



Développement d'un couple d'applications bureau et mobile

Génération et lecture de QR Codes pour l'institut Montéclair

David DEMBELE
Alassane DIOP
Jules LEGUY
Rahmatouwalet MOHAMEDOUN

Décembre 2017

Table des matières

1	Introduction	2
1.1	Contexte	2
1.2	Cahier des charges	2
2	Organisation	3
2.1	Répartition des tâches	3
2.2	Planning	3
3	Conception	4
3.1	Types de QR Codes	4
3.2	Représentation des données	4
3.3	Chargement et modification de QR Codes	4
4	Réalisation	6
4.1	Application de bureau	6
4.1.1	Architecture	6
4.1.2	Modèle	6
4.1.3	Vue	6
4.1.4	Contrôleur	6
4.2	Application mobile	6
4.2.1	Application existante	6
4.2.2	Adaptation aux nouveaux QR Codes	6
4.2.3	Téléchargement Drive	6
4.3	Contraintes et choix	6
5	Conclusion	7

Chapitre 1

Introduction

1.1 Contexte

Dans le cadre du module Concrétisation Disciplinaire de notre première année de Master 1 mention Informatique à l'Université d'Angers, nous avons développé une application de bureau et poursuivi le développement d'une application mobile.

Ces applications répondent à une demande de l'Institut Montéclair à Angers, dans le but d'apporter une dimension ludique à l'enseignement pour les mal-voyants et non-voyants inscrits à l'institut.

Ce projet a été encadré par Corentin TALARMAIN et Thomas CALATAYUD, deux étudiants en deuxième année de Master, dans le cadre de leur module Gestion de Projet.

L'objectif du projet est de fournir la possibilité aux transcripteurs de l'institut un moyen de générer des QR Codes contenant du texte et des sons et pouvant être interprétés par une application mobile. Une première version de l'application mobile avait déjà été développée par Corentin TALARMAIN lors de son TER¹ de fin de première année de Master. Celle-ci fournissait déjà la possibilité de lire avec une voix de synthèse du texte contenu dans des QR Codes. Nous avons donc développé une application de bureau permettant de générer ces QR Codes selon les contraintes décrites dans le cahier des charges. Nous avons également adapté l'application mobile existante à notre nouveau format de données.

1.2 Cahier des charges

Nous avons structuré notre projet pour répondre aux deux demandes principales qui nous ont été faites. La première demande était de pouvoir utiliser les QR Codes comme légendes de cartes de géographie. Cela implique de pouvoir gérer la lecture de QR Codes adjacents dans un ordre prédéfini (LIEN FAMILLES).

L'autre demande était de pouvoir utiliser des QR Codes pour lire des sons dans les pages d'un livre, sans avoir nécessairement accès à internet au moment de la lecture. Pour cela, un ou plusieurs QR Codes sur la page de couverture doivent permettre le téléchargement de tous les sons contenus dans le livre (LIEN ENSEMBLE).

De plus, un espace central contenant du texte en braille devait pouvoir être inséré au centre des QR Codes, pour permettre aux non-voyants de les localiser (LIEN GENERATION QR).

1. Travail Encadré de Recherche

Chapitre 2

Organisation

2.1 Répartition des tâches

2.2 Planning

Chapitre 3

Conception

3.1 Types de QR Codes

3.2 Représentation des données

Les QR Codes devant stocker plus d'informations que dans la version initiale de l'application mobile, il nous a fallu définir une structure de données pour les représenter. Cette représentation s'est construite parallèlement à l'élaboration des types de QR Codes (LIEN TYPES QR) et à celle du Modèle (LIEN MODÈLE), afin d'assurer la cohérence de l'ensemble.

Notre choix s'est porté sur une structure XML¹. Cette structure a montré ses faiblesses par la suite (voir REF CHOIX XML), mais avait déjà créé trop de dépendances pour être modifiée.

Afin de minimiser la quantité de données stockées dans le QR Code, seules les données indispensables à l'interprétation du QR Code par l'application mobile y sont stockées. Les données annexes nécessaires au chargement d'un QR Code par l'application de bureau sont stockées dans les métadonnées de l'image (REF CONTROLLEUR/METADONNÉES). On y trouve les données relatives au nom et aux couleurs du QR Code, ou encore au nom des fichiers contenus.

La structure fait apparaître clairement la dichotomie entre les données stockées dans le QR Code (noeud `<donneesUtilisateur>`) et les données stockées dans les métadonnées de l'image (noeud `<metadonnees>`).

Le type de QR Code (atomique ou ensemble) est indiqué par un attribut dans le noeud `<<donneesUtilisateur>>`. Ce noeud contient un noeud `<contenu>` contenant lui-même un ensemble de textes (`<texte>`) et de fichiers (`<fichier>`).

L'appartenance à une famille (REF TYPES QR) de QR Codes est indiquée par la présence d'un noeud `<famille>` ayant pour attributs le nom de la famille et la place du QR Code.

Un exemple complet de représentation XML d'un QR Code est visible dans la figure ci-dessous.

3.3 Chargement et modification de QR Codes

1. Hypertext Markup Language

```

<qrcode>
  <donneesutilisateur type="atomique">
    <contenu>
      <texte>champ1</texte>
      <fichier url="URLFICHIER"></fichier>
      <texte>champ2</texte>
    </contenu>
    <famille nom="famille" ordre="4"></famille>
  </donneesutilisateur>
  <metadonnees>
    <fichiers>
      <fichier url="URLFICHIER" nom="fichier1"></fichier>
    </fichiers>
    <colorqrcode color="#085a0c"></colorqrcode>
    <textebraille texte="QR"></textebraille>
    <colorbraille color="#f11313"></colorbraille>
  </metadonnees>
</qrcode>

```

FIGURE 3.1 – Représentation d'un QR Code

Chapitre 4

Réalisation

4.1 Application de bureau

4.1.1 Architecture

4.1.2 Modèle

4.1.3 Vue

4.1.4 Contrôleur

4.2 Application mobile

4.2.1 Application existante

4.2.2 Adaptation aux nouveaux QR Codes

4.2.3 Téléchargement Drive

4.3 Contraintes et choix

- Stockage des sons en distant dans le drive
- Utilisation structure XML
- Compression XML
- ...

Chapitre 5

Conclusion