

Application mobile Android de consultation de médicaments pour le laboratoire GSB

Développer les modules d'une application Java Android existante à partir d'une base de données SQLite embarquée

Propriétés	Description
Intitulé	Développement en Java et SQL dans un environnement Android
Formation concernée	BTS Services Informatiques aux Organisations
Matières	SI3 : Exploitation des données SI4 : Bases de la programmation (fin de semestre)
Présentation	<p>L'entreprise GSB a besoin, pour ses visiteurs médicaux, d'une application mobile recensant les médicaments mis sur le marché en France (un peu comme l'application mobile de l'entreprise <i>Vidal</i>, mais construite à partir de données libres de droit).</p> <p>Une équipe de techniciens a été déléguée pour le projet. Ils travaillent dans un processus Agile Scrum, avec les technologies Android Studio, git et SQLite. Les données, récupérées en csv à partir du site data.gouv.fr, ont été migrées dans SQLite. Une application de test programmée en Java - <i>Material Design</i> est fournie sur git.</p> <p>L'ordre proposé des missions est le suivant :</p> <p>Mission 1 Installation et manipulation de l'environnement de développement Android Studio</p> <p>Mission 2 Exploitation d'une base de données SQLite embarquée sur le smartphone Android</p> <p>Mission 3 Cours d'introduction à la programmation objet (création d'objets en Java)</p> <p>Mission 2 Utilisation d'ArrayList en Java</p>
Transversalité	U22- Mathématiques, algorithmique appliquée
Pré-requis	Modèle relationnel SQL Langage d'Interrogation des données, Langage de Manipulation des Données Bases de la programmation structurée (variables, structures conditionnelle et itérative, typage, fonction) Utilisation de base de git (clone et branches) Vocabulaire de base Scrum
Outils	Environnement de développement intégré (IDE) Android Studio, git, SQLite
Mots-clés	SQL, SQLite, Langage d'Interrogation des Données, intégration d'une base SQLite dans une application Android Android Studio, AVD, API, console Logcat Initiation à la programmation orientée objet, classe, objet, constructeur, new, signature et argument d'une méthode, paramètre d'une méthode, classe métier, classe technique, classe DAO, adresse mémoire, référence, hash code, git, Material Design Parcours d'ArrayList

Prolongements possibles	<p>En SLAM2 :</p> <ul style="list-style-type: none"> - programmation de classes métiers et de leurs liens (en faisant évoluer le diagramme de classes initial) <p>En SLAM4 :</p> <ul style="list-style-type: none"> - modification de la couche de DAO afin de récupérer les données sur un service web REST (et plus sur SQLite en local) - programmation des tests unitaires des classes métiers déjà implémentées - évolutions de l'interface graphique <i>Material Design</i> - migration de la couche de DAO en utilisant un l'ORM Java Android - notifications en <i>push</i> sur l'application avec Firebase - photographie et capture de codes-barres des médicaments pour effectuer des recherches (bibliothèque <i>zxing</i> par exemple) <p>En SLAM5 :</p> <ul style="list-style-type: none"> - apprentissage du modèle MVC sur une application Java Android - développement en équipe (<i>sprints</i>) sur une architecture MVC en utilisant git
Durée	20 heures
Auteur(es)	Fabrice Missonnier, relectures Hervé Le Guern, Olivier Capuozzo et Yann Barrot
Version	v 1.0
Date de publication	mai 2019

I. Contexte

L'activité majeure du laboratoire GSB est la production et la distribution des médicaments. L'entreprise compte 480 visiteurs médicaux en France métropolitaine (Corse comprise) et 60 dans les départements et territoires d'outre-mer. Les territoires sont répartis en 6 secteurs géographiques (Paris-Centre, Sud, Nord, Ouest, Est, DTOM Caraïbes-Amériques, DTOM Asie-Afrique).

Ces visiteurs médicaux doivent connaître les molécules des produits qu'ils vendent. Il existe des milliers de médicaments en vente pour les particuliers et les professionnels. Il leur est impossible d'apprendre l'ensemble des compositions des médicaments (dont ceux de leurs concurrents).

Lorsqu'ils sont en clientèle, les visiteurs médicaux souhaiteraient avoir à leur disposition une petite application simple qui recense les médicaments, leurs compositions ainsi que les formes commerciales dans lesquels ils sont vendus (boîte, comprimé, etc.).

Ce genre d'application existe déjà dans le commerce : l'entreprise *Vidal* fournit aux médecins une application mobile. Par contre, les données brutes ne sont pas accessibles et il est impossible de les manipuler. Le directeur marketing de GSB serait intéressé par un travail plus complet de statistiques sur les médicaments. Par exemple, il souhaiterait connaître le nombre de médicaments dont le service médical est insuffisant, ou encore les médicaments des concurrents qui contiennent certaines substances actives fournies par GSB.

L'application mobile doit être programmée pour le système d'exploitation Android. Elle permettra aux différents personnels de GSB de visualiser un descriptif de médicaments. Le choix technologique imposé par le directeur de la branche « développement » est une application fonctionnant en *mode déconnecté*.

Même si un certain nombre de développements d'applications sont externalisés par l'entreprise GSB dans une Entreprise de Services du Numérique (ESN), ce logiciel sera développé en interne.

I. Equipe de développement

L'équipe de développement du laboratoire dans laquelle vous allez être intégré se compose de 3 personnes. Ces informaticiens travaillent, parfois, sur plusieurs projets en même temps. Ils ont déjà programmé un certain nombre de fonctionnalités pour le projet. Le code source est versionné à partir du système de gestion *git*.

Il arrive que ces développeurs interviennent parfois avec l'ESN en tant que *spécialistes métiers*. Dans ces projets, ils utilisent la méthode agile *Scrum*. Les termes Scrum étant spécifiques, des définitions succinctes sont proposées dans le lexique annexé à la fin de ce document (les termes suivis d'un * sont définis dans ce lexique, voir document 1 p.8).

L'équipe est composée de cinq membres, dont vous.

- Bryan Stylemon, développeur *back-office* qui connaît très bien la partie métier des différentes bases de données de GSB. A 48 ans, il travaille depuis plus de 25 ans dans des environnements hétérogènes. Il est particulièrement compétent sur les bases de données Oracle et SQL Server, où il a aussi une fonction d'administrateur de bases de données. Il a travaillé sur de nombreux scripts pour la récupération et la migration de données (dans les langages *bash* et Java). Il est le *scrum master** de l'équipe. Il fait partie de *l'équipe de développement**.
- Anne-Lise Malet est graphiste. C'est elle qui intervient sur le *design* des sites web et des applications GSB. Elle est très occupée car l'équipe marketing lui demande beaucoup : logos, web marketing, etc. Elle occupe aussi la fonction de *community manager* pour GSB. Son rôle, dans le projet, sera de créer et de valider la charte graphique de l'application. Elle fait aussi partie de l'équipe de développement.
- Kévin Viti : développeur Java depuis 5 ans, Kévin a un profil DevOps. Il développe sous Android et connaît bien les différentes architectures applicatives de GSB.
- Norleen Nosry est médecin conseil. Elle est salariée par GSB depuis deux ans. Elle travaille sur la partie technique des nouveaux produits destinés aux personnels médicaux (médecin, sage-femme et infirmier). Elle a été désignée par sa hiérarchie pour être *product owner** (PO) du projet.

Avant de débiter le projet, Bryan a établi la matrice de compétences de l'équipe. Norleen étant PO et non technicienne, elle n'apparaît pas.

	<i>Bryan</i>	<i>Anne-Lise</i>	<i>Vous</i>	<i>Kévin</i>
<i>Android (OS)</i>	😊	😊	😊	💕
<i>Java Android Studio (front -> design et interfaces graphiques)</i>	😡	😊	😐	😐
<i>Java Android Studio (back -> data)</i>	😡	😡	😐	😐
<i>SQL & SQLite</i>	💕	😡	😊	😊
<i>Parsing de données CSV</i>	😐	😡	😡	😡

- 😡 : je ne sais pas faire et ça ne m'intéresse pas
- 😐 : je ne sais pas faire mais pourquoi pas
- 😐 : je sais faire mais si je pouvais un peu travailler sur autre chose...
- 😊 : j'ai des connaissances et je veux bien m'impliquer sur la technologie
- 💕 : j'aime, j'ai très envie de continuer

II. Descriptif des missions

Le projet a débuté il y a un mois : le *sprint** 0 est terminé et une *revue de sprint** a déjà été effectuée avec le PO. Le sprint 0 correspond à la première itération du projet. Au vu de la matrice de compétences, il a été négocié dans l'équipe que chacun effectuerait les tâches suivantes.

Résultats du sprint 0

Kévin a travaillé 4 semaines à 100% sur le projet, tandis qu'Anne-Lise a programmé une trentaine d'heures.

Plusieurs applications Java Android ont été réalisées en parallèle par Kévin et Anne-Lise. La première permet de tester l'environnement Android Studio. La seconde intègre l'utilisation d'un module graphique particulier sous Java Android, le *material design* :

<https://developer.android.com/guide/topics/ui/look-and-feel/>

L'utilisation de cette technologie correspond aux attentes du directeur marketing.

Bryan a programmé la migration des données brutes des médicaments. Il avait déjà pas mal de code en Java qui permettait de *parser* des fichiers CSV, ce travail n'a pas été techniquement trop difficile pour lui.

Les données brutes qu'il a utilisées sont disponibles sur le site

<https://www.data.gouv.fr/fr/datasets/base-de-donnees-publique-des-medicaments-base-officielle>
<http://base-donnees-publique.medicaments.gouv.fr/telechargement.php>

Il a travaillé avec la cellule juridique de GSB pour être sûr de pouvoir réutiliser les données. L'entreprise n'a, par contre, pas le droit de les dénaturer (les modifier par exemple). Elles sont réutilisables commercialement (licence ouverte) :

<https://www.etalab.gouv.fr/licence-ouverte-open-licence>

Bryan a programmé en Java une application qui permet, à partir de ces données initiales, de générer des scripts SQL. Les données ont été insérées dans une base SQLite. Bryan tient à préciser que l'insertion des tuples dans la base SQLite prend environ 5 heures. S'il faut modifier la base (en cas d'erreur), il peut difficilement en fournir une nouvelle rapidement.

Norleen s'est occupée du *backlog de produit** de l'application, où elle a référencé les principaux besoins de l'application.

Vous arrivez sur le projet au début du nouveau sprint (appelé sprint 1). N'ayant pas le profil de développeur Java sous Android, l'équipe a bien entendu, lorsque vous discutiez à la machine à café, que vous seriez très intéressé par cette technologie. Le DSI de GSB a accepté de libérer 50% de votre temps sur ce projet.

Les missions suivantes répondent à deux objectifs techniques :

- savoir si la base de données fournie SQLite répond aux besoins du PO,
- développer une application Java sous Android dont la finalité sera de traiter les données de la base SQLite en fonction des besoins des utilisateurs sur un smartphone ou une tablette.

Mission 1 (2 heures)

Il a bien été noté que vous saviez déjà utiliser des environnements de développement standard (Visual Studio, Eclipse, Netbeans, etc.). Dans ces conditions, la première étape est de comprendre le fonctionnement de l'exécution d'une application dans un environnement Android.

- 1- Installer l'environnement de travail et récupérer le projet en cours de développement sous git. Voir la mission « Mission n°1, Utilisation d'Android Studio. »

Mission 2 (6 heures)

La seconde étape est de valider si les données insérées dans la base SQLite répondent aux besoins du product owner. La base a été déployée dans le projet Android Studio afin de tester en réel dans l'environnement Android.

- 2- Installer SQLite sur une plateforme virtuelle Android et vérifier les données SQLite. Voir la mission « Mission n°2, SQLite ».

Mission 3 (6 heures)

Cette troisième mission consiste à programmer en Java des classes et des objets correspondant aux données de la base. Kévin a travaillé sur la maquette d'une application Android. S'occupant de l'architecture du code applicatif, il souhaite que vous repreniez le code de la maquette pour y développer la partie « métier ».

Cette maquette a été programmée pour évaluer la faisabilité du projet. Ce n'est, bien sûr, pas un produit fini.

Vous allez intervenir dans le sprint 1 en tant que développeur back-office. L'objet de la mission est de reprendre le code de Kévin et de comprendre le fonctionnement des objets manipulés dans l'application.

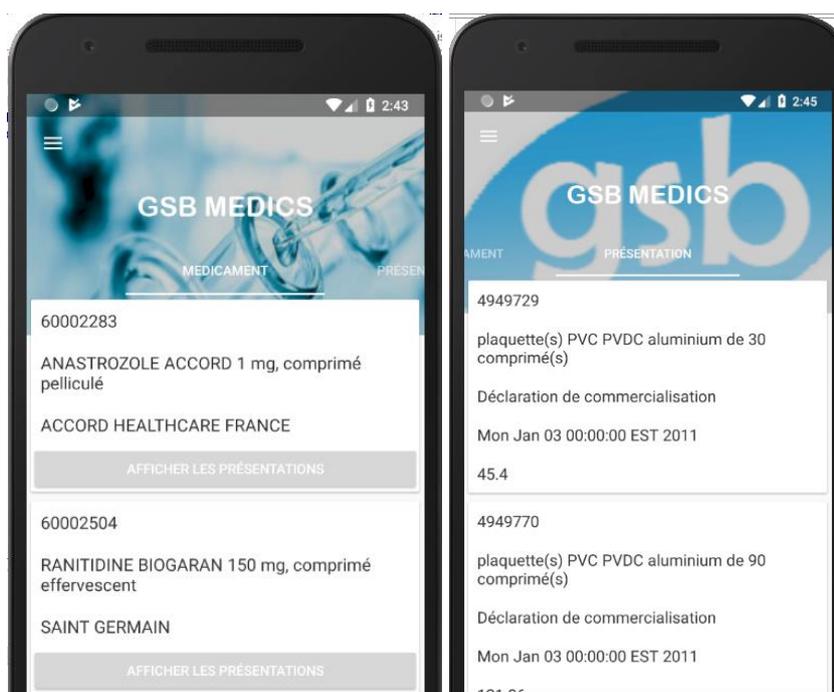
- 3- Récupérer la branche créée par Kévin sous git et créer les différents objets nécessaires à l'application. Voir la mission « Mission 3 Java - apprentissage POO ».

Mission 4 (6 heures)

En parallèle, Anne-Lise a développé une nouvelle maquette d'interface graphique utilisant une technologie de front-office proposée par Google, le *material design*
https://fr.wikipedia.org/wiki/Material_design

Après quelques recherches sur Internet, elle a trouvé un projet permettant de tester le material design :
<https://github.com/florent37/MaterialViewPager>

Anne-Lise a récupéré le code source, l'a corrigé et intégré au projet. La maquette d'où sont issues ces captures d'écran est disponible sur une branche git. Les médicaments sont affichés sous forme de « carte » :



Lors d'une *daily scrum**, elle explique que chaque carte affiche un objet Java (un médicament, un laboratoire, etc.). Elle a déjà effectué des tests rapides sur des objet de type `Medicament`.

Techniquement, dans le code de la couche métier, il faut obligatoirement instancier l'ensemble des objets pour qu'ils soient affichés. C'est le travail que vous avez déjà effectué dans la mission précédente. Cependant, les données sont stockées dans la base SQLite et il n'est pas possible de garder l'instanciation des objets « en dur » dans le code Java. Il faut donc créer automatiquement ces nouveaux objets à partir de la base de données.

Kévin a travaillé en même temps qu'Anne-Lise sur la partie d'accès aux données SQLite à partir de l'application Java :

- il a programmé l'ensemble des classes métiers de l'application correspondant au modèle relationnel,
- il a créé une nouvelle classe DAO qui fait la correspondance (*mapping*) entre les données de la base et les objets Java.

Par exemple, chaque médicament de la base de données sera extrait sous forme d'un objet médicament (d'une instance de la classe).

Les références à ces objets métiers sont ensuite stockées dans des tableaux et disponibles pour l'affichage.

Dans cette mission, les objectifs sont de comprendre le code source qui a été produit et de travailler sur ces tableaux.

- 4- Analyser le code source d'Anne-Lise et celui de Kévin. Voir la mission dans le répertoire `Mission 4 Java - tableaux et ArrayList`.

Document 1 : Lexique Agile Scrum

<https://agiliste.fr/lexique-agile-scrum/>

Voici un lexique des termes les plus couramment utilisés sur des projets agiles menés avec le cadre méthodologique Scrum [...].

Dev Team / Equipe de Développement : Terme couramment utilisé pour désigner l'équipe de développement qui réalise le produit.

Mêlée Quotidienne / Daily Meeting / Stand-Up Meeting : Réunion quotidienne de moins de 15 minutes permettant à l'équipe de développement de se synchroniser, d'identifier les obstacles éventuels et de mesurer son avancement sur le Sprint en cours.

Product Backlog / Backlog : Liste ordonnancée (priorisée) des besoins (généralement formulés sous forme de User story) du projet.

Product Owner : Représentant des utilisateurs qui porte la vision du produit que l'on souhaite réaliser dans le cadre du projet. Il est responsable du Product Backlog et en interaction directe avec l'équipe de développement.

Release / Version : Une release est une nouvelle version du produit, livrée aux utilisateurs. Elle est le fruit de plusieurs Sprints.

Scrum : Cadre méthodologique de gestion de projet agile.

Scrum Master : Garant du respect du cadre méthodologique Scrum. Il a également un rôle de coach vis à vis de l'équipe de développement et du Product Owner afin d'aider ces derniers à jouer pleinement leur rôle.

Scrum Team / Equipe Scrum : Rassemble en partenariat la Dev Team, le Product Owner et le Scrum Master.

Sprint / Itération : Intervalle de temps court (1 mois maximum, souvent appelé itération), pendant lequel la Dev Team va concevoir, réaliser et tester de nouvelles fonctionnalités.

Sprint Review / Revue de Sprint : Réunion de travail consistant à présenter aux parties prenantes les fonctionnalités terminées au cours du Sprint afin de recueillir leurs feedbacks. Et à faire le point sur l'avancement global du projet.

User Story / Story : Technique permettant de formaliser synthétiquement les besoins sans perdre de vue l'essentiel : le besoin concerne QUI, en QUOI il consiste et dans quel BUT.