



Rapport de stage

« Création d'une liseuse ePub accessible sous Android »



Licence 3^e année
Ingénierie informatique
Année universitaire 2015/2016

Stagiaire : Adrien Fallot
Maître de stage : M. Casson
Tuteur : M. Ed-Dbali

Remerciements

Je souhaite avant tout remercier David Casson, expert en accessibilité numérique, pour m'avoir proposé un stage au sein de la Direction des Systèmes d'Information (DSI) de l'université d'Orléans, ainsi que l'ensemble de son équipe pour m'avoir accueilli durant ces trois mois.

Il en est de même pour Fabienne Meducin et Philippe Godiveau, respectivement chargée de mission « handicap et accessibilité » et professeur de langue à l'Unité de Formation et de Recherche (UFR) de Lettres, Langues et Sciences Humaines (LLSH), pour leur expertise et leurs conseils tout au long de ce projet.

S'ajoute également à ces remerciements toutes les entreprises partenaires de ce projet qui ont permis son financement, à savoir : Adecco, Atos, Banque Polulaire, Crédit Agricole, Le Groupe La Poste, Caisse d'épargne, Groupama, Capgemini, Sogeti, SNCF, WorldLine, Thales, l'Université François Rabelais de Tours et enfin l'Académie d'Orléans-Tours.

Et enfin je tiens à remercier tout le personnel de la licence et l'université pour avoir été présents et aidant tout au long cette année de formation.

Table des matières

Remerciements.....	3
Introduction.....	5
1. Lieu de stage.....	6
1.1. Passerelle handicap.....	6
1.2. Direction des Systèmes d'Information.....	7
2. Présentation du sujet.....	9
2.1. Origines.....	9
2.2. Pourquoi Android ?.....	10
2.3. Pourquoi le format ePub ?.....	11
3. Présentation des objectifs.....	13
3.1. Analyse.....	13
3.2. Développement.....	15
3.3. Présentations et conférences.....	20
3.4. Résultat final et documentation.....	21
3.5. Perspectives.....	23
4. Bilan technique et humain.....	24
Annexes.....	25
A.1. Diagramme de GANTT.....	25
A.2. Fiche de comparaison des applications existantes.....	26
A.3. Maquette de l'accueil « connecté » et « invité ».....	27
A.4. Premier diagramme de classe.....	28
A.5. Premier MLD-R.....	29
Sitographie.....	30

Introduction

Dans le cadre de la troisième année de formation en licence spécialité ingénierie informatique à l'université d'Orléans, nous avons la possibilité d'effectuer un stage de treize semaines. Mon maître de stage, le référent accessibilité numérique de l'université, possède aussi bien des missions associées à la passerelle handicap qu'à la DSI, j'ai donc effectué mon stage au sein de cette dernière.

Ce stage m'a permis d'approfondir mes connaissances en accessibilité numérique. J'y ai appris les règles de présentation du texte pour certaines personnes atteintes de troubles spécifiques du langage (exemple la dyslexie : difficulté de lecture). Il faut notamment : ne pas justifier le texte, utiliser un interligne de 1,5, appliquer de préférence une police spécifique et placer deux espaces entre chaque mot. Mon stage portant sur l'accessibilité numérique et plus particulièrement sur ces troubles, c'est tout naturellement que j'ai adapté mon rapport en conséquence. Ce dernier est également disponible au format *electronic publication* (ePub).

Le stage porte sur la création d'une liseuse de livres électroniques – au format ePub – sous Android. Beaucoup d'applications du même type sont déjà disponibles sur le marché mais l'avantage de notre application est qu'elle permet à l'utilisateur de créer un compte puis d'y associer un profil de lecture modifiable. L'utilisateur peut changer les règles précédemment énoncées ainsi que les couleurs ou la taille du texte. Deux profils prédéfinis – dyslexique et malvoyant – ainsi qu'un compte invité sont configurés nativement.

1. Lieu de stage

Comme indiqué précédemment, j'ai réalisé mon stage pour le service « passerelle handicap » de l'université d'Orléans mais dans les locaux de la DSI.

1.1. Passerelle handicap

Elle est située à côté du centre culturel « Le Bouillon » et dans le même bâtiment que le restaurant universitaire « Le Forum ». Ceci la place au cœur d'un des lieux clés du campus, ce qui est un atout considérable au vu ses différentes missions.



Figure 1 : La passerelle handicap vue de l'extérieur

Elle se charge d'accompagner les étudiants en situation de handicap, permanent ou temporaire, tout au long de leur parcours universitaire. Durant les études en mettant en place différentes aides telles que des preneurs de notes, des interprètes ou des aménagements lors des examens comme une majoration de temps. Ainsi que pour effectuer des démarches administratives, trouver un logement adapté ou durant dans leur vie quotidienne (déplacements, prise des repas, etc.).

Elle s'occupe également de mettre à disposition des ordinateurs, d'adapter les différents documents pédagogiques (transcription en braille, agrandissement, etc.).

Le service possède également toute une batterie de matériels spécialisés qui m'ont permis d'en apprendre plus sur l'accessibilité en général. Par exemple un télé-agrandisseur qui permet d'afficher une feuille de papier de manière agrandie et d'en modifier avec les couleurs. Ces spécificités sont implémentés dans notre application.

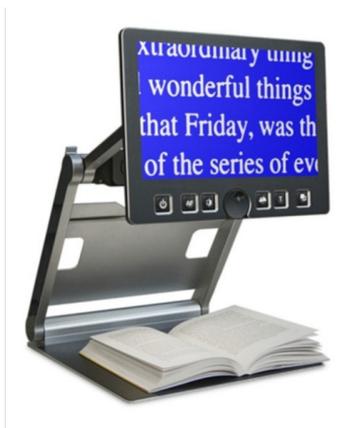


Figure 2 : Un télé-agrandisseur

1.2. Direction des Systèmes d'Information

La DSI se situe sur le campus, à quelques mètres du Laboratoire d'Informatique Fondamentale d'Orléans (LIFO) dans le bâtiment de l'Institut de Recherche pour le Développement (IRD).



Figure 3 : La DSI vue de l'extérieur

La DSI a pour mission de gérer les systèmes d'information de l'université. Que ce soit l'administration des serveurs, le dépannage de poste, la gestion des différentes bases de données, le maintien des applications métiers ou encore la carte Atout'Centre.

Voici un organigramme représentant l'organisation de DSI de l'Université d'Orléans

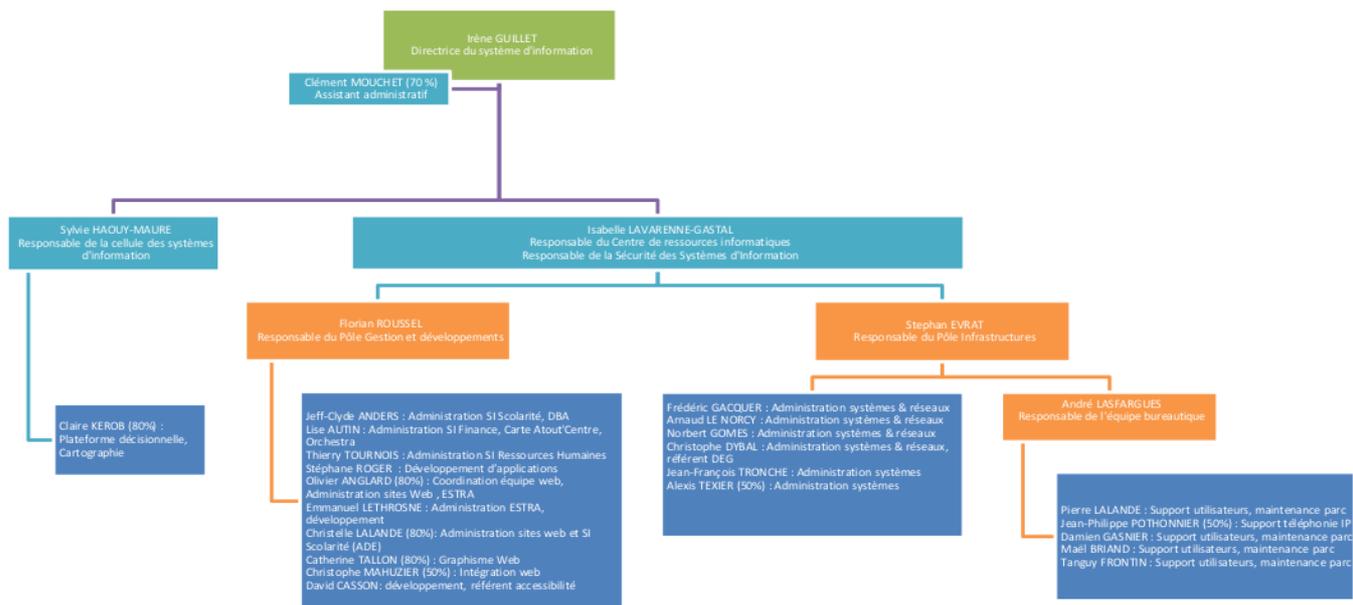


Figure 4 : Organigramme de la DSI de l'université d'Orléans

Comme mon maître de stage appartient du pôle gestion et développement, j'y ai donc effectué mon stage. La partie développement de ce pôle a pour mission la création et le maintien des applications web de l'université (par exemple : la consultation des emplois du temps en ligne, l'annuaire du personnel, la prise de rendez-vous préalable aux inscriptions, etc.).

2. Présentation du sujet

2.1. Origines

M. Casson a eu l'idée d'implémenter les normes d'accessibilité web dans le format ePub. Ce dernier est établi sur le langage de balisage *Hypertext Markup Language* (HTML), il peut ainsi s'adapter et répondre aux standard d'accessibilité (Référentiel général d'Accessibilité pour les Administrations (RGAA) basé sur les *Web Content Accessibility Guidelines 2.0* (WCAG 2.0)). Selon l'expérience utilisateur le support le plus approprié pour la lecture du format ePub s'avère être une tablette. Il a donc demandé au Dr Meducin, chargée de mission de la passerelle handicap, si ce développement pouvait s'inscrire dans la liste des projets financés par les entreprises en partenariat avec la passerelle handicap.

Après validation par ces dernières, le projet à pu débuter. Un cahier des charges, un diagramme de GANTT provisoire (voir annexe A.1.) ainsi que les premières maquettes furent réalisés. M. Godiveau, professeur de langue à l'UFR LLSH, spécialiste des troubles spécifique du langage rejoint l'équipe, pour apporter ses connaissances dans ce domaine.

Ayant déjà travaillé avec la passerelle, pour mon stage de Diplôme Universitaire de Technologie (DUT), dans un projet d'accessibilité web (MediaPlayerA11y : lecteur multimédia HTML5 accessible au clavier pour les personnes malvoyantes). Il m'a été proposé de rejoindre l'équipe pour assumer les missions de développement. Étant moi-même atteint de certains troubles spécifiques du langage, j'ai tout de suite été intéressé pour apporter mon expertise informatique au service des personnes en situation de handicap.

2.2. Pourquoi Android ?

Selon le graphique ci-dessous, nous pouvons constater que le système Android est le plus présent parmi les utilisateurs de *smartphone* avec plus de 80 % des parts de marché.

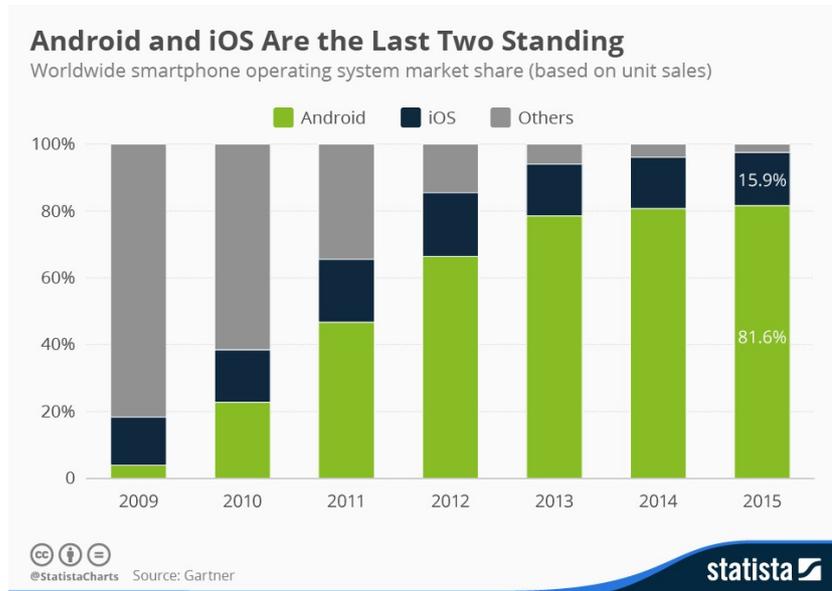


Figure 5 : Graphique représentant le marché mobile mondial entre 2009 et 2015

À l'aide du graphique suivant nous avons choisi la version KitKat (4.4). Ce choix nous permet d'être disponible au près de suffisamment d'utilisateurs – environ 70 % – et de ne pas se soucier des fonctionnalités qui ne sont pas implémentées dans les versions antérieures à celle-ci (comme les droits de stockage, et le HTML5 dans une *WebView*)

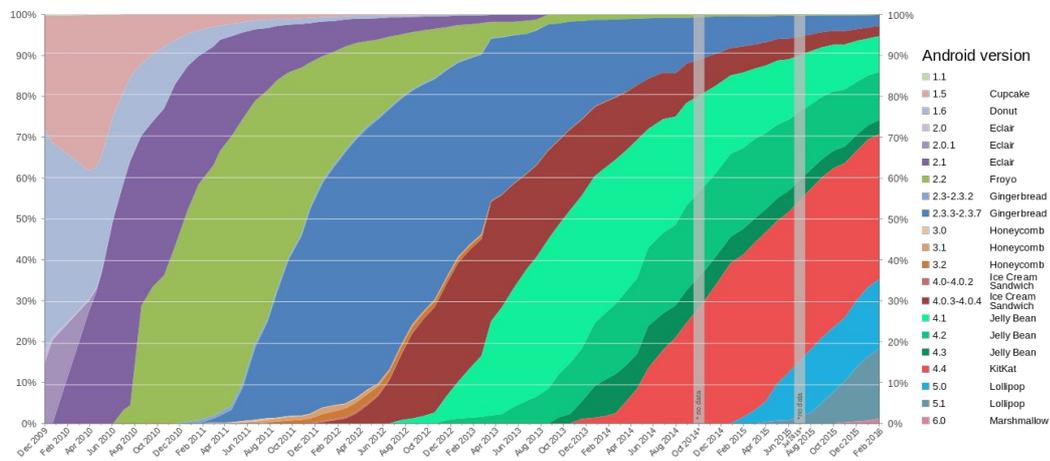


Figure 6 : Répartition des différentes version d'Android depuis décembre 2009

Nous pouvons également noter que le coût de développement plus faible ainsi que la connaissance de la technologie ont été des critères de décision. J'ai eu l'occasion d'apprendre la programmation sous Android et le langage Java durant ma formation à l'IUT. Un gros avantage de l'IDE Android Studio est qu'il est porté sur différents systèmes d'exploitation. Contrairement au développement sur iOS (Apple) qui demande un ordinateur Mac et une licence à payer chaque année. Cependant il est envisagé par la suite un portage sur iOS, ceci sera détaillé dans les perspectives.

2.3. Pourquoi le format ePub ?

Le format ePub est un format libre proposé par l'*International Digital Publishing Forum* (IDPF) destiné aux publications électroniques. Cette spécificité permet de ne pas demander d'autorisation d'utilisation aux ayant-droit.

Mais le plus gros avantage est sa structure. Nous pouvons voir sur le schéma ci-après, qu'un ePub est structuré comme un site internet. Il possède également certaines spécificités comme une table des matières, un fichier contenant les métadonnées (le(s) auteur(s), la date, la langue ou toute autres informations générales à l'ePub).

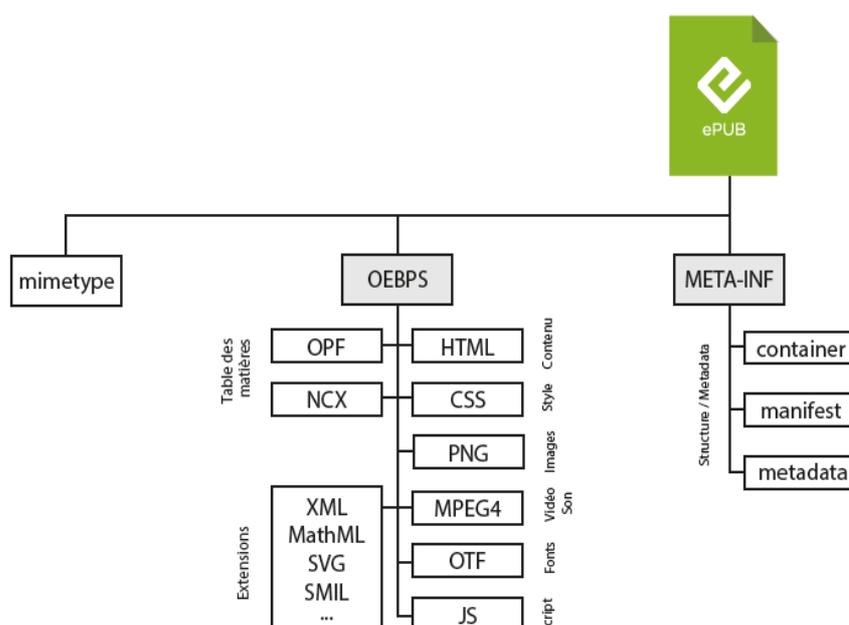


Figure 7 : Schéma représentant le contenu d'un .epub

Le format ePub est une archive comme pourrait l'être le zip ou le rar. À l'intérieur de celui-ci, se trouve du *Extensible HyperText Markup Language* (XHTML), *Cascading Style Sheets* (CSS) et JavaScript. Cette spécificité permet d'ouvrir ses chapitres dans un navigateur web. Ce qui se fait simplement sous Android pour peu que l'application ait décompressé l'archive auparavant. Nous le verrons ceci plus en détails dans la partie 3.2.

3. Présentation des objectifs

3.1. Analyse

Dans un premier temps j'ai commencé par prendre connaissance du projet. J'ai lu le cahier des charges rédigé par M. Casson avec l'aide de M. Godiveau. Dans celui-ci j'ai appris les spécificités de chacun des profils, de la fenêtre d'option et du fonctionnement des comptes utilisateurs. J'ai aussi pu également étudier les premières maquettes de l'application réalisées par M. Casson.

Ensuite, je suis passé à l'analyse de l'existant en téléchargeant plusieurs applications sur la tablette mise à disposition par la DSI. Après les avoir analysées, deux ont retenues mon attention : « PageTurner » et « EPUB3 Reader ». J'ai réalisé une fiche comparative listant les points jugés positifs et ceux à améliorer (document en annexe A.2.).

À partir de toutes ses informations, j'ai amélioré les maquettes de l'application en y ajoutant les fonctionnalités auxquelles on a pu penser durant la première semaine. Ces dernières sont très détaillées – il y a même deux versions en fonction de l'utilisateur, s'il est invité ou connecté (voir en annexe A.3. pour la différence) – du fait de leur immense utilité par la suite. En effet elles permettront de développer l'application plus rapidement.

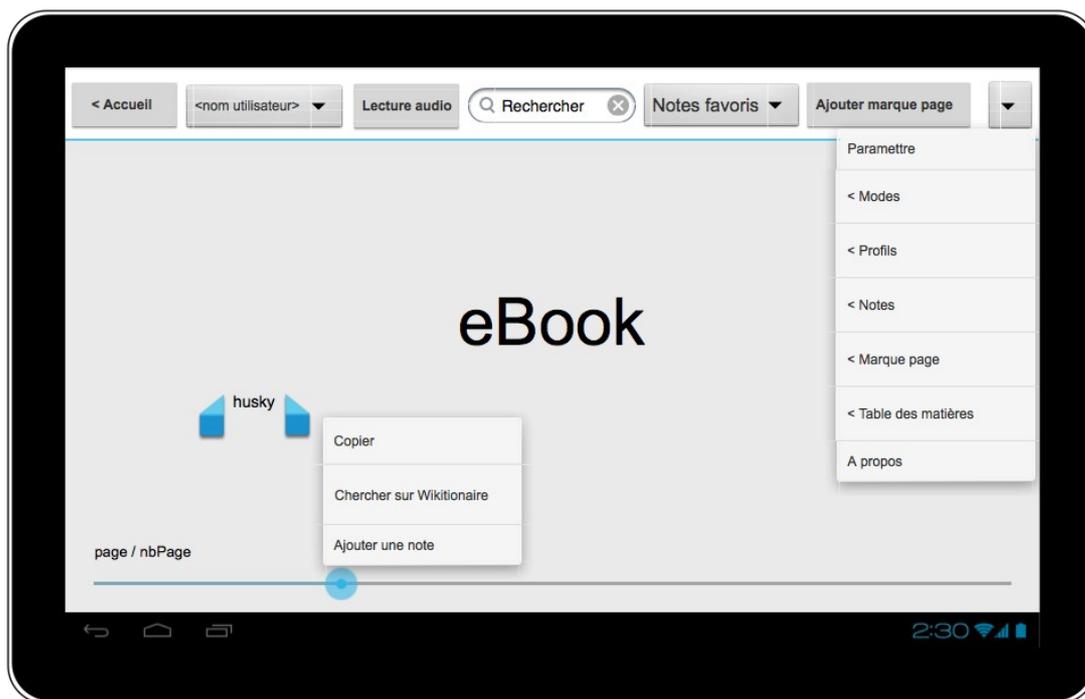


Figure 8 : Maquette de l'écran de lecture

Sur la maquette de l'écran de lecture de l'application nous pouvons y voir les fonctionnalités principales de ce celui-ci. Comme la possibilité de se déplacer de page en page à l'aide d'une barre de navigation, le changement du profil de lecture mais également le mode d'affichage – jour ou nuit.

Après les maquettes, je suis passé à la création de diagrammes détaillant la structure interne de l'application et de la base de donnée. J'ai tout d'abord réalisé un diagramme de classe (disponible en annexe A.4.) à l'aide du logiciel StarUML. Au cours de l'avancement du projet, l'organisation des classes de l'application à beaucoup changé. Par exemple le programme contient également beaucoup plus d'activités. La lecture et l'accueil qui sont maintenant séparés mais héritent désormais d'une activité abstraite qui contient les livres, l'utilisateur courant et de nombreuses fonctionnalités.

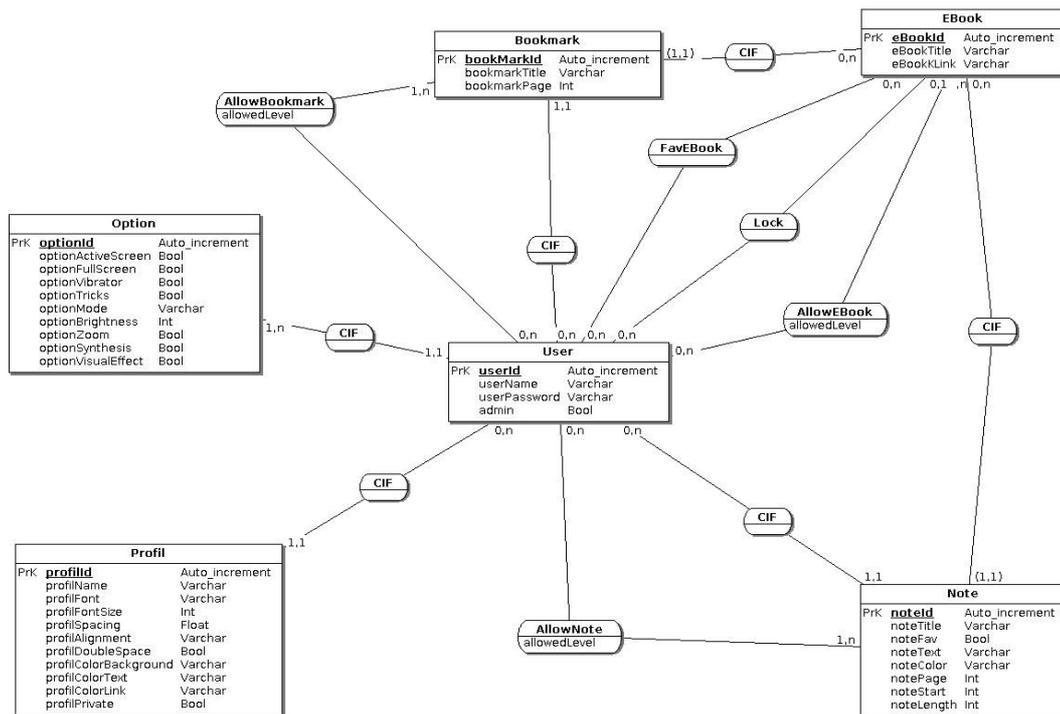


Figure 9 : Premier MCD de l'application

J'ai ensuite effectué le Modèle Conceptuel de Données (MCD) ci-dessus ainsi qu'un Modèle Logique de Données Relationnel (MLD-R) (disponible en annexe A.5.) à l'aide du logiciel JMerise pour me faire une première idée de la base de données. Au final, cette dernière n'est pas exactement comme ces schémas, car nous avons retiré la possibilité de verrouiller un eBook pour son compte ou encore de partager ses marques-pages. Mais nous avons rajouté la possibilité que l'utilisateur reste actif pour les personnes n'ayant qu'un compte sur l'application. Toute cette analyse a été très utile, durant celle-ci beaucoup de questions ont trouvé leurs réponses ce qui a permis d'avancer rapidement durant le début du développement.

3.2. Développement

Le premier objectif a été de trouver un moyen de convertir un fichier ePub stocké sur la tablette en classes Java utilisable par Android. Mon stage étant surtout axé sur les fonctionnalités ainsi que sur l'accessibilité, je me suis directement orienté vers la recherche d'une bibliothèque qui proposerait ces fonctionnalités. Après

quelques recherches j'en ai trouvé une : ePubLib. Celle-ci est très connue pour la manipulation d'ePub en Android ou en Java ce qui sera très pratique si j'ai besoin de précision sur une fonctionnalité. Pour créer un nouveau livre il faut utiliser la méthode *EpubReader.readEpub* avec comme paramètre un *InputStream* pointant sur le livre. Pour les besoins de cette application j'ai utilisé une surcouche pour stocker les livres nommés « EPubReaderA11yBook ». Celle-ci contient un indicateur pour savoir si ce livre est en favoris, connaître la date d'ajout et celle de lecture.

Après avoir réussi à ouvrir un livre en Java, j'ai essayé d'afficher ses détails directement dans l'application en suivant les maquettes pensées précédemment. Pour ce faire, ePubLib nous propose de récupérer le titre avec la méthode *getTitle()* ou les auteurs avec la fonction *getAuthors()*. Je les ai utilisées pour remplir des champs textes situés dans la vue écrite en *Extensible Markup Language* (XML), ceci à très bien marché au premier abord. Mais quand j'ai commencé à implémenter les *pop-up*, pour afficher les détails d'un livre, j'ai eu un problème lors du passage de ses informations d'une activité à l'autre. En effet pour ce faire il faut utiliser le système d'*intent* et de *bundle* à l'aide de la sérialisation d'un objet. Or les classes *Uri* et *Context* contenu dans *EPubReaderA11yBook* ne peuvent pas l'être. Toutes les solutions me venant à l'esprit me demandait soit de recréer l'objet (ce qui prend du temps) ou directement la position du livre pour le retrouver plus tard (en allant le chercher dans la première activité). Mais étant convaincu par aucune de ces solutions peu élégantes. J'ai effectué quelques recherches et j'ai trouvé la solution : le *dada-binding* du XML sous Android.

Ce dernier permet de créer des variables dans le XML, comme ci-après, pour les utiliser par la suite.

```
<data>
  <variable name="book" type="com.ally.and.dev.epubreaderally.EPubReaderA11yBook"/>
</data>
```

Figure 10 : Déclaration d'une variable de data-binding dans le XML

Nous pouvons ensuite s'en servir directement dans les autres élément du XML – tel qu'un texte. Par exemple pour accéder aux auteurs nous pouvons utiliser *book.authors*, nous avons aussi la possibilité de faire des conditions – comme sur la capture d'écran si dessus avec une condition ternaire.

```

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@{book.authors != "" ? @string/by + " " + book.authors : @string/unknownAuthor}"
    android:id="@+id/authorEBookItem_TextView"
    android:textSize="20sp"
    android:textColor="?colorTextBookAuthor" />

```

Figure 11 : Utilisation d'une variable de data-binding dans le XML

Et enfin pour ajouter le livre dans la variable du XML il suffit d'ajouter les lignes suivante dans le code Java.

```

//init item layout using data-binding (http://developer.android.com/tools/data-binding/guide.html)
EbookShelfItemBinding binding = DataBindingUtil.inflate(vi, R.layout.ebook_shelf_item, container, false);
binding.setBook(EBOOKS.get(i));
View item = binding.getRoot();

```

Figure 12 : Initialisation d'une variable de data-binding dans le Java

J'ai utilisé le *data-binding* dans pratiquement toutes les vues ayant du texte dynamique durant ce projet, car ce dernier permet une simplification de code Java ainsi que le passage de variables à fonction prédéfini par Android.

Je me suis ensuite attelé à la lecture de l'ePub. Celui-ci contenant des pages XHTML, je me suis directement orienté vers les *WebView* d'Android qui permettent d'afficher du code HTML en utilisant un navigateur interne à l'application. ePubLib propose la méthode *book.getSpine().getResource(<numéro de page>)* pour récupérer un des chapitres du livre. Ceci a très bien marché jusqu'au moment où il y a fallu pouvoir charger des images ou du CSS. Ces derniers sont normalement affichés à l'aide d'un lien relatif situé dans le code HTML et menant dans l'archive, sauf qu'Android n'a pas accès au contenu de celle-ci. La solution a donc été de copier ces ressources dans le dossier de stockage temporaire de l'application accessible via la méthode *getCacheDir().getAbsolutePath()*. Il a ensuite fallu changer les liens dans le fichier HTML pour qu'ils pointent vers cette copie.

À partir de ce moment il a donc été possible d'implémenter les profils. Les deux profils prédéfinis ont été réalisés sous la forme de fichiers CSS présent dans le dossier *assets*. Pour y faire référence j'ai ensuite ajouté une nouvelle ligne (balise *link* pour CSS) dans la page HTML ayant pour lien « *file:///android_asset/<nom du profil>.css* ». Les caractéristiques de ces profils sont les suivantes :

```
* {
  font-size: 14pt;
  line-height: 1.5;
  word-spacing: 0.5em;
  text-align: left;
}
```

Figure 13 : Fichier CSS pour le profil dyslexique

```
* {
  font-size: 18pt;
  line-height: 1.5;
  word-spacing: normal;
  background-color:black;
  color:white;
}
```

Figure 14 : Fichier CSS pour le profil malvoyant

Pour les profils créés par les utilisateurs, j'ai développé une nouvelle activité permettant de choisir les paramètres suivant : taille de la police (de 0 à 100), alignement du texte, l'interligne (simple, 1,5, double), le double espacement entre les mots ainsi que la couleur (le *ColorPicker* viens du projet open-source *Ambilwarna*). Après les avoir récupérés, ses valeurs sont ajoutées dans un fichier CSS ayant le nom choisi par l'utilisateur. Ce fichier est enregistré dans le dossier de stockage interne de l'application – accessible via la fonction *getFilesDir()*.

Cette activité contient une zone d'aperçu en temps réel. Pour ce faire chacun des éléments possède un *listener* personnalisé qui détecte chaque changement de valeur. Une fois cette détection effectuée, une page web contenant un texte d'exemple et les valeurs actuelles – sous forme de CSS – est créée. Celle-ci est ensuite chargée dans la *WebView* de la zone d'aperçu. De plus dans cette activité nous pouvons autoriser ou non les autres utilisateurs de l'application à utiliser notre profil. Il a donc fallu pouvoir associer un profil et un utilisateur. Et pour ce faire le prochain objectif a donc été de développer la base de données de l'application.

Je me suis servi la fonction de génération de script *Structured Query Language* (SQL) de JMerise. Mais j'ai quand même dû effectuer des ajouts de nouvelles tables ainsi que des modifications pour qu'il fonctionne avec la version de SQLite utilisé par Android. Les requêtes on était réalisées à l'aide des fonctions *insert*, *delete*, *update* et *query* de la classe *SQLiteOpenHelper* présente dans Android.

La base de données permet de stocker – à l’aide de leur *Uniform Resource Identifier* (URI) – les livres déjà ouverts dans l’application. Mais également de créer des comptes utilisateurs contenant des mots de passe cryptés (stocké sous forme de « hash » à l’aide de l’algorithme SHA-256). Ces comptes permettent d’enregistrer les profils et les marque-pages que l’utilisateur a créés. Elle permet aussi de mettre en favori les livres pour les retrouver plus facilement à l’aide des fonctions de tri (le dernier utilisé est également stocké). Ou encore d’enregistrer la page où utilisateur a arrêté sa lecture. Et enfin la base de données possède une fonction « se souvenir de moi » qui permet aux personnes de se reconnecter automatiquement même après arrêt de la tablette.

Un des objectifs était de réaliser deux thèmes pour l’application un « jour » et un « nuit ». N’ayant pas connaissance de la méthode à utiliser j’ai dû chercher dans la documentation. J’y ai appris qu’Android possède un système d’attribut. Dans un fichier *attr.xml* nous pouvons déclarer plusieurs variables, celles-ci peuvent être, une couleur, une image, un *layout*, etc. Il faut ensuite initialiser ces variables dans le fichier *style.xml* comme ci-après.

```

<style name="EpubReaderAllly" parent="Theme.AppCompat.Light">
  <item name="colorPrimary">@color/colorPrimary</item>
  <item name="android:textColorPrimary">@color/textColorPrimary</item>
  <item name="colorAccent">@color/colorAccent</item>
  <item name="android:colorBackground">@color/colorBackground</item>
  <item name="android:textColor">@color/colorText</item>
  <item name="android:alertDialogTheme">@style/dialog</item>
  <item name="android:editTextColor">@color/colorText</item>

  <item name="android:navigationBarColor" tools:targetApi="lollipop">@color/colorNavigationBar</item>
  <item name="android:statusBarColor" tools:targetApi="lollipop">@color/colorStatusBar</item>

  <item name="android:actionOverflowButtonStyle">@style/MyActionButtonOverflow</item>
  <item name="bookmarkIcon">@drawable/bookmark</item>
  <item name="favoriteIconInactive">@drawable/fav_icon_inactive</item>
  <item name="favoriteIconActive">@drawable/fav_icon_active</item>
  <item name="activeIcon">@drawable/active</item>

  <item name="customButton">@drawable/custom_button_day</item>
  <item name="customEditText">@drawable/custom_edit_text_day</item>
  <item name="customBorder">@drawable/custom_border_day</item>
</style>

```

Figure 15 : Code du thème « jour » dans le fichier *style.xml*

Dans les fichiers XML définissant les vues, nous pouvons accéder à ces valeurs avec la syntaxe suivante *?<nom de l'attribut>*. Ces dernières proviennent

automatiquement du thème actif. Celui-ci peut être changé depuis le code Java à l'aide de la méthode *setTheme()*.

3.3. Présentations et conférences

Le 8 juin, nous avons présenté l'application auprès des entreprises partenaires. Nous leur avons montré ce qu'il était possible de faire avec cette première version de l'application. Les représentants de celles-ci ont été enthousiasmés par cette présentation. Une personne du rectorat nous a proposé de mettre à disposition cette application dans les écoles les lycées. Puisque suite au « plan numérique » annoncé le 7 mai, certains seront équipés de tablettes d'ici la rentrée 2016 – soit environ 175 000 élèves.

Durant mon stage j'ai eu l'opportunité d'assister à deux journées de conférence sur l'accessibilité numérique. La première, le 30 mai, a été le 10^e Forum Européen de l'Accessibilité Numérique à la Cité des sciences et de l'industrie de Paris. Celui-ci a eu pour thème « L'accessibilité numérique dans un monde connecté ». J'ai pu assister à des tables rondes ou des présentations – en français et en anglais – sur l'accessibilité dans les objets connectés. Durant ce colloque nous avons rencontré des responsables de l'accessibilité numérique chez Microsoft France. Ces derniers nous ont proposé d'aller à la journée « Handicap, et alors ? Quelle accessibilité numérique ? » organisée dans leurs bureaux le 22 juin. Durant celle-ci nous avons pu constater que les éditeurs de manuels scolaires ne peuvent pas répondre à toutes les normes d'accessibilité du fait de leur très grande diversité. Notre application serait une solution efficace, une seule version du livre au format ePub – sur les tablettes fournies par le rectorat – avec un profil par élève.

Ces journées ne m'ont pas vraiment appris de chose en rapport avec le développement sur Android, mais ont été très enrichissantes et sensibilisantes sur l'accessibilité dans le numérique.

3.4. Résultat final et documentation

À ce jour l'application contient environs 90 % des fonctionnalités demandées. À savoir l'ajout de livres dans une bibliothèque – en mode étagère et liste –, la mise en favoris et les tris.

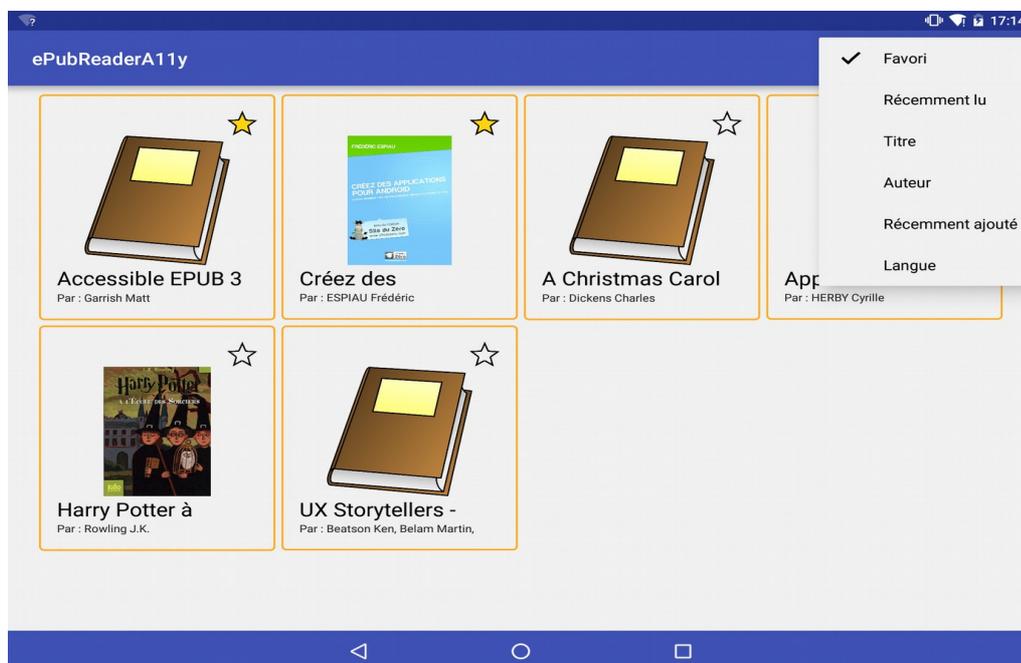


Figure 16 : La bibliothèque en vue « étagère » avec le thème « jour »

Ainsi que le fait de lire un livre, en y ajoutant le profil de son choix. Il y a également possibilité de se déplacer à l'aide de la table des matières et d'une barre de navigation



Figure 17 : Lecture avec un profil personnalisé et le thème « nuit »

Et enfin la création de compte, et la sauvegarde dans celui-ci de plusieurs paramètres comme le dernier tri effectué, la mise en favori, les profils, les marques-pages et la dernière page lue.

Parmi les fonctionnalités manquantes se trouvent la prise de note – en surlignant le texte dans une certaine couleur – ainsi que le déplacement page par page. Celui-ci est actuellement de chapitre en chapitre et demande donc de *scroller* ce qui peut s'avérer désagréable pour certains utilisateurs. Nous avons pensé mettre des astuces – sous forme de *pop-up* – lors de la première utilisation d'une fonctionnalité mais ceci n'a malheureusement pas été implémenté par manque de temps. Les options telle que la lecture en plein écran et le réglage de la luminosité ne sont également pas disponibles dans la version actuelle.

Une grosse partie du temps de développement a été utilisé pour la rédaction d'une documentation en anglais – sous le format JavaDoc. Pour ce faire il a fallu annoté toutes les fonctions et les classes de commentaires spécifiques. De plus comme vous pouvez le voir sur le code ci-dessous j'ai également commenté beaucoup des lignes dans les méthodes en elles-mêmes.

```
/**
 * Load default profile and "create profile"
 *
 * @param profileMenu the root menu
 */
private void loadDefaultProfilesItem(Menu profileMenu)
{
    //Add "create new profile" to menu
    profileMenu.add(getResources().getString(R.string.createNewProfile));
    MenuItem itemCreate = profileMenu.getItem(0);
    //Add listener
    itemCreate.setOnMenuItemClickListener((item) -> {
        createProfile(item);
        return false;
    });

    //toggle create new profile
    if (CURRENT_USER == null) {
        itemCreate.setVisible(false);
    }
    else {
```

Figure 18: Exemple de documentation et de commentaire de code

D'ici la fin du stage je pense également refaire un diagramme de classe et un MCD reprenant la structure réelle actuelle. Toute cette documentation sera très utile pour le futur de cette application. La personne qui reprendra le code plus tard pourra le

comprendre beaucoup plus vite et simplement.

3.5. Perspectives

Un des objectifs futurs est l'ajout une fonction de *text-to-speech* (TTS) qui permettrait la lecture à haute voix du texte. Mais tout d'abord nous voulons rendre cette application compatible à 100 % avec TalkBack, le lecteur d'écran natif d'Android. (C'est est un logiciel destiné aux personnes ne pouvant pas voir l'écran, il retranscrit par synthèse vocale et/ou sur un afficheur braille ce qui y est affiché.)

Après que la version Android soit sortie, nous envisageons également de porter l'application sous iOS. Ceci pour deux raisons, tout d'abord parmi les déficients visuels beaucoup utilisent ce système pour son ergonomie et son accessibilité. Il serait dommage de pas proposer ce projet à ces personnes. La deuxième raison concerne notre possible déploiement auprès de l'éducation nationale. Dans le « plan numérique » il est dit que les établissements doivent s'équiper de tablette mais n'est pas précisé si celles-ci seront sous Android ou iOS. Les élèves de certains établissements n'auraient donc pas accès au projet, ce qui ne respecte pas la loi de 2005 sur l'égalité des chances.

4. Bilan technique et humain

Actuellement l'application n'est pas disponible sur le *PlayStore*. Du fait des améliorations qui reste à y faire. Mais nous espérons que celle-ci le soit – en version beta – pour la fin de l'année.

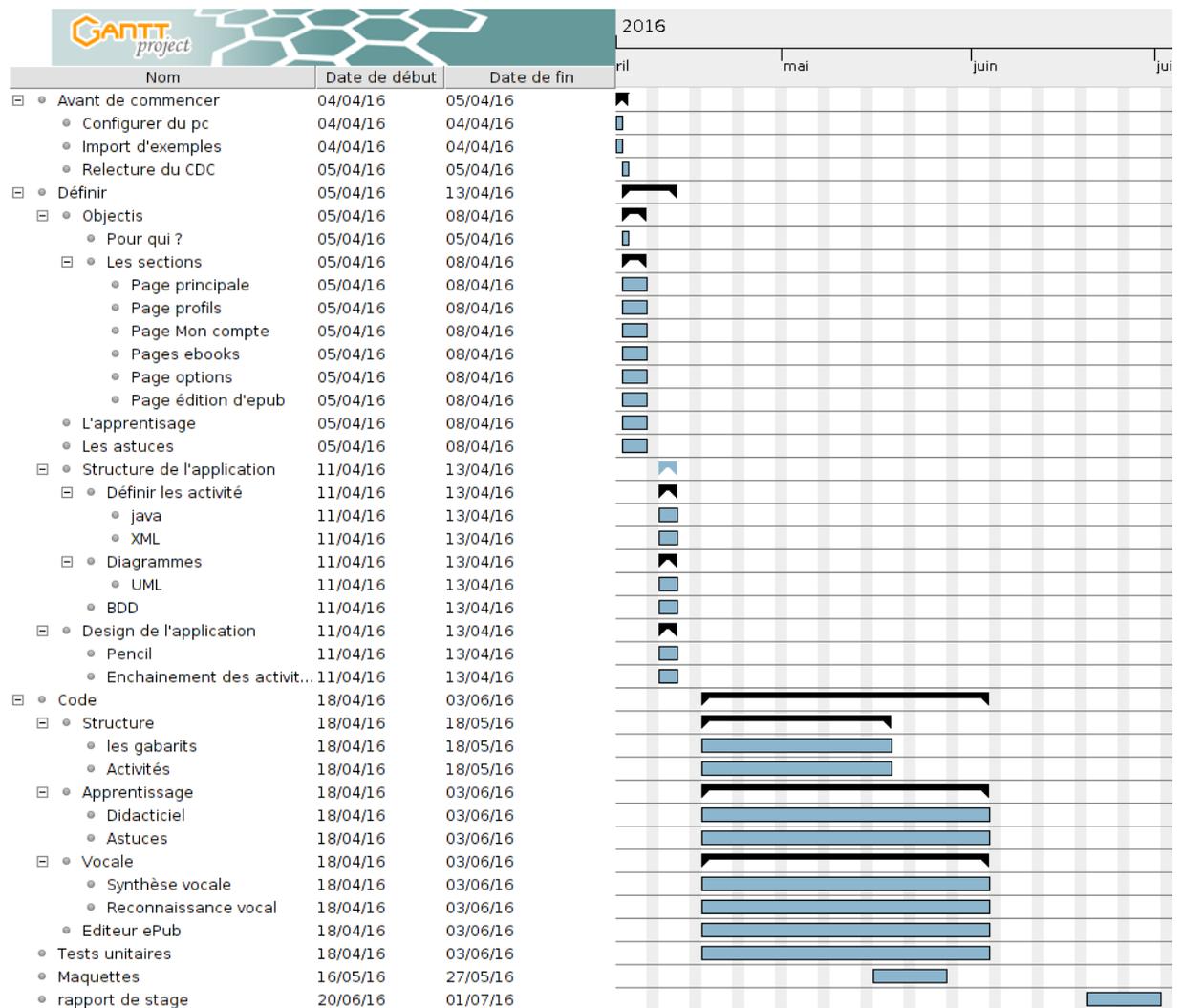
Durant ce projet j'ai pu mettre à profit mes compétences en Java et Android apprises tout le long de ma formation. Mais j'y ai également appris énormément sur la programmation en Android – par exemple le *data-binding* ou le fonctionnement des *DialogFragment*. J'ai pu également approfondir mes connaissances en accessibilité – CSS et Android. Le développement mobile ainsi que l'accessibilité numérique sont deux domaines en plein essor, en connaître les spécificités dès maintenant me donnera un plus non négligeable par la suite.

Ce stage m'a également permis d'utiliser mes connaissances en gestion de projets, en réalisant plusieurs diagrammes de modélisation. Ainsi que lors des réunions quotidiennes d'avancement avec mon maître de stage. J'ai pu également y découvrir la vie en entreprise, que ce soit tous les jours, en réunion ou en colloques.

Avec ce projet, nous espérons ainsi promouvoir le plus possible l'accessibilité numérique qui est très importante et pas forcément très compliqué à mettre en place. Uniquement 5 min de temps de développement peut économiser plusieurs heures de perdue à une personne handicapée. C'est pour cela que je vous encourage dès à présent à y penser. Et n'hésitez surtout pas à parler de cette application ou de l'accessibilité en général autour de vous pour que d'autres prennent conscience de son importance.

Annexes

A.1. Diagramme de GANTT



A.2. Fiche de comparaison des applications existantes

EPub3 reader :

- + possibilité d'afficher 2 livres à la fois
- + modification de l'affichage comme on veut (mais menu pas clair)

-+ plein d'options pour synchroniser les 2 livres mais pas d'explication clair sur leur fonctionnement

- pas d'erreur ni de proposition de charger un ePub s'il y a rien de charger
- charge uniquement les ePub du dossier « Download »

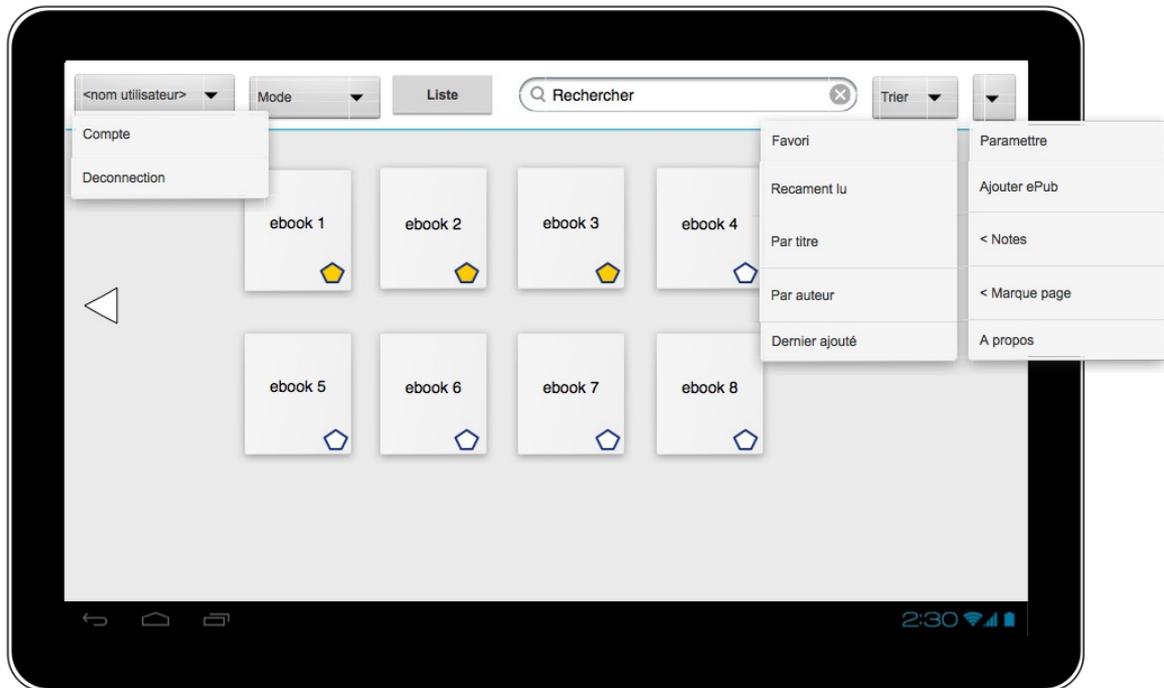
PageTurner :

- + sélection d'ePub dans un explorateur
- + possibilité de défilement vertical et horizontal
- + affichage bibliothèque et liste
- + ajout de marque-page
- + ajout de soulignement de plusieurs couleurs (possibilité d'y aller directement)
- + mode nuit
- + synchro avec un cloud
- + possibilité de chercher un mot dans un ePub
- + possibilité de télécharger des ePub en ligne
- + trier les livres par titre ou auteur
- + afficher uniquement les livres jamais lu
- + changement de langue de l'application
- + possibilité de lancer auto du dernier livre
- + possibilité de chercher du texte directement sur Wikipédia, Wikitionnaire
- + possibilité de partager du texte

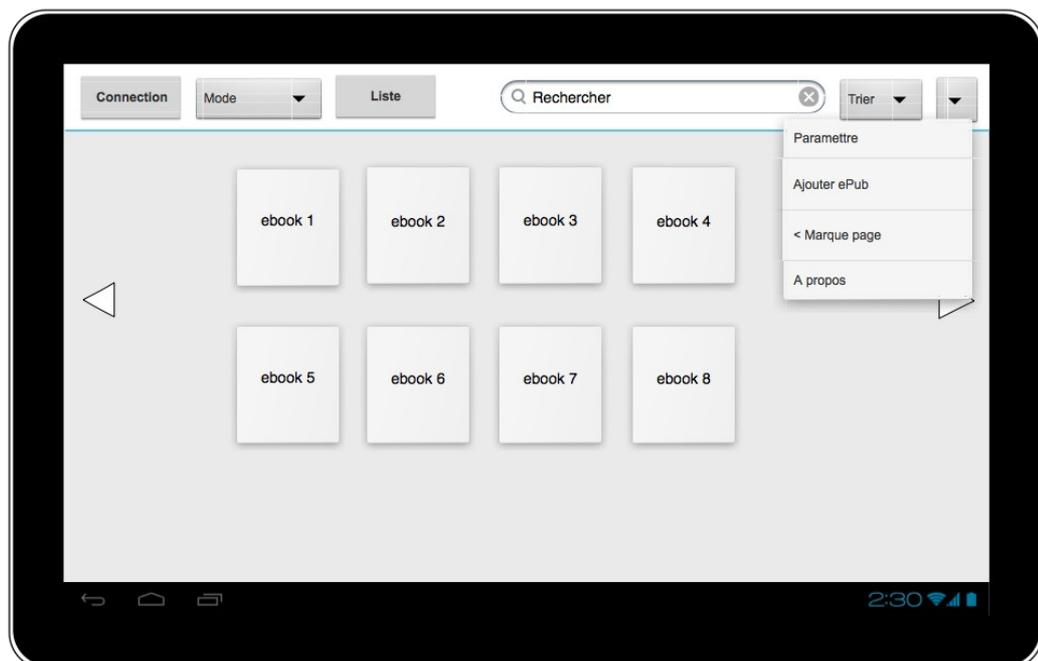
-+ possibilité de modifier tout l'affichage de l'ePub mais uniquement dans les options générales

A.3. Maquette de l'accueil « connecté » et « invité »

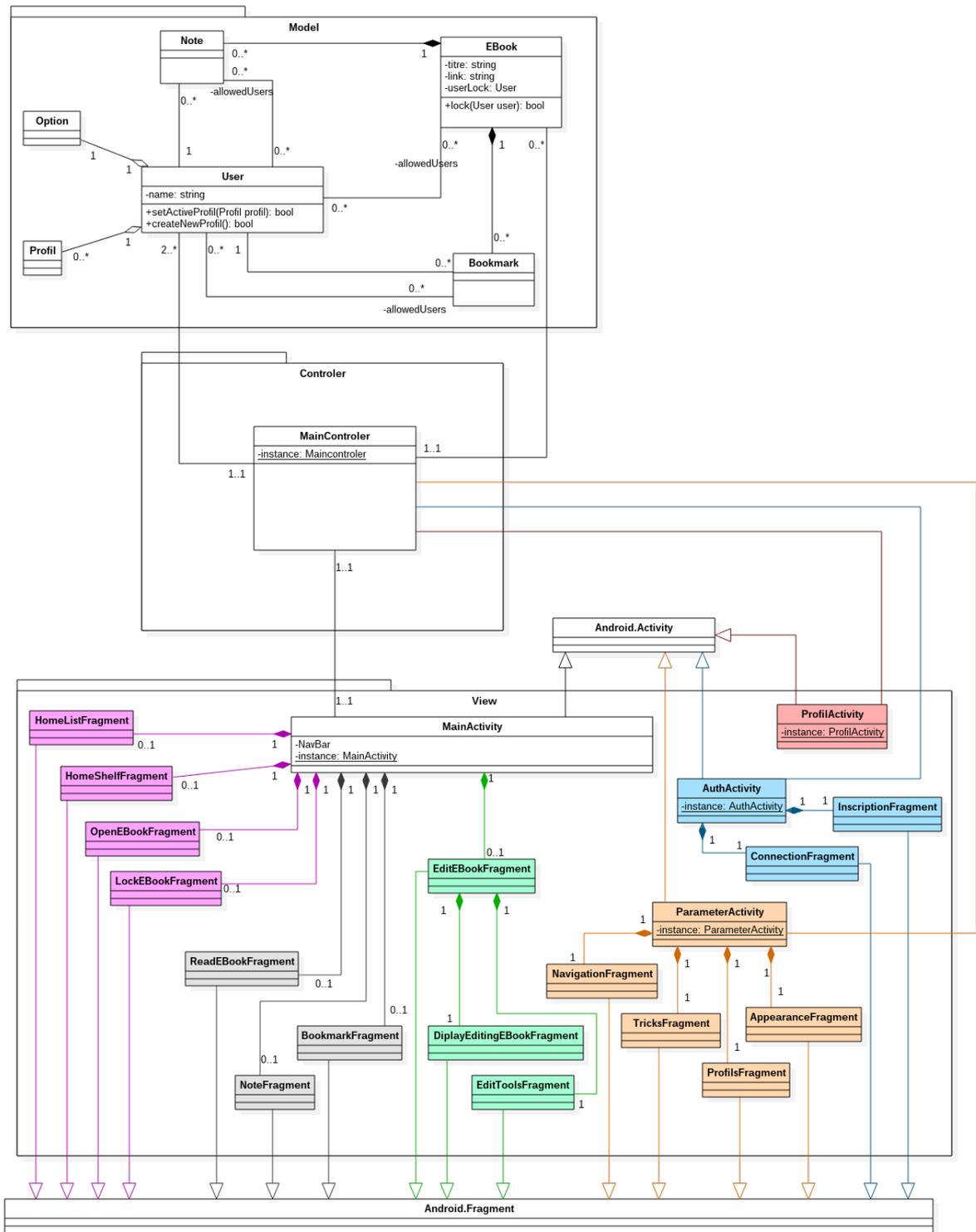
Version « connecté » :



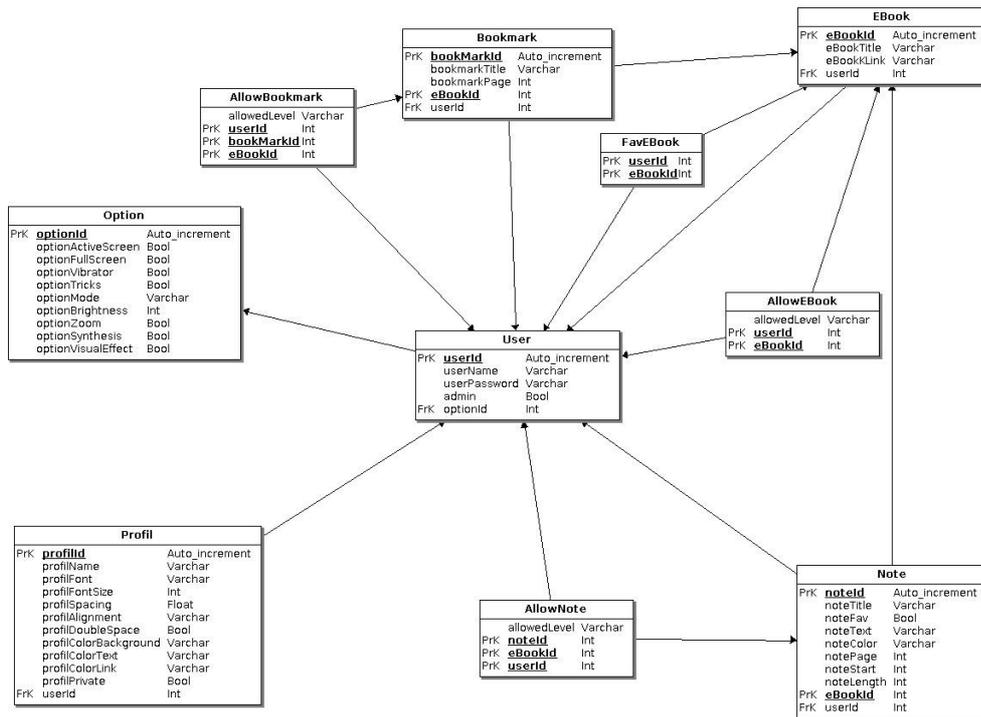
Version « invité » :



A.4. Premier diagramme de classe



A.5. Premier MLD-R



Sitographie

Image « télé-agrandisseur » : http://magasin.avh.asso.fr/270-thickbox_default/visio-bookteleagrandisseur-portable.jpg

Image « marché mobile mondial entre 2009 et 2015 » :
https://d28wbuch0jlv7v.cloudfront.net/images/infografik/normal/chartoftheday_4431_smartphone_operating_system_market_share_n.jpg

Image « Répartition des différentes version d'Android depuis décembre 2009 » :
https://upload.wikimedia.org/wikipedia/commons/thumb/e/ee/Android_historical_version_distribution_-_vector.svg/1280px-Android_historical_version_distribution_-_vector.svg.png

Image « Schéma représentant le contenu d'un .epub » : <http://cerig.pagora.grenoble-inp.fr/memoire/2014/images/structure-epub.gif>