
Projet Lourd E5 : Pantheon



Réalisé par Lucas Estingoy



Sommaire de la documentation :

Projet Lourd E5 : Pantheon	1
Sommaire de la documentation :	2
Projet : Pantheon	3
Introduction :	3
Spécifications fonctionnelles :	4
Spécification Technique :	6
Architecture du site :	7
Conception du site :	9
Charte graphique :	10
Création de l'application :	11
Partie XML :	11
Partie Java :	12
Création de l'APK :	13
Suite de l'application :	14
Mise à jour :	14

Projet : Pantheon

Introduction :

Le projet Pantheon consiste en la création d'une application mobile à l'aide du logiciel Android Studio. Cette application permettra à l'utilisateur d'accéder à son programme de musculation personnalisé ainsi qu'à une liste complète d'exercices avec des guides visuels explicatifs, sous forme de vidéos ou de photos.

Le fichier d'installation (.apk) sera généré automatiquement à la fin du développement, permettant ainsi à l'utilisateur d'installer ou de mettre à jour l'application sur n'importe quel appareil Android.

Rendu de l'accueil de l'application :



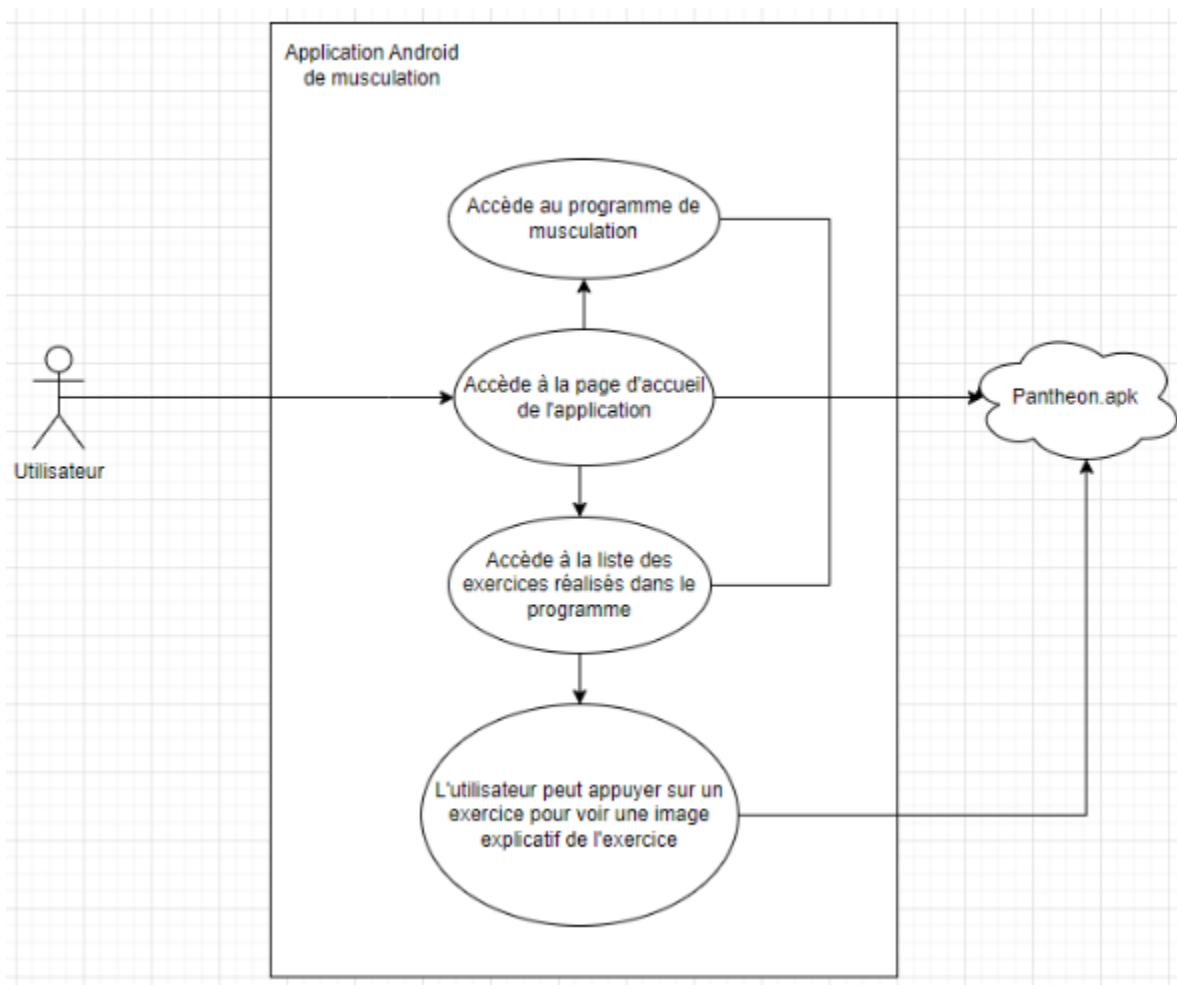
Spécifications fonctionnelles :

L'application Pantheon offrira à l'utilisateur la possibilité de naviguer entre trois pages distinctes : la page d'accueil, la page du programme de musculation et la page des exercices.

En cliquant sur l'onglet dédié, l'utilisateur pourra accéder facilement à son programme de musculation personnalisé, qui sera présenté de manière claire et organisée.

La liste des exercices sera également disponible sur une page dédiée, avec des guides visuels explicatifs pour chaque exercice. Lorsque l'utilisateur cliquera sur un exercice, une image explicative s'affichera pour l'aider à comprendre la bonne exécution du mouvement.

Enfin, la page d'accueil permettra à l'utilisateur de revenir facilement à ces différentes sections de l'application, et d'accéder à d'autres fonctionnalités pratiques, telles que la configuration de son profil utilisateur.



En somme, Pantheon offrira une expérience utilisateur fluide et intuitive, en permettant un accès facile à toutes les fonctionnalités essentielles de l'application.

Spécification Technique :

Le développement de l'application Pantheon a été réalisé à l'aide d'Android Studio, un environnement de développement intégré (IDE) conçu pour créer des applications Android.

. Java : Le langage de programmation utilisé pour la création de l'application est Java, qui est un langage informatique couramment utilisé pour la création d'applications web, mobiles et de bureau.



. XML : Le format de balisage XML (Extensible Markup Language) a également été utilisé pour la création de l'interface utilisateur. Cela a permis de créer une présentation visuelle claire et organisée pour l'ensemble de l'application.

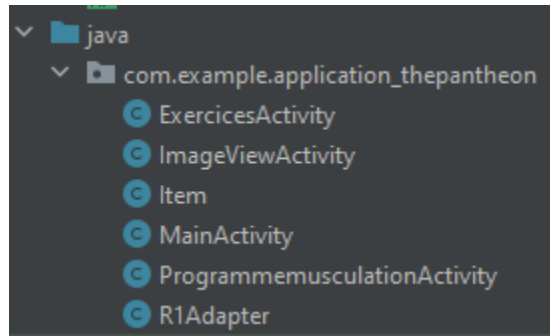


Enfin, Android Studio a été utilisé pour la compilation de l'application et la génération de son fichier d'installation, le .apk. Cette étape finale est essentielle pour permettre à l'utilisateur d'installer et de mettre à jour facilement l'application sur n'importe quel appareil Android.

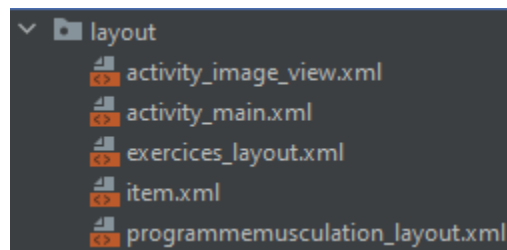


Architecture du site :

L'application Pantheon utilise une architecture de six pages, codées en Java, pour sa création. Chacune de ces pages a une fonction spécifique qui contribue à la fonctionnalité globale de l'application.



La page ExercicesActivity est utilisée pour créer différents exercices, tandis que la page ImageViewActivity permet la création du programme de musculation. L'item.xml est utilisé pour définir les attributs de chaque exercice, tandis que le MainActivity est utilisé pour créer la page d'accueil. La page Programmemusculatation est utilisée pour créer la page du programme de musculation, tandis que R1Adapter est utilisé pour détecter où l'utilisateur appuie.

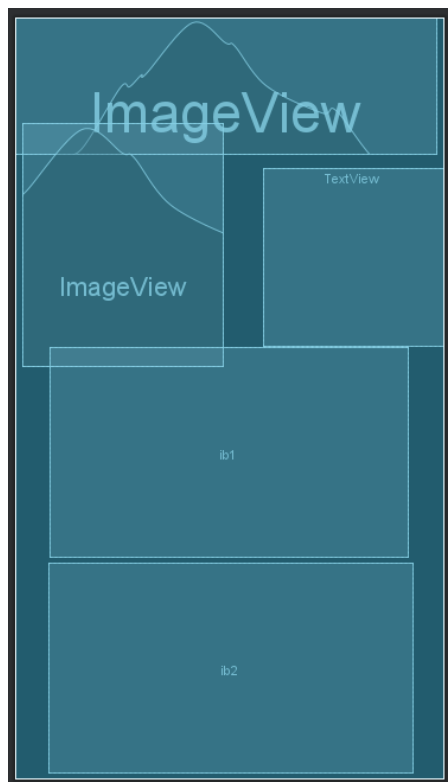
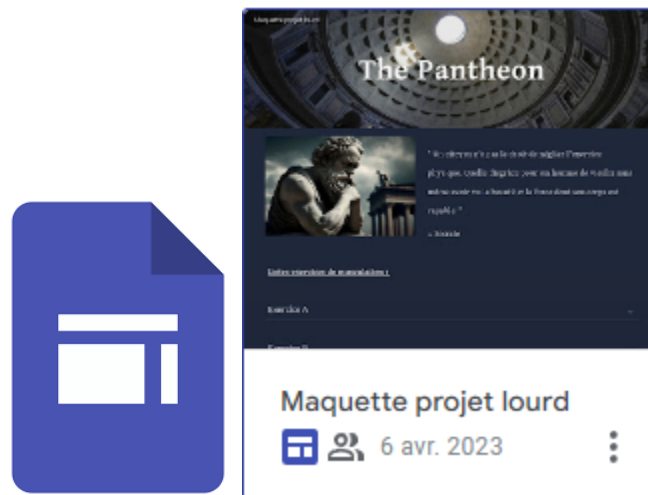


En plus de ces pages Java, l'application utilise également une architecture XML pour créer les différentes pages visuelles. L'activitymain est utilisée pour créer la page d'accueil, l'exerciceslayout est utilisé pour créer la page des exercices, et le programmemusculaton est utilisé pour créer la page du programme de musculation. La page activityimage est utilisée pour créer une image lorsqu'un utilisateur appuie sur un exercice, l'image change en fonction de l'URL définie. Enfin, item.xml est utilisé pour définir comment chaque exercice sera affiché dans l'application.

Cette architecture permet à l'application de fournir une interface utilisateur claire et organisée pour les utilisateurs, tout en offrant des fonctionnalités puissantes pour la création de programmes de musculation et l'accès à des listes d'exercices détaillées.

Conception du site :

L'application a été créée en deux parties, la première est la création d'une maquette grâce à l'outil Google Site permettant facilement la création de maquette.



La seconde est la création sur Visual Studio du visuel en XML pouvant être fait de façon manuel en déplaçant les différentes parties tel qu'une image ou un texte.

Charte graphique :

Le but graphique de l'application est de se rapprocher du style grec, pour cela une image de pantheon et d'une statue de Socrate, un philosophe grec, on été utilisé. Ensuite la couleur du fond est un bleu foncé pour évité d'aggraver les yeux de l'utilisateur si la salle est peu éclairé.

La couleur utilisé pour le fond est :
. Le bleu foncé : #1f283b

Style de police : Le texte est réalisé en blanc avec la police natif d'Android Studio.



Ainsi avec cette charte graphique nous retrouvons bien le style grec recherché. Pour le logo l'utilisation d'un panthéon blanc sous fond noir a donc été utilisé.



Création de l'application :

Partie XML :

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:clickable="true">

    <TextView
        android:id="@+id/titre"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="20sp"
        android:textStyle="bold"
        android:textColor="@android:color/white"/>

    <TextView
        android:id="@+id/description"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="16sp"
        android:textColor="@android:color/white"/>

    <ImageView
        android:id="@+id/image"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:adjustViewBounds="true"
        android:scaleType="centerCrop" />

    <View
        android:layout_width="match_parent"
        android:layout_height="1dp"
        android:background="@android:color/darker_gray" />

</LinearLayout>
```

Le code XML permet la création visuel de l'application.

Comme l'exemple ci-join elle implémente les différentes parties visuels, dans ce cas là celui des exercices, avec deux textes, une image et un view simple servant à mettre un fond gris.

Il y a donc 5 codes XML différents permettant le rendu visuel global de l'application.

Le code XML se mélangera alors au code Java pour fusionner le visuel de l'application avec le fonctionnement de celui-ci et rendre alors chaque élément appuyable par l'utilisateur.

Partie Java :

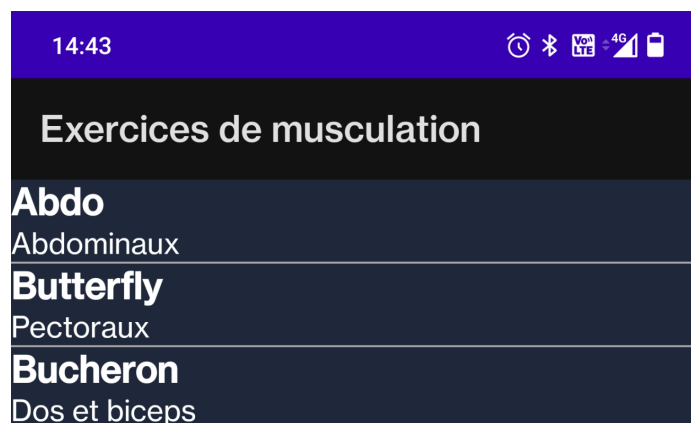
```
public void writeToParcel(Parcel dest, int flags) {
    dest.writeString(name);
    dest.writeString(description);
    dest.writeInt(imageId);
}
```

```
items.add(new Item( name: "Abdo", description: "Abdominaux", R.drawable.abdo));
```

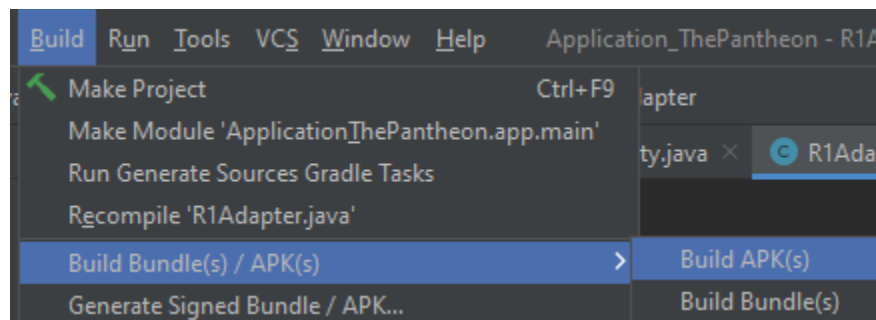
La partie Java permet le fonctionnement de l'application, c'est lui qui rendra les exercices responsifs ou donnera aux items les différents éléments de l'exercice étant donc le nom, la description et l'image correspondante.

```
if (listener != null) {
    int position = getAdapterPosition();
    if (position != RecyclerView.NO_POSITION) {
        listener.onClick(position);
    }
}
```

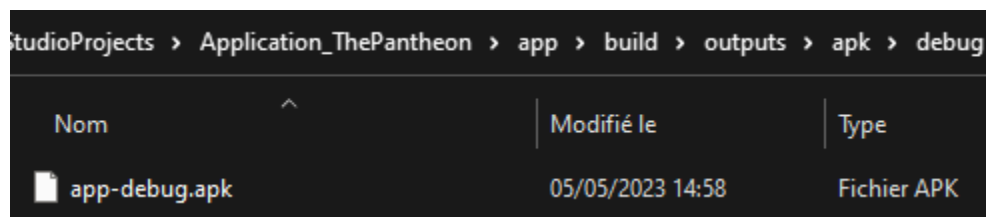
Ainsi grâce à la partie R1Adapter il est possible de détecter où l'utilisateur appuie et ainsi afficher l'image dans l'application.



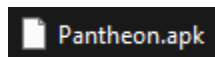
Création de l'APK :



Pour créer un .apk avec Visual Studio il suffit seulement d'utiliser la fonction Build Apk de l'application, en effet, Visual Studio a été fait pour faciliter la création d'application mobile et permet ainsi de transformer facilement un code en apk.



Une fois l'apk créé il est possible de "localiser" l'apk depuis l'application pour arriver ainsi sur l'emplacement de l'apk sur l'ordinateur. Ainsi il vous est aussi possible de renommer ce dossier et ainsi le faire télécharger à votre utilisateur pour installer l'application sur l'appareil.



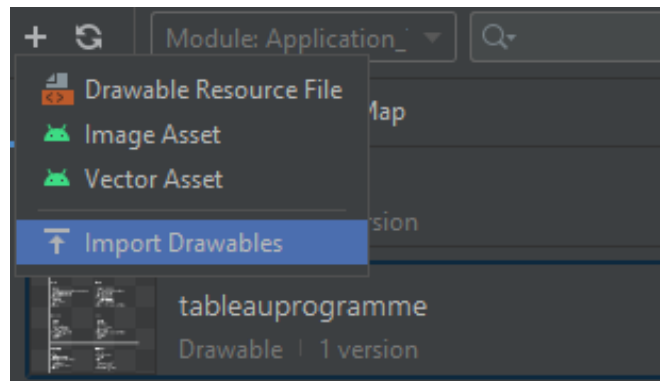
<https://github.com/Itsweeazie/Pantheon>

Suite de l'application :

Mise à jour :

L'application pourra être mise à jour en modifiant le .apk avec la nouvelle version. Que ce soit le rajout d'une page, le rajout d'un exercice ou la modification du programme de musculation.

Pour modifier le programme de musculation il suffit de modifier le tableau se trouvant le fichier "maintenance" au nom de "Tableau programme maintenance". Une fois le tableau modifié et un imprime écran effectué (maj+alt+s) il suffit avec un logiciel comme Photofiltre de retirer le fond noir en le transformant en transparent puis d'importer l'image comme ci-dessous.



Une fois l'image importée il suffit de transformer l'imageView avec la nouvelle image, transformer en .apk et la nouvelle version de l'application avec le nouveau programme est fait.

Lien vers le github : <https://github.com/Itsweeazie/Pantheon>