

Projet d'option MyScript

-

SYNTHESE

Présentation de l'application

« Mes petits exercices »



Tuteur Entreprise : Anthony Laurence

Tuteur Ecole : Xavier Lorca

Equipe :

Ali BELGHITI

Matthieu POSMYK

Thibault ROMAAN

Aurélien ZAKOWIC

Lien Github du projet :

<https://github.com/aurelienzakowic/ProjetOptionMyscript>

TABLE DES MATIERES

I – Présentation du contexte

I – a – L’entreprise MyScript

I – b – La mission

I – c – Analyse des contraintes et choix de l’application à développer

II – Présentation de l’application « Mes petits exercices »

II – a – Introduction

II – b – La partie « Application »

II – c – L’architecture Client-Serveur pour la phase de collecte des données

II – c – i) Motivation

II – c – ii) Présentation des technologies utilisées

II – c – iii) Etude sur un exemple

II – d – Module d’analyse et de visualisation de données

III – Analyse des résultats

III – a – Le résultat obtenu

III – a – i) Comparatif avec l’objectif défini en début de projet

III – a – ii) Axes d’amélioration

III – b – Analyse du travail en équipe et de la gestion de projet

IV – Conclusion générale

I – Présentation du contexte

I – a - L'entreprise MyScript

Ce projet d'option est en collaboration avec l'entreprise **MyScript**, entreprise Nantaise leader dans les technologies de reconnaissance d'écriture manuscrite. L'entreprise développe différents moteurs de reconnaissance d'écriture : **MyScript Text**, **MyScript Math**, **MyScript Shape** et **MyScript Music**. La technologie **MyScript** est disponible sur les principales plateformes de développement d'applications (ios, android, windows) et est utilisée par beaucoup d'entreprises souhaitant intégrer des fonctionnalités de reconnaissance d'écriture dans leurs services.

I – b - La mission

L'objectif de ce projet d'option est de développer une application utilisant la technologie **MyScript** et permettant d'évangéliser l'utilisation de la reconnaissance d'écriture. Ce projet s'inscrit dans le scope de l'option GIPAD, et inclue donc une partie collecte de données.

Nous avons porté notre choix sur le développement d'une application à visée pédagogique. En effet, la technologie de reconnaissance d'écriture trouve parfaitement sa place dans des problématiques d'apprentissage. Par exemple, apprendre à calculer en utilisant une application permettant de numériser les chiffres écrits et d'afficher un résultat offre un gain en interactivité très intéressant. Un enfant apprenant à calculer sera plus intéressé par un exercice à réaliser sur tablette que sur une feuille de papier.

I – c - Analyse des contraintes et choix de l'application à développer

Au début du projet, nous avons centré notre étude sur 3 des 4 technologies **MyScript** décrites ci-dessus : les technologies Text, Math et Shape.

En premier lieu, nous voulions intégrer plusieurs technologies **MyScript** au sein de notre application et proposer des exercices de géométrie et d'algèbre à des élèves de niveau primaire et collège. Après analyse des contraintes (temporelles et techniques) et discussion avec notre tuteur en entreprise Monsieur Anthony Laurence, **notre choix définitif s'est porté sur le développement d'une application permettant d'effectuer des exercices d'algèbre et s'adressant à des élèves d'école primaire**. La technologie utilisée dans notre application est donc **MyScript Math**, permettant de créer des exercices interactifs.

II – Présentation de l'application « Mes petits exercices »

II – a - Introduction

Notre application est une application **Android** intégrant le Math widget développé par **MyScript**. Elle donne accès à des exercices d'algèbre à résoudre par un utilisateur de niveau école primaire. L'application permet aussi de collecter les données utilisateurs afin de les analyser et de permettre ainsi de produire des informations sur le niveau d'un utilisateur (pour chaque exercice réalisé, le nombre de tentatives, le nombre de réponses fausses et vraies, ou encore l'évolution en fonction du temps du niveau d'un utilisateur). Notre application intègre donc également un serveur et une base de données permettant de traiter cette partie collecte de données.

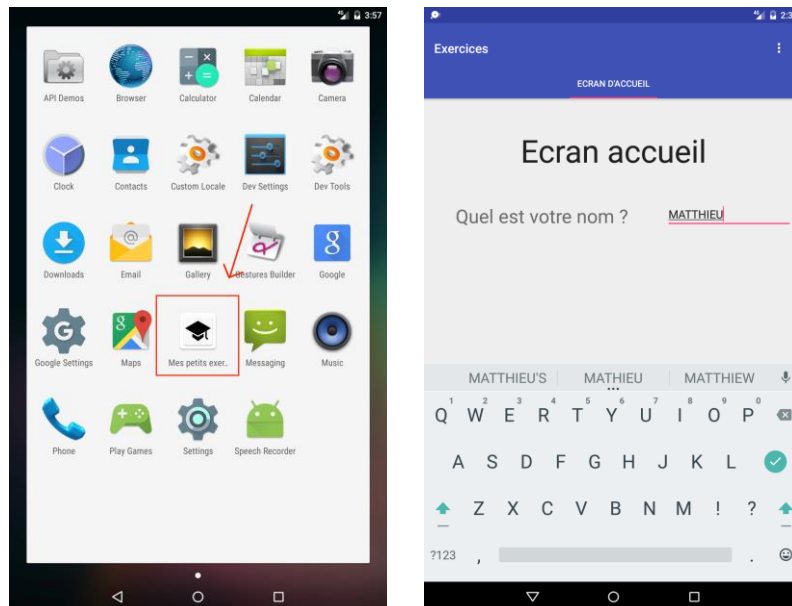
Nous pouvons diviser notre application en 3 grandes parties :

- L'application **Android** intégrant le math widget, les exercices et l'interaction avec l'utilisateur et permettant l'envoi de données utilisateur
- Une base de données permettant de stocker les données utilisateurs
- Un serveur permettant de faire l'interface entre l'application et la base de données

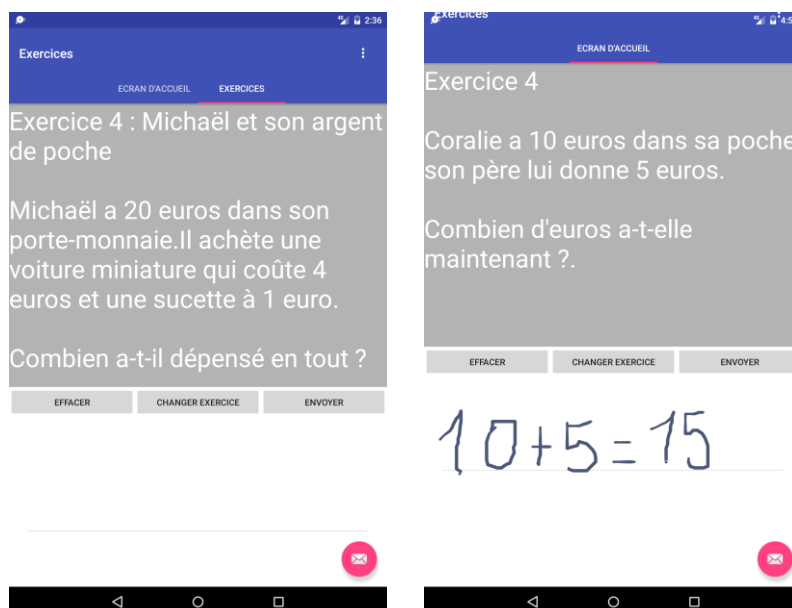
II – b – La partie Application

Pour présenter l'application « Mes petits exercices », nous allons expliquer un scénario d'utilisation simple qui permet de montrer comment utiliser l'application.

Etape 1 : Lancer l'application et arriver sur le menu principal ou l'utilisateur s'identifie



Etape 2 : L'utilisateur effectue un exercice, il écrit le résultat de l'exercice au stylet dans la zone blanche



Etape 3 : L'utilisateur envoie la réponse (en cliquant sur le bouton rose « envoyer » qui est analysée par le widget Math de MyScript. A cette étape, la donnée utilisateur est également envoyée via le serveur vers la base de données Mongo. On voit ci-dessous que l'écriture manuscrite est dans un premier temps transformée au format LaTeX, et que l'application indique à l'utilisateur s'il a réussi ou non l'exercice.



II – c – L'architecture client-serveur pour la phase de collecte de données

II – c – i) Motivation

D'un point de vue pédagogique, il est intéressant de récupérer les données d'utilisation sur l'application afin de pouvoir évaluer la performance d'un utilisateur. Ainsi, la partie collecte de données est une fonctionnalité critique, donnant une valeur ajoutée à l'application.

Pour mettre en place ce processus de collecte des données, nous avons choisis une architecture client-serveur. Le but étant qu'à chaque fois qu'un utilisateur effectue un exercice et appuie sur le bouton « envoyer », un fichier avec les informations sur le résultat de l'exercice soit envoyé par l'application à la base de données.

II – c – ii) Présentation des technologies utilisées

Au niveau de l'application **Android** (côté client donc), nous utilisons la bibliothèque **OkHttp** qui permet d'envoyer des données en utilisant le protocole HTTP.

Côté serveur, nous avons développé une API REST en utilisant **Node.js** et le module **Express** pour permettre l'échange de données entre l'application et la base de données. L'architecture REST respecte un certain nombre de règles et utilise les méthodes traditionnelles (que nous avons donc implémentées) du protocole HTTP :

- GET : pour la lecture des données
- POST : pour l'écriture de nouvelles données
- PUT : pour la modification des données
- DELETE : pour la suppression des données

Notre serveur est déployé sur un cloud, nous utilisons le service **Heroku**.

Côté base de données, nous utilisons MongoDB, qui est un SGBD de type NoSQL. Les bases de données Mongo reposent sur des collections de documents au format JSON et sont particulièrement adaptées à notre cas, étant donné que la donnée produite par notre application est au format JSON. La base de données Mongo est hébergée par le service **MLab**.

Nous allons expliquer dans l'exemple ci-dessous le fonctionnement de cette architecture client-serveur.

II – c – iii) Etude sur un exemple

Explication du processus d'envoi de données de l'application vers la base de données.

Considérons un utilisateur de l'application résolvant un exercice. Cet utilisateur possède un identifiant utilisateur (**idutilisateur**), l'exercice un identifiant exercice (**idexercice**). L'application est en mesure d'envoyer à la base de données l'information suivante : l'utilisateur **idutilisateur** effectué l'exercice **idexercice** à une certaine date et a donné une réponse vraie ou fausse. Ainsi, lorsque l'utilisateur clique sur « envoyer » (voir ci-dessus) à la fin de l'exercice, l'application **Android** produit le fichier JSON suivant :

```
{
  "idclient" : "Matt",
  "idexo" : "Exo1",
  "reponse" : "faux",
  "date" : "17-02-18"
}
```

Voici comment fonctionne le processus :

- 1 – Envoie du fichier JSON de l'application Android vers le serveur en utilisant le client **OkHttp**
- 2 – Le serveur reçoit le fichier et l'envoie vers la base de données en utilisant la méthode **POST**
- 3 – La donnée utilisateur est stockée dans la base de données Mongo

Voici un schéma récapitulatif présentant l'architecture client-serveur :



Remarque : Nous insistons ici sur le fait que nous avons utilisé deux services de cloud-computing **Heroku** et **MLab**. Ce choix est dû à la fiabilité de ces services en termes d'hébergement de serveur (Heroku) et de bases de données (MLab) et surtout à la facilité de déploiement d'application en utilisant ces services.

II – d – Module d'analyse et de visualisation de données

Afin de valoriser les données collectées, nous avons décidé de mettre en pratique nos cours de visualisation de données et réalisant deux visualisations. Ces visualisations s'appuient chacune sur une collecte JSON de la base puis sur un script javascript utilisant une bibliothèque externe.

Comme visualisation, nous avons retenu une représentation des pourcentages de bonnes réponses pour chaque exercice afin de pouvoir équilibrer au besoin la difficulté des exercices.

Nous avons aussi réalisé une courbe représentant le volume de POST journalier afin d'évaluer l'activité de l'application.

III – Analyse des résultats

III – a – Le résultat obtenu

III – a – i) Comparatif avec l'objectif défini en début de projet

Nous avons dû faire un recentrage de notre objectif vers une application n'utilisant qu'un seul widget. L'application obtenue atteint plusieurs objectifs définis en début de projet :

- développer une application à visée pédagogique
- développer une application fonctionnelle intégrant de manière optimale la technologie MyScript
- mettre en place un processus de collecte de données pour la visualisation et l'analyse de la performance des utilisateurs de l'application

III – a – ii) Axes d'amélioration

L'application pourrait dans le future intégrer d'autres widgets MyScript et ainsi élargir le spectre de ses utilisateurs (collégiens, lycéens). La partie analyse de données pourrait être aussi approfondie en utilisant des techniques de data-mining avancées.

III – b – Analyse du travail en équipe et de la gestion de projet

Notre travail s'est organisé en 2 phases principales :

Phase 1 : De Octobre à Décembre

- Prise en main de la technologie MyScript
- Montée en compétences de l'ensemble de l'équipe sur le développement Android
- Choix des technologies à utiliser pour la partie serveur (JavaScript, Node.js, choix des services de cloud)
- Montée en compétences sur le langage JavaScript pour les personnes responsables de cette partie

Phase 2 : Mois de Janvier et mois de Février

- L'équipe a été divisée en 2 : Ali et Aurélien ont travaillé sur la partie application, Matthieu et Thibault ont travaillé sur la partie serveur.

Difficultés rencontrées :

- Prise en main de nouvelles technologies (peu de connaissances en développement d'application pour les membres de l'équipe en début de projet)
- Difficultés à mettre en place l'environnement de développement
- Difficultés à établir la connexion entre l'application et le serveur

Tous ces problèmes ont cependant été résolus.

Nombres d'heures passées à travailler sur le projet : Nous avons travaillé en équipe lors de chacune réservées au projet d'option sur notre emploi du temps Oasis. En plus de ces heures nous avons travaillé plusieurs week-ends sur le projet. Enfin, un travail personnel d'apprentissage a été effectué par chacun des membres de l'équipe.

IV – Conclusion générale

L'ensemble de l'équipe a beaucoup apprécié travailler sur ce projet. En effet, celui-ci a permis une montée en compétences sur un panel de technologies que nous n'avions pas utilisées auparavant : développement Android, développement JavaScript, mise en place d'API REST, connexion entre une application et un serveur, gestion d'une base de données Mongo.