

Projet S2 – Compte rendu

D3.2 : Rapport du développement/test du
02/06/2016

2015/2016

Groupe 1.2 - “GoForm”



Les fonctionnalités de l'application :

Selon nous, l'application est fonctionnelle. En effet nos deux objectifs principaux ont été atteints.

Le premier était de gérer un profil d'utilisateur (inscription et connexion) avec une base de données pour stocker toutes les informations propres à ce dernier (suivi de performance au fil des semaines, création de séances personnalisées, informations personnelles). Toutes ces informations nécessitant bien sûr une identification (par mail et mot de passe) lors du lancement de l'application dans le but de garantir une certaine confidentialité à l'utilisateur.

Le deuxième était de donner la possibilité à un utilisateur de créer ses propres entraînements et séances d'entraînements afin de lui offrir une certaine liberté au sein de son programme sportif. D'abord basé sur des entraînements et des séances créées par nos soins, ce dernier peut modifier à sa guise les quantités et les types d'exercice s'il y trouve de l'intérêt. Il peut ensuite bien évidemment réaliser ses exercices, avec à la fin un suivi mis à jour lorsqu'il valide son entraînement.

La réalisation des « fonctions » :

Certaines fonctions de notre application sont totalement réussies : L'inscription d'un nouvel utilisateur, la connexion d'un utilisateur déjà connu de la base de données, la création d'une séance d'entraînement personnalisé, l'affichage du profil, la barre de progression dynamique indiquant l'IMC de l'utilisateur ainsi que la déconnexion de ce dernier.

En revanche nous avons dû abandonner certaine fonctionnalité par manque de temps. C'est le cas par exemple des réglages : Il est impossible pour un utilisateur de modifier ses informations personnelles à partir de l'application à cause du rafraichissement des informations saisis.

De plus nous voulions inclure un player de musique afin de dynamiser et motiver l'utilisateur lors de la pratique de ses exercices. Nous aurions aussi aimé permettre à l'utilisateur de changer la couleur de l'application et générer des confirmations d'adresse mail.

La notion de calendrier a aussi été laissée de côté dès le début du codage. Ces fonctionnalités ont été abandonnées au profit des autres.

Les difficultés rencontrées :

De nombreuses difficultés ont été rencontrées lors de la production du code de notre application notamment pour tous ce qui était relatif à la base de données et à l'IHM. Le manque de connaissances sur le sujet et le temps nous ont beaucoup pénalisés. Nous avons dû passer un certains nombres d'heures à lire de la documentation et suivre des tutoriels. Evidemment tous ces ralentissements nous ont empêchés de suivre la progression prévisionnelle du code. Lorsque nous estimions que nous passions vraiment trop de temps sur une difficulté et que cette perte de temps allait être réellement dommageable, nous allions chercher les avis des experts M.GENTHIAL et M.JEAN ainsi que M.GELIBERT.

De plus, le fait de travailler sur une base de données nous oblige à être présent à l'IUT car nous ne pouvons pas y avoir accès depuis l'extérieur. Pour pallier à cela nous avons fait des tests en local afin de pouvoir avancer chez nous, mais une fois de retour il fallait à nouveau transformer le code pour qu'il

corresponde à la base de données de l'IUT.

Un point que nous n'avions pas prévu était le fait que les méthodes décidées en conception, sont devenue la plus part des requêtes SQL. Ceci provoque notamment une modification drastique de notre schéma de conception. De plus, ce code est plus compliqué à factoriser.

Sur ce dernier point justement, pris par le temps, nous n'avons pas pu faire de factorisation de code.

Lancement de l'application :

Notre application se lance via la Classe principal « IHMConnexionUser ». Lorsqu'on exécute cette dernière, une fenêtre s'ouvre et trois choix sont proposés à l'utilisateur :

Soit il possède déjà un compte et il remplit les champs relatifs à une connexion (adresse mail et mot de passe) après quoi une requête est envoyée à la base de données pour vérifier si cette utilisateur est connu de cette dernière. Si la requête est acceptée, une autre fenêtre s'ouvre avec l'interface utilisateur. Sinon, un pop-up s'ouvre indiquant qu'au moins un des deux champs est incorrect.

Le deuxième choix est celui de l'inscription : Si l'utilisateur ne possède pas de compte, il clique sur le bouton inscription et remplit les champs nécessaires pour générer cette dernière.

Le troisième choix est un bouton lui permettant de quitter l'application.

Tests unitaires :

En ce qui concerne les tests unitaires nous avons dédié 3 Classes à cet effet : « Inscription » , « Exercice » et « Connexion ». Il semblait évident de tester en premier une inscription car c'était le premier défi à relever dans notre application. Nous avons donc redéfini une méthode qui permet de faire un appel à des tests de pattern c'est-à-dire une chaine de caractère à suivre comme dans un mail : « Cedric@hotmail.fr ». Le test porte sur le fait qu'on doit faire attention à l'arobase et au point entre le « fr » et le « hotmail ». Il visait donc à faire respecter la casse standard d'une adresse mail.

De plus, nous avons inséré des intervalles pour tester les paramètres d'entiers. Dans cette méthode de test nous avons mis les paramètres qu'on aurait souhaités lorsqu'un individu s'inscrit sur notre application. Nous n'avons pu appliquer cela sur notre réelle inscription car nous avons eu un manque de temps. Les deux autres classes restante sont relatifs à une connexion et un exercice avec les mêmes procédées. Il y a eu un autre type de test qui a était écrit en dur pour notre application qui se nomme « testSeance ».