

18/12/2018



Fabian Descampe  
Robin Gielen  
Sébastien Raemdonck  
Tanguy Alexandre  
Maximilien Lambert  
Arno Godart  
Adelin Gauthier

# 1.0 Fonctionnalités, intérêts

## 1.1 Présentation

SmartScan est tout d'abord une application qui va vous permettre de vous débarrasser de vos cartes de visite en version papier. Comment ? Tout simplement en les scannant et en les stockant sur votre smartphone.

Vous pouvez introduire toutes vos cartes de visites, même celles qui vous paraissent difficile à lire. En plus de cela, vous avez le choix entre les classer de façon alphabétique ou chronologique.

## 1.2 Application

Le but premier de l'application est de permettre à l'utilisateur de scanner ses cartes de visites afin de les ajouter à notre compte sous la forme de contacts.

Pour ce faire, l'utilisateur a la possibilité de se connecter à son compte (s'il en possède déjà un) ou bien de créer un compte. Cela nécessite une adresse mail non encore liée à un compte smartscan, que ce soit sur le site web ou l'application.

Lorsque l'utilisateur est connecté à l'application, il aura la possibilité d'afficher ses informations, afficher les différents contacts qu'il possède ou bien ajouter un nouveau contact.

- Si l'utilisateur affiche ses informations, il pourra les voir ainsi que les modifier.
- Si l'utilisateur affiche ses contacts, il verra tous les contacts qui auront été ajoutés via l'application ou le site web.
- Si l'utilisateur souhaite ajouter un nouveau contact, il aura l'occasion de le faire via une entrée manuelle ou bien via le scan d'une carte de visite.

L'application nécessite pour le moment une connexion à internet pour fonctionner.

## 1.3 Site Web

En plus d'être une application, SmartScan est également un site internet qui fonctionne en parallèle avec l'application. C'est-à-dire que toutes les cartes de visite que vous mettez sur l'application se retrouveront sur le site internet et vice-versa. Il suffira juste de vous connecter avec votre compte que vous aurez au préalable créé. Vous pouvez évidemment créer un compte sur le site ou l'application, comme bon vous semble.

Nous avons mis en place, pour les professionnels et les PME, un système de mise en avant de cartes de visite. C'est-à-dire que n'importe quelle personne qui voudraient mettre en avant sa carte de visite pourrait le faire. Cette exposition se situe sur la page vitrine du site Internet.

De plus, l'entreprise qui aura payé pour mettre sa carte en avant, aura la possibilité, moyennant un abonnement pro, d'avoir son propre site vitrine personnalisé réalisé par notre équipe.

L'utilisateur a la possibilité de supprimer son compte dans le cas où il ne souhaiterait plus se retrouver dans notre base de données. Et s'il le souhaite, il peut également modifier ses infos personnelles en cas de changement de mot de passe, de nom d'utilisateur ou encore d'adresse mail.

L'application n'étant pas (encore) sur iOS (le système d'exploitation mobile d'Apple), nous avons rendu le site web accessible sur n'importe quel smartphone, il est responsive design. De plus, nous avons pensé à ces utilisateurs en ajoutant la fonctionnalité d'ajouter manuellement une carte de visite. Ça signifie que pour l'instant, même s'il n'y pas encore d'application sur iOS, ces utilisateurs peuvent tout de même utiliser le site internet.

## **1.4 Intérêt**

L'intérêt de ce projet est simple, c'est de débarrasser les entrepreneurs qui se retrouvent avec une multitude de cartes de visite dans leur portefeuille après une conférence ou même plus simplement après être allé à la boulangerie du coin. La probabilité de perdre ces cartes est grande. C'est là que notre application entre en jeu pour recenser toutes ces cartes et ne plus devoir remplir son portefeuille.

## **2.0 Qualité de réalisation**

### **2.1 Objectif**

Notre objectif concernant le site était de faire quelque chose destiné aux professionnels, donc nous avons fait un site épuré et efficace.

### **2.2 Améliorations possibles**

Il y a cependant des pistes d'améliorations qui sont envisagées. La plus importante est la possibilité de créer des groupes de contacts afin de faire un tri, de pouvoir les regrouper par métier, par conférence etc. Il y a déjà une alternative disponible sur le site, lorsqu'on ajoute une carte manuellement, on doit rajouter une catégorie et celle-ci s'ajoute automatiquement dans la liste des catégories.

Une autre également qui nous semble utile, c'est le fait de pouvoir modifier les données d'une carte de visite. Il arrive souvent que les données sur la carte ne soit pas mise à jour, qu'un numéro de téléphone ai été modifié.

Au niveau de l'application, les deux aspects les plus importants à améliorer sont l'accessibilité hors ligne de l'application et le scan des cartes de visites, qui pourrait être automatisé plus qu'il ne l'est actuellement.

Enfin, d'autres améliorations qui sont moins essentielles mais toujours appréciables comme le partage de cartes de visite, le changement de langue ou encore le transfert de données en fichier .csv

### **2.3 Difficultés rencontrées**

Les principales difficultés rencontrées étaient d'adapter notre site web à tous les types d'écrans. En effet, il fallait rendre notre site responsive, ce qui a demandé beaucoup de temps pour adapter notre code.

De plus, nous voulions utiliser l'OCR de Google Vision mais sans que l'utilisateur doive sans arrêt se connecter à internet. Il nous a fallu donc développer une solution pour palier à ce problème.

En outre, l'aspect entrepreneuriat et marketing nous a également pris beaucoup de temps. En effet, nous avons eu peu de cours à ce sujet et devons développer nombres de concepts marketings assez difficile à comprendre. Sous les conseils de quelques élèves de marketing ainsi que de comptas, nous avons réalisé tant bien que mal une enquête quantitative.

## 3.0 Gestion de projet

### 3.1 Présentation

La réalisation de ce projet a grandement été aidé par le cours *Gestion de projet*.

Il y a vraiment eu une bonne dynamique pour ce projet, tous les membres ont eu un réel intérêt et tout le monde a participé de manière égale.

### 3.2 Communication

Concernant notre communication, nous avons effectué chaque jeudi une rétrospective du sprint en cours ou du sprint précédent/suivant. Nous mettions chaque fois en avant les avantages et désavantages de ce sprint.

En plus de cela, nous avons essayé de se voir au minimum une fois par semaine afin de discuter ensemble, d'aider ceux qui avaient des problèmes.

Enfin, dans le cas où nous n'avions pas la possibilité de se voir et que nous avions des questions/problèmes, nous utilisons *Discord* pour pouvoir se parler vocalement et/ou montrer notre écran. Et s'il n'était pas nécessaire de parler vocalement, nous posons nos questions sur Messenger car nous avons une discussion propre à ce projet.

### 3.3 Méthodologie Scrum

Nous n'avons pas compté les heures de travail de chacun car nous avons peur que ça fasse de la concurrence entre membres du groupe. Nous savons que chacun a travaillé de manière égale, et quand quelqu'un avait davantage travaillé durant un sprint, il travaillait légèrement moins au sprint suivant.

Nous n'avons également pas effectué le planning poker car nous avons déjà défini, lors de la création des User Stories, l'importance de celles-ci.

D'autre part il a été difficile de mettre cette méthodologie dans le cadre d'un projet durant notre cursus scolaire étant donné qu'on a d'autres cours (d'autres projets) et des activités en dehors des cours alors que si celui-ci était dans le cadre d'un projet d'entreprise nous aurions pu prendre plus de temps pour établir des réunions plus régulières et avancer de manière plus dynamique.

### 3.4 Organisation

L'utilisation de l'outil Trello nous a été très bénéfique. Ça nous a permis de présenter différemment les fonctionnalités de l'application/site web, de les classer par ordre d'importance et si elles étaient à faire/ en cours / terminées.

### 3.5 Méthodologie Agile

Nous avons utilisé différents principes d'Agiles :

- Pair Programming : Il est arrivé plusieurs fois que 2 d'entres nous travaillent sur la même machine. Un codant et l'autre observant. Après quelques instants, l'observateur donnait son avis et donnait les imperfections dans le code. Les rôles changeaient régulièrement.
- Nous donnions davantage d'importance au concret qu'à la documentation inutile. Nous savions qu'il fallait faire des rapports, des comptes-rendus mais donnions moins d'intérêt à ces documents.
- Nous nous adaptions au changement. Certes, nous avions un plan, des fonctionnalités à faire. Mais il arrivait qu'on doive changer des choses, qu'on doive supprimer des fonctionnalités.

## 4.0 RSE

### 4.1 Présentation

Nous avons intégré plusieurs fonctionnalités sur notre site afin que les personnes malvoyantes puissent naviguer facilement. De plus, nous avons tout fait afin que notre site soit le plus optimisé, énergétiquement parlant.

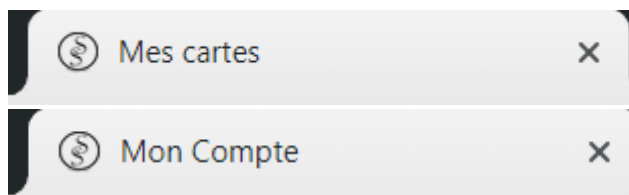
Nous allons partager ce document en deux parties, d'abord la partie faite pour les personnes ayant un handicap et enfin la partie pour l'environnement.

### 4.2 Partie accessibilité

- **Les balises <title> correspondent à leur page**

Nous savons que pour les malvoyants, il n'est pas évident de s'y retrouver sur le web. Pour ce faire, nous avons, pour chaque page de notre site web, associé chaque balise <title> à la page correspondante.

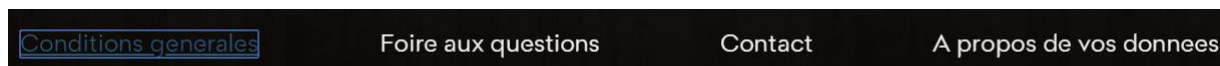
Voici 2 exemples :



- **Le site est accessible via la touche *Tabulation***

Il est très important que le site soit entièrement accessible via la touche Tabulation et que celle-ci soit visible.

Tout lien et fonctionnalité sur notre site est accessible via cette touche et est très visible.



- **Ecriture claire sur fond foncé**

Nous avons pensé aux malvoyants en essayant de faire un site accessible à tout le monde, y compris pour les malvoyants.

Une écriture claire sur un fond noir est beaucoup plus lisible qu'une écriture noire sur un fond clair. Nous avons donc fait notre choix du design en fonction de ça.

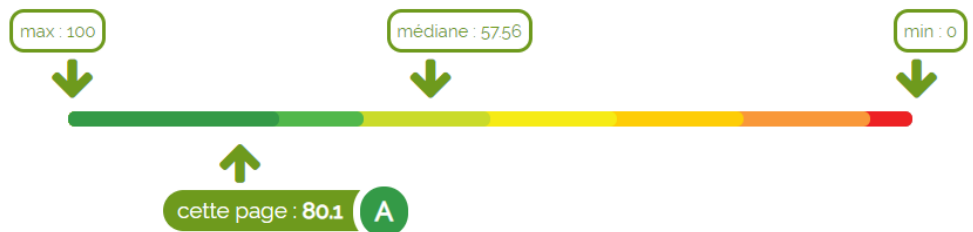
## 4.3 Partie environnement

→ [www.smartscan-bc.ovh/index.php](http://www.smartscan-bc.ovh/index.php)

calculé le 06 / 12 / 2018

Renouveler le test pour  
[www.smartscan-bc.ovh/index.php](http://www.smartscan-bc.ovh/index.php) ...

→ Performance environnementale ⓘ



Nous avons audité notre site avec l'outil ecoIndex, voici notre résultat.

Nous sommes à la 2416<sup>ème</sup> place sur 13419 pages, ce qui est un résultat honorable.

- **Business Network International Alliance**

D'autre part nous comptons mettre en place dans les réunions du style BNI Alliance (Business Network International) : réunion qui permet l'échange de carte de visites entre des entrepreneurs membres de ce groupe, une poubelle spéciale pour les cartes de visites ce qui permettra de diminuer la circulation de carte de visite physique et donc éviter l'abandon de carte de visite et amènera vers la digitalisation de celle-ci grâce à notre application.



- Compression des images

Toutes les images se trouvant sur notre site ont été compressé afin de prendre le moins d'espace possible. Nous avons gagné 70% d'espace grâce à cette compression.

- Message préventif

Sur notre site web, un message préventif est présent. En effet, celui-ci invite à ne pas jeter sa carte sur la voie publique. De ce fait, nous espérons sensibiliser les utilisateurs à la pollution.

- HTML/CSS checker

Nous avons fait validé notre site avec les 2 validateurs connus de CSS et HTML et nous avons aucune erreur.

## Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

**Showing results for <http://www.smartscan-bc.ovh/index.php>**

Checker Input

Show ☐ source ☐ outline ☐ image report

Check by

**Document checking completed. No errors or warnings to show.**

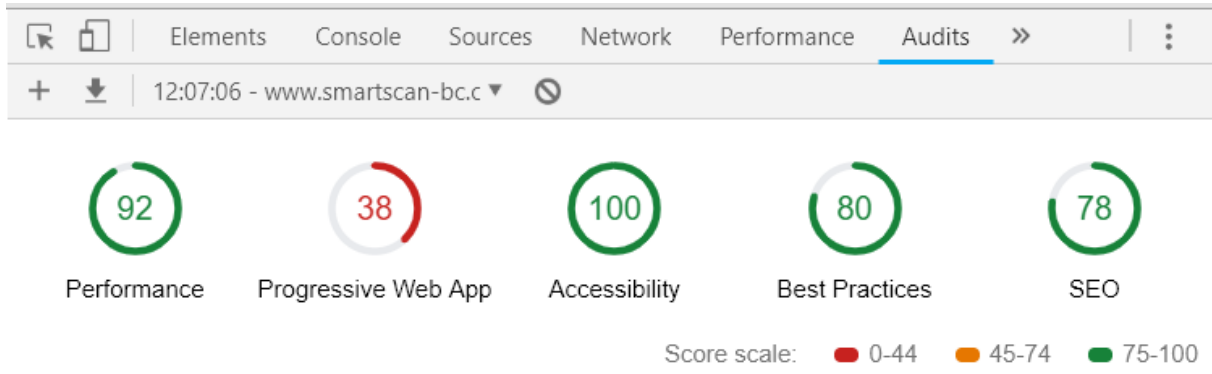
Used the HTML parser. Externally specified character encoding was UTF-8.

Total execution time 202 milliseconds.

- **Audit de Google**

Comme vous pouvez le voir, nous avons 100 % dans l'accessibilité.

En plus de cela, nous avons 92 % dans l'onglet performance. Nous n'arrivons pas à 100% à cause des images qui ne sont pas compressibles à l'infini.

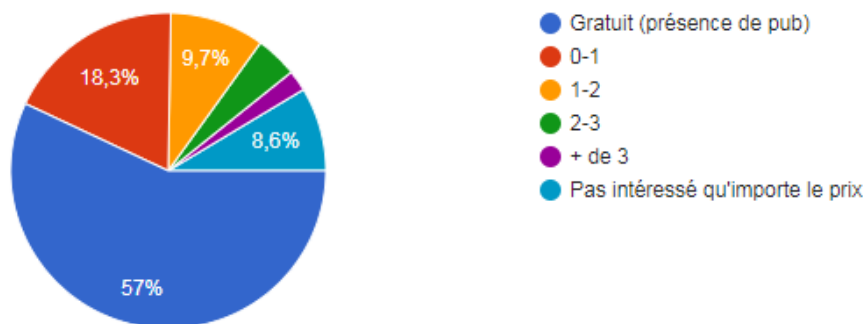


## 5.0 Aspect entrepreneuriale

### 5.1 Etude quantitative

Cette étude quantitative a été réalisée sur un panel de 100 personnes. Pour rappel, notre application vise tous les membres actifs de la société (salariés, indépendants, aidants) soit un total de 1.346.840 personnes. Néanmoins, certains services visent plutôt les indépendants comme la mise en place d'un site vitrine.

Combien seriez vous prêt à payer pour notre application (en euros)



-57% gratuit mais présence de pub : Il faut savoir que les pubs sur application sont beaucoup moins rentables qu'il n'y paraît. Après de nombreuses recherches, il en ressort qu'un clic sur une pub rapporte environ 10 cents. Sachant que les utilisateurs cliquent très peu sur les pubs (2%), il nous faut donc un très grand nombre d'utilisateurs pour considérer ces pubs comme une rentrée d'argent fiable.  $614159 * 2\% * 0.10 = 1228,3$  euros /jour, soit un total de **36849.5 euros par mois**.

-18.3% entre 0 et 1 euro. Soit  $246472 * 0.50 =$  **123 235.8 euros**

-9.7% entre 1 et 2 euros. Soit  $104515 * 1.5 =$  **156 772.2 euros**

-8.6% Pas intéressé qu'importe le prix.

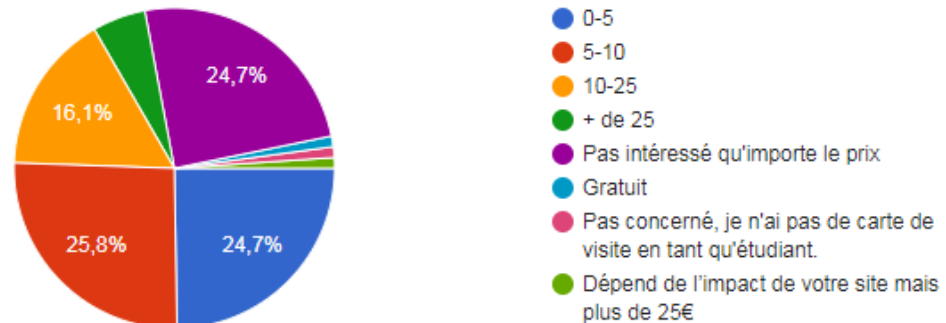
-4.3% entre 2 et 3 euros. Soit  $92667 * 2.5 =$  **231 656.4 euros**

-2.2% plus de 3 euros. Soit  $23704 * 3 =$  **71 112.2 euros**

Note : il faut enlever à ces chiffres, la TVA appliquée en Belgique mais également les frais de Google qui s'élève à 30%. On ne touchera donc qu'environ 58% du prix de vente TTC.

Le prochain graphique nous montre le nombre de personnes qui serait prêt à payer pour mettre en avant la carte de visite sur notre site internet, cette fonctionnalité vise aussi bien les PME qui sont aux nombres de 72.435 en Wallonie que les indépendants qui eux sont 182.745 en Wallonie.

Combien seriez vous prêt à payer pour mettre votre carte de visite en priorité lors des recherches sur notre site Web?



-25,8% entre 5 et 10 euros par mois. Soit  $52669 \times 7.5 = \mathbf{395\ 018.64\ euros}$

-24.7% entre 0 et 5 euros par mois. Soit  $50423 \times 2.5 = \mathbf{126\ 058.92\ euros}$

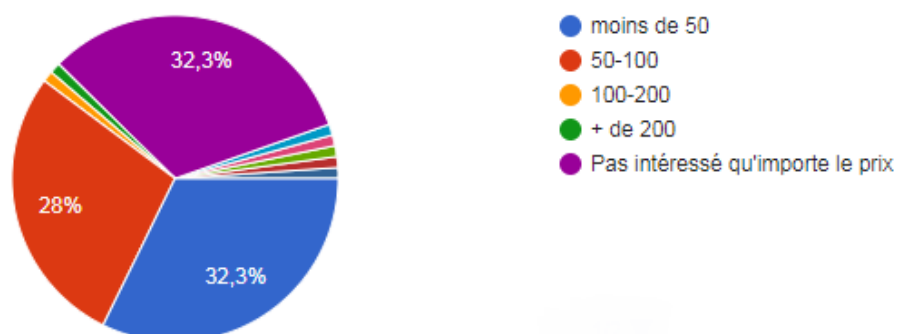
-24.7%+1% pas intéressé qu'importe le prix

-16.1% entre 10 et 25 euros par mois. Soit  $32867 \times 17.5 = \mathbf{575\ 175.72\ euros}$

-5.4%+1% plus de 25 euros par mois. Soit  $13065 \times 25 = \mathbf{326\ 630.4euros}$

Le prochain graphique nous montre le nombre de personnes qui serait prêt à payer pour un « site vitrine » sur notre site web. Cette fonctionnalité vise aussi bien les entreprises que les indépendants. Néanmoins il est clair que peu d'entreprises ont besoin d'un site vitrine car presque toutes les entreprises de nos jours possèdent leur propre site web. Nous avons donc considéré uniquement les indépendants pour utiliser les résultats de ce sondage mais il est clair qu'une entreprise pourrait tout à fait bénéficier de cette fonctionnalité au même titre qu'un indépendant.

Combien seriez vous prêt à payer pour un "site vitrine" sur notre site Web?



-32.3% moins de 50 euros par mois. Soit  $47221 \times 25 = \mathbf{1\ 180\ 532.7\ euros}$

-32.3% pas intéressé qu'importe le prix.

-28% entre 50 et 100 euros par mois. Soit  $40934 \times 75 = \mathbf{3\,070\,116\text{ euros}}$

-1.1% entre 100 et 200 euros par mois. Soit  $1608 \times 150 = \mathbf{241\,200\text{ euros}}$

-1.1% plus de 200 euros par mois. Soit  $1608 \times 200 = \mathbf{321\,600\text{ euros}}$

Si l'on en suit notre étude quantitative, le plus judicieux serait de mettre l'application gratuite au téléchargement, de fixer le prix de la mise en avant des cartes sur notre site entre 5 et 10 euros et finalement la mise en place du site vitrine à moins de 50 euros. Ce qui nous ferait un total de 1 612 399 euro par mois. Toutefois, nous sommes bien conscients que ce chiffre est loin de refléter la réalité et ce pour plusieurs raisons. La première étant l'échantillonnage qui n'a été fait que sur 100 personnes et trop peu d'entre eux étaient dans le monde du travail. La deuxième est le nombre de personnes effectivement prêtent à payer, même si nous avons retiré 20% lors de nos calculs nous pensons qu'il serait plus judicieux de retirer 40%. Enfin, notre manque de connaissance dans ce domaine, en effet c'était la première étude quantitative que nous avons réalisée et nous avons commis quelques imprécisions dans les questions, ce qui a pu également fausser les résultats du sondage. Pour conclure, ces chiffres sont de très bons augures pour notre projet mais nous sommes bien conscients des nombreux facteurs d'erreurs qui ont influencés nos résultats.

## **Rentabilité**

Nos dépenses s'élèvent à 16415 euros par mois si l'on en suit le tableau ci-dessous :

	Dépenses par mois (euros)
VPS	<b>15</b>
Salaires	$7 \times 2000 = \mathbf{14\,000}$
Pub	<b>2500</b>
<b>Total</b>	<b>16415</b>

S'il l'on considère les préférences des utilisateurs misent en lumière par notre étude quantitative et que nous appliquons donc un tarif de moins de 50 euros pour le site vitrine, entre 5 et 10 euros pour la mise en avant sur notre site et une application gratuite nous aurions besoin de 100 000 utilisateurs.

Parmi ces 100 000 utilisateurs, 1000 souhaiterons mettre leur carte de visite en avant sur notre site et 60 un site vitrine. Nous pensons que ces chiffres sont largement atteignables et nous pensons donc être rentables sur le long terme. Nous sommes conscients que nos dépenses sont à revoir à la hausse néanmoins nous dispensons de nombreuses solutions pour palier à ces problèmes comme une diminution de la pub, des salaires, modification de notre système de répartition au sein de l'entreprise, ...

## 5.2 Business Model Canvas

Date: 26/10/2018		<b>BMC</b>		Nom de l'entreprise: SMARTSCAN BC	
<b>Partenaires clés</b> Qui sont vos partenaires clés ? Qui sont vos fournisseurs clés ? <ul style="list-style-type: none"> <li>• Entreprise qui regroupe des professionnels (LinkedIn)</li> <li>• OVH pour le VPS et le nom de domaine</li> </ul>		<b>Activités clés</b> Quelles sont les activités clés nécessaires pour produire l'offre ? (logistique, marketing, production...) Sans ces tâches l'activité meurt. <ul style="list-style-type: none"> <li>• Maintenance de la reconnaissance de caractère</li> <li>• Maintenance de l'application</li> <li>• Gestion de la Base de données</li> <li>• Mettre à jour le site WEB</li> <li>• Maintenance du VPS</li> </ul>		<b>Offre (proposition de valeur)</b> Quelles solutions concrètes l'offre apporte-t-elle aux clients ? <ul style="list-style-type: none"> <li>• Regroupement des cartes de visite</li> <li>• Numérisation de carte de visite</li> <li>• Recherche de professionnels</li> <li>• page web personnalisé</li> </ul>	
		<b>Ressources clés</b> Quelles ressources clés la production de l'offre requiert-elle ? <ul style="list-style-type: none"> <li>• VPS</li> <li>• Nom de domaine</li> <li>• MySQL</li> <li>• Magasin d'application</li> <li>• Internet</li> </ul>		<b>Relation client</b> Quel type de relation chaque segment de clientèle attend-il de l'entreprise ? Lister les types de relations client. <ul style="list-style-type: none"> <li>• FAQ</li> <li>• Email</li> </ul>	
				<b>Segments de clientèle</b> Quels sont les segments de clientèle cibles ? Pour qui crée-t-on de la valeur ? <ul style="list-style-type: none"> <li>• Les professionnels</li> <li>• Particuliers qui recherchent des professionnels</li> </ul>	
<b>Structure des coûts</b> Quels sont les coûts (dépenses) les plus importants générés par le modèle économique ? <ul style="list-style-type: none"> <li>• Nom de domaine</li> <li>• VPS</li> <li>• Employés</li> </ul>		<b>Sources de revenus</b> Quelle est la source de revenus de l'entreprise ? <ul style="list-style-type: none"> <li>• Création de page web personnalisé</li> <li>• Mise en avant de carte de visite</li> </ul>			

## 6.0 Aspects techniques et intégration

### 6.1 VPS

#### 6.1.1 Serveur web (nginx)

Nous avons utilisé NGINX parce que NGINX à l'inverse d'APACHE, n'utilise pas le modèle Thread Driven mais le modèle Event Driven.

La différence principale entre ces modèles :

##### Modèle Thread Driven

Un thread est créé dès qu'un client demande une page web. Ce thread aura pour responsabilité le traitement de l'entièreté de la page web. Quand ce thread attend une entrée / sortie, il se bloque et donne son tour au prochain thread. Un très grand nombre de connexions simultanées, engendre un très grand nombre de threads qui se bloque entre eux.

##### Modèle Event Driven

A l'inverse on ne trouve « qu'un seul thread » qui traite toutes les requêtes puis les répartit sur des handlers approprié qui eux même rerépartissent si nécessaire. Dans cette approche, très peu de threads sont créées, ce qui améliore les performances générales.

#### 6.1.2 Base de données MySql

Nous avons utilisé MySql car

- Le serveur MySQL est très rapide (voir test sur le site de mysql)
- MySQL est beaucoup plus simple d'utilisation que la plupart des autres serveurs de bases de données.
- On peut effectuer beaucoup d'opérations sur une base MySQL en utilisant des interfaces développées en C, Perl, C++, Java, Python, PHP.
- MySQL dispose d'un système de sécurité permettant de gérer les personnes et les machines pouvant accéder aux différentes DB.
- MySQL tourne sur la plupart des systèmes tels que Unix, Windows, Linux ou mac OS.

### 6.1.3 DNS (bind)

BIND bénéficie de plusieurs avantages favorables à son utilisation, qui en fait de loin le serveur DNS le plus utilisé sur internet, voici les différentes raisons de l'utilisation de bind.

- BIND est un standard pour le DNS dans les systèmes Linux et est activement pris en charge par une grande communauté open source.
- BIND est utilisé dans des millions de serveurs DNS de production et est connu pour son fonctionnement stable et prévisible.
- BIND supporte Linux, NetBSD, FreeBSD, OpenBSD, macOS et Windows.

BIND est l'un des seuls serveurs DNS à couvrir toutes les fonctionnalités DNS de base.

Pour tous ces services nous avons utilisé docker parce que :

Cette solution est légère car elle ne demande que quelques centaines de Mo de disque et l'empreinte mémoire est réduite étant donné qu'elle utilise que la mémoire utilisée pour l'application (pas de couche os).

Cette solution est rapide et facilite le déploiement des applications car après la récupération de template (conteneur), il suffit d'une commande pour l'exécuter.

## 6.2 Site Web

### 6.2.1 PHP

Tout d'abord, nous avons utilisé le langage de programmation PHP car nous avons déjà eu, pendant un quadrimestre, un cours sur ce langage. Nous avons donc tous au minimum des bases, et ça nous a aidé.

De plus, PHP est un langage très populaire et assez facile à apprendre. De ce fait, internet nous a beaucoup aidé via des tutoriels et/ou des forums où nous pouvions poser nos questions en cas de problèmes. Ce langage dispose d'une énorme communauté d'utilisateurs, qui sont prêts à entraider.

En outre, PHP gère très bien, depuis sa création, les requêtes SQL. Comme nous utilisons justement beaucoup SQL, on peut donc facilement développer notre site afin d'extraire les données SQL.

Enfin, la plateforme Linux est connue pour être d'une très grande stabilité. Le site peut tourner pendant des semaines voir des semaines sans avoir besoin d'être redémarré.



## 6.2.2 JavaScript

Le JavaScript permet la conception d'applications et sites web dynamiques interactifs fonctionnant sur tous les navigateurs web que ce soit Google Chrome, Mozilla Firefox ou encore Safari. De plus, la communauté très active et les nombreux outils et modules disponibles pour utiliser ce langage facilitent son utilisation et lui permet d'innover constamment.

## 6.3 Application

### 6.3.1 Langage de programmation

Le premier choix qui s'est offert à nous lorsque l'on s'est posé la question de la technologie à utiliser pour le développement de l'application a été le langage de programmation. Nous avons le choix entre Java et Kotlin.

Les trois aspects principaux à prendre en compte ont été l'utilisation des deux langages, leur compatibilité avec les différents appareils sur le marché et notre accessibilité à ces technologies.

Pour ce qui est de la compatibilité, nous savions qu'avec Java l'application ne serait développée que pour les appareils sous android, tandis que Kotlin permettrait une éventuelle adaptation sur les appareils sous iOS.

Ensuite, pour ce qui est des langages en eux-mêmes, nous avons déjà travaillé avec Java tandis que nous n'avons jamais utilisé Kotlin. Étant donné que nous avons prévu d'utiliser plusieurs technologies que nous n'avons jamais utilisées pour le projet nous avons trouvé plus judicieux d'utiliser Java pour l'application. De plus, Java nous à sembler être le langage le plus utilisé, ce qui nous permettrait de disposer de plus d'exemple et de correction d'erreur.

Pour finir, si nous avons utilisé Kotlin dans le seul but d'avoir une adaptation possible vers iOS, il nous aurait fallu le matériel compatible. Malheureusement, aucun d'entre nous ne possédions d'appareil Apple, que ce soit un mac ou un iPhone.

Pour conclure, nous avons choisi d'utiliser Java. D'une part car nous n'aurions pas pu mettre en place l'application sous iOS et d'autre part car nous avons préféré le langage le plus répandu pour notre première application.

### 6.3.2 Bibliothèques utilisées

Lors du développement de notre application, nous avons utilisé plusieurs bibliothèques.

La liste complète étant la suivante :

- GSON
- Retrofit
- Android Image Cropper
- Google Vision

#### GSON

GSON est une bibliothèque utilisée pour convertir des objets Java en leur représentation JSON et inversement. Elle nous permet de convertir les objets que nous envoyons à l'API ainsi que ceux que nous recevons.

Cette bibliothèque est utilisée directement par la bibliothèque Retrofit afin de faciliter les échanges entre le client et le serveur en s'assurant que les données rentrent dans les modèles que nous avons fixés.

#### Retrofit

Nous utilisons cette bibliothèque afin de créer et gérer les requêtes que nous envoyons et recevons de notre API.

Nous avons créé les méthodes qui correspondent aux différents Endpoints de notre API ainsi que les modèles qui correspondent aux données de notre base de données. Nous avons ensuite établi la correspondance entre les méthodes et les Endpoints via Retrofit et, à partir de là, nous avons pu utiliser ces méthodes et interagir directement avec notre base de données à partir de notre application.

#### Android Image Cropper

Nous utilisons cette bibliothèque dans notre application afin de pouvoir sélectionner une certaine zone d'une image. Nous effectuons ensuite notre reconnaissance à partir de cette zone.

#### Google Vision

Pour la reconnaissance de texte, nous devons implémenter un OCR (Optical Character Recognition ou Reconnaissance Optique de Caractères). Deux choix s'offraient à nous lorsque nous voulions le mettre en place. En effet, nous voulions créer une application sous Android Studio permettant de scanner en mode « Offline » car cela pouvait être un plus pour notre application par rapport à nos concurrents. Nous nous sommes donc tournés vers Tesseract qui offrait cette possibilité sous les conseils de nos professeurs ainsi que les coachs externes. Parallèlement, nous savions qu'un autre OCR existait : Google Vision. Cet OCR permet d'avoir de bien meilleurs résultats mais devait être utilisé en mode « Online ». Nous avons donc mis en place une solution permettant d'utiliser Google Vision de façon que l'utilisateur n'aie pas tout le temps besoin de se connecter à internet pour pouvoir scanner les cartes de visites. Cette solution fut développée en Java.

### 6.3.3 Design

La partie design de l'application a été en grande partie réalisée grâce à l'outil intégré de Android Studio. Nous avons utilisé cet outil pour définir le design général de l'application, qui serait appliqué à chaque nouvelle page que nous ajouterions. Ensuite, nous avons-nous même ajouté et modifié les différents éléments inhérents à chaque page.

En effet, nous souhaitons un design épuré et assez classique pour cette application.

## 6.4 Connection à la base de données

### 6.4.1 API

Afin que l'utilisateur puisse sauvegarder ses données, les récupérer et les utiliser sur le site web et l'application, nous avons bien sûr mis en place une base de données.

Afin que l'application puisse envoyer à et récupérer les données depuis cette base de données, nous avons mis en place une API.

Dès le début, nous avons choisis de structurer cette API selon le format REST. En effet, nous avons remarqué qu'une API basé sur ce format nous serait bénéfique étant donné que nous développons notre application en Java et qu'il y avait une grande compatibilité entre une application Android et une API REST.

Nous avons ensuite dû choisir quel langage utiliser pour développer cette API. A cette étape du développement, nous avons déjà mis en place notre serveur web. Nous avons alors choisi de développer l'API en PHP afin que celle-ci utilise le langage que nous avons choisi jusque-là, étant donné qu'elle serait mise sur le même serveur web.

Plusieurs autres solutions s'offraient à nous (Python, Ruby, .NET, Java, ...). Les différences majeures entre ces différents langages étant toujours les différentes bibliothèques disponibles dans chaque langage. Mais la faible ampleur de notre API ne justifiait pas l'utilisation d'un langage en particulier pour pouvoir utiliser une bibliothèque en particulier.

## 6.4.2 Endpoints

Une API se présente donc au « client » sous la forme d'une série de points de contact qui peuvent être contactés afin d'envoyer ou de récupérer des données depuis notre base de données.

Chaque point de contact est appelé un endpoint. Nous avons, dans le cadre de notre application, mis en place 5 endpoints :

- CreateUser
- GetUser
- UpdateUser
- CreateContact
- GetContacts

### CreateUser

Cet endpoint est utilisé afin de créer un nouvel utilisateur dans la base de données. Il sera contacté lorsqu'un nouvel utilisateur va s'inscrire sur l'application.

### GetUser

Cet endpoint est utilisé afin de récupérer les données d'un utilisateur. Il sera contacté dans deux situations. Lorsque l'utilisateur voudra se connecter à l'application et lorsque l'utilisateur voudra accéder à ses données.

### UpdateUser

Cet endpoint est utilisé afin de mettre à jour les données d'un utilisateur. Il sera contacté lorsque l'utilisateur ira via l'application, modifier ses données d'utilisateur.

### CreateContact

Cet endpoint est utilisé afin de créer un nouveau contact pour un utilisateur donné. Il sera contacté lorsqu'un utilisateur scannera une carte de visite ou ajoutera manuellement une nouvelle carte.

### GetContacts

Cet endpoint est utilisé afin de récupérer les données des contacts d'un utilisateur. Il sera contacté lorsqu'un utilisateur voudra afficher les différents contacts qu'il a déjà enregistrés sur son compte

## **7.0 Analyse de sécurité**

### **7.1 Fail2ban**

Nous avons mis en place fail2ban. C'est un framework de prévention contre les intrusions, qui bloque les adresses IP appartenant à des hôtes qui tentent de casser la sécurité du système. Ceux-ci sont repérés grâce au fait qu'ils aient obtenus un trop grand nombre d'échecs lors de l'authentification.

### **7.2 Firewall**

D'autre part nous avons mis en place un firewall, le service est firewalld qui permet d'assurer la sécurité des informations du réseau en filtrant les entrées et en contrôlant les sorties d'après les règles que l'on a défini.

### **7.3 Base de données**

#### **7.3.1 Sauvegarde en cas de perte de données**

Du point de vue de la sauvegarde de notre base de données MySQL, nous n'avons pour le moment pas mis de sécurité en place. Nous avons en effet estimé qu'il n'était pas nécessaire de prévoir un deuxième espace (payant) pour empêcher la perte des données avant que l'application ne soit lancée.

Nous avons toutefois cherché des pistes de solutions pour palier à un éventuel problème en cas de crash de notre base de données dans le cas où nous lançons le projet.

Nous avons imaginé un script de sauvegarde automatique dans un répertoire protégé. Le répertoire ne sera pas accessible par les internautes même s'ils connaissent le lien direct. Notre base de données serait sauvegardée automatiquement dans un fichier .zip. Nous pouvons gérer facilement le temps entre deux sauvegardes et également après combien de temps les anciennes sauvegardes seront supprimées.

Une solution alternative serait de ne sauvegarder que les mises à jour de la base de données par rapport à la version que nous avons déjà archivée. De cette façon, la sauvegarde prendrait moins de temps et rendrait la base de données non accessible pendant une durée plus courte.

#### **7.3.2 Accès à la base de données**

Nous n'avons malheureusement pas eu le temps de mettre en place une sécurité pour limiter l'accès depuis l'extérieur à notre base de données. Cela se traduit par un utilisateur externe capable d'accéder aux données de tous les utilisateurs.

Nous avons cherché les différentes méthodes pour résoudre cela et nous avons plusieurs pistes.

La mise en place de tokens d'identification et la restriction d'accès aux endpoints en dehors de l'application, et donc d'une connexion obligatoire, étant la solution la plus adaptée à notre cas.

### **7.3.3 Application**

Lors du développement l'application, nous avons réfléchi afin d'éviter au maximum les problèmes liés aux données stockées localement. Pour cela, nous avons mis en place quelques sécurités afin que l'utilisateur ne puisse pas se trouver sur les pages qui requiert une connexion sans être connecté ou inversement, mais également afin que les données locales soient réinitialisées dans ces situations.

## 8.0 Conclusion Générale

Le but principal de ce projet était de lier, au travers d'un projet concret, un maximum de technologie apprise durant nos études tout en utilisant la méthodologie de gestion de groupe SCRUM.

Nous avons choisi de nous lancer dans la création d'un site web et d'une application avec toutes les implications techniques que cela peut avoir, comme la mise en place des différents serveurs.

Il existe des aspects qui peuvent être améliorés, comme nous l'avons montré durant ce rapport, mais nous avons également réfléchi à ces aspects et chercher des éventuelles solutions pour pallier aux différents problèmes que nous avons trouvés.

Ce projet nous a montré à chacun qu'il y avait de nombreux aspects d'un travail de groupe qu'il est important de prendre en compte et d'apprendre à maîtriser.

Pour finir, nous pensons, au terme de ce projet, avoir une solution fonctionnelle et présentable qui correspond aux attentes que nous avions au lancement du projet.

## 9.0 Annexes

### 9.1 Business plan

#### Notre projet

##### 1. Résumé

Notre idée était de créer une fonctionnalité pour stocker et référencer facilement les cartes de visite. Nous avons développé une application qui permet de scanner des cartes de visite et de toutes les stocker sur un compte privé.

La mise en place d'un site web nous a paru important pour retrouver nos cartes aussi sur un PC. L'implémentation d'une application était aussi un élément important pour pouvoir scanner facilement les cartes sur son smartphone.

Ces cartes pourront être classées comme le client le désire dans des catégories. L'utilisateur pourra effectuer des recherches d'une carte sur base des données de la cartes comme le lieu, le nom de la personne, le métier, ... Toutes les cartes de visite de l'utilisateur et ses répertoires seront synchronisés sur son compte.

Le client pourra mettre sa carte de visite en avant sur le site en payant une première formule. La personne aura aussi possibilité pour payer pour une deuxième formule qui reprend la mise en avant et la création d'un site personnel qui sera réalisé en fonction de ses envies.

##### 2. Origine

Plusieurs indépendants dans notre entourage se sont plaints d'avoir trop de carte de visite dans leurs portefeuilles. Il était difficile pour eux de s'y retrouver avec toutes ces cartes et ce n'était pas pratique. C'est ainsi que nous avons eu l'idée de stocker ces cartes sur leur GSM et internet.

##### 3. Objectif du projet

Nous voudrions que cette entreprise grandisse et devienne un outil utile pour les professionnels comme les particuliers de trouver quelqu'un de compétent dans un domaine. Il permettrait aussi de trouver des particuliers ou entrepreneurs.



# Le marché

## 1. Présentation de l'équipe

Nous sommes 7 étudiants étant en dernière année de bachelier à l'EPHEC, dévorés par notre passion commune qu'est l'informatique. Nos profils sont complémentaires, certaines ayant des bonnes connaissances dans la conception d'une application, d'autres dans le web et encore dans le réseau. Nous réalisons cette start-up dans le cadre de notre cours de projet d'intégration qui nous permettra d'avoir une vue d'ensemble des différents domaines étudiés durant nos 3 années.

## 2. Le marché

Nous avons 2 clients ciblés :

- Le B2B, c'est-à-dire les entreprises. Nous comptons aller à des réunions de Networking comme BNI alliance.
- B2C, c'est-à-dire à destination des consommateurs particuliers. Par exemple un jardinier, un plombier.

Nous pensons réellement que notre produit apportera une solution au problème des cartes de visite qu'on ne retrouve plus ou que l'on perd trop facilement.

Plusieurs personnes de nos entourages qui sont indépendantes, ont encouragé cette idée. Il trouvait ça utile et leur faciliterais la vie.

Notre zone géographique ciblée est dans un premier temps la Wallonie. Notre site et application sortira en français. Nous prévoyons de s'étendre ensuite vers Bruxelles en traduisant nos services en anglais et pour finir nous nous dirigerons vers les clients situés en Flandre. Notre objectif sera de s'étendre sur ces différents marchés dans un laps de temps de 2 ans. Nous nous rendrons dans les différentes réunions d'affaires dans les différentes régions au fur et à mesure de notre expansion en Belgique.

### 3. Le modèle économique

Nous comptons rentabiliser notre produit avec le système des cartes de visite misent en avant sur le site vitrine. Les clients pourront payer pour première formule « premium » qui mettra leur carte de visite en avant sur le site.

Les clients pourront payer pour deuxième formule « pro » plus cher qui donne droit à un site personnalisé en fonction de ces besoins. Nous prendrons en compte les critères de la personne pour lui fournir un site basique.

Afin de faire de communiquer avec nos clients, nous avons créer une page Facebook et une page Twitter que nous tiendrons quotidiennement à jour.

De plus, ces pages serviront de contact direct avec le consommateur en cas de questions ou de problèmes.

### 4. La concurrence

Sur le marché des scannes de carte de visite, il y a un gros acteur c'est l'application « CamCard Lite » qui récence plus de 10 millions de téléchargement et selon eux 100 millions de profil. Il y d'autres plus petit concurrent comme « ABBYY » ou « Haystack » avec respectivement 500 000 et 100 000 téléchargements.

Nous nous sommes intéressés à nos concurrent sur la plateforme Google Play Store car nous prévoyons de se déployer d'abord sur celle-ci puis sur IOS.

Nos différents concurrents proposent de se créer un compte pour stocker les cartes de visites mais elles ne sont que disponible sur l'application. Ils permettent aussi de créer un contact à partir de la carte. Il y a moyen de partager la carte sur les réseaux sociaux. Le concurrent « CamCard Lite », lui fonctionne comme un réseau social avec un fil d'actualité et de suivre des entreprises ou particulier.

Nous nous démarquerons grâce à la disponibilité de notre service peu importe la plateforme (sur GSM, une tablette ou un ordinateur). Pour attirer les entrepreneurs et indépendant à souscrire à notre offre payante, nous mettrons en place un site personnalisable. Les clients qui payeront auront droit à un site en fonction de leur demande. Nous ne ferons pas de site d'e-commerce pour le prix de base mais bien un site mettant leur service ou produit en avant.

## Plan d'action

### 1. Stratégie marketing

Nous savons que la demande de ce type de service intéresse les entrepreneurs, particulier ou petite entreprise. Nos points de rencontre avec le client seront les réunions d'affaires (exemple : BNI, UCM, ...). Le lieu sera idéalement trouvé car durant ces réunions les entrepreneurs échangent leurs cartes entre eux. Ils pourront ainsi utiliser notre application pour stocker leurs cartes récupérées.

La prise de contact avec les particuliers se fera via des affiches ou des folders chez les commerçants de proximité qui seront soit déjà client. Nos clients nous permettront aussi de faire connaître nos services de recherche pour entrepreneur via leur site qui sera sur le nôtre. Nos utilisateurs payants partageront par mail ou sur leur carte de visite papier leur propre site vitrine.

### 2. Communication

Ces dernières années, la communication à travers les réseaux sociaux entre le client et les entreprises est primordiale. C'est pour cela que nous avons créé une page Facebook et Twitter qui permettra de prendre facilement contact. Les clients pourront nous envoyer des messages via Messenger sur Facebook ou sur Twitter via l'onglet « Tweets & réponses ».

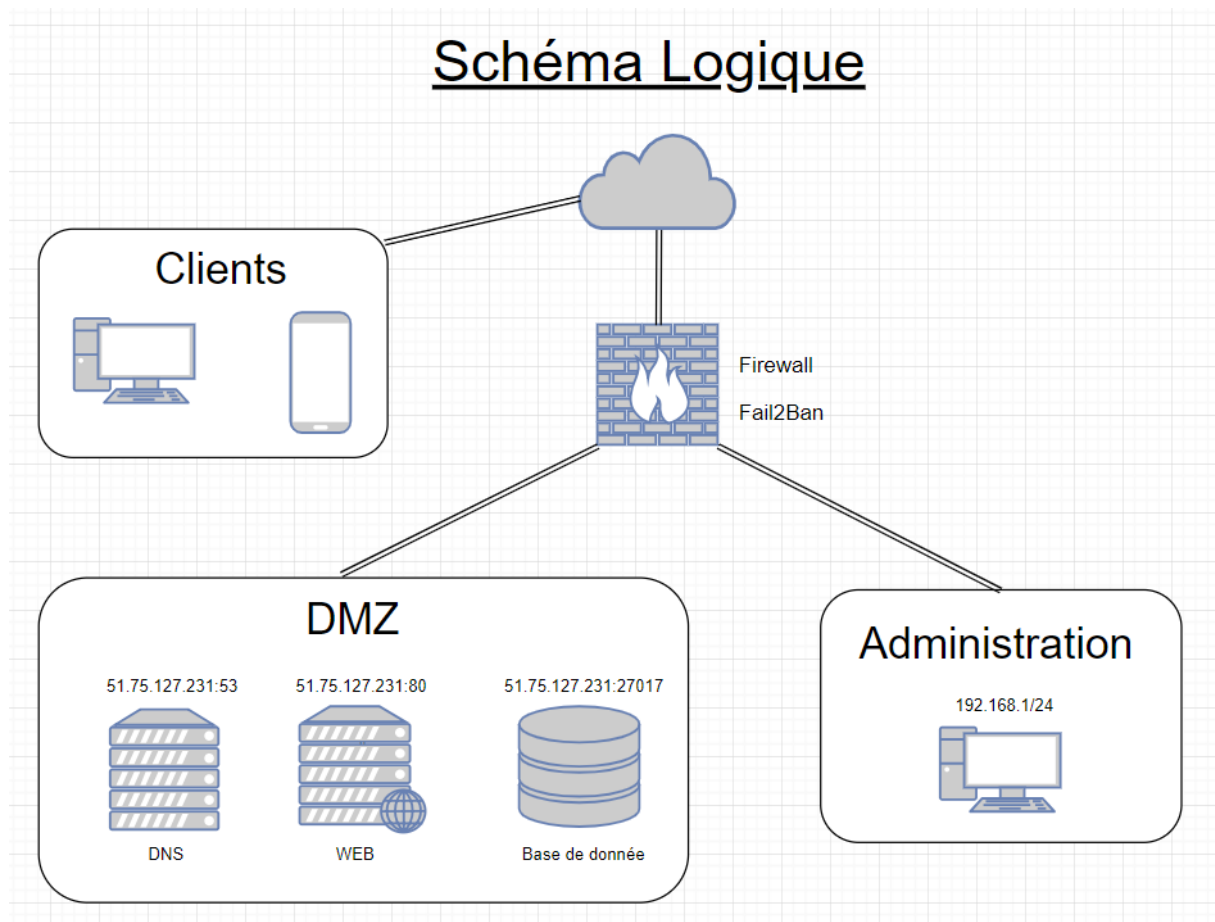
Nous aurons bien évidemment un téléphone privé si nos clients veulent avoir un échange de vive voix.

## Responsabilité sociale de l'entreprise

Notre entreprise permettra à l'entrepreneur d'éviter de distribuer leur carte de visite inutilement. Ils pourront ainsi partager leur carte de visite par mail à leur client pour que ceux-ci l'enregistrent sur l'application « SmartScan BC » ou le site. Nous aurons donc un peu d'impact sur l'utilisation de papier qui est lié à la déforestation des forêts.

## 9.2 Schémas

### 9.2.1 Schéma réseau logique



## 9.3 Sources

GSON : <https://github.com/google/gson>

Retrofit :

<https://github.com/square/retrofit>

<http://www.vogella.com/tutorials/Retrofit/article.html>

Android Image Cropper : <https://github.com/ArthurHub/Android-Image-Cropper>

Google Vision :

<https://developers.google.com/android/guides/releases?fbclid=IwAR07B3AFK76J4s5ejOMnEUkYsf1cEBNj-KTTdxMNZsQKvuxNBIVUfd6LdJs>

REST API :

<https://www.phpflow.com/php/create-php-restful-api-without-rest-framework-dependency/>

<https://www.phpzag.com/how-to-create-simple-rest-api-in-php/>

<https://stormpath.com/blog/secure-your-rest-api-right-way>

<https://coderwall.com/p/8wrxfw/goodbye-php-sessions-hello-json-web-tokens>

<https://guides.codepath.com/android/consuming-apis-with-retrofit>

<https://blog.cloudflare.com/using-guzzle-and-phpunit-for-rest-api-testing/>

<https://github.com/guzzle/guzzle>

<https://github.com/sebastianbergmann/phpunit>

<https://auth0.com/docs/api-auth/why-use-access-tokens-to-secure-apis>

[https://www.owasp.org/index.php/REST\\_Security\\_Cheat\\_Sheet](https://www.owasp.org/index.php/REST_Security_Cheat_Sheet)

Site Web :

<http://www.codeurjava.com/2016/12/formulaire-de-login-avec-html-css-php-et-mysql.html>

<http://stylescss.free.fr/couleurs.php>

<https://www.petite-entreprise.net/P-159-88-G1-faire-un-business-plan-exemple-de-business-plan.html>

<https://www.webtolosa.com/2015/06/17/exporter-une-table-mysql-au-format-csv-excel/>

<https://developers.google.com/web/tools/lighthouse/#devtools>

Étude quantitative :

<https://www.iweps.be/indicateur-statistique/population-active-taux-dactivite-taux-demploi-taux-de-chomage-administratifs-commune/>

<http://www.entrepriseswallonnes.be/pdf/Q01.pdf>