



المدرسة الوطنية للمهندسين بتونس

école nationale d'ingénieurs de Tunis

Ecole Nationale d'ingénieur de Tunis

&

*Département de Technologie de l'information et de
la communication*

Développement d'une application mobile
hybride présentant une bibliothèque
numérique sélective de livres

Elaboré par

Kais LABIEDH

Nadine BEN ALAYA

Au sein du

Département TIC



Année Universitaire

2015/2016

Sommaire

Introduction Générale

1. Chapitre 1 Spécification des besoins

- 1.1 Besoins fonctionnels
- 1.2 Besoins non fonctionnels

2. Chapitre 2 Conception du système

- 2.1 Architecture logicielle du système
- 2.2 Diagramme de classes

3. Chapitre 3 Réalisation de l'application

- 3.1 Environnement de travail
- 3.2 résultat de l'implémentation

Conclusion

Introduction Générale

Auparavant le livre représentait le véhicule par excellence du savoir et une source indisponible de culture. Il est vrai que par la suite, l'apparition de l'internet a fait exploser les puits des connaissances et de la cognition, mais il s'est avéré que pas mal de personnes demeurent bouquineurs et accros à la lecture, traditionnelle soit elle ou virtuelle. Ces personnes là représentent une tranche considérable de la popularité. Cette dernière atteigne, en 2014, jusqu'à 30 % du peuple Tunisien et aux alentours de 50% de celui de la France.

D'autre part, Internet a contribué à la véritable révolution dans l'univers des livres, du traditionnel format papier on passe aux livres virtuels, où il s'avère possible aux férus de lecture d'en acheter une quantité phénoménale de toutes sortes, dont plusieurs parmi eux sont disponibles gratuitement sur le web.

En outre, depuis la fin du dernier siècle, l'univers de la haute technologie a connu une évolution spectaculaire qui a littéralement métamorphosé le quotidien des millions de personnes à travers le monde.

Cet accroissement est perceptible à travers les applications mobiles qui ont pris le dessus sur les applications de bureau. En plus d'être pratiques, accessibles à tout moment et à portée de main, elles couvrent quasiment toutes les catégories des applications informatiques. De plus, la simplicité d'utilisation des téléphones mobiles, l'accès permanent à Internet grâce à l'évolution des technologies de communication et la rapidité de téléchargement des applications depuis les différents stores sont des facteurs essentiels de cette apogée. Par ailleurs, les téléphones classiques disparaissent petit à petit du marché et 82% des téléphones mobiles seront des smartphones à la fin de l'année 2016.

Parmi les différentes catégories des applications mobiles existantes sur le marché les bibliothèques numériques se définissent comme des outils permettant aux passionnés de la lecture électronique des retrouvailles plus faciles et des choix sélectifs plus rapides des e-books . Notons qu'un e-book est une version numérique d'un livre traditionnel, qui pourrait être lu en utilisant un ordinateur personnel, un smartphone, etc ...

Le projet consiste à réaliser une application mobile hybride qui permet à l'utilisateur de consulter une bibliothèque en ligne, filtrer les livres qu'il cherche par genre, ajouter un livre et noter les livres déjà existants.

Le rapport de ce projet se divise en trois chapitres. Le premier consiste en une spécification des besoins fonctionnels et non fonctionnels de l'application. Le deuxième quant à lui détaille l'architecture logicielle du système conçu et finalement le troisième chapitre contient les détails de l'implémentation et le résultat obtenu du prototype développé.

Chapitre 1

Spécification des besoins

Introduction :

La spécification des besoins est une étape hyper importante pour s'assurer de la conformité des résultats qu'on obtiendra par la suite avec les besoins établis dès le début. Ce chapitre sera alors dédié à la spécification des besoins fonctionnels et non fonctionnels de l'application.

1.1 Les besoins fonctionnels

L'application doit permettre à l'utilisateur de :

- Consulter une liste des livres cités par (titre, genre, description et note)
- Ajouter un nouveau livre
- Noter un livre
- Rechercher un livre par titre
- Filtrer l'affichage par genre

1.2 Les besoins non fonctionnels

Le système à concevoir doit être :

- Multiplateforme (compatible avec iOS, Android, etc ..)

- Rapide
- Robuste
- Utilisable même si l'utilisateur n'est pas connecté à Internet
- Ergonomique

Conclusion

Dans ce chapitre nous avons cité et restreint les caractéristiques et les exigences que doit satisfaire le prototype auquel nous optons vers la fin du projet.

Dans le chapitre suivant, nous allons détailler les étapes de conception de l'application.

Chapitre 2

Conception du système

Introduction

La conception est une étape cruciale dans le cycle de développement de tout logiciel. En effet, il cette étape consiste à définir les réponses résolvant les comment-faire techniques correspondants aux spécifications qu'on a dégagées auparavant.

2.1 Architecture logicielle du système

L'architecture REST (Representational State Transfer) est un style l'architecture permettant de construire les applications web, Intranet, Webservice, etc ...

Il s'agit d'une multitude de conventions et de bonnes pratiques à respecter et non pas d'une technologie à part entière. L'architecture REST utilise les spécifications originelles du protocole HTTP, plutôt que de réinventer une surcouche (comme le font SOAP ou XML-RPC par exemple).

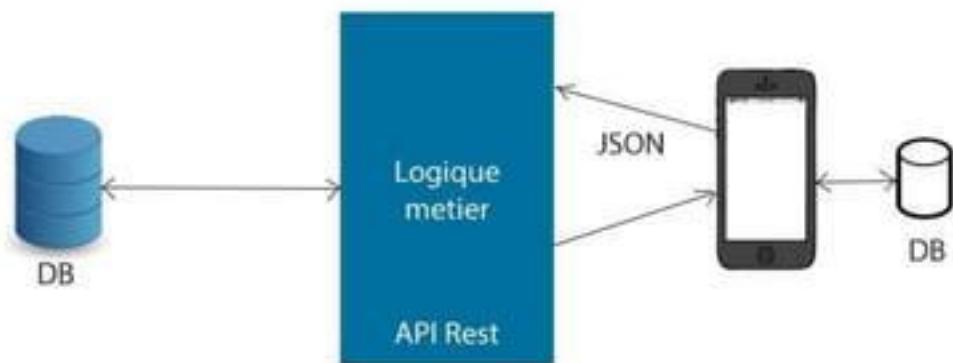


Figure1 : Architecture REST adaptée à l'application EasyRead

Nous avons eu recours à cette architecture, démontrée dans la figure 1, pour permettre à notre application de se connecter à une base de données distante et en même temps synchroniser, au moment où il y ait connexion d'un certain utilisateur, les données de la base de données locale avec celles de la base distante.

2.2 Diagramme de classe

On montre dans la figure ci-dessous le diagramme de classes de l'application qui décrit le comportement, et les responsabilités de ses objets.

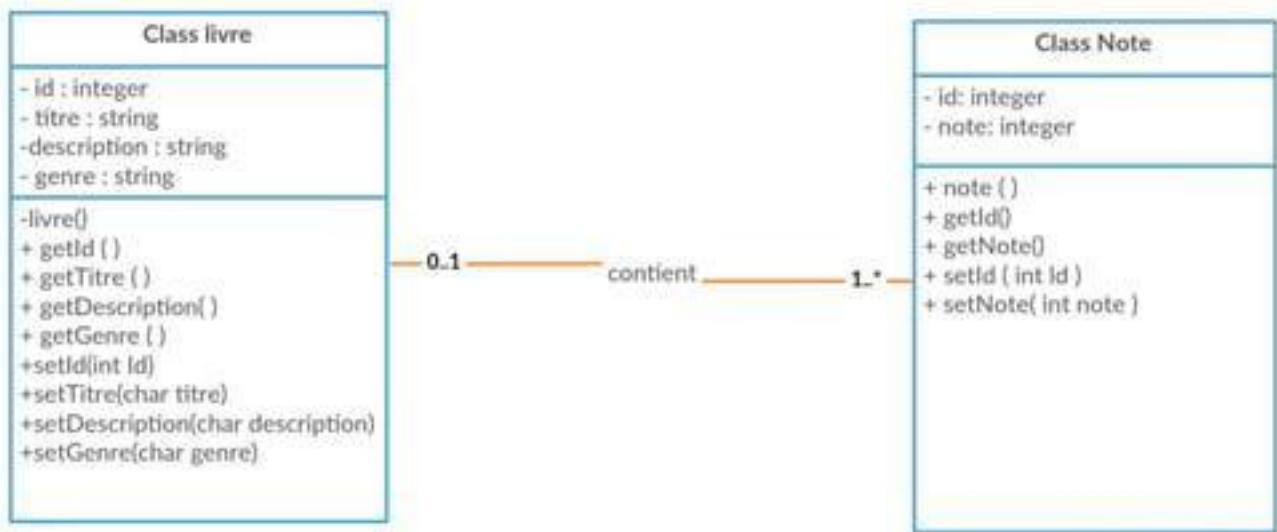


Figure 2 : Diagramme de classes

Le diagramme comporte deux classes, une pour les livres et l'autre pour les notes de ces derniers. Elles sont toutes les deux reliées par la relation « contient » (Chaque livre possède une ou plusieurs notes)

Conclusion

Dans cette partie du rapport nous avons cité les étapes de la conception de la solution pour pouvoir modéliser le diagramme de classes de l'application.

En outre, dans la section suivante nous allons entamer la phase de l'implémentation de la solution et nous allons exposer les résultats de l'exécution obtenus.

Chapitre 3

Réalisation de l'application

Introduction

Après avoir spécifié les besoins du système et conceptualiser la solution requise, nous allons, dans ce chapitre, citer les détails de l'implémentation, justifier nos choix technologiques et exposer vers la fins les résultats de l'exécution de l'application.

3.1 Environnement de travail

- ❖ Quant au développement de l'API (Application Programming Interface) nous avons opté à :
 - Node.js : Une plateforme logicielle OpenSource écrite en JavaScript et convergente principalement vers les applications réseaux qui ont à monter en charge. Cette plateforme se sert de la machine virtuelle V8 et implémente, les spécifications CommonJS, sous licence MIT.



Figure 3 : Logo de la plateforme Node.js

Nous avons utilisé en particulier le mini-Framework Express.js qui prend en charge la partie Routing.

- MySQL : Système de Gestion de base de données distante



Figure 4 : Logo de la plateforme Node.js

- ❖ Pour le développement de l'application et le traitement des données nous avons utilisé :
 - IONIC : Un Framework complet et Open source qui sert à développer des applications mobiles Hybrides. IONIC nous offre des composants CSS prêts à utiliser. La partie logique métier est offerte en Angular.js



Figure 5 : Logo du Framework Ionic

- Angular JS est un Framework libre principalement retenu par Google et désigné au développement des applications web one page.



Figure 6 : Logo de la plateforme Angular.js

- SQLite : Un système de gestion de base de données relationnelle contenue dans une bibliothèque de programmation C. Contrairement à de nombreux autres SGBD, SQLite n'est pas un moteur de base de données client-serveur, mais il est incorporé dans le programme final, autrement dit, c'est une sorte de fichier de sauvegarde.



Figure 7 : Logo de SQLite

3.2 Résultat de l'implémentation

L'application comporte en tout cinq interfaces. Une pour l'accueil, deux pour l'affichage de liste des livres et du menu principale, et finalement deux autres pour

L'ajout d'un livre et pour la visualisation de la description complète d'un livre qui existe déjà.

➤ Interface D'accueil



Figure 8 : Page d'accueil de EasyRead

La figure ci-dessus montre la page d'accueil qui va apparaître dès qu'un utilisateur lance l'application. Cet interface comporte un bouton « Go », qui, en cliquant dessus, va nous rediriger vers l'interface de l'affichage des livres existant. Le bouton contenant trois lignes parallèles nous permet de visualiser le Menu Principal.

➤ Menu principal

L'interface du menu principale va servir de guide à l'utilisateur, et ce en lui présentant les services offerts par l'application et en lui permettant de retourner vers l'interface précédente.

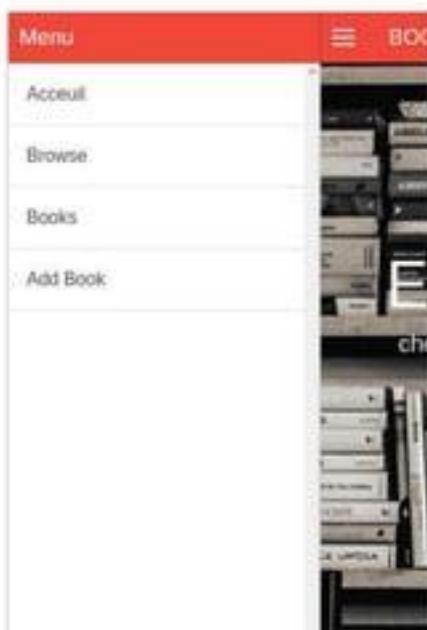


Figure 9 : Menu principale de EasyRead

Comme l'illustre la figure ci-dessus, le menu de notre application comporte quatre options :

- o « Accueil » : qui permet de revenir à la page d'accueil
- o « Browse » : Permet de nous amener à l'interface de recherche des livres par titres
- o « Books » : L'option qui affiche la liste des livres existants sans aucune sélection particulière
- o « Add Book » : Nous permet d'accéder au menu d'ajout de livre à la bibliothèque

Notons que le Bouton à trois tirets parallèles permet à l'utilisateur de faire disparaître la barre des menus

➤ Affichage de la liste des livres

La façon d'afficher des livres qu'offre EasyRead se fait en listant tous les œuvres existants



Figure 10 : Interface de l'affichage Liste des livres

Les livres qui existent dans la base de données locale (fichier de sauvegarde) de l'application sont affichés dans la liste par nom et revue de description. En cliquant sur l'un des livres, on migre vers l'interface qui contient tous les détails de ce dernier.

L'interface contient aussi un bouton de mise à jour de la liste.

➤ Afficher les détails d'un livre



Figure 11 : Description détaillée d'un livre « X »

L'interface ci-dessus est celle qui apparaît quand on clique sur l'un des livres de la liste précédente. Elle contient le titre, sa description complète, son genre et sa note. L'utilisateur aura tout de même le droit de noter un livre qu'il consulte. La note qui apparaît juste après la description est la moyenne de toutes les notes accordées par les utilisateurs à ce livre là.

➤ Ajouter un livre

Figure 12 : Menu d'ajout de livre à la bibliothèque de EasyRead

Pour ajouter un livre, l'utilisateur ne doit qu'ajouter le titre, la description et le type de l'œuvre en question. En cliquant sur le bouton « Add » le livre sera ajouté à la base et il apparaîtra, suite à une mise à jour, dans la liste des livres existants.

Conclusion

Dans cette section, nous avons détaillé nos choix technologiques et nous avons cité les fonctionnalités desquels jouit notre application, tout en montrant les résultats de l'exécution de cette dernière.

Conclusion Générale

L'application EasyRead, qu'on peut considérer comme une mini-bibliothèque numérique est un produit qui peut toucher une marge non-négligeable de la popularité. Elle peut servir, les accros à la lecture, les débutants cherchant un moyen pour leur faciliter le choix et la retrouvaille des œuvres et ainsi de suite.

En outre, l'application peut être considérée comme une plateforme d'échange d'avis ou de sondage d'avis au sujet des livres, et ce grâce au concept de note attribuée par chaque utilisateur et de la note moyenne qui s'affiche à la fin.

Ce projet peut certainement subir des améliorations grâce à l'existence de plusieurs pistes exploitables dans les technologies connexes à ce système. On pourrait rajouter un module de reporting, qui a comme but d'afficher à l'utilisateur, les temps forts de l'application. Ce sont les périodes au cours desquelles il y a accès dense à l'application. L'ajout de ce module permettra à l'utilisateur de mieux choisir le temps de sa publication pour maximiser la visibilité de l'œuvre qu'il publie.