser en mémoire et ainsi éviter de créer un fichier java additionnel pour chaque catégorie d’âge. L’économie de mémoire faisait également partie du challenge à relever.

Ainsi, par cette architecture, l'application ne pèse que 12 Mo. Au début de notre projet, nous souhaitions réaliser un fichier java et un fichier XML pour chacune des fonctionnalités. Cependant, nous nous sommes rapidement rendu compte que l'application prenait trop de place en mémoire (26 Mo pour la première version).

IV) Les fiches XML

Dans chaque fiche XML, nous travaillons sur l'affichage de notre application. Le code XML ressemble beaucoup au code HTML avec balises, boutons, placements dans l'espace...

a) Les "Layout"

Celui-ci est très modulable et nous permet d'ajuster nos Layout facilement avec des LinearLayout, AbsoluteLayout, RelativeLayout.

Le LinearLayout : spécification des objets les uns par rapport aux autres

L'AbsoluteLayout : disposition sous forme de cellules modulables

Le RelativeLayout : spécification des objets horizontalement ou verticalement

Pour ne citer que ces trois exemples-ci car ce sont les seuls dont nous avons eu l'utilité.

b) Les "View"

Les fichiers XML permettent aussi d'intégrer des vues comme les Webview ou les ScrollView.

Le Webview : affichage d'une page Web

Le ScrollView : permet de "scroller" le contenu d'une vue

Ainsi tout le contenu que l'on souhaite mettre dans une vue doit être encapsulé dans celle-ci par les balises adéquates.

La Webview a été utilisée dans notre cas pour l'affichage du Google Maps ou des fiches HTML. Pour pousser un fichier sur une WebView, nous avons utilisé la méthode `getSetting()`.

```
WebView fiche educ = (WebView) findViewById(R.id.fiche educ);  
WebSettings webSetting = fiche educ.getSettings();
```

Ici, nous instançons une Webview “fiche_educ” et la lions à celle du XML avec le même principe que la liaison faite entre un fichier “java” et un fichier “xml”. Ensuite, nous appliquons la méthode “getSettings()” à “fiche_educ” qui ici prend une URL de type :

```
"file:///sdcard/SSR_docs/fiche_medecin1.html"
```

Ainsi, le chemin du fichier est connu. Le fichier HTML sera affiché dans la WebView “fiche_educ”.

c) Les boutons

Les boutons sont omniprésents dans notre application. Du menu au plan de l’hôpital, nous les utilisons comme interactions.

Nous avons utilisé deux méthodes de détection d’un appui :

- soit par une fonction “OnClick” renvoyée par le fichier XML et définie dans le “java” :

```
ImageButton btnScan = (ImageButton)  
findViewById(R.id.mode_aventure);
```

- soit par instantiation de bouton dans le “java” qui récupère l’id de l’objet sur lequel on applique la méthode “setOnClickListener” :

```
btnScan.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        IntentIntegrator integrator = new IntentIntegrator(activity);  
        integrator.initiateScan();  
    }  
})
```

d) Adaptation des “Layout”

Les “Layout” doivent être adaptés à un certains nombre d’écrans comme les :

- small
- normal
- large
- xlarge

Pour cela, le SDK de Android studio nous permet de créer des fichiers “values” dans ces différentes catégories dans lesquelles nous pouvons définir les tailles de chaque paramètre pour chaque type d’écran.

```
<resources>
    <dimen name="txtmenu">15sp</dimen>
    <dimen name="hauteurimage">0dp</dimen>
</resources>
```

Le "txtmenu" est ensuite utilisé pour définir la taille des boutons dans le fichier "activity_main.xml" :

```
android:textSize="@dimen/txtmenu"
```

Ensuite, lors de l'upload de l'application sur l'appareil, la reconnaissance du type d'écran se fera automatiquement.

III) Le code QR

Le centre Marc Sautélet souhaitait qu'une des fonctionnalités de l'application permette aux patients d'obtenir des informations dans le centre tout en découvrant les différents pôles d'activités. Ainsi lors de notre première réunion et suite à une visite, nous avons choisi d'utiliser les codes QR.

Chaque code QR correspond à une chaîne de caractère comme "médecin", "orthoptiste", "infirmier"... En plaçant ces différentes balises dans le complexe, cela permettra aux patients, s'ils flashent par exemple de code "médecin" d'ouvrir dans l'application une page XML avec un webview et d'afficher la page HTML métier spécialement créée pour la catégorie d'âge dans laquelle il se trouve.

Concernant la partie technique de la lecture de code QR, il existe une librairie java nommée Zxing mettant à notre disposition de nombreuses méthodes de traitement.

Lancement d'un scan :

```
IntentIntegrator integrator = new IntentIntegrator(activity);
integrator.initiateScan();
```

Récupération du code scanné sous forme de chaîne de caractères :

```
i.putExtra("result_flash", result.getContents());
```

A partir de la chaîne de caractères récupérée, nous passons d'un fichier java à un autre par le biais de putExtra. Nous traitons les fiches dans educ1.java pour la première tranche d'âge, educ2.java pour la seconde tranche d'âge et educ3.java pour la dernière. Nous avons donc des explications adaptées à chacun avec une méthode d'apprentissage qui permet aux patients de découvrir dans le centre.

Par exemple si nous flashons la le code “médecin” dans la catégorie 3-6 ans :

```
if (result flash.equals("medecin"))
{
fiche educ.loadUrl("file:///sdcard/SSR docs/fiche medecin1.html");
}
```

Nous avons créée une fiche pour chaque corps de métier. Cependant, si un nouveau métier souhaite être rajouté, il faudra ajouter une méthode de détection du code QR de la profession dans le code source de l’application.

V) Google Maps

Les informations au sujet du centre SSR Marc Sautelet dans la partie “en savoir plus”, sont placées à l’intérieur d’un Google Maps inclu dans un Fragment. Sur le Google Maps, nous avons placé sur les coordonnées du centre un “Marker” (*cliquable*) avec la description du centre, qui apparaît lorsque l’on clique dessus. En ce qui concerne le WebView on peut y trouver les coordonnées du centre ainsi que les crédits de l’application.

Concernant la réalisation du Google Maps, nous avons utilisé le “*Google Maps JavaScript API*”. L’avantage de cette méthode est de ne pas avoir besoin d’une clef google pour réaliser un Google Maps; en comparaison avec la méthode du MapFragment qui requiert une clef. Un des avantages pour notre projet est de pouvoir utiliser des codes HTML pour réaliser ce Maps, vu que l’affichage de presque toutes les données se fait sous forme de Webview via les fiches HTML.

Quelques méthodes utilisées pour aboutir au résultat:

```
var infowindow = new google.maps.InfoWindow({
  content: contentString
});
```

ajout d’une fenêtre d’information sur le Maps

```
var marker = new google.maps.Marker({
  position: SSRCentre,
  map: map,
});
```

placement d’un Marker sur les coordonnées du centre

```
marker.addListener('click', function() {
  infowindow.open(map, marker);
});
```

rendre le Marker cliquable

```
google.maps.event.addDomListener(window, 'load', initMap);
```

lancement du Maps avec tous les paramètres définis

VI) Mise à jour des fichiers

Nous souhaitons mettre à jour les fichiers HTML et les images utilisées dans l'application afin de faciliter les modifications de celles-ci par l'équipe du centre Marc Sautelet. Pour cela, nous avons choisi de déposer sur un serveur Web les fichiers à télécharger et de les rendre modifiables par notre application Web.

a) Le serveur Web

L'objectif principal de la création d'un serveur web est la modification des fiches html autour de l'application. En effet les employés du centre vont pouvoir modifier les fiches, par exemple lorsqu'ils voudront y ajouter une information, remplacer une photo ancienne, rajouter du ou en effacer.

Afin de réaliser ces opérations nous avons développé un espace web où les utilisateurs vont pouvoir visualiser les fiches et les images qui se trouvent sur le serveur. Ils auront la possibilité de cliquer sur le nom d'une fiche afin de regarder son contenu ou de cliquer sur l'option "éditer" afin de travailler sur cette même fiche (l'édition est faite sur un "textarea").

Comment fonctionne l'édition ? Nous affichons le contenu du fichier sur le "textarea", ensuite nous pouvons effacer ou ajouter des informations. Une fois l'édition fini nous créons un nouveau fichier avec le même nom que celui du fichier ancien, cette opération écrasera l'ancien fichier donnant naissance au nouveau.

Par question de sécurité, afin que personne d'autre ne puisse modifier les fiches (sauf utilisateurs référencés), nous avons mis en place une base de données avec un mot de passe crypté (SHA1) contenant les comptes de personnes qui auront l'accès au données qui se trouvent sur serveur. A cet effet seulement ces personnes auront la permission de modifier, effacer ou téléverser les données sur le serveur. L'administrateur peut ouvrir un compte à partir de la page principale.

La table "users" de la base de données "login":

```
CREATE TABLE `users` (  
  `id` int(11) NOT NULL,  
  `unique_id` varchar(23) NOT NULL,  
  `name` varchar(50) NOT NULL,  
  `username` varchar(20) NOT NULL,  
  `encrypted_password` varchar(80) NOT NULL,  
  `salt` varchar(10) NOT NULL /* la deuxième partie du mot de passe */  
);
```

Les fonctions crée en php afin de gérer la base de données:

```
public function storeUser($name, $username, $password)
```

Cette fonction permet d'ajouter un membre sur la BDD

```
public function getUserByLoginAndPassword($username, $password)
```

Cette fonction, est utilisée lors de la connexion d'un utilisateur, retourne null si id ou mot de passe ne correspond pas à celui d'un utilisateur enregistré sur la BDD

```
public function getUsers()
```

renvoie tous les utilisateurs enregistré dans la BDD

```
public function isUserExisted($username)
```

vérifie si un utilisateur existe dans la BDD

```
public function deleteUser($username)
```

efface un utilisateur

```
public function hashSSHA($password)
```

cryptage du mot de passe (clé sha1)

```
public function checkhashSSHA($salt, $password)
```

décryptage du mot de passe lors de la connexion

b) L'application

Pour mettre à jour l'application, tout part d'un bouton sur lequel il faut appuyer pour lancer la procédure. Le bouton est simple d'usage est peut être sécurisé par un code afin d'éviter que les patients (en bas âge) ne s'amuse avec la mise à jour des fichiers. De plus, il ajoute une sécurité supplémentaire contrairement à la mise à jour par détection de Wi-Fi puisque l'encadrant ne l'utilisera que s'il a modifié récemment un fichier sur le serveur.

Concernant la partie technique de la mise à jour des fichiers par l'application, voici comment cette méthode a été mise en place. Pour débiter, nous listons dans un dossier tous les fichiers téléchargeables depuis le serveur :

```
String[] dir = {  
    "contact_map.html",  
    "fiche_amb1.html",  
    "fiche_amb2.html",  
    "fiche_amb3.html",
```

Nous avons réalisé des tentatives pour rendre cette procédure automatique mais malheureusement, une application de ne peut lister un dossier distant directement. La tableau est mis sous forme de "String" car nous allons ajouter les chaînes de caractères à la fin d'un URL déjà défini.

Ensuite nous appliquons la méthode "execute()" à l'ensembles de fichiers à télécharger :

```
new DownloadFileAsync().execute(  
    url+dir[0],url+dir[1],url+dir[2],url+dir[3],url+dir[4],  
    url+dir[5],url+dir[6],url+dir[7],url+dir[8],url+dir[9],  
    url+dir[10],url+dir[11],url+dir[12],url+dir[13],url+dir[14],  
    url+dir[15],url+dir[16], url+dir[17]...
```

Ainsi, l'URL est `String url = "http://192.168.43.197/documents/"` avec notre adresse de serveur apache installé sur notre ordinateur personnel.

"DownloadFileAsync()", fait appel à une méthode DoInBackground qui va :

- établir la connexion :

```
URL url = new URL(params[i]);  
URLConnection connexion = url.openConnection();  
connexion.connect();
```

- mettre les données du fichier à télécharger dans un buffer :

```
InputStream input = new  
BufferedInputStream(url.openStream());
```

- enregistrer les données du buffer sous un nouveau nom :

```
OutputStream output = new FileOutputStream("/sdcard/SSR_docs/" +  
dir[i]);
```

Nous bouclons dans "dir" afin de conserver les même noms que précédemment et que les fichiers soient toujours utilisables dans les flash de codes QR ou autre activité qui requiert l'utilisation de WebView.

En début d'exécution, nous créons un nouveau dossier en racine appelé "SSR_docs" afin d'y stocker tous les fichiers téléchargés et ne pas déborder sur d'autres dossiers déjà utilisés. Pour cela, nous avons créé une méthode :

```
public static File CreateFile()  
{  
    File path = new  
File(Environment.getExternalStorageDirectory(), "/SSR_docs/");  
    path.mkdirs();  
    return path;  
}
```

Cette méthode sera appelée lors du "setOnClickListener" du bouton de mise à jour en tout début de procédure.

AJOUT D'UN MOT DE PASSE TÉLÉCHARGEMENT

Pour ajouter un mot de passe bloquant l'accès au téléchargement par les utilisateurs de l'application, nous avons créé une nouvelle vue demandant via un "textEdit" une chaîne de caractères. Ensuite nous réalisons une comparaison par rapport au mot de passe prédéfini. Si celle-ci renvoie 1, alors nous lançons la procédure de téléchargement décrite précédemment.

VII) Difficultés rencontrées

Les principales difficultés rencontrées ont été de trouver les méthodes nécessaires à la mise en place des différentes fonctionnalités de l'application. Il existe de nombreuses bibliothèques certaines avec des méthodes plus complexes que d'autres.

Android Studio a également été parfois un frein au développement du projet. En effet, lorsqu'une partie du code est commentée, puis décommentée, toutes les librairies doivent être redéclarées. Une situation pénible lors des situations de test. De plus, le logiciel nécessite une capacité de RAM importante, parfois trop importante pour réaliser des simulations. L'utilisation du GIT laisse à désirer (problèmes de push, de pull). Nous recommandons à l'avenir l'utilisation d'Eclipse.

Pour finir, la plus grande difficulté de notre projet a été de cibler les besoins du centre Marc Sautelet et de leur proposer un compromis entre leurs idées et nos capacités. En tout début de projet, l'équipe de rééducation souhaitait la création de jeux vidéos en 3D, ce qui aux vues du temps imparti mais aussi de nos capacités, rendait le projet irréalisable. Nous nous sommes donc concentrés sur la partie mise à jour des fichiers, garantissant la pérennité du projet.

VIII) Conclusion

Le projet “Amélioration de l’accueil des enfants hospitalisés” en collaboration avec le centre SSR Marc Sautelet a été l’opportunité pour nous de travailler sur un outil concret. Cet outil tire ses fonctionnalités des différentes réunions d’information qui ont eu lieu entre janvier et avril. En effet, les besoins exprimés par les professionnels du centre ainsi que leurs retours nous ont permis d’avancer pas à pas dans la bonne direction. Ainsi, nous avons créé une application adaptée aux téléphone et aux tablettes “larges” et “xlarges”. Cette application permet aux utilisateurs de découvrir le centre en entrant en interaction via les QR codes mais aussi de l’occuper lorsque celui-ci sera amené à patienter.

Nous avons gagné de l’expérience en capacité d’écoute et formulation des besoins. Le travail collaboratif a été enrichissant pour nous qui n’avions, à ce jour, travaillé avec le secteur professionnel que lors de stages. Toute l’application est codée en Java ainsi qu’en XML. Cela nous a donc permis de renforcer nos connaissances en programmation dans ces langages. La partie mise à jour via serveur Web sur laquelle nous avons travaillé en PHP et HTML a été très enrichissante puisqu’elle nous a permis de revoir des fonctions utilisées lors des séances de projet IMA3.

Nous vous remercions de l’attention que vous avez porté à ce rapport, en espérant que notre travail jouera son rôle dans l’accueil des patients.

Edmur LOPES
Robin CAVALIERI

Annexes

Tutoriel sur la modification des fiches

Etape 1: Identifiez-vous sur la page principale de connexion avec votre nom d'utilisateur et votre mot de passe :

Username

Password

Etape 2: Ensuite vous avez accès à la page de manipulation des fichiers. Pour modifier un fichier en question comme ici :

[fiche amb1.html](#) ---> [Edit](#) ---> [Delete](#)

Vous pouvez :

- visualiser la fiche en cliquant sur son nom
- Modifier la fiche en cliquant sur Edit
- Supprimer la fiche en cliquant sur Delete

Etape 3 : Pour modifier une fiche, vous n'avez qu'à modifier les parties situées entre les balises :

```
<!doctype html>
<head>
  <meta charset="utf-8">
</head>
<h1><big>Ambulancier</big></h1>
<center></center>
<p>
L'ambulancier est la personne qui conduit une ambulance.
Parfois, il peut être accompagné par un infirmier ou une infirmière.
</p>
</html>
```

Ici par exemple vous pouvez modifier le titre "Ambulancier" mais aussi le contenu "L'ambulancier est la personne...infirmière."

Vous pouvez également changer l'image en donnant le nom du fichier.

Une fois vos modifications effectuées, cliquez sur “Save”.

Etape 4 : Rendez-vous sur l’application avec une tablette connectée en Wi-Fi et cliquez sur l’onglet “Administrateur”, entrez le mot de passe et attendez la fin du téléchargement. Voilà, vos fichiers sont désormais mis à jour.

Veillez à conserver les même noms pour les fiches html : comme les explications sur les différents métiers mais aussi les fiches contenant l’organigramme.