Récupération et traitement d'information via le crowdsourcing et la reconnaissance d'images

Matthieu Lombard

le cnam
École d'ingénieurs eicnam

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

CENTRE REGIONAL RHÔNE-ALPES

CENTRE D'ENSEIGNEMENT DE GRENOBLE

MEMOIRE

présenté en vu d'obtenir

le **DIPLOME D'INGENIEUR C.N.A.M.**

SPECIALITE : **INFORMATIQUE**

OPTION : **SYSTEMES D'INFORMATION**

par

Matthieu Lombard

Récupération et traitement d'information via le crowdsourcing et la reconnaissance d'images

Soutenu le : 3 Juin 2014

JURY

Président : M. Eric GRESSIER-SOUDAN (CNAM)

Membres : M. Claude GENIER (CNAM)
M. Philippe GARRAUD (CNAM)
M. Thierry FERRANDIZ (CEO - Le Bon Côté des Choses)
M. Rémy AMOUROUX (CTO - Le Bon Côté des Choses)

Claude Genier
Enseignant CNAM

Remerciements

Je tiens à remercier tout d'abord mon tuteur M. Rémy Amouroux, directeur technique et M. Thierry Ferrandiz *PDG du Bon Côté des Choses, de m'avoir donné l'opportunité de mener à bien ce projet au sein de leur société.*

Je voudrais également remercier toute l'équipe du Bon Côté des Choses, avec qui j'ai collaboré pendant presque 1 an, qui ont rendu ce stage très enrichissant professionnellement et humainement.

Je remercie mon tuteur du CNAM M. Claude Genier pour ses conseils dans la rédaction de mon mémoire ainsi que tous les membres du jury qui ont pris le temps de juger mon travail.

Pour terminer, un grand merci à ma famille et à toutes les personnes, professeurs du CNAM et connaissances professionnelles qui m'ont soutenu pendant ces 5 années d'étude d'ingéniorat.

Sommaire

Liste des figures

Liste des tableaux

Liste des abréviations

API	Applications Programming Interface
BCC	Le Bon Côté des Choses
BYOD	Bring Your Own Device
CSS	Cascading Style Sheets
DPH	Droguerie, Parfumerie et Hygiène
EAN	European Article Numbering
GPS	Global Positioning System
HTML	HyperText Markup Langage
HTTP	Hypertext Transfer Protocol
IHM	Interface (ou Interaction) Homme Machine
JSON	JavaScript Object Notation
LAMP	Linux, Apache, MySQL, PHP
Mo	MégaOctet
MOA	Maîtrise d'OuvrAge
NFC	Near Field communication

OCR	Optical Character Recognition
PHP	Hypertext Preprocessor
R&D	Recherche et Développement
REST	REpresentational State Transfer
S2LO	Social Shopping List Optimizer
SI	Système d'information
SURF	Speeded-Up Robust Features
SDK	Software Development Kit
SOA	Service-Oriented Architecture
URL	Uniform Resource Locator

Introduction

Dans un contexte mondial où la collecte de données est devenue décisive face au marché et à la concurrence, toutes les entreprises participent à cette course et agrègent de grandes quantités d'informations.

Ne faisant pas exception, la société Le Bon Côté des Choses propose un service mettant à disposition une très grande base d'information dans le domaine de l'alimentaire et DPH (Droguerie, Parfumerie et Hygiène). L'entreprise permet aux consommateurs et aux professionnels de comparer et d'analyser des produits de grande consommation au sein d'applications web et mobile. Ce double enjeu nécessite un système robuste permettant la récupération, l'optimisation et la gestion de l'information dans le but de délivrer un service de qualité. Effectivement, dans ce domaine fonctionnel, les informations sont très variées et il existe de nombreuses typologies de données (produits, enseignes, marques, prix, etc.) demandant une mise à jour quotidienne afin d'apporter un réel service utile et fédérateur.

L'objet du projet de ce mémoire consiste à proposer de nouveaux éléments technologiques innovants de récupération et de traitement de l'information qui reste une des principales problématiques de l'entreprise.
Au-delà d'informations publiques récupérables sur Internet ou d'informations monétisables, il existe une source d'information différente non exploitée encore sur laquelle est basé un concept appelé le « crowdsourcing » : la connaissance individuelle. En proposant une intégration complète d'un écosystème de gestion de l'intelligence et du savoir-faire des personnes utilisatrices du service, nous devrons être en mesure d'exploiter ce savoir et de répondre à ces exigences :

- Récupérer de nouvelles informations ;
- Qualifier des données existantes ;
- Classifier des données existantes ;
- Attirer de nouveaux utilisateurs et capitaliser sur les anciens.

Ce mémoire est structuré en 3 chapitres. Après une présentation du Bon Côté des Choses et du contexte de travail, nous définirons le crowdsourcing et présenterons les axes de conduite du projet ainsi que son positionnement au sein du système d'information existant. Enfin nous étudierons les différentes méthodes mises en œuvre via le crowdsourcing pour répondre au besoin de l'entreprise.

1 Le contexte

Ce projet de mémoire a été réalisé dans le cadre d'un stage de 9 mois dans la société Le Bon Côté des Choses. Nous allons présenter les services qu'elle propose et l'environnement dans lequel s'est inscrite cette étude.

1.1 Le Bon Côté des Choses (Le BCC)

1.1.1 Présentation

Le Bon Côté des Choses est une Startup[1] fondée en juin 2009 par Thierry Ferrandiz et basée sur le campus de Savoie Technolac au Bourget-du-Lac. Elle a développé son expertise dans le traitement de toutes les informations qui circulent depuis le consommateur jusqu'à l'ensemble des professionnels du « retail[2] ».

Le BCC se concentre en priorité sur l'optimisation des achats récurrents, alimentaires et DPH, selon une approche centrée sur les intentions d'achats déclarées et les préférences affinitaires de chaque individu.

Pour ce faire, la société a mis au point son Social Shopping List Optimizer (S2LO), un algorithme exclusif et propriétaire, en collaboration avec un laboratoire de recherche grenoblois, G-SCOP. Cet algorithme est au cœur du premier service de l'entreprise, un comparateur de prix de listes de courses. Il permet en effet d'effectuer d'importants calculs d'optimisation multicritères complexes sur des paniers de course d'environ 80 produits.

Ce service est en ligne en mode béta depuis le 7 Janvier 2013. Une version plus aboutie disponible depuis le 26 août 2013 réunit chaque mois plus de 50 000 utilisateurs dont une moitié sur le site Internet, l'autre via les applications mobiles disponibles sur l'Apple Store et Google Play (Figure 1).

Figure 1 : Développement du BCC

[1] Jeune entreprise non lancée sur le marché commercial et en forte phase de développement d'un produit ou d'une idée qui fait la plupart du temps l'objet de levée de fonds.
[2] Commerce de détail. Maillon final de la chaîne de distribution, il va de l'achat de produits auprès d'un fournisseur, jusqu'à la revente de la marchandise à un client en magasin.

1.1.2 Offre de services

Le BCC recueille et agrège de l'information provenant des fabricants, des supermarchés, des magasins de proximité et des acteurs du e-commerce.

Ces informations sont ensuite proposées aux consommateurs par l'intermédiaire de son site Internet (www.leboncotedeschoses.fr) et de ses applications mobiles. Sur ces plateformes, l'utilisateur peut préparer sa liste de courses en naviguant dans les catégories de produits au sein de l'application et ainsi obtenir un comparatif précis de son panier sur un panel de marchands et de critères importants. Les utilisateurs constituent leur foyer (regroupement de personnes) et partagent leurs listes de courses entre eux. Ils précisent leurs préférences produits, leurs contraintes logistiques et leurs obligations de santé qui apporteront des filtres précis au résultat du comparateur. Le BCC combine sa base de données de produits au plus près des déclarations de chaque utilisateur, de ses historiques de comportement et de sa localisation géographique au sein du Social Shopping List Optimizer (S2LO).

Cette technologie propriétaire permet au consommateur de trouver et de décider où et comment acheter au meilleur rapport qualité/coût, en ligne ou chez un commerçant de proximité (Figure 2).

Figure 2 : L'offre du BCC

Le consommateur utilise également les applications mobiles du BCC jusqu'en magasin pour scanner ses produits, compléter l'information rendue disponible aux autres utilisateurs et rayer leurs produits de la liste une fois ajoutés dans leur caddy.

Cette approche orientée vers le consommateur permet au BCC de devenir un précieux outil de compréhension du marché pour les professionnels de la grande consommation :

- Le BCC est une plateforme de business intelligence qui combine les technologies d'exploration et d'analyse de très fortes volumétries d'information provenant de sources multiples. Elle permet de fournir aux professionnels des marqueurs dans des domaines aussi stratégiques que le positionnement concurrentiel ou les évolutions de tendances de consommation ;
- Le BCC est aussi un canal d'amélioration des ventes pour les professionnels au moyen :
    - du relai d'opérations promotionnelles numérisées (bons de réductions)
    - de transferts de commandes (liste de courses) clients vers des marchands en ligne
    - d'acquisition de trafic en magasin pour les distributeurs traditionnels

Fort d'une des plus grosses bases de données produits dans le domaine, le BCC cherche en permanence de nouveaux moyens de l'enrichir, de la qualifier et de la maintenir à jour. Points sur lesquels portera le projet de mémoire.

1.2 Environnement de travail

1.2.1 L'équipe

Le BCC est une équipe de douze personnes segmentée en deux pôles de compétences principaux regroupés sous son directeur général Thierry Ferrandiz (Figure 3).

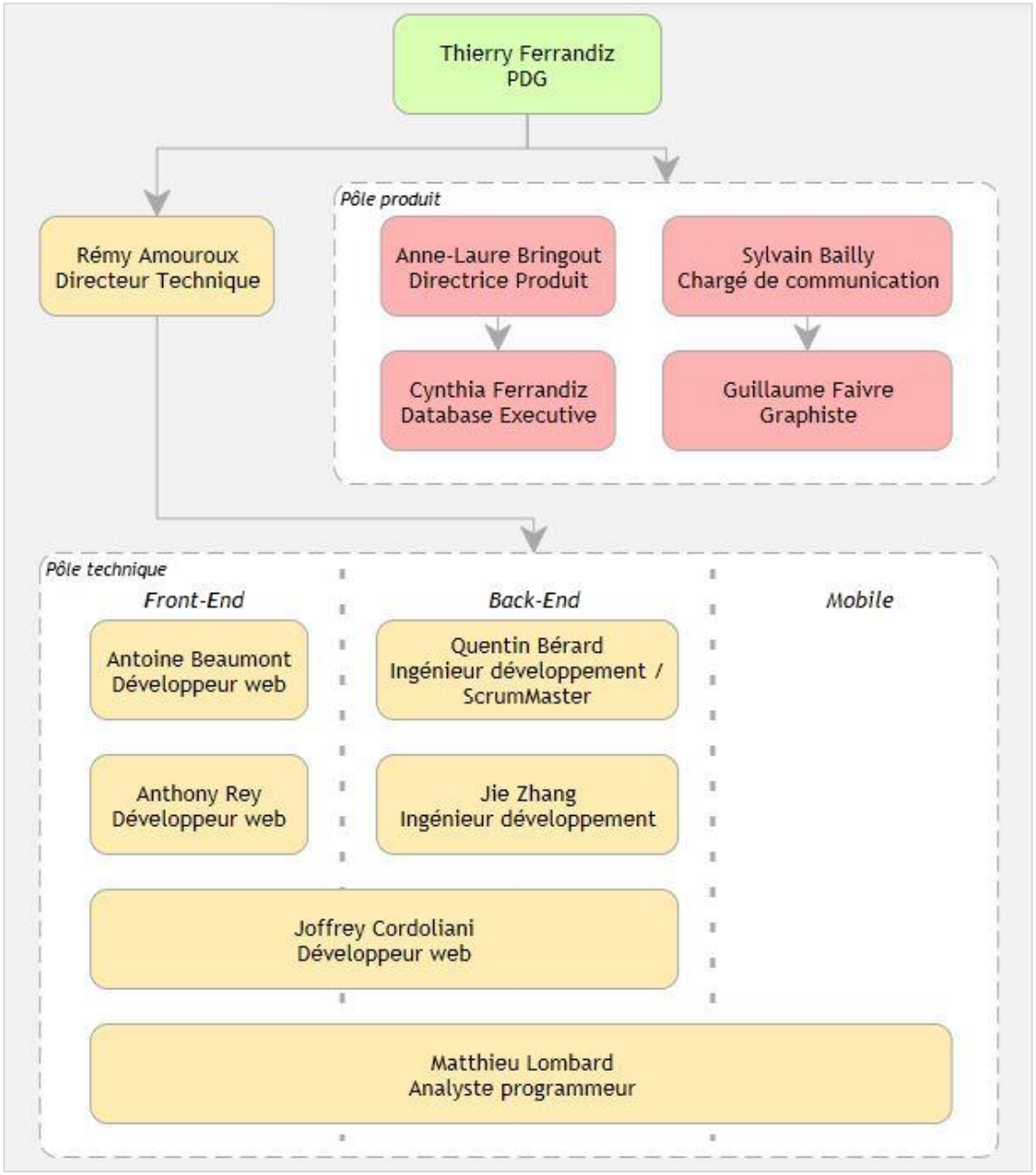


Figure 3 : Organigramme du BCC

Le pôle « Produit » est garant de l'aspect métier et des données des applications promulguées par la société. Cette équipe a la connaissance des lignes directrices de la stratégie de l'entreprise. Elle joue le rôle de MOA et représente le client (le BCC) et les utilisateurs en étant le support des autres pôles pour maximiser la valeur du produit développé. Elle intègre aussi une partie « Communication / Design » qui gère toutes les réalisations graphiques web (site Internet), mobile et print (affiches, flyers, plaquettes) représentant l'image du BCC. Elle est aussi en charge de l'animation des réseaux sociaux et la communication presse.

Le pôle « Technique », dirigé par Rémy Amouroux directeur technique et tuteur de mon stage, est scindé en trois compétences :

- La partie « Front-End » en charge du développement des IHM Web, intégration des designs et interactions utilisateurs ;
- La partie « Back-End » en charge du développement des processus métiers, flux et base de données ;
- Pour ce qui est des développements mobiles, ils sont externalisés auprès d'experts dans les deux solutions propriétaires que sont Apple iOS et Google Android.

1.2.2 Ma fonction au sein de l'entreprise

Arrivé en janvier 2013, l'objectif fixé par le directeur du BCC et mon tuteur a été la gestion du projet dans son ensemble. Ce projet devait s'intégrer comme une brique de l'architecture en place construite et maintenue quotidiennement par les équipes de développement du BCC. Au sein de cette start-up aux priorités très en mouvement, mon travail fut de positionner des ressources en fonction des disponibilités pour aider à la mise en place du programme. Une grosse part du travail a aussi été la définition du concept ainsi que la conception et le développement de plusieurs de ces fonctionnalités.

Le projet demandant des développements allant d'une action utilisateur web et mobile aux traitements de données, j'intègre donc le pôle « Technique » et travaillerai transversalement sur des fonctionnalités couvrant les trois compétences du pôle technique.

1.2.3 Gestion de projet au BCC

Au-delà de l'entrée dans une équipe, je rentre aussi dans un processus de gestion de projet cadré, basé sur une méthode agile de type Scrum. Ce type de méthode repose sur des cycles de développement itératifs et adaptatifs en fonction des besoins évolutifs du client. Elles permettent d'impliquer l'ensemble des collaborateurs ainsi que le client dans le développement du projet.

La méthode Scrum, créée en 2002, dont le nom est un terme emprunté au rugby qui signifie « la mêlée », s'appuie sur le découpage du projet en « releases » qui correspondent à des livraisons de versions. Chaque version est ensuite découpée en itérations encore nommées « sprints » (Figure 4).

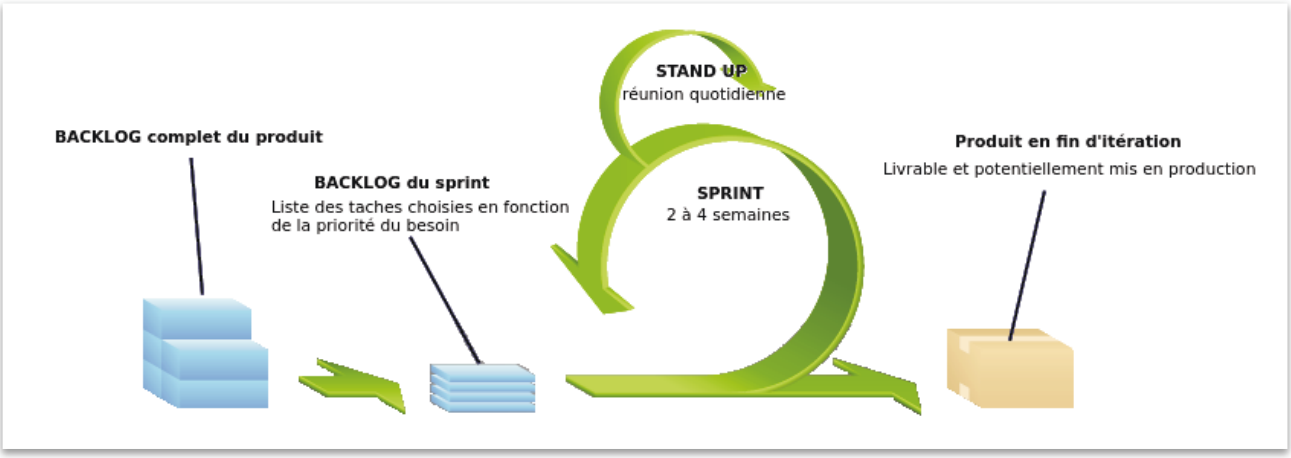


Figure 4 : Une itération selon la méthode agile Scrum [1]

Un sprint peut avoir une durée qui varie généralement entre une semaine et un mois. Avant chaque sprint les tâches sont estimées en termes d'effort. Ces estimations permettent à la fois de planifier les livraisons mais aussi d'estimer le coût de ces tâches auprès du client ; le client étant ici le BCC donc l'équipe « Produit ». Les tâches qui font l'objet d'un sprint constituent ce que l'on appelle un « backlog » du produit éventuellement livrable à la fin du sprint. La méthode Scrum est aussi caractérisée par un « stand up meeting » quotidien, dans lequel les collaborateurs (développeurs, graphistes, et équipe produit) indiquent tour à tour les tâches qu'ils ont effectuées la veille, les difficultés rencontrées et enfin ce sur quoi ils vont poursuivre leur travail le jour suivant. Cela permet d'évaluer l'avancement du projet, de mobiliser des ressources là où cela est le plus nécessaire mais aussi de venir en aide aux collaborateurs rencontrant des difficultés lorsque celles-ci ont déjà été rencontrées auparavant par d'autres membres de l'équipe.

L'avancement d'un sprint est jugé sur un reporting visuel avec affichage mural à base de post-it (Figure 5).

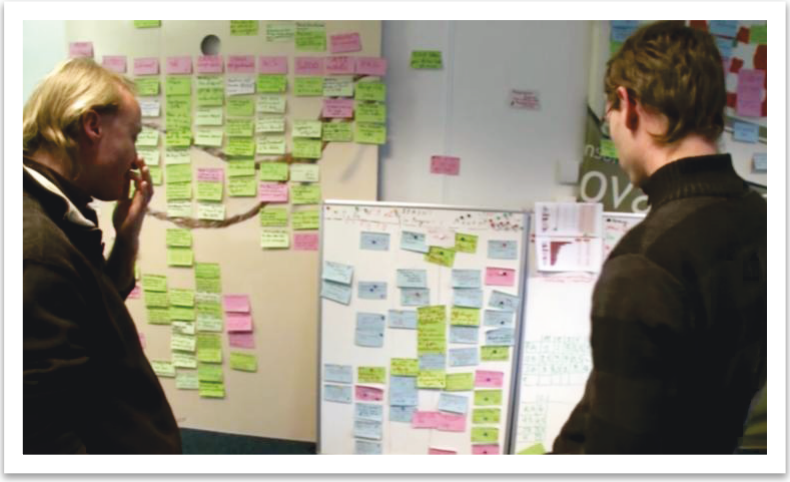


Figure 5 : Daily stand up meeting

Enfin, la méthode Scrum est également caractérisée par la présence, parmi les collaborateurs, d'un « Scrum master » qui est une personne chargée de veiller à la mise en application de la méthode et au respect des objectifs. Un membre du pôle technique a cette responsabilité. [1]

2 Présentation du projet

Le BCC propose des services autour des produits de grande consommation et le projet va s'articuler au cœur de cette notion. De ce concept nous allons sortir la problématique, cadrer et définir l'orientation du projet.

2.1 La problématique

Les services du BCC se déclinent selon deux axes proposant des offres complémentaires à destination des consommateurs et des professionnels. Au centre de ces offres de services se trouvent une entité dominante qui constitue le cœur du métier, c'est le produit alimentaire ou produit DPH.
Autour de ces produits vont s'articuler d'autres entités qui vont venir donner de la valeur à cette donnée.
Des propriétés intrinsèques à l'article :

- L'EAN (code produit)
- Le nom
- La marque
- La description
- La composition
- L'emballage (on parlera de packaging)

Des propriétés fortes rattachées à l'article :

- Le prix
- La promotion
- Le magasin de distribution

La variable la plus importante est la qualité de ces données sur le produit, qui sera restituée à l'utilisateur et qui constituent la base d'informations du BCC. Elles sont tirées pour la plupart de « webcrawlers » (robot d'indexation) qui parcourent les sites Internet marchands d'enseignes nationales (Figure 6) dans le but de les récupérer.

Figure 6 : Enseignes référencées par le BCC

Afin de proposer un réel service utile à l'utilisateur, le BCC se doit de répertorier le maximum de produits. Concrètement, un site marchand peut proposer jusqu'à 30 000 références produits. Pour se rendre compte du volume, il faut faire la distinction entre une référence produit et une offre produit (Figure 7). Une offre produit est la déclinaison de la référence produit dans chaque magasin de l'enseigne avec un prix potentiellement différent. Ce qui monte, pour une enseigne de 400 magasins avec 10 000 références produits, à un total de 4 millions d'offres différentes à récupérer.

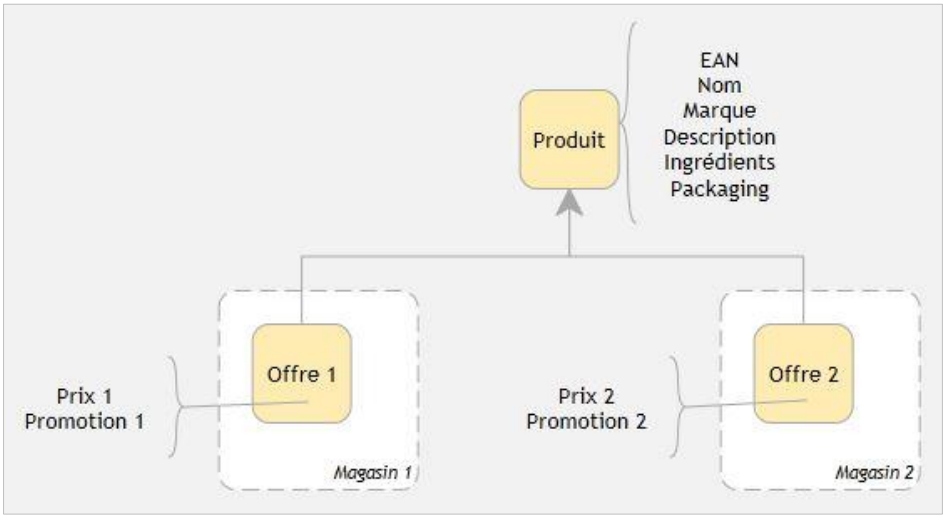


Figure 7 : Caractéristiques produit et offre

L'un des objectifs final est d'arriver à comparer les prix d'un produit dans différents magasins de même ou de différente enseigne. Unitairement, cette comparaison doit donc se faire entre deux offres strictement identiques du point de vue de leurs caractéristiques produit. Il faut donc dans ce flux de produits provenant de différentes sources arriver à raccrocher les offres similaires. L'EAN, un code à 13 chiffres attribués aux produits est là pour assurer la similitude des offres entre elles car il caractérise l'identification unique d'un produit commercialisé en Europe (généralement associés à un code-barres imprimé sur le produit).
Malheureusement, d'un site marchand à l'autre, la pertinence des attributs liés au produit récupéré n'est pas du tout la même. Certains sites ne proposeront pas certains produits ou l'EAN ce qui ne permettra pas sans complément d'information de proposer ce produit à l'utilisateur dans l'ensemble de ses comparaisons. Sans ce code il faut donc trouver d'autres méthodes pour rapprocher ces offres. D'autres sites ne proposeront pas les caractéristiques d'un produit comme la description, l'image, etc., il faut donc trouver de nouveaux moyens pour y avoir accès.

En plus des sites Internet marchands, il y a d'autres éléments pouvant apporter des informations sur un produit :

- Les tickets de caisse : nom, prix, magasin de distribution
- Les étiquettes en rayon : EAN, nom, prix, poids ou volume
- Les emballages : EAN, nom, marque, description, ingrédients, valeurs énergétiques, poids ou volume

Dans cette optique, l'objectif est de proposer de nouvelles façons de récupérer des données de produits et de qualifier l'information déjà à notre disposition. Nous verrons par la suite que le concept du « crowdsourcing » est au centre de la solution à cette problématique et que l'analyse et la reconnaissance d'images feront partie des nombreux outils permettant de développer cet environnement.

Dans les prochaines parties nous définirons ce qu'est le crowdsourcing et les différentes formes sous lesquelles il peut apparaître. Nous verrons de quelle manière ces mécanismes ont pu être mis en place au BCC en termes de ressources projet, autant techniquement au sein du système d'information existant que fonctionnellement au sein de la communauté d'utilisateurs.

2.2 Orientation du projet

A l'initiative le projet se concentrait sur la récupération d'informations sur trois supports bien précis au travers de photos prises par les utilisateurs via leurs smartphones :

- Le ticket de caisse
- L'étiquette en rayon
- Le packaging du produit

Après une analyse du besoin dans sa globalité, le projet demandait l'intégration de nombreux outils supplémentaires en amont de la récupération d'informations au travers d'images. Une rapide réflexion a montré qu'il fallait un environnement complet basé autour du crowdsourcing et intégrable au sein de l'architecture des services du BCC pour accueillir cette fonctionnalité avancée. Le traitement d'image a donc été réorienté comme étant une partie de la réalisation sur la totalité du projet. Il a donc fallu revoir les priorités pour absorber la création de cette mécanique.

Parmi ces trois cibles, la priorité a donc été posée sur l'extraction d'informations des images de packagings de produits qui permet de qualifier fortement la base de données des offres produits. Ces images pouvant provenir de photos utilisateurs ou d'images des sites marchands. Nous détaillerons plus loin les types d'informations concernés.

En ce qui concerne le traitement des tickets de caisse, il était envisagé de permettre aux utilisateurs du BCC de prendre en photo leurs tickets grâce à leurs smartphones afin de mettre à jour le prix des produits en base de données. Sur la photo devaient être reconnus les textes, les produits et leur prix, grâce à un logiciel d'OCR. La solution a été mise en attente car nous avons préféré lancer une première analyse de la concurrence et des solutions existantes qui permettrait d'effectuer cette tâche sans développement coûteux. Cette étude sort du cadre du projet.

De la même manière pour les étiquettes en rayon, un résultat plus fiable et moins coûteux en termes de réalisation pourrait être réalisé grâce à des technologies sans-fil type NFC plutôt qu'à l'aide d'algorithmes de reconnaissance d'images. Une première analyse est en cours d'étude avec certains distributeurs alimentaires qui comptaient investir dans un parc de scannettes[3]. Plutôt que d'investir dans ce parc coûteux, l'idée est de proposer une solution en mode BYOD[4], c'est-à-dire via une application installée sur les smartphones des clients qui servirait de scannettes personnelles lorsque l'on fait ses courses. Cela permettrait de se servir de la puissance et de la technologie des smartphones de tout un chacun. Cette étude sort aussi du cadre du projet.

[3] Scannette est le nom donné au lecteur de code barre fourni par les grandes surfaces pour permettre à ses clients de scanner les produits qu'ils mettent en caddie.
[4] Bring Your Own Device ou Apportez vos appareils personnels en français.

2.3 Le crowdsourcing

La première étape du projet est déjà de donner un cadre au projet en définissant les différents modes de crowdsourcing que l'on va pouvoir mettre en place ainsi que la manière de l'inclure et de le promouvoir au sein des services du BCC. Effectivement, il existe de nombreuses typologies à étudier selon les résultats que l'on souhaite obtenir.

2.3.1 Définition

Le mot Crowdsourcing est construit à partir du mot anglais « crowd » qui signifie « la foule » et d'une contraction du mot « sourcing » que l'on peut traduire par « approvisionnement ». Le Crowdsourcing est une forme d'externalisation (Outsourcing) qui ne s'adresse pas à d'autres entreprises mais à la foule (Figure 8). Il s'agit d'une notion qui apparait pour la première fois en 2006 lors d'une discussion sur un Forum Internet. C'est l'article, paru dans le journal en ligne Wired, écrit par Jeff Howe et Mark Robinson qui popularisera ce nouveau terme.

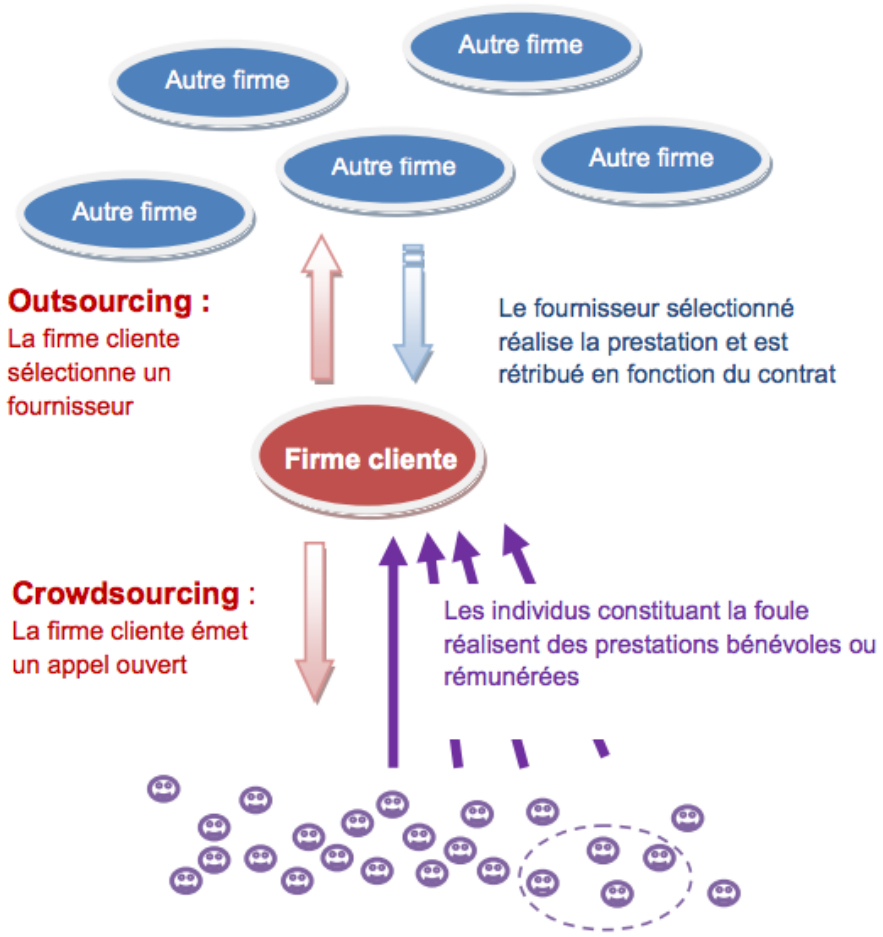


Figure 8 : Crowdsourcing et Outsourcing [2]

Le service peut être produit individuellement ou par des communautés informelles, il peut reposer sur un talent individuel ou sur des capacités collectives. La particularité du fonctionnement du Crowdsourcing est que plusieurs individus/communautés peuvent travailler simultanément sur un même projet, l'entreprise choisissant à la fin le projet qui correspond le mieux à ses besoins. Pour l'entreprise cliente l'avantage est conséquent puisqu'elle permet d'externaliser le risque d'échec, puisqu'elle ne paye que les produits ou services qui répondent à ses attentes. [3]

Il est important de souligner que l'appel à la foule doit être ouvert à tous et non pas limité à un public présélectionné. Une autre particularité constitutive du Crowdsourcing est que l'entreprise cliente n'opère pas de sélection a priori des contributeurs éventuels.

2.3.2 Les typologies de crowdsourcing

Nous pouvons observer depuis des années une expansion du champ d'application du crowdsourcing. Avant Internet, les concours photos ou d'affiches sont de bons exemples de manifestations par la foule en tant qu'apporteur de contenus. Maintenant, le Crowdsourcing peut se décliner en différentes typologies, en voici une liste non-exhaustive :

Crowdvoting : il intervient quand un site rassemble les opinions et jugements d'un grand groupe de personnes sur un sujet précis. Par exemple, le site lafraise.com sélectionne des t-shirts qu'il imprime et vend suivant les votes sur les designs préférés des utilisateurs.

Creative Crowdsourcing : il promeut des projets créatifs tels que la conception graphique, l'architecture, la conception de vêtements, l'écriture, etc. lafraise.com est aussi un exemple dans ce domaine car il propose au designer d'exprimer sa créativité au sein de leur plateforme. Leur réalisation sélectionnée par vote est ensuite imprimée en édition limitée et vendu par le site web contre une rétribution financière et cession des droits d'utilisation de l'œuvre.

Crowdfunding : c'est un processus de financement de projets par une multitude de personnes qui contribuent via des « petits » montants à l'attente d'un objectif monétaire. Un outil crowdfunding bien connu est kickstarter.com, qui est le plus grand site de financement de projets créatifs. Il a recueilli plus de 100 millions de dollars, en dépit de son modèle tout-ou-rien qui exige que l'on atteigne l'objectif monétaire proposé afin d'acquérir de l'argent. Tout un chacun peut participer au financement de projets en lesquels il

croit par l'intermédiaire de la plateforme. Le BCC a lancé dans le cadre de son financement une opération auprès d'un organisme de crowdfunding : Anaxago [4].

"Wisdom of the crowd" : c'est un procédé qui consiste à prendre en compte l'opinion d'un groupe d'individus plutôt que celui d'un seul expert afin de répondre à une question. L'idée de l'intelligence collective prend tout son sens sur Internet où des gens venant de tous horizons peuvent répondre en temps réel sur une même plateforme.

Implicit crowdsourcing : cette méthode est un peu à part et moins évidente car les utilisateurs ne savent pas nécessairement qu'ils contribuent. Cependant elle peut être très efficace couplée à la méthode de « Microwork » dans la réalisation de certaines tâches. Plutôt que de demander aux utilisateurs de participer activement dans la résolution d'un problème complexe ou de fournir des informations, l'implicit crowdsourcing consiste à laisser l'utilisateur faire entièrement une autre tâche et récupérer des informations en fonction de ses actions. Une autre utilisation bien connue de cette méthode est reCAPTCHA [5] qui demande aux utilisateurs de résoudre des captchas[5] afin d'améliorer un processus de numérisation de livres.

Nous nous attarderons plus longuement sur l'approche « **Microwork** » qui est dans notre cadre la plus adaptée au traitement de données en masse correspondant à la stratégie de l'entreprise.

Le microwork est une série de petites tâches qui, unifiées, contribuent à l'avancée d'un plus gros projet. Elles sont réalisées par de nombreuses et diverses personnes au travers d'Internet. Considéré comme la plus petite unité de travail dans une ligne d'assemblage virtuel, il est le plus souvent utilisé dans la résolution de tâches pour lesquels aucun algorithme efficace n'a encore été mis au point et exige la compétence humaine pour le compléter de manière fiable. Le processus de division et de distribution de ce travail au travers d'Internet est appelé « microtasking ». C'est sur cette technique que vont se baser nos développements.

Plusieurs projets basés sur cette technique montrent toute l'efficacité de la solution et nous ont apporté des concepts très intéressant pour modéliser notre propre service. Voici quelques exemples :

[5] Un captcha est un test permettant de différencier de manière automatisée un utilisateur humain d'un ordinateur au travers d'images déformées

reCAPTCHA

Un captcha est un système de filtrage visant à établir une distinction entre internautes et robots virtuels (exemple Googlebot). Ainsi l'utilisateur d'un site est amené à décrypter une séquence de caractères afin de pouvoir poursuivre sa visite du site. Un reCAPTCHA comprend quant à lui deux termes à déchiffrer (Figure 9). Le premier est un mot connu qui sert à s'assurer que « l'individu » en face de l'écran est bien un internaute. Le second est un mot rejeté par un logiciel de reconnaissance de caractère et qu'il s'agir de déchiffrer. Le 17 Septembre 2009, Google a annoncé l'acquisition de la société reCAPTCHA. En mobilisant des compétences de déchiffrage d'internautes à travers le monde, reCAPTCHA contribue de manière très significative[6] au programme de numérisation d'ouvrages et de périodiques mené par Google. [5]

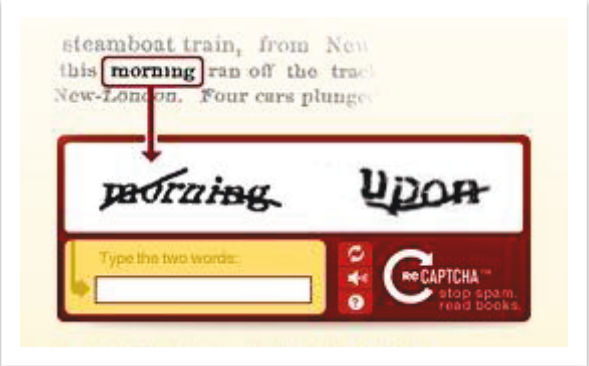


Figure 9 : Exemple d'un bloc reCAPTCHA affiché sur un site Internet [5]

Amazon Mechanichal Turk

Amazon Mechanical Turk est un marché pour le travail qui nécessite l'intelligence humaine. Le service Web Mechanical Turk permet aux entreprises d'accéder via un programme à ce marché et à une main d'œuvre variée à la demande. Les développeurs peuvent tirer profit de ce service pour construire une intelligence humaine directement dans leurs applications. Même si la technologie informatique continue de progresser, il reste encore beaucoup de choses que les humains peuvent faire bien plus efficacement que les ordinateurs, comme identifier des objets dans une photo ou une vidéo, effectuer une déduplication de

[6] Le site ReCaptcha revendique plus de 30 millions de ReCaptcha rempli par jour sur plus de 100 000 sites utilisant le système. ReCaptcha sert à la numérisation des archives du New York Times et de Google Books. En 2012, 30 ans d'archives du New York Times ont été numérisés et les responsables du projet espèrent avoir complètement numérisé les années restantes avant fin 2013.

données, transcrire des enregistrements audio ou rechercher des détails de données. Les entreprises ou développeurs nécessitant des tâches effectuées (appelées Human Intelligence Tasks ou « HIT ») peuvent utiliser les API robustes de Mechanical Turk pour accéder à des milliers d'employés hautement qualifiés, à faible coût, mondiaux et à la demande et intégrer ensuite via un programme les résultats de ce travail directement dans leurs processus et systèmes d'entreprise. Mechanical Turk permet aux développeurs et aux entreprises d'atteindre leurs objectifs plus rapidement et à un coût moindre qu'il ne le leur était possible auparavant. [6]

Wikisource

Wikisource est une bibliothèque libre qui propose des textes dans le domaine public provenant de livres numérisés et passés dans un logiciel de reconnaissance de texte. Les livres sont ensuite corrigés manuellement et collaborativement par les internautes pour qualifier la mise en page ou les erreurs de reconnaissance de l'OCR. Les lecteurs ont alors ensuite à disposition des textes librement diffusables et interrogeables en recherche textuelle. Plusieurs bibiothèques d'archives utilisent le service et plus de 135 000 textes sont actuellement disponibles. [7]

Pour utiliser le service, il faut se rendre sur le site de wikisource et choisir un livre puis une page qui n'ont pas encore été traités. Apparaissent ensuite en parallèle la page numérisée et le texte extrait de cette page par OCR (Figure 10).

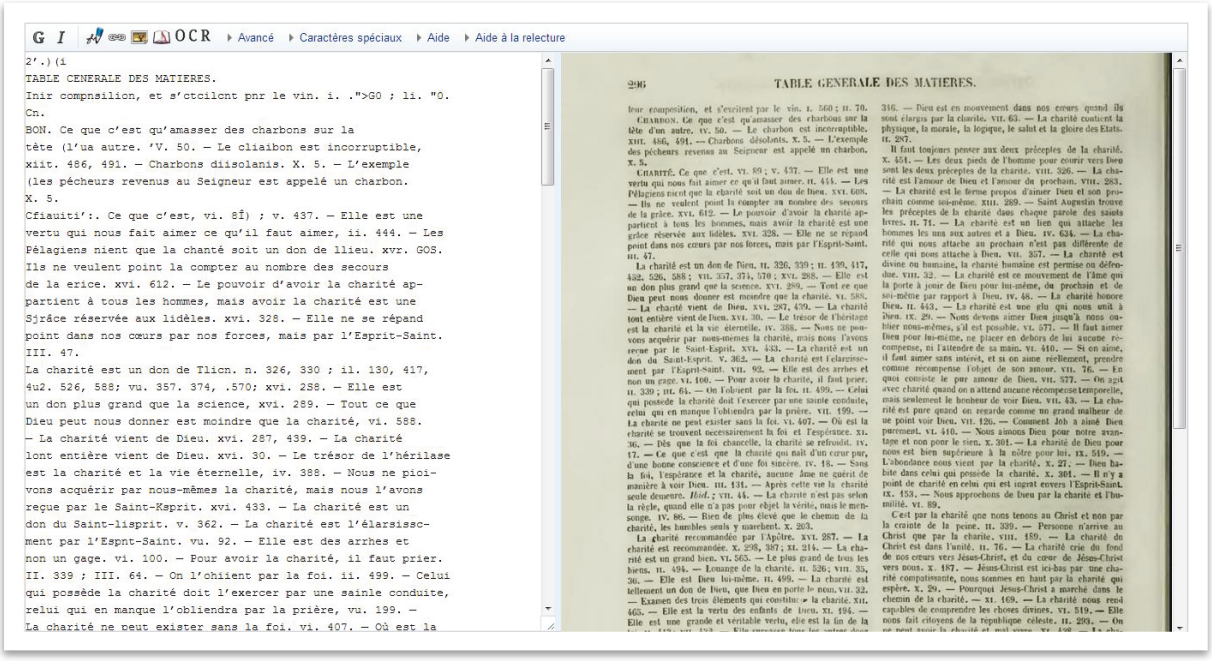


Figure 10 : Wikisource - Création d'une page

Nous pouvons voir en zoomant sur une zone de la page à numériser que la reconnaissance de texte n'est pas optimale : « leur composition » est reconnu par l'OCR en « Inir compnsilion » (Figure 11). Des contributeurs peuvent donc venir et éditer le texte pour le faire correspondre à l'image et ainsi participer à la qualification la base de données de wikisource. Ces actions sont faites bénévolement.

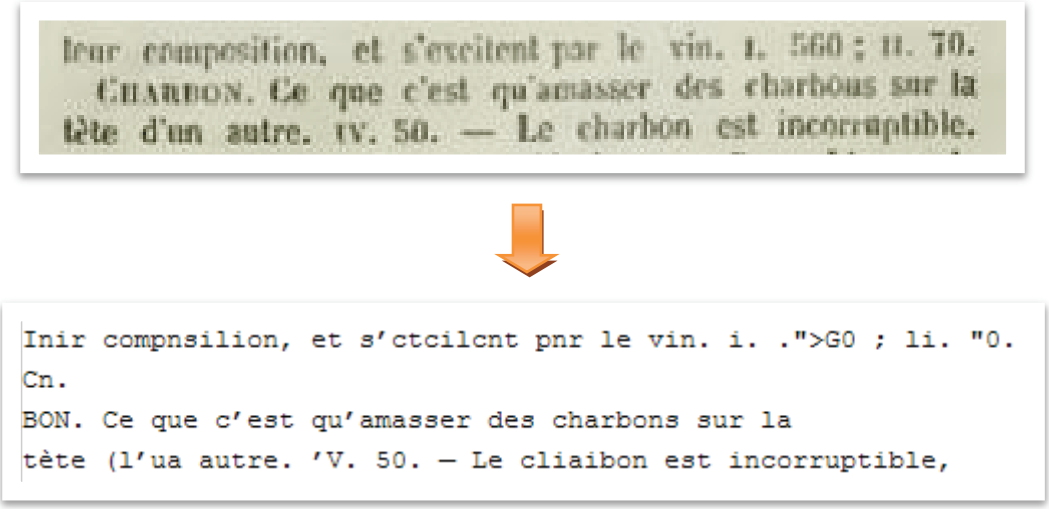


Figure 11 : Wikisource – Zoom sur la reconnaissance de texte

Après avoir définit le concept de Crowdsourcing, nous allons voir la mise en place qui en a été faite au BCC.

2.3.3 Le « Caddy Trophy »

2.3.3.1 Le concept

Avant d'entamer une réflexion purement technique, il convient de créer un écosystème permettant d'attirer les utilisateurs et de leur donner envie de participer activement. Le but étant d'allier sous une même bannière toutes les mécaniques de crowdsourcing qui seront diffusées aux membres du BCC. Pour que les retours soient intéressants, l'enjeu est d'arriver à créer une communauté qui participerait de manière récurrente à l'accomplissement de multiples micro-tâches variées. Pour que l'impact et l'adhésion soient les plus forts possible, il fallait créer une véritable identité et un intérêt réel pour les participants. Nous avons donc créé une compétition, le « **Caddy Trophy** » sous laquelle seront diffusées sous la forme d'un jeu participatif, toutes les actions de crowdsourcing que le BCC mettra en place. Cette compétition doit donc être évolutive dans l'ajout ou le retrait de tâches mais elle doit aussi pouvoir récompenser les gagnants. Dans le document les tâches utilisateurs seront appelées « actions ». La récompense devra avoir un rapport avec le périmètre fonctionnel du BCC, c'est-à-dire, en relation avec l'alimentaire, droguerie, parfumerie et hygiène (DPH).

Dans sa première version, les règles du « Caddy Trophy » sont simples. La compétition sera hebdomadaire et chaque action sera récompensée par une valeur en points. Celui qui remportera le plus de points dans la semaine sera déclaré gagnant et sera remboursé de ses courses alimentaires et DPH de la semaine sur présentation de justificatifs et dans la limite de 120€.

2.3.3.2 Les actions réalisables par les utilisateurs

Afin de donner du contenu à la compétition, il a fallu définir des actions permettant de gagner des points et remporter les gains de la semaine. Pour définir ces actions, avec l'équipe produit, nous avons répertorié toutes les fonctionnalités déjà en place dans les services web et mobile puis nous avons extrait et trié les actions utiles en termes d'apport d'information pour le BCC (Tableau 1).

Actions	WEB	MOBILE	Points
Parrainage, invitation à s'inscrire au BCC	✔		200
Ajout de nouveaux produits non référencés		✔	200
Catégorisation d'un produit		✔	200
Ajout d'une photo à un produit qui n'en a pas		✔	50
Partage du lien vers le site Internet sur Facebook	✔		40
Signalisation d'une anomalie sur un produit		✔	35
Partage des vidéos promotionnelles sur Facebook	✔		30
Inscription au BCC	✔		25
Utilisation du service, lancement du comparateur S2IO sur une liste de course	✔	✔	20

Tableau 1 : Fonctionnalités à prendre en compte dans le cadre du Caddy Trophy

Chaque action se voit attribuée des points qui récompenseront les utilisateurs et de nouvelles actions devront pouvoir être intégrées aisément.

2.3.4 Diffusion et communication

Ce programme d'« incentive[7] » doit devenir un pilier de la stratégie de fidélisation et de communication du BCC. Il doit permettre de faire entrer l'utilisateur dans un processus d'engagement et de créer une dynamique dans la communauté.

Afin d'avoir rapidement cet outil à disposition sur la plateforme du BCC, nous avons planifié la mise en ligne d'une première version au bout de 3 mois qui contenait la gestion des actions utilisateurs sur le web et sur la version iOS des applications mobiles. La seconde version est sortie ensuite à l'automne 2013 permettant d'inclure de nouvelles fonctionnalités et de statuer sur l'accueil accordé à la première version.

Chaque diffusion doit donc être accompagnée par des actions hebdomadaires récurrentes de communication sur les réseaux sociaux que nous avons donc mis en place avec les acteurs de cette partie au sein de la société. Ces annonces doivent mettre en avant le top 5 des gagnants de la semaine, le gagnant de la récompense, son nombre de points et promouvoir la semaine suivante.
Des actions plus ponctuelles doivent aussi être menées pour essayer de créer du contenu sur Internet autour du concept. Exemple d'une interview auprès d'une blogueuse8 pour la sortie de la nouvelle version à l'automne 2013 en Annexe 1.

2.4 Conduite du projet

2.4.1 Planning

L'architecture technique du BCC est un système établit sur lequel travaillent en parallèle plusieurs développeurs selon leurs domaines de compétences. La particularité du projet est qu'il va avoir des impacts à plusieurs niveaux de cette architecture. Il doit pouvoir s'intégrer avec le minimum d'incidence sur les autres modules et fonctionnalités du service mais nécessitera l'intervention de développeurs front-end, back-end et mobile.

[7] « Incentive » qui donne le nom au projet est un mot anglais signifiant « motivation ». En e-business, cela représente le fait d'inciter une personne à effectuer une action précise selon un système basé sur des récompenses.
[8] Rédactrice de blog.

Le BCC ne dispose pas de crédit temps homme conséquent dans la réalisation de ce programme car l'équipe de développement doit constamment faire évoluer le service principal. Donc, en termes de ressources humaines sur le projet, quelques tâches seront affectées par pôle de compétences à chaque sprint. Afin d'avoir une vision globale sur la mobilisation des ressources, nous avons établi un macro-planning (Figure 12).

Le projet a été scindé en trois releases de 12 semaines comprenant différentes fonctionnalités :

- V1 du Caddy Trophy
    - Concept et règles ;
    - Implémentation des règles métiers ;
    - Intégration sur le site Internet ;
    - Intégration sur la version mobile iOS ;
    - Supervision via l'intranet ;
    - Communication V1 du Caddy Trophy.
- V1.2 du Caddy Trophy
    - Amélioration des règles métiers ;
    - Intégration sur la version mobile Android ;
    - Concept et premiers prototypes sur le traitement des emballages alimentaires.
- R&D sur le traitement d'images
    - Mise en place de l'outil de reconnaissance de texte (OCR) ;
    - Réalisation d'une librairie iOS de prise de vue et de segmentation d'image ;
    - Mise en place d'un système de validation communautaire ;
    - Développement d'un programme de reconnaissance d'images ;
    - Développement d'un intranet de supervision de match produit.

Le fonctionnement selon la méthode SCRUM demande ensuite de détailler à chaque début de sprint les efforts[9] à fournir par personne concernant cette fonctionnalité. Chaque release est segmentée en sprints de 1 semaine.

[9] Unité de modélisation de l'effort à fournir dans la réalisation d'une tâche durant un sprint.

Actions	Acteurs	JANVIER				FEVRIER				MARS				AVRIL				MAI					JUIN				JUILLET				AOUT				SEPTEMBRE			OCTOBRE					NOV.		
		02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
V1 du Caddy Trophy																																													
Cadrage du projet	ML - RA																																												
Appropriation de l'environnement de développement	ML																																											
Recherche d'informations : Crowdsourcing	ML																																											
Définition du périmètre du Caddy Trophy	ML - Prod																																											
Implémentation des règles métiers (back-end)	ML - Back - Mobile																																											
Intégration du Caddy Trophy sur le site Internet	ML - Front																																											
Développement de l'outil de supervision sur l'intranet	ML - Back																																											
Pilotage de l'intégration sur iOS	ML - Mobile																																											
V1.2 du Caddy Trophy																																												
Amélioration des règles métiers (back-end)	ML - Back - Mobile																																											
Pilotage de l'intégration sur Android	ML - Mobile																																											
Recherche d'informations : Reconnaissance d'image et OCR	ML																																											
Premiers prototypes sur la reconnaissance de texte (OCR)	ML																																											
R&D sur le traitement d'images																																												
Mise en place de l'outil de reconnaissance de texte	ML																																												
Réalisation d'une librairie iOS de segmentation d'image	ML																																											
Mise en place du système de validation communautaire	ML																																											
Développement d'un programme de reconnaissance d'image	ML																																											
Développement d'un intranet de supervision de match produit	ML																																											
Tâches transverses																																												
Communication externe	ML - Prod																																											
Rédaction du mémoire	ML																																											

ML : Matthieu Lombard
Prod : Equipe produit
Front : Equipe front-end (PHP/HTML-CSS-JS)
Back : Equipe back-end (JAVA)
Mobile : Prestataires mobiles (iOS / Android)
RA : Rémy Amouroux

Figure 12 : Macro-planning

2.4.1 Rôles et responsabilités

Le planning présenté précédemment, est segmenté en trois temps principaux sur les neufs mois prévus pour le projet. Des points d'avancement réguliers ont été prévus environ tous les mois pour ajuster le macro-planning pour les mois suivants.

2.4.1.1 Initialisation du projet

Cette phase a permis de poser les bases du projet et de définir les premières idées autour du crowdsourcing. Ce travail a demandé de répertorier les différentes techniques de crowdsourcing ainsi qu'un cadrage précis du périmètre de la solution à mettre en place avec l'équipe produit.

Rôle	Responsabilité
Matthieu Lombard	Gestion du projet Définition du « crowdsourcing ».
Equipe produit	Définition du périmètre du projet.

2.4.1.2 Pilotage du développement du Caddy Trophy V1

Cette deuxième phase avait pour but de mettre en place le Caddy Trophy au sein des applications web et mobiles. Pendant cette période, les équipes de développement ont dû travailler en collaboration. L'équipe back-end a pris en charge le développement des services métiers. L'équipe front-end de son côté a intégré les fonctionnalités du site Internet pendant que les prestataires mobiles s'occupaient de l'intégration des mêmes fonctionnalités sur les deux applications mobiles.

Rôle	Responsabilité
Matthieu Lombard	Gestion du projet. Pilotage des prestataires en développement mobile. Développement front-end et back-end.
Equipe back-end	Développement des services métiers pour le traitement des actions utilisateurs web et mobile. Développement des services métiers pour le traitement des fonctionnalités des outils de supervision dans l'intranet.
Equipe front-end	Intégration de l'espace communautaire au site Internet. Intégration des outils de supervision sur l'intranet.
Equipe mobile	Intégration des actions et de l'espace communautaire dans les applications mobiles.

2.4.1.3 Traitement des images

Cette phase a été axée recherche et développement pour la mise en place d'un système de reconnaissance de texte et de reconnaissance d'emballages alimentaires produits. Cela a été un travail plus personnel au départ comprenant la recherche d'informations sur le traitement des images et la reconnaissance de texte. Le choix des outils a ensuite donné lieu à plusieurs prototypes.

Rôle	Responsabilité
Matthieu Lombard	Gestion du projet Recherche sur la reconnaissance d'image et de texte. Prototypage. Intégration back-end, front-end et mobile d'une solution.
Equipe back-end	Support dans la mise en place et la compréhension de l'architecture en place.

2.5 Analyse de l'existant

Afin d'intégrer au mieux de nouvelles fonctionnalités, nous avons étudié l'architecture actuelle du BCC au niveau physique, technique, ainsi que les flux entrants et sortants.

2.5.1 Architecture physique

L'architecture physique se compose de trois serveurs web frontaux permettant de supporter la charge occasionnée par le site Internet www.leboncotedeschoses.fr ainsi que le trafic venant des applications mobiles. Les serveurs web accèdent eux même aux bases de données présentes sur un autre serveur. Un intranet d'administration est aussi présent sur celui-ci afin de gérer des données non publiques ou des données brutes provenant des sites marchands et devant être traitées avant leur publication sur les services en ligne (Figure 13). L'architecture physique ne demande pas de modification.

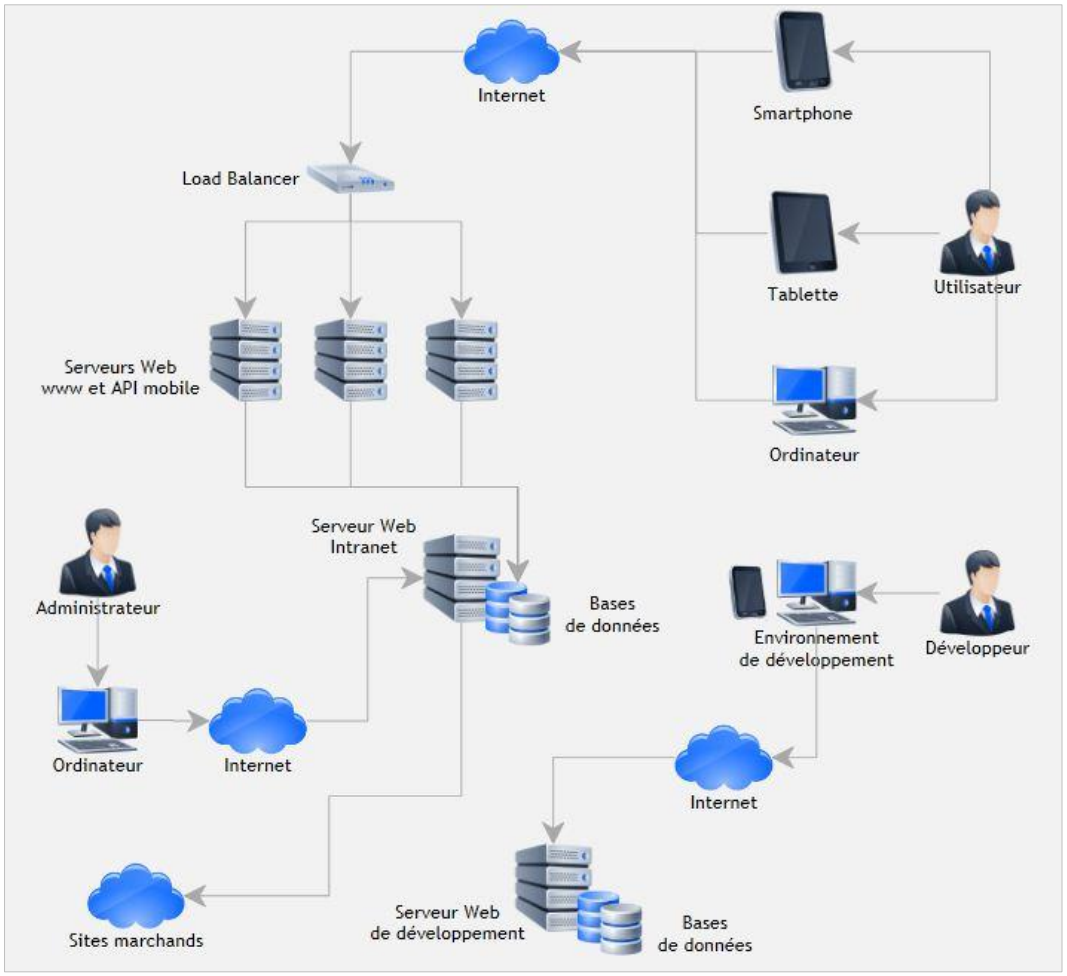


Figure 13 : Architecture physique au BCC

2.5.2 Architecture applicative

L'architecture de l'environnement technique du BCC est basée sur un modèle multi-tiers. Le système est divisé en 3 couches (Figure 14) :

- Couche de présentation (Client – navigateur)
- Couche de logique métier (Application)
- Couche d'accès aux données (Base de données)

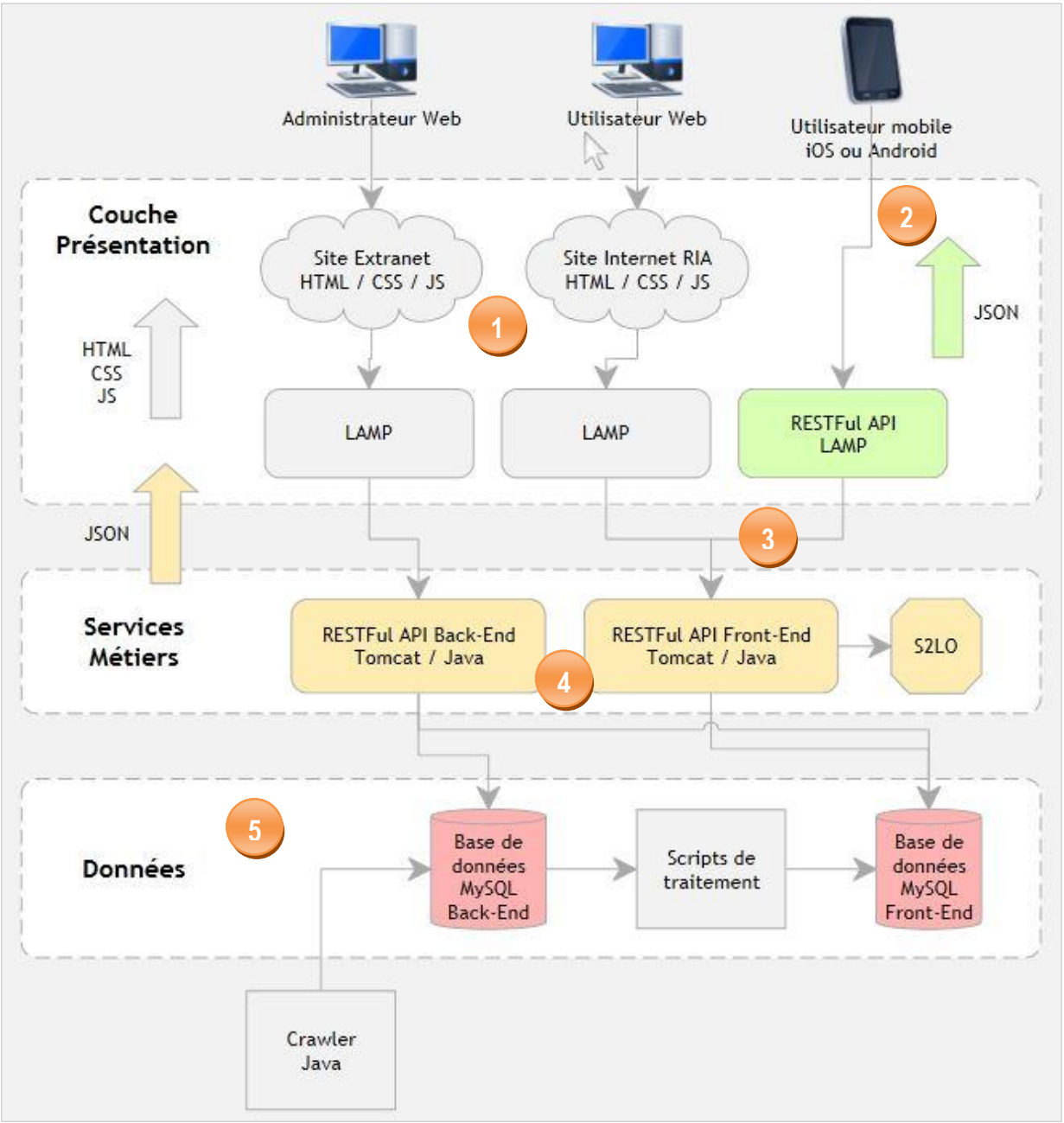


Figure 14 : Architecture applicative au BCC

On désignera dans le document la couche de présentation : « Front-End », et les couches métier et données : « Back-End », en rapport avec la segmentation en deux domaines d'expertise de l'équipe technique du BCC.

2.5.2.1 Front-End

La couche présentation correspond à l'affichage, la restitution sur le poste fixe ou mobile, le dialogue avec l'utilisateur.

1. Les parties site Internet et intranet sont structurées autour des langages clients, HTML, CSS, javascript et d'une plateforme serveur, LAMP. Toutes les actions effectuées sur les sites coté navigateur réalisent des appels vers la partie serveur qui se chargera d'appeler ensuite la couche métier pour les traitements fonctionnels et retourner au navigateur les données à afficher au format web. Le fait de réaliser les appels vers la couche métier côté serveur donne un premier niveau de sécurisation en masquant ces requêtes aux utilisateurs chevronnés et malintentionnés.

2. La partie mobile communique avec une première passerelle écrite en PHP qui implémente des services web de type RESTful mettant à disposition des développeurs mobile une API d'appel permettant d'accéder à des ressources ciblées et de récupérer des données destinées à l'affichage mobile. Ces requêtes HTTP suivent les opérations classiques : GET, PUT, POST et DELETE lancées sur une URL précise. Les retours de ces services sont faits au format JSON pour un affichage spécifique au mobile. Comme pour la partie web, les requêtes vers la couche métier sont effectuées depuis la plateforme serveur LAMP.

3. Le fait que les appels à la couche métier se fassent sur le même point d'entrée de la couche métier pour le site Internet et les applications mobiles permet la standardisation des transactions en une seule API métier. Les plateformes LAMP web et mobile s'occupent de transmettre à l'utilisateur les données génériques retournées par la couche métier et formatées spécifiquement pour l'outil de navigation : navigateur web ou smartphone.

2.5.2.2 Back-End

Le traitement métier des données correspond à la mise en œuvre de l'ensemble des règles de gestion et de la logique applicative.

4 La partie métier, frontale aux requêtes des serveurs LAMP web et mobile, implémente aussi des services web de type RESTful, écris eux en JAVA. Les retours des services sont renvoyés au format JSON.

Deux API sont accessibles, une permettant de récupérer des données métiers destinées au Front-End et à l'affichage aux utilisateurs et une seconde permettant l'affichage des données dans l'Intranet. C'est l'API destinée au Front-End qui permet d'appeler le moteur S2IO et de retourner les résultats de calcul pour la comparaison des listes de courses.

5 La couche d'accès aux données correspond aux données stockées servant à la vie des différentes applications. Deux bases de données sont accessibles, une pour les données front-end qui sont les données affichables publiquement et une seconde qui sont les données back-end non encore affichables publiquement. Cette dernière contient les informations provenant des robots (crawler) qui doivent être traitées par des scripts intermédiaires ou encore des données à traiter via l'intranet avant d'apparaitre en ligne.

2.5.2.3 Exemple d'une requête REST

Prenons comme exemple la récupération des informations d'un utilisateur depuis le web ou le mobile.

Envoie d'une requête REST en GET sur l'URL http://leboncotedeschoses.fr:9080/rest/user/get/33050 depuis la couche présentation vers la couche métier :

- La première partie de l'URL http://leboncotedeschoses.fr:9080/rest/ permet l'accès à l'API REST.
- La seconde partie de l'URL user/get/33050 détermine les actions à effectuer au niveau de l'API métier.

Ici, nous ciblons la récupération des informations d'un utilisateur suivant son identifiant. Nous obtenons en retour une réponse JSON correspondante avec laquelle seront créées indépendamment les structures de données remontées pour le site Internet et le mobile. (Figure 15)


```
{
        "s2loparams":{
                "drive":"on",
                "livraison":"on",
                "magasin":"on",
                "plus_vite_moins_cher":"moins_cher",
                "adr_dep":"Grenoble,France",
                "adr_dep_lat":"45.188529",
                "adr_dep_lon":"5.724523999999974",
                "adr_dep_ville":"Grenoble"
        },
        "email":"matthieu@webinbox.fr",
        "pseudo":"Matt",
        "prenom":"Matthieu",
        "date_ann":1985,
        "date_inscription":"2012-10-0215:40:36.0",
        "date_activation":"2012-10-0215:40:36.0",
        "taille":173,
        "poids":80,
        "preference":[4,8]
}
```


Figure 15 : Exemple de réponse JSON renvoyée par la couche métier

Les informations sur l'utilisateur sont donc contenues dans ce format de données. Dans cet exemple, nous retrouvons l'email, le prénom, différentes dates, des préférences et caractéristiques de la personne ainsi que des paramètres techniques servant à la configuration du moteur S2IO ; tout cela présent sous la forme « clé-valeur » caractéristique du format JSON.

2.5.3 Langages et outils utilisés

Le BCC met à disposition une architecture qui demande la connaissance de différents systèmes et technologies :

	Front-End		Back-End	OCR et Analyse d'images
	Mobile	Web		
Plateforme et environnement	iOS, Android	Linux (Ubuntu)	Linux (Ubuntu)	Linux (Ubuntu)
Serveur	Apache 2	Apache 2	Tomcat	-
Base de données	SQLite	-	MySQL	-
Langages	Objective-C, Java	PHP, HTML, CSS, JS	Java	C++, Java
Librairies	-	jQuery	Jersey (REST/JSON), Hibernate	Tesseract OCR, OpenCV

Tableau 2 : Langages et outils

Après avoir défini le déroulement, la base du projet qu'est le crowdsourcing, ainsi que l'environnement d'intégration au sein du système d'information du BCC nous allons détailler son implémentation. De la mise en place de l'environnement de travail à l'intégration de chaque brique du système de crowdsourcing.

3 Réalisation du projet

Après avoir défini les bases du projet, nous allons commencer son intégration technique au sein du système d'information du BCC. Ensuite nous détaillerons l'implémentation de chaque fonctionnalité du projet.

3.1 Mise en place de l'environnement de développement

3.1.1 Fonctionnalités et intégration

Les différentes fonctionnalités (Figure 16) à mettre en place viendront s'intégrer à l'architecture applicative (Figure 14) en place au BCC.

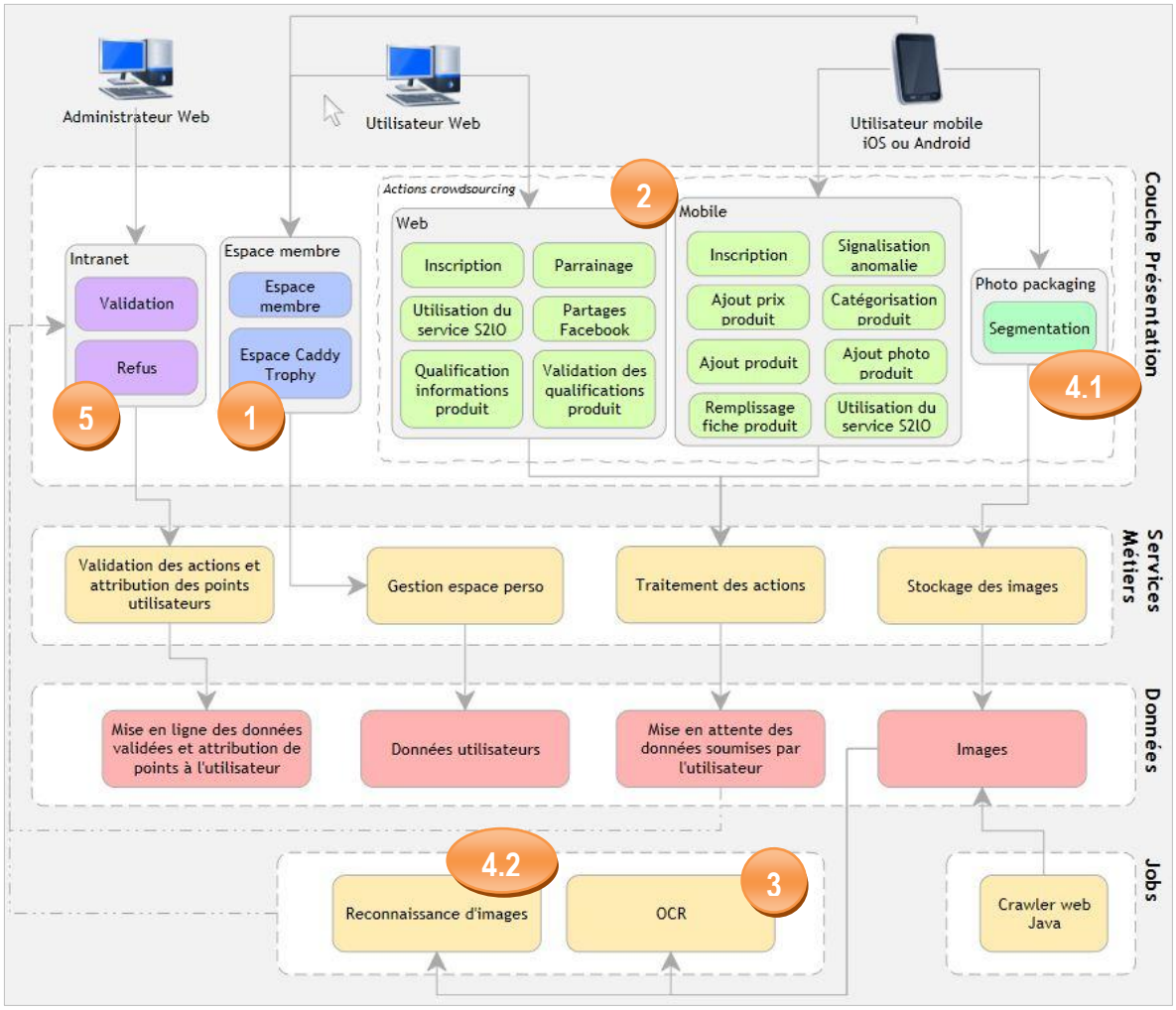


Figure 16 : Diagramme d'intégration des fonctionnalités

On distingue 5 modules qui recoupent transversalement les 3 couches et qui seront le fondement du projet de crowdsourcing :

1. L'espace Caddy Trophy dédié aux membres du BCC sur le site Internet et le mobile. Cette rubrique va permettre aux utilisateurs de configurer leur compte et d'accéder à un tableau de bord personnel du Caddy Trophy.

2. Les actions de crowdsourcing du web et du mobile. Le fondement du Caddy Trophy demande aux utilisateurs la réalisation de tâches réparties aux seins des applications du BCC. Ces actions qui valorisent la base de données du BCC vont aussi permettre le calcul des points de chaque utilisateur, en vu de maintenir un classement dans le jeu.

3. La reconnaissance de texte (OCR) sur les emballages produits. Cette fonctionnalité entraîne la création d'une brique applicative mobile qui permettra aux utilisateurs de prendre en photo un emballage produit et ensuite de la segmenter en fragments d'image à analyser. Une brique technique va ensuite permettre le traitement de ces fragments afin d'en extraire le texte. Ces traitements seront soumis à un processus de validation manuel par les administrateurs du BCC et les utilisateurs web des services.

4. La reconnaissance d'images sur les photos utilisateurs et images web des emballages produits. Cette brique technique va permettre le traitement des images provenant des crawlers JAVA et des images remontées par les utilisateurs des applications mobiles. Ces traitements seront soumis ensuite à un processus de validation manuel par les administrateurs du BCC.

5. L'intranet de validation/refus. Cet espace réservé aux administrateurs du BCC permettra de confirmer ou de refuser les propositions faites par les utilisateurs dans le cadre du Caddy Trophy. Les propositions validées seront ensuite diffusées publiquement sur les différentes plateformes du service et des points seront attribués au participant. Les propositions issues du programme de reconnaissance d'images seront aussi soumises à validation au sein de cet espace. Des systèmes de validation communautaire telle que la validation des textes issus du programme d'OCR seront inclus au sein de l'espace Caddy Trophy des membres.

Après la vision de l'intégration dans l'architecture du SI, voyons cela en termes de structuration de code.

3.1.2 Création des projets « Git »

Le code du BCC est organisé autour de plusieurs dépôts « Git ». Git est un logiciel de gestion de versions décentralisé qui permet le travail d'une équipe sur un projet autour d'un même code. Chaque dépôt Git représente une fonctionnalité ou un aspect métier de l'application permettant de segmenter pour un travail et des mises à jour simplifiées.

Actuellement, pour les projets qui nous intéressent, voici le diagramme des composants Git mis en place ainsi que leurs dépendances fortes au niveau du code, ou faibles au travers de requêtes HTTP (Figure 17).

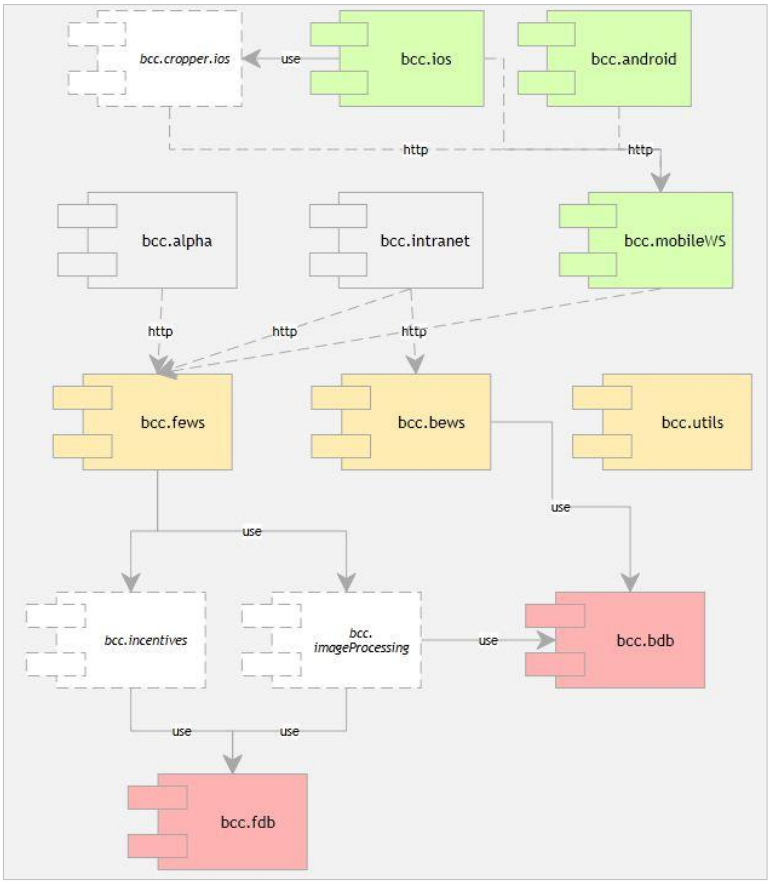


Figure 17 : Diagramme des composants Git

Trois dépôts Git contiennent les projets web formant la couche de présentation :

- **« bcc.alpha » :** code du site Internet, écrit en PHP, HTML, CSS et Javascript ;
- **« bcc.intranet » :** code de l'intranet d'administration, écrit en PHP, HTML, CSS et Javascript ;
- **« bcc.mobileWS » :** code des webservices REST présentés aux applications mobiles iOS et Andoid, écrit en PHP.

Deux dépôts Git représentent les interfaces des webservices REST de la couche métier :

- **« bcc.fews » :** code de l'API REST métier pour le site internet, les applications mobiles et quelques fonctionnalités de l'intranet, écrit en JAVA ;
- **« bcc.bews » :** code de l'API REST pour l'intranet, écrit en JAVA.

Deux dépôts Git représentent les objets métiers mappés en base de données :

- **« bcc.fdb » :** objets métiers pour le front-end, écrit en JAVA ;
- **« bcc.bdb » :** objets métiers pour le back-end, écrit en JAVA.

Le dépôt **« bcc.utils »** propose des utilitaires statiques tels que l'accès à la base de données, la configuration, etc.

Pour notre projet, nous avons créé de nouveaux dépôts Git (en blanc et bordures pointillées sur le diagramme Figure 17) :

- **« bcc.incentives »** : intégrera le code permettant de traiter les manipulations métiers pour la gestion des actions utilisateurs, la manipulation des points, etc., écrit en JAVA ;
- **« bcc.imageProcessing » :** code permettant de gérer l'OCR et la reconnaissance des images d'emballages alimentaires (fichiers et base de données), écrit en JAVA ;
- **« bcc.cropper.ios » :** librairie iOS permettant la segmentation des images au sein de l'application iOS du BCC, écrit en Objective-C.

Des modifications sur les autres projets Git seront bien entendu à prévoir.

Une fois le découpage des fonctionnalités réalisé au travers de différents projets techniques, nous pouvons entamer la réalisation des premières tâches.

3.2 Refonte de l'espace membres du site Internet

Afin de me familiariser avec l'environnement technique et la gestion de projet au sein de l'équipe, les premières tâches ont été l'immersion dans le code du site Internet en vue de la refonte graphique partielle de celui-ci. Plusieurs modules du site Internet ont été revus tant au niveau ergonomique que données à afficher. Cela m'a permis de commencer à travailler sur de l'intégration HTML, CSS, javascript au niveau du projet Git « bcc.alpha » ainsi que sur la modification de web services JAVA de « bcc.fews ». Un des modules importants de cette refonte graphique qui permettra d'accueillir quelques briques du « Caddy Trophy » est l'espace « membre ».

L'espace membre pour les utilisateurs est l'une des briques principales du BCC. Il permet aux utilisateurs de la solution de s'inscrire sur le site Internet ou le mobile et d'avoir accès à des fonctionnalités supplémentaires et de participer au « Caddy Trophy ».

Une fois l'utilisateur connecté, il va avoir accès à la gestion de son foyer, ses informations personnelles et ses préférences alimentaires. Une fois la mise en place des actions pour le « Caddy Trophy » effectuée, cette espace contiendra aussi un tableau de bord des récompenses du concours.

Figure 18 : Espace membre - Fiche membre

Les systèmes d'inscription et de connexion étaient déjà présents dans les premières versions du site et du mobile. Pour le site Internet, il a été décidé de rajouter plusieurs fonctionnalités à l'espace membres lui permettant de configurer plusieurs paramètres qui seront ensuite utiles pour ajouter de la pertinence dans les recherches du moteur de comparaison de listes de course : S2IO. (Figure 18)

Après avoir refondu l'espace personnel des utilisateurs, nous avons commencé la modélisation de la couche d'« incentive » qui viendra se câbler dessus et apporter un esprit de compétition fédérateur au sein de la communauté.

3.3 Mise en place du système d'« incentive »

La mise en place de ce système d'« actions-récompenses » va se greffer dans l'architecture du BCC et plus particulièrement au niveau de sa base de données, son API REST et ses interfaces web et mobile.

3.3.1 Prise en compte des actions utilisateurs

3.3.1.1 Modélisation du modèle de données

Afin de se prémunir de dépendances fortes au modèle de données en place, nous sommes partis sur une modélisation autour de l'entité « **UserInfo** ». Cette entité représente un utilisateur du service. Le reste de la conception est donc effectué autour de l'utilisateur.

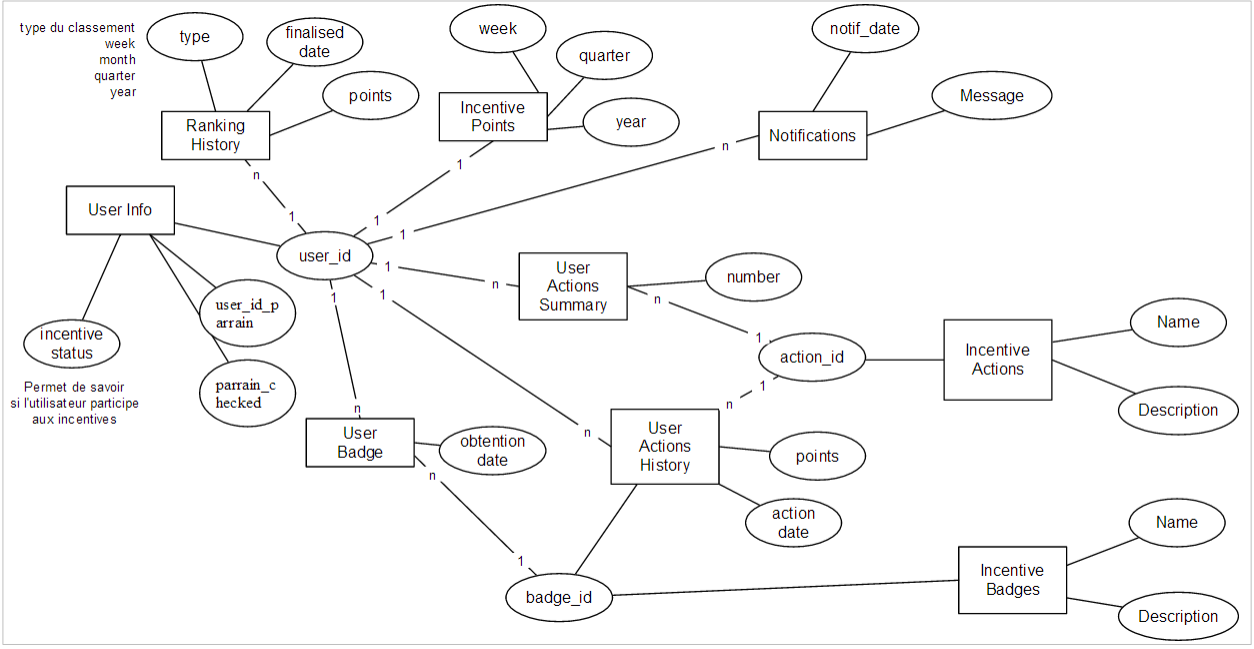


Figure 19: Modèle de données du programme d'incentive

A cette entité utilisateur sont liées plusieurs tables :

- **IncentivePoints :** référence les points en cours d'acquisition pour la semaine, le trimestre et l'année ;
- **UserActionsHistory :** recense l'historique unitaire de toutes les actions effectuées par l'utilisateur depuis le lancement du programme ;
- **UserActionsSummary :** contient une extraction de la table UserActionsHistory pour la semaine passée afin de générer le classement de la semaine ;
- **IncentiveActions :** contient la liste des actions possibles ;
- **UserBadge :** contient la liste des badges gagnés par l'utilisateur ;
- **IncentiveBadges :** contient la liste des badges existants ;
- **Notifications :** liste les notifications envoyées à l'utilisateur (gains de X points, gain d'un badge, etc.) ;
- **RankingHistory :** recense l'historique des classements par semaine, mois, trimestre et année depuis le lancement du programme.

3.3.1.2 Modélisation des webservices métiers

Les services de type REST vont être implémentés au niveau du projet « bcc.fews » et utiliseront les entités métiers définies dans le modèle de données et générées en JAVA au sein du projet « bcc.fdb ». Ils vont permettre de récupérer des données particulières en fonction des demandes front-end.

Plusieurs fonctionnalités ont été modélisées dans l'API REST grâce à la librairie « jersey ». L'API est accessible en http via une URL de type « http://leboncotedeschoses.fr:9080/rest ». Jersey fournit ensuite un système d'annotations permettant de taper sur des méthodes précises de classes JAVA en spécifiant des directives à la fin de cette URL. Nous avons donc créé une **IncentiveWebService.java** qui sera le point d'entrée de toutes les requêtes liées à l'incentive. Elle est définit de cette manière :

```
@Path("/incentive")
public class IncentiveWebService
```

Et donc accessible via http://leboncotedeschoses.fr:9080/rest/**incentive**.

Pour l'accès aux méthodes de la classe, le même principe est proposé, ici pour la récupération des points pour un utilisateur, on déclare la méthode **userGetPoints** :

```
@GET
@Path("/get/points/{user_id}")
public Response userGetPoints(@PathParam("user_id") int user_id)
```

Accessible via http://leboncotedeschoses.fr:9080/rest/incentive**/get/points/5** où 5 est une donnée dynamique de l'URL représentant l'identifiant de l'utilisateur. Il est récupéré en paramètre de la fonction pour ensuite effectuer les traitements et requêtes nécessaires à la récupération de ses points.

La réponse renvoyée formatée au format JSON propose les points de la semaine ainsi que les points du trimestre, de l'année et totaux (ici, moins d'une année a été effectuée donc les données du trimestre, de l'année et totaux sont identiques :

```
{
    "week": 250,
    "quarter": 29615,
    "year": 29615,
    "total": 29615
}
```

La documentation de l'API d'interrogation détaillée a été rédigée et est proposée en Annexe 2.

L'API créée est là pour permettre la récupération de données liées aux incentives et non pour l'insertion ou la mise à jour de ces points. Effectivement, les différentes actions utilisateurs prévues récompensées dans le programme (inscription, ajout de produits, etc.) sont déjà traitées côté back-end par différents webservices. Pour traiter l'attribution de points et des badges, il conviendra donc de créer des « hook[10] » intelligents et évolutifs dans les différentes méthodes de traitements de ces fonctionnalités. Chaque hook correspondra à la vérification de l'éligibilité de l'utilisateur pour une action ou un badge précis :

```
new ActionInscription().verifAndApply(idUser);
```

[10] Un hook dans un programme informatique permet d'insérer des fonctionnalités supplémentaires à des moments déterminés pour personnaliser le fonctionnement de ce dernier.

Suivant l'éligibilité, des règles métiers spécifiques à chaque action sont effectuées. Chaque action hérite d'une classe abstraite **AbstractActionBadge** permettant de définir les méthodes à surcharger obligatoirement par les actions et les badges. Elle implémente aussi des méthodes génériques tel que l'envoi de notification ou l'attribution d'un badge (Figure 20 : Diagramme de classe des actions et badges).

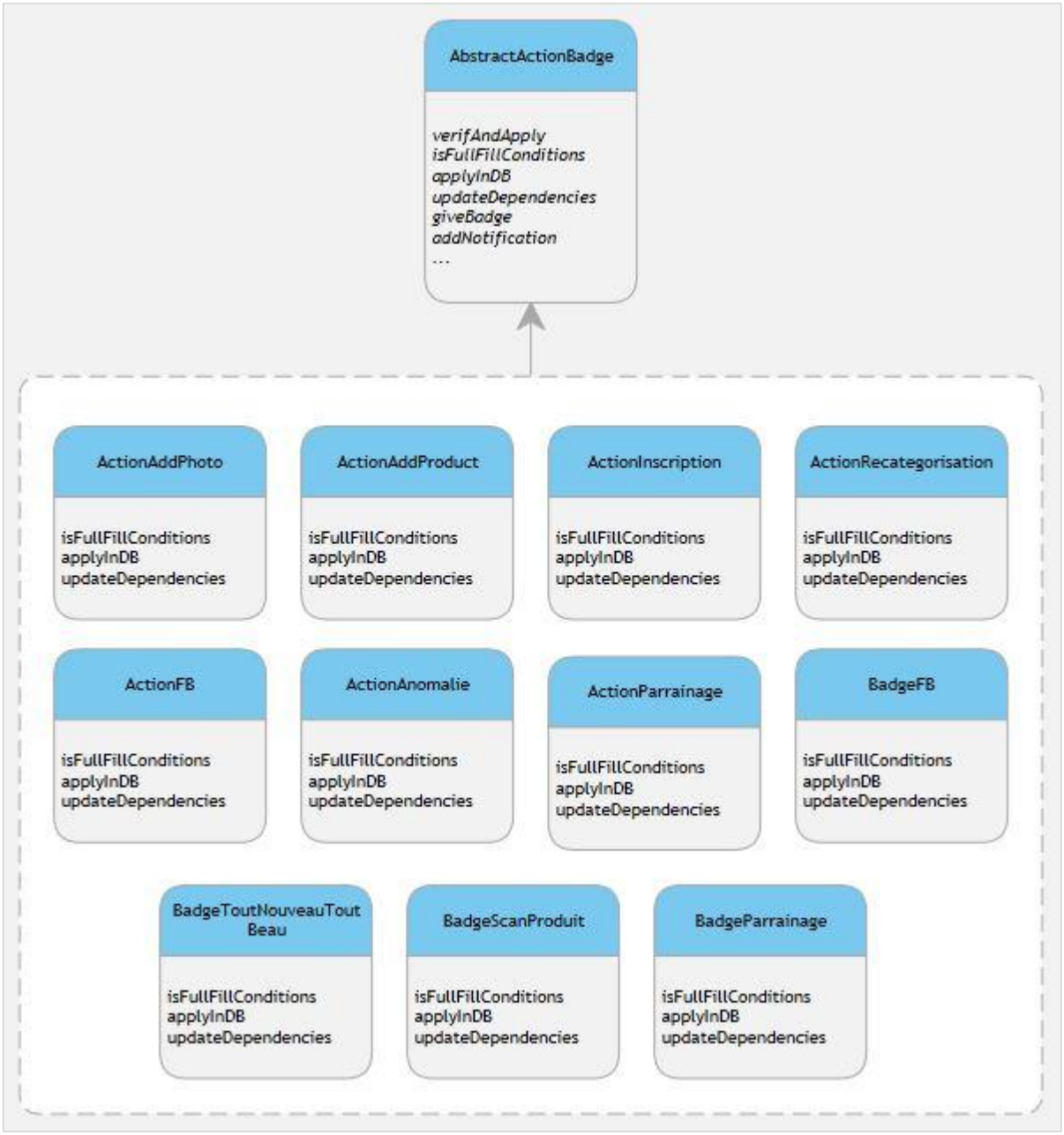


Figure 20 : Diagramme de classe des actions et badges

3.3.1.3 Implémentation pour le web

En ce qui concerne le site Internet, il ne demande pas de modification pour accéder aux nouveaux webservices. L'application des points et l'attribution des badges se feront grâce aux « hook » côté back-end pour chaque action web réalisée.

3.3.1.4 Implémentation pour le mobile

Pour les applications mobiles, à l'image du site Internet, elles ne nécessitent pas de modifications structurelles en ce qui concerne le traitement des points par action réalisée. En revanche, le mobile interroge une API REST passerelle en PHP qui sert d'interface de transformation entre les données renvoyées par la couche métier et les données à renvoyer pour l'affichage sur le mobile. L'application mobile ne fonctionnant pas sur un système de page comme le site Internet, nous avons donc créé un unique webservice REST permettant de retourner toutes les données d'incentive structurées dans un objet global. Ce webservice mobile appelle les différents webservices métiers pour construire son objet et le renvoyer pour affichage dans l'application.

L'API est accessible en http via une URL de type « http://m.leboncotedeschoses.fr:9080/rest ». Un fichier **.htaccess**[11] permet la redirection vers le bon service suivant l'appel réalisé dans l'URL. Ici, une règle permet de rediriger vers le bon fichier PHP **incentivebcc.php** :

```
RewriteRule ^incentive/get$ incentivebcc.php [L]
```

Accessible via http://m.leboncotedeschoses.fr:9080/rest**/incentive/get**. Le script appelle ensuite tour à tour les webservices métiers pour les agréger dans un objet JSON compréhensible par l'application mobile.

La documentation de cette partie de l'API mobile ainsi que l'objet JSON retourné sont proposés en Annexe 3.

[11] Fichier de configuration des serveurs Apache qui peuvent être positionnés dans des répertoires de site web et qui servent à gérer les droits d'accès, créer des règles de redirection, etc.

3.3.2 Tableau de bord des utilisateurs

En parallèle de la mécanique de gestion des actions, il faut proposer à l'utilisateur une interface chaleureuse et ergonomique donnant de l'attrait au Caddy Trophy. Cette interface est une sorte de tableau de bord récapitulatif de toutes ses actions dans la compétition : le nombre de points gagnés, les classements, les badges récoltés, etc.

3.3.2.1 Intégration web

Pour accompagner les utilisateurs lors de leur participation au Caddy Trophy, nous avons créé deux pages servant de tableau de bord.

En relation avec le pole produit, nous avons créé un environnement très visuel mettant en avant le score personnel de la semaine et promulguant de belles icônes ainsi que des badges décalés issues du monde du jeu (coupe, sablier).

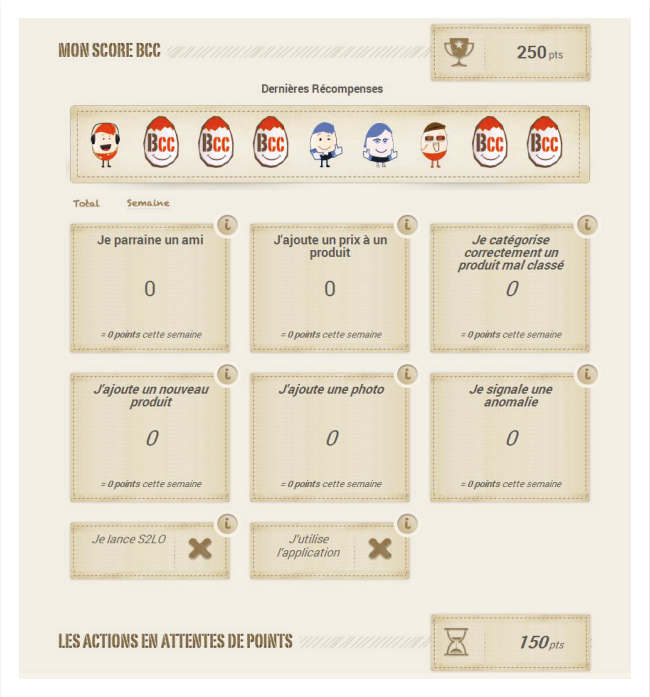


Figure 21 : Espace membre Web - Scores Caddy Trophy

Chaque action est représentée par un bloc donnant le nombre de fois que l'action a été effectuée ainsi que le nombre de points que cela a rapporté. Les actions en attente de points sont là pour permettre à l'utilisateur de voir ce qui n'a pas encore été validé par les administrateurs du BCC (Figure 21).

La seconde page affiche le score total du foyer (cumul des scores de chaque utilisateur du foyer) ainsi que le classement des foyers, total ou par semaine (Figure 22).

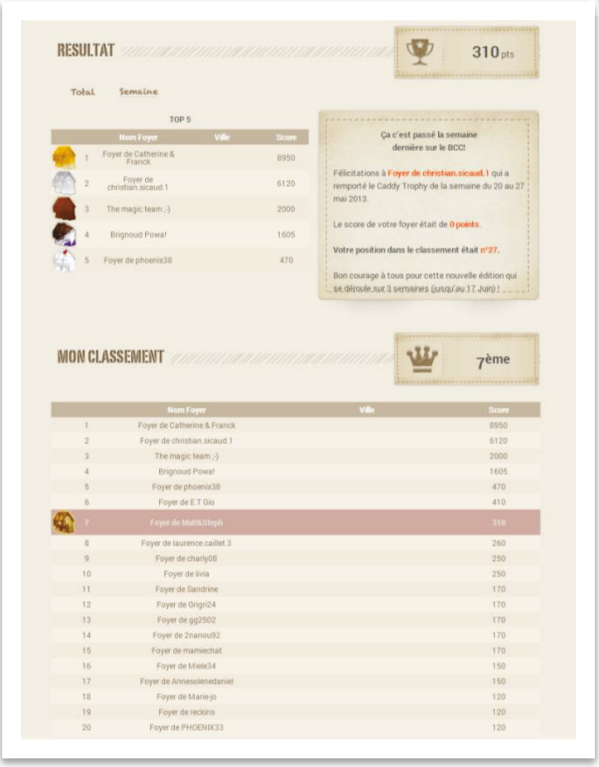


Figure 22 : Espace membre Web - Classement Caddy Trophy

3.3.2.2 Intégration mobile

Comme la plupart des actions sont présentes sur mobile, il a aussi fallu intégrer un espace Caddy Trophy aux applications mobiles. En relation avec les prestataires pour les développements iOS et Android, nous avons réalisé les pages de score, de badge et de classement qui permettent un affichage d'un tableau de bord de toutes les informations présentes sur le web : scores, classements, statistiques, notifications, badges ; mais ne permet aucune intervention de l'utilisateur (Figure 23).

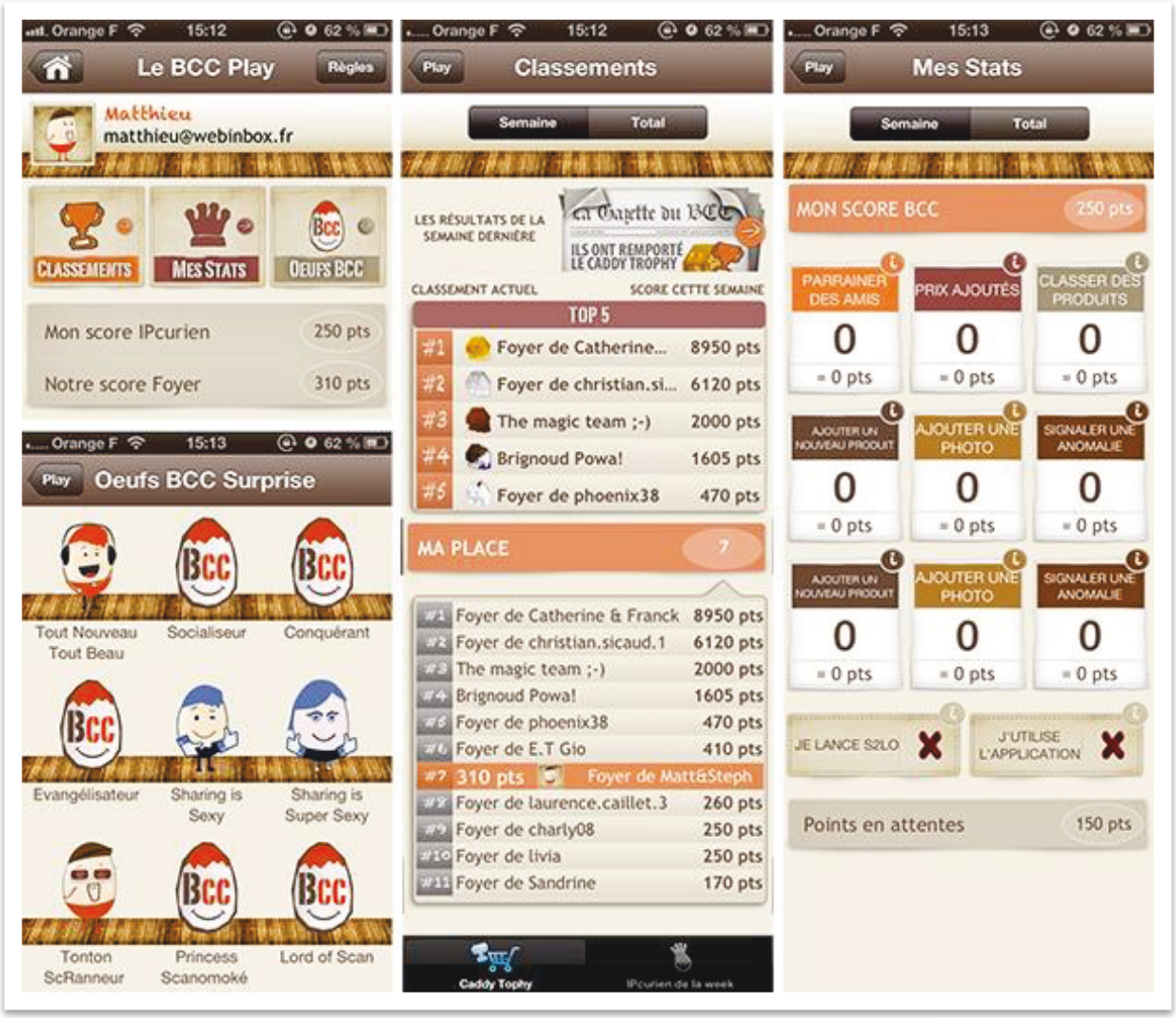


Figure 23 : Espace membre mobile iOS - Caddy Trophy

3.4 Récupération d'informations via les images d'emballages alimentaires

Afin d'apporter encore plus d'informations sur nos produits en base de données, nous avons réfléchi à des méthodes permettant d'exploiter les images d'emballages produits en notre possession. Deux solutions ont été pensées :

- Exploiter le crowdsourcing en mandatant les utilisateurs pour prendre des photos d'emballages de produits. Chaque utilisateur via son propre smartphone doit pouvoir compléter la fiche d'un produit en prenant une photo de certaines parties de l'emballage. En l'occurrence, deux parties pour nous sont importantes dans une première version de l'application : la **description** et les **ingrédients** du produit. Charge à nous d'extraire ces contenus textuels des images. Ce processus sera inclus en tant qu'action au sein du projet « bcc.incentives » et chaque utilisateur se verra attribuer des points comme pour les autres actions ;
- Utiliser la reconnaissance d'images sur les visuels des emballages produits provenant du crowdsourcing et des « webcrawlers » afin d'attribuer automatiquement des données manquantes à certains produits.

Ces mécanismes seront inclus au sein du projet « bcc.imageProcessing » comme des programmes de traitement autonomes et seront lancés automatiquement de manière régulière.

3.4.1 Extraction de texte

L'objectif est d'arriver à lire convenablement des portions de texte présentes sur une image issue de la prise d'une photo via un smartphone.

La première complexité qui se présente à nous vient de la diversité importante des formes, couleurs et mise en page des emballages (Figure 24). Effectivement, chaque emballage a ses propres couleurs et tailles de police d'écriture, des couleurs de fond différentes, les informations voulues sur des faces opposées qui font que nous ne connaissons pas sa structure au préalable pour cibler des régions précises à extraire. Si l'on applique un logiciel de reconnaissance

Figure 24 : Photo emballage alimentaire

de caractère (OCR) directement sur l'image globale avec un panel de styles de texte aussi variés, le texte reconnu en sortie sera très approximatif. Il faut donc préalablement mettre en place une mécanique de segmentation et de traitement de l'image.

Pour confirmer cela, nous avons exécuté un logiciel OCR **tesseract-ocr** [8] en ligne de commande sur un emballage pris dans sa globalité (Figure 24) :

```
tesseract image.jpg fichier_text -l fra
```

Sans traitement préalable de l'image ni suppression des éléments inutiles (couleurs de fond, images, etc.), nous pouvons voir que sans configuration précise de l'OCR l'exécution renverra un texte déstructuré en sortie :

```
mflnnlnilun’ ‘U,’
5 æryzçfiïgfigäzfäuäfiflmrazm
-‘-*...._-M,.m.....‘m.. Muflwwhm
’ zz."._=_,._.‘__,à_‘,_r‘..‘...‘.nu.‘-".............._
ääjgîrmmmmmwzzzmmzzzu:
MW‘ '
gâmfi‘ t;  ,;f
‘LHÆË’? ‘f f ‘f:
1a...‘ ‘.‘......;._‘;:..._  ‘.
,  » Q 'Î":';Z..,."L..ÄÄM..ÏL-
Ï. lgfiu.‘ n n-nuun
 -".’-‘ ’="“..‘-‘,:;,:::..._.‘..‘::‘

i-nwuflnwm" 7*"‘"‘.""' "'î"_’u"""‘m"«‘ä"wfifä.‘l"lliïhffä‘flïi*w"'
(IŒSIMIIIPIIIÇIIIHÜI
........,..‘..._.W
11:»
51%
```

Nous nous sommes donc posé la question d'effectuer une reconnaissance des zones de texte au sein des emballages en les identifiant grâce à leur fond de couleur ou la forme de leur bloc contenant. Nous avons soumis l'idée auprès de plusieurs communautés développant ces librairies de traitement d'images. Les fondateurs de la librairie OpenIMAJ [9] de l'université de Southampton au Royaume-Uni nous ont permis d'avancer dans notre réflexion. Un emballage alimentaire peut contenir beaucoup de texte à des tailles et couleurs très variées dans une même prise de vue. Certains textes peuvent même contenir plusieurs couleurs sur plusieurs fonds différents. D'après leur expérience, le traitement d'image se base

principalement sur l'intensité et la couleur des pixels que l'on manipule au sein d'une image. Si la photo contient trop d'informations et qu'au sein de celle-ci plusieurs zones de texte sont présentes et qu'elles apparaissent en police très réduite (Figure 24), il sera très complexe de créer un algorithme permettant d'éliminer les éléments inutiles et de trier sémantiquement par la suite les zones de texte que l'on souhaite récupérer réellement.
De plus, un autre élément que nous ne maîtrisons pas est la qualité des photos prises par les utilisateurs. Chaque version de smartphone propose des caractéristiques d'appareil photo très différentes et surtout, chaque utilisateur prendra des photos plus ou moins exploitables : surexposition, flou, cadrage, etc. Nous pourrions faire le parallèle avec la reconnaissance de ticket de caisse pour laquelle nous aurions la possibilité d'aider au cadrage et à la prise de vue car le ticket de caisse a une forme standard rectangulaire qui permet sa détection plus aisément. Il faut donc trouver un moyen de qualifier les photos en aval de la prise de vue plutôt que d'essayer de détecter des patterns trop nombreux et complexes.

Dans le temps imparti et avec les connaissances de base en analyse d'image que nous avons actuellement dans l'équipe, il parait trop coûteux de développer une solution permettant d'absorber tous les modèles d'emballages alimentaires existants. Il semble donc que la diversité des éléments à reconnaitre va nous pousser à accompagner l'utilisateur assez loin dans sa prise de vue pour l'aider à cibler en amont les informations qui nous intéressent. Le ciblage devra nous permettre de ne récupérer qu'une zone de texte le plus uniforme possible en termes de style mais aussi en termes de contexte fonctionnel : description ou ingrédients.

Le dernier élément qu'il va falloir résoudre est la validation de l'authenticité de la photo par rapport au produit pour lequel elle est soumise. Actuellement, en ce qui concerne la photo principale du produit qui peut aussi être proposée par les utilisateurs des applications mobiles, celle-ci est validée dans l'intranet par l'administrateur des actions afin de ne pas mettre en ligne publiquement des visuels erronés. Le but de la mise en place de cette nouvelle action de crowdsourcing est aussi d'essayer de la rendre auto-suffisante sans aucune intervention de modération quelconque demandant des ressources en temps de plus en plus importantes. Il faut donc penser à la mise en place d'un système de validation communautaire en bout de chaîne.

Après cette première analyse, il ressort quatre points principaux à traiter :

- Segmentation de l'image, aide à la prise de vue côté utilisateur ;

- Traitement d'image, nettoyage du texte ;
- Reconnaissance de texte : OCR ;
- Système de validation communautaire.

3.4.1.1 Choix des outils

Nous avons réalisé un tour d'horizon des technologies existantes afin de choisir des outils intégrables aisément au sein de l'architecture BCC et proposant le plus possible une abstraction d'algorithmes complexes au profit d'une API accessible aux développeurs.

3.4.1.1.1 Segmentation d'image

Afin de cibler les portions d'image à extraire, il faut penser à un outil intégrable directement lors de la prise de vue par l'utilisateur au sein des applications mobiles du BCC. Comme ces applications sont réalisées en langages natifs nous développerons une première librairie pour iOS en Objective-C.

3.4.1.1.2 Reconnaissance de texte

Le procédé des systèmes OCR fournit la capacité de transformer les images comprenant des caractères imprimés à la machine en caractères lisibles par la machine. Il est évident que cette technique apporte un gain important en termes de productivité et de coût face à la réalisation de saisie manuelle par des opérateurs.

En termes d'outils, il convient de trouver un bon OCR open source. Un moteur sort du lot :

Tesseract [8]. Il a été développé par HP de 1985 à 1995 puis repris par Google. Il peut lire de nombreux formats d'images et les convertir dans 60 langues différentes. Il est utilisé dans de nombreux projets open source ainsi que dans le moteur de numérisation de livres numériques initié par Google. Nous avons tout de même testé le meilleur concurrent sur le marché des solutions propriétaires qui propose un OCR sous la forme d'un SDK accessible en mode web API : ABBYY Cloud OCR SDK [10]. Il s'avère très performant en termes de précision pour des documents structurés. Pour le type de documents que nous souhaitons analyser, comme Tesseract, il va demander beaucoup de configurations et probablement de post-traitements. De plus la facturation de cet

outil est effectuée à la requête, le budget risquerait d'être trop élevé pour des données qu'il sera malgré tout nécessaire de retraiter par la suite.

Tesseract fonctionne selon un traditionnel traitement « pipeline » étape par étape (Figure 25). La première étape consiste à réaliser une analyse de la présentation du document afin de définir des régions de texte. La seconde étape vise à repérer les composants connectés et à les stocker. Les contours sont ensuite réunis par emboitement dans des « blobs » (regroupement en boîte). Ces blobs sont organisés en lignes de texte qui sont analysés pour définir la chasse[12] des caractères. Les lignes de texte sont coupées en différents mots selon l'espacement des caractères. Pour les polices à chasse fixe, le caractère est directement découpé en cellule. Pour les polices à chasse variable, le texte est divisé en mots en utilisant les espaces fixes et les espaces flous. La reconnaissance procède ensuite en deux passages. Dans le premier passage, une tentative est faite pour reconnaître chaque mots un à un. Chaque mot considéré comme satisfaisant est conservé comme données d'apprentissage afin d'optimiser ses chances de précision dans la reconnaissance plus bas dans la page. Une deuxième passe est ensuite réalisée pour profiter de l'apprentissage réalisée trop tard en bas de page dans la passe précédente.

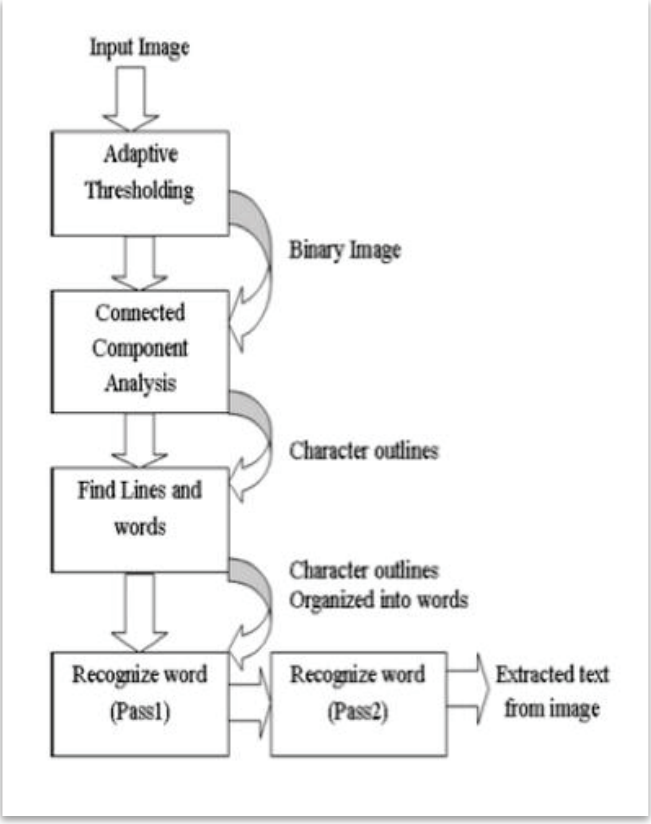


Figure 25 : Fonctionnement du moteur Tesseract [11]

[12] La chasse est, en typographie, la largeur d'un caractère augmentée des espaces entre deux de ses lettres consécutives.

Afin d'améliorer la reconnaissance de caractère il existe plusieurs méthodes, il faut bien évidemment le moins de « bruit », le moins de symboles sur la page, qui ne sont pas des caractères. C'est pour cela que pour des textes complexes à traiter, comme pour notre cas, des textes provenant d'images non uniformément formatées, il faudra appliquer un prétraitement afin de supprimer les éléments inutiles.

De plus, Tesseract propose deux moyens supplémentaires, l'apprentissage de nouveaux langages ou plutôt de nouvelles polices et tailles d'écriture et l'amélioration des dictionnaires de données utilisés par le logiciel.
Tesseract traitera correctement certaines polices d'écriture assez communes telles que les polices Roman (Times New Roman) ou Sans-serif (Arial, Helvetica) mais demandera un apprentissage des polices par taille pour d'autres plus complexes.

3.4.1.2 Installation de Tesseract OCR

Tesseract sera installé sur une machine serveur du BCC sous l'environnement Linux Ubuntu. Nous pouvons donc installer une version du logiciel suffisante grâce aux commandes :

Installation de l'outil :

```
apt-get install tesseract-ocr
```

Cette installation apporte aussi directement les dépendances avec la librairie graphique Leptonica [12] et le dictionnaire anglais par défaut. Pour avoir le dictionnaire français proposé, il faut installer un autre paquet :

```
apt-get install tesseract-ocr-fra
```

Afin de pouvoir développer une librairie C++ permettant de lancer la reconnaissance de texte et tous les traitements nécessaires de manière plus industrialisée qu'en ligne de commande, un paquet proposant une API est disponible pour le développement :

```
apt-get install libleptonica-dev
apt-get install libtesseract-dev
```

A présent l'outil est disponible, il faut le configurer.

3.4.1.3 Apprentissage de Tesseract OCR

L'apprentissage [13] consiste à fournir au moteur une ou plusieurs images contenant la liste complète de tous les caractères qui seront utilisés dans les futures images à numériser, écrits dans la police voulue (Figure 26).

Figure 26 : Image d'entrainement pour Tesseract OCR (fra.myfontitalic.exp0.tif)

Pour la suite, Tesseract a besoin d'un fichier « box » pour chaque image d'entrainement. Le fichier « box » est un fichier texte qui liste les caractères de l'image d'entrainement, une par ligne, avec les coordonnées de la boîte englobante du caractère dans l'image. Pour générer ce fichier, il faut lancer une commande spécifique sur notre image nommée [lang].[fontname].exp[num].tif :

```
tesseract fra.myfontitalic.exp0.tif fra. myfontitalic.exp0 batch.nochop makebox
```

Un fichier est généré : fra.myfontitalic.exp0.box

Lorsque l'on édite le fichier on peut voir nos caractères listés les uns en dessous des autres avec leurs coordonnées associées :

A 29 127 50 160 0
B 63 127 88 159 0
C 99 127 119 159 0
D 135 128 158 161 0
E 171 128 194 159 0
F 222 128 244 161 0
S 257 127 280 159 0
H 294 127 316 160 0
…

Avec ce format de fichier, il est complexe de voir si l'OCR s'est trompé dans son analyse de notre image. Il existe pour cela plusieurs outils mis en place par la communauté qui permettent de manipuler graphiquement ces résultats.

Nous avons utilisé un de ces outils en ligne et voilà le résultat [14] :

Figure 27 : Création d'un fichier "box" pour Tesseract OCR

Grâce à cet outil, nous pouvons déjà voir que chaque caractère a bien été délimité grâce aux boîtes englobantes oranges. Nous pouvons ensuite cliquer sur cette boîte et voir le caractère trouvé associé et le modifier le cas échéant si la correspondance est mauvaise. Ici par exemple le caractère « G » a été relié à la lettre « S ». Il est aussi possible d'ajouter, de redimensionner ou de supprimer des boîtes afin de modéliser un fichier « box » au plus proche de l'image.

Ensuite avec notre fichier image et son fichier « box », il faut lancer l'apprentissage du moteur :

```
tesseract fra.myfontitalic.exp0.tif fra.myfontitalic.exp0 box.train
```

Un fichier est généré : myfontitalic.tr

Dans notre cas particulier, nous n'avons pas de visibilité sur le nombre et le type de polices d'écriture que nous rencontrerons au travers de toutes les photos que feront les utilisateurs. C'est pour cela que nous allons réaliser une première version de l'application avec la connaissance par défaut incluse dans l'OCR. En capitalisant sur les nombreuses remontées des utilisateurs, nous pourrons dans une future version apporter un système d'apprentissage automatique. Nous aurons sous la main un panel d'échantillons important qui nous donnera une vision plus précise de la diversité des éléments à traiter et ainsi développer une solution plus robuste.

Cependant, nous allons enrichir fortement le dictionnaire français fourni par défaut dans l'OCR avec des termes liés au domaine fonctionnel du BCC présent dans notre base de données.

3.4.1.4 Elargissement du dictionnaire

Les dictionnaires sont facultatifs et Tesseract peut reconnaitre les lettres une à une sans cela. Ils aident simplement l'OCR en lui fournissant des probabilités de combinaisons de caractères possibles pour arriver plus précisément à des mots. Tesseract fournit un dictionnaire dans la langue française qui contient déjà beaucoup de mots. Cependant certains mots très spécifiques comme : acidifiant, curcumine, disulfite, etc. qui peuvent se retrouver dans les ingrédients d'un produit ne sont pas des mots qui alourdissent ce dictionnaire par défaut. Le BCC possède un très grand nombre de données textuelles sur les produits alimentaires et DPH qui vont permettre d'enrichir ce dictionnaire classique.

Pour avoir le plus grand nombre de mots nouveaux et spécifiques au métier, nous avons ciblé deux sources principales de données du SI :

- la base de données contenant les produits uniques afin de récupérer leurs titres et leurs descriptions ;
- les dictionnaires de « text mining[13] » afin de récupérer des groupes de mots de différents champs sémantiques : additifs (E322, lécithines), allégations nutritionnelles (Oméga 3), animaux (bœuf, poisson), antioxydants (E300, acide ascorbique), attributs (peaux sensibles, pasteurisé), catégorie (épicerie salée, surgelés), colorants alimentaires (E131, bleu patenté), conditionnement (bâton, tranche), conservateurs (E200, acide sorbique), domaines / châteaux (Château Margaux), fromages (Banon, Camembert), gammes (Activia, Coca Cola zéro), goûts (grillé, nature), marques (Evian, Ariel), nutriments (lipides, fer), packaging (barquette, bouteille plastique), parfums (amande, olivier), parties (filets, côtelette), pays (France, Thaïlande), unité (milligrammes, semaines), etc.

Le dictionnaire de Tesseract est fourni dans un format utilisable par l'OCR et non humainement lisible donc pour ajouter nos données à celui-ci, il faut exécuter quelques commandes :

[13] Technique faisant partie du domaine de l'intelligence artificielle permettant grâce à un ensemble de traitements informatiques d'extraire des connaissances d'un texte selon des critères de similarité complexes. Le BCC a construit au fil du temps ses propres dictionnaires de text mining pour inclure dans son processus de récupération un mécanisme de reconnaissance des schémas redondant dans les données récupérées (identification d'un poids, d'un fruit, d'un ingrédient, d'un pays, d'une marque, etc.)

Dé-combiner le fichier compilé/compressé fra.traineddata en fichiers fra.* bruts (format DAWG[14]) :

```
combine_tessdata –u fra.traineddata fra.
```

Récupérer le dictionnaire de mots au format texte (wordlist) depuis les fichiers bruts :

```
dawg2wordlist fra.unicharset fra.word-dawg wordlist
```

Nous avons ensuite créé un petit programme qui récupère toutes nos informations de la base de données et des dictionnaires de text mining, les convertit en mots et les ajoute au dictionnaire de Tesseract en supprimant les doublons. Le dictionnaire passe de 100 000 mots à 248 642 mots. Ce nouveau dictionnaire « new_wordlist » contient maintenant en plus de mots classiques, de nombreux mots supplémentaires présents sur les packagings des produits.
Il faut maintenant le rendre utilisable par l'OCR en le générant au bon format :

Convertir en fichiers bruts :

```
wordlist2dawg new_wordlist fra.word-dawg fra.unicharset
```

Combiner les fichiers bruts en un nouveau fichier combiné fra.traineddata compréhensible par l'OCR :

```
combine_tessdata fra.
```

L'OCR est configuré pour notre besoin et reconnait à présent des mots issus du domaine de l'alimentaire. Nous avons donc mis en place une librairie iOS permettant de récupérer des fragments de photos des utilisateurs de l'application du BCC.

[14] Directed Acyclic Word Graphs est une structure de données qui permet la recherche très rapide de mots.

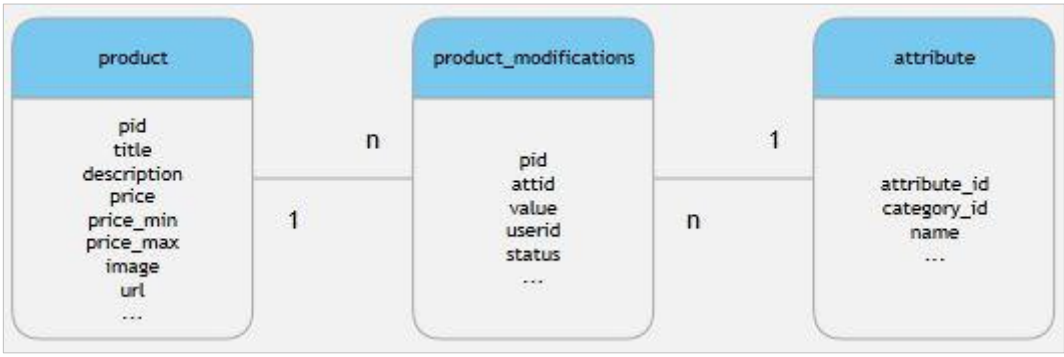
3.4.1.5 Librairie iOS de segmentation d'image

L'objectif est d'arriver à proposer un outil simple d'utilisation permettant d'extraire deux caractéristiques que sont la description et les ingrédients, d'une photo d'un emballage produit prise par un smartphone. Il faut que cette extraction permette d'appliquer par la suite des traitements OCR simples sans traitement complexe de nettoyage de l'image.

3.4.1.5.1 Conception de la récupération d'information

Tout d'abord, le principe de la librairie est qu'elle puisse être incluse aisément dans l'application iPhone native existante du BCC. Actuellement chaque modification d'une entrée dans une fiche produit au sein d'une application mobile est insérée en base de données (back-end) pour être validée par les administrateurs du BCC et mise ensuite à disposition de tous les utilisateurs (base de données front-end). Il faut que cette nouvelle fonctionnalité puisse entrer dans ce même processus.

Dans le schéma de données actuel, chaque produit est lié à une modification dans la table « product_modifications » grâce à son identifiant unique « pid ». Chacune de ces modifications a un attribut qui correspond au type de modification proposée par l'utilisateur : nom du produit, marque, image, catégorie, conditionnement, etc. et même les ingrédients et description au format texte. La table « attribute » contient ces types.



Afin d'ajouter les ingrédients et description au format image, nous allons créer deux nouveaux attributs : « Description image » et « Ingrédients image ».

Les webservices appelés depuis le mobile sauvegarderont l'image puis une nouvelle entrée dans la table « product_modifications » avec l'attribut correspondant au type de modification effectué dans la fiche produit et l'url vers l'image proposée comme « value ». Ces modifications suivront ensuite comme les autres le chemin de validation via les outils Intranet.

3.4.1.5.2 Outil mobile

Afin d'extraire des zones d'image uniformes lors de la prise de vue d'un emballage produit, nous avons réfléchi à un système de positionnement manuel de zones sur l'image en question.

En premier lieu, l'utilisateur prend une photo d'un produit qu'il a en sa possession. Une interface apparait et lui permet de définir des zones précises sur l'image grâce à des rectangles redimensionnables et déplaçables (Figure 28).

Sur la figure ci-contre ont été positionnés un rectangle orange pour la description et un rectangle vert pour les ingrédients. Ils peuvent être sélectionnés, déplacés et redimensionnés : cadre bleu autour du rectangle vert sélectionné.

Figure 28 : Segmentation d'une image

Pour se faire, nous avons utilisé comme base de développement l'implémentation d'une UIView[15] redimensionnable : SPUserResizableView [15] ; que nous avons adaptée à nos besoins afin qu'il soit possible d'ajouter ou de masquer plusieurs vues redimensionnables et de les déplacer sur les zones voulues de l'aperçu de la photo.

Ensuite, il faut réaliser la correspondance entre les zones sélectionnées sur l'image affichée à la résolution du téléphone (ici iPhone 4) de 640×960 et la photo sortie de l'appareil photo d'une résolution de 1936x2592 pour extraire la zone sur l'image en haute résolution.

[15] Object vue dans une application iOS. Définit une zone rectangle à l'écran et les interfaces pour gérer son contenu.

Une fois la sélection effectuée et les zones extraites de l'image originale, nous obtenons deux nouvelles images qui constituent les précédentes sélections (Figure 29).

Figure 29 : Segmentation d'une image et résultats

Si toutefois les deux informations clés ne se trouvent pas sur la même face du packaging produit, il est possible de ne définir qu'une seule zone à extraire dans la photo et de prendre une seconde photo par la suite pour créer la seconde zone.

Le résultat que nous obtenons correspond à deux photos uniformes ne contenant qu'une couleur de texte et un fond de couleur uni. Il sera donc beaucoup plus aisé d'appliquer un traitement OCR sur des images comme celles-ci. Il faut donc envoyer ces résultats au serveur en créant le webservice mobile.

3.4.1.5.3 Webservice d'ajout d'image « description » et « ingrédients »

Ce nouveau service mobile sera ajouté à l'API REST mobile. Il sera appelé via l'url « http://m.leboncotedeschoses.fr:9080/rest/**product/photo4ocr/<pid>/add?type=<type>** » où **<pid>** correspondra à l'identifiant du produit modifié et **<type>** au type de modification effectuée : description ou ingrédients.

Dans le fichier **.htaccess** nous mettrons la règle suivante qui permettra de rediriger vers le bon fichier PHP **productPhotoOcrAdd.php** :

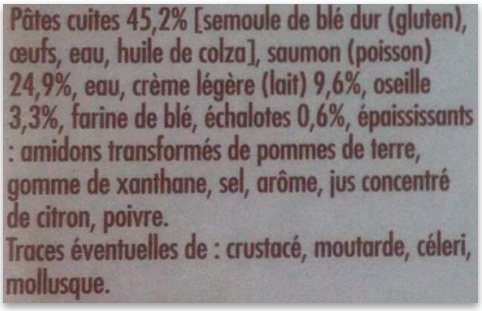
```
RewriteRule ^product/photo4ocr/(\d+)/add$
productPhoto4OcrAdd.php?pid=$1&%{QUERY_STRING} [L]
```

Des webservices métiers JAVA sont ensuite appelés pour se charger de la mise à jour de la base de données.

3.4.1.6 Reconnaissance du texte avec tesseract-ocr

Une fois en possession de parties d'images précises extraites proprement des emballages produits, il est possible de lancer le processus d'extraction de texte.

Le traitement d'images par un OCR nécessite souvent un prétraitement visant à supprimer les imperfections de l'image. Avec nos fragments d'image cette étape est tout de même nécessaire de manière moins poussée afin d'améliorer la netteté (éclaircissement, flou) des caractères. Effectivement, si nous lançons l'analyse OCR sur notre fragment d'image, voici le texte retourné :

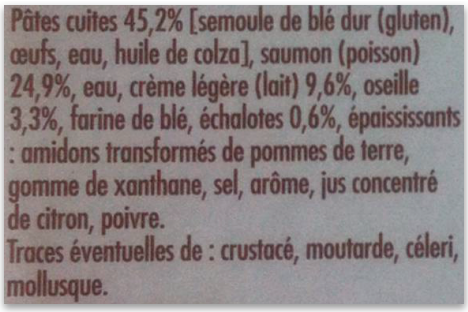



```
Pâtes cuites 45,2% [semoule de blé dur (gluten),
œufs, eau, huile de colzul, saumon (poisson)
24,9%, euu, crème légère (lait) 9,6%, oseille
3,3%, farine de blé, éfhulotes 0,6%, épuississnnts
: umklons trunslomtês de pommes (le terre,
gomme de xunthune, sel, arôme, lus concentré
de citron, poivre.
Traces éventuelles de : crustucé, moutarde, céleti,
mollusque.
```

Sans traitement, l'analyse a déjà été améliorée par l'ajout de mots spéciaux dans le dictionnaire mais l'image restant trop sombre et flou, certains caractères restent mal reconnus.
Nous avons créé une librairie C++ qui procède à ces traitements (grâce à la librairie de traitement d'images Leptonica) et lance ensuite l'OCR. Voici les étapes de traitement d'une image :

Chargement du fragment d'image extrait depuis l'application mobile et conversion de l'image en niveau de gris :

```
Pix* pixs = pixRead("fragment-image.jpg");
PIX* pixsg = pixConvertRGBToLuminance(image);
```

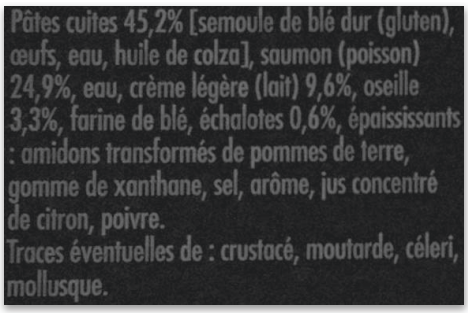


Application de l'algorithme de transformation « Top-Hat » [16] qui permet d'extraire des petits éléments d'images. Nous avons utilisé le type de transformation top-hat : « black top-hat transform ».

```
PIX* pixg = pixTophat(pixsg, 15, 15, L_TOPHAT_BLACK);
```

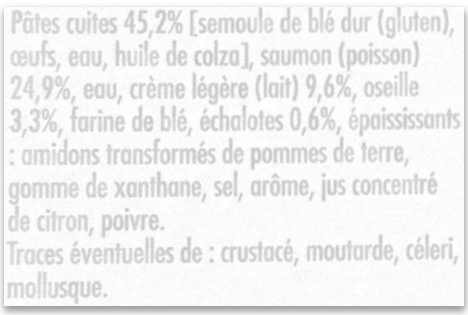
Cette transformation applique une différence entre l'image d'entrée et la même image traitée par un algorithme de fermeture morphologique mathématiques [17]. Cette fermeture permet grâce à un élément structurant d'une certaine taille d'appliquer une érosion qui fermera l'intérieur des lettres en ne laissant apparaître que l'ombre des mots :

La différence avec l'image d'origine restitue ensuite une image nette :



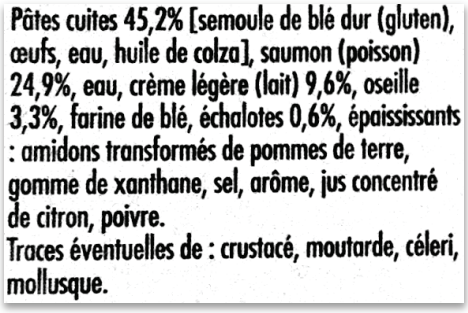
Nous appliquons ensuite une inversion des couleurs :

```
pixInvert(pixg, pixg);
```



Puis pour finaliser notre traitement, nous adaptons modifions le contraste de l'image :

```
pixd = pixGammaTRC(NULL, pixg, 1.0, 200, 245);
```



Le programme complet est joint en Annexe 4.

A l'issu de ces prétraitements et l'amélioration de la clarté de l'image, le texte en sortie de l'OCR est nettement plus lisible :

```
Pâtes cuites 45,2% [semoule de blé dur (gluten),
œufs, eau, huile de colza], saumon (poisson)
24,9%, eau, crème légère (lait) 9,6%, oseille
3,3%, farine de blé, échalotes 0,6%, épaississants
: amidons transformés de pommes de terre,
gomme de xanthane, sel, arôme, ius concentré
de citron, poivre.
Traces éventuelles de : crustacé, moutarde, céleri,
mollusque.
```

Nous remarquons que seul le mot « jus » n'a pas été reconnu et s'est transformé en « ius » ; ceci certainement dû à la police d'écriture.

Ces erreurs, bien que minimes, ne peuvent être propagées directement en visibilité des internautes. Nous avons donc décidé de nous servir du « Caddy Trophy » et de sa communauté pour mettre en place un système de vérification communautaire et ainsi qualifier les résultats de l'OCR.

3.4.1.7 Qualification des résultats

Ce qui donne du travail aux équipes du BCC au travers du « Caddy Trophy », c'est la qualification des résultats. Il est donc intéressant de réfléchir à une solution qui serait autosuffisante depuis la récupération de l'information jusqu'à sa mise en ligne.

3.4.1.7.1 Crowdsourcing de données « Crowdsourcées »

La prise de vue des packagings produit rentre dans le cadre du programme de crowdsourcing mais renvoie des résultats nécessitant un travail de relecture. Nous avons donc pensé pousser plus loin le travail de la communauté en proposant directement ces nouveaux textes erronés pour relecture au reste de la communauté.

3.4.1.7.2 Validation communautaire

Afin de proposer un outil ergonomique et surtout rapide à utiliser, nous sommes partis sur le même principe que ce que propose Wikisource [7] en apportant quelques améliorations. Le but étant au sein d'une page de l'espace utilisateur du site Internet de proposer aléatoirement les fragments de photos pris

par les autres utilisateurs accolés à leur texte fournis par l'OCR. Ainsi, chaque correction effectuée par les utilisateurs sera comptabilisée et lui apportera des points dans le cadre de sa participation au « Caddy Trophy ». Afin d'apporter un peu plus de sécurité aux corrections des utilisateurs, un système de seuil de validation devra être atteint pour chaque texte ; un texte ne sera mis en ligne que si 3 utilisateurs différents ont réalisé exactement la même correction sur le texte.
Afin de guider l'utilisateur et d'identifier des corrections similaires qui peuvent être très variées sur un texte assez gros, nous avons ajouté une fonctionnalité de modification par mots.

Pour cela, Tesseract OCR est capable de nous renvoyer à la place d'un texte brut, le texte au format HTML :

```
<p class='ocr_par' dir='ltr' id='par_1' title="bbox 6 1 577 291">

  <span class='ocr_line' id='line_1' title="bbox 10 1 569 40">

    <span class='ocrx_word' id='word_1' title="bbox 10 4 70 33">Pâtes</span>
    <span class='ocrx_word' id='word_2' title="bbox 79 6 143 34">cuites</span>
    <span class='ocrx_word' id='word_3' title="bbox 152 4 226 40">45,2%</span>
    <span class='ocrx_word' id='word_4' title="bbox 236 2 341 37">[semoule</span>
    <span class='ocrx_word' id='word_5' title="bbox 350 1 376 33">de</span>
    …
  </span>

  <span class='ocr_line' id='line_2' title="bbox 9 42 526 82">

    <span class='ocrx_word' id='word_9' title="bbox 9 43 71 82">œufs,</span>
    <span class='ocrx_word' id='word_10' title="bbox 82 55 130 81">eau,</span>
    <span class='ocrx_word' id='word_11' title="bbox 141 43 196 75">huile</span>
    <span class='ocrx_word' id='word_12' title="bbox 206 43 232 75">de</span>
    …
  …
</p>

<p class='ocr_par' dir='ltr' id='par_2' title="bbox 4 288 574 376">

  <span class='ocr_line' id='line_8' title="bbox 4 288 574 329">
    <span class='ocrx_word' id='word_47' title="bbox 4 297 75 326">Traces</span>
    …
  …
</p>
```

Ce qui est intéressant dans le format HTML proposé par l'OCR, c'est la modélisation de la structure visuelle du texte sur le packaging. Chaque paragraphe est représenté par une balise <p class='ocr_par'> et chaque ligne est représentée par un et chaque mot par une . Chaque balise contient des attributs renseignant sur le sens de lecture, « dir='ltr' » (left to right = de gauche à droite), le numéro de l'élément, « id », mais aussi une propriété qui va nous servir plus particulièrement, « title ». Cet attribut contient la « bounding box » (boîte englobante) de l'élément (paragraphe, ligne ou mot) encadré par la balise. Cela représente les coordonnées de chaque coin d'un rectangle englobant l'élément. Pour les mots, l'OCR renvoie en réalité chaque groupe de caractères, qu'il définit comme étant des mots, au sein d'une balise associée à sa boîte contenante. Ces coordonnées sont basées par rapport au fragment d'image, à sa taille originale, provenant de l'appareil photo du smartphone.

Grâce à ces coordonnées nous pouvons calculer proportionnellement les coordonnées sur l'image affichée et réaliser une interface web pour qu'au survol de chaque mot dans le texte, sur l'image adjacente soit encadré le mot correspondant. Cela permet à l'utilisateur de repérer directement sur l'image le mot qu'à voulu traduire l'OCR et ainsi faire lui-même la traduction unitaire du mot erroné (Figure 30).

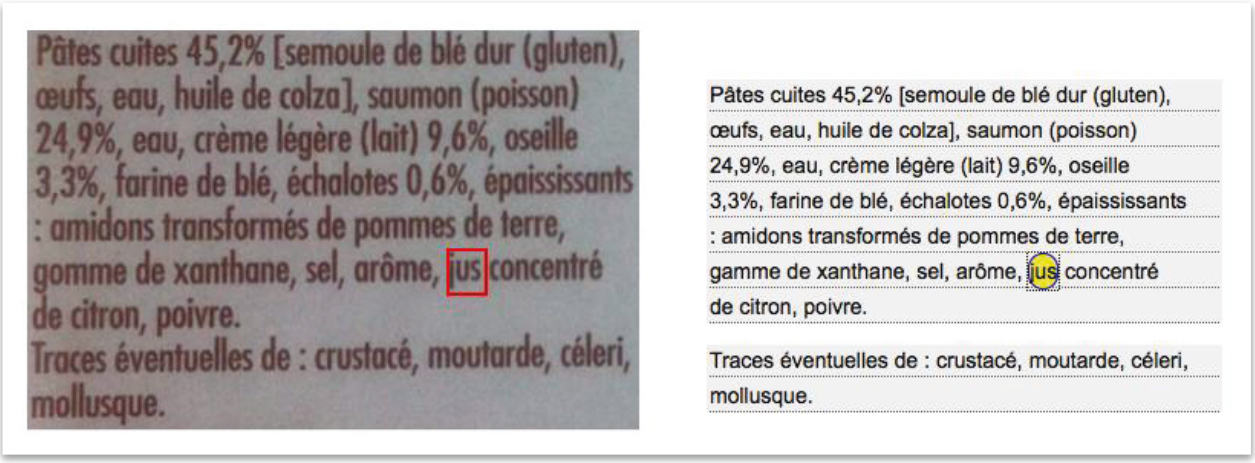


Figure 30 : Outils de validation des textes issus d'OCR

Les propositions de l'utilisateur sont ensuite enregistrées par texte et par mot et une fois que chaque mot du texte a été corrigé 3 fois de la même manière, le texte peut être mis en ligne. Bien sûr ce mode de fonctionnement de validation par 3 personnes devra être validé à moyen terme afin de repérer les dérives éventuelles du système.

3.4.2 Reconnaissance d'images

Le BCC possède un total de 430 000 images qui représente chaque image de chaque offre récupérée par les crawlers ou le crowdsourcing en place sur les applications mobiles. Ce qui est important pour le comparateur de prix est le rapprochement des offres des différents marchands par leur EAN, l'identifiant unique du produit. A l'inverse d'images, toutes les offres ne disposent pas toutes d'un EAN qui est une donnée fréquemment indisponible sur les sites marchands. Nous avons donc réfléchi à la création d'un rapprochement des offres avec EAN et des offres sans EAN grâce à leur image. En partant du principe que les photos des produits représentent le packaging du produit et que les sites marchands disposent des mêmes photos ou de photos prises sous le même angle, nous pouvons donc arriver à créer un programme permettant de trouver un nombre suffisant de similitudes pour considérer le produit comme identique.

Il convient de trouver des outils qui permettent d'absorber une grande partie du travail de reconnaissance d'image et qui s'inclut aisément à l'architecture logicielle du BCC. Après quelques recherches et analyse des solutions open source, une librairie de traitement d'images est sortie du lot : **OpenCV** [18]. Cette librairie écrite en C++ maintenu par la société Willow Garage, est spécialisée dans le traitement d'images en temps réel. Nous l'avons choisie face à quelques alternatives comme OpenIMAJ ou SimpleCV car elle propose le traitement bas niveau des images et implémente de base quelques algorithmes puissants. De plus, la librairie propose des interfaces en C++, C, Python et Java et peut s'intégrer facilement à l'environnement Linux présent au BCC.

3.4.2.1 Installation d'OpenCV

OpenCV sera installé sur une machine serveur du BCC sous l'environnement Linux Ubuntu. Nous pourrions installer une version grâce à la commande :

```
apt-get install libopencv-dev
```

Le problème est que par défaut la librairie n'est pas compilée avec le support Java. En effet, pour la réalisation de ce programme, nous avons besoin d'inclure la procédure au sein même des programmes Java du BCC. A l'inverse de la librairie de reconnaissance de texte sur les packagings qui peut être appelée simplement avec une image en entrée et un texte en sortie.

Il faut donc compiler les sources d'OpenCV avec le support Java à destination de notre plateforme Linux Ubuntu.

Récupération des sources :

```
git clone git://github.com/Itseez/opencv.git
cd opencv
git checkout 2.4.6.2
mkdir build
cd build
```

Installation des librairies nécessaires à la compilation :

```
sudo apt-get install build-essential
sudo apt-get install cmake
sudo apt-get install make
```

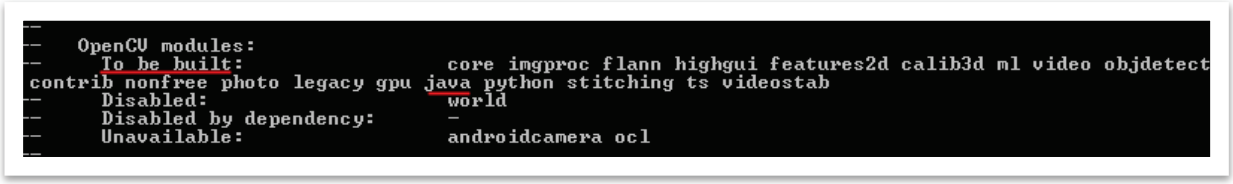
Définition du répertoire Java pour permettre la compilation du module Java :

```
export JAVA_HOME=/usr/lib/jvm/java-7-openjdk-amd64/
```

Génération du makefile et préparation de la compilation :

```
cmake -DBUILD_SHARED_LIBS=OFF ..
```

Dans le résultat du cmake vérifier que le module Java est bien dans les modules à compiler (To be built) :



Lancer la compilation :

```
make -j8
```

La compilation créera entre autre le fichier « jar » (bin/opencv_2.4.6.jar) contenant l'interface Java et une librairie native (bin/libopencv_java246.so) contenant les liaisons Java et toutes les fonctionnalités OpenCV.

3.4.2.2 Algorithme de matching

Pour réaliser notre programme de matching, nous avons décidé d'utiliser l'algorithme SURF (Speeded-Up Robust Features) qui est basé sur une méthode de détection par concordance de points, par détection de zones d'intérêts (Feature Detection) [19]. Cette technique vise à détecter des zones d'une image jugées « intéressantes » à analyser. Ces zones peuvent être identifiées par des points, des courbes, etc., selon la méthode utilisée. Les premières méthodes de détection se basaient sur l'analyse des contours et des arêtes qui sont des zones où la luminance (couleur) dans l'image change brusquement.

Dérivé de l'algorithme SIFT [20], SURF est considéré comme le plus puissant parmi les algorithmes de détection d'objet dans une scène. Il a été présenté pour la première fois en 2006 par Herbert Bay qui propose une amélioration en 2008. [21,22]

Afin de mettre en œuvre l'algorithme SURF plusieurs étapes successives de traitements sont nécessaires (Figure 31).

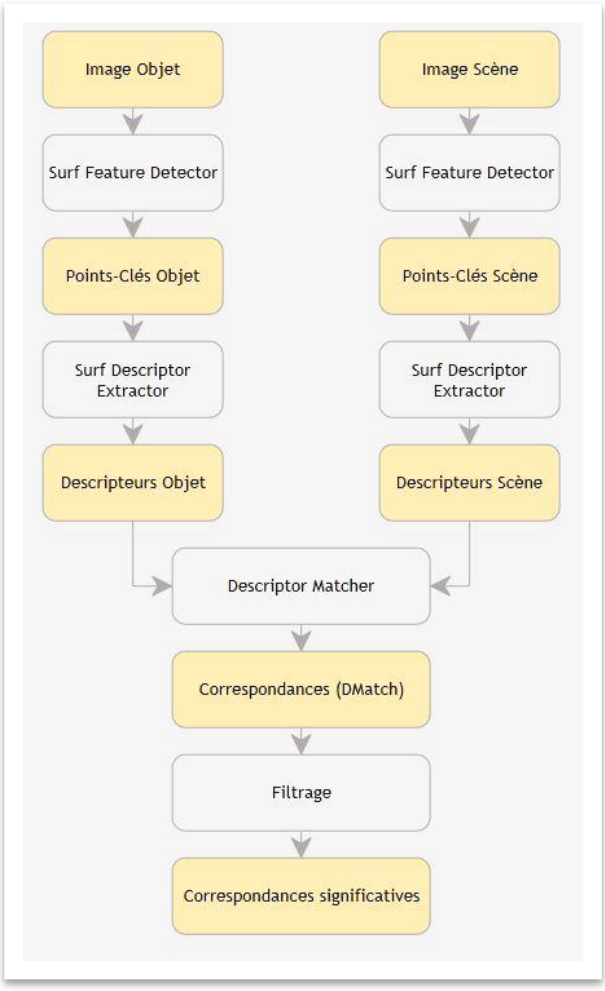


Figure 31 : Etapes de mise en œuvre de l'algorithme SURF

Chacune de ces étapes peut être traitée grâce à la librairie OpenCV, prenons deux images :

- A gauche, La photo ciblée d'un élément à rechercher que nous appellerons « objet » ;
- A droite, une photo, dans laquelle se trouve l'objet, que nous appellerons « scène ».

L'objectif de l'algorithme est de nous permettre de trouver l'objet dans la scène. Nous allons suivre les étapes une à une avec ces deux images pour comprendre le processus de reconnaissance. Les exemples proposés par la suite seront implémentés avec l'API Java d'OpenCV.

3.4.2.2.1 Surf Feature Detector

La première étape consiste à trouver les zones d'intérêt des deux images sous la forme de points clés. Cet ensemble constitue une sous partie de l'image qui résulte de décisions informatiques locales prises en chaque point de l'image. Un point clé se caractérise par :

- Un angle : orientation calculée du point clé ;
- Une position : coordonnées x et y du point ;
- Une taille : diamètre de la zone d'intérêt adjacente ;
- Une octave : couche de la pyramide à partir de laquelle le point clé a été extrait.

Premièrement, nous chargeons les images objet et scène :

```
Mat imgObject = Highgui.imread("object.jpg", 1);
Mat imgScene = Highgui.imread("scene.jpg", 1);
```

Ensuite, nous lançons la détection des points clés sur nos deux images en créant un détecteur de type SURF :

```
FeatureDetector detector = FeatureDetector.create(FeatureDetector.SURF);
MatOfKeyPoint keypointsObject = new MatOfKeyPoint();
MatOfKeyPoint keypointsScene = new MatOfKeyPoint();
detector.detect(imgObject, keypointsObject);
detector.detect(imgScene, keypointsScene);
```

Une fois nos points clés récupérés, nous pouvons avoir un aperçu de leur représentation sur les images avec des cercles de couleur (Figure 32).

Figure 32 : Représentation graphique des points clés de 2 images

Nous avons obtenu de nombreuses régions d'intérêts sur la scène et sur l'objet. Après l'algorithme de détection, nous appliquons un algorithme d'extraction de caractéristiques.

3.4.2.2.2 Surf Descriptor Extractor

Cette étape consiste pour chaque point clé détecté à calculer un « vecteur caractéristique » qui résumera le contenu de la zone en question. Effectivement, lorsque les données en entrée d'un algorithme sont trop grandes pour être traitées et peuvent contenir des redondances qui n'apportent pas plus d'informations, elles peuvent être transformées en une représentation numérique réduite de caractéristiques : c'est l'extraction de caractéristiques.

OpenCV nous donne accès à des fonctionnalités afin d'extraire ces caractéristiques depuis les points clés précédemment récupérés :

```
DescriptorExtractor extractor = DescriptorExtractor.create(DescriptorExtractor.SURF);
Mat descriptorsObject = new Mat();
Mat descriptorsScene = new Mat();
extractor.compute(imgObject, keypointsObject, descriptorsObject);
extractor.compute(imgScene, keypointsScene, descriptorsScene);
```

Une fois les descripteurs des deux images récupérés, nous pouvons commencer à détecter les correspondances. Ces descripteurs ont l'avantage d'être invariants à l'orientation, à la résolution de l'image, son exposition, sa netteté et sa perspective.

3.4.2.2.3 Descriptor Matcher

Dans cette étape, il s'agit de trouver les correspondances entre les descripteurs de deux images. Chaque correspondance (DMatch) est caractérisée par l'index du point de l'objet, l'index du point de la scène correspondant et une grandeur « distance » entre les deux points.
Pour réaliser la correspondance nous avons utilisé la librairie « FLANN » (Fast Library for Approximate Nearest Neighbors) qui permet de réaliser des recherches très optimisées des plus « proches voisins » sur un large jeu de données. Cette méthode permet de trouver les points les plus proches ou les points similaires.

Nous exécutons donc la méthode OpenCV en spécifiant que l'on souhaite récupérer les 2 correspondances les plus proches :

```
DescriptorMatcher matcher = DescriptorMatcher.create(DescriptorMatcher.FLANNBASED);
List<MatOfDMatch> matches = new ArrayList<MatOfDMatch>();
matcher.knnMatch(descriptorsObject, descriptorsScene, matches, 2);
```

Une fois les correspondances récupérées, il faut trier les « bonnes » correspondances, celles qui pour nous seront utiles pour la suite.

3.4.2.2.4 Filtrage

Cette étape va permettre de récupérer seulement les correspondances qui paraissent pertinentes pour nous. Nous allons comparer les distances de correspondances entre nos deux points avec un ratio acceptable ; 0.7 ici :

```
LinkedList<DMatch> goodMatches = new LinkedList<DMatch>();
float nndrRatio = 0.7f;
for (int i = 0; i < matches.size(); i++)
{
        if (matches.get(i).toList().size() < 2)
                continue;

        DMatch m1 = matches.get(i).toList().get(0);
        DMatch m2 = matches.get(i).toList().get(1);

        if (m1.distance <= nndrRatio * m2.distance)
                goodMatches.addLast(m1);
}
```

Voici une représentation des correspondances trouvées entre les deux images (Figure 33).

Figure 33 : Représentation graphique des correspondances entre 2 images

Nous pouvons donc voir que les correspondances entre les images s'orientent vers le bon objet dans la scène malgré quelques faux positifs. En sortie nous obtenons donc nos correspondances significatives et allons pouvoir rechercher notre objet dans la scène.

3.4.2.2.5 Recherche de l'objet dans la scène

Chaque « DMatch » correct récupéré garde un lien vers le point clé d'origine qui va nous permettre d'identifier notre objet sur la scène.

Pour chaque descripteur valide, nous récupérons les coordonnées de chaque keypoint sur l'objet et sur la scène :

```
List<Point> objectPoints = new ArrayList<Point>();
List<Point> scenePoint = new ArrayList<Point>();
for (int i = -1; ++i < goodMatches.size();)
{
        objectPoints.add(keypointsObject.toArray()[goodMatches.get(i).queryIdx].pt);
        scenePoint.add(keypointsScene.toArray()[goodMatches.get(i).trainIdx].pt);
}
```

Notre image d'objet et l'objet présent sur la scène ne sont pas forcément à la même taille ou à la même perspective. Il faut donc trouver grâce aux points de correspondance, la perspective entre les deux plans.

OpenCV fournit une méthode qui permet de trouver la perspective de transformation entre les plans source et destination. Cependant, ce ne sont pas toutes les paires de points (source et destination) qui réalisent une transformation de perspective rigide. Il existe dans le lot des paires considérées comme aberrantes, des valeurs isolées ou extrêmes par rapport au modèle global. Nous pouvons d'ailleurs voir dans la représentation graphique des correspondances (Figure 33) qu'il existe 5 points isolés du modèle cohérent. OpenCV propose donc l'utilisation de la méthode RANSAC (RANdom SAmple Consensus) appliquée à cette transformation. C'est une méthode itérative qui permet dans un jeu de données d'identifier les valeurs d'un modèle et d'écarter les valeurs aberrantes de ce modèle. [23]

Ainsi nous pouvons récupérer la perspective entre les points :

```
Mat H = Calib3d.findHomography(
new MatOfPoint2f(objectPoints.toArray(new Point[objectPoints.size()])),
        new MatOfPoint2f(scenePoint.toArray(new Point[scenePoint.size()])),
        Calib3d.RANSAC,
        3
);
```

Une fois la perspective récupérée, il faut l'appliquer aux 4 coins de notre image objet afin d'obtenir sur notre scène un rectangle identifiant l'emplacement de l'objet :

```
Point[] objectCorners = new Point[4];
objectCorners[0] = new Point(0, 0);
objectCorners[1] = new Point(imgObject.cols(), 0);
objectCorners[2] = new Point(imgObject.cols(), imgObject.rows());
objectCorners[3] = new Point(0, imgObject.rows());
MatOfPoint2f sceneCorners2f = new MatOfPoint2f();
Core.perspectiveTransform(new MatOfPoint2f(objectCorners), sceneCorners2f, H);
```

On dessine le rectangle en récupérant les nouvelles coordonnées des coins, transformées par la perspective :

```
Point[] sceneCorners = sceneCorners2f.toArray();
Point[] sceneCornersNorm = new Point[4];
sceneCornersNorm[0] = new Point(sceneCorners[0].x, sceneCorners[0].y);
sceneCornersNorm[1] = new Point(sceneCorners[1].x, sceneCorners[1].y);
sceneCornersNorm[2] = new Point(sceneCorners[2].x, sceneCorners[2].y);
sceneCornersNorm[3] = new Point(sceneCorners[3].x, sceneCorners[3].y);
Core.line(imgMatches, sceneCornersNorm[0], sceneCornersNorm[1], new Scalar(0, 255,
0), 4);
Core.line(imgMatches, sceneCornersNorm[1], sceneCornersNorm[2], new Scalar(0, 255,
0), 4);
Core.line(imgMatches, sceneCornersNorm[2], sceneCornersNorm[3], new Scalar(0, 255,
0), 4);
Core.line(imgMatches, sceneCornersNorm[3], sceneCornersNorm[0], new Scalar(0, 255,
0), 4);
```

Voici l'identification de l'objet trouvé sur la scène :

Figure 34 : Représentation graphique d'un objet trouvé dans une image

Nous avons un programme utilisable (le programme complet est joint en Annexe 5 et une version C++ est présentée dans la documentation OpenCV [24]) qu'il faut maintenant appliquer au panel de données du BCC. Dans notre cas de figure, les données ne seront pas réellement dans une scène comme l'exemple ci-dessus mais simplement une autre photo d'objet seul.

3.4.2.3 Données à traiter

Avant de s'attaquer aux offres dont nous n'avons pas toutes les images, nous allons prendre comme jeu de données cibles, les produits en doublon. Normalement un produit est unique mais lors de la récupération de l'offre, si elle n'a pas d'EAN, il nous est impossible de l'associer à son produit correspondant. Elle est donc créée comme produit unitaire et potentiellement en doublon.

Nous avons en notre possession deux groupes de produits représentés par 240 000 images, 15% sont des produits sans EAN et 85% sont des produits avec EAN qu'il faut réussir à rapprocher pour n'en créer plus qu'un.

La solution exhaustive serait d'appliquer notre algorithme de matching au produit cartésien des deux ensembles de produits.

$$(240\,000 * 15\%) * (240\,000 * 85\%) = 7\,344\,000\,000$$

Avec cette méthode, cela va donc entrainer dans le pire des scénarios, plus de 7 milliards de confrontations d'images. Suivant le poids de l'image que chaque procédure va avoir à charger, cela peut prendre énormément de temps.

Après plusieurs tests nous avons remarqué que pour que le matching donne des résultats corrects il faut une taille d'image pas trop petite et avons choisi de partir sur une taille d'image de 255x255 pixels pour un poids d'image aux alentours de 20-25ko.

Pour comprendre le choix de cette taille minimum, prenons un exemple avec deux packagings du même type de produit avec des saveurs différentes :

Si les images sont trop petites, nous allons perdre le détail important qui caractérisera notre produit, ici ce sont le titre et les photos de fruits sur l'emballage qui permettent de faire la différence entre les deux et de faire une correspondance fiable avec notre image en entrée.

Si nous réduisons la taille :

L'algorithme de matching va certainement donner le même nombre de points communs entre notre image en entrée et ces deux images, surtout par rapport à la forme et non par rapport à des détails intrinsèques au produit.

Avec ce format d'image en entrée de notre programme, le traitement de chaque image s'effectue en moyenne en 40 millisecondes. Si nous le soumettons au nombre d'exécutions maximales que nous aurions à effectuer, nous atteignons un temps de traitement inacceptable :

$$7\,344\,000\,00 * 0{,}04 \approx 9 \text{ ans}$$

C'est pour cela que nous avons réfléchi à la mise en place d'un traitement des produits par lots plus fins.

3.4.2.4 Optimisation fonctionnelle du temps de traitement des images

Les produits du BCC sont classés de manière très fine au sein de 3 niveaux arborescents de catégories. Logiquement, chaque offre récupérée par les « webcrawlers » est passée dans un programme de catégorisation afin de lui attribuer ses différentes catégories. Comme chaque enseigne des sites marchands affiche son propre système de catégorisation, le BCC à réaliser un « mapping » (correspondance une à une) des catégories de chaque magasin avec son propre modèle de données. Cela permet que chaque offre extraite d'une catégorie d'un site marchand se retrouve dans la même catégorie qu'une offre identique extraite sur un site marchant concurrent.

Schéma de catégories du BCC :

- 12 catégories de niveau 1 ;
- 149 catégories de niveau 2 ;
- 2000 catégories de niveau 3.

Exemple d'arborescence :

- Epicerie salée
    - Conserve de légumes
        - Conserve champignon
        - Conserve tomates
    - Pâtes, Riz
        - Spaghettis
        - Riz Thaï
- Crèmerie
    - Fromage
        - Fromage de brebis
        - Fromage frais
    - Yaourt
        - Yaourt 0%
        - Yaourt aux fruits

Voici le nombre de produits en moyenne répertoriés par niveau de catégorie :

- Niveau 1 : 12 000 produits ;
- Niveau 2 : 900 produits ;
- Niveau 3 : 40 produits.

Nous émettrons une réserve sur la catégorisation en niveau 3 qui n'est pas réalisée sur tous les produits car certains sites marchands n'ont pas de découpage aussi fin. Même si au premier abord c'est ce niveau qui nous permettrait de réaliser le moins de traitements possibles, ces catégories de niveau 3 ne fournissent pas un panel assez important de produits pour donner des résultats de match finaux très intéressants. On peut déjà imaginer réduire considérablement nos tests de correspondances en réalisant le mapping seulement entre les produits de même catégorie de niveau 2 :

$$(900 * 15\%) * (900 * 85\%) = 103\,275$$

Avec ce traitement par lots, nous obtenons une moyenne de 103 275 confrontations d'images par catégorie de niveau 2. Si nous le multiplions par le nombre de catégories :

$$103\,275 * 149 = 15\,387\,975$$

Dans le pire des scénarios, cela demande tout de même plus de 15 millions de confrontations pour couvrir l'ensemble des produits soit un total d'environ 7 jours :

$$15\,387\,975 * 0{,}04 \approx 7 \text{ jours}$$

Dans une prochaine évolution, il est prévu que pour le filtrage des produits nous prenions en compte la marque du produit en plus des catégories de niveau 2. Cela permettra de diviser encore le temps des reconnaissances et leur fiabilité en ne comparant que des produits de même marque et de même catégorie. D'autres critères pourraient encore être ajoutés à ces filtres comme les conditionnements (boîtes, bouteilles, etc.) ou poids/volumes mais ces caractéristiques ne sont pas encore assez fiables pour tous les produits. En attendant, il est possible d'apporter des améliorations techniques à notre programme afin de gagner encore un peu de temps de traitement.

3.4.2.5 Optimisation technique du temps de traitement des images

Avec notre programme de matching, nous nous rendons compte qu'un produit cartésien entre tous les produits (même filtrés) demande beaucoup d'instructions qui pourraient être réduites à une seule par produit recherché (objet).

Comme vu précédemment, dans notre cas de figure nous n'avons pas réellement de « scène ». C'est une image produit seule qui sert de lieu de recherche à notre programme. Nous avons donc décidé d'étudier plus fortement cette notion et de créer finalement cette scène avec tous les produits issus d'un filtrage. Plus précisément, nous avons créé des images représentant des grilles de produits de catégories de niveau 2 (Figure 35).

Figure 35 : Grille de produits (scène) de la catégorie « Compote »

Cette grille présentée en Figure 35 est formée de 257 produits qui ont été redimensionnés à la taille choisie de 255x255 pixels. Le zoom montre que l'image dans sa taille réelle contient bien des produits de cette

taille. Avec le nombre de produits par ligne et par colonne, cela donne une image en taille réelle de 4335x4080 pixels et d'un poids de 5,50Mo.
Ce type d'image pourrait contenir des centaines de produits supplémentaires et devenir relativement grande en termes de taille (+7000px) et de poids (+20Mo). Il faut donc que le programme puisse arriver à réaliser les traitements de détection de points clés et d'extraction de caractéristiques en mémoire avec de telles images ; cela peut aller pour chaque traitement à des manipulations de plus de 200Mo de données. Après quelques tests sur les serveurs puissants du BCC, ces traitements peuvent être absorbés, mais cette contrainte pourra être levée par un filtrage encore plus fin dans la construction des grilles comme expliqué précédemment.

Afin de réaliser le gain de temps espéré, nous avons créé un mécanisme permettant de générer les grilles régulièrement de manière automatique et d'enregistrer préalablement les points clés et descripteurs au format binaire afin de les réutiliser seulement lors de la comparaison sans avoir à les extraire à nouveau lors de chaque matching.

Grâce à cette grille, nous ne comparons donc plus tous les produits sans EAN avec tous les produits avec EAN mais tous les produits sans EAN avec une grille de produits avec EAN. Le calcul du nombre de confrontations d'images passe de 103 275 à 135 :

$$(900 * 15\%) * 1 = 135$$

Ce qui donne pour l'ensemble des catégories de niveau 2 :

$$135 * 149 = 20\,115$$

Avec la taille des grilles, les données traitées en mémoire sont plus conséquentes et le temps de comparaison est 100 fois plus important qu'entre des images de produits unitaires. Donc en se basant sur 4 secondes, avec nos grilles préalablement générées, nous obtenons un temps total de traitement d'environ 22h :

$$20\,115 * 4 \approx 22\text{h}$$

Sur le même principe qu'avec le matching par produit unitaire, nous retrouvons l'objet dans la scène grâce à ses coordonnées car au moment de la création de la grille, nous avons aussi stockés au format binaire, une matrice des identifiants des produits, permettant de les retrouver si une correspondance apparait. Voici le résultat de la recherche d'un produit dans une grille (Figure 36) :

Figure 36 : Matching d'un produit dans une grille

La prochaine étape sera d'exécuter le programme sur l'ensemble des offres avec photos afin de qualifier encore plus le service sur le panel de marchands proposé à la comparaison.

3.5 Outil de supervision

Toutes les actions prises en compte dans le « Caddy Trophy » permettant de gagner des points sont pour certaines soumises à validation d'administrateurs du BCC. Effectivement, certaines actions nécessitent une vérification des données proposées par l'utilisateur avant insertion en base de données et publication sur les différents flux publics du BCC. Cette validation permet de ne pas publier n'importe quoi sur des actions nécessitant un œil expert de l'équipe produit.

Par exemple, les nouveaux produits recatégorisés par les utilisateurs sont listés dans un tableau par ordre anti-chronologique. Pour chacun des produits, on peut visualiser l'utilisateur qui l'a soumis, son titre, la photo, l'ancienne catégorie déterminée par les crawls web marchands puis la nouvelle catégorie proposée par l'utilisateur. L'administrateur peut ensuite accepter ou refuser cette action. (Figure 37)

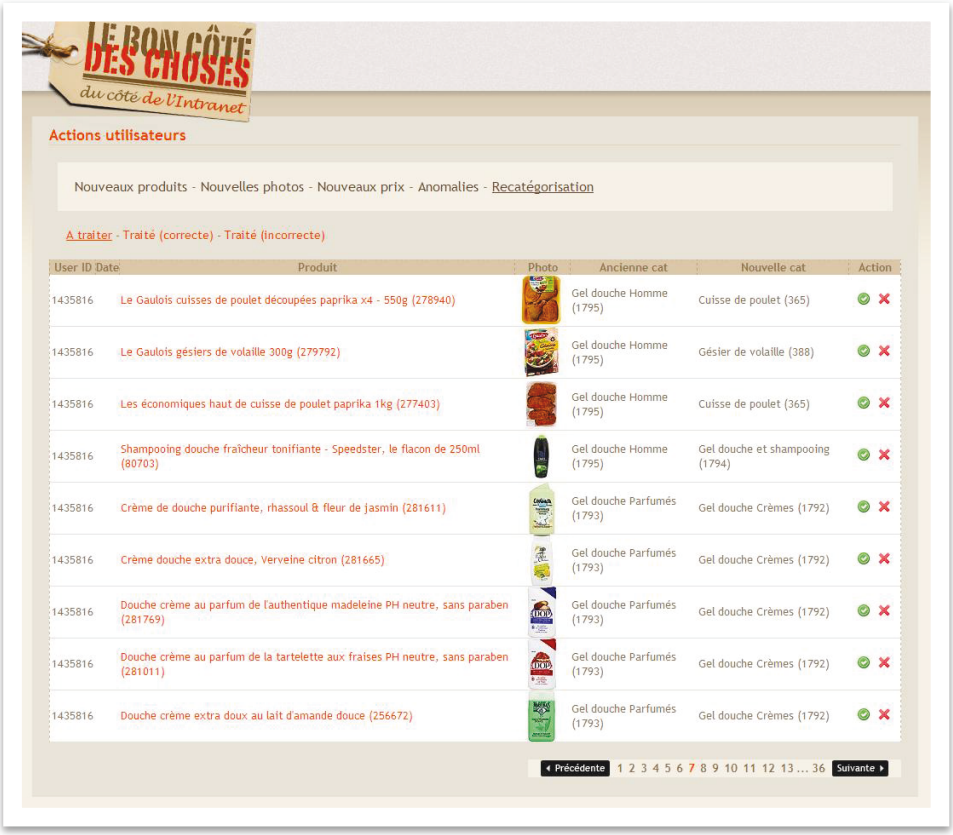


Figure 37 : Intranet, validation des actions utilisateurs - Recatégorisation

Nous pouvons cependant réfléchir pour certaines actions demandant peu de connaissance métier à les soumettre à validation de personnes lambda en reprenant le principe de validation communautaire utilisé pour les emballages alimentaires. Certaines validations telles que celles de l'ajout de nouveaux produits sont cependant difficiles à externaliser car elles demandent une vérification plus poussée de la cohérence des informations.

Pour conclure sur l'implémentation du projet, nous pouvons faire une synthèse sur l'apport des différentes briques développées.

Une fois l'intégration du Caddy Trophy au cœur des services du BCC effectuée, les premiers tests du système ont rapidement débuté grâce au panel récurrent d'utilisateurs. Le jeu a été accueilli favorablement comme une récompense aux aides que les membres actifs apportaient déjà, les poussant à « jouer » encore plus assidument. La refonte ergonomique et l'ajout de nouvelles fonctionnalités au site Internet et aux applications mobiles ainsi que la diffusion de l'information sur les réseaux sociaux d'un jeu concours récompensé, a permis d'attirer de nouveaux participants.

Grâce aux fonctionnalités de l'intranet, les administrateurs du BCC ont commencé la validation des propositions de nouvelles données s'accumulant proportionnellement au taux de participation. Globalement la pertinence était au rendez-vous, aussi bien du côté des remontées utilisateurs que du côté du programme de reconnaissance d'images appliqué aux fichiers binaires récupérés par les web crawlers.

Effectivement, avec ce programme de matching, nous avons pu rapprocher des dizaines de milliers d'EAN et de doublons nous permettant de mettre en ligne autant d'offres supplémentaires pour les produits existants, comparables grâce au moteur S2IO. Le raccrochement a permis d'avoir de nouvelles offres comparables mais il a aussi permis de récupérer des caractéristiques (description, ingrédients, etc.) entre les offres liées.

Seul le programme de reconnaissance de texte et son système de validation communautaire n'a pas encore été mis en exploitation. Le BCC a choisi d'attendre la refonte de son application iOS pour intégrer la brique de segmentation d'image.

Conclusion

Il est très intéressant de remarquer que l'information est accessible facilement de nos jours mais que pour la rendre utilisable, cela nécessite de la rendre pertinente et formatée à notre cas d'utilisation.
Le cas de figure du BCC entre totalement dans cet esprit là. Actuellement, de nombreuses données sont récupérables et classables informatiquement afin de créer un ensemble cohérent et structuré au sein du système d'information. Néanmoins, un pourcentage de ces données ne détient pas de connexion fonctionnelle intéressante avec le reste de la structure, le rendant superflu pour notre utilisation. Afin d'activer ces liaisons entre nos données, nous avons dû faire appel à la communauté du BCC pour apporter certains traitements humains sur l'ensemble de nos données et miser sur leur implication.

Après 3 mois de lancement du Caddy Trophy, plus de 45 000 actions (parrainage d'amis, recatégorisation, ajout de nouveaux produits, ajout de prix, ajout de photos, ajout de descriptions produits, etc.) ont été effectuées, soit plus de 15 000 participations par mois des membres du BCC. C'est là que l'on voit tout l'intérêt des méthodes de crowdsourcing et de l'apport que cela peut engendrer dans l'élaboration de micro-tâches. En termes de valorisation de ces actions sur ces 3 mois, le BCC a pu mettre en avant de nouveaux produits, des produits mieux qualifiés et voir plus de 8 000 nouveaux membres inscrits.
Il reste encore des mécanismes à roder et à améliorer sur le long terme. Une seconde version du Caddy Trophy est déjà en train d'être réfléchie pour palier certaines règles du système dans la participation trop active de certains membres par rapport à d'autres. La première version des règles du Caddy Trophy V2 est disponible en Annexe 6.

Durant ces 9 mois, nous avons mis en place plusieurs mécanismes de récupération et de traitement de l'information basés sur des concepts innovants. Des mécanismes présentés à l'utilisateur sur son ordinateur et son smartphone lui permettant de nous proposer des mises à jour de notre système. Des programmes de traitements de données textuelles et binaires dans le but d'augmenter la qualité de nos données. Tous ces travaux ont demandé une connaissance verticale totale de toute l'architecture du BCC afin de proposer une intégration optimale et de conserver un système cohérent et maintenable.

Le résultat de ce projet est très positif pour l'entreprise et pour moi-même. Effectivement, le BCC a en sa possession de nouveaux outils de qualification de sa base de données ainsi que de nouveaux axes de développement et d'amélioration pour le futur. Personnellement, l'environnement technique et humain dans lequel j'ai évolué durant cette période de stage m'a apporté énormément. La diversité des technologies à

mettre en place m'a permis d'élargir mon expertise et d'approfondir mes connaissances dans des domaines variés. La gestion du projet dans son ensemble a demandé la synchronisation de plusieurs personnes et tâches dans un contexte très mouvant pour la société, me permettant ainsi de me confronter à la réalité des priorités d'une entreprise en plein développement.

Glossaire

API	Interface de programmation normalisée par laquelle le programmeur a directement accès aux fonctions d'un système ou d'un autre programme.
CSS	Cascading Style Sheet : format de fichier lu par les navigateurs Internet dans lequel est regroupé l'ensemble des informations de style liées à une page Web.
Device	De l'anglais, périphérique qui dans notre cas vient de l'utilisation technique dans du code qui vise les différents types d'écrans des matériels.
DPH	Sigle utilisé par les professionnels de la grande distribution pour désigner le ou les rayons concernés par les produits de Droguerie, de Parfumerie et d'Hygiène.
EAN	Un code EAN (European Article Numbering) est un code barre utilisé par le commerce et l'industrie à des fins d'identification d'objets. Il identifie des articles de façon unique.
HTML	L'Hypertext Markup Language, est le format de données conçu pour représenter les pages Web. C'est un langage de balisage qui permet d'écrire la base de la représentation de site Internet.
HTTP	Protocole de communication client-serveur développé pour le Web. Les clients HTTP les plus connus sont les navigateurs Web (Internet explorer, Chrome, Firefox, etc.)
IHM	L'Interface Homme Machine est un ensemble de dispositifs matériels et logiciels permettant à un utilisateur d'interagir avec un système informatique.
JSON	Format de données textuelles dérivé de la notation des objets du langage Javascript.
OCR	La reconnaissance optique de caractères (ROC, en anglais optical character recognition : OCR) désigne les procédés informatiques pour la traduction d'images de textes imprimés ou dactylographiés en fichiers de texte.
Open Source	La désignation open source s'applique aux logiciels dont la licence respecte les critères de libre redistribution, d'accès aux sources et aux travaux dérivés.
Plateforme	En informatique, c'est un lieu physique ou virtuel qui centralise un ensemble de services. Une plateforme d'exécution pour une application est un espace où l'application peut se lancer et accéder à différents services.
PHP	Langage de programmation interprété. Il est utilisé pour produire des pages Web dynamiques.
REST	Style d'architecture basé sur le protocole HTTP permettant de construire la communication client-serveur d'applications (Web, mobile, etc.)
S2LO	« Social Shopping List Optimizer » est un algorithme propriétaire du BCC réalisé en collaboration avec un laboratoire de recherche grenoblois, G-SCOP. Il met sous contrôle d'importantes exigences de calculs permettant de calculer, suivant des paramètres multicritères complexes (allergies, géolocalisation, etc.), le coût d'une liste de course.

SDK	Kit de développement logiciel permettant de créer des applications de type défini.
SOA	Type d'architecture permettant la conception, l'intégration et la manipulation de différentes briques d'applications autour de services (SOAP, REST).
Smartphone	Téléphone dit « intelligent » qui donne accès à des fonctionnalités évoluées et autorise l'ajout de programmes (applications) spécifiques.
URL	Chaîne de caractères représentant l'adresse Web d'accès à une ressource (image, document, site Internet, etc.).

Bibliographie

[1] Neumann I. *Présentation des méthodes agiles et Scrum*. http://ineumann.developpez.com/tutoriels/alm/agile_scrum/

[2] *Academic representations of crowdsourcing, co-creation and open innovation*. http://yannigroth.com/2011/10/18/academic-representations-of-crowdsourcing-co-creation-and-open-innovation/

[3] Schenk E., Guittard C. *Crowdsourcing: What can be Outsourced to the Crowd, and Why ?* 7 décembre 2009. Disponible sur : http://halshs.archives-ouvertes.fr/halshs-00439256

[4] *ANAXAGO - Investir | Devenir business angel*. https://www.anaxago.com/

[5] *reCAPTCHA*. http://www.google.com/recaptcha/intro/index.html

[6] *Amazon Mechanical Turk*. https://www.mturk.com/mturk/welcome

[7] *Wikisource*. http://fr.wikisource.org/wiki/Wikisource:Accueil

[8] *Tesseract-OCR - An OCR Engine that was developed at HP Labs between 1985 and 1995... and now at Google*. https://code.google.com/p/tesseract-ocr/

[9] *OpenIMAJ - Intelligent Multimedia Analysis*. http://openimaj.org/

[10] *ABBYY Cloud OCR SDK*. http://ocrsdk.com/

[11] Cédric Verstraeten. *How to train Tesseract 3.01*. http://blog.cedric.ws/how-to-train-tesseract-301

[12] *Leptonica - Image processing tool*. http://www.leptonica.com/

[13] *TrainingTesseract3 - tesseract-ocr - How to use the tools provided to train Tesseract3 for a new language*. https://code.google.com/p/tesseract-ocr/wiki/TrainingTesseract3

[14] *Tesseract OCR Chopper*. http://pp19dd.com/tesseract-ocr-chopper/

[15] Poletto S. *SPUserResizableView - User-resizable, user-repositionable UIView subclass built for iOS*. In : *GitHub* [En ligne]. https://github.com/spoletto/SPUserResizableView

[16] Bloomberg D., Vincent L. *Document Image Applications*. In : Najman L, Talbot H (éd.). *Math. Morphol.* [En ligne]. p. 407‑420. Disponible sur : http://onlinelibrary.wiley.com/doi/10.1002/9781118600788.ch18/summary ISBN : 9781118600788.

[17] *Morphologie mathématique*. [En ligne]. *Wikipédia*. 2 avril 2014. Disponible sur : http://fr.wikipedia.org/w/index.php?title=Morphologie_math%C3%A9matique

[18] *OpenCV - Open Source Computer Vision*. http://opencv.org/

[19] *Feature detection (computer vision)*. [En ligne]. *Wikipedia Free Encycl.* Disponible sur : http://en.wikipedia.org/w/index.php?title=Feature_detection_(computer_vision)

[20] *SIFT - Scale-invariant feature transform*. [En ligne]. *Wikipedia Free Encycl.* Disponible sur : http://en.wikipedia.org/w/index.php?title=Scale-invariant_feature_transform&oldid=600615509

[21] Bay H., Tuytelaars T., Van Gool L. *SURF: Speeded Up Robust Features*. In : *Proc. 9th Eur. Conf. Comput. Vis. - Vol. Part I* [En ligne]. p. 404–417. Disponible sur : http://dx.doi.org/10.1007/11744023_32 ISBN : 3-540-33832-2, 978-3-540-33832-1.

[22] Bay H. et al. *Speeded-Up Robust Features (SURF)*. *Comput Vis Image Underst* [En ligne]. juin 2008. Vol. 110, n°3, p. 346–359. Disponible sur : http://dx.doi.org/10.1016/j.cviu.2007.09.014

[23] *RANSAC*. [En ligne]. *Wikipedia Free Encycl.* Disponible sur : http://en.wikipedia.org/w/index.php?title=RANSAC

[24] *Feature Matching with FLANN*. http://docs.opencv.org/doc/tutorials/features2d/feature_flann_matcher/feature_flann_matcher.html

Table des annexes

Annexe 1 : Interview avec Ferréole Lespinasse, rédactrice Web

Du crowdsourcing réussi : le Caddy Trophy du Bon Côté des Choses

Depuis l'origine, en 2010, le BCC interroge constamment ses utilisateurs (crowdsourcing) pour évoluer. Plus les utilisateurs s'impliquent, mieux le BCC répond à leurs besoins.

C'est ainsi que le Caddy Trophy a vu le jour. En échange de la contribution des IpCuriens pour mieux qualifier les produits de la base de données, le foyer qui totalise le plus grand nombre de points se voit rembourser un caddy de courses (jusqu'à 120 EUR).

Comment est venue cette idée de Caddy Trophy ?

Le Bon Côté des Choses, compte tenu de son activité de référencement de produits, est caractérisé par un nombre important de données complexes à mettre à jour.
Notre besoin initial était de nous appuyer sur les utilisateurs pour mieux qualifier les produits de nos bases de données et en ajouter de nouveaux. Nous voulions aussi que les utilisateurs participent à faire connaître la communauté.

Comment participer au Caddy Trophy ?

Plusieurs types d'actions, en mode connecté, peuvent être réalisés :
Classifier des produits, ajouter une photo, rectifier un prix, parrainer des amis ou signaler une anomalie. Un barème de points est attribué en fonction des actions. D'autres actions comme partager une vidéo sur Facebook permettent également de gagner des points. L'utilisateur, qui a totalisé le nombre maximal de points, remporte le Caddy Trophy.

Côté traitement des données entrées par les utilisateurs ?

Nous validons, bien sûr, les données entrées, ce qui prend un certain temps. Nous réfléchissons actuellement à une automatisation de cette validation, pour que le système soit autosuffisant. Par exemple, si 10 utilisateurs signalent un même prix et qu'il n'y a pas d'autres prix émis, alors, le prix est automatiquement rectifié.

Quel taux de participation ?

Sur la période de mi-mars à mi-juin 2013 (12 semaines), voici les chiffres
- En moyenne, 90 foyers pour 200 utilisateurs ont participé par semaine

- 1440€ et un smartphone ont été gagnés

Au niveau des actions les plus importantes nous ayant permis de fortement qualifier notre base de données de produits :

- 12 500 reclassifications de produits mal renseignés dans les catégories du BCC
- 3 700 nouveaux produits proposés par les utilisateurs

Les prochaines échéances

La manière, dont est fait le jeu, conduit à avoir sur plusieurs semaines le même gagnant.

Afin que le jeu soit égalitaire, nous réfléchissons à le réorienter, de manière à ce que chaque utilisateur puisse avoir une possibilité de gagner. Pour certains, cela ne prendra que deux semaines et pour d'autres plusieurs mois.

Que chacun tente sa chance !

Annexe 2 : API front-end des webservices d'« incentive »

Récupération des points d'un utilisateur

* METHOD : GET

- PATH : /fews/v1/incentive/get/points/{user_id}?
- PARAMS : aucun
- RETURNS : json sur les points de l'utilisateur
  - week : point sur la semaine courante
  - quarter : point sur le trimestre
  - year : point sur l'année
  - total : tous les points jamais acquis
- Exemple :

```
curl   'http://localhost:9080/fews/v1/incentive/get/points/22'

{"week":8,"quarter":80,"year":100,"total":120}
```

Récupération des badges d'un utilisateur

* METHOD : GET

- PATH : /fews/v1/incentive/get/badge/{user_id}
- PARAMS :
  - mobile, 0 = web, 1=mobile optionnel default 0
- RETURNS : json sur les badges de l'utilisateur
  - bid
  - obtained
  - date
  - name
  - description
  - ordre d'affichage
  - image_acquis image à afficher si badge acquis
  - image_non_acquis image à afficher si badge non acquis
- Exemple :

```
curl  'http://localhost:9080/fews/v1/incentive/get/badge/22'

[
```

```
      {
         "bid":1,
         "obtained":true,
         "date":"2013-01-22 11:16:49.0",
         "name":"Tout Nouveau Tout Beau",
         "description":"Je remplis mes préférences  (étapes 1,2,3)",
         "id":1,
         "ordre":2,
         "image_acquis":"/contents/badges/badge1.png",
         "image_non_acquis":"/contents/badges/badge1.png"
      }
]
```


Récupération des badges d'un utilisateur V2

* METHOD : GET

- PATH : /fews/v1/incentive/get/badge/{user_id}?
- PARAMS :
  - mobile, 0 = web, 1=mobile optionnel default 0
- RETURNS : json sur les badges de l'utilisateur, ordonné suivant la colonne ordre
  - **bid** id du badge
  - **obtained** true ou false
  - **date** date d'obtention, uniquement si obtained=true
  - **name** nom du badge
  - **description** description du badge
- Exemple :


```
curl  'http://localhost:9080/fews/v1/incentive/get/badge/22'

[
   {
      "date":"2013-01-22 11:16:49.0",
      "obtained":true,
      "bid":1,
      "name":"Tout Nouveau Tout Beau",
      "description":"Je remplis mes préférences  (étapes 1,2,3)"
   },
   {
      "obtained":false,
      "bid":2,
      "name":"Socialiseur",
      "description":"Je parraine un ami"
   },...
]
```

Récupération du rang d'un utilisateur et du classement des utilisateurs sur une plage

* METHOD : GET

- PATH : /fews/v1/incentive/get/ranking/{user_id}?
- PARAMS :
  - indexmin début de la pagination (optionnel, défaut 0)
  - indexmax fin de la pagination (optionnel, défaut 0)
  - type int (optionnel, défaut 1)
    - 1 : week
    - 2 : quarter
    - 3 : year
    - 4 : total
- RETURNS : json sur le rang de l'utilisateur
  - nbUser : nombre d'utilisateurs participants
  - point : nombre de points sur la période
  - rank : rang de l'utilisateur sur la période
  - table : tableau des users et leur place
    - place : place de l'utilisateur
    - point : nb de points
    - name : son prénom
    - ville : sa ville
    - user_id : id du user
    - pseudo : pseudo
- Exemple :

```
curl
'http://localhost:9080/fews/v1/incentive/get/ranking/276989?indexmin=0&indexmax=1'

{
   "nbUser":291246,
   "point":170,
   "rank":3,
   "table":[
      {
         "place":1,
         "point":260,
         "name":" Erualenna",
         "ville":"",
         "pseudo":""
      },...
   ]
}
```

Récupération du rang d'un utilisateur et du classement des utilisateurs autour de lui

Renvoie le classement centré sur l'user

* METHOD : GET

- PATH : /fews/v1/incentive/get/rankingAroundHim/{user_id}?
- PARAMS :
  - nombre d'éléments à remonter (optionnel, défaut 0)
  - type int (optionnel, défaut 1)
    - 1 : week
    - 2 : quarter
    - 3 : year
- RETURNS : json sur le rang de l'utilisateur
  - nbUser : nombre d'utilisateurs participants
  - point : nombre de points sur la période
  - rank : rang de l'utilisateur sur la période
  - table : tableau des user et leur places
    - place : place de l'utilisateur
    - point : nb de point
    - name : son prénom
    - ville : sa ville
    - user_id : id du user
    - pseudo : pseudo
- Exemple :

```
curl
'http://localhost:9080/fews/v1/incentive/get/rankingAroundHim/276989?indexmin=0&index
max=1'

{
   "nbUser":291246,
   "point":170,
   "rank":3,
   "table":[
      {
         "place":1,
         "point":260,
         "name":" Erualenna",
         "ville":"",
         "pseudo":""
      },...
```

```
    ]
}
```

Récupération du rang d'un foyer et du classement des foyers sur une plage

* METHOD : GET

- PATH : /fews/v1/incentive/get/rankingFoyer/{foyer_id}?
- PARAMS :
  - indexmin début de la pagination (optionnel, défaut 0)
  - indexmax fin de la pagination (optionnel, défaut 0)
  - type int (optionnel, défaut 1)
    - 1 : week
    - 2 : quarter
    - 3 : year
- RETURNS : json sur le rang du foyer
  - nbFoyer : nombre de foyers participants au Caddy Trophy
  - point : nombre de points sur la période
  - rank : rang de l'utilisateur sur la période
  - table : tableau des foyers participants
    - place : place de l'utilisateur
    - point : nb de point
    - name : son prénom
    - ville : sa ville
    - foyerId : id du foyer
- Exemple :

```
curl
'http://localhost:9080/fews/v1/incentive/get/rankingFoyer/1791?indexmin=0&indexmax=1'

{
   "nbFoyer":2,
   "point":170,
   "rank":1,
   "table":[
      {
         "place":1,
         "point":170,
         "name":"foyer 1791"
      },
      {
         "place":2,
         "point":0,
```

```
            "name":"foyer 520"
        }
    ]
}
```

Récupération du rang d'un foyer et du classement des foyers autour de lui

Renvoie le classement centré sur l'user

* METHOD : GET

- PATH : /fews/v1/incentive/get/rankingFoyerAroundIt/{foyer_id}?
- PARAMS :
  - nombre d'éléments à remonter (optionnel, défaut 0)
  - type int (optionnel, défaut 1)
    - 1 : week
    - 2 : quarter
    - 3 : year
- RETURNS : json sur le rang du foyer
  - nbFoyer : nombre de foyers participants au caddy trophy
  - point : nombre de points sur la période
  - rank : rang de l'utilisateur sur la période
  - table : tableau des foyers participants
    - place : place de l'utilisateur
    - point : nb de point
    - name : son prénom
    - ville : sa ville
    - foyerId : id du foyer
- Exemple :

```
curl
'http://localhost:9080/fews/v1/incentive/get/rankingFoyerAroundIt/1791?indexmin=0&ind
exmax=1'

{
   "nbFoyer":2,
   "point":170,
   "rank":1,
   "table":[
      {
         "place":1,
         "point":170,
         "name":"foyer 1791"
```

```
        },
        {
            "place":2,
            "point":0,
            "name":"foyer 520"
        }
    ]
}
```


Récupération du classement des foyers de la semaine précédente sur une plage

* METHOD : GET

- PATH : /fews/v1/incentive/get/oldRankingFoyer?
- PARAMS :
  - indexmin début de la pagination (optionnel, défaut 0)
  - indexmax fin de la pagination (optionnel, défaut 0)
  - date_fin_classement date de fin de classement du caddy trophy précédent (nombre de millisecondes depuis 1 janvier 1970)
  - date_debut_classement date de début de classement du caddy trophy précédent (nombre de millisecondes depuis 1 janvier 1970)
- RETURNS : json sur le rang du foyer
  - nbFoyer : nombre de foyers participants au Caddy Trophy
  - table : tableau des foyers participants
    - place : place de l'utilisateur
    - point : nb de point
    - name : le pseudo du foyer
    - ville : sa ville
    - foyerId : id du foyer
- Exemple :


```
curl
'http://localhost:9080/fews/v1/incentive/get/oldRankingFoyer?indexmin=0&indexmax=1'

{
   "nbFoyer":2,
   "table":[
      {
         "place":1,
         "point":170,
         "name":"foyer 1791",
         "ville":"Chambéry",
         "foyerId":2
      },
```

```
        {
            "place":2,
            "point":0,
            "name":"foyer 520",
            "ville":"Grenoble",
            "foyerId":3
        }
    ]
}
```

Récupération du classement last week du foyer d'un utilisateur

Renvoie le classement du foyer de l'utilisateur tel que déterminé le lundi passé à minuit

* METHOD : GET

- PATH : /fews/v1/incentive/get/history_foyer/{uid}
- PARAMS :
  - uid : l'id de l'utilisateur
- RETURNS : json incluant les champs suivants
  - winner_point : point du foyer vainqueur pour la semaine dernière.
  - winner : le pseudo du foyer vainqueur pour la semaine dernière.
  - point : nombre de points du foyer de l'utilisateur tel que calculé dans ce classement
  - rank : rang du foyer de l'utilisateur tel que calculé dans ce classement
- Exemple :

```
curl   'http://localhost:9080/fews/v1/incentive/get/history_foyer/106'

{
    "winner":"Foyer de Matt&Steph",
    "point":170,
    "rank":7
}
```

Récupération du classement last week d'un utilisateur

Renvoie le classement de l'utilisateur tel que déterminé Lundi dernier à minuit

* METHOD : GET

- PATH : /fews/v1/incentive/get/history_user/{uid}
- PARAMS :
  - uid : l'id de l'utilisateur

- RETURNS : json incluant les champs suivants
  - winner : le pseudo du foyer vainqueur pour la semaine dernière.
  - point : nombre de points du foyer de l'utilisateur tel que calculé dans ce classement
  - rank : rang du foyer de l'utilisateur tel que calculé dans ce classement
- Exemple :

```
curl   'http://localhost:9080/fews/v1/incentive/get/history_user/22'

{
   "winner":"Foyer de Matt&Steph",
   "point":170,
   "rank":7
}
```

Récupération de la gazette des scores

Renvoie la gazette personnalisée pour le foyer/l'utilisateur

* METHOD : GET

- PATH : /fews/v1/incentive/get/gazette/{uid}
- PARAMS :
  - (path) uid : l'id de l'utilisateur
  - type
    - foyer
    - user
  - html est ce qu'on veut la gazette sous un format html ?
    - 0 non
    - 1 oui
- RETURNS : un texte ou un html
- Exemple :

```
curl   'http://localhost:9080/fews/v1/incentive/get/gazette/22?type=foyer&html=1'

la gazette !!
```

Récupération de toutes les actions d'un utilisateur

Renvoie le classement centré sur l'user

- METHOD : GET

- PATH : /fews/v1/incentive/get/action/{user_id}?
- PARAMS :
  - user_id dans l'url
  - mobile, 0 = web, 1=mobile optionnel default 0
- RETURNS : json
  - summary un sommaire sur les actions validées
    - nombre : nombre de fois que l'action a été réalisée
    - point : nombre de points rapportés par l'action
    - name : nom de l'action
    - description : description de l'action
    - action_unitaire
      - 0 : ce n'est pas une action unitaire,
      - 1 c'est une action unitaire
    - visible
      - 0 non
      - 1 oui
  - history un historique des actions faites
    - point : nombre de points rapportés par l'action
    - date : date de réalisation de l'action
    - name : nom de l'action
    - description : description de l'action
    - type :
      - 0=validé,
      - 1=à valider,
      - 2=invalidé
    - visible
      - 0 non
      - 1 oui
  - notdone les actions jamais faites
    - point : nombre de points rapportés par l'action
    - name : nom de l'action
    - description : description de l'action
    - visible
      - 0 non
      - 1 oui
  - pendingPoints : nombre total de points en attente
- Exemple :

```
curl   'http://localhost:9080/fews/v1/incentive/get/action/276989'

{
   "summary":[
      {
         "nombre":1,
         "point":10,
         "name":"Je m'inscris",
         "description":"",
         "action_unitaire":1
      },...
   ],
   "history":[
      {
         "point":100,
         "date":"2013-01-22 12:26:09.0",
         "name":"Je remplis mes préférences (niveau 3)",
         "description":"",
         "type":0
      },...
   ],
   "notdone":[
      {
         "point":20,
         "name":"Je valide mon inscription",
         "description":""
      },...
   ]
}
```

Récupération des actions d'un utilisateur sur une période

Renvoie le classement centré sur l'user

* METHOD : GET

- PATH : /fews/v1/incentive/get/actionForPeriod/{user_id}?
- PARAMS :
  - user_id dans l'url
  - type int (optionnel, défaut 1)
    - 1 : week
    - 2 : quarter
    - 3 : year
    - 4 : all
  - mobile, 0 = web, 1=mobile optionnel default 0
- RETURNS : json d'actions

- summary un sommaire sur les actions faites sur la période, ordonné suivant le champ ordre de la table
  - numbre : nombre de fois que l'action a été réalisée
  - unitPoint : nombre de points rapportés si l'utilisateur effectue une fois l'action
  - point : nombre de points déjà rapportés par l'action à l'utilisateur
  - name : nom de l'action
  - description : description de l'action
  - id : id de l'action
  - action_unitaire
    - 0 : ce n'est pas une action unitaire,
    - 1 c'est une action unitaire
  - visible
    - 0 non
    - 1 oui
- pendingPoints : nombre total de points en attente

- Exemple :

```
curl   'http://localhost:9080/fews/v1/incentive/get/actionForPeriod/276989'

{
   "summary":[
      {
         "numbre":0,
         "point":0,
         "unitPoint":10,
         "name":"Je m'inscris",
         "description":"",
         "id":1,
         "action_unitaire":1
      },
      {
         "numbre":0,
         "point":0,
         "unitPoint":20,
         "name":"Je valide mon inscription",
         "description":"",
         "id":2,
         "action_unitaire":1
      },...
   ]
}
```

Récupération si un foyer et un user est admissible

* METHOD : GET

- PATH : /fews/v1/incentive/get/isFoyerAdmissible/{foyer_id}?user_id=?
- PARAMS :
  - foyer_id (in url) le foyer id
  - user_id (optionnel), l'id de l'user
- RETURNS : json d'actions
  - foyerAdmissible : true ou false
  - userAdmissible : true ou false
- Exemple :

```
curl   'http://localhost:9080/fews/v1/incentive/get/isFoyerAdmissible/{foyer_id}'

 {"foyerAdmissible":true,"userAdmissible":true}
```

Vérification sur le pseudo

* METHOD : GET

- PATH : /fews/v1/user/verif_pseudo?pseudo={pseudo}
- PARAMS :
- RETURNS : json
  - admissible : 0=invalidé, 1=validé
- Exemple :

```
curl   'http://localhost:9080/fews/v1/user/verif_pseudo?pseudo=Matthieu'

 {"valide":0}
```

Récupération des notifications

* METHOD : GET

- PATH : /fews/v1/incentive/get/notification/{user_id}?indexmin={indexmin}&indexmax={indexmax}
- PARAMS :
  - user_id : l'id de user
  - indexmin : l'index minimum
  - indexmax : l'index maximum
- RETURNS :
  - le nombre de notifications
  - La liste de notifications contient l'id de notification, le message et le date de création

- Exemple :

```
curl
'http://localhost:9080/fews/v1/incentive/get/notification/2?indexmin=0&indexmax=20

 {"nb":1,"list":[{"unid":20,"message":"Votre nouveau produit est
accepté","date":""}]}
```

Suppression d'une notification

* METHOD : GET

- PATH : /fews/v1/incentive/delete/notification/{unid}
- PARAMS :
  - unid : l'id de notification
- RETURNS : {"message":"ok"}
- Exemple :

```
curl   'http://localhost:9080/fews/v1/incentive/delete/notification/2

 {"message":"ok"}
```

Classement des foyers de la semaine courante (+points en attente)

* METHOD : GET

- PATH : /fews/v1/incentive/getFoyerRankingWeekPointWaitingPoint/
- RETURNS : JSONArray
- Exemple : [{"fid": 153, "weekPoint": 12, "pseudo": "bob", "waitingPoint": 66514}, {"fid": 155, "weekPoint": 12, "pseudo": "po", "waitingPoint": 27}]

```
 [{"fid": 153, "weekPoint": 12, "pseudo": "bob", "waitingPoint": 66514}, {"fid": 155,
"weekPoint": 12, "pseudo": "po", "waitingPoint": 27}]
```

Annexe 3 : API mobile des webservices d'« incentive »

Récupération des informations d'incentives

* METHOD: GET
  - PATH : /incentive/get
  - RETURNS :
    - rank_foyer :
      - lastweek :
        - winner : le nom du gagnant de la semaine dernière
        - rank : le classement de la semaine dernière
      - total : contient le top5 et le classement autour de lui
        - nbFoyer : nombre de foyers
        - point : le point total
        - rank : le classement actuel, 0 si le foyer n'est pas éligible au concours
        - table : la liste (top5/autour de lui) de l'utilisateur ; contient la place, les points, l'identifiant de foyer, le nom de foyer et la ville
          - place : la place
          - point : les points
          - foyerId : l'identifiant de foyer
          - name : le nom de foyer
          - ville : le nom de ville
      - week : contient le top5 et le classement autour de lui
        - nbFoyer : nombre de foyers
        - point : le point total
        - rank : le classement actuel, 0 si le foyer n'est pas éligible au concours
        - table : la liste (top5/autour de lui) de utilisateur contient la place, les points, l'identifiant de foyer, le nom de foyer et la ville
          - place : la place
          - point : les points
          - foyerId : l'identifiant de foyer
          - name : le nom de foyer
          - ville : le nom de ville
    - rank : contient le classement de semaine et le classement total
      - week : contient le top5 et le classement autour de lui
        - nbUser : nombre d'utilisateurs
        - point : les points de la semaine
        - rank : le classement de semaine

        - table : la liste (top5/autour de lui) de l'utilisateur ; contient la place, les points, l'identifiant de l'utilisateur, le nom, le pseudo et la ville
            - place : la place
            - point : les points
            - user_id : l'identifiant de l'utilisateur
            - name : le nom
            - pseudo : le pseudo
            - ville : le nom de ville
    - total : contient le top5 et le classement autour de lui
        - nbUser : nombre d'utilisateurs
        - point : le point total
        - rank : le classement actuel
        - table : la liste (top5/autour de lui) de utilisateur contient la place, les points, l'identifiant de l'utilisateur, le nom, le pseudo et la ville
            - place : la place
            - point : les points
            - user_id : l'identifiant de l'utilisateur
            - name : le nom
            - pseudo : le pseudo
            - ville : le nom de ville
- scores : contient les scores totaux et les scores de la semaine
    - total :
        - summary : la liste de points (le nombre d'actions, les points obtenus, le nom de l'action, les points unitaires, les descriptions et la visibilité)
            - numbre : le nombre d'actions réalisées
            - point : les points obtenus
            - unitPoint : les points unitaires
            - action_unitaire : le nombre de fois limité pour cette action
                - 0 : ce n'est pas une action unitaire,
                - 1 : c'est une action unitaire
            - name : le nom de l'action
            - description : les descriptions
            - visible : la visibilité
            - id : l'identifiant de l'action
            - order : l'ordre de l'action
        - pendingPoints : les points en attente
        - Points : les points totaux

- week :
  - summary : la liste de points (le nombre d'actions, les points obtenus, le nom de l'action, les points unitaires, les descriptions et la visibilité)
    - numbre : le nombre d'actions réalisées
    - point : les points obtenus
    - unitPoint : les points unitaires
    - action_unitaire : le nombre de fois limité pour cette action
      - 0 : ce n'est pas une action unitaire,
      - 1 c'est une action unitaire
    - name : le nom de l'action
    - description : les descriptions
    - visible : la visibilité
    - id : l'identifiant de l'action
    - order : l'ordre de l'action
  - pendingPoints : les points en attente
  - Points : les points de la semaine
- Badge : contient une liste de badge
  - bid : l'identifiant de badge
  - name : le nom
  - description : description
  - obtained : true si obtenue, false sinon
  - date : la date d'obtention (pour les badges obtenus)
  - ordre : ordre d'affichage des badges
  - image_acquis : image à afficher si le badge a été acquis :
    - exemple : /contents/badges/badge1.png
    - ⇒ test : 'http://localhost:9080//slir/w90-h90-c1:1//contents/badges/badge1.png
    - ⇒ prod : 'http://localhost:9080//slir/w90-h90-c1:1//contents/badges/badge1.png
    - slir permet de changer la taille directement dans l'url.
    - les images seront carrées.
  - image_non_acquis : image à afficher si le badge n'a pas été acquis
- Notification : contient le nombre total de messages et une liste de messages
  - unid : l'identifiant de notification
  - message : message
  - date : la date de création
  - type : le type de notification :
    - 0= ajout de produit, validé
    - 1= parrainage,

        - 2= badge,
        - 3= ajout photo validé
        - 4= recatégorisation validée
        - 5= signalement anomalie validé
        - 6= message incentive de lancement
        -
        - 100= ajout de produit, non validé
        - 103= ajout photo non validé
        - 104= recatégorisation non validée
        - 105= signalement anomalie non validé
    - home :
        - email : email
        - pseudo : pseudo
        - nb_notif : le nombre de notifications
        - notif1 : la 1ère notification
        - notif2 : la 2eme notification
        - points : les points de l'utilisateur
        - points_foyer : les points du foyer
        - avatar : l'avatar de l'utilisateur, éventuellement facebook, sinon bcc.
    - gazette_foyer le contenu text de la gazette du foyer
    - gazette_user le contenu text de la gazette du user

Exemple :

```
curl 'http://localhost:9080/rest/v2/incentive/get?token=...'

{
"rank_foyer" :
    {
    "week" : {
              "top5" : {
                        "nbFoyer":14,
                        "point":40,
                        "rank":4,
                        "table":[
                                 {
                                   "place":1,
                                   "point":700,
                                   "foyerId":520,
                                   "name":" foyer 520",
                                   "ville":"GRENOBLE"},
                                   ....
                                 ]
                       },
              "aroundhim" : {
```

```
                              ....
                              }
             },
    "total" : {
              ....
              }
    },
"rank" :
    {
    "week" : {
              "top5" : {
                        "nbUser":291397,
                        "point":60,
                        "rank":28,
                        "table":[
                                  {
                                    "place":1,
                                    "point":700,
                                    "user_id":22,
                                    "name":"Matthieu",
                                    "pseuso":"",
                                    "ville":"GRENOBLE"},
                                    ....
                                  ]
                       },
              "aroundhim" : {
                              ....
                              }
             },
    "total" : {
              ....
              }
    },
"scores" :
     {"total": {
                "summary":[
                           {
                             "numbre":0,
                             "point":0,
                             "unitPoint":200,
                             "name":"Je parraine un ami",
                             "description":"..",
                             "id":10,
                             "action_unitaire":0,
                             "visible":1
                           },
                           .....
                          ],
                 "pendingPoints":50,
                 "Points":60
               },
      "week": {
              ...
               }
     },
"badge":
```

```
    [
        {"bid":1,
         "name":"Tout Nouveau Tout Beau",
         "description":"Je remplis mes pr\u00e9f\u00e9rences  (\u00e9tapes 1,2,3)",
         "obtenu":false,
         "ordre":2,
         "image_acquis":"/contents/badges/badge1.png",
         "image_non_acquis":"/contents/badges/badge1.png"},
         ....
    ],
"notification":
    {"list":[
        {"unid":3,
         "message":"Votre scan produit est valid\u00e9",
          "date":"2013-01-25 14:11:36.0"},
          "type":0
          ....],
      "nb":2},
"home":
    {"email":"test01@leboncotedeschoses.fr",
     "pseudo":"test01",
     "nb_notif":3,
     "notif1": {
                  "unid":3,
                  "message":"Votre scan produit est valid\u00e9",
                  "date":"2013-01-25 14:11:36.0"
               },
     "notif2":{
                  "unid":6,
                  "message":"Votre scan produit est refus\u00e9",
                  "date":"2013-01-25 17:09:46.0"
               },
     "points":60,
     "points_foyer":40,
     "avatar":"http://test-medias.leboncotedeschoses.fr/contents/avatars/oeufs-
avatars.png"
    },
    "gazette_foyer":"Le texte de la gazette du BCC &#8211; caddy trophy.\nMerci de ne
pas mettre de mise en forme !\n",
   "gazette_user":"Gazette du BCC &#8211; IPcurien\n"
}
```


Récupération des règles du Caddy Trophy

* METHOD: GET

- PATH : /incentive/rules
- RETURNS :
  - Règles au format html

Annexe 4 : Programme C++ de traitement d'images pour OCR

```
PIX *pixs, *pixsg, *pixg, *pixd;

/* Read the image. */
if ((pixs = pixRead("fragment-image.jpg")) == NULL)
        return;

/* Convert the RGB image to grayscale. */
pixsg = pixConvertRGBToLuminance(pixs);

/* Black tophat (closing - original-image) and invert */
pixg = pixTophat(pixsg, 15, 15, L_TOPHAT_BLACK);
pixInvert(pixg, pixg);

/* Set black point at 200, white point at 245. */
pixd = pixGammaTRC(NULL, pixg, 1.0, 200, 245);

/* Clear. */
pixDestroy(&pixg);
pixDestroy(&pixd);
pixDestroy(&pixs);
pixDestroy(&pixsg);
```

Annexe 5 : Programme Java de reconnaissance d'images via SURF

```
System.loadLibrary(Core.NATIVE_LIBRARY_NAME);

String filePath = "/tmp/";

// Load images
Mat imgObject = Highgui.imread(filePath + "object.jpg", 1);
Mat imgScene = Highgui.imread(filePath + "scene.jpg", 1);

if (imgObject.empty() || imgScene.empty())
{
        System.out.println("(!) Error reading images");
        return;
}

//-- Step 1: Detect the keypoints using SURF Detector
FeatureDetector detector = FeatureDetector.create(FeatureDetector.SURF);

MatOfKeyPoint keypointsObject = new MatOfKeyPoint();
MatOfKeyPoint keypointsScene = new MatOfKeyPoint();

detector.detect(imgObject, keypointsObject);
detector.detect(imgScene, keypointsScene);

// Save image with keypoints
Mat objectKeypoints = imgObject.clone();
Features2d.drawKeypoints(imgObject, keypointsObject, objectKeypoints);
Highgui.imwrite(filePath + "objectKeypoints.jpg", objectKeypoints);
Mat sceneKeypoints = imgObject.clone();
Features2d.drawKeypoints(imgScene, keypointsScene, sceneKeypoints);
Highgui.imwrite(filePath + "sceneKeypoints.jpg", sceneKeypoints);

//-- Step 2: Calculate descriptors (feature vectors)
DescriptorExtractor extractor = DescriptorExtractor.create(DescriptorExtractor.SURF);

Mat descriptorsObject = new Mat();
Mat descriptorsScene = new Mat();

extractor.compute(imgObject, keypointsObject, descriptorsObject);
extractor.compute(imgScene, keypointsScene, descriptorsScene);

//-- Step 3: Matching descriptor vectors using FLANN matcher
DescriptorMatcher matcher = DescriptorMatcher.create(DescriptorMatcher.FLANNBASED);
List<MatOfDMatch> matches = new ArrayList<MatOfDMatch>();
matcher.knnMatch(descriptorsObject, descriptorsScene, matches, 2);

//-- Step 4: Filter matched descriptors

// Draw only "good" matches
// (i.e. whose distance is less than 0.7 * scene point distance)
LinkedList<DMatch> goodMatches = new LinkedList<DMatch>();
float nndrRatio = 0.7f;
```

```
for (int i = 0; i < matches.size(); i++)
{
        if (matches.get(i).toList().size() < 2)
                continue;

        DMatch m1 = matches.get(i).toList().get(0);
        DMatch m2 = matches.get(i).toList().get(1);

        if (m1.distance <= nndrRatio * m2.distance)
                goodMatches.addLast(m1);
}

Mat imgMatches = new Mat();
MatOfDMatch gm = new MatOfDMatch();
gm.fromList(goodMatches);
Features2d.drawMatches(imgObject, keypointsObject, imgScene, keypointsScene, gm,
imgMatches, Scalar.all(-1), Scalar.all(-1), new MatOfByte(),
Features2d.NOT_DRAW_SINGLE_POINTS);

//-- Step 5: Localize the object
List<Point> objectPoints = new ArrayList<Point>();
List<Point> scenePoint = new ArrayList<Point>();
for (int j = 0; j < goodMatches.size(); j++)
{
        // Get the keypoints from the good matches
        objectPoints.add(keypointsObject.toArray()[goodMatches.get(j).queryIdx].pt);
        scenePoint.add(keypointsScene.toArray()[goodMatches.get(j).trainIdx].pt);
}

// Save image with matches
Highgui.imwrite(filePath + "matches.jpg", imgMatches);

Mat H = Calib3d.findHomography(
        new MatOfPoint2f(objectPoints.toArray(new Point[objectPoints.size()])),
        new MatOfPoint2f(scenePoint.toArray(new Point[scenePoint.size()])),
        Calib3d.RANSAC,
        3
);

// Get the corners from the imgObject ( the object to be "detected" )
Point[] objectCorners = new Point[4];
objectCorners[0] = new Point(0, 0);
objectCorners[1] = new Point(imgObject.cols(), 0);
objectCorners[2] = new Point(imgObject.cols(), imgObject.rows());
objectCorners[3] = new Point(0, imgObject.rows());

MatOfPoint2f sceneCorners2f = new MatOfPoint2f();
Core.perspectiveTransform(new MatOfPoint2f(objectCorners), sceneCorners2f, H);

// Draw rectangle around the corners (the mapped object in the scene - imgScene)
Mat objectFoundOnScene = imgScene.clone();
Point[] sceneCorners = sceneCorners2f.toArray();
Point[] sceneCornersNorm = new Point[4];
sceneCornersNorm[0] = new Point(sceneCorners[0].x, sceneCorners[0].y);
sceneCornersNorm[1] = new Point(sceneCorners[1].x, sceneCorners[1].y);
sceneCornersNorm[2] = new Point(sceneCorners[2].x, sceneCorners[2].y);
```

```
sceneCornersNorm[3] = new Point(sceneCorners[3].x, sceneCorners[3].y);
Scalar green = new Scalar(0, 255, 0);
Core.line(objectFoundOnScene, sceneCornersNorm[0], sceneCornersNorm[1], green, 4);
Core.line(objectFoundOnScene, sceneCornersNorm[1], sceneCornersNorm[2], green, 4);
Core.line(objectFoundOnScene, sceneCornersNorm[2], sceneCornersNorm[3], green, 4);
Core.line(objectFoundOnScene, sceneCornersNorm[3], sceneCornersNorm[0], green, 4);

// Save image with object found
Highgui.imwrite(filePath + "objectFoundOnScene.jpg", objectFoundOnScene);
```

Annexe 6 : Spécifications Caddy Trophy V2

Règles

- Mise en jeu de 120€ par semaine dans une cagnotte globale qui correspondra à XXX points à atteindre ;
- Chaque action effectuée par l'utilisateur décrémente la cagnotte globale et incrémente une cagnotte personnelle ;
- Toutes les actions seront décrémentées de la cagnotte globale, actions validées et non validées
    - Comme seulement 10% des actions sont refusées, 10% de la cagnotte par semaine ne sera pas distribuée (transparent pour les utilisateurs car les refus sont dilués sur tous les joueurs et au final, gain de 10% par semaine pour le BCC) ;
    - Les points de parrainages dépendent de la date de validation du filleul, donc on décrémente tout de suite les points en attente de la valeur max que peut gagner le parrain sur la cagnotte de la semaine en cours *(à approfondir)*
        - PS : un user peut aussi faire 100 invitations et vider la cagnotte semaine. Euros qui ne seront jamais gagnés s'ils ne sont pas validés ;
        - PS2 : comment gérer les points que le BCC retire aux utilisateurs en cas de fraude ? Les points ne sont pas remis dans la cagnotte de la semaine comme les parrainages non validés.

  La cagnotte personnelle et le classement semaine seront incrémentés avec les actions validées ;
  Les actions non validées seront dans une cagnotte en attente (comme actuellement) ;
  Un utilisateur est rétribué lorsqu'il atteint la somme de 120€ au niveau de sa cagnotte personnelle
    - Envoie d'une notification d'attente des justificatifs pour recevoir la récompense ;
    - Remise à zéro (décrément de 120€) automatique de la cagnotte personnelle ;
    - Envoie d'un mail au responsable Caddy Trophy du BCC pour notifier d'un gagnant ;
    - Sur présentation de justificatifs de courses du mois *(à valider)*

Si la cagnotte est consommée avant la fin de la semaine :

- Ajuster au besoin les semaines suivantes la valeur des actions ;
- Arrêt du cumul des points côté back-end une fois la cagnotte globale atteinte.

Si la cagnotte n'est pas consommée à la fin de la semaine :

- Pas d'incidence, on réinitialise la cagnotte à 120€ la semaine suivante. Tout l'argent de la semaine n'est donc pas dépensé (gain pour le BCC) et permet de remotiver les gens à jouer encore plus la semaine d'après.

Classements

- Classement total pas nécessaire ou simplement pour info afin de voir le meilleur joueur du BCC de tous les temps !
    - Est-ce que les points acquis chaque semaine après que la cagnotte soit vidée sont pris en compte ?

  Classement de la semaine précédente conservé ;
  Classement par semaine : Le gagnant de la semaine remporte le droit de demander le montant de sa cagnotte personnelle en cours avant l'atteinte des 120€. Cela entraîne la remise à zéro du compteur de sa cagnotte personnelle (Sur présentation de justificatifs de courses de la semaine ?)

Problématiques :

- Si le classement de la semaine se base sur les gains des utilisateurs cumulés de semaine en semaine et que le plus gros joueur ne demande jamais sa cagnotte et n'est jamais remis à zéro, les nouveaux inscrits ne lui passeront jamais devant pour remporter les semaines suivantes.
- Si le classement de la semaine se base sur les gains de la semaine :
    - L'utilisateur peut demander la totalité de sa cagnotte : gains de la semaine + gains accumulés sur les semaines précédentes et sa cagnotte est remise à zéro automatiquement *(à approfondir)*
    - Cela fait maintenir trois types de points indépendants :

  Points totaux ;
  Points de la semaine ;
  Points de la cagnotte personnelle.
- Si on atteint 120€, est-ce que le classement de la semaine est décrémenté ?
    - Si oui, l'utilisateur a moins de chance d'être le gagnant de la semaine car il repasse en fin de tableau au passage à 120€ (compliqué à comprendre pour l'utilisateur) :

Temps	Classement semaine	Cagnotte personnelle
J1	0€	115€
J4 : Atteinte 120€	5€ -> 0€	120€ -> 0€
J7	5€	5€
J7+1	5€ -> 0€	5€

        - Remarque, le joueur aura bien gagné 10€ dans la semaine ;
        - L'utilisateur passe dernier au classement semaine à l'atteinte des 120€ ;
        - Il peut demander 5€ de la semaine s'il est premier du classement semaine.
    - Si non et s'il gagne le Caddy Trophy de la semaine, il gagne 120€ + l'argent restant de la semaine s'il a dépassé 120€ :

Temps	Classement semaine	Cagnotte personnelle
J1	0€	115€
J4 : Atteinte 120€	5€	120€ -> 0€
J7	10€	5€
J7+1	10€ -> 0€	5€

        - Il peut aussi demander 5€ de la semaine s'il est premier du classement semaine. Attention au calcul du 5€.

Affichage

Au niveau de l'interface, les gains totaux (décrémentés) et personnels (incrémentés) seront affichés sous la forme d'une monnaie virtuelle (ex : BCC Coins). Elle sera une conversion visuelle des 120€ à gagner.

Mémoire d'Ingénieur C.N.A.M

Récupération et traitement d'information via le crowdsourcing et la reconnaissance d'images

Matthieu Lombard Grenoble, juin 2014

Résumé :

De nos jours, l'information est partout et facilement accessible. Malgré cela, cette abondance nous demande de réfléchir à des moyens de l'exploiter et de la traiter de manière intelligente.

L'objectif principal de cette étude est de proposer la mise en place d'un écosystème de récupération et d'adaptation de diverses données, au sein du système d'information de la société Le Bon Côté des Choses.

En se basant sur des méthodes de « crowdsourcing », le projet développe la création d'un environnement communautaire participatif Web et mobile en vue d'apporter de nouveaux axes de récupération, de traitement et de qualification d'informations textuelles et binaires. Des techniques de traitement d'images et de reconnaissance de texte viendront appuyer plusieurs de ces mécanismes.

Mots-clés : crowdsourcing, reconnaissance d'images, traitement d'images, reconnaissance optique de caractères, récupération d'information

---

Abstract:

Nowadays, a large amount of data are generated and available everywhere. However, there is a huge challenge on how to efficiently exploit, process and leverage this immense flow of information.

The main objective of this disseration is to demonstrate how we can efficiently aggregate various data sources and smoothly integrated them in order to create a new data ecosystem for Le Bon Côté des Choses.

The purpose of this project is to develop new approaches to gather, process and classify textual and binary data supported by a Web & mobile community. Novatives crowdsourcing technologies, automatic image processing and text recognition methodologies have been used throughout this project.

Keywords: crowdsourcing, image recognition, image processing, OCR, information retrieval

---