



RAPPORT DE STAGE

Du 15 Mars au 31 Août 2015



"Développement de fonctionnalités sur une application RIA/SPA"

Bastien VANDERCHMITT
Master II Informatique - Spécialité AIGLE

Encadrants : Simon BAUDRY - Abdelhak SERIAI

Sommaire

1. Introduction
 2. Présentation
 - 2.1. Contexte du stage
 - 2.1.1. Entreprise
 - 2.1.2. Equipe
 - 2.1.3. Very Good Moment : le client
 - 2.2. Objectifs détaillés du stage
 - 2.3. Etat des lieux
 3. Cahier des charges
 - 3.1. Présentation générale
 - 3.2. Expression fonctionnelle du besoin
 - 3.3. Cadre de réponse
 4. Organisation
 - 4.1. Formation
 - 4.2. Outils
 - 4.3. Méthode Agile
 - 4.4. Planification
 5. Conception
 - 5.1. Analyse
 - 5.1.1. Modèle Conceptuel des Données
 - 5.1.2. Modèle Physique des Données
 - 5.2. Architecture
 - 5.2.1. Couches logicielles
 - 5.2.2. Services Php
 - 5.2.3. Services REST
 - 5.2.4. Interface Web
 6. Réalisation
 - 6.1. Méthodologie
 - 6.2. Développement
 - 6.2.1. Librairies et Frameworks
 - 6.2.2. Widgets
 - 6.2.3. Exemple de la Timeline
 7. Résultat et discussion
 - 7.1. Application finale
 - 7.2. Problèmes rencontrés
 - 7.3. Apports à l'entreprise et personnels
 8. Conclusion
- Annexes

Remerciements

Je voudrais d'abord remercier Simon Baudry et Baptiste Gaillard pour l'accueil chaleureux dont j'ai bénéficié au sein de leur entreprise ainsi que pour le partage de leurs connaissances et leurs précieux conseils. De même pour Rolland Houssou qui apportait sa bonne humeur chaque matin lors des réunions.

Je tiens également à remercier l'ensemble des enseignants et du personnel de la Faculté des Sciences de Montpellier pour les connaissances qu'ils m'ont permis d'acquérir au cours de ma formation et plus particulièrement Abdelhak Seriai pour m'avoir guidé lors de ce stage.

Résumé

De nos jours l'informatique est omniprésente dans notre quotidien et continue d'évoluer à chaque seconde qui s'écoule. De plus en plus de monde ressent le besoin d'avoir un accès internet et de "surfer" sur le web. Seulement, si beaucoup d'individus comprennent comment se servir des sites internet et des outils informatiques en général, moins nombreux sont ceux qui savent comment sont réalisés ces "logiciels du web". C'est pour cette raison que j'ai orienté mes études dans un cursus universitaire en informatique qui m'a mené droit vers ce stage en entreprise.

Un stage de fin d'études permet à l'étudiant de s'imprégner du monde de l'entreprise et de s'approprier les mécanismes indispensables pour son futur développement personnel dans le monde du travail. Ce rapport de stage consiste à montrer et expliquer mon expérience dans l'entreprise Gomoob qui fût l'occasion de mettre en pratique les techniques et connaissances apprises à la Faculté des Sciences de Montpellier tout en continuant d'apprendre auprès de professionnels. L'objet de ce stage porte sur le développement une application web riche et intégrable dans d'autres sites web pour un client, la start-up Very Good Moment.

Abstract

These days Information Technology is omnipresent in our daily lives and continues to evolve with every second that passes. More and more people feel the need to have an internet access and "surf" the web. But if many people understand how to use websites and computer tools in general, fewer people know how they are made these "Web software." That is why I directed my studies in a university course in computer science that led me right to this internship.

An internship graduation allows students to impregnate the world of business and to appropriate the necessary mechanisms for its future personal development in the workplace. This internship report is to show and explain my experience in the Gomoob company that was an opportunity to put into practice skills and knowledge learned in the Faculty of Sciences of Montpellier while continuing to learn from professionals. The purpose of this internship is to develop a rich monospace web application integrable in other websites for an exclusive client named VeryGoodMoment.

1. Introduction

Dans le cadre du programme de Master 2 Informatique de la Faculté des Sciences de Montpellier II, il nous est demandé de réaliser un stage en entreprise ou en laboratoire (stage de recherche ou stage professionnel). J'ai choisi un stage professionnel de manière à pouvoir appliquer les connaissances acquises lors de ma formation à la Faculté ainsi que pour développer des compétences afin d'être mieux préparé au monde professionnel.

C'est dans cette optique que j'ai contacté la société GOMOOB, une entreprise spécialisée dans le secteur d'activité du conseil en systèmes et logiciels informatiques, dirigée par Simon Baudry et Baptiste Gaillard qui m'ont accepté dans leur équipe suite à un entretien. Ma mission au sein de l'entreprise consiste à concevoir et développer des fonctionnalités sur une application web RIA/SPA.

Ce rapport a pour objet de présenter et d'expliquer le travail accompli lors de ce stage en entreprise.

2. Présentation

2.1. Contexte du stage

2.1.1. Entreprise

L'agence Gomoob a été créée par Baptiste Gaillard et Simon Baudry en janvier 2012 suite à leurs expériences dans des grands groupes tels que Thales et Sopra Group. C'est une entreprise de développement web et mobile capable de répondre à des besoins spécifiques et des problématiques complexes. Elle se positionne sur des technologies innovantes pour apporter à ses clients des solutions sur mesure à forte valeur ajoutée et des conseils personnalisés.

Les domaines d'expertise de l'entreprise sont :

- Plateformes Web & mobiles (Responsive Web Design)
- Applications Web RIA (Rich Internet Applications)
- Applications mobiles HTML5 (iOS, Android, Windows Phone)
- Solutions complètes Client Side & Server Side

2.1.2. Lieu

Les locaux de l'entreprise Gomoob se situent dans le centre de Montpellier à la Maison de Heidelberg. Ces locaux servent également de logement à plusieurs entreprises de type 'start-up'. Ces entreprises sont spécialisées dans le développement web, la sécurité informatique, et l'architecture en 3D.

2.1.3. Very Good Moment : Le client

L'application sera au service de Very Good Moment, une jeune entreprise de home marketing. Leur activité consiste à mettre en relation les marques et les consommateurs autour d'un même produit. C'est-à-dire que les marques sollicitent l'entreprise lorsqu'elles ont besoin de promouvoir une nouvelle référence. La plateforme de marketing collaboratif met alors en place un "very good moment". C'est un événement où plusieurs hôtes devront accueillir le même jour leurs amis autour d'un thème (et d'un produit indirectement).

A mi-chemin entre événementiel et publicité, Very Good Moment donne l'opportunité aux marques de créer du buzz autour de leurs produits en les faisant tester en situation réelle par des consommateurs volontaires lors d'un événement. Seule contrainte pour ces derniers : se mettre en scène grâce au "kit" (le colis) envoyé spécialement par la marque à ces ambassadeurs d'un jour en se filmant et en se photographiant. Ces photos et vidéos, sont alors publiées sur la plateforme où les "very good people" (les utilisateurs) peuvent voter pour élire le plus beau moment afin que son organisateur remporte un cadeau offert par la marque.

Very Good Moment est donc une plateforme permettant aux marques de montrer une image proche de ses consommateurs tout en véhiculant une bonne publicité pour ses produits. Voici quelques grandes marques en partenariat avec Very Good Moment : Microsoft, HP, TicTac, Acer, Ferrero, Rocher, Milka, La Redoute, ...

2.2. Objectifs détaillés du stage

Création d'une application web RIA intégrable sur un composant html pour le client Very Good Moment.

“ Une application Internet riche (en anglais Rich Internet Application for R.I.A.), est une application Web qui offre des caractéristiques similaires aux logiciels traditionnels installés sur un ordinateur. La dimension interactive et la vitesse d'exécution sont particulièrement soignées dans ces applications Web. ”

Citation - Wikipédia

L'application sera également de type monopage (SPA) dont le principe est expliqué ci-dessous.

“ Une application web monopage (en anglais single-page application ou SPA) est une application web accessible via une page web unique. Le but est d'éviter le chargement d'une nouvelle page à chaque action demandée, et de fluidifier ainsi l'expérience utilisateur. ”

Citation - Wikipédia

L'application finale devra donc répondre aux critères suivants :

- Rapidité d'exécution
- Navigation sur une seule page web
- Intégrable sur n'importe quel conteneur (site web)
- Facilité d'utilisation (bonne ergonomie)

2.3. Etat des lieux

Avant de présenter le travail accompli, il faut que je vous décrive en détail le contenu existant et le mode opératoire mise en place par le client et la société Gomoob.

Un site web est déjà en ligne et opérationnel à l'adresse <https://www.verygoodmoment.com> Ce site internet permet à l'utilisateur de s'inscrire à la plateforme pour pouvoir participer aux moments proposés par les marques en partenariat avec Very Good Moment. Lors de l'inscription, un questionnaire avec une série de questions sur le profil consommateur de la personne souhaitant s'inscrire doit être rempli. Une fois l'inscription achevée, l'utilisateur peut se connecter au site internet avec son identifiant (un pseudo) et son mot de passe. A partir de là, le Very Best People (c'est de cette façon que sont désignés les utilisateurs de Very Good Moment) peut participer à un moment. Mais attention, pas n'importe quel moment.

En effet, les moments peuvent avoir différents états :

- **Les moments à saisir** : ce sont des moments qui viennent juste d'être créés par la marque. Les utilisateurs peuvent y participer en s'y inscrivant. L'inscription à un moment peut se faire de deux manières. La première consiste à être l'ambassadeur du moment. Il faut alors remplir un questionnaire réalisé par la marque. La deuxième, plus simple, consiste à être invité par un ami ambassadeur à participer à son moment. Il n'y a environ que 20 colis disponibles pour un moment donc une sélection des équipes qui profiteront réellement des produits de la marque s'impose. Celle-ci est réalisée en fonction des réponses aux questionnaires remplis ainsi qu'en fonction du niveau de points et de l'expérience de l'utilisateur sur la plateforme Very Good Moment.
- **les moments en cours** : comme son nom l'indique, ces moments ont passé l'étape de la sélection et sont maintenant en cours. L'ambassadeur et ses invités vont recevoir le colis de la marque chez eux, passer un "very good moment" (une soirée, un goûter, un repas, bref un rassemblement entre amis) avec les produits et les goodies (produits dérivés) de la marque, et prendre des photos et des vidéos de ce moment.

- **les moments en élection** : une fois le moment passé, les participants à celui-ci peuvent venir poster les vidéos et photos qu'ils ont prises sur la plateforme web. Ainsi, chaque utilisateur du site web peut voter pour un feedback (une photo ou vidéo) pour une catégorie de vote de son choix. La catégorie du vote est choisie par la marque mais Very Good Moment propose les catégories suivantes par défaut : very friendly, funny, sexy, stylé et mignon.
- **les moments passés** : quand la période des votes est terminée, l'ambassadeur qui à obtenu le meilleur score est nommé Very Best Hôte et remporte un gros lot offert par la marque associée. L'image ou la vidéo ayant le nombre de votes le plus élevé est appelée Very Best Moment et est affichée sur la page d'informations du moment. Lors de l'élection, la marque dispose également d'un rôle important puisqu'elle peut voter pour un feedback ce qui apporte un sérieux bonus (20%) à son bénéficiaire. Une fois le moment terminé, les utilisateurs ne peuvent plus effectuer d'actions dessus mais il reste visible par l'ensemble de la communauté.

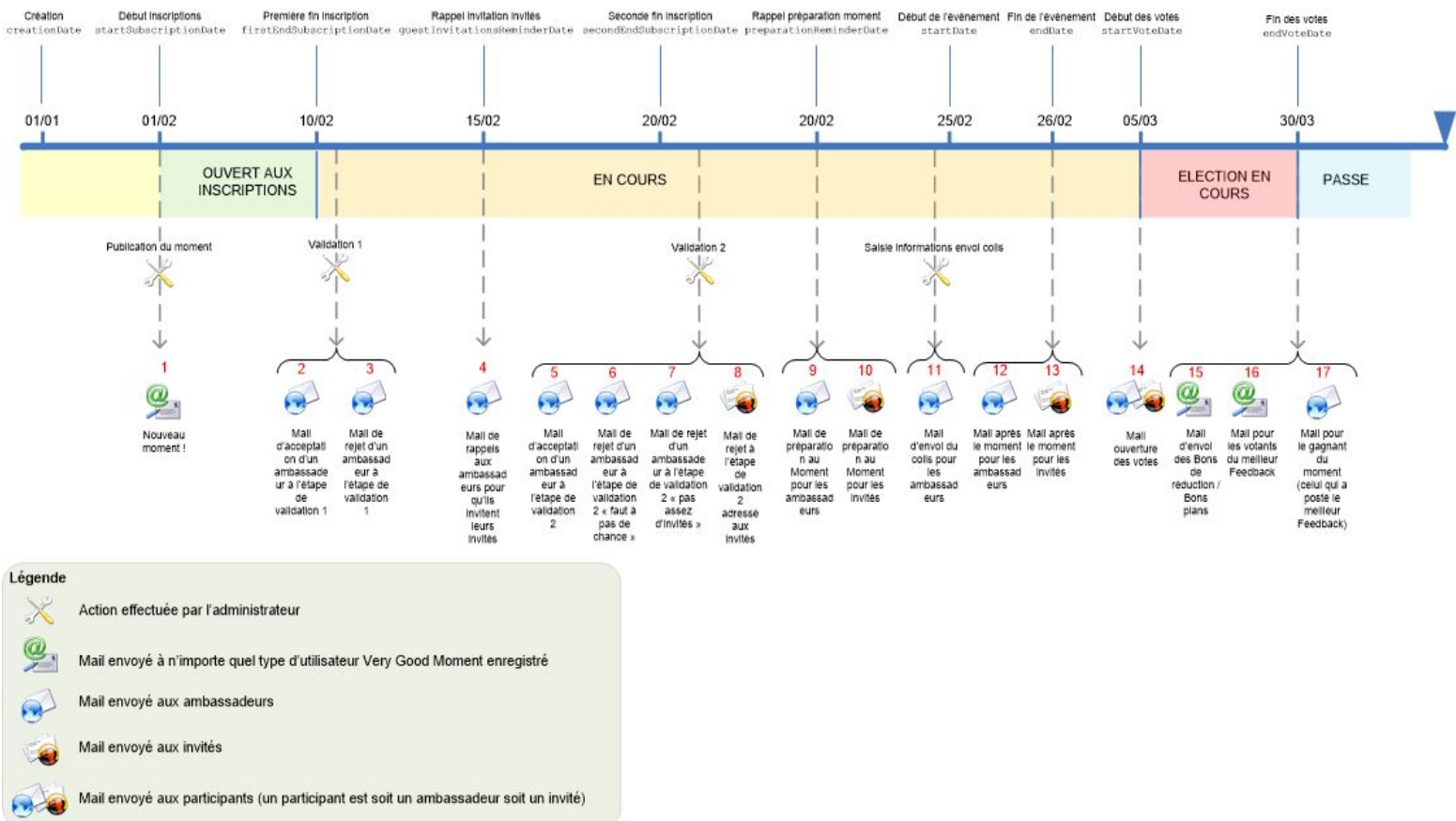


Figure 1 - Vie d'un moment

Comme on peut le constater sur le schéma, des emails sont envoyés lors de moments clés de la "vie" d'un évènement. Par exemple un mail de validation de sélection à un moment est envoyé à l'ambassadeur pour confirmer sa participation à l'étape suivante. Plusieurs actions doivent être effectuées par un administrateur : la mise en ligne du moment, la sélection des ambassadeurs, la sélection des utilisateurs (ambassadeurs + invités) qui recevront un colis, et la saisie des informations sur le colis.

Plus l'utilisateur participe à des moments, plus il gagne des points et augmente son niveau, ce qui lui permettra d'être sélectionné en priorité par la marque. Une fonction de recherche des Very Best People est disponible sur le site web de manière à donner la possibilité aux utilisateurs de consulter le profil de ses amis, par exemple, ou des meilleurs Very Best People.

3. Cahier des charges

3.1. Présentation générale

Enoncé du besoin :

Réponse à la problématique sous forme de plateforme web intégrable. Cette plateforme doit mettre à disposition de l'utilisateur des fonctionnalités lui permettant de réaliser les mêmes actions que celles déjà disponibles sur le site internet.

L'application devra également implémenter un design 'Responsive'. Ceci implique que la conception de l'application vise, grâce à différents principes et technologies, à offrir une expérience de consultation confortable même pour des supports différents. L'utilisateur peut ainsi consulter le même site web à travers une large gamme d'appareils (moniteurs d'ordinateur, smartphones, tablettes, TV, etc.) avec le même confort visuel et sans avoir recours au défilement horizontal ou au zoom avant/arrière sur les appareils tactiles notamment, manipulations qui peuvent parfois dégrader l'expérience utilisateur, tant en lecture qu'en navigation.

Environnement de développement :

- Ordinateur portable acer (Disque dur SSD, Système d'exploitation : Windows 8)
- Smartphone Sony Xperia Go avec Android 4.1.2
- Php 5.0
- IDE : Eclipse
- Test : Serveur local WAMP
- Serveur de production APACHE
- Client MySQL Workbench
- Data storage : Amazon Web Services

Contraintes :

- Temps de developpement : le développement de l'application démarre le 15 Mars 2015, et se termine le 31 Aout 2015.
- Performance du résultat : l'application finale devra être rapide à l'exécution.
- L'interface de l'application doit être "responsive".
- Exécution sur tous les navigateurs du marché : l'application web devra avoir un rendu uniforme sur les différents navigateurs existants tels que google chrome, mozilla firefox, internet explorer, opéra, ...

3.2. Expression fonctionnelle du besoin

Cadre d'utilisation :

Le cadre d'utilisation de l'application se situe dans le secteur de la publicité et plus précisément dans le "Home Marketing". L'application web s'adresse à plusieurs types d'utilisateurs : les premiers utilisateurs concernés par notre application seront les visiteurs (des particuliers) qu'ils soient connectés (Very Best People) ou non. Ensuite viennent les administrateurs de Very Good Moment qui sont chargés de gérer le contenu de l'application. Et enfin, les marques (des professionnels) qui participent et proposent des moments.

L'application doit être accessible depuis n'importe quel terminal, que ce soit sur ordinateur fixe, portable ou smartphone (à condition que l'utilisateur dispose d'un navigateur internet).

Fonction d'usage :

L'application web doit proposer une alternative moderne au site web existant de Very Good Moment. C'est donc une interface graphique web qui doit permettre aux différents utilisateurs d'effectuer les mêmes actions que sur le site internet. C'est-à-dire la saisie d'un moment, la recherche d'autres utilisateurs, la visualisation et l'upload (téléchargement côté serveur) de photos et vidéos, etc...

Fonctions de service :

L'application finale devra comporter les fonctionnalités suivantes :

- Connexion via identification par un login et un mot de passe.
- Authentification d'un utilisateur quelconque et d'un administrateur.
- Gestion du transfert (upload) et du stockage des photos et vidéos des utilisateurs.
- Gestion de l'affichage des photos et vidéos des utilisateurs sous forme de patchwork trié aléatoirement ou par score.
- Gestion d'un système de commentaires pour les photos et vidéos.
- Gestion des notifications (simples et push) lors de la mise en ligne d'un nouveau moment, d'un nouveau commentaire et d'une invitation à un événement.
- Gestion d'une "timeline" : affichage de contenu spécifique à un ambassadeur sur un moment en fonction de l'état du Moment
- Recherche d'utilisateurs (les Very Best People) et affichage de leur profil publique.
- Envoi automatique d'emails aux clients.
- Partage sur les réseaux sociaux avec intégration de widgets Facebook et Twitter.

Critères d'appréciation :

- La rapidité de l'exécution de l'application est un des critères qui rends le projet effectif et opérationnel. Par exemple, l'application finale ne devra pas "freezer" ou avoir des temps de chargement trop lourd à supporter pour l'utilisateur.
- Le rendu visuel devra être similaire au site web existant de façon à ne pas destabiliser l'utilisateur (charte graphique identique).
- La prise en main de l'application par un nouvel utilisateur ne devra pas être trop fastidieuse. Elle devra donc disposer d'une bonne ergonomie.

3.3. Cadre de réponse

Perspective d'evolution technologique :

Par la suite, de nouvelles fonctionnalités spécifiques pourront être ajoutés comme les notifications à chaque fois qu'un moment change d'état par exemple.

Selon l'avancement et le succès de Very Good Moment il faudra sûrement prendre en compte le problème que pose le "Big Data" lorsque le volume de données deviendra trop conséquent.

Evolution de l'architecture système (base de données, serveurs, etc.) pour mieux répondre à l'accroissement de la fréquentation de la plateforme sur les différents supports.

4. Organisation

4.1. Formation

Lors de mon arrivée dans l'entreprise, mon maître de stage m'a affecté des petites tâches de manière à pouvoir m'imprégner des différentes technologies utilisées. Ces tâches consistaient en la résolution de bugs de complexité croissante, et des évolutions mineures de l'interface graphique du site internet et de l'application mobile. Ces petits travaux m'ont permis de découvrir de nouveaux outils de programmation très utiles, comme "marionette js" (voir page 46), "grunt" ou "bower" (voir page 47); mais également de me familiariser avec les différentes briques logicielles qui constituent le projet Very Good Moment.

J'ai dû aussi m'habituer aux méthodes de travail de l'entreprise telles que la méthode agile et les réunions "scrums" que je détaille dans la partie 4.3. Puis, j'ai dû réaliser un système de commentaires avec états (en attente de validation, accepté, rejeté, supprimé) ce qui m'a permis de vraiment plonger dans la conception et la réalisation d'une fonctionnalité à part entière.

4.2. Outils

Dans cette section je vais vous énumérer les différents outils qui m'ont aidé dans l'organisation du travail et la gestion de projet :

Git est un logiciel de gestion de versions décentralisées. Il est conçu pour être efficace tant avec les petits projets, que les plus importants. Git fonctionne de façon décentralisée, c'est-à-dire que le développement ne se fait pas sur un serveur centralisé, mais chaque personne peut développer sur son propre dépôt. Git facilite ensuite la fusion (merge) des différents dépôts.

Dans le cas de l'entreprise Gomoob, la plupart du temps, pour chaque projet deux branches sont disponibles. Une branche appelée "develop" dans laquelle sont effectuées les modifications (comprendre développement). Et une autre branche "master" dans laquelle sont poussés les changements une fois effectués. L'utilisation de Git s'est effectuée principalement en ligne de commande windows.

Bitbucket est un outil de gestion de projet flexible et évolutif pour les équipes de développement logiciel et qui fournit en particulier la gestion des sources avec Git.

Google Drive m'a permis de stocker, mettre en commun et synchroniser de nombreux fichiers tels que les maquettes réalisées pour le client, des fichiers de configuration, des images et autres.

Gmail est le client de messagerie utilisé par l'entreprise. Dès mon arrivée dans les locaux une adresse personnelle à été créée et m'a été affectée : bastien.vanderchmitt@gomoob.com

4.3. Méthode Agile

Dans le cadre de la gestion de projet, l'entreprise Gomoob a fait le choix de la planification de réunions '**scrum**' tous les matins afin de mieux structurer les tâches. Ces réunions consistent à communiquer ce que tous les intervenants ont réalisé la veille, si ils ont eu des problèmes ou bugs notables, et ce qu'ils comptent réaliser dans la journée. Cette méthode agile permet une meilleure synchronisation des tâches entre les différents membres de l'équipe. En effet, de cette manière il est plus facile de travailler sur des ressources communes et d'éviter de répéter les mêmes erreurs.

Dans le cadre de cette méthode agile, l'entreprise à également mise en place la création de tâches quantifiées et chronologiquement dépendantes grâce à un logiciel se nommant **Jira**. Associé à l'utilitaire **Zend-desk**, le client peut directement créer une tâche qui sera automatiquement affectée dans jira et visible par les développeurs afin d'être traitée. De plus, BitBucket permet de lier directement un commit de Git à un bug JIRA ou une tâche de développement ce qui permet de voir rapidement tout le processus de traitement d'un bug, d'une évolution ou d'une tâche.

4.4. Planification

Taches	Mars	Avril	Mai	Juin	Juillet	Aout
Formation						
Conception						
Maquettes						
Services Php						
Web services						
Front End						
Rapport						

Figure 2 - Diagramme de Gantt

5. Conception

Afin de répondre au mieux à la problématique, il faut modéliser le besoin du client et notamment les données que nous aurons à manipuler : c'est la phase de conception. Dans cette partie sont exposés les principaux modèles de données réalisés dans cette perspective. A savoir : les modèles pour un évènement et pour un moment.

5.1. Analyse

5.1.1. Modèle Conceptuel des Données

Modèle de données pour un évènement (event) :

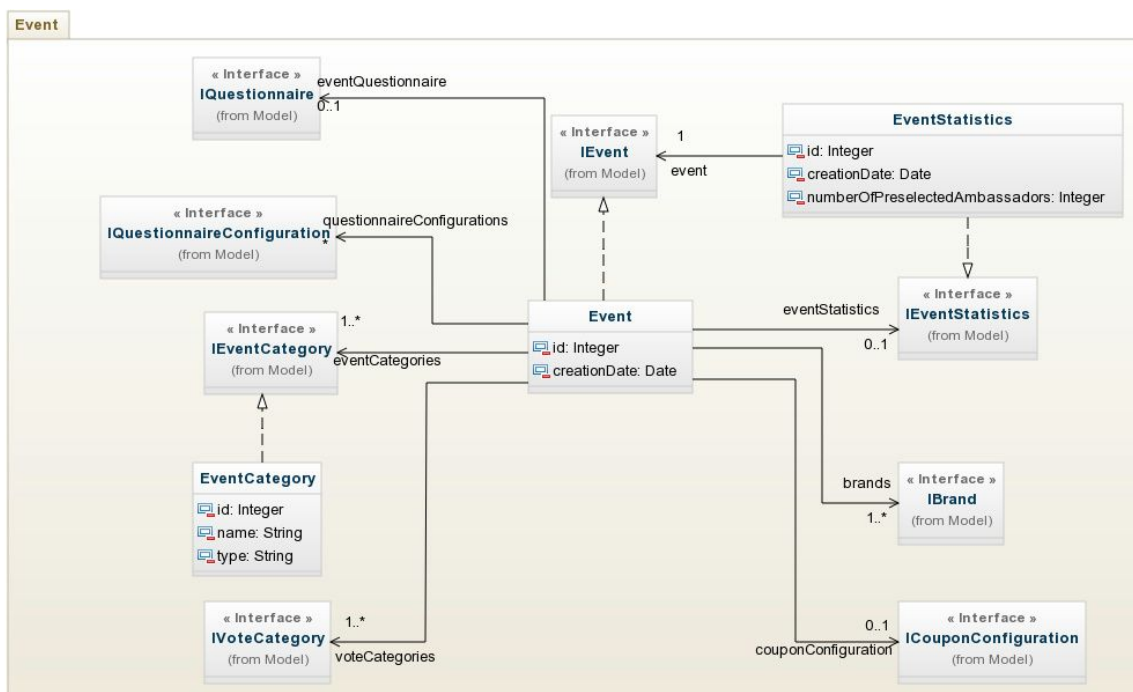


Figure 3 - MCD Event

Ce schéma montre qu'un évènement est rattaché à une ou plusieurs marques (brand). Chaque évènement hérite de l'interface IEvent et possède un identifiant "id" ainsi qu'une date de creation "creationDate".

Un évènement possède de nombreux autres attributs qui sont ici masqués pour un souci de visibilité. Un évènement est également rattaché à une catégorie, un questionnaire, une configuration de questionnaire, une catégorie de votes et finalement à des statistiques.

Le modèle conceptuel des données complet est disponible en annexe.

Modèle de données pour un moment (party) :

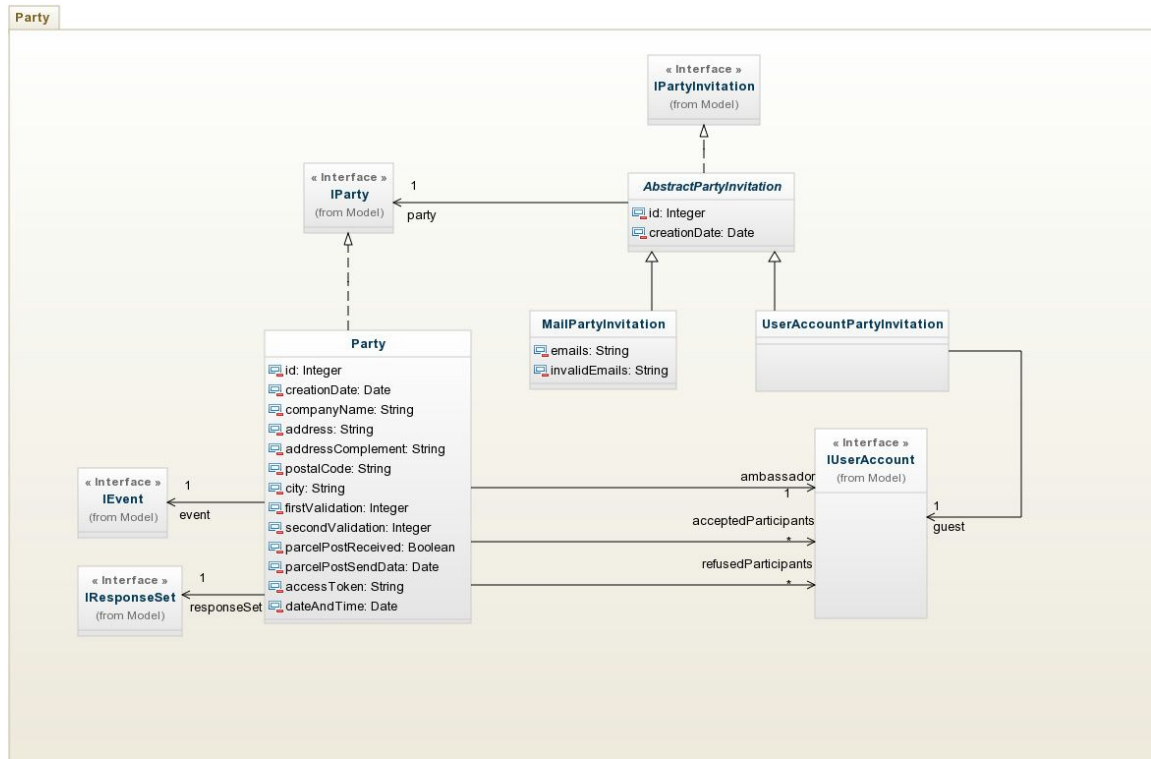


Figure 4 - MCD Party

Ce schéma montre qu'un moment est lié à un seul et unique évènement et possède en attribut toutes les informations utiles au déroulement d'un moment. Un moment est également associé à une interface IUserAccount représentant un utilisateur de l'application. Un moment possède ainsi un seul et unique ambassadeur, plusieurs participants ayant acceptés une invitation et plusieurs participants ayant refusés une invitation de l'ambassadeur. Ces contraintes sont représentés respectivement par les liaisons "ambassador", "acceptedParticipants" et "refusedParticipants".

La modélisation d'une invitation à un moment se fait par l'intermédiaire d'une classe abstraite "AbstractPartyInvitation", qui hérite d'une interface "IPartyInvitation" et de deux classes "MailPartyInvitation" et "UserAccountPartyInvitation" héritant de cette classe abstraite. La première classe instanciable représente une invitation par mail tandis que la deuxième représente une invitation directement par Very Good Moment. En effet, comme on peut le voir sur le schéma, chaque invitation de ce type est associé à une interface de type "invité" (guest).

5.1.2. Modèle Physique des Données

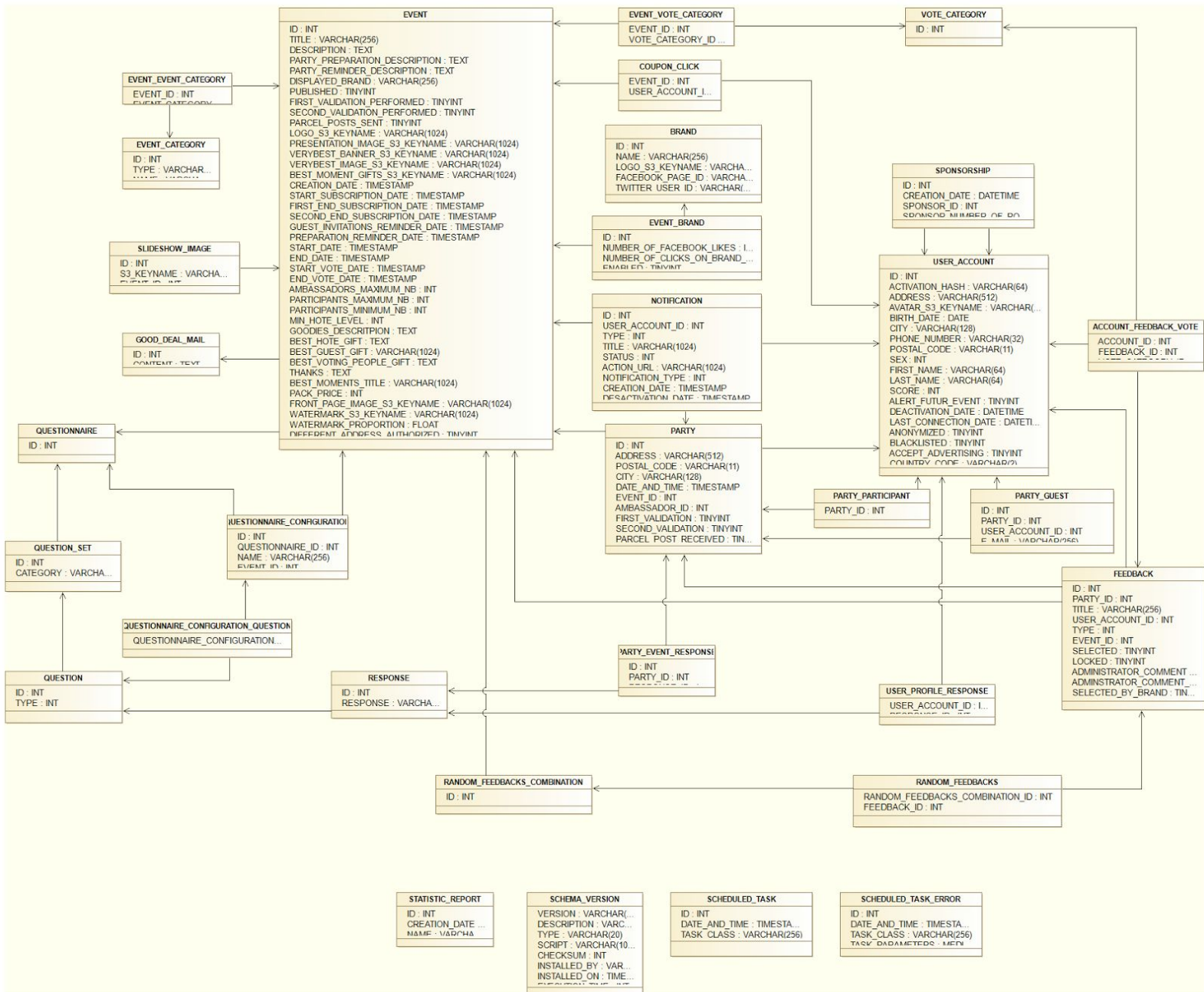


Figure 5 - Modèle Physique des Données

Le modèle physique de données représente les tables et leurs attributs telles qu'elles sont stockées en base de données. On y retrouve les clefs primaires et étrangères qui permettent de gérer des relations entre plusieurs tables et garantissent la cohérence des données.

5.2. Architecture

Cette partie présente l'architecture mise en place sur les différents projets réalisés.

5.2.1. Couches logicielles

Le schéma suivant représente de manière générale et simple la structure de l'application.

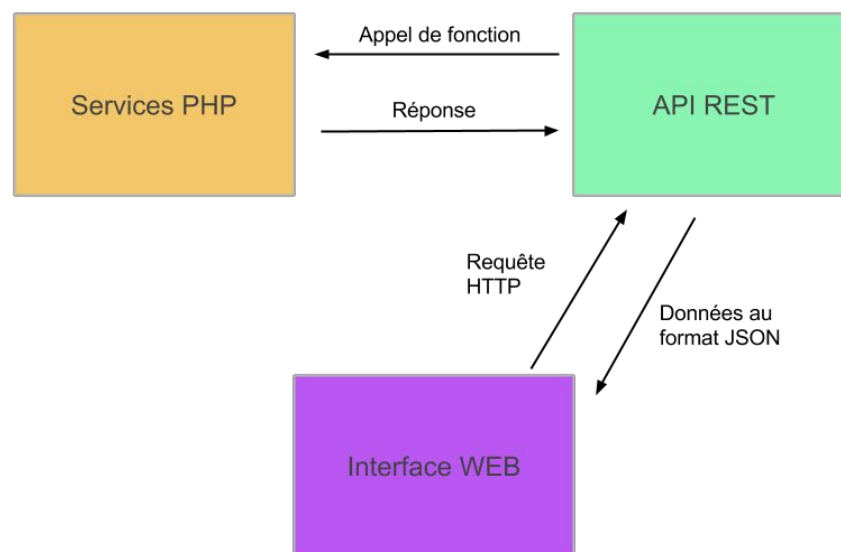


Figure 6 - Structure de l'application

La structure de l'application est donc constituée de la manière suivante :

- Services php (Serveur) : services appelés par l'API REST elle même sollicités par l'application cliente via une requête HTTP.
- API Rest (Serveur) : couche logicielle servant à intercepter les requêtes HTTP de l'interface graphique web cliente est d'assurer la communication avec les services PHP.
- Interface Web (Client) : application cliente emetteuse des requêtes HTTP dans l'attente des données demandées au format JSON.

Le diagramme de *package* suivant montre les différents *package* applicatifs créés côté serveur (code PHP) pour séparer les couches logicielles de l'application.

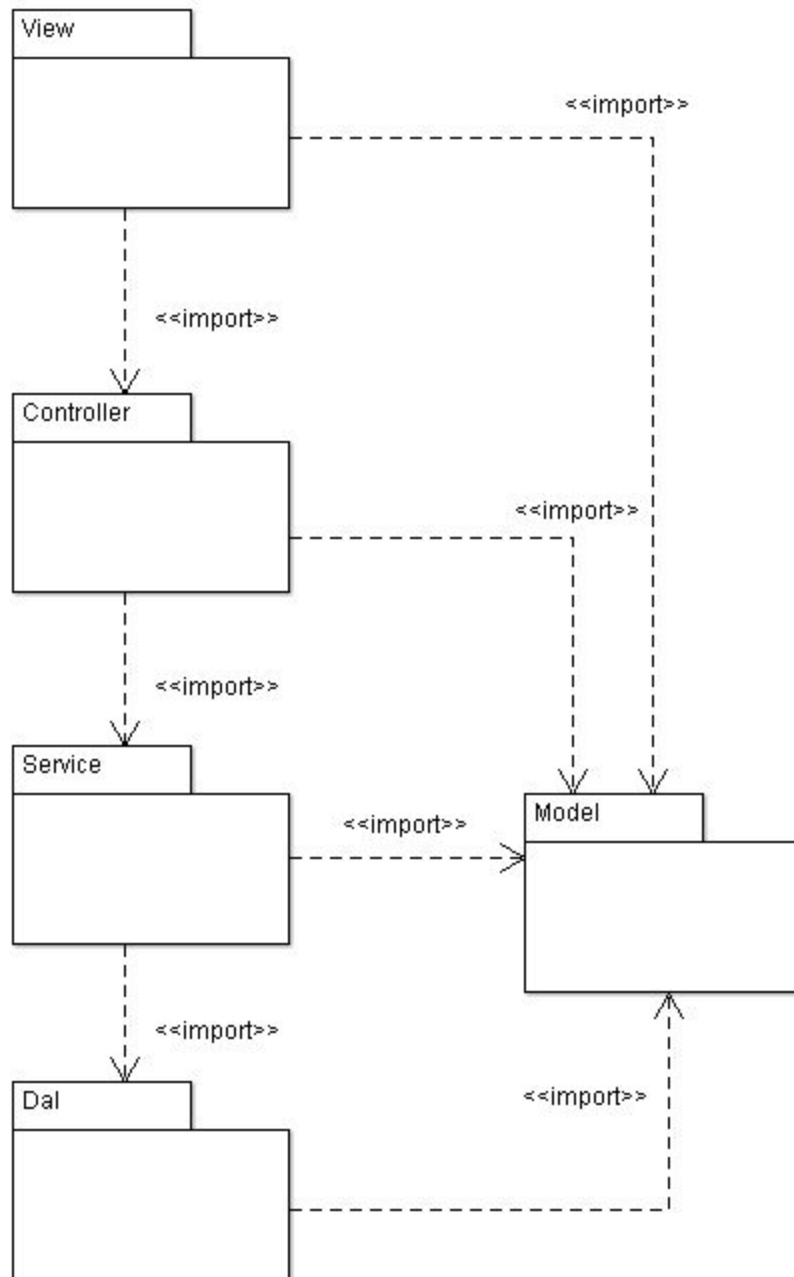


Figure 7 - Packages de l'application

L'application Web utilise un modèle MVC (Modèle Vue Controlleur). Un avantage apporté par ce modèle est la clarté de l'architecture qu'il impose. Cela simplifie la tâche du développeur qui tenterait d'effectuer une maintenance ou une amélioration sur le projet. La description de chacun des *packages* mis en oeuvre dans l'application est la suivante :

- View : Package contenant les vues de l'application. Ce *package* correspond à la partie Vue du modèle MVC.
- Controller : Package contenant les contrôleurs de l'application, les contrôleurs sont utilisés à la fois pour répondre aux requêtes de pages mais aussi pour exposer l'API REST de la plateforme. Ce *package* correspond à la partie Controlleur du modèle MVC.
- Service : Package correspondant à la couche de service de l'application (*Service Layer*). Les services apportent les fonctionnalités métier de l'application.
- Dal : Package représentant la couche d'accès aux données de l'application (*Data Access Layer*). Cette couche logicielle est responsable du *mapping* (correspondance) entre le modèle physique de la base de données et le modèle de données de l'application.
- Model : Package contenant les entités du modèle de données de l'application. Ce *package* correspond à la partie Modèle de l'architecture MVC.

5.2.2. Services Php

Ce chapitre met en place la description des services logiciels utilisés sur la plateforme Very Good Moment. Chaque service est défini par une interface et à chaque interface correspond son implémentation.

- **Package Dal (Data mappers et Row mappers)** : Cette couche logicielle est responsable du *mapping* entre le modèle physique de la base de données et le modèle de données de l'application.
- **Package Model** : *package* contenant les entités du modèle de données de l'application.
- **Package Service** : *package* contenant les interfaces et implémentations des services, c'est à dire actions sur les modèles (par exemple création, modification, suppression, ...) mais aussi les règles métiers spécifiques à Very Good Moment.

Prenons l'exemple du service dédié aux commentaire "CommentService". Parmi les fonctions dont dispose ce service nous allons nous intéresser plus particulièrement à celle qui permet de récupérer tous les commentaires pour un utilisateur et un *feedback* (comprendre photo ou vidéo) donnés que voici :

```
/**
 * {@inheritdoc}
 */
public function fetchAllByUserAccountAndFeedback($userId, $feedbackId, $offset = 0, $pageSize = 12, $embeddedResources = array())
{
    $comments = $this -> commentDataMapper -> fetchAllByUserAccountAndFeedback($userId, $feedbackId, $offset, $pageSize);

    if ($comments !== null && count($embeddedResources) > 0) {
        foreach ($comments as $comment) {
            $this -> _fillEmbeddedResources($comment, $embeddedResources);
        }
    }

    return $comments;
}
```

On remarque que cette fonction appelle la fonction du même nom implémentée dans le "CommentDataMapper" afin de récupérer les commentaires recherchés. Puis, si des commentaires ont été trouvés, le service associe à chacun d'eux des ressources embarqués appelés "embeddedResources". Ces ressources servent à récupérer des informations complémentaires.

Penchons nous maintenant sur le cas du Data Mapper.

```

* @param Application\Model\UserAccount $user the connected user.
* @param int $feedback_id the id of the feedback associated.
* @param int $offset (optionnal) an offset used to paginate the results.
* @param int $pageSize (optionnal) a page size used to paginate the results.
*
* @return [\Application\Model\IComment] the resulting comments.
*/
function fetchAllByUserAccountAndFeedback($userId, $feedbackId, $offset = 0, $pageSize = 12){

    // Checks that ids of user and feedback associated to the comment are not null
    ValidationUtils::isNotNullInteger('$userId', $userId);
    ValidationUtils::isNotNullInteger('$feedbackId', $feedbackId);

    // Comments with WAITING state associated to the user and feedback
    $query1 = 'select c1.* from comment c1
              where c1.user_account_id = ?' .
              ' and c1.feedback_id = ?' .
              ' and c1.state = "' . CommentState::WAITING .
              '" group by c1.id';

    // Comments with ACCEPTED state associated to the feedback
    $query2 = 'select c2.* from comment c2
              where c2.feedback_id = ?' .
              ' and c2.state = "' . CommentState::ACCEPTED .
              '" group by c2.id';

    // Apply the union between query 1 and 2
    $query = $query1 .' union ' .$query2;

    // Apply order by date of creation and apply pagination
    $query .= ' order by creation_date desc
              limit ' . $offset . ', ' . $pageSize;

    $rowSet = $this -> tableGateway -> getAdapter() -> createStatement(
        $query, array($userId,$feedbackId,$feedbackId) -> execute());

    return $this -> toEntities($rowSet);
}

```

Comme on peut le voir, le "CommentDataMapper" et les Data Mappers en général, s'occupent de créer la requête SQL correspondante aux données recherchés. Dans ce cas, la fonction "fetchAllByUserAccountAndFeedback" fait l'union entre les commentaires en attente de validation par un administrateur et les commentaires acceptés tout deux associés à un utilisateur et un feedback, puis rajoute ensuite un tri en fonction de la date de création décroissante du commentaire.

Puis, cette requête est exécutée et le résultat retourné est envoyé à la fonction "toEntities" qui appelle le Row Mapper qui se charge de convertir les données reçues en instances d'un modèle préalablement créé. Dans ce cas, le modèle ainsi récupéré est un "FeedbackComment" qui hérite d'une classe abstraite "AbstractComment". Le choix d'implémenter un héritage ici a été fait dans un souci d'évolution de l'application dans le cas où le client aurait besoin d'avoir d'autres types de commentaires (comme des commentaires du réseau social Facebook par exemple).

5.2.3. Services REST

REST (Representational State Transfer) est un style d'architecture qui repose sur le protocole HTTP. On accède à une ressource par son URI unique pour procéder à diverses opérations supportées nativement par HTTP :

- GET pour la lecture
- POST pour l'écriture
- PUT pour la modification
- DELETE pour la suppression

Un des plus gros avantages des services REST réside dans le fait qu'ils permettent de réaliser plusieurs applications clientes pour un même serveur. En effet, l'absence de gestion d'état du client sur le serveur conduit à une plus grande indépendance entre le client et le serveur. Dans le cas de Very Good Moment c'est un critère majeur qui nous a décidé à choisir cette architecture serveur au vu du fait qu'une Application Mobile et une Application Web RIA sont en développement.

Cette architecture permet également de ne pas avoir à maintenir une connexion permanente entre le client et le serveur. Le serveur peut ainsi répondre à d'autres requêtes venant d'autres clients sans saturer l'ensemble de ses ports de communication. Cela devient essentiel dans un système distribué. L'absence de gestion d'état du client sur le serveur permet également une répartition des requêtes sur plusieurs serveurs : une session client n'est pas à maintenir sur un serveur en particulier, ou à propager sur tous les serveurs. Cela permet aussi une meilleure évolutivité et tolérance aux pannes (un serveur peut être ajouté facilement pour augmenter la capacité de traitement, ou pour en remplacer un autre).

Utilitaires et librairies :

Afin de créer les différentes ressources de manière à être facilement compréhensibles et réutilisables nous avons utilisés les outils et standards suivants :

- **Hal** : (Hypertext Application Language) c'est un format simple qui permet de créer des liens hypertexte de manière cohérente et facile entre les ressources dans l'API.
- **Apigility** : c'est un outil de création d'API REST grâce à une interface graphique web. Dans notre cas, il est utilisé pour créer les fichiers de base afin de commencer le développement et la "personnalisation" des web services REST.
- **Postman** : c'est une extension du navigateur web Google Chrome qui permet de tester les web services ainsi créés. On peut choisir le type de requête, les données à envoyer et tester la requête avec un affichage du résultat qui en découle. Cet outil garde également en mémoire l'historique des requêtes testées.

Fonctionnement :

- **Hydrator** : chaque ressource dispose d'un hydrator qui se charge de convertir les données au format JSON en tableaux et vice-versa.
- **ResourceFactory** : fabrique (ou *factory*) permettant de créer un objet ressource dédié à une ressource REST spécifique.
- **Resource** : cette classe permet de spécifier quelles actions effectuer pour chaque méthode (create, fetch, delete, ...) d'une ressource. C'est ici que seront appelés les services PHP correspondants.

Ressources de type collection :

Ce chapitre permet de décrire les comportements communs à toutes les ressources REST de type collection. La taille des pages utilisée pour la navigation dans les collections définies dans cette spécification est égale à 12 par défaut. 12 est un multiple de 2, 3, 4 et 6, ceci permet l'affichage de listes à 2 ou 3 colonnes aisément. Les représentations JSON des collections définies dans ce document sont associées à des ensembles de liens dédiés à la pagination. Ces liens sont déclarés dans la section `_links` (standard HAL) des représentations JSON des collections.

La description de ces liens de pagination est la suivante :

- **first** : Lien vers la première page de la collection.
- **prev** : Lien vers la page suivante de la collection (présent uniquement si une page précédente existe).
- **next** : Lien vers la page précédente de la collection (présent uniquement si une page précédente existe).
- **last** : Lien vers la dernière page de la collection.

Paramètres de pagination :

Par défaut la taille des pages de résultats dans les ressources de collection doit être égale à 12. Afin de permettre aux applications clientes de choisir des tailles de pages différentes les requêtes associées aux collections doivent pouvoir utiliser les paramètres d'URL suivants :

- **page** : L'offset de la page de résultat à récupérer (commence à 1), si ce paramètre n'est pas défini explicitement sa valeur implicite est égale à 1.
- **page_size** : La taille en nombre de résultats de la page à récupérer, si ce paramètre n'est pas défini explicitement sa valeur implicite est égale à 12. Si la valeur des paramètres "page" et "page_size" fournie est négative alors le serveur doit retourner une erreur HTTP 400 « Bad Request ». Si la valeur du paramètre 'page_size' est supérieure à 100 alors le serveur doit faire de même afin de ne pas surcharger.

Exemple de réponse JSON :

L'exemple suivant est une réponse à une requête de type POST qui sert à la création d'un commentaire. L'url de la requête est la suivante : <https://staging.gomoob.com/vgm/rest/comments/2>

La réponse à cette requête confirme la création du commentaire dont les données sont passées et envoyés au format JSON. Cette réponse est la suivante :

```
{
  "id": 2,
  "creationDate": "2015-02-14T09:00:00+0100",
  "message": "Ceci est un test de commentaire.",
  "state": "accepted",
  "_links": {
    "self": {
      "href": "https://staging.gomoob.com/vgm/rest/comments/2"
    },
    "userAccount": {
      "href": "https://staging.gomoob.com/vgm/rest/user_accounts/3"
    },
    "feedback": {
      "href": "https://staging.gomoob.com/vgm/rest/feedbacks/1300"
    }
  }
}
```

Figure 8 - Exemple de réponse JSON

5.2.4. Interface Web

Après connexion à la plateforme les utilisateurs sont « redirigés » vers une application RIA basée notamment sur le *framework* Backbone JS. Une couche graphique RIA est créée car elle procure plusieurs avantages :

- Les navigateurs Web peuvent mettre en cache un maximum de code, ceci inclut le code qui permet la génération des vues.
- Etant donné que la génération des vues est faite essentiellement côté client ceci à l'avantage d'alléger le serveur au maximum.
- Le réseau déployé sur le terrain lors du déroulement d'un événement peut entraîner des lenteurs, principalement en raison du nombre d'utilisateurs connectés à la fois aux mêmes antennes Wifi mais aussi aux mêmes antennes 3G / H+ / 4G. La communication entre la couche graphique RIA et le serveur se fait via des données en JSON beaucoup plus légères qu'une page HTML complète. Ainsi les problèmes de lenteur du réseau peuvent être mieux supportés qu'avec une génération de pages HTML classique côté serveur.

L'architecture côté client de l'application est représentée sur le schéma suivant.



Figure 9 - Architecture client

Les différentes briques de ce schéma sont les suivantes :

- **RequireJS** Loaders de modules AMD (*Asynchronous Module Definition*), l'application est séparée en modules Javascript chargés de manière asynchrones. Ceci permet d'accroître les performances de chargement et de séparer clairement les responsabilités dans le code. RequireJS est aussi utilisé pour minifier le code source Javascript et charger des fichiers de ressources côté client (fichiers de traductions, fichiers de *templates* de vues, etc.) ;
- **Twitter Bootstrap** Bibliothèque de composants graphiques utilisés dans l'application ;
- **jQuery** Framework de manipulation du DOM dans les pages HTML
- **UnderscoreJS** framework permettant la gestion de *templates* de vues. Ce *Framework* apporte aussi un grand nombre de méthodes utilitaires pour la gestion des tableaux et collections ;
- **BackboneJS** Framework proche d'un *Framework* MVC (Modèle Vue Contrôleur), il permet de séparer l'application grâce à une couche modèle, vue et un ensemble d'autres composants. Le *router* permettant de rediriger une requête vers une vue précise ;
- **Marionette JS** Framework apportant un cadre applicatif au-dessus du Framework Backbone JS. Marionette JS est un *framework* conçu pour les « Large Javascript Applications » et est particulièrement adapté aux interfaces graphiques RIA (*Rich Internet Application*) complexes.

6. Réalisation

6.1. Méthodologie




Après la phase de conception, nous entrons dans l'étape de la réalisation de maquettes, parfois plusieurs pour une même page, qui sont soumises au client pour validation. Pour ce faire, l'entreprise utilise le logiciel MockFlow.

MockFlow WireframePro est un outil basé sur le Web pour concevoir des plans ainsi que des maquettes et collaborer sur des interfaces utilisateur pour des sites web et des applications . Il aide à visualiser l'interface graphique, la navigation et la structure du site web en peu de temps.

Une fois la maquette validée par le client, le développement de la page en question peut commencer.

Exemple : la maquette ci-contre représente la Timeline d'un ambassadeur pour un Moment à l'étape de la sélection et de la mise en ligne des photos et vidéos prises lors de l'évènement en mode "desktop".

☰
⋮
Party Game en Fête




Moment proposé par **Asmodee**

Le **01.05.2015**

Pour **20** hôtes de marque et **10 à 15** invités entre 18 et 30 ans.

ETAPE 4 - PHOTOS / VIDEOS
 INFOS UTILES **A GAGNER**

ETAPE




PHOTOS/VIDEOS

LES PHOTOS/VIDEOS DE MES INVITES

<div style="border: 1px solid #ccc; padding: 5px; text-align: center;">VIDEO</div> <p style="font-size: small;">Moment-passé-toto-2 Toto Legrand</p> <p style="background-color: #FFC000; color: white; text-align: center; padding: 2px;">SELECTIONNER</p>	<div style="border: 1px solid #ccc; padding: 5px; text-align: center;">IMAGE</div> <p style="font-size: small;">Moment-passé-toto-2 Toto Legrand</p> <p style="background-color: #FFC000; color: white; text-align: center; padding: 2px;">SELECTIONNER</p>	<div style="border: 1px solid #ccc; padding: 5px; text-align: center;">IMAGE</div> <p style="font-size: small;">Moment-passé-toto-4 Toto Legrand</p> <p style="background-color: #00A0C0; color: white; text-align: center; padding: 2px;">DESELECTIONNER</p>
<div style="border: 1px solid #ccc; padding: 5px; text-align: center;">IMAGE</div> <p style="font-size: small;">Moment-passé-toto-3 Toto Legrand</p> <p style="background-color: #FFC000; color: white; text-align: center; padding: 2px;">SELECTIONNER</p>	<div style="border: 1px solid #ccc; padding: 5px; text-align: center;">IMAGE</div> <p style="font-size: small;">Moment-passé-toto-4 Toto Legrand</p> <p style="background-color: #00A0C0; color: white; text-align: center; padding: 2px;">DESELECTIONNER</p>	<div style="border: 1px solid #ccc; padding: 5px; text-align: center;">IMAGE</div> <p style="font-size: small;">Moment-passé-toto-3 Toto Legrand</p> <p style="background-color: #FFC000; color: white; text-align: center; padding: 2px;">SELECTIONNER</p>
<div style="border: 1px solid #ccc; padding: 5px; text-align: center;">IMAGE</div> <p style="font-size: small;">Moment-passé-dédé-1 Dédé Legros</p> <p style="background-color: #00A0C0; color: white; text-align: center; padding: 2px;">DESELECTIONNER</p> <p style="font-size: x-small; background-color: #ccc; padding: 2px;">Message administrateur</p>	<div style="border: 1px solid #ccc; padding: 5px; text-align: center;">IMAGE</div> <p style="font-size: small;">Moment-passé-titi-3 Titi Lepetit</p> <p style="background-color: #00A0C0; color: white; text-align: center; padding: 2px;">DESELECTIONNER</p>	

MES PHOTOS / VIDEOS AJOUT



Tu peux uploader jusqu'à **10 photo(s)** et **1 vidéo(s)** (taille maximum: 75Mo). Attention, si tu viens juste d'uploader une vidéo il se peut qu'un message t'indique que ta vidéo n'est pas disponible. Tu dois attendre quelques minutes pour que ta vidéo apparaisse.

AJOUTER MES PHOTOS/VIDEOS +

LISTE

<div style="border: 1px solid #ccc; padding: 5px; text-align: center;">VIDEO</div> <p style="font-size: small;">Moment-passé-toto-2 Toto Legrand</p> <p style="background-color: #FFC000; color: white; text-align: center; padding: 2px;">SELECTIONNER</p>	<div style="border: 1px solid #ccc; padding: 5px; text-align: center;">IMAGE</div> <p style="font-size: small;">Moment-passé-toto-2 Toto Legrand</p> <p style="background-color: #FFC000; color: white; text-align: center; padding: 2px;">SELECTIONNER</p>	<div style="border: 1px solid #ccc; padding: 5px; text-align: center;">IMAGE</div> <p style="font-size: small;">Moment-passé-toto-4 Toto Legrand</p> <p style="background-color: #00A0C0; color: white; text-align: center; padding: 2px;">DESELECTIONNER</p>
<div style="border: 1px solid #ccc; padding: 5px; text-align: center;">IMAGE</div> <p style="font-size: small;">Moment-passé-toto-3 Toto Legrand</p> <p style="background-color: #FFC000; color: white; text-align: center; padding: 2px;">SELECTIONNER</p>	<div style="border: 1px solid #ccc; padding: 5px; text-align: center;">IMAGE</div> <p style="font-size: small;">Moment-passé-toto-4 Toto Legrand</p> <p style="background-color: #00A0C0; color: white; text-align: center; padding: 2px;">DESELECTIONNER</p>	<div style="border: 1px solid #ccc; padding: 5px; text-align: center;">IMAGE</div> <p style="font-size: small;">Moment-passé-toto-3 Toto Legrand</p> <p style="background-color: #FFC000; color: white; text-align: center; padding: 2px;">SELECTIONNER</p>

MES INFORMATIONS

<p> 5 Photos / Vidéos sélectionnées Tu peux choisir jusqu'à 10 photos et 1 vidéo.</p>	<p> 12 Temps restant Il te reste 12 jours pour poster les meilleures photos/vidéos.</p>	<p>Cloture des votes le 15/10/2018 D'ici là toutes les photos/vidéos seront soumises au vote des internautes .</p>
<p> 6 Photos / Vidéos restantes Tu peux envoyer jusqu'à 10 photos</p>	<p> Information, seules les photos/vidéos sélectionnées à la date du 01/10/2014 seront publiées sur Very Good Moment. Jusqu'à cette date, tu peux modifier tes choix à ta convenance !</p>	

6.2. Développement

Les dirigeants de Gomoob ont choisi d'utiliser des technologies récentes et en constante évolution. C'est ce choix qui leur permet d'être une entreprise innovante et la pointe de la technologie. Parmi celles-ci, je vais vous décrire les principales librairies et frameworks que j'ai utilisés afin d'améliorer ma productivité et la performance du produit final.

6.2.1. Librairies et Frameworks



Node est une bibliothèque Javascript qui permet de construire facilement des applications de réseau rapides et évolutives. Concrètement, permet d'utiliser le langage JavaScript côté serveur. Il bénéficie ainsi de la puissance de JavaScript pour proposer une toute nouvelle façon de développer des sites web dynamiques. Node.js est donc idéal pour les applications en temps réel avec un gros volume de données comme celle de Very Good Moment.



Bower permet de gérer les dépendances Javascript d'un projet. Il dépend de Node et de NPM (Node Package Manager), l'outil permet de télécharger et installer facilement tous les modules de la communauté tout en gérant les dépendances. Il fonctionne en s'appuyant sur github pour télécharger les composants grâce à des tables de correspondances "name" : "url". On peut lui indiquer de forcer la version de la librairie si besoin est.



Grunt est un outil d'automatisation des tâches. Il permet de réduire l'effort nécessaire pour préparer un "build" en développement ou en production. Grunt peut réaliser la concaténation et la minification de JavaScript. Cet outil dispose d'une communauté active de développeurs, avec de nouveaux plugins qui sortent régulièrement ainsi qu'une vaste gamme d'outils et de tâches automatisées qui sont déjà disponibles et prêts à utiliser.



Backbone est une bibliothèque Javascript donne la structure des applications web en fournissant des modèles avec la liaison de clé-valeur et des événements personnalisés, des collections avec une API riche de fonctions énumérables , vues avec événement déclarative manipulation , et se connecte tout à votre API existante sur une interface RESTful JSON.



Marionette est une bibliothèque Javascript qui va sur le dessus de Backbone.js . Il permet de construire une application à grande échelle , l'introduction de nombreux modèles et de bonnes pratiques.



JSHint est un outil communautaire visant à détecter les erreurs et les problèmes potentiels dans le code JavaScript et de faire respecter les conventions de codage.



Twitter Bootstrap est une collection d'outils utile à la création de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript.



WampServer est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Il m'a permis de tester les développements effectués en local.



PHPUnit

PHPUnit est un framework de tests unitaires open source dédié au langage de programmation PHP. Cet outil m'a permis de tester chaque service Php crée avec des données de test au format "xml".

C'est un outil très important dans le sens où chaque service doit être testé pour un maximum de cas possibles de manière à ne pas avoir de mauvaises surprises plus tard et devoir se replonger dans un code parfois oublié et complexe.



Jenkins est un outil open source d'intégration continue. Écrit en Java, Jenkins fonctionne dans un conteneur de servlets tel qu' Apache Tomcat, ou en mode autonome avec son propre serveur Web embarqué. Un certain nombre de scripts existants permettent de réaliser un "build" de l'application web en mode développement ou production.

PHP-DI 5 Cette librairie permet l'injection automatique des dépendances php. Elle est principalement utilisée dans les projets php-services et web-services.



Composer est l'équivalent de Bower côté serveur. En effet, composer permet de gérer les dépendances PHP d'un projet. Concrètement, voici comment s'utilise Composer : On définit dans un fichier la liste des bibliothèques dont le projet dépend, ainsi que leur version; On exécute une commande pour installer ou mettre à jour ces bibliothèques (et leurs propres dépendances également) ; On inclut alors le fichier d'*autoload* généré par Composer dans notre projet.



Pushwoosh est un service web qui permet d'envoyer des notifications *push* à des terminaux mobiles de type smartphones. La création de notification se fait par une requête HTTP POST avec des données (message, terminaux à notifier et date/heure) au format JSON. Nous avons utilisé ce service pour l'envoi de notifications lors de la mise en place d'un nouveau moment, d'invitations à un évènement et de la création d'un nouveau commentaire.

6.2.2. Widgets

Lors du développement de l'application web, nous nous sommes aperçus que quelques éléments devraient apparaître de façon redondante dans l'application. Nous avons donc décidé de créer un projet annexe dans lequel nous réaliserons des widgets. Un widget est une collection d'éléments ayant pour but une fonctionnalité bien précise. Un widget est générique ce qui permet d'avoir moins de redondance dans le code de l'application principale. De plus, les widgets servent également à l'application mobile puisque celle-ci est similaire à l'application web. Les widgets doivent également disposer d'un design responsive pour s'adapter en fonction de la taille du conteneur.

A ce jour, les widgets existants sont :

- Champ de recherche (search-field)
- Bannière d'informations (banner-info)
- Roue de chargement (load-indicator)
- Very Best People : sert à afficher les informations d'un very best people.
- Timeline info : sert à afficher des informations en mettant un nombre en avant.
- Timeline step viewer : widget servant à montrer l'étape de la timeline en cours.
- Timeline guest info : permet d'afficher les informations d'un invité dans la timeline.

6.2.3. Exemple de la Timeline

Voici deux captures d'écrans de la timeline d'un ambassadeur en mode "Desktop" (largeur du conteneur > 768 pixels) et "Mobile" :

The screenshot shows a desktop view of a selection process timeline. At the top, there is a header with a menu icon, the text "MON MOMENT", and a share icon. The main heading is "SELECTION EN COURS". To the right, there are three informational items: "Moment proposé par ACER", "Le 01.09.2030", and "Pour 20 Hôtes de marque et 15 à 21 invité(e)s de tous âges". Below this is a progress bar with five steps, where step 1 is highlighted in orange and labeled "INSCRIPTIONS". A navigation bar contains three tabs: "ETAPE 1 - INSCRIPTIONS", "INFOS UTILES", and "A GAGNER". The main content area features a clock icon and the text "TES AMIS ET TOI ÊTES EN ATTENTE DE VALIDATION DÉFINITIVE !". Below this, a message states: "Un email te sera envoyé pour t'indiquer si tu fais partie de la liste définitive des heureux élus ! Il reste actuellement [objet Object] candidats pour 20 cols. Ton groupe d'amis et toi serez avertis par email du résultat des ultimes sélections. Bonne chance !". The interface then displays two categories of candidates: "Ils ont acceptés" and "Ils ont refusés".

Catégorie	Nom	Âge	Points
Ils ont acceptés	TITI	41 ANS	90 POINTS
	MBLANC	39 ANS	110 POINTS
Ils ont refusés	TLHERMITTE	35 ANS	150 POINTS

Figure 10 - Timeline en mode "Desktop"

Général :

Cette page est composée de plusieurs vues :

- Une vue représentant les informations liées au moment ainsi que l'image du moment.
- Une vue englobant les onglets.
- Une vue pour le contenu de chaque onglet.

Le chargement de cette page se fait en plusieurs étapes. Premièrement on récupère la "Party" associée à l'identifiant passé en paramètre de la requête de la page. Grâce aux données ainsi récupérées on détermine l'étape en cours requise pour instancier la vue contenant les onglets. On génère aussi à cet instant la vue représentant les informations du moment. La vue contenant les onglets se charge alors d'instancier le contenu de ceux-ci en fonction de l'état du moment (il y a plusieurs états dans une même étape de la timeline) à l'exception du dernier onglet qui reste inchangé lors des différentes étapes.

Mode desktop :

Sur la partie supérieure de l'écran on retrouve les informations liés au moment. Ici il s'agit d'un moment ayant passé les premières sélections et dont l'ambassadeur à déjà invité ses amis et attends d'être sélectionné définitivement. Un code couleur à été associé à chaque état d'un moment, ici c'est la fine bande orange.

L'utilisateur dispose ensuite de trois onglets : l'onglet "Infos Utiles" qui explique en détail les actions que celui-ci peut réaliser. L'onglet "A Gagner" qui récapitule les produits du présents dans le colis et sa valeur. Et enfin l'onglet principal sur lequel on effectue les actions. La frise située juste en dessous des onglets représentant l'étape courante de la Timeline résulte de l'instanciation du widget "TimelineStepViewer" et se retrouve sur chaque onglet de manière à toujours indiquer sa position à l'utilisateur.

Viens ensuite une bannière informative qui est aussi un widget paramétré avec le texte correspondant et un chemin vers une image. Finalement, une liste des amis ayant acceptés les invitations et une autres avec ceux ayant refusés l'invitation sont affichés. Chaque "fiche" représentant un invité est également un widget résumant les informations de celui-ci.

Mode mobile :

Sur la capture d'écran suivante on remarque que certains éléments ont été modifiés :

- Les informations du moment ont été enlevés et transférées dans l'onglet "Infos Utiles".
- Le nom des onglets à disparu en faveur d'un titre de page car il n'y a pas la place sur mobile pour afficher le nom entier dans l'onglet.
- Le widget représentant l'étape de la timeline s'est ajusté à la taille de l'écran.
- La disposition des éléments dans la bannière d'information à changé (l'image est maintenant centrée au dessus du texte et la taille s'est réduite).
- Les invités des listes sont dorénavant affichés les uns en dessous des autres alors qu'ils était représentés par lignes de trois en mode Desktop.

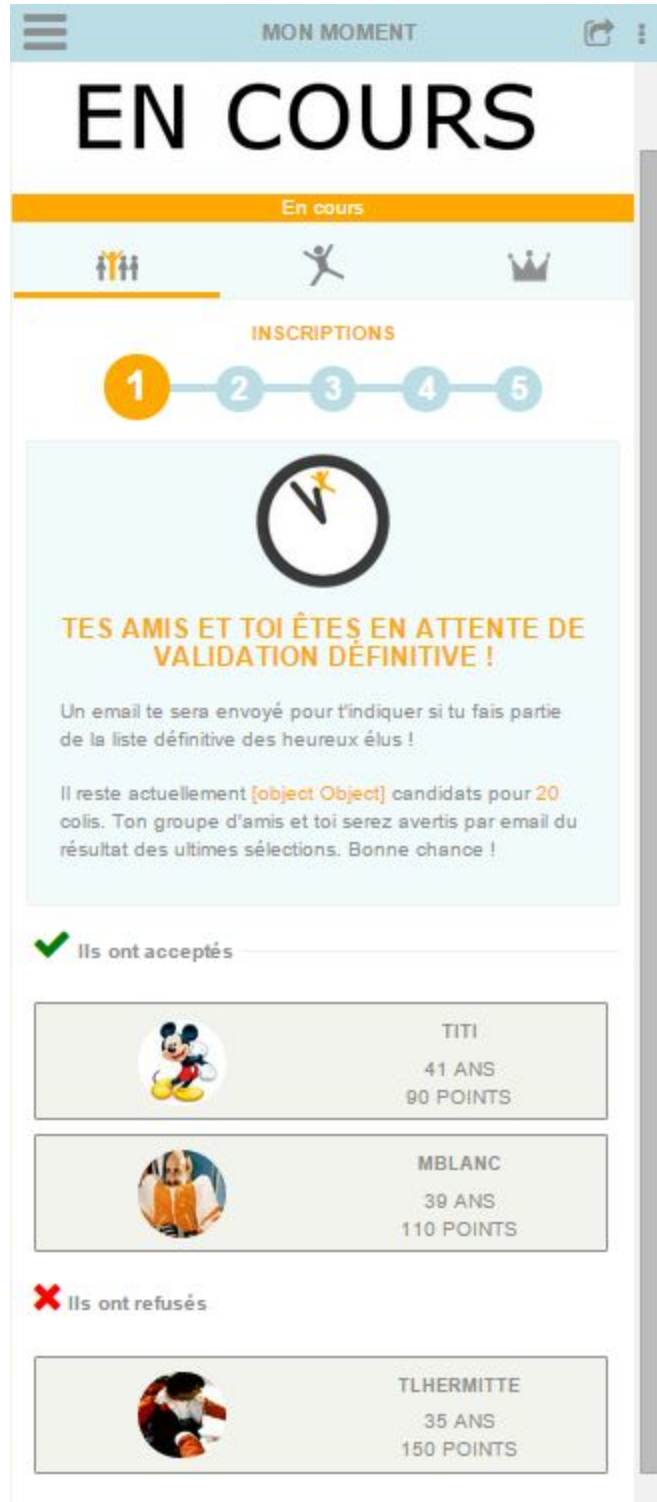


Figure 11 - Timeline en mode "Mobile"

7. Résultat et discussion

7.1. Application finale

Toutes les fonctionnalités ont bien été développées à temps. Deux grosses mises à jour de la production ont été réalisées en Juillet et Aout. Dans un premier temps nous avons réalisés la partie essentielle de l'application. C'est-à-dire la disposition des menus, l'identification d'un utilisateur, les pages d'informations (mentions légales, nous contacter, découvrir), et la saisie d'un moment. Puis nous avons travaillé sur l'affichage des very best people et l'implémentation des widgets very good moment. Enfin, durant le mois d'Aout nous avons développé la fonctionnalité de "Timeline" qui permet à un utilisateur de gérer un moment auquel il participe ou est invité en temps réel.

On peut voir des exemples de l'application dans deux conteneurs de sites internet différents. Le premier concerne un moment organisé par la marque Justin Bridou. Dans ce cas précis l'application web Very Good Moment est intégrée sur le réseau social Facebook : https://www.facebook.com/justinbridou/app_114497452235318

Le deuxième exemple de l'application web Very Good Moment se situe sur le site web du quotidien 20Minutes pour la marque de levures et farines Francine : <http://verygoodmoment.20minutes.fr/francine/cuisinez-en-couleur>

La fluidité et la rapidité de l'application sont au rendez-vous même si lors de grosses opérations comme le chargement de la Timeline de l'utilisateur le temps de réponse est un peu plus élevé (environ une seconde).

7.2. Problèmes rencontrés

L'entreprise a fait le choix judicieux d'utiliser beaucoup de bibliothèques modernes. Il a donc fallu s'habituer et s'adapter le plus vite possible à ces nouvelles technologies mais la plupart du temps la documentation de celles-ci était conséquente et permettait une adaptation rapide. Les connaissances acquises à la Faculté des Sciences lors de mon cursus universitaire comme les architectures logicielles m'ont été ici très utiles.

L'application web Very Good Moment est un gros projet avec un existant conséquent et une terminologie bien particulière. Il m'a fallu quelques semaines pour bien comprendre les différents principes et fonctionnalités à réaliser.

7.3. Apports à l'entreprise et personnel

Aide au développement pendant une période de croissance de l'entreprise. En effet, la société Gomoob est en plein essor et de nombreux projets affluent. Il était donc devenu indispensable pour les dirigeants de décharger un peu de leur tâches. J'espère avoir su combler leurs attentes et avoir été fiable quand au rôle qu'on m'a confié.

De mon côté j'ai appris de très nombreuses choses. A compter une expérience très enrichissante et non négligeable avec de nouvelles technologies telles que Backbone, Marionette, Bootstrap, Grunt, Jenkins, etc ... Ce stage m'a également permis d'avoir une meilleure vision d'ensemble sur le développement d'application web ou mobiles dans le sens où lors de la conception, le développeur doit vraiment penser à tout les cas possibles en passant de la base de donnée au rendu final. Enfin, j'ai pu découvrir certaines facettes de la création d'entreprise comme l'importance des relations et la bonne entente entre les différents acteurs (comprendre employés et employeurs) qui fût très facile. Un autre aspect de la "start-up" qui s'est porté à ma connaissance sont les tâches administratives comme la mise en place des tarifs et des devis par exemple.

8. Conclusion

Ce stage en entreprise fût l'occasion d'apprendre de nouveaux concepts comme l'intégration continue et de découvrir de nouvelles technologies comme pushwoosh, Marionette ou Bower. Je me suis aussi rendu compte de l'importance des tests unitaires pour ne pas avoir à corriger un problème une fois la tâche finie et pour avoir un produit fiable et robuste. Mon maître de stage m'a fait part de ses connaissances en matière de principes et de normes de programmation que je me suis appliqué à mettre en place. Ces bonnes pratiques m'ont permis de progresser sur l'efficacité et la rapidité de développement. Cependant, force m'a été d'accepter le fait qu'il me reste encore énormément à apprendre et à découvrir au vu des connaissances et de la motivation de mon maître de stage Simon Baudry.

Cette expérience m'a également permis de mieux comprendre les tenants et les aboutissants d'une application RIA/SPA. En effet, notre client Very Good Moment avait besoin d'une application web avec de nombreuses fonctionnalités qui requéraient de lourds traitements si ils avaient été effectués avec des technologies du web classiques. Heureusement, grâce à la solution de rassembler toutes ces nouvelles technologies, nous avons pu répondre à son besoin en lui fournissant une application riche et réactive. L'avenir des applications web en général et plus particulièrement les applications RIA se situe donc dans l'implémentation de bibliothèques toujours plus récentes et originales afin de réduire les traitements et accélérer l'échange de données.

Pour conclure, ce stage m'a permis à la fois de faire un grand pas vers le monde professionnel et à la fois de découvrir l'univers des start-up et de la création d'entreprise. Je vais ainsi pouvoir mettre à profit toutes les bonnes pratiques et connaissances acquises dans mes futurs emplois et projets.

ANNEXES :

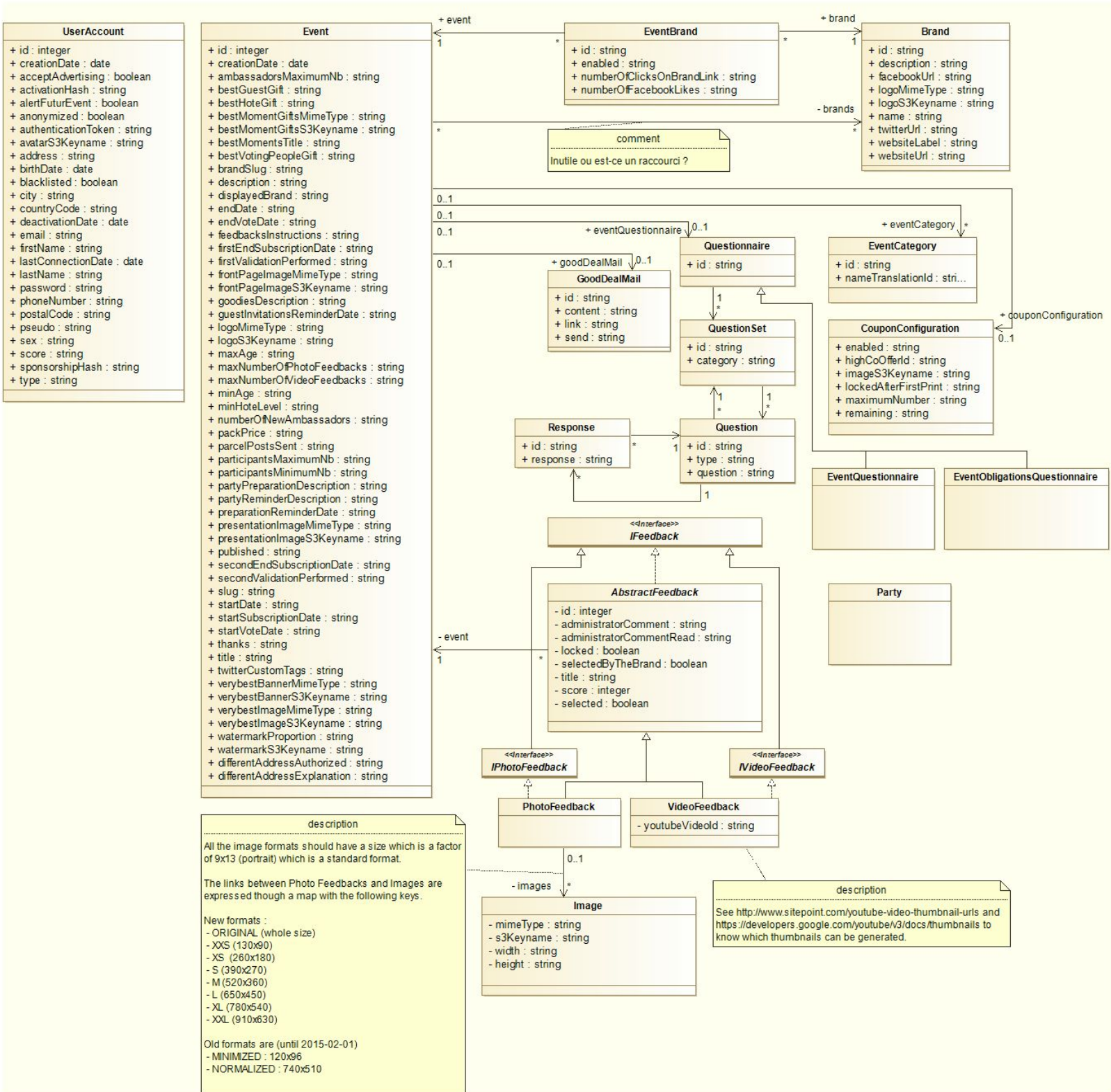


Figure 12 - Modèle Conceptuel des Données de l'application