



UNIVERSITÉ
**PARIS
DESCARTES**



Université de Paris

DOSSIER APPMOB

**I.U.T. De Paris
Département informatique
Année Universitaire 2020-2021
Groupe 209**



*Boufares Adam
IN Keryann
Cardos Calin
Bashiti Muntaser*

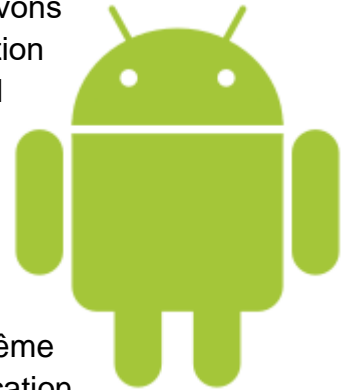


Table des matières

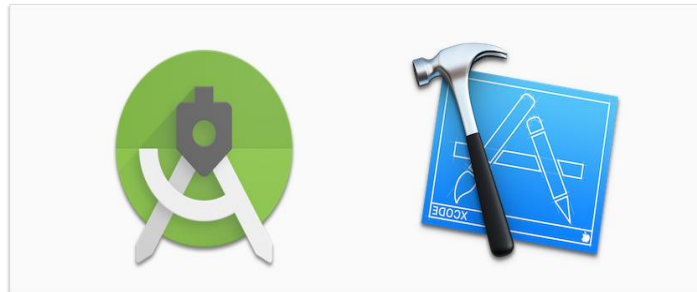
| | |
|--|---|
| 1. Introduction | 3 |
| 2. Fonctionnalités principales de l'application..... | 4 |
| 2.1. Présentation du but de l'application | 4 |
| 3. Langages et Outils utilisés | 5 |
| 3.1. Langage | 5 |
| 3.2. Outils | 5 |
| 3.2.1. Android Studio | 5 |
| 3.2.2. Firebase | 6 |
| 3.3. API | 6 |
| 4. Conclusion | 7 |
| 4.1. Bilan | 7 |
| 4.2. Difficultés rencontrées | 7 |
| 4.3. Améliorations possibles | 7 |

1. Introduction

Ce projet du module AppMob a pour but de nous faire découvrir le développement d'applications Android. Comme nous le savons depuis quelques années, Android est le système d'exploitation le plus utilisé au monde. De plus depuis un certain temps, il est aussi considéré comme l'un des systèmes d'exploitation les plus sécurisés, puisqu'il existe une multitude de version.



Android a toujours été un système d'exploitation, qui permet une énorme liberté au développeur, et il en est de même pour Android Studio qui permet de développer une application Android, mais qui est aussi une multiplate-forme contrairement à son concurrent Xcode, qui est dédié au développement sur macOS.



Pour notre projet, Cutube, différents critères ont dû être respectés, pour être en accord avec ce module d'App Mob.

- Les éléments suivants ont dû être respectés :
 - Un schéma d'architecture simple
 - une IHM
 - passer d'une activité à une autre
 - Stocker de l'information
 - utiliser des libs
 - "qualité" du travail en dev

2. Fonctionnalités principales de l'application

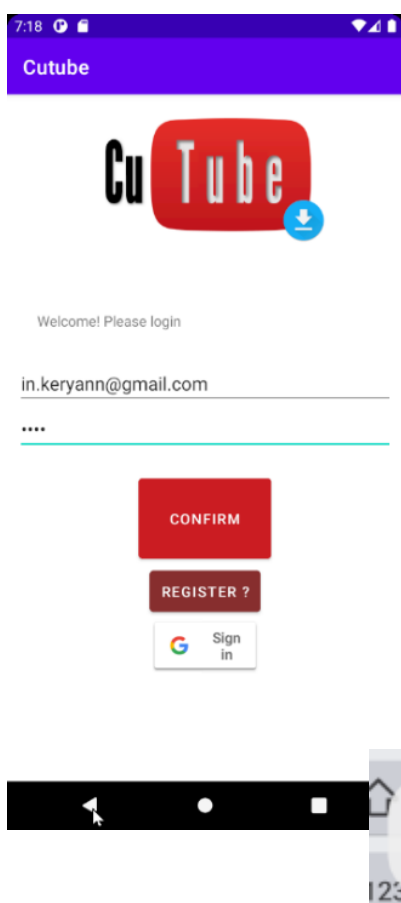
2.1. Présentation du but de l'application

Notre application a pour but de permettre à l'utilisateur de télécharger une vidéo Youtube en format MP4 de la manière, la plus simple possible pour l'utilisateur. Grâce à l'utilisation d'une API, le téléchargement de la vidéo sous un format MP4 est possible, et cela ne nécessite que, que le client rentre le lien https://.



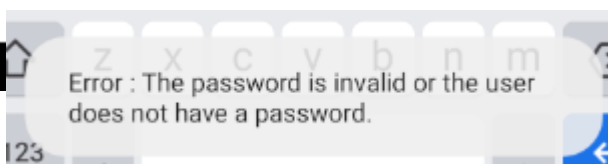
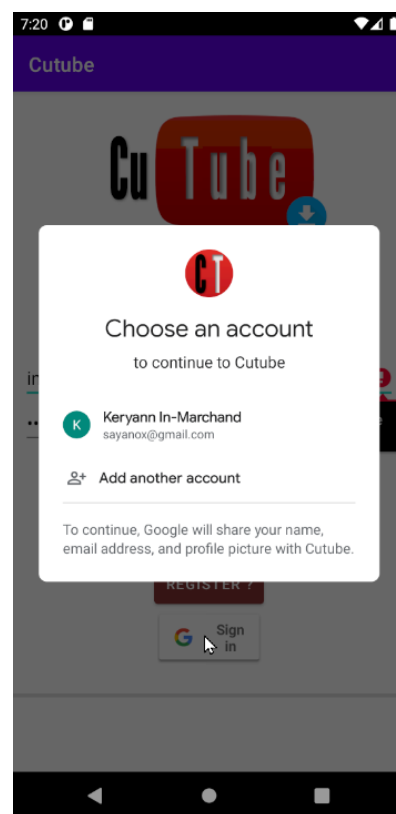
2.2. Présentation des fonctionnalités

2.2.1. Connection



Pour pouvoir avoir accès à nos services, l'utilisateur doit posséder au minimum une d'adresse google. De plus si l'utilisateur n'a pas d'adresse google ou veut pas la connecter à l'application, nous lui offrons la possibilité de pouvoir créée un compte sur notre application de manière très simpliste.

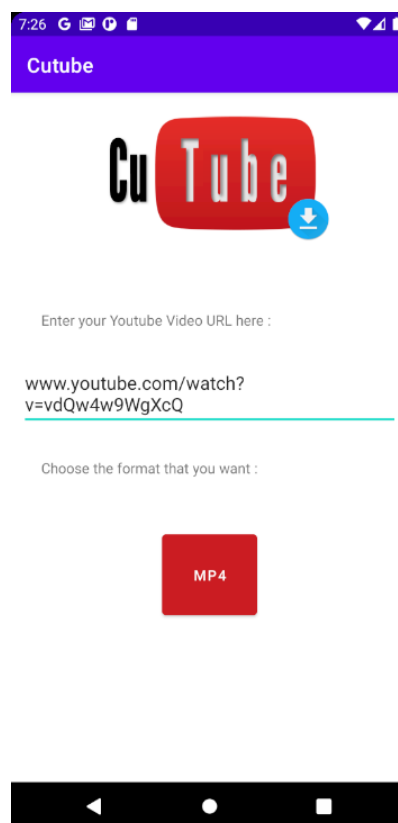
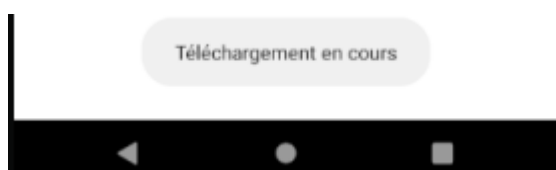
En cas d'erreur sur l'adresse mail ou le mot de passe, et si l'utilisateur n'existe pas un message d'erreur va prévenir l'utilisateur d'un problème de connexion



2.2.2. Service téléchargement

Le service de téléchargement permet, à l'utilisateur de télécharger sous un format MP4, la vidéo Youtube, du lieu qu'il doit saisir ou coller dans la zone se saisit dédié au service.

À la suite de quoi il doit valider, et si le lien est correct, et que la vidéo existe, alors l'application télécharge sous format MP4 la vidéo du lieu. De plus l'utilisateur est prévenu du début du téléchargement.



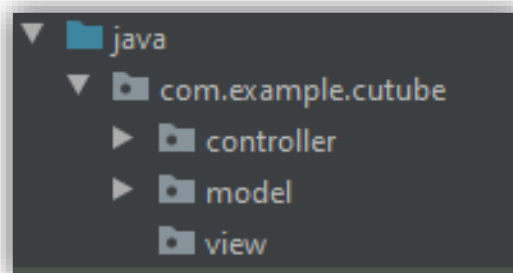
3. Langages et Outils utilisés

3.1. Langage

Le langage que nous avons utilisé pour développer l'application en elle-même est **Java**, c'est l'un des langages éligibles à Android Studio et le plus conseillé d'après nos recherches.

Pour l'aspect **IHM** de l'application les pages/activités sont « codées » en langage **XML**, c'est ce que l'on nomme les **layout**.

Aussi, nous avons adopté une architecture **MVC (Model, View, Controller)** pour structurer l'application.

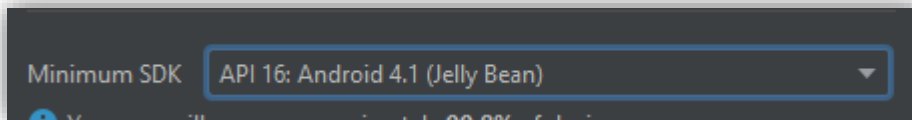


3.2. Outils

3.2.1. Android Studio

Le logiciel utilisé pour développer l'application est **Android studio** vu que cette dernière est faite pour fonctionner tout d'abord sur l'OS de Google.

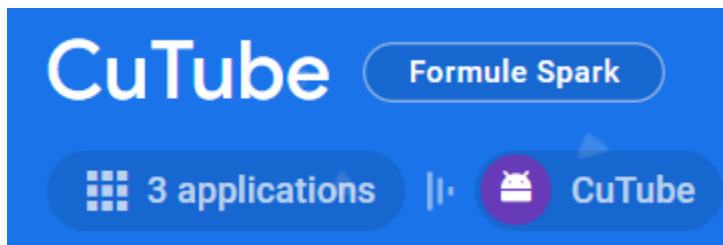
La **version minimale d'Android** que nous avons choisie est « **Android 4.1 (Jelly Bean)** » sorti en 2012 pour englober le maximum d'appareils Android compatibles.



3.2.2.

Firebase

Afin d'authentifier nos utilisateurs, un système d'authentification a été réalisé grâce à l'**API Firebase**. Ainsi toutes les connexions et la liste de nos utilisateurs sont conservées et accessibles depuis la console Firebase.



| Identifiant | Fournisseurs |
|----------------------|--------------|
| test@test.test | ✉ |
| in.keryann@gmail.com | ✉ |
| test2@test.test | ✉ |

3.3. API

Notre API est constituée de deux APIs :

- Le premier est une bibliothèque qui décode le lien YouTube vers un lien téléchargeable
- La deuxième c'est pour sauvegarder la vidéo après qu'il est décodé par la bibliothèque



Lors du téléchargement le fichier sera nommé « vidéo téléchargé par CuTube ».

Notre problème pour nous c'était d'avoir une API qui marche, pour pouvoir avoir le téléchargement.

Puisque c'est YouTube, il y a plusieurs API pour le téléchargement donc la recherche sur Google était difficile et il nous fallut 5 heures pour trouver les bonnes APIs.

Ensuite le deuxième problème c'étaient les versions des SDK. Le problème général a cause de ça c'est « import android.support.v7.app.AppCompatActivity cannot resolve symbol » car la deuxième API était trop ancienne donc c'était un problème. Pour résoudre on a dû faire un changement dans l'import de v7 par androidx.appcompat avec quelques changements pour que l'ancien API marche bien avec notre SDK 30.

4. Conclusion

4.1. Bilan

Ce projet fut vraiment très enrichissant, avec la découverte d'un tout nouveau système d'exploitation. Il nous a permis d'approfondir nos connaissances sur les IHM avec des émulateurs qui simule les smartphones Android qui nous facilite la tâche en tant que développeurs.

De plus le projet nous a permis de voir qu'il est très compliqué de trouver un sujet d'application qui est assez innovant car il existe des applications pour un nombre incalculable de choses. De même que pour les million API le choix est immense.

4.2. Difficultés rencontrées

Pour ce qui est de l'**authentification** avec Firebase, la mise en place fût rude mais c'est surtout la tentative de Sign In avec l'**api Google services** qui à freiner l'évolution de notre système d'authentification. Malgré l'ajout de la clé SHA 1 et de l'activation de ce service dans la console Firebase le Sign IN google n'a pas marché.

4.3. Améliorations possibles

Au niveau de l'interface Homme Machine, l'ajout d'un **fond d'écran** à l'application pourrait améliorer le rendu actuel. Aussi, il serait possiblement préférable d'ajouter certains **éléments graphiques** voir d'en espacer certains pour occuper au mieux l'espace sur l'écran.

Pour ce qui est de la gestion des utilisateurs, étendre les **moyens de connexion**, sécurisé la création de compte avec par exemple des **mails de vérifications** ou encore des sms peut pousser la sécurité au sein de notre app. Enfin, si l'**api Google** fonctionnait avec notre projet, on aurait pu chercher à lier nos utilisateurs à leur compte YouTube pour permettre une utilisation fluide de l'app.