



*Faculté des Sciences de Montpellier*



## **Rapport de projet android**

### **Ecole en ligne**

Réalisé par:

Thierno Barry  
Karim Dahdouh

Encadrant:

























































M. Abdelhak-Djamel Seriai

# Sommaire

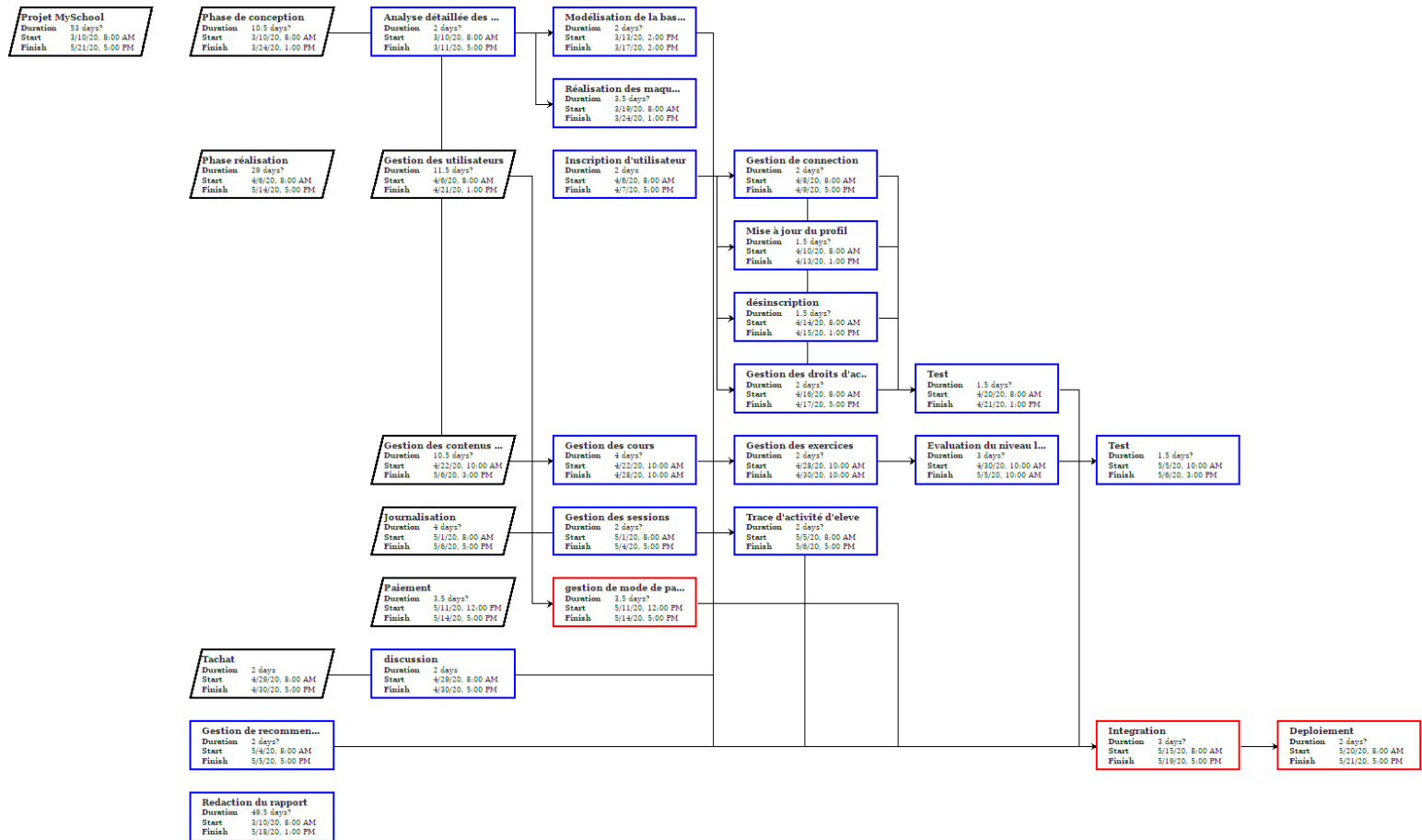
Conduite de projet MySchool.....	3
Liste des tâches.....	3
Diagramme de Perte.....	5
Diagramme de Gantt .....	6
I. Conception.....	7
Base de données.....	9
Architecture technique.....	10
II. Outils de développement.....	11
IDE Eclipse STS (Spring Tool Suite) :.....	11
Android Studio :.....	12
Gestionnaire de dépendance Maven.....	12
Spring Boot & Spring Initializr.....	12
SGBD MySQL.....	12
Postman pour tester les les web services RESTful.....	13
Github pour la gestion des versions.....	13
Cloud Foundry pour déployer la partie backend du projet MySchool.....	13
III. Frontend : Fonctionnalités.....	13
Choix du profil.....	13
Inscription de parent et d'étudiant.....	14
Login.....	15
Validation des données saisies.....	16
Récupérer le mot de passe.....	16
Deconnexion d'utilisateur.....	17
Activation compte par mail.....	17
Création des profils : élève et Parent.....	18
Profil d'utilisateur connecté.....	19
Gestion des cours.....	19
Historiques des cours.....	26
Paieement.....	26
Desinscription.....	27
Recommandation.....	28
Internationalisation.....	28
Chat en temps réel.....	30
Partage.....	30
IV. Version offline et synchronisation.....	32
V. Backend:RESTful webservices (framwork spring).....	33
Webservice : Login ou authentification.....	33
Webservice: Inscription.....	33
webservice: Mot de passe oublier.....	34
webservice: Déconnexion.....	35
VI. Déploiement.....	35
Consommation des webservices RESTful coté Android.....	37
Liens du projet MySchool.....	38
Version en ligne des webservices RESTful.....	38
Github .....	38

# Conduite de projet MySchool

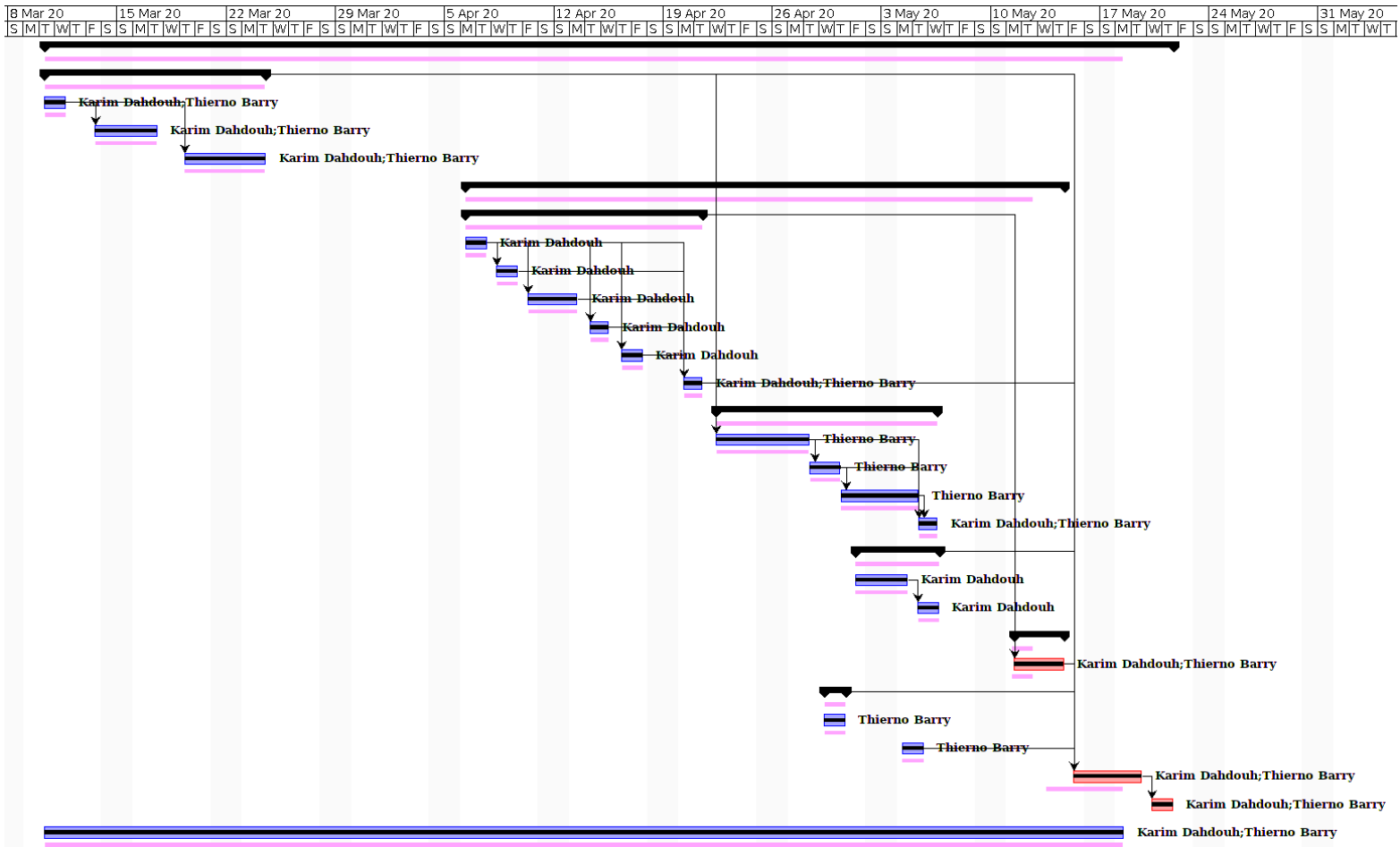
## Liste des tâches

	Work		Name	Percent C...	Duration	Start	Finish	Resource Names	Predecessors
1	1,288 h...	 	☐Projet MySchool	100%	53 days?	3/10/20, 8:00 AM	5/21/20, 5:00 PM		
2	120 hours	 	☐Phase de conception	100%	10.5 days?	3/10/20, 8:00 AM	3/24/20, 1:00 PM		
3	32 hours	 	Analyse détaillée des besoins	100%	2 days?	3/10/20, 8:00 AM	3/11/20, 5:00 PM	Karim Dahdouh;Thierno Barry	
4	32 hours	 	Modélisation de la base de données	100%	2 days?	3/13/20, 2:00 PM	3/17/20, 2:00 PM	Karim Dahdouh;Thierno Barry	3
5	56 hours	 	Réalisation des maquettes	100%	3.5 days?	3/19/20, 8:00 AM	3/24/20, 1:00 PM	Karim Dahdouh;Thierno Barry	3
6	280 hours	 	☐Phase réalisation	100%	29 days?	4/6/20, 8:00 AM	5/14/20, 5:00 PM		
7	96 hours	 	☐Gestion des utilisateurs	100%	11.5 days?	4/6/20, 8:00 AM	4/21/20, 1:00 PM		
8	16 hours	 	Inscription d'utilisateur	100%	2 days	4/6/20, 8:00 AM	4/7/20, 5:00 PM	Karim Dahdouh	
9	16 hours	 	Gestion de connexion	100%	2 days?	4/8/20, 8:00 AM	4/9/20, 5:00 PM	Karim Dahdouh	8
10	12 hours	 	Mise à jour du profil	100%	1.5 days?	4/10/20, 8:00 AM	4/13/20, 1:00 PM	Karim Dahdouh	8
11	12 hours	 	désinscription	100%	1.5 days?	4/14/20, 8:00 AM	4/15/20, 1:00 PM	Karim Dahdouh	8
12	16 hours	 	Gestion des droits d'accès	100%	2 days?	4/16/20, 8:00 AM	4/17/20, 5:00 PM	Karim Dahdouh	8
13	24 hours	 	Test	100%	1.5 days?	4/20/20, 8:00 AM	4/21/20, 1:00 PM	Karim Dahdouh;Thierno Barry	9;8;11;12;10
14	96 hours		☐Gestion des contenus pédagogiques	100%	10.5 days?	4/22/20, 10:00 AM	5/6/20, 3:00 PM		
15	32 hours	 	Gestion des cours	100%	4 days?	4/22/20, 10:00 AM	4/28/20, 10:00 AM	Thierno Barry	2;7
16	16 hours	 	Gestion des exercices	100%	2 days?	4/28/20, 10:00 AM	4/30/20, 10:00 AM	Thierno Barry	15
17	24 hours	 	Evaluation du niveau l'eleve	100%	3 days?	4/30/20, 10:00 AM	5/5/20, 10:00 AM	Thierno Barry	16
18	24 hours		Test	100%	1.5 days?	5/5/20, 10:00 AM	5/6/20, 3:00 PM	Karim Dahdouh;Thierno Barry	15;16;17
19	32 hours	 	☐Journalisation	100%	4 days?	5/1/20, 8:00 AM	5/6/20, 5:00 PM		
20	16 hours	 	Gestion des sessions	100%	2 days?	5/1/20, 8:00 AM	5/4/20, 5:00 PM	Karim Dahdouh	
21	16 hours	 	Trace d'activité d'eleve	100%	2 days?	5/5/20, 8:00 AM	5/6/20, 5:00 PM	Karim Dahdouh	20
22	56 hours	 	☐Paieement	100%	3.5 days?	5/11/20, 12:00 PM	5/14/20, 5:00 PM		
23	56 hours	 	gestion de mode de paiement	100%	3.5 days?	5/11/20, 12:00 PM	5/14/20, 5:00 PM	Karim Dahdouh;Thierno Barry	7
24	16 hours		☐Tachat	100%	2 days	4/29/20, 8:00 AM	4/30/20, 5:00 PM		
25	16 hours	 	discussion	100%	2 days	4/29/20, 8:00 AM	4/30/20, 5:00 PM	Thierno Barry	
26	16 hours	 	Gestion de recommandation	100%	2 days?	5/4/20, 8:00 AM	5/5/20, 5:00 PM	Thierno Barry	
27	48 hours	 	Integration	100%	3 days?	5/15/20, 8:00 AM	5/19/20, 5:00 PM	Karim Dahdouh;Thierno Barry	13;2;19;24;23;26
28	16 hours	 	Deploiemet	100%	2 days?	5/20/20, 8:00 AM	5/21/20, 5:00 PM		27
29	792 hours	 	Redaction du rapport	100%	49.5 days?	3/10/20, 8:00 AM	5/18/20, 1:00 PM	Karim Dahdouh;Thierno Barry	

# Diagramme de Perte

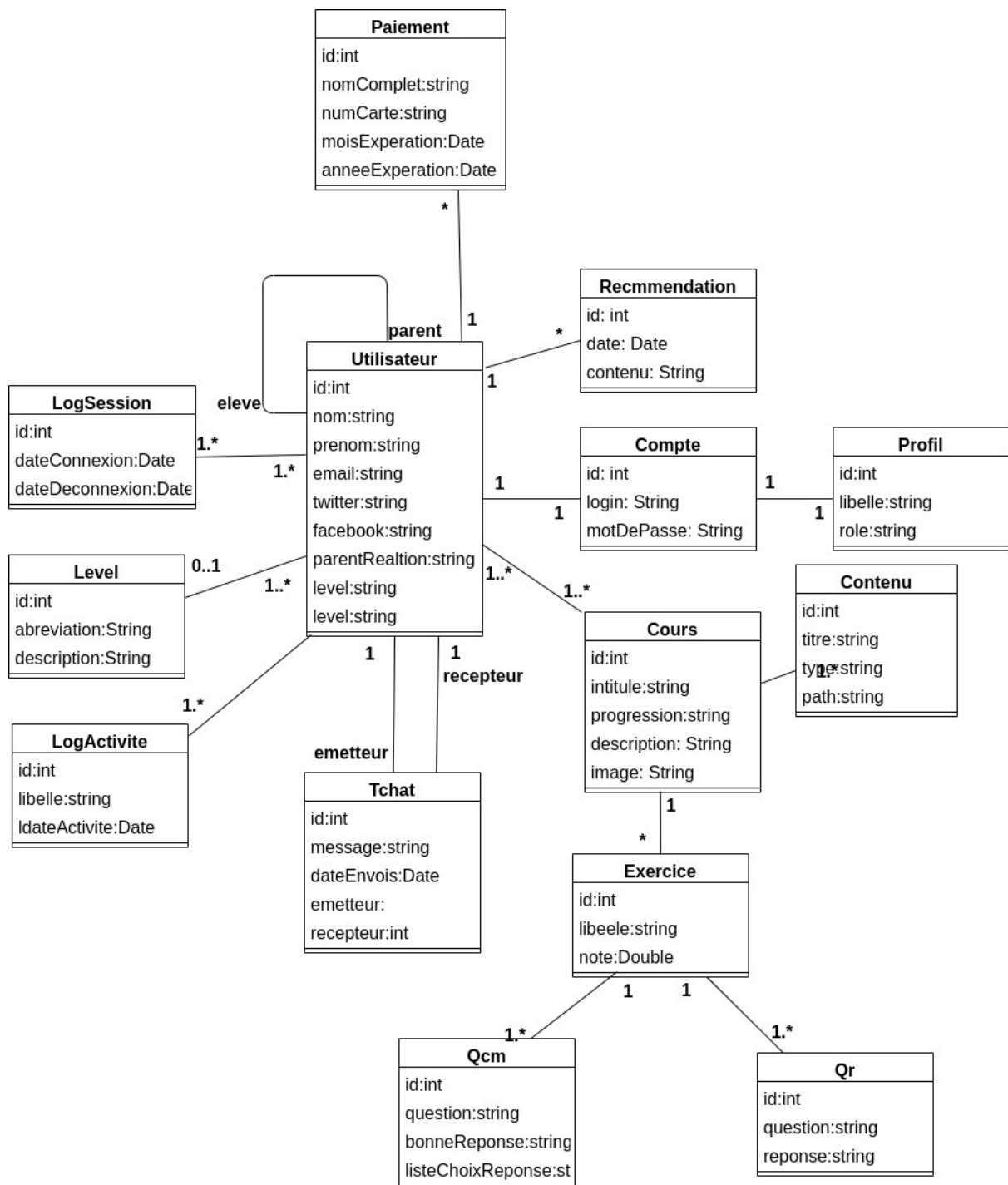


## Diagramme de Gantt



## I. Conception

Afin de concevoir le projet Myschool, Nous nous sommes basés sur le langage UML. Le diagramme de classe suivant illustre dans la figure ci-dessous l'ensemble des entités de notre base de données.



**Classe utilisateur** : permet de contenir les informations personnelles de l'utilisateur connecté au système telles que le nom, prénom, adresse e-mail, etc. Dans le cas où l'utilisateur est un élève, on devrait fournir son niveau scolaire et son parent.

**Classe compte** : contient le login et le mot de passe de l'utilisateur, ainsi qu'un attribut activé qui représente l'état de compte (actif ou non).

**Classe profile** : Il y a deux profils parent avec le rôle « ROLE\_PARENT » et élève avec le rôle « ROLE\_ELEVE » ou professeur « ROLE\_PROF ».

**Classe LogSession** : permet de sauvegarder les sessions de connexion de chaque utilisateur, surtout la date de connexion et déconnexion au système MySchool.

**Classe LogActivité** : garde les traces d'activités de l'utilisateur connecté, y compris les cours, les exercices, etc.

**Classe Level** : représente tous les niveaux scolaires à partir de cours préparatoire (CP) jusqu'au baccalauréat.

**Classe Cours** : contient la liste des cours dispensés aux élèves. Plus particulièrement, l'intitulé, la description et l'image de cours. Plus la progression de l'élève dans le cours.

**Classe Contenu** : représente le contenu de cours : pdf, vidéo, ou autre.

**Classe Exercice** : contient les exercices liés à chaque cours. Un cours peut être sous forme de QR (Question-réponse) ou de type QCM.

Ce diagramme de classe contient l'ensemble de classes permettant aux parents de bien gérer les contenus pédagogiques de ses enfants, ainsi qu'aux élèves de suivre leurs cours n'importe quand, n'importe où à partir de leur smartphone.

## Base de données

Pour implémenter notre modèle de classe UML, nous avons choisi d'employer de SGBD MySQL. L'implémentation de notre modélisation au niveau de la base de données est illustrée dans la figure :

```
+-----+
| Tables_in_myschool |
+-----+
| compte              |
| course              |
| course_content      |
| course_progress     |
| exercice            |
| exercice_progress   |
| listchoices         |
| log_activity        |
| log_session         |
| paiement            |
| profile             |
| qcm                 |
| qr                  |
| recommendation      |
| tchat               |
| user                |
| user_course         |
+-----+
17 rows in set (0.01 sec)

mysql>
```

Ci-dessous, un aperçu des données stockées dans notre base de données. (**user, compte et profile**)

```
mysql> select * from user;
+-----+
| id | email | facebook | firstname | lastname | twitter | compte_id | parent_id | level | parent_relatio |
+-----+
| 81 | karim.dahdouh.fr@gmail.com | test face | karim | dahdouh | test twitter | 56 | NULL | NULL | NULL |
| 83 | barryzainouli4@gmail.com | jhklm | ooo | ppp | jkjl | 58 | NULL | NULL | NULL |
| 86 | NULL | NULL | yanis | Bonnal | NULL | 61 | 81 | Cours préparatoire (CP) | Father |
| 88 | NULL | NULL | sebastiane | Martinize | NULL | 63 | 81 | Cours préparatoire (CP) | Mother |
| 90 | NULL | NULL | lean-luc | Dalman | NULL | 65 | 81 | Cours préparatoire (CP) | Father |
| 91 | test323@gmail.com | facebook | sebastiane | bounnal | twitter | 66 | NULL | NULL | NULL |
+-----+
6 rows in set (0.00 sec)

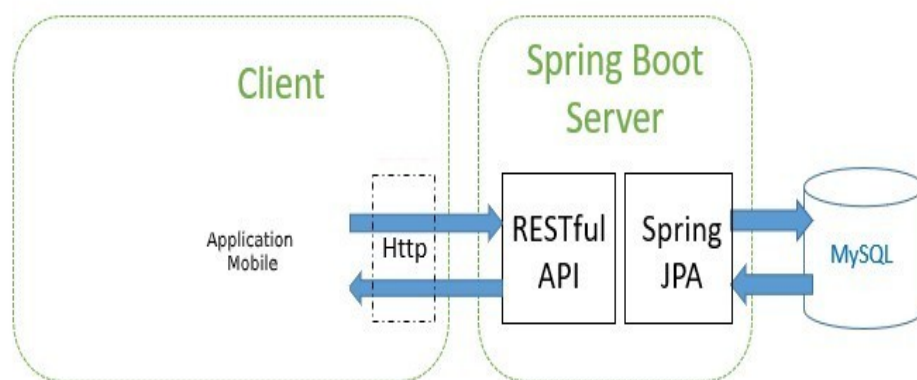
mysql> select * from compte;
+-----+
| id | login | password | profile_id | activate |
+-----+
| 56 | karim | karim | 1 | yes |
| 58 | barry | barry | 1 | no |
| 61 | yanis | yanis | 2 | yes |
| 63 | sebastiane | siabasian1 | 2 | yes |
| 65 | jeanluc | jeanluc | 2 | yes |
| 66 | logintest | passtest | 1 | no |
+-----+
6 rows in set (0.00 sec)

mysql> select * from profile;
+-----+
| id | libelle | role |
+-----+
| 1 | parent | ROLE_PARENT |
| 2 | eleve | ROLE_ELEVE |
+-----+
2 rows in set (0.00 sec)
```

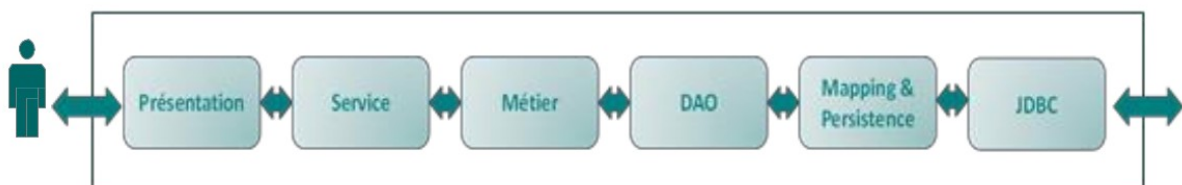


## Architecture technique

Notre architecture est une architecture client serveur. Le client est représenté par notre application android qui communique avec le serveur web. Ce dernier est chargé de faire les traitements nécessaires en intégrant une base de données gérée par le SGBD MySQL. Donc, la logique métier se trouve au niveau du serveur publiée sous forme des webservices RESTful. Le schéma suivant donne un récapitulatif de l'architecture du projet MySchool.



Vu que l'application se base sur l'architecture J2EE, et sachant que, pour les meilleurs résultats, toute application java devra être découpée en couches chacune ayant un rôle bien défini. Alors notre architecture technique mise en place est décomposée en plusieurs couches. Le schéma ci-dessous donne un aperçu de cette architecture :



Cette architecture comporte six couches, chacune d'entre elle a un rôle bien déterminé :

- ◆ **Couche Présentation:** c'est la couche qui dialogue avec l'utilisateur, via une interface graphique web. Elle a pour rôle de fournir des données provenant de l'utilisateur à la couche métier ou bien de présenter à l'utilisateur des données fournies par la couche métier.

- ◆ **Couche Services:** Cette couche traduit les scénarios identifiés lors de la phase de conception sous forme de services mis à disposition des utilisateurs
- ◆ **Couche Métier:** cette couche applique les règles dites métier, c'est-à-dire la logique spécifique de l'application, sans se préoccuper de savoir d'où viennent les données qu'on lui donne, ni où vont les résultats qu'elle produit.
- ◆ **Couche DAO:** Data Access Object est la couche qui fournit à la couche métier des données préenregistrées (fichiers, bases de données...) et qui enregistre certains des résultats fournis par couche métier.
- ◆ **Couche Mapping et Persistance:** Cette couche se charge de la persistance des objets et du mapping (association des objets aux éléments de la BD relationnelle), elle se place entre la couche DAO écrite par le développeur et la couche JDBC. Elle présente un outil qui fait le pont entre le monde relationnel des bases de données et celui des objets manipulés par Java. Ce pont est fait principalement dans notre cas par des annotations Java dans le code.
- ◆ **Couche JDBC:** C'est la couche standard utilisée en Java pour accéder à des bases de données. Elle isole la couche DAO du SGBD qui gère la base de données. On peut théoriquement changer de SGBD sans changer le code de la couche DAO.

## II. Outils de développement

Le développement de notre projet MySchool se fait en utilisant plusieurs technologies Backend et Frontend. La partie Frontend est réalisé à l'aide du langage java et le système android. Le Frontend est développé principalement avec le Framework JEE et d'autres technologies web, à savoir : le framework Spring (java), Jpa, hibernate, bootstrap.

### **IDE Eclipse STS (Spring Tool Suite) :**

Spring Tool Suite (STS) est un IDE étendu pour Eclipse. Il se spécialise dans le développement des applications Spring. Il permet d'intégrer efficacement Spring dans l'environnement de développement.

## Android Studio :

Android Studio est un environnement de développement pour développer des applications mobiles Android. Il est basé sur IntelliJ IDEA et utilise le moteur de production Gradle. Il peut être téléchargé sous les systèmes d'exploitation Windows, macOS, Chrome OS et Linux4.

## Gestionnaire de dépendance Maven

Maven est un outil de gestion de projet technique. Son développement est assuré principalement par la fondation Apache. Maven permet de standardiser la forme d'un projet ainsi que son utilisation. Il permet également de gérer les dépendances d'un projet, d'effectuer une livraison complète et automatique, de déployer une application, et de faciliter le déploiement d'un projet au sein d'une plateforme d'intégration continue. Maven offre également une intégration poussée de nombreux outils de reporting (Surefire, PMD, CheckStyle, NCSS...). Il est destiné aux projets Java en général et aux projets Java EE en particulier. Il est distribué sous la licence Apache. Maven est écrit en Java. Dans notre cas, nous avons utilisé Maven pour gérer les dépendences de la partie Backend de notre projet.

## Spring Boot & Spring Initializr

Spring Boot est un framework qui facilite le développement d'applications fondées sur Spring en offrant des outils permettant d'obtenir une application packagée en jar, totalement autonome. Ce qui a beaucoup aidé dans le développement de des services web du partie Backend du projet MySchool. Pour créer rapidement un projet Spring avec toutes les dépendances nécessaires, Spring Boot fournit Spring Initializr. Il s'agit d'une interface web permettant de choisir la liste de dépendences pour démarrer un projet.

## SGBD MySQL

MySQL est un système de gestion de base de données relationnelle (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, Informix et Microsoft SQL Server. MySQL est un serveur de bases de données relationnelles SQL développé dans un souci de performances élevées en lecture, ce qui signifie qu'il est

d'avantage orienté vers le service de données déjà en place que vers celui de mises à jour fréquentes et fortement sécurisées.

## Postman pour tester les web services RESTful.

Postman est un logiciel qui se focalise sur les tests des API. EN effet, Postman est très pratique qui nous a permis de tester les services RESTful de notre projet Myschool. L permet d'envoyer et de recevoir toutes sortes de requêtes(PUT, POST, GET, DELETE, ...) sous différents formats XML, JSON, etc.

## Github pour la gestion des versions

Pour assurer le travail collaboratif, nous avons travaillé à distance via la plateforme GitHub. Il s'agit d'un logiciel de contrôle et de gestion de versions décentralisé, comme Subversion, CVS, Arch, qui utilise des dépôts décentralisés. Il permet ainsi de travailler tout en gardant une trace des modifications apportées successivement, et de retrouver ainsi l'état antérieur de ses données. Il permet également de travailler à en groupe sur un même projet.

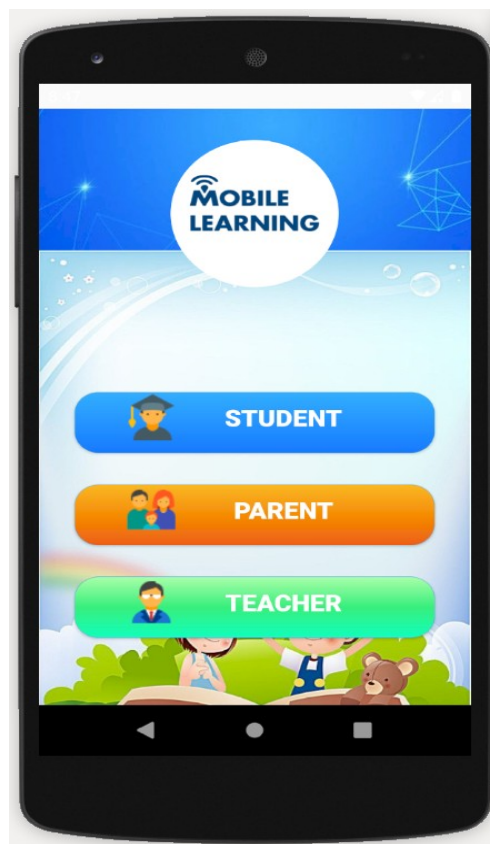
## Cloud Foundry pour déployer la partie backend du projet MySchool.

Cloud Foundry est la toute première PaaS (Plateforme en tant que Service) open source. Ces services cloud permettent d'accélérer le développement d'applications en fournissant aux utilisateurs une plateforme déjà configurée et prête à l'usage. Cette plateforme supporte les technologies Java et permet de déployer facilement des applications basées sur le Framework Spring. Il propose également plusieurs services gratuitement, comme le service de base de données pour publier les données du projet Myschool à distance.

## III. Frontend : Fonctionnalités

### Choix du profil

L'utilisateur, dans un premier lieu, devrait choisir son profil. Il y a trois profils : parent, étudiant et le professeur, comme il est montré dans la figure suivante.



## Inscription de parent et d'étudiant

 This is the 'Register' screen for a parent. It has a purple header with a back arrow and the title 'Register'. The form includes fields for:
 

- First Name: Luka
- Last Name: Martin
- Email: sebasiane.martin@gmail.com
- Phone number: 0765456765
- City: Montpellier
- Username: Luka
- Password: Luka (with a toggle to show/hide the password)

 A blue 'REGISTER' button is at the bottom, followed by a link that says 'ALREADY HAVE AN ACCOUNT? LOGIN'.


 This is the 'NEW STUDENT' registration screen. It has a blue header with the title 'NEW STUDENT'. The form is divided into two sections:
 

- Parent section:** Includes radio buttons for 'Mother', 'Father', and 'Autre'. Fields for 'Firstname' (Sebasiane) and 'Lastname' (Martin).
- Student section:** Includes radio buttons for 'boy' (selected) and 'girl'. A 'Classroom' dropdown menu set to 'Cours moyen première année (CM1)'. Fields for 'Username' (sebasiane) and 'Password' (sebasiane) with a toggle to show/hide the password.



Après l'authentification du parent ou eleve, on sauvegrade la date de sa connexion dans la base de données. Puis, On le redirige vers le une activité de bienvenue.

## Login

Afin de bien gérer la connexion de l'utilisateur à l'application, nous avons mise en place la gestion des sessions à travers SharedPreferences. Il s'agit d'une API qui permettant enregistrer une collection de données sous forme des paires clé-valeur. Cette API fournit des méthodes simples pour lire et écrire les informations stockés dans la session de l'utilisateur connecté telles que son login et son profil.

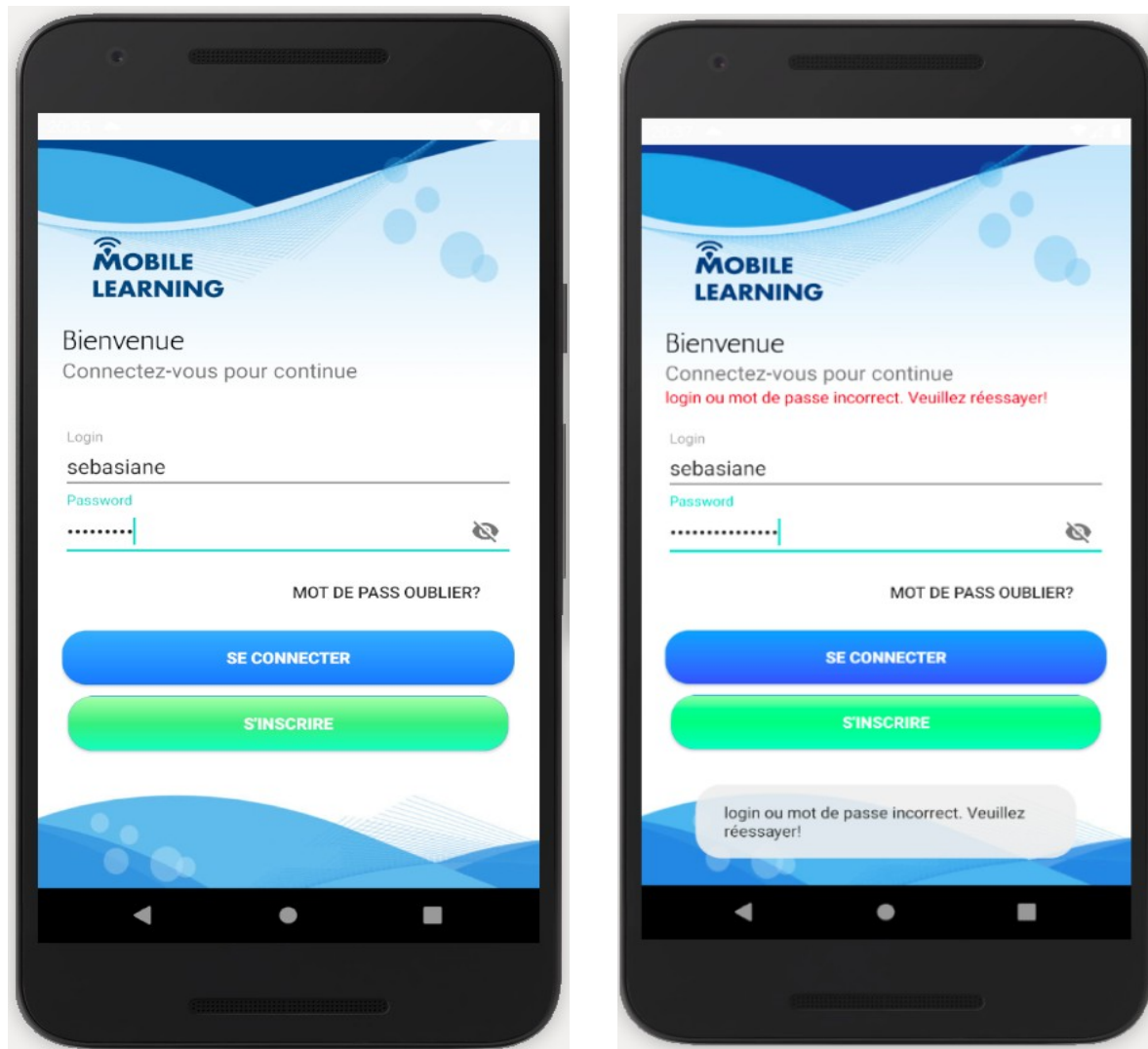
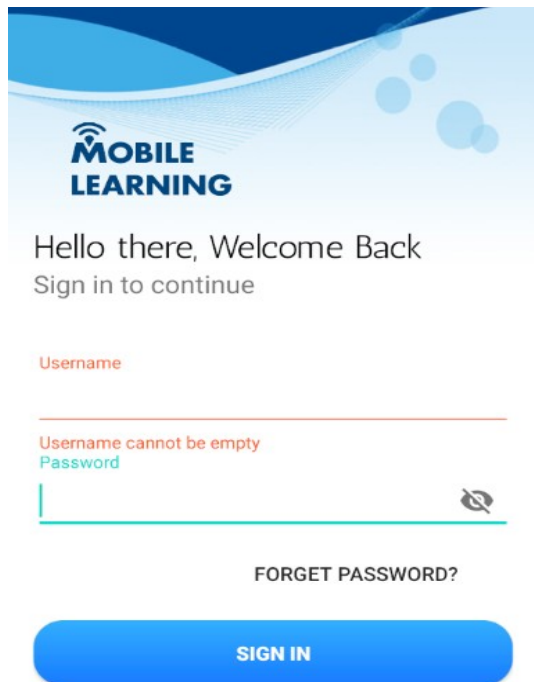


Figure: Interface de connexion

Si login ou mot de passe incorrect, un message d'erreur s'affiche qui invite l'utilisateur de réessayer de se connecter. Pour l'affichage, le message d'erreur apparaît dans dialogue box, il s'affiche également dans un textView dédié en couleur rouge.

## Validation des données saisies



**MOBILE LEARNING**

Hello there, Welcome Back  
Sign in to continue

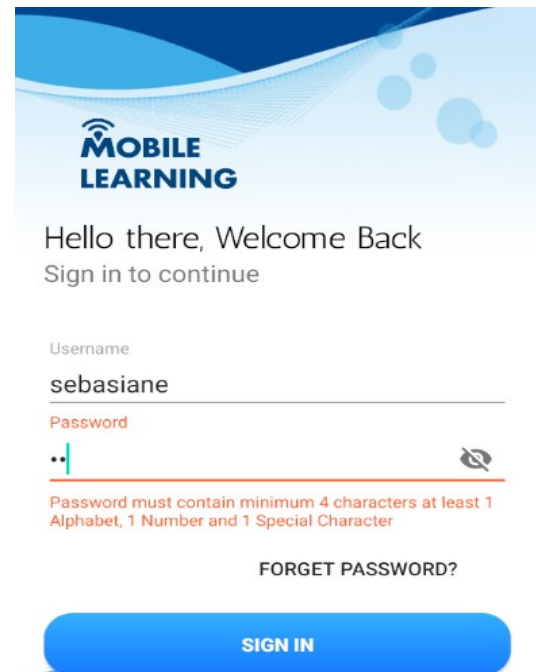
Username

Username cannot be empty

Password

FORGET PASSWORD?

**SIGN IN**



**MOBILE LEARNING**

Hello there, Welcome Back  
Sign in to continue

Username

sebasiane

Password

Password must contain minimum 4 characters at least 1 Alphabet, 1 Number and 1 Special Character

FORGET PASSWORD?

**SIGN IN**

## Récupérer le mot de passe

Cette fonctionnalité permet à l'utilisateur de récupérer son password en saisissant son adresse email.



**MOBILE LEARNING**

Récupérer le mot de passe  
Pour récupérer le mot de passe, veuillez saisir votre adresse e-mail.

Email

**ENVOYER MOT DE PASSE**

## Deconnexion d'utilisateur

Une fois l'utilisateur se déconnecte, une trace de son activité de connexion est enregistrer dans la base de données. La mise à jour de date de déconnexion se fait au niveau de la table log\_session.

```
mysql> select * from log_session;
+-----+-----+-----+-----+
| id | date_connexion          | date_deconnexion | user_id |
+-----+-----+-----+-----+
| 15 | 2020-04-10 16:09:59.019000 | NULL             | 46      |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select * from log_session;
+-----+-----+-----+-----+
| id | date_connexion          | date_deconnexion          | user_id |
+-----+-----+-----+-----+
| 15 | 2020-04-10 16:09:59.019000 | 2020-04-10 16:10:19.203000 | 46      |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

## Online school registration Boîte de réception x



**karim.dahdouh.fr@gmail.com**

15:35 (il y a 0 minute)



À moi ▾

Hi test9, thank you for your registration in our online school courses. In order to activate your account please click in this link:  
<http://localhost:8085/activateAccount/71>

## Activation compte par mail.

Une fois l'utilisateur(élève ou parent) clique sur le lien d'activation, il va être rediriger vers la page d'activation. Un message d'activation du compte sera affiché à l'utilisateur. L'interface d'application web permettant de gérer les services RESTful du projet MySchool .



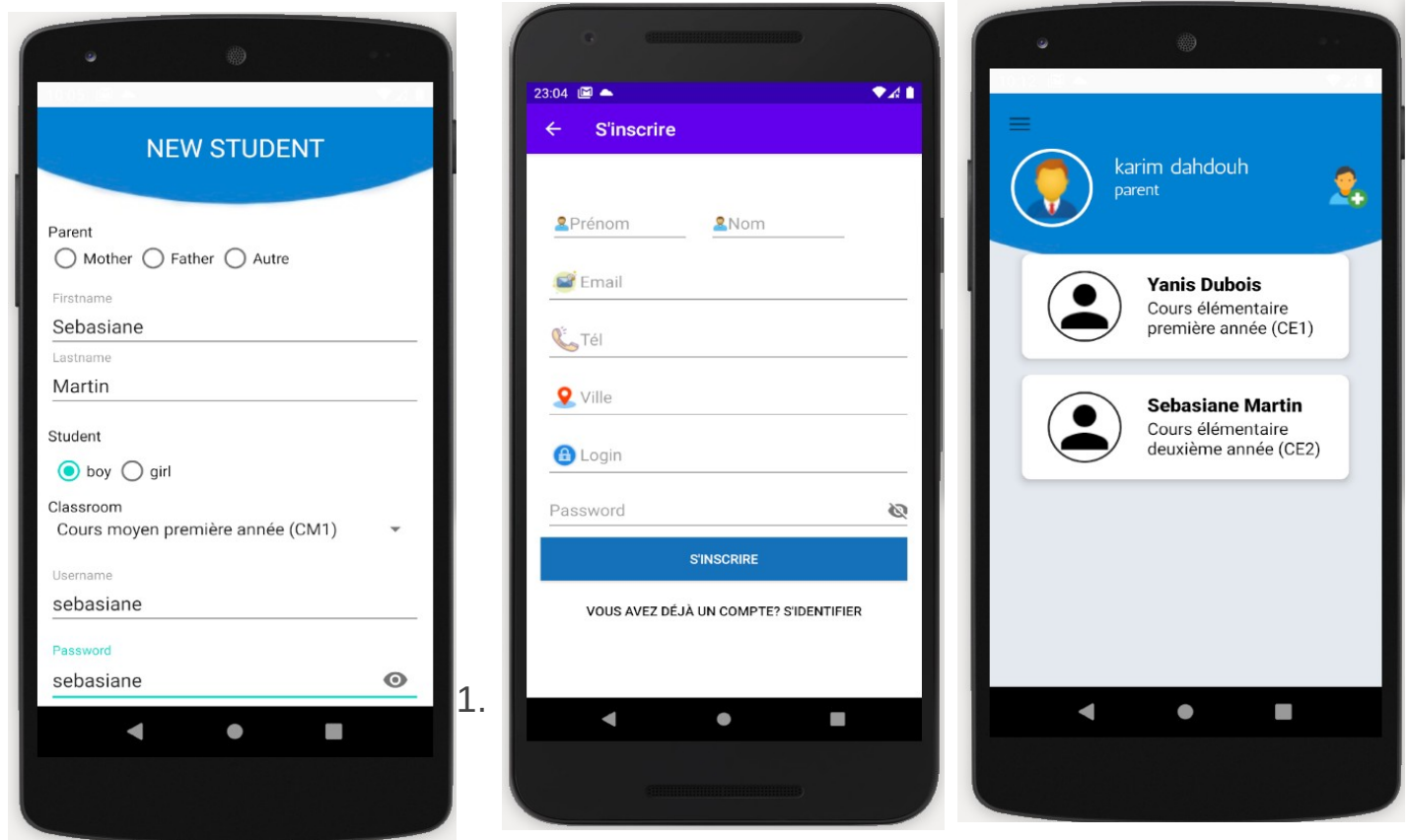
Activation account

You account has been successfully activated

© 2020 Karim Dahdouh & Theirno Barry-Tous droits réservés

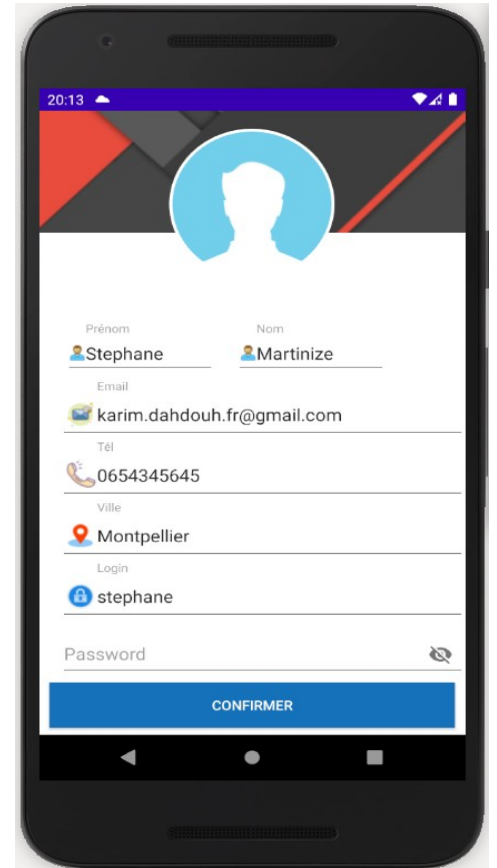
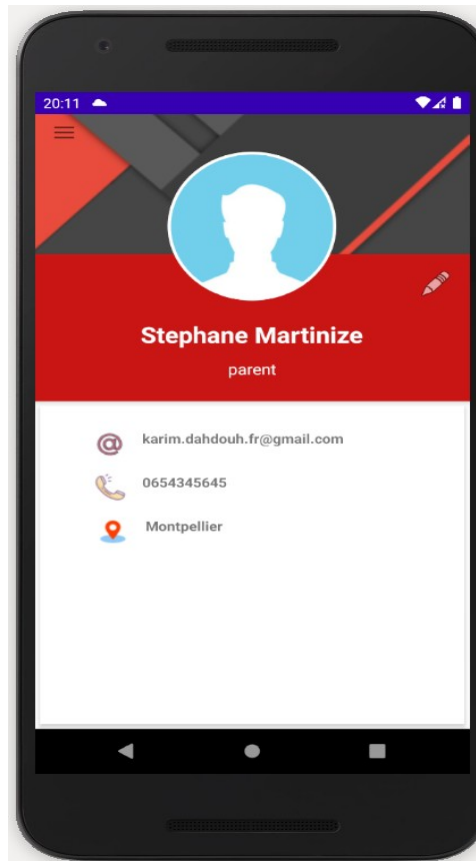
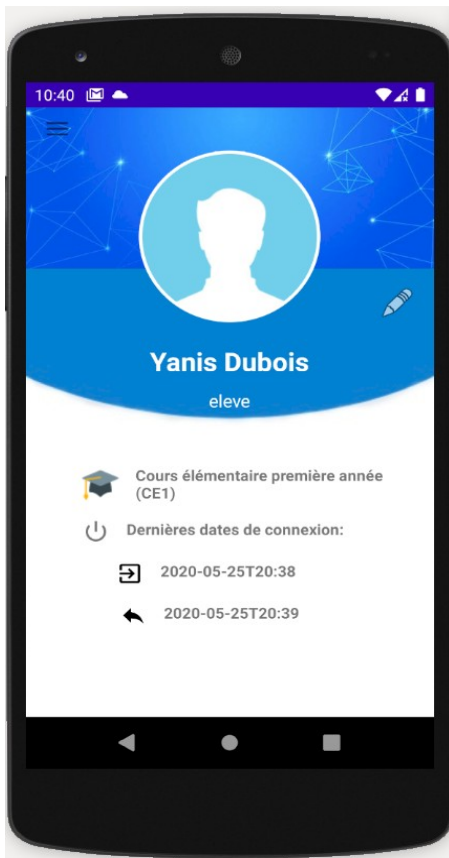
## Création des profils : élève et Parent

Les parents à le possibilité de créer des comptes pour leurs enfants. Pour cela, un parent doit fournir les informations de sons fils/fille (nom, prénom, niveau scolaire, login et password). La figure suivante (à gauche) présente l'activité « StudentAdd » et celle de droite représente la créaion du profil parent, et la dernière montre la liste des élèves créé par un parent.



La liste des élève créent par un parent seront affichés dans le dashboard du profil du parent, comme présente la figure ci-dessus(à droite).

## Profil d'utilisateur connecté



L'étudiant ou parent peut voir son profil, ainsi que de le modifier via le menu. En plus des informations personnelles comme nom, prénom etc, le parent peut consulter les horaires de connexion et déconnexion de son élève.

## Gestion des cours

Comme indiqué dans les premières parties du rapport, nous avons utilisé le web service RESTful pour implémenter notre serveur car nous essayons de nous adapter au mieux à ce qui se passe en entreprise en utilisant les bonnes pratiques. Notre Application Android est donc un consommateur des différentes APIs que nous faisons avec la technologie java plus particulièrement avec son extra-ordinaire Framework spring.

Nous allons donc commencer par les captures des données qu'on a enregistrées dans notre

```
mysql> use myschool
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from course;
+-----+-----+-----+-----+-----+-----+
| id | libelle | progress | level | description | image |
+-----+-----+-----+-----+-----+-----+
| 1 | Maths | 0% | NULL | Cours de maths avec plusieurs chapitres: algèbre, arithmétique, etc | https://image.shutterstock.com/image-ve |
| 2 | Français | 0% | NULL | Cours de Français avec plusieurs chapitres: Grammaire, Orthographe etc | https://t3.ftcdn.net/jpg/01/28/45/16/24 |
| 3 | Physique | 0% | NULL | Cours de Physique avec plusieurs chapitres: Electricité, Mécanique etc | https://www.studyrama.com/IMG/arton8928 |
| 4 | Chimie | 0% | NULL | Cours de Chimie avec plusieurs chapitres: les composants chimiques, réactions etc | https://cdn.futura-sciences.com/buildsv |
| 5 | Anglais | 0% | NULL | Cours d' Anglais avec plusieurs chapitres: Grammaire, Speaking, reading etc | https://vraiesecolesdelangues.com/wp-co |
| 6 | Histoire | 0% | NULL | Cours d'Histoire avec plusieurs chapitres: Guerres mondiales, crise des années 30 etc | https://i.f1g.fr/media/eidos/805x453_cr |
| 7 | Géographie | 0% | NULL | Cours de Géographie avec plusieurs chapitres: la démographie mondiale etc | https://www.geodus.com/globe-map/tresgr |
| 8 | Biologie | 0% | NULL | Cours de Biologie avec plusieurs chapitres: les plantes, la reproduction etc | https://www.pr.f.upol.cz/typo3temp/asset |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.01 sec)

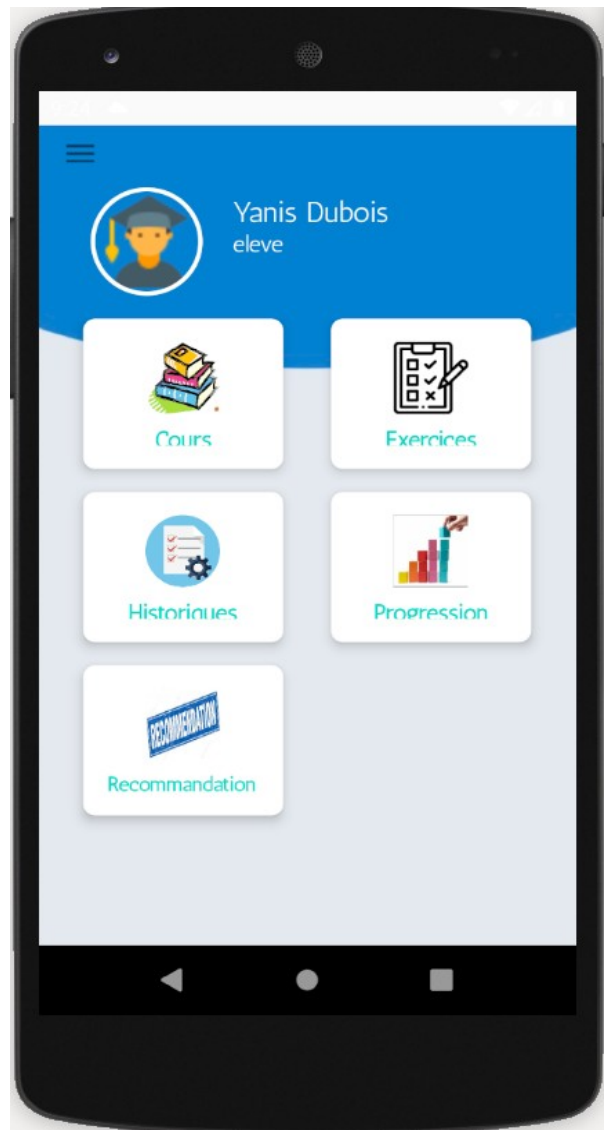
mysql>
```

base de données. La capture est un peu trop grande vu qu'on voulait capture tout son contenu et que les liens de nos données(images, vidéos) soient longs. Nous pouvons donc voir dans cette capture la liste de huit(8) cours que nous avons ajouté et qui seront présents dans l'application.

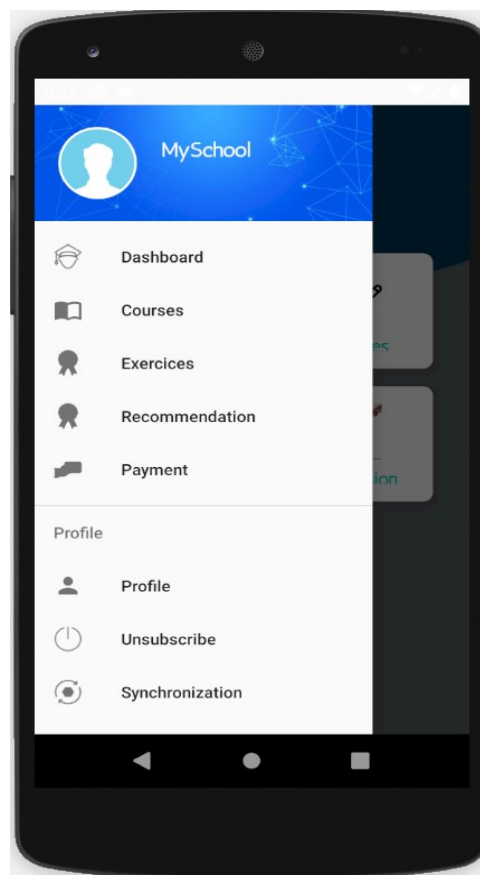
La capture suivante concerne les contenus de ces différents cours en vidéos et en pdf . Comme vous pouvez le constater, nous voulons rendre notre application le plus réaliste que possible en utilisant des données externes vu que notre service web est déployé en ligne .

```
mysql> select * from course_content;
+-----+-----+-----+-----+-----+
| id | path | title | course_id | type |
+-----+-----+-----+-----+-----+
| 1 | http://www.ancestralauthor.com/tutorial/tutorial.pdf | premier cours | 1 | 1 |
| 2 | http://www.ancestralauthor.com/download/large.pdf | deuxième cours | 1 | 1 |
| 3 | http://www.ancestralauthor.com/download/sample.pdf | troisième cours | 1 | 1 |
| 4 | http://www.ancestralauthor.com/tutorial/tutorial.pdf | Français élémentaire | 2 | 1 |
| 5 | https://webusers.imj-prg.fr/~alessandro.chiodo/LU1MA002/exomov/F05E06.mp4 | calcul fréquentiel | 1 | 2 |
+-----+-----+-----+-----+-----+
5 rows in set (0.05 sec)
```

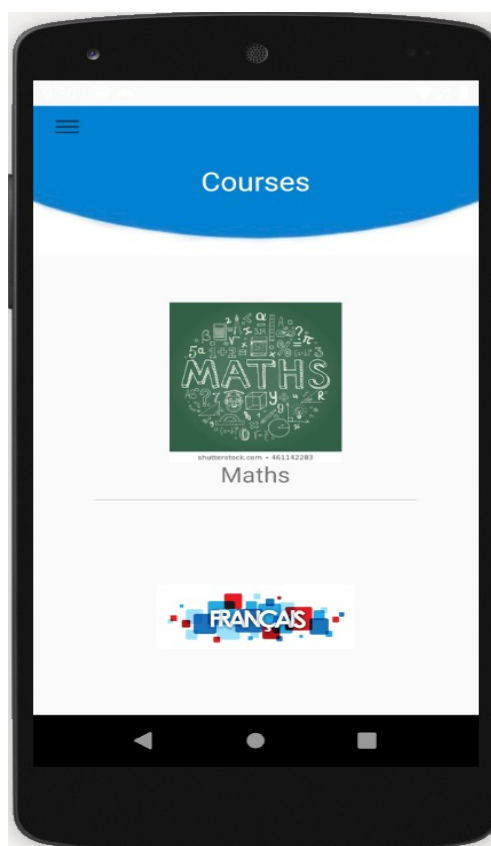
Nous allons maintenant commencer les captures de l'interface utilisateur(UI). La capture suivante est la première version de notre page d'accueil. c'est pas visible sur la capture mais il y a une petite animation avant le lancement de la page.



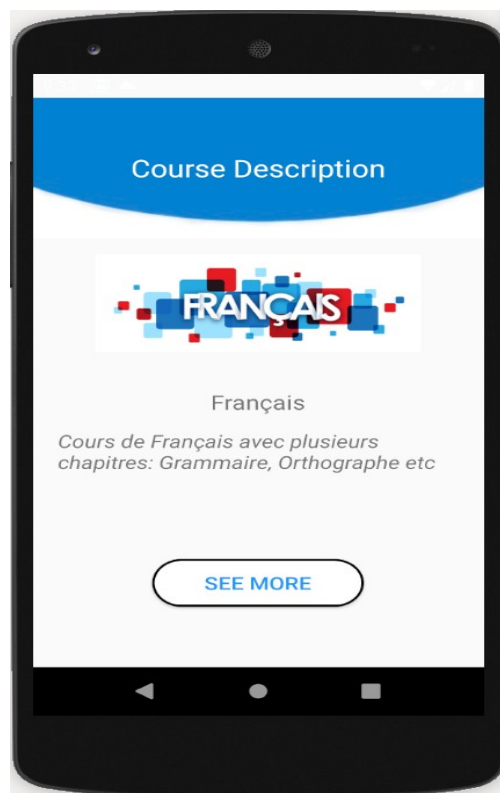
Nous avons en suite notre menu de gauche



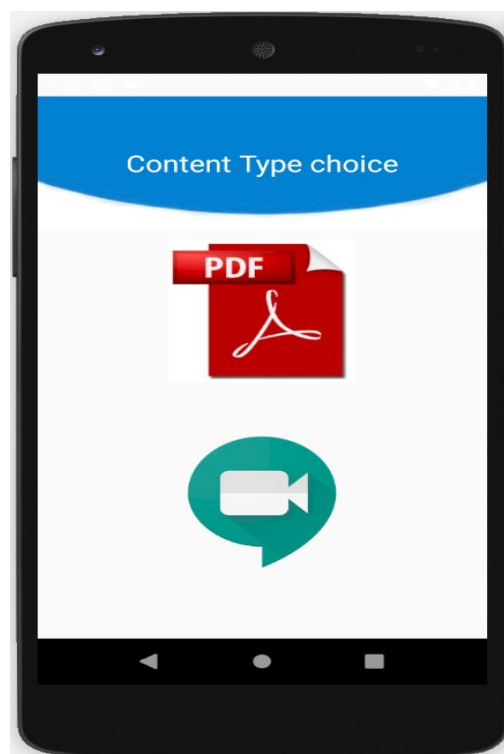
Nous pouvons donc aller sur les captures qui nous intéressent dans cette partie à savoir les cours . cliquons donc sur l'item courses de notre menu.



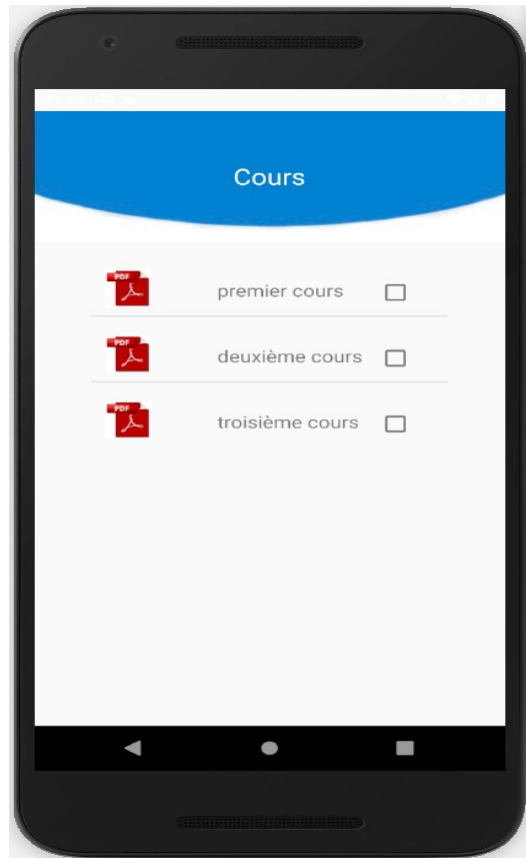
Nous pouvons voir sur la capture la liste de nos cours qui nous viennent de notre API et qui tirent leur source de notre base de données où nous pouvons retrouver leur liens complets. Nous allons choisir un cours pour lequel on a déjà enregistré quelques cours en pdf et vidéo . Quand c 'est notre première fois nous avons cette activité qui s'affiche avec l'intitulé et les détails de la matière puis un bouton pour s'inscrire afin qu'on puisse accéder aux cours et exercices



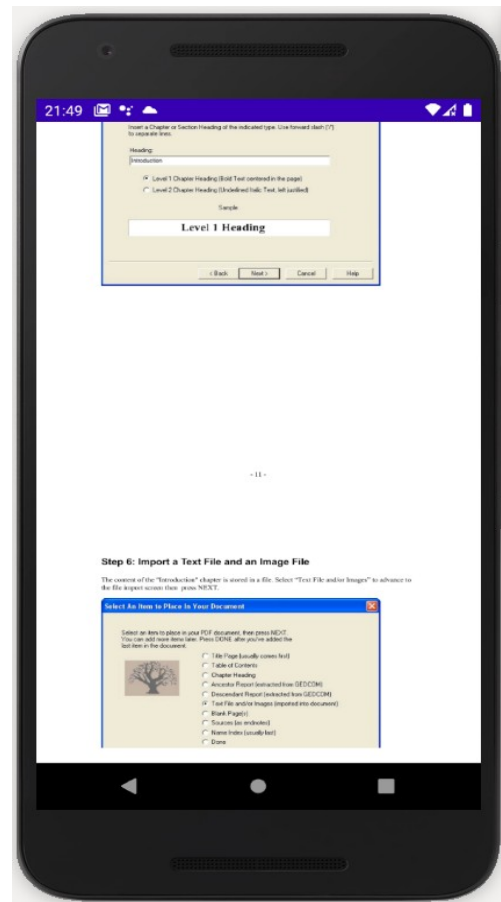
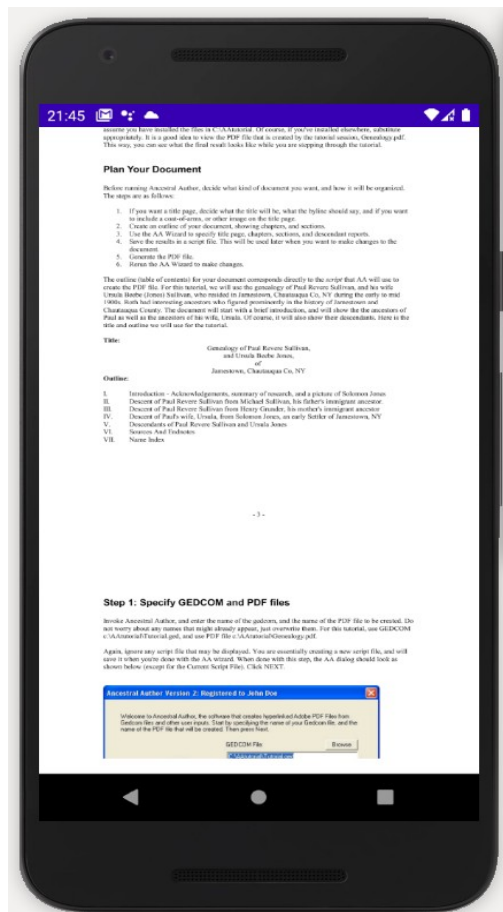
Nous allons passer maintenant au choix du type de cours qu'on veut consulter à savoir pdf ou vidéo.



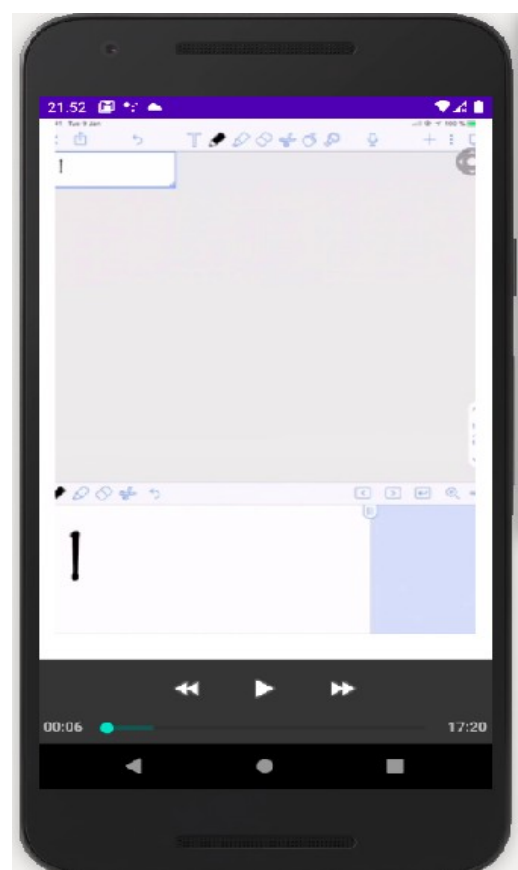
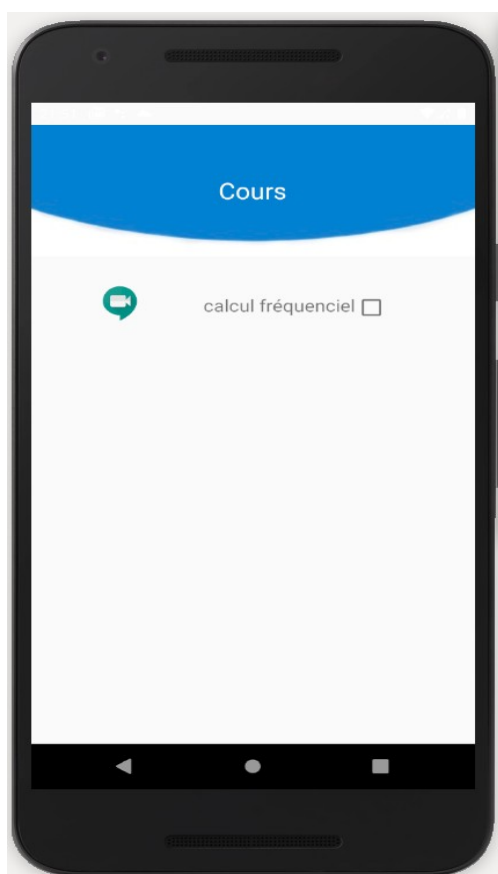
Dirigeons nous vers les cours PDFs en premier lieu . Nous avons ici la liste de nos cours en pdf et une case à cocher qui sera utilisée prochainement pour la progression de l'élève dans un cours donné.



Nous allons maintenant lire un cours PDF. Pour lire les fichiers pdf on a fait une injection de dépendance que vous retrouverez dans le code. Dans ces deux capture nous voyons un fichier pdf(qu'on a pris sur internet) ouvert.



Nous allons maintenant nous diriger vers les cours vidéos. Nous avons un seul cours vidéo pour l'instant.

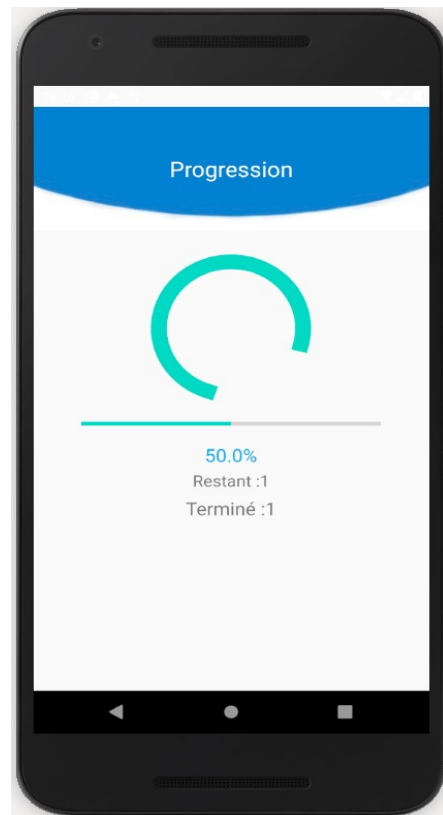
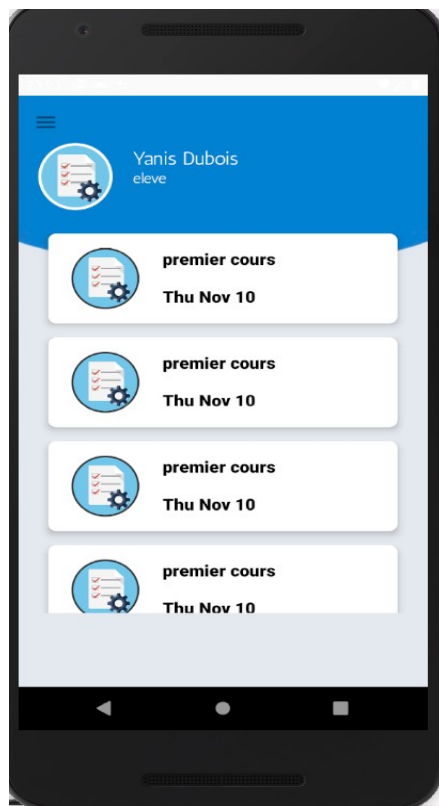




Passons maintenant à sa lecture. Nous avons ici la lecture de notre vidéo qui nous vient aussi de l'externe comme toutes autres données. La figure ci-dessus à droite.

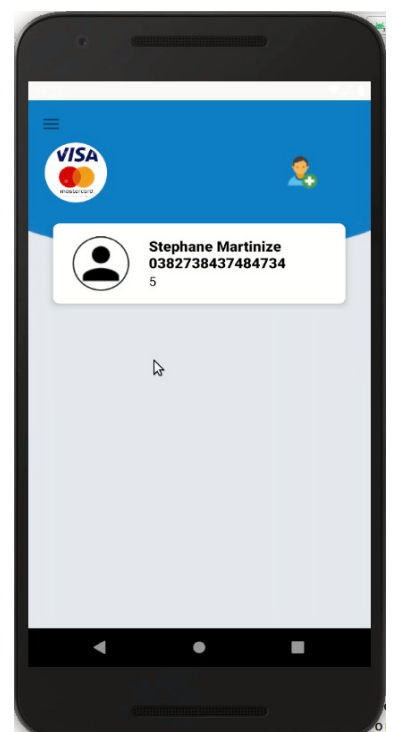
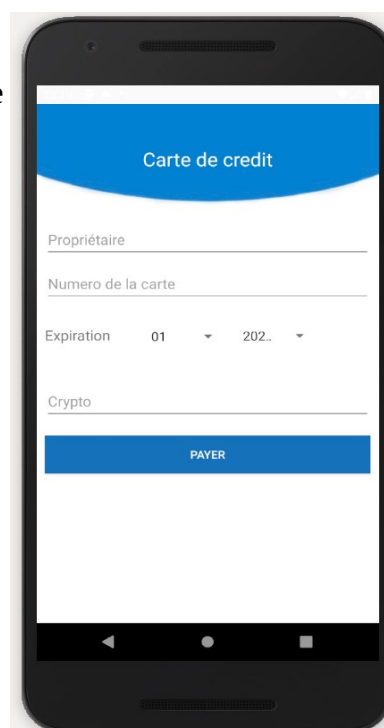
## Historiques des cours

Aussi l'interface permettant d'observer la progression d'un élève dans une matière donnée en fonction des cours et exercices.

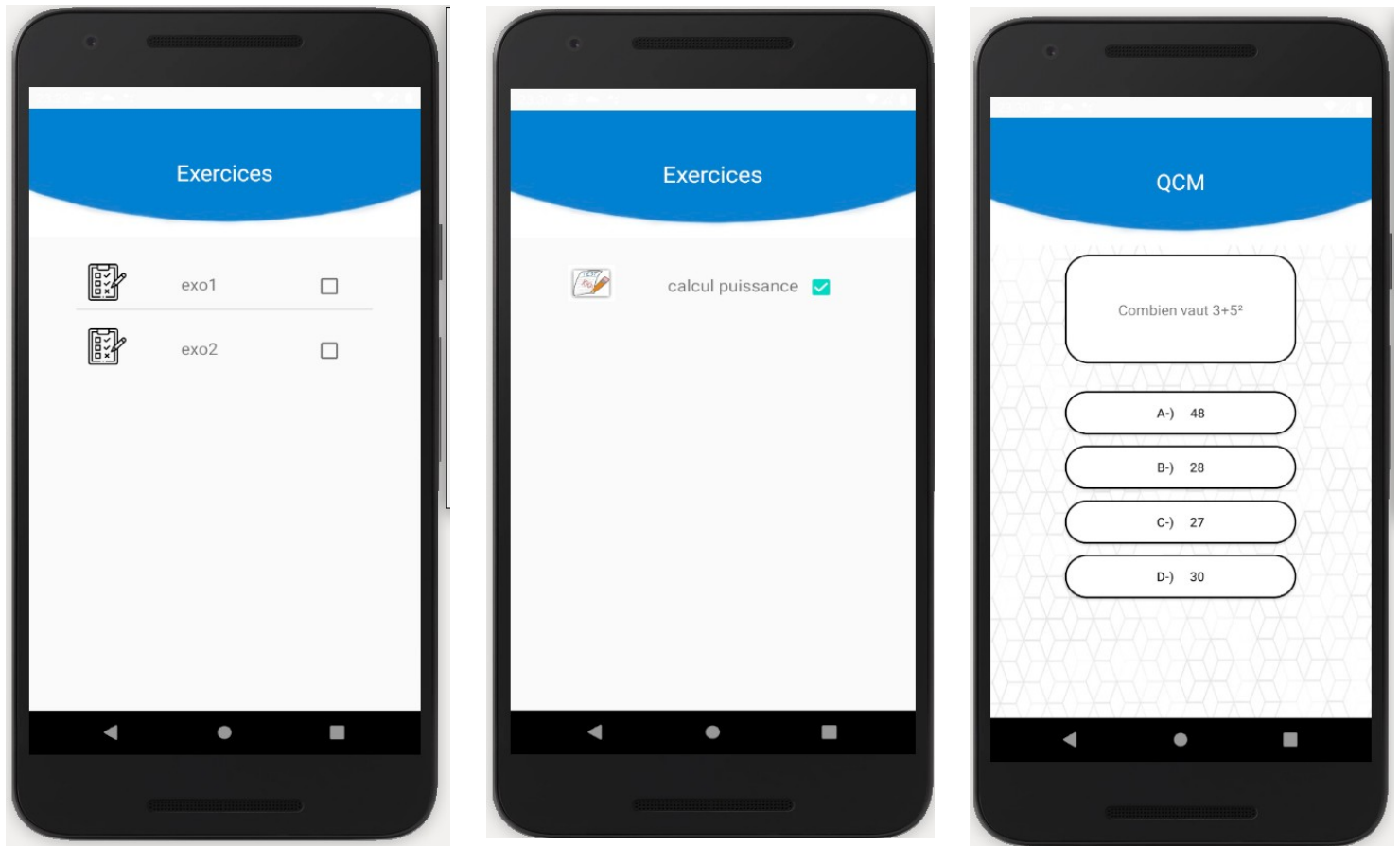


## Paieement

Nous avons ensuite l'enregistrement d'une carte de crédit afin qu'un élève puisse s'acquitter de ses droits concernant l'abonnement.

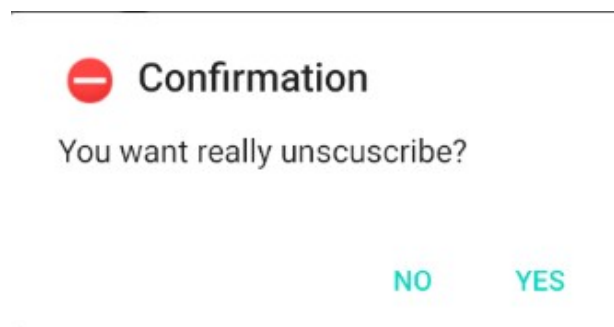


## Gestion des Exercices :



## Desinscription

Nous avons développé un processus de désinscription permettant à l'utilisateur (étudiant ou parent) de désinscrire de l'application. Cela se fait après la confirmation du message d'information qui demande à l'utilisateur s'il vraiment souhaite désinscrire de notre application (ce dialogue box sera amélioré dans la dernière version).



## Recommandation

La fonctionnalité de recommandation notifie les utilisateur en fonction de son mode d'apprentissage. Elle sert également comme un assistant pour aider et encourager d'utilisateur durant l'activité d'apprentissage.

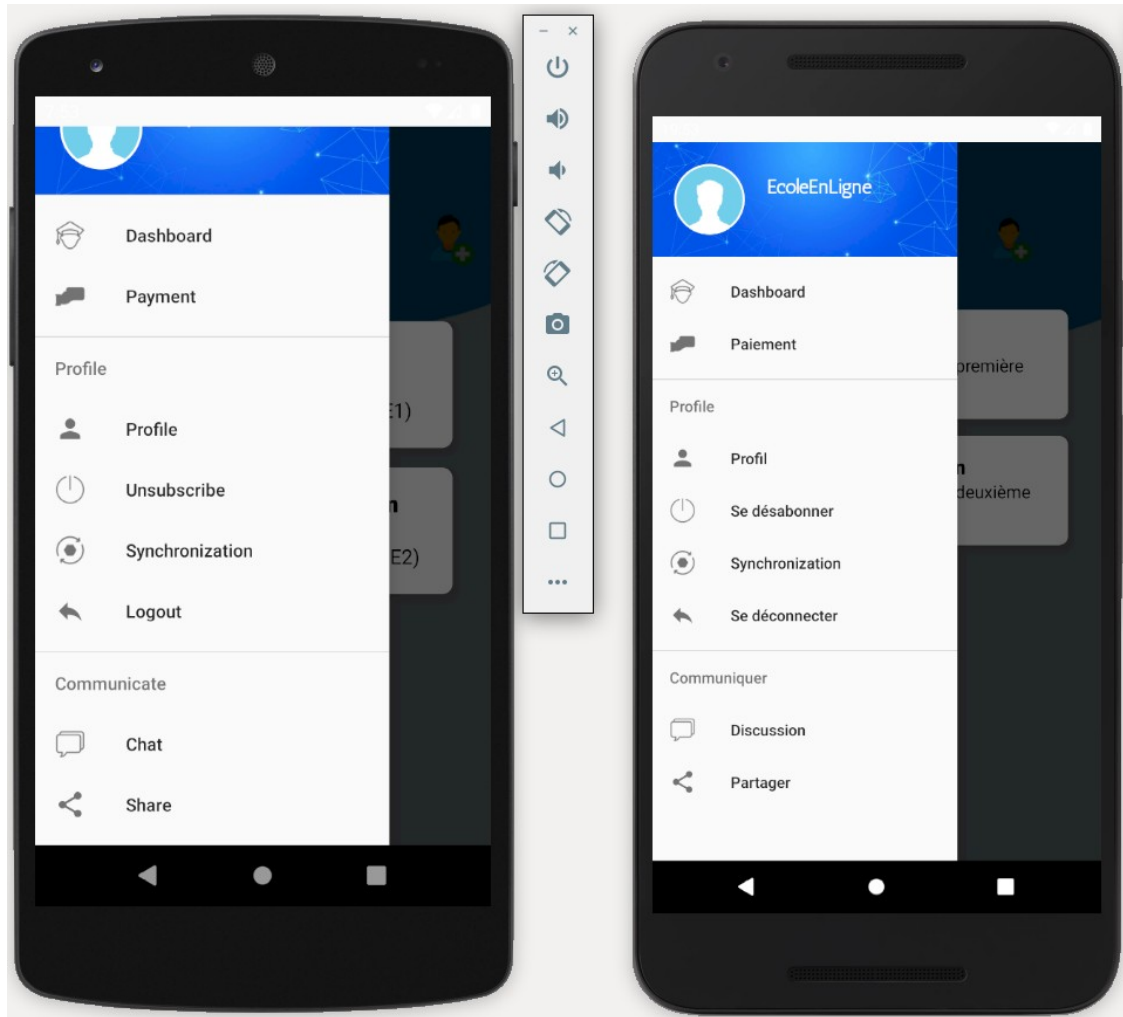


La fonctionnalité de recommandation notifie les utilisateur en fonction de son mode d'apprentissage. Elle sert également comme un assistant pour aider et encourager d'utilisateur durant l'activité d'apprentissage.

## Internationalisation

Afin d'adapter notre application mobile à différentes langues, nous avons mis en place la technique l'internationalisation. Cette technique permet de rendre notre application disponible dans plusieurs langues, en ajoutant simplement un fichier de traduction.

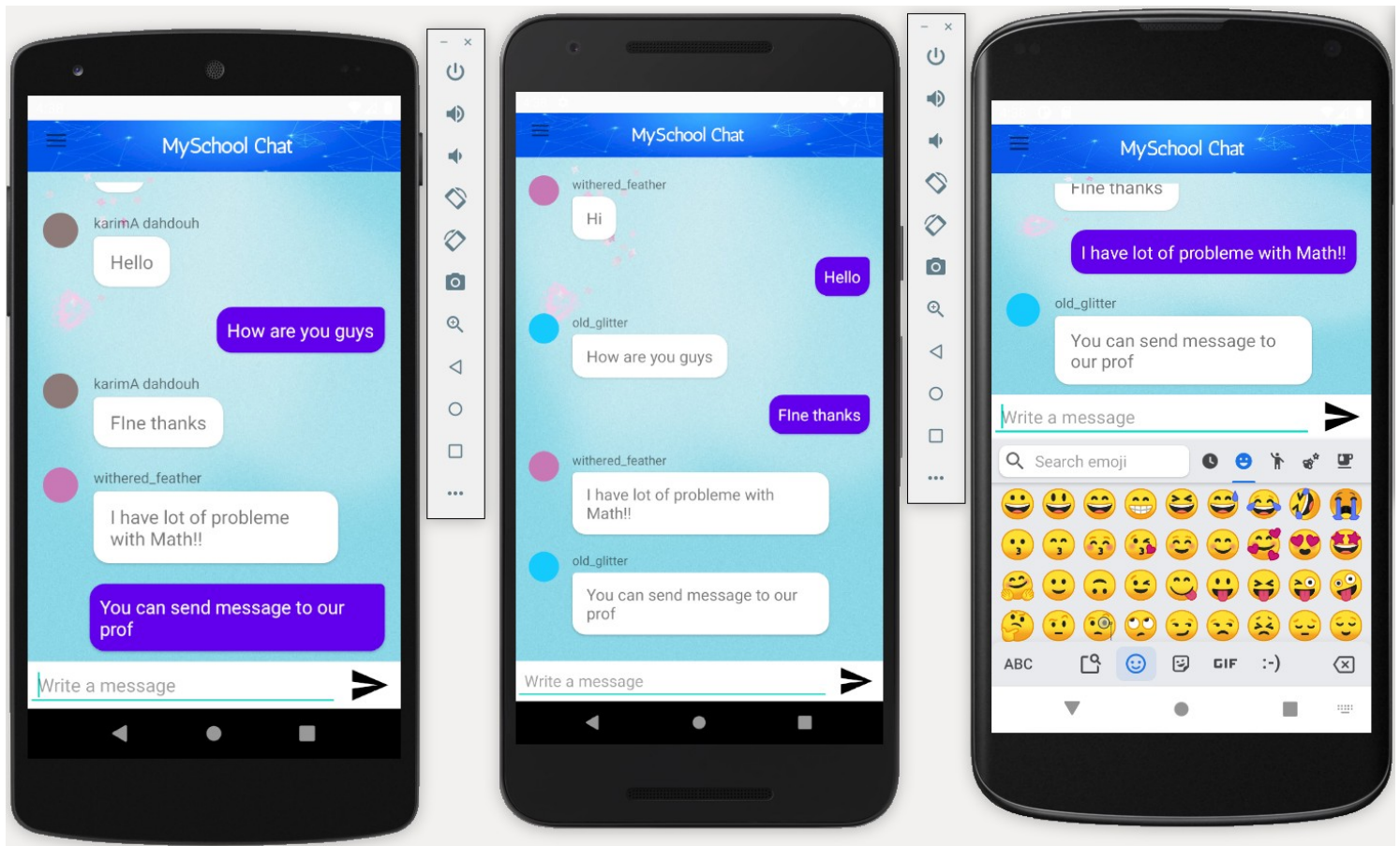
Jusqu'à l'instant le projet MySchool a été traduit en deux langues, qui sont le français et l'anglais. Nous envisagerons d'intégrer d'autres langues dans les prochaines versions.



La figure ci-dessus présente les deux versions du projet MySchool. La première version est traduit en français, et la seconde en anglais. Il suffit juste changer la langue du système android de l'utilisateur connecté pour que l'application s'adapte automatiquement selon le profil utilisateur.

## Chat en temps réel

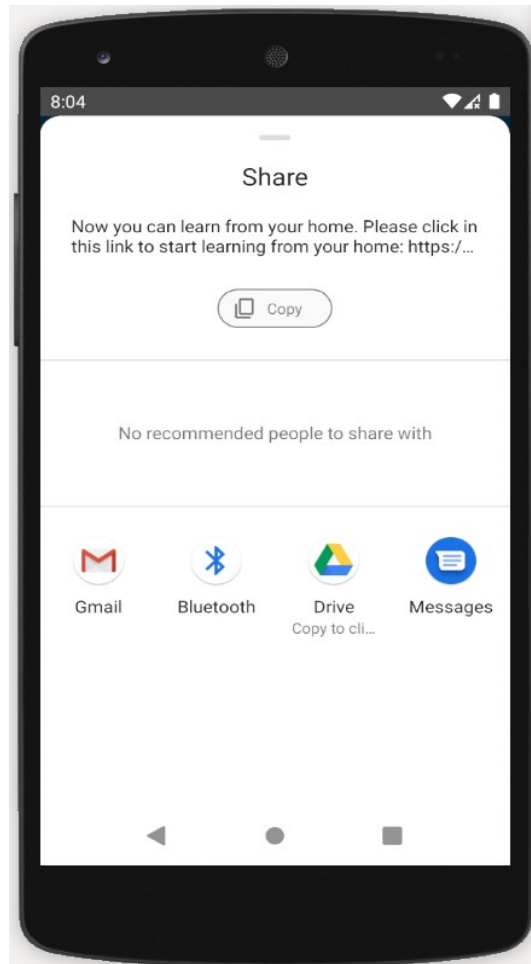
Notre version de chat à été faite en n'utilisant le service Scaledrone. Il s'agit d'une plateforme de messagerie en temps réel permettant de créer des salons (rooms) de discussion et des outils collaboratifs. Elle supporte plusieurs langages de programmation, y compris Java, JavaScript, Python, PHP, Go, etc. Dans un premier temps, nous avons créé un compte Scaledrone. Puis, nous avons s'abonné à un salon de discussion.



Ces figures illustrent une discussion entre trois étudiants dans un salon de discussion. Les messages envoyés par chaque étudiant apparaissent en temps réel dans les smartphones des autre étudiants connectés au même room.

## Partage

Cette fonctionnalité est accessible via le menu princpiale.



La fonctionnalité de partage permet partager le lien de notre application sur différents moyens, y compris Gmail, Messenger, Bluetooth, Message, Drive, etc. Le message partage, change selon la langue de de l'appareil d'utilisateur.

EcoleEnLigne



Boîte de réception x



**school online** <onlineschool606@gmail.co... 00:01 (il y a 0 minute)

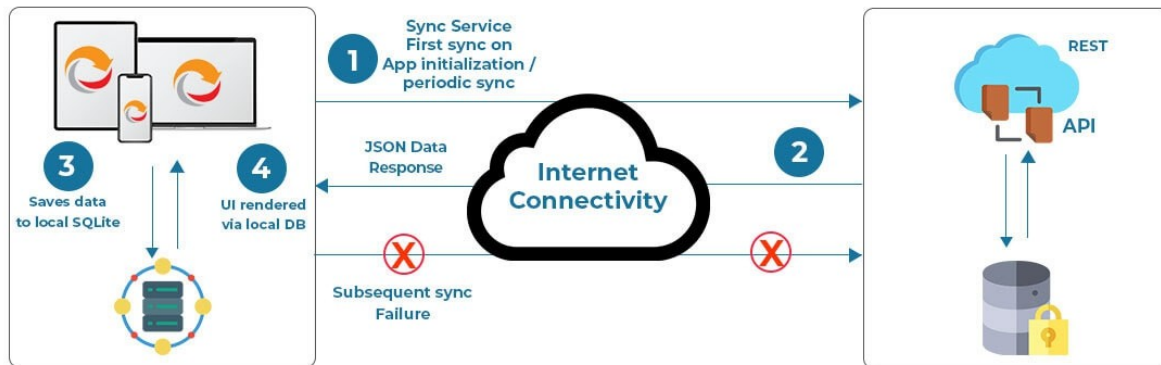


À moi ▼

Vous pouvez maintenant apprendre de chez vous. Veuillez cliquer sur ce lien pour commencer l'apprentissage: : <https://github.com/dahdouh/android/tree/master/APK>

## IV. Version offline et synchronisation

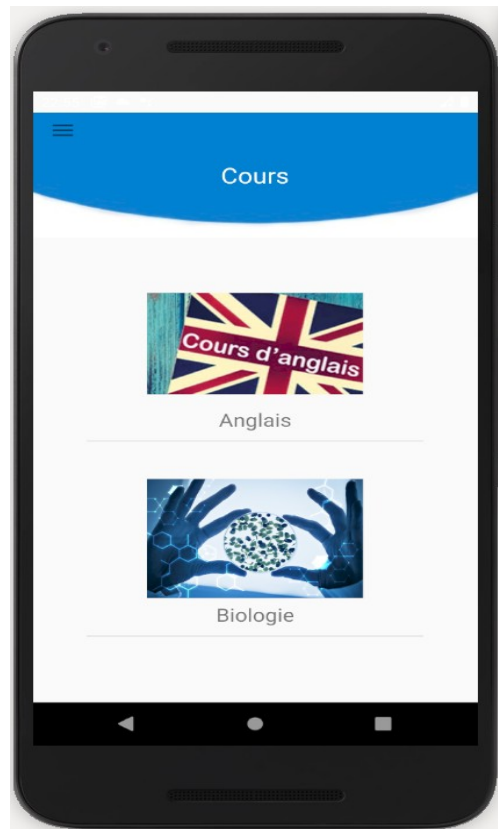
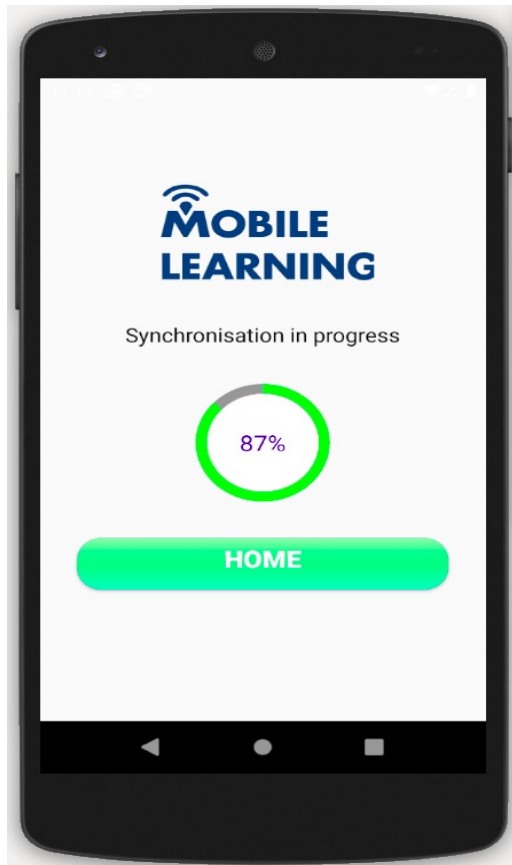
Afin de permettre à l'utilisateur travailler avec notre application dans toutes les conditions, avec ou sans raison disponible, nous avons développé une version offline de celle-ci. Il s'agit de mettre en place un service de synchronisation permettant de synchroniser les données entre l'appareil utilisateur et le serveur de façon asynchrone. Le schéma suivant récapitule brièvement le processus de synchronisation.



Pour que notre application fonctionne bien mode hors connexion, nous avons créé une base de données locale en utilisant SQLite. L'idée derrière cette base de données locale est de faire une copie de la base de données distante et de l'utiliser dans le cas de l'absence de connexion, puis de transférer les données via les webservices RESTful une fois que la connexion est disponible.

Généralement, la synchronisation se fait automatiquement, par contre l'utilisateur peut également l'effectuer explicitement via le menu « Synchronisation ». Dans ce cas, l'utilisateur sera redirigé vers une activité qui affiche le pourcentage de synchronisation de données. (voir la figure ci-dessous).





La deuxième figure (à droite) montre que même si il n'y a pas de connexion dans l'appareil mobile l'application fonctionne bien et n'affiche aucune erreur et nous ramène la liste des cours à partir de notre base de données locale sqllite.

## V. Backend:RESTful webservice (framework spring)

Afin de tester les service web du projet MySchool, on a utilisé Postman, comme il est montré ci-dessous.

### Webservice : Login ou authentification

Ce webservice permet aux utilisateurs de l'application MySchhol de se connecter au serveur en fournissant un login et un mot de passe.

### Webservice: Inscription

Il permet aux nouveaux utilisateurs de s'inscrire aux application MySchool en remplissant les informations telles que prénom, nom, email etc. Ce web service permet de sauvegarder les données de l'utilisateur au niveau de bases de données.



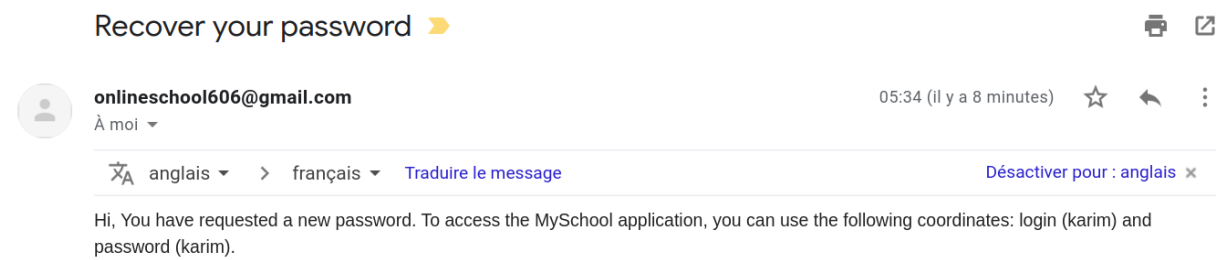
```
1 {
2   "id": 91,
3   "firstname": "sebasiane",
4   "lastname": "bounnal",
5   "email": "test323@gmail.com",
6   "facebook": "facebook",
7   "twitter": "twitter",
8   "level": null,
9   "parent": null,
10  "parentRelation": null,
11  "compte": {
12    "id": 66,
13    "login": "logintest",
14    "password": "passtest",
15    "activate": "no",
16    "profile": {
17      "id": 1,
18      "libelle": "parent",
19      "role": "ROLE_PARENT"
20    }
21  },
22  "courses": [],
23  "recommendations": null
24 }
```

## webservice: Mot de passe oublier

On a développé un service afin de permettre à l'utilisateur de récupérer son mot de passe au cas ou il a oublié.

```
1 {
2   "id": 81,
3   "firstname": "karim",
4   "lastname": "dahdouh",
5   "email": "karim.dahdouh.fr@gmail.com",
6   "facebook": "test face",
7   "twitter": "test twitter",
8   "level": null,
9   "parent": null,
10  "parentRelation": null,
11  "compte": {
12    "id": 56,
13    "login": "karim",
14    "password": "karim",
15    "activate": "yes",
16    "profile": {
17      "id": 1,
18      "libelle": "parent",
19      "role": "ROLE_PARENT"
20    }
21  },
22  "courses": [],
23  "recommendations": []
24 }
```

Le mot de passe est envoyé à l'adresse email d'utilisateur.



## webservice: Déconnexion

Il permet à un utilisateur connecté à l'application MySchool de se déconnecter. Une mise à jour de date de déconnexion se fait au niveau de la table log\_Session pour garder la trace de la session d'utilisateur.

## VI. Déploiement

Le déploiement de des webseices à été fait à l'aide de plate-forme open source Cloud Foundry. Il s'agit d'un PaaS open source qui permet de créer, de déployer, d'exécuter et de faire évoluer des applications sur des modèles de Cloud public et de Cloud privé. Cloud Foundry a été créé à l'origine par VMware et appartient désormais à Pivotal Software.

On a, tout d'abord, créé une base de données de type ClearDB MySQL Database, permettant de manipuler données de nos services web. Pour cela, on un service dédié appelé « myschool\_db » comme montre la figure suivante :

The screenshot shows the Pivotal Web Services console interface. On the left is a sidebar with navigation links: Space, App (1), Service (1), Member (1), Settings, Networking, Route (1), and Marketplace. The main content area displays the 'development' space. At the top, there's a search bar and a notification banner stating: 'Info: Your org, "myonlineschool", is still in trial. To get access to 25GB of memory and paid service plans, [upgrade now](#)'. Below the notification, the 'Services' section shows a table with one service: 'ClearDB MySQL Database' with name 'myschool\_db', 1 bound app, and a 'free - Spark DB' plan. The 'Last Operation' is 'create succeeded'. A status bar at the top right shows 'RUNNING 1', 'STOPPED 0', and 'DOWN 0'. At the bottom, there's a footer with 'Pivotal' logo, copyright information, and links for Tools, Support, and other resources.

Service	Name	Bound Apps	Plan	Last Operation
ClearDB MySQL Database	myschool_db	1	free - Spark DB	create succeeded

Ensuite, on a déployé la partie backend du projet MySchool. L'objectif est de mettre en ligne les webservice MySchool pour nous permettre de travailler en équipe sur la même version. Cette version contient les web services nécessaires au bon fonctionnement de l'application mobile MySchool. Il est accessible via le lien :<https://onlineschool.cfapps.io/>. La figure suivante donne un aperçu du déploiement de l'application jee.

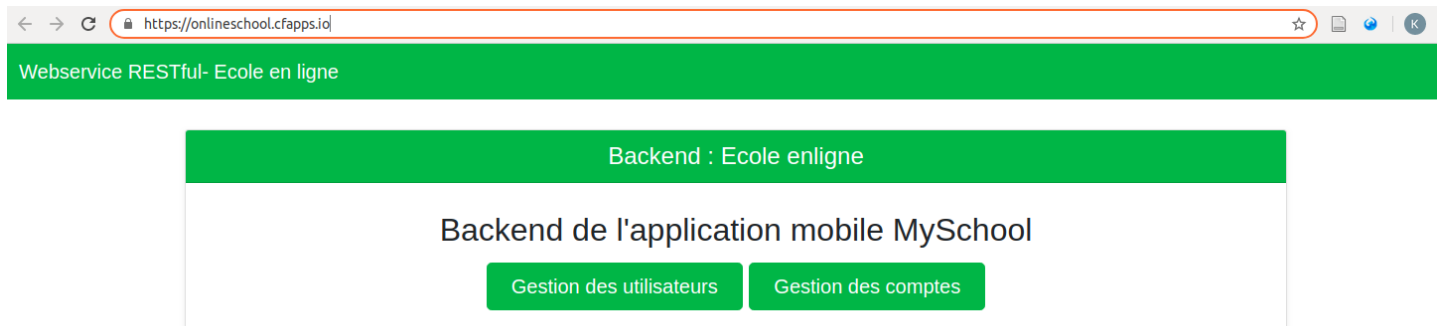
This screenshot shows the Pivotal Web Services console with the 'Apps' section selected. The 'onlineschool' app is listed as 'RUNNING' with 1 instance and 1 GB of memory. The 'Route' column shows the URL 'https://onlineschool.cfapps.io/'. The interface is similar to the previous screenshot, with the same sidebar and notification banner. The footer also contains the Pivotal logo and various links.

Status	Name	Instances	Memory	Last Update	Route
RUNNING	onlineschool	1	1 GB	9 min	<a href="https://onlineschool.cfapps.io/">https://onlineschool.cfapps.io/</a>

## Consommation des webservices RESTful coté Android

Pour appeler les webseices RESTful, nous avons employé une API spécifique appelée Volley. En effet, Volley est une librairie HTTP qui permet facilement de faire des appels réseaux rapides sur Android. Elle utilise les méthodes GET / POST / PUT / DELETE du protocole HTTP et est particulièrement efficace pour des téléchargements courts et rapides tels que le téléchargement d'images ou de données utilisateurs. Volley permet aux applications android communiquer avec un service web RESTful en JSON.

Page d'accueil :



# Liens du projet MySchool

## Version en ligne des webservices RESTful

- ◆ Les web services RESTful: <https://onlineschool.cfapps.io/>

## Github

Les liens github suivants contiennent l'APK et le code source complet de des parties Backend et Frontend du projet MySchool:

## Video :

démo : drive

<https://drive.google.com/file/d/1fyxssj5k4J739ukBUDXA9sZVvagAwcUV/view?usp=sharing>

démo : Youtube

[https://youtu.be/-2\\_LlqECv6w](https://youtu.be/-2_LlqECv6w)

APK :

<https://github.com/dahdouh/android/tree/master/APK>

Frontend : application android

<https://github.com/dahdouh/android/tree/master/EcoleEnLigne>

Backend : Webservices RESTful (Spring)

<https://github.com/dahdouh/spring/tree/master/MySchool>

Déploiement des webservices en ligne :

<https://onlineschool.cfapps.io/>