

## MGL7130 - Développement d'applications mobiles - Rapport Final

**Nom de l'équipe : Dream Team**

**Nom de l'application : CoopBiblio**

**Membres (3) :**

Nom	Code Permanent
Salim HARIZ	HARS17089101
Moez DRIRA	DRIM11119009
Dylan COHEN TANNUGI	COHD20049403

### Introduction :

Ce document est réalisé dans le cadre du cours MGL7130 (Développement d'applications mobiles), et présente un bilan de fermeture du projet réalisé tout au long de la session, durant laquelle on a réalisé une application mobile en utilisant Cordova.

Les objectifs suivants seront adressés dans la suite du document :

- Les modifications apportées lors de développement de l'application par rapport à ce qui a été planifié lors de la phase d'analyse.
- Le modèle d'affaire potentiel pour cette application.
- La réflexion sur le choix de l'architecture technologique mobile.
- Les leçons apprises lors de la réalisation de l'application.

### Modifications apportées :

Comparé au rapport d'analyse, nous avons effectué quelque changement. Tout d'abord, nous avons modifié notre stratégie hors-ligne. En effet, dans le rapport d'analyse, nous voulions laisser la possibilité à l'utilisateur d'utiliser la fonctionnalité d'affichage de la liste d'emprunt. Cependant, il a été décidé que notre application ne devrait pas être utilisée sans connexion, nous avons donc décidé d'afficher un message d'alerte si l'appareil n'est pas connecté à Internet.

Comme nous avons modifié notre stratégie hors-ligne, avoir une base de données local pour la liste des emprunts si l'appareil n'est pas connecté n'est plus nécessaire. Nous avons donc supprimé l'idée de cette base de données locale et avons uniquement gardé la base de données NoSQL de type document. Pour finir, l'interface générale de l'application n'a presque pas changé. Les couleurs principales sont le violet, le blanc et le gris et les icônes utilisées sont celles décrites dans le rapport d'analyse. La structure générale est elle aussi inchangé. Nous avons décidé de séparer l'application en trois parties selon les trois différentes fonctionnalités qui sont : l'emprunt d'un livre, le retour d'un livre et l'affichage des emprunts. Le seul changement général est l'emplacement des boutons de sélection des fonctionnalités. En effet, ceux-ci se trouve toujours en bas de l'écran, même lorsque l'application est en mode paysage. De plus, l'interface de chaque fonctionnalité a elle subit des changements.

### Interfaces d'emprunt d'un livre :

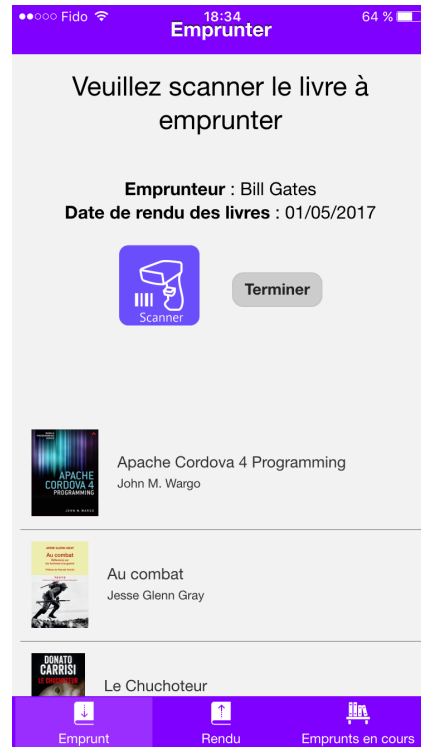


Image 1 : Interfaces de la fonctionnalité d'emprunt de livres sur mobile au format portrait

Contrairement à ce que nous avons décrit dans notre rapport d'analyse, la fonctionnalité d'emprunt ne se décompose que de deux parties. La première est restée inchangée et consiste à scanner la carte d'un membre pour l'identifier et ainsi pouvoir commencer l'emprunt de ses livres. La deuxième partie est un assemblage de la deuxième et troisième parties décrites dans le rapport d'analyse. Celle-ci permet de scanner les livres à emprunter. Elle affiche en plus le nom de l'emprunteur, la date de remise (calculée automatiquement) et la liste des livres qui viennent d'être scannés. Le bouton "Terminer" permet de terminer les emprunts pour ce membre et de revenir à la première partie. (Voir image 1).

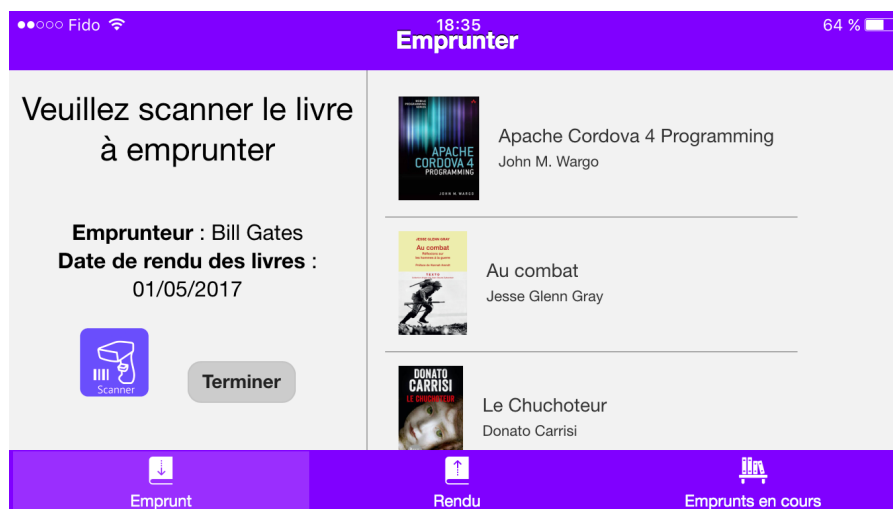
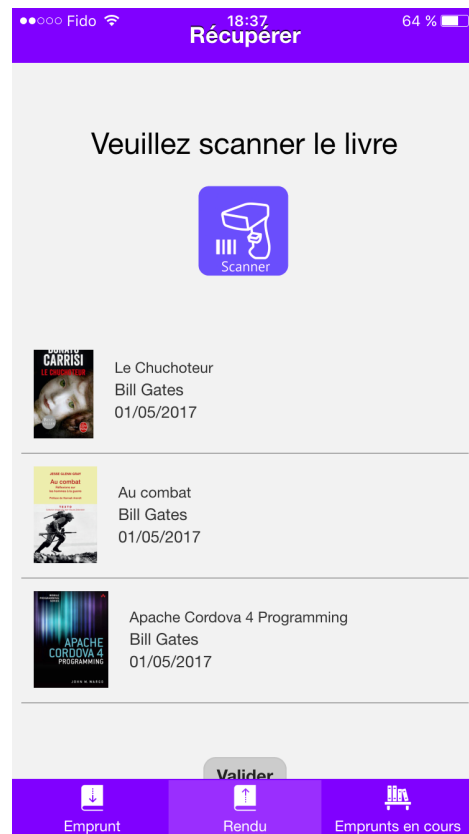


Image 2 : Interfaces de la fonctionnalité d'emprunt de livres sur mobile au format paysage

Le format paysage est quant à lui très semblable à l'idée d'origine que nous avons de cette interface. La seule différence est l'emplacement du nom de l'emprunteur et de la date de retour. En effet, initialement, nous pensions placer ces informations du côté droit avec la liste des livres. Mais, avec les changements de l'interface au format portrait et l'espace qui restait du côté droit de l'interface, nous avons finalement décidé de placer ces informations de ce côté. (Voir image 2).

*Interface de retour de livres :*



*Image 3 : Interfaces de la fonctionnalité de retour de livres sur mobile au format portrait*

Contrairement à ce que nous avons décrit dans le rapport d'analyse, l'interface de retour de livres est différente de celle de l'emprunt de livres. Celle-ci est composée d'une seule partie. Sur celle-ci, l'utilisateur peut scanner autant de livres que voulu. Il doit ensuite valider le rendu pour que celui-ci s'effectue. De plus, une liste affiche les livres scannés avant la validation. (Voir image 3).

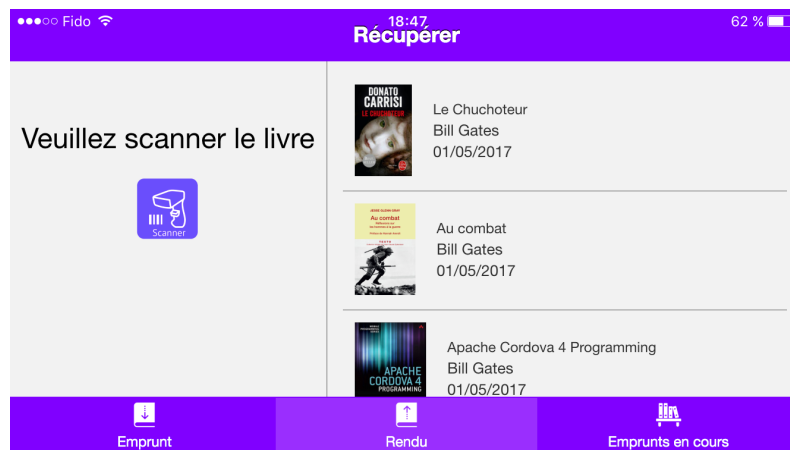


Image 4 : Interfaces de la fonctionnalité de retour de livres sur mobile au format paysage

L'idée pour l'interface au format paysage est quant à elle similaire à celle utilisée pour l'emprunt. En effet, l'interface est décomposée en deux parties. Une à droite contenant la liste des livres et une à gauche contenant le bouton de scan.

Interface d'affichage des emprunts :

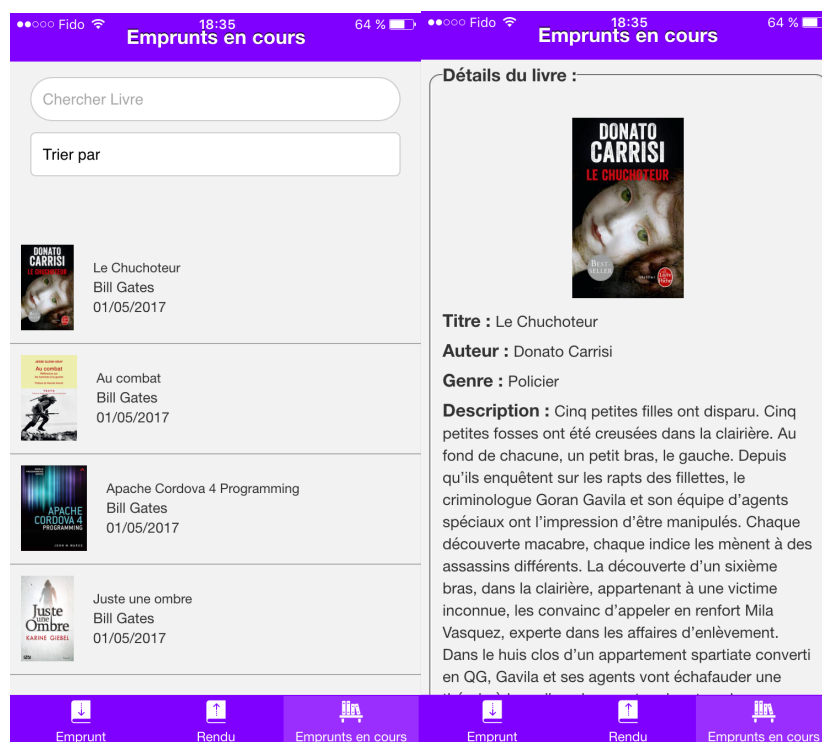


Image 5 : Interfaces de la fonctionnalité d'affichage des emprunts sur mobile au format portrait (à gauche : partie Maître, à droite : partie Détail)

Pour l'affichage des emprunts, nous avons finalement décidé d'utiliser le patron maître-détail. La partie maître contient donc la liste des livres ainsi qu'une barre de recherche et de tri. La partie détail contient les détails de l'emprunt séparés en trois parties : les détails du livre, les détails de

l'emprunteur et les détails de l'emprunt. (Voir Image 5).

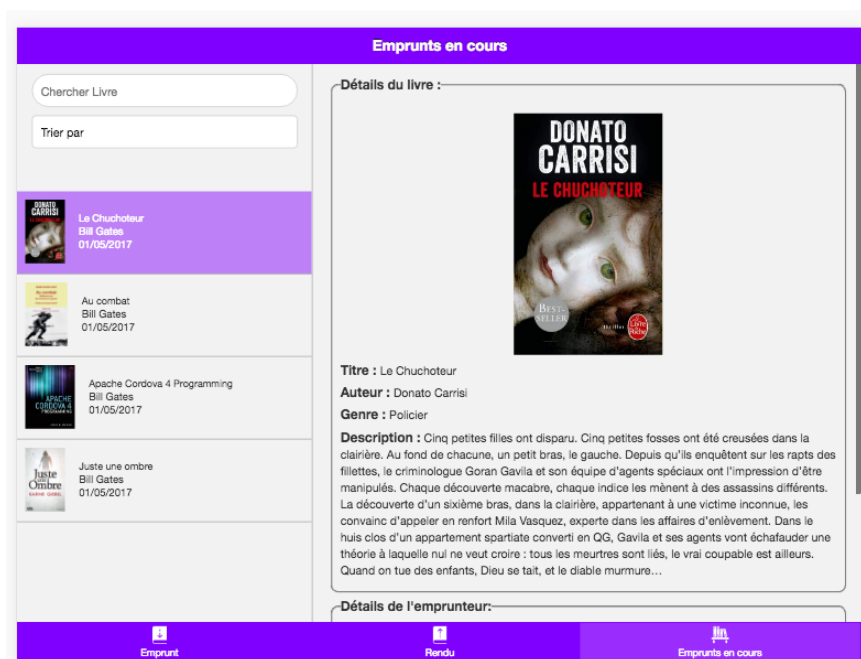


Image 6 : Interfaces de la fonctionnalité d'affichage des emprunts sur tablette au format paysage

Pour le format paysage pour les tablettes, nous avons donc aussi appliqué le patron maître-détail. C'est-à-dire que la partie de gauche de l'interface contient le maître et celle de droite le détail. Lorsque l'utilisateur sélectionne un des emprunts, la partie détail se met à jour pour afficher les détails correspondant. (Voir image 6).

Un autre changement subtil pour cette fonctionnalité, par rapport au rapport d'analyse, est l'apparence de la recherche et du tri. Dans le rapport d'analyse, ces fonctionnalités se trouvaient dans un menu pouvant être caché au format portrait et à droite de la liste des emprunts au format paysage. De plus, le tri se faisait à l'aide de bouton. Maintenant, ces fonctionnalités se trouvent toujours au-dessus de la liste des emprunts, que ce soit au format paysage ou portrait, et le tri se fait grâce à une liste de sélection. De plus, le tri ne peut être fait que selon le nom de l'emprunteur, le titre du livre ou la date de rendu.

#### Modèle d'affaire :

Le modèle d'affaire potentiel pour cette application est le modèle payant. Ce modèle est conforme à CoopBiblio car cette dernière est orientée vers un seul type de clients qui est le responsable des prêts de livres dans une bibliothèque. En effet, l'application ne sera utilisée que par les bibliothécaires, elle sera donc uniquement livrée pour les bibliothèques qui le demandent. En effet, l'installation de cette application demande une pré configuration pour se lier à la base de données de la bibliothèque afin d'avoir accès aux membres, aux livres et aux emprunts. De plus, cette application est destinée à donner un service bien précis de gestion des emprunts pour la bibliothécaire, il semble donc logique que la bibliothèque paye pour avoir accès à cette application.

### Réflexion sur l'architecture mobile :

Vu que cette application est destinée vers le marché Canadien en particulier et le marché Nord-Américain en général, il fallait avoir une application qui fonctionne sur les deux systèmes d'exploitation Android et iOS.

Apache Cordova permettait d'avoir une version de l'application sur les deux plateformes en développant un seul code source. De plus, l'application ne nécessite pas que des ressources et capteurs basiques et disponible sous Cordova. Le développement natif n'est donc pas nécessaire et l'utilisation de Cordova était donc optimal pour notre application.

### Leçons apprises :

- Nouvelles connaissances technologiques : durant la réalisation de ce projet, les membres de l'équipe se sont confrontés à des nouveaux principes et techniques. En effet, ce projet était une bonne occasion pour développer une "Single Page Application" et de manipuler les données nuagiques en utilisant la base de données DynamoDB d'Amazon Web Services et le service de stockage de fichiers Amazon S3.
- Durant ce projet, l'équipe a dû utiliser un des capteurs des appareils mobiles. Nous avons donc pu nous familiariser avec l'utilisation du lecteur de code barre utilisable grâce à un plugin Cordova.
- Le projet était aussi une occasion pour développer une application mobile hybride avec l'utilisation de Cordova. Certains membres de l'équipe sont familiers avec le développement d'applications mobiles natives Android. Ce projet a montré les différences de réalisation des deux types d'application et a permis d'appliquer les notions des patrons de design mobile (Master-Detail) et de l'expérience usager (le choix des couleurs, l'adaptabilité des interfaces et des éléments graphiques pour les différents systèmes et appareils).
- Le développement mobile en utilisant Cordova n'est pas simple car la détection des erreurs et des bugs, l'exécution et le test de l'application est plus compliqué que pour une application non-mobile. Normalement, pour savoir ce qui ne fonctionne pas dans un projet NodeJS et JavaScript, il faut regarder la console du navigateur web (fureteur) ou utiliser les instructions "alert" et "console.log". Seul l'instruction "alert" est utilisable sur mobile. De plus, notre application est basée particulièrement sur le scannage des codes à barres, donc il nous était impossible d'exécuter le scan sur le fureteur ou sur un émulateur mobile. C'est pour cela qu'il fallait, à chaque fois que le code était modifié, le compresser et l'uploader vers Phonegap pour tester l'application. Tout cela a provoqué une grande perte de temps en plus du processus répétitif et ennuyeux.

### Conclusion :

Ce rapport présente le bilan de fermeture de réalisation de l'application mobile "CoopBiblio" en utilisant le framework de développement mobile Apache Cordova. Essentiellement, ce rapport énumère les modifications apportées à l'application par rapport à ce qui a été planifié lors de la phase d'analyse, décrit le modèle d'affaire probable qu'on peut appliquer, justifie le

choix d'utiliser Cordova au lieu d'applications natives, et détaille quelques leçons apprises de ce projet.