

PROJET : Digital Business Card

Ce projet comprend la gestion de cartes de visite professionnelles avec trois fonctionnalités :

- L'affichage et la modification de la carte de visite pro
- L'import et l'envoi de carte de visite à partir des contacts enregistrés sur l'appareil
- L'envoi et la réception des cartes de visite depuis et vers un appareil Android (QRCode + NFC)

I – Diagramme de cas d'utilisation de l'application

Diagramme de cas d'utilisation pour l'authentification /inscription :

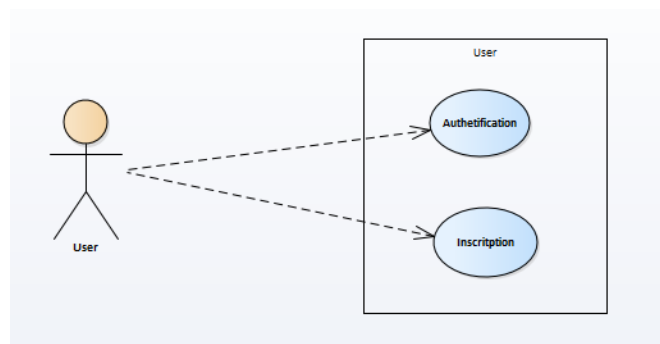


Figure 1: Authenfication/inscription d'un utilisateur

Diagramme de cas d'utilisation pour partage de la carte :

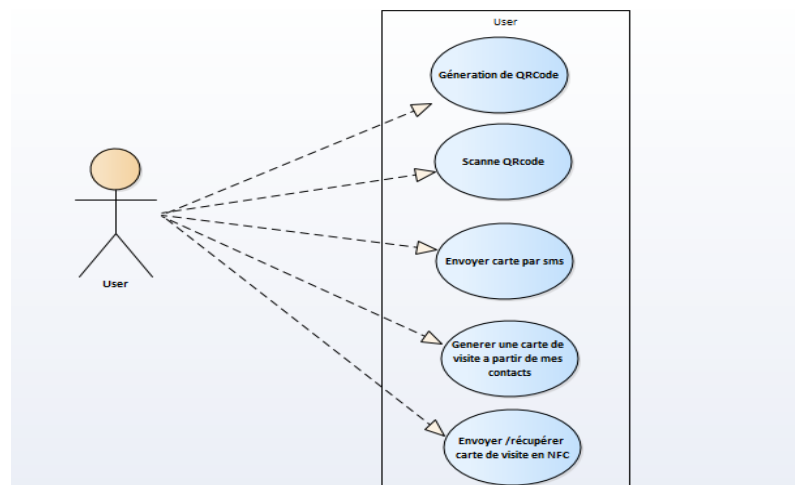
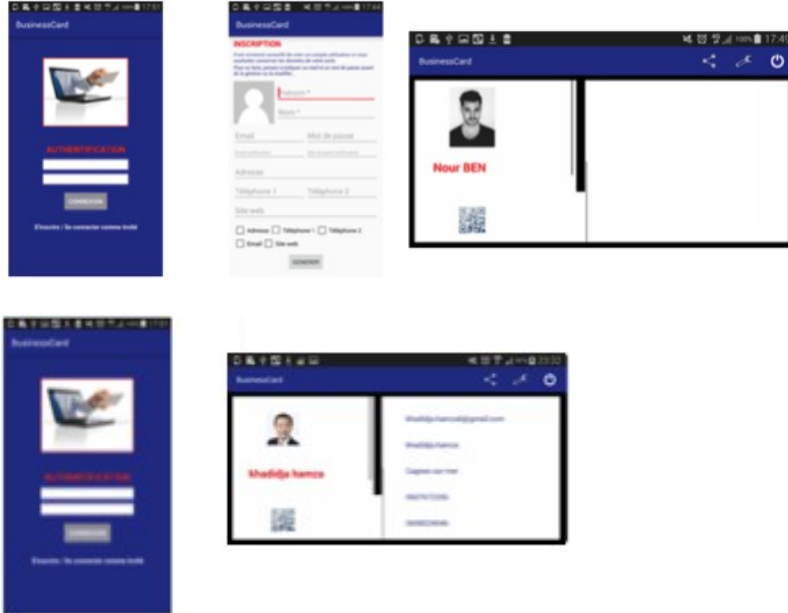


Figure 2: Différents types de partage de la carte

II - Scénarios réalisés

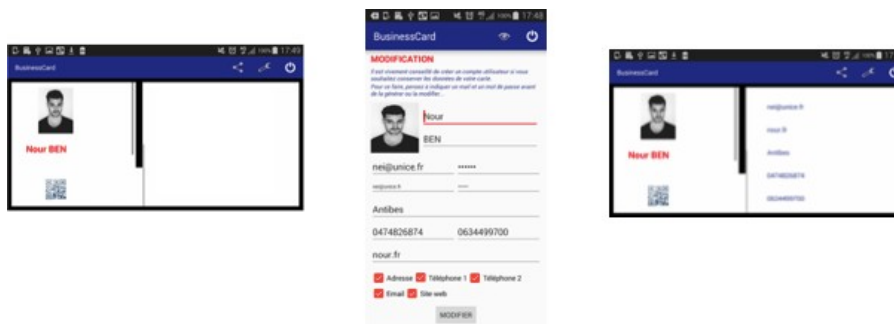
1) Stocker mes données et y accéder à tout moment



Au lancement de l'application, deux possibilités s'offrent à l'utilisateur :

- S'authentifier (appel de la fonction login()) pour vérifier si l'utilisateur existe bien)
- S'inscrire en indiquant mail et mot de passe pour une récupération de l'ensemble de ses données ultérieurement ou se connecter en tant qu'invité sans aucune volonté de stocker ses données.

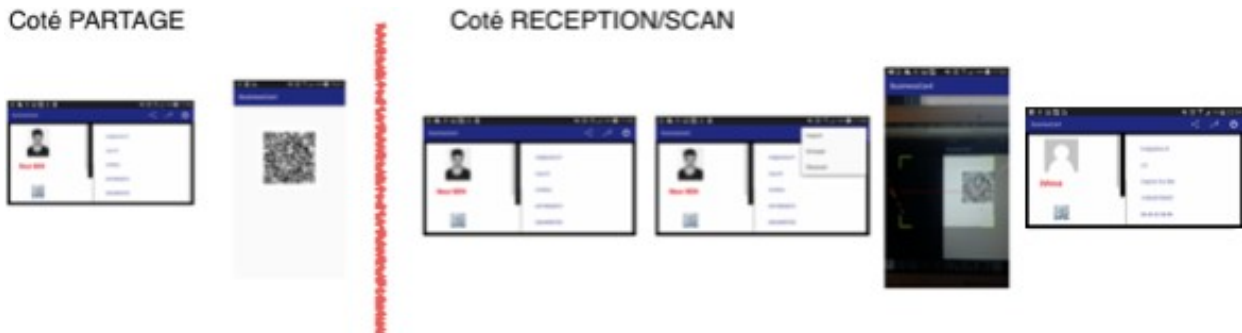
2) Choisir les éléments à afficher sur ma carte de visite



Après clic sur l'icône « Paramètres », il suffit de cocher les données que l'utilisateur souhaite afficher. Pour ce faire, nous avons utilisé les SharedPreferences.

Une fois avoir cliqué sur le bouton « MODIFIER », la carte de visite est mise à jour avec affichage des éléments sélectionnés.

3) Partager sa carte de visite avec une personne ayant la même application



***Partage

Un QRCode est généré et correspond à la carte de visite actuelle juste en cliquant sur l'image représentant un QRCode. Une activité se lance prenant en compte que nous sommes dans le cas où l'utilisateur souhaite partager sa carte de visite.

***Scan

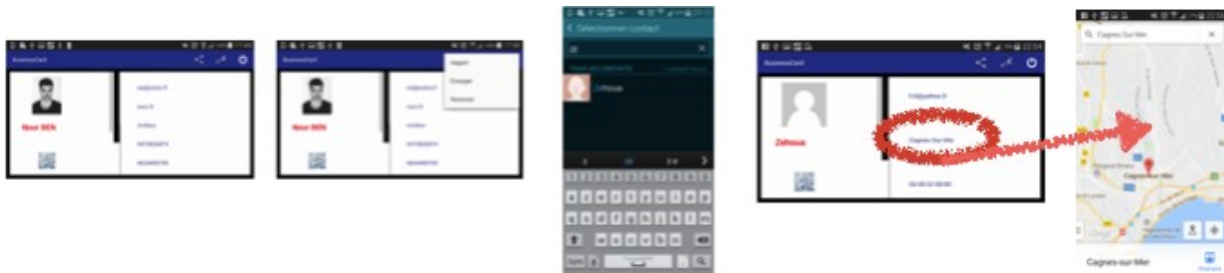
Pour recevoir une carte de visite d'un autre utilisateur, il suffit de cliquer sur l'icône « Paramètres » puis « Scan ». La caméra se lance et cherche à détecter un QRCode. Une fois détecté, une nouvelle carte de visite avec les données de l'autre utilisateur est générée à partir de l'objet *Card*. Seconde approche, lecture via tag NFC.

4) Envoyer ma carte de visite par SMS



Pour envoyer sa carte de visite, il suffit de cliquer sur l'icône « Paramètres » puis « Envoyer ». Un intent se lance sur la liste de contact. Il ne reste plus qu'à le sélectionner et le mécanisme de SMS s'exécute avec le body et le numéro de téléphone préremplis.

5) Générer une carte de visite à partir de l'un de mes contacts du téléphone



Il est possible de générer une carte de visite à partir des contacts du téléphone.

Après avoir cliquer sur l'icône « Paramètres » puis « Import ». Un intent se lance sur la liste de contact. Une fois le contact sélectionné, une carte de visite est à nouveau générée et affichée avec les données de ce contact.

A savoir : un accès à la géolocalisation (intent implicite) de l'adresse affichée sur la carte de visite est toujours disponible.

III - Description des fonctionnalités réalisées

1) Mise en place de la base de données

Nous avons défini également un modèle de base de données pour le stockage des fichiers qui est le suivant :

CARD
<u>CARD_id</u> : String CARD_mail : String CARD_password : String CARD_name : String CARD_surname : String CARD_address : String CARD_tel1 : String CARD_tel2 : String CARD_website : String CARD_photo : String

Dictionnaire de données

Champs	Type	Description
CARD_id	Integer	identifiant unique de la carte de visite
CARD_mail	String	mail de l'utilisateur/contact unique dans la base
CARD_password	String	mot de passe de l'utilisateur
CARD_name	String	prénom de l'utilisateur
CARD_surname	String	nom de l'utilisateur
CARD_address	String	adresse de l'utilisateur
CARD_tel1	String	téléphone de l'utilisateur
CARD_tel2	String	téléphone de l'utilisateur
CARD_website	String	URL du site de l'utilisateur
CARD_photo	String	chemin de la photo de l'utilisateur visible sur la carte

Cette unique table nous permet de gérer l'ensemble des scénarios demandés et est maintenable. On peut facilement ajouter d'autres éléments et les gérer dans le modèle par exemple.

A savoir :

- L'identifiant est autoincrémenté.
- Le mail doit être unique dans la base, elle permet de s'identifier à l'application.
- De manière arbitraire, le nom et le prénom du contact doit systématiquement être renseignée pour pouvoir accéder à l'application mais aussi pour importer/recevoir une carte. Il est pour nous évident qu'il n'est pas possible de partager une carte professionnelle sans au moins ces deux éléments.
- L'image représentant la photo n'est pas l'image en elle-même mais seulement le chemin nous permettant d'y accéder sur le téléphone.
- Les champs non précisés dans cette partie sont gérés de manière standard.

2) Mécanisme inter appareil : SMS

Nous avons utilisé pour l'envoi des SMS, l'application existante en envoyant la carte sérialisée (la carte est mise en type primitif).

Pour la réception d'un message contenant les éléments d'une carte de visite, nous a vous procédé de la même manière en faisant pour ce cas l'inverse, c'est-à-dire désérialiser le message reçu pour pouvoir l'afficher sur la carte de visite via un objet Card.

A savoir : *tout comme pour l'extraction des contacts, le code est inspiré de celui vu en cours.*

3) Mécanisme NFC : Read/Write mode

Nous avons utilisé le protocole NFC pour les cas d'usage : l'envoi et la réception d'une carte de visite depuis et vers un autre appareil Android par le biais d'un tag NFC.

Le protocole en mode Read/Write est implémenter de façon à lire ce tag.

Pour ce faire, nous avons donné les autorisations et le filtre de l'activité nécessaires sur le manifeste Android.

La classe NFCAdapter est la classe utilisée pour définir l'ensemble des méthodes

4) Séparation du code Android avec le code Java pure

Notre application considère 4 activités différentes pour la gestion des différents scénarios exposés précédemment. Toutes la partie utilitaire, c'est-à-dire la validation des champs, des messages d'erreur, de la partie SMS ou encore de localisation a été séparé du code propre aux activités afin de les alléger.

Nos activités appellent différentes classes ayant du Java pure comme les classes gérant notre base de données d'une part avec l'ensemble du mécanisme nous permettant de stocker/extraire les éléments d'une carte de visite.

D'autre part, avec notre classe Card, nous pouvons entre autres de sérialiser l'ensemble des cartes de visite pour chaque scénario.

IV - Répartition du travail effectué

La répartition du travail n'a débuté qu'après l'élaboration des scénarios à réaliser.

A partir de cela, nous avons établi les layouts et menus. Nous avons également enrichi le fichier *strings.xml* pour avoir une application potentiellement « internationalisable ».

1) Nour

Nour a mis en place et réalisé :

- la gestion de la base de données SQLite
- la mise en place du modèle pour la lecture de carte par QRCode et NFC
- la classe User pour rendre possible la sérialisation
- le système de création de compte et de validation des éléments de la carte

2) Khadidja

Khadidja a mis en place et réalisé :

- le mécanisme d'extraction de contact complètement
- la gestion d'envoi de SMS
- l'ensemble du mécanisme lié au QRCode et NFC (import/partage)