LagtProjet : Générateur de Codes QR

Cahier de conception

*Solution partielle*

Une image contenant texte, Téléphone mobile, capture d’écran, gadget

Description générée automatiquement

Raphaël Tremblay-Lessard

# Notes d’édition

Première édition @2024

Création de l’œuvre : Raphaël Tremblay-Lessard

Document modèle original : Patrick Simard

Collaboration(s) : Aucun

Actualisation(s) : Aucun

## Conditions d’utilisation

L’auteur autorise les enseignants du département d’informatique du Cégep de Chicoutimi à utiliser cette œuvre exclusivement dans le cadre de leur cours de la formation régulière en présentielle au même collège.

En cas d’appropriation d’une portion mineure afin de contribuer à la création d’une nouvelle œuvre ou en ajout à une œuvre existante, l’auteur demande que sa participation soit reconnue et qu’il soit mentionné dans la nouvelle œuvre à titre de collaborateur. La consultation de l’auteur pour cet acte est fortement suggérée.

L’appropriation d’une portion majeure afin de contribuer à la création d’une nouvelle œuvre ou en ajout à une œuvre existante n’est pas autorisée par l’auteur. L’auteur demande à être consulté au préalable afin d’obtenir son autorisation.

En cas de modification mineure (ajout mineur, retrait ou modification d’élément), l’auteur invite l’enseignant concerné à le mentionner à la section *Actualisation(s)* en ajoutant son nom et la ou les année(s) où cette action a été posée.

## Reconnaissance et protection du droit d’auteur

Selon l’article 8-1.03 de la convention collective des enseignants, les cahiers de notes ou notes de cours ne peuvent être utilisés sans le consentement explicite de l’auteur. De plus, selon l’annexe V-4 la reproduction sous une forme quelconque, la présentation en public et la publication d’un œuvre est un droit exclusif appartenant à l’auteur.

[Source de l'image de la page de garde : www.louisianafcu.org](https://www.louisianafcu.org/articles/the-polite-persons-guide-to-splitting-the-bill)

Table des matières

Notes d’édition ii

Conditions d’utilisation ii

Reconnaissance et protection du droit d’auteur ii

Rappel du projet 1

Rappel 1

Solution retenue 1

Modélisation de la solution 2

Diagramme de cas d’utilisation 2

Scénarios 3

Déterminer la chaîne de caractères en binaire 3

Description 3

Séquencement 3

Séquence alternative ou Exceptions 3

Coder la correction des erreurs 4

Description 4

Séquencement 4

Séquence alternative ou Exceptions 4

Structurer le message final 5

Description 5

Séquencement 5

Séquence alternative ou Exceptions 5

Placement du module dans la matrice 6

Description 6

Séquencement 6

Séquence alternative ou Exceptions 6

Masquage des données 7

Description 7

Séquencement 7

Séquence alternative ou Exceptions 7

Créer le format et la version chaînes, puis placez-les aux bons endroits dans le code QR 8

Description 8

Séquencement 8

Séquence alternative ou Exceptions 8

Diagramme de classes 9

Maquettes de la solution 11

Personnalisation d’un code QR 11

Génération d’un code QR 11

Plan de tests 12

Tests unitaires (boîte blanche) 12

Tests de fonctionnalités (boîte noire) 12

# Rappel du projet

## Rappel

Un groupe d’amis cherche un moyen simple et efficace pour partager les dépenses liées à leurs activités communes. Pour effectuer ces partages, ils ont besoin d’une application permettant la gestion de leurs activités et des dépenses liées.

## Solution retenue

Afin de répondre à leur besoin et tel que présenté dans le cahier des charges, il a été convenu de développer une application native sans base de données. Cette approche sera plus économique en termes de développement et d’entretien puisqu’il n’y aura pas de base de données ni de système de gestion associé à mettre en place. L’installation et la portabilité de l’application seront très facile aussi puisqu’il n’y aura pratiquement qu’à copier les fichiers essentiels pour faire fonctionner l’application.

# Modélisation de la solution

## Diagramme de cas d’utilisation

Le diagramme à la figure 1 représente toutes les actions que pourra effectuer l’utilisateur lors de son utilisation de l’application. Le code de couleurs permet de distinguer les principaux états de l’application (démarrage/arrêt, gestions des participants, gestion des activités, gestion des dépenses d’une activité).

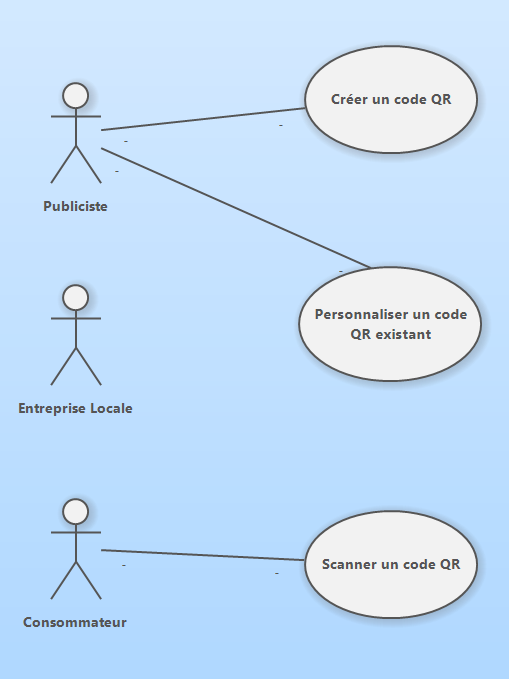


Figure 1 Diagramme de cas d'utilisation

## Scénarios

Normalement, les scénarios décrivent globalement les actions entreprises par les utilisateurs et le système pour l’accomplissement d’une interaction ciblée dans l’application. Les spécificités du système de code QR étant assez pointues, on va plutôt **découper chaque étape de la génération** et écrire des scénarios pour bien les détailler.

Pour notre programme nous allons commencer avec le mot : **HELLO WORLD**

La première étape est : déterminer le mode de code QR, soit alphanumérique dans notre cas. Nous allons suivre le tutoriel proposé et aller ensuite avec un niveau de correction d’erreur Q. Toujours en suivant le tuto, la version la plus petite pour les données **HELLO WORLD** est version 1 (21 pixels par 21 pixels).

### Déterminer le mode d’encodage

|  |
| --- |
| Description Nom du cas : Déterminer le mode d’encodage  But : sélectionner le mode le plus efficace pour le code QR  Acteur principal : Système  Acteur(s) secondaire(s) : Aucun |
| Séquencement Précondition(s) :   * Aucune   Séquence nominale   1. Examinez les caractères de la chaîne d'entrée. 2. Choisir le mode d’encodage qui répond au mieux aux critères prédéfinis   Postcondition(s)   * Le mode d’encodage le plus optimal a été choisi et c’est lui qui sera utilisé dans toutes les étapes suivantes de création du code QR |
| Séquence alternative ou Exceptions Aucune |

### Déterminer la version

|  |
| --- |
| Description Nom du cas : Déterminer la version du code QR  But : Sélectionner la plus petite version pour le code QR  Acteur principal : Système  Acteur(s) secondaire(s) : Aucun |
| Séquencement Précondition(s) :   * Le mode d’encodage, le niveau de correction sont choisis.   Séquence nominale  1. Compter le nombre de caractères de la chaine d’entrée.  2. Selon le mode et le niveau de correction d’erreur, choisir la plus petite version pouvant contenir ce nombre de caractères.  Postcondition(s)   * La plus petite version a été choisie et c’est elle qui sera utilisée dans toutes les étapes suivantes de création du code QR |
| Séquence alternative ou Exceptions Aucune |

### Générer les mots de code de correction d'erreur

|  |
| --- |
| Description Nom du cas : Coder la correction des erreurs  But : Créer les mots de code de correction d’erreur qui vont permettre aux lecteurs de code QR de détecter et corriger les erreurs.  Acteur principal : Système  Acteur(s) secondaire(s) : Aucun |
| Séquencement Précondition(s)   * Toute la chaine en binaire a été formée.   Séquence nominale   1. Splitter la chaine en octets qui seront des mots de codes de données. 2. Repartir les mots de code de données en deux groupes de différents blocs. 3. Pour chaque Bloc utiliser la bibliothèque de Reed Solomon pour générer les mots de code de correction d’erreur correspondants.   Postcondition(s)   * Les mots de code et les mots de code de correction d’erreurs à utiliser sont générés. |
| Séquence alternative ou Exceptions Aucune |

### Structurer le message final

|  |
| --- |
| Description Nom du cas : Structurer le message final  But : Structurer le message final qui sera utilisé par la matrice.  Acteur principal : Système  Acteur(s) secondaire(s) : Aucun |
| Séquencement Précondition(s)   * Les mots de code de correction d’erreur ont été crées pour chaque bloc.   Séquence nominale   1. Entrelacer les mots de code de données. 2. Entrelacer les mots de code de correction d’erreur. 3. Placer les mots de code de correction d'erreur entrelacés après les mots de code de données entrelacées. 4. Convertir en binaire.   Postcondition(s)   * La chaîne binaire finale est formée. |
| Séquence alternative ou Exceptions Séquence alternative #1 –   1. Ajouter des bits restant si nécessaire |

### Masquage des données

|  |
| --- |
| Description Nom du cas : Masquage des données  But : Déterminer et appliquer le meilleur motif de masque pour faciliter sont scannage par les lecteurs.  Acteur principal : Système  Acteur(s) secondaire(s) : Aucun |
| Séquencement Précondition(s)   * Placement des modules (modèles de fonction et message final) dans la matrice   Séquence nominale   1. Appliquer tous les huit modèles de masque à la matrice. 2. Pour chaque modèle calculer le score de pénalité. 3. Choisir le modèle de masque ayant le score de pénalité le plus bas. 4. Appliquer effectivement ce modèle de masque selon sa formule.   Postcondition(s)   * Le meilleur motif de masque a été appliqué à la matrice. |
| Séquence alternative ou Exceptions Aucune |

### Générer l’image finale de code QR

|  |
| --- |
| Description Nom du cas : Générer l’image finale de code QR.  But : Générer un format d’image sur lequel sera scanner le code QR.  Acteur principal : Système  Acteur(s) secondaire(s) : Aucun |
| Séquencement Précondition(s)   * Le meilleur motif de masque a été appliqué, et les zones réservées ont été remplies.   Séquence nominale   1. Générer une image avec la couleur blanche pour 0 et la couleur noire pour 1.   Postcondition(s)   * Un code QR fonctionnel |
| Séquence alternative ou Exceptions Séquence alternative #1 – |

## Diagramme de classes

Diagramme de classes de l’application console

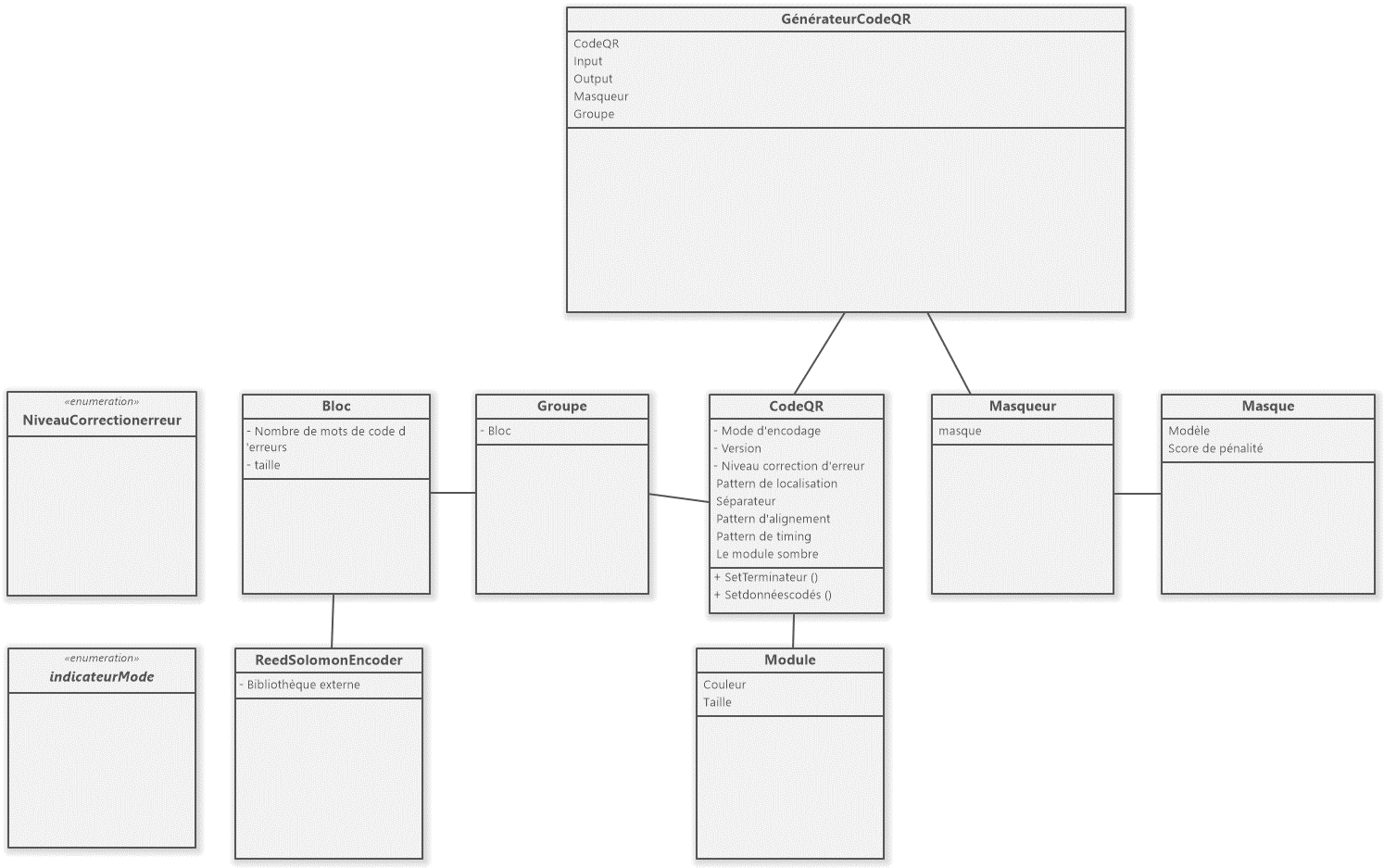
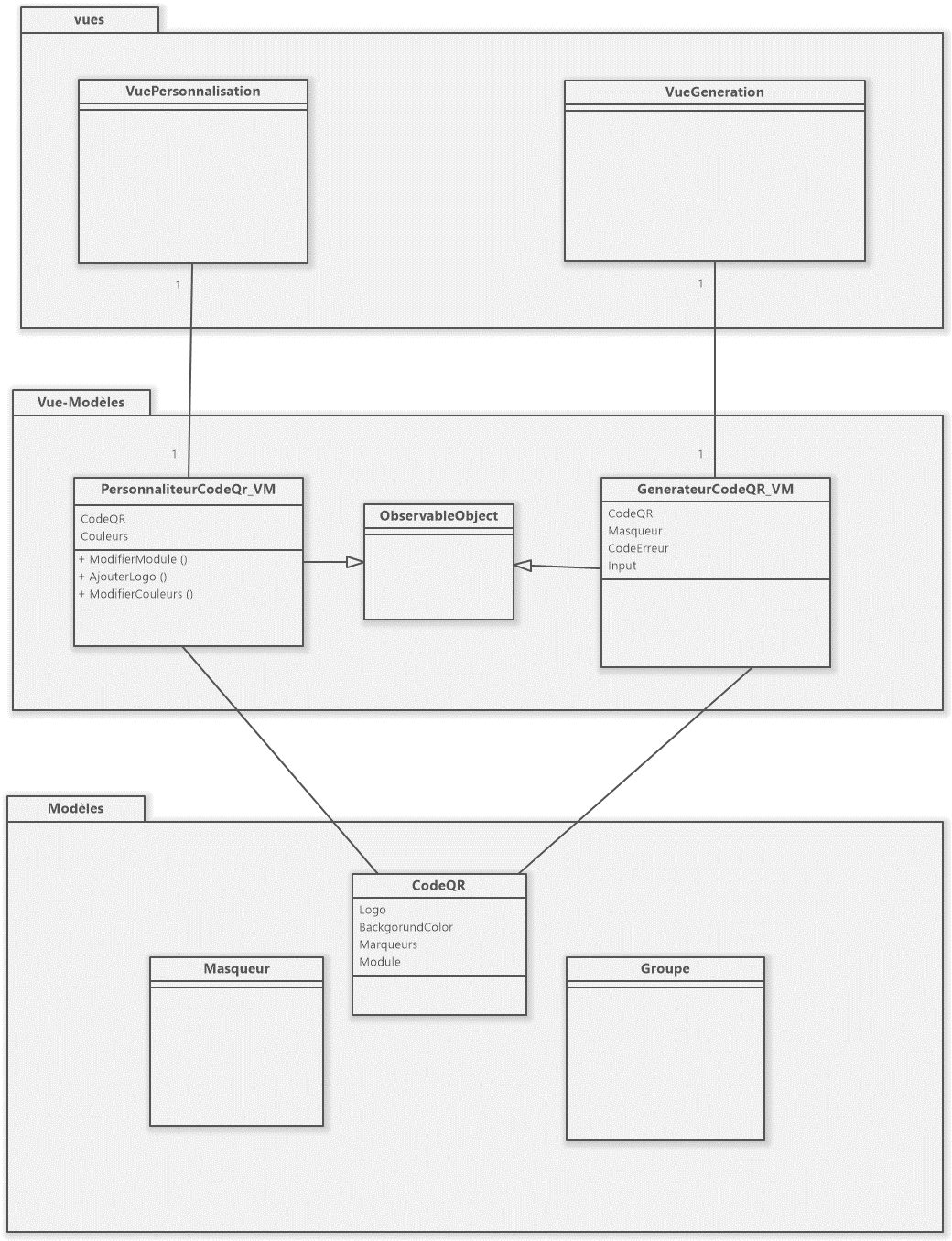


Diagramme de classes de l’application visuelle 

# Maquettes de la solution

## Personnalisation d’un code QR

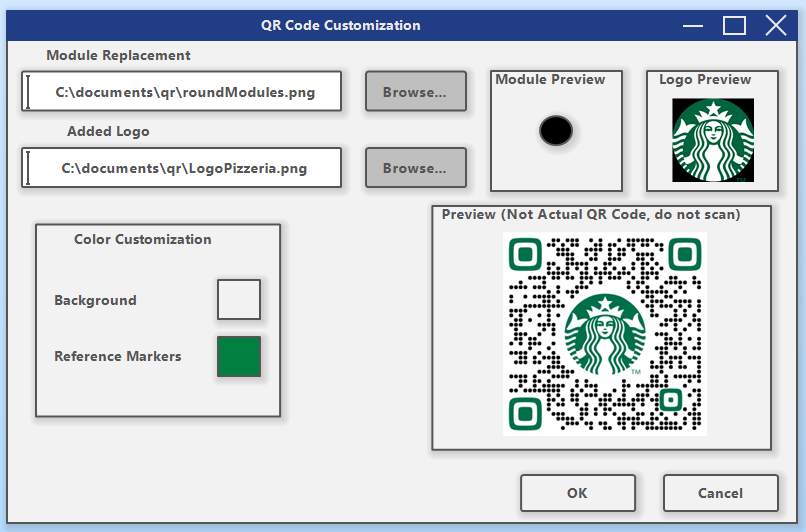


Figure 4 Vue de la personnalisation d’un code QR

## Génération d’un code QR

**Vue en console**

**Exemple de génération :**

**(Input reçu à l’externe)**

**\*Affichage imagePNG\***

Figure 5 Vue de la génération d’un code QR

# Plan de tests

## Tests unitaires (boîte blanche)

En parallèle au développement de l’application, des tests unitaires seront mis en place afin de vérifier le bon fonctionnement de chacune des classes tout au long du développement. Ces tests unitaires seront exécutés minimalement à la fin de chaque *sprint* et avant toute livraison au client, que ce soit pour une démonstration, un livrable d’étape (prototype) ou pour la livraison finale. Un rapport sur les résultats des tests unitaires sera fourni lors de la livraison du produit final.

## Tests de fonctionnalités (boîte noire)

Les tests de fonctionnalités suivants seront réalisés lors de la période de *tests et débogage* ainsi qu’avant la livraison du produit final. Un rapport sur les résultats des tests de fonctionnalités sera fourni lors de la livraison du produit final