

Projet de Fin d'Etudes

Pour l'Obtention Du Diplôme D'Ingénieur D'Etat

Filière : Génie Informatique

Réalisé au sein de : APPARTOO



**DEVELOPPEMENT D'UNE APPLICATION WEB DE MATCHING ENTRE
COLOCATAIRES**

Période de stage : Du 13/02/2017 Au 13/07/2017

Réalisé par :

- **Mohamed NACHIT**

Encadré par :

- M. KENZI Adil
- M. CROMBEZ Jimmy

Présenté le /07/2017 Devant le Jury composé de :

- M. Kenzi Adil

Année Universitaire : 2016-2017

Remerciements

Au terme de mon projet de fin d'études, je tiens à exprimer ma profonde gratitude et mes sincères remerciements pour tous ceux qui ont aidé de près ou de loin à la réalisation de ce travail.

Je tiens à remercier M. Jimmy Crombez, de m'avoir permis d'effectuer mon projet de fin d'études au sein d'Appartoo Paris, et qui m'a fait part de son temps et de son énergie et qui m'a fait bénéficier de son savoir-faire et ses conseils estimables. Je le remercie plus particulièrement pour sa patience et sa gentillesse.

Je ne manquerai pas de remercier aussi, M. KENZI Adil mon encadrant interne à l'ENSA Fès, pour ses efforts et ses conseils, et pour le suivi qu'il m'a apporté tout au long de cette période de stage.

Mes sincères gratitude pour tout le cadre professoral de l'ENSA Fès, pour la formation qu'il m'a inculquée.

Que messieurs les membres du jury trouvent ici l'expression de ma reconnaissance d'avoir accepté de juger ce travail.

Mes expressions distinguées, pour tous ceux qui ont contribué de près ou de loin à l'accomplissement de ce projet.

Merci encore.

Listes des figures

Figure 1: Architecture du projet	7
Figure 2 Architecture Scrum	10
Figure 3 : Diagramme Grantt du projet	10
Figure 4 : Colonnes Trello du projet	11
Figure 5 : Colonne Tâche du Sprint	12
Figure 6 : Colonne en cours Trello.....	12
Figure 7 : Colonne prêt à tester Trello	12
Figure 8 : Diagramme de Classe	13
Figure 9 : Schéma explicative du scénario d'un chercheur	14
Figure 10 : Schéma explicative du scénario d'un Proposeur	14
Figure 11 : Schéma explicative des rôles d'un administrateur	15
Figure 12 : Routes du service Web	25
Figure 13 : Page d'accueil.....	28
Figure 14 Page d'inscription.....	28
Figure 15 : Page d'informations du profil.....	29
Figure 16 Page de création de colocation virtuelle	30
Figure 17 Page de recherche	30
Figure 18 Page du profil.....	31
Figure 19 Page d'invitation.....	31
Figure 20 Interface de messagerie.....	32
Figure 21 Interface de paiement.....	33
Figure 22 Interface ADmin Profil	33
Figure 23 Interface admin annonce	34
Figure 24Admin Statistique	34

Table des matières

Introduction générale.....	1
Chapitre 1	2
I- Présentation de l'organisme d'accueil.....	3
II- Présentation du projet	4
III- Structure du projet	5
IV- Répartition des tâches	7
Chapitre 2	8
I- Approche Agile	9
Définition.....	9
II- Scrum.....	9
III- Application de Scrum sur notre projet.....	10
IV- Déroulement de l'application de la méthode	11
V- Conception.....	13
Chapitre 3	16
I- Technologies et outils de développement	17
II- Déroulement de Mon stage.....	24
III- Démonstration	27
Conclusion.....	35

Introduction générale

En France, trouver un logement est une tâche très difficile à cause des restrictions et des exigences des propriétaires dont la majorité impose aux candidats d'avoir un revenu important afin de garantir le paiement du loyer sans retard.

Par conséquent, la majorité des candidats se présentent en groupe de colocataires afin d'avoir une force financière supérieure, et ainsi convaincre les propriétaires de leurs éligibilités.

Malheureusement, les colocataires ne s'entendent pas forcément, la différence des personnalités et des modes de vies crée en effet des problèmes entre les individus, ce qui pourrait conduire à leur séparation, et ainsi, présenter un risque quant au paiement du loyer.

Afin de remédier à ces problèmes, APPARTOO a décidé de créer une application qui agira sur tout le processus de recherche d'un logement :

En premier lieu, aider ces utilisateurs à trouver un groupe de colocataires dont la personnalité est compatible avec la leur, afin de garantir une entente idéale.

En second lieu, leur proposer des logements qui correspondent à leurs critères de recherches.

Pour atteindre ce but, APPARTOO repose sur un réseau social qu'elle a acquis grâce à la première version de son site, qui avait la simple fonction de proposer des annonces de location à ces utilisateurs.

C'est dans cette optique que s'inscrit mon projet de fin d'études au sein de la société APPARTOO Paris visant le développement de cette application en trois parties :

La première est l'implémentation d'un service WEB en PHP (Symfony) qui se chargera d'exécuter les différents algorithmes et traitements relatifs aux objectifs de l'entreprise.

La deuxième est la création d'une interface graphique à l'aide du framework AngularJS qui permettrait à l'utilisateur d'avoir une expérience agréable.

La troisième consiste à implémenter une interface administrateur afin de gérer les utilisateurs, détecter les fraudes et récupérer des statistiques afin d'avoir une meilleure visibilité sur l'évolution du site.

Après le développement de l'application, il faudrait alors effectuer une migration de la base de données de la première version du site vers la nouvelle base de données.

A travers ce rapport, fruit de mon effort, j'essaierai de décrire les différentes étapes parcourues afin de venir à bout de mon projet. Je commencerai par un premier chapitre à travers lequel je présenterai le cadre général du projet, notamment l'organisme d'accueil APPARTOO, une description générale du projet et de ses objectifs. Après j'entamerai un chapitre qui décrira la planification, conception et la méthodologie suivie afin de réaliser le projet. Enfin, le dernier chapitre avant la conclusion contiendra toutes les informations sur la mise en œuvre du projet, notamment les technologies utilisées et les étapes du développement.

Chapitre 1

Cadre général du projet

Ce chapitre introduit le contexte général dans lequel s'est déroulé le projet. Il présente d'abord, APPARTOO Paris, l'organisme d'accueil. Ensuite il décrit le projet ainsi que ses objectifs.

I- Présentation de l'organisme d'accueil

Cet axe est consacré à la présentation de l'organisme d'accueil Appartoo. Nous allons d'abord présenter globalement Appartoo avant d'exposer les services qu'elle offre à ses clients, ensuite nous allons dévoiler l'organisation générale de la société.



Appartoo est une jeune Start-up créée en 2015 par Crombez Jimmy spécialisée dans le secteur immobilier. Elle est née d'une page Facebook qui a réussi à réunir en quelques mois seulement, une communauté de plus de 100 000 personnes autour du partage de « bons plans colocations » sur Paris.

En plus de sa page Facebook, Appartoo possède un site web qui permet à ses utilisateurs de retrouver des annonces de logement.

- **Services :**

Aider ses clients à trouver des logements et les mettre en contact avec leurs colocataires idéals.

Présenter des chercheurs de location aux propriétaires d'appartement.

Assister les agences immobilières pour trouver des candidats éligibles

Fournir aux assurances des dossiers qui correspondent à leurs critères

Ces services sont mis en place en utilisant plusieurs méthodes :

L'organisation des évènements de rencontres entre les colocataires potentiels

La publication des offres d'appartements sur leur site web

Application mobile Appartoo

- **L'équipe :**

Appartoo est une entreprise à hiérarchie horizontale, ce qui était un plus dans ce stage car cela a permis d'établir une communication directe avec tous les membres et créer un climat amical entre nous.

Les collaborateurs de Appartoo sont :

Jimmy Crombez : Fondateur

Vincent Péliissier : Chef de projet et développeur de l'application iOS
Equipe de développement : regroupe 4 personnes, 3 étudiants d'écoles supérieures en France en stage de fin d'études, et moi-même.

II- Présentation du projet

Contexte

Afin d'augmenter les chances des candidats pour retrouver un logement, Appartoo a décidé de créer une application web et mobile qui permettraient aux utilisateurs de trouver leurs colocataires idéals pour contourner les critères qu'exigent les propriétaires. Après avoir trouvé leurs amis, ils peuvent alors chercher un appartement ensemble.

Cahier des charges

Pour achever ces objectifs, plusieurs fonctionnalités ont été prévues :

L'application doit permettre aux :

Personnes qui cherchent un logement :

- a. Lors de son inscription, l'utilisateur doit pouvoir fournir des informations sur son état (Age, situation maritale, fumeur/non fumeur, fonction ...) et sa personnalité (organisé ou bordélique, aime la musique, paresseux, aime les animaux, propre, etc.)
- b. Interface de recherche : l'utilisateur peut choisir entre chercher des appartements directement, ou bien chercher des colocataires selon sa personnalité + ses critères.

Cette interface contiendra une carte qui affiche l'emplacement des résultats.

- c. L'utilisateur doit pouvoir créer une colocation virtuelle, elle contiendra les personnes avec qui il veut habiter.
- d. L'utilisateur doit pouvoir ajouter des colocataires à sa colocation virtuelle depuis Facebook.
- e. Un utilisateur peut postuler pour rejoindre une colocation virtuelle
- f. L'utilisateur peut postuler pour une annonce d'appartement, soit seul ou bien avec son groupe virtuel de colocataires.

Personnes qui proposent un logement :

- a. Création d'une annonce de logement
- b. Pouvoir recevoir les candidatures des utilisateurs en forme de messages
- c. Possibilité d'accepter ou refuser des candidats
- d. Possibilité de chercher des candidats adaptés à ses critères

Fonctionnalités communes

- a. Une page de connexion et d'inscription des utilisateurs
- b. Interface de messagerie : l'utilisateur doit pouvoir discuter avec ses colocataires potentiels en temps réel.
- c. Système de mailing : l'utilisateur doit pouvoir recevoir des mails dans des événements précis.
- d. Système de notifications : l'utilisateur doit pouvoir recevoir des notifications en temps réel.
- e. Système de paiement : l'application doit contenir un système de gestion des utilisateurs basique et premium.

L'administrateur doit pouvoir :

- a. Voir la liste de toutes les annonces créées et leurs informations
- b. Voir la liste des utilisateurs et leurs informations
- c. Modifier, bannir, désactiver des utilisateurs et des annonces
- d. Filtrer les résultats
- e. Envoyer des mails aux utilisateurs
- f. Voir les statistiques de l'utilisation de l'application
- g. Exporter les données dans différents formats

III- Structure du projet

Afin de garantir une modularité et une indépendance aux langages de programmation, le projet adoptera une architecture orientée services, en utilisant les services WEB REST.

« L'architecture orientée service est un style d'architecture dont l'objectif est d'organiser un ensemble d'applications isolées, en un ensemble de services interconnectés, chacun étant accessible à travers des interfaces et des protocoles de communication standards. »

(M Kenzi Adil - la technologie des services web)

Back end : Service Web

En effet, un service web RESTful sera mis en place afin de permettre au site web et à l'application mobile de communiquer avec la même unité de traitement des données, et ainsi gagner du temps et garantir la réutilisabilité du service web.

Notre service web se chargera des fonctions suivantes :

Authentification des utilisateurs avec le standard JSON Web Token

Communiquer avec la base de données : Ajout – Suppression – Modification des profils, utilisateurs et annonces par le billet de l'orm doctrine

Utiliser elasticsearch afin de garantir une recherche rapide et optimisée, en plus de mettre en œuvre l'algorithme de matching.

Envoyer des mails aux utilisateurs

Envoi des notifications aux utilisateurs via web sockets

Ce service web sera implémenté en utilisant le langage PHP, à l'aide du Framework Symfony.

Le choix de cette technologie est dû au fait que Symfony offre des modules (bundles) qui facilitent largement la création des services web, en plus de gérer l'authentification et le cache.

Front End : Site Web

Afin de consommer notre service web, nous aurons besoin d'une application qui prendra le rôle de l'interface entre l'utilisateur et nos données. Pour cela, le développement d'un site web a été mis en place. Ce site se chargera d'être l'intermédiaire entre l'utilisateur et le service web tout en offrant à l'utilisateur une interface simple et complète afin de lui permettre de manipuler ses données, tout en respectant les règles de l'UX.

Cette application contiendra aussi une interface de messagerie instantanée qui utilisera Firebase comme base de données et système temps réel.

Ce site web sera développé en utilisant le framework Angular JS, ce framework permet d'augmenter la performance du site et facilite la manipulation de l'arbre DOM. En plus, il fournit des modules prêts pour des API comme Googlemaps Firebase et Facebook ce qui nous sera très utile.

Le front end contiendra les pages suivantes :

Une page d'accueil

Une page de recherche qui permettra à l'utilisateur d'insérer des critères spécifiques d'annonces ou bien de personnes.

Une page de connexion et d'inscription.

Une page descriptive d'une annonce (Colocations virtuelles ou bien d'appartements) où l'utilisateur trouvera toutes les informations par rapport à l'annonce et aux membres de celle-ci.

Une page descriptive du profil où l'utilisateur trouvera toutes les informations par rapport à celui-ci.

Une page de création d'annonce qui permettrait à l'utilisateur de créer une annonce d'appartement ou de colocation virtuelle à travers un Wizard.

Une page de gestion de compte où l'utilisateur pourrait changer son mot de passe et désactiver son compte.

Un tableau de bord où l'utilisateur pourrait changer ces informations à tout moment.

Une page de messagerie pour que les utilisateurs puissent échanger des messages en temps réel avec leurs candidats, tout en ayant la possibilité d'accepter/refuser leurs candidatures.

Front End Mobile

A l'instar du site web, l'application mobile offrira une solution simple et ergonomique aux utilisateurs des smartphones, le choix de l'iOS est basé sur les statistiques des utilisateurs de la première version du site, où plus de 70% des utilisateurs étaient sur des iPhone.

Communication entre le service WEB et le front End

La communication entre le service WEB et le Front END se fera par le biais des requêtes http qui porteront des données au format JSON. Afin de garantir la persistance de l'authentification des utilisateurs, une méthode d'identification par jeton sera mise en place, chaque requête http vers le service WEB inclura un jeton dans l'entête ' Authorization ', que le service web utilisera pour identifier l'utilisateur.

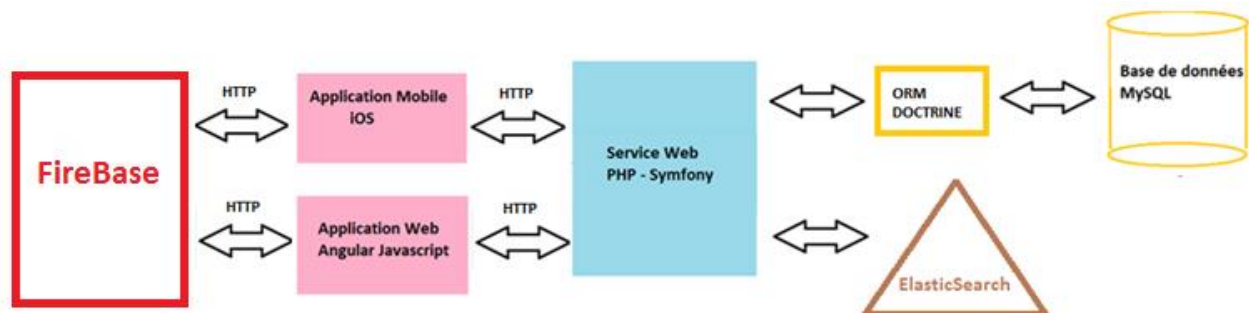


Figure 1: Architecture du projet

IV- Répartition des tâches

Afin de mettre en place ces fonctionnalités, chaque membre de l'équipe prendra une charge une partie du développement :

Vincent Pélissier : Application mobile

Thomas Ducroix : Design et migration des données de la première version du site.

Jeremy Beuvry et Timothée Robic : Développement du web service et du frontend AngularJs.

Mohamed Nachit : Développement du web service, du frontend AngularJs et de l'interface Administrateur.

Chapitre 2

Méthodologie et conception du projet

Ce chapitre décrit les méthodes suivies pour la réalisation du projet, les outils utilisés pour organiser le travail en équipe, ainsi que la conception choisie afin de modéliser le projet.

Afin d'aboutir aux objectifs fixés du projet, en s'adaptant rapidement aux changements des demandes du client en plus d'assurer une coopération permanente entre le client et l'équipe projet.

On a adopté l'approche Agile.

I- Approche Agile

Définition

Le terme "agile" définit une approche de **gestion de projet** qui prend le contre-pied des approches traditionnelles prédictives et séquentielles de type **cycle en V** ou waterfall (en cascade).

Une approche dite "traditionnelle" attend généralement du client une expression détaillée et validée du besoin en entrée de réalisation, laissant peu de place au **changement**. La réalisation dure le temps qu'il faut et le rendez-vous est repris avec le client pour la recette. Cet **effet tunnel** peut être très néfaste et conflictuel, on constate souvent un déphasage entre le besoin initial et l'application réalisée.

L'approche Agile propose au contraire de réduire considérablement voire complètement cet **effet tunnel** en donnant davantage de visibilité, en impliquant le client du début à la fin du projet et en adoptant un **processus itératif et incrémental**. Elle considère que le besoin ne peut être figé et propose au contraire de s'adapter aux changements de ce dernier. Mais pas sans un minimum de règles.

(Introduction aux méthodes agiles et Scrum, s.d.)

II- Scrum

Scrum est une méthode agile dédiée à la « gestion de projet ». Cette méthode de gestion, ou plutôt ce *Framework de management de projet*, a pour objectif d'**améliorer la productivité de son équipe**.

- Scrum définit 3 rôles : Le « Product Owner » qui porte la vision du produit à réaliser (représentant généralement le client).
- Le « Scrum Master » garant de l'application de la méthodologie Scrum.
- L'équipe de développement qui réalise le produit.

L'application de cette méthode comporte plusieurs étapes :

La première étape consiste à effectuer une première planification de l'itération (Sprint). Cette réunion fera ressortir les éléments prioritaires de la liste des exigences fonctionnelles du produit. Chaque exigence représente une user story (US) ou "histoire utilisateur". Elle doit également contenir les cas de tests ainsi que les critères de validation de la fonctionnalité développée.

En accord avec le client, aussi appelé Product Owner, les premières livraisons devraient être effectuées à la fin de cette itération (une itération à une durée d'environ 2 à 3 semaines suivant

le nombre d'US présent dans le backlog). Le backlog est l'ensemble des US à développer durant l'itération en cours.

Une autre réunion appelée Revue de Sprint est organisée à la fin de chaque Sprint durant laquelle les développeurs présentent au client les fonctionnalités développées. Ce dernier pourra ainsi tout de suite donner son feedback, ce qui présente l'avantage de gagner beaucoup de temps et d'ajuster les fonctionnalités ou les méthodes de travail le cas échéant.



Figure 2 Architecture Scrum

III- Application de Scrum sur notre projet

Dans notre projet, les rôles Scrum étaient répartis comme suit :

- Le « Product Owner » : Mr Jimmy Crombez, fondateur d'Appartoo.
- Le « Scrum Master » : Vincent Pélissier.
- L'équipe de développement : Moi-même, Jeremy, Timothée et Thomas.

Un sprint = 1 semaine

Planification du Projet

Le diagramme de Gantt suivant décrit la planification du projet ainsi que la durée prévue pour chaque étape

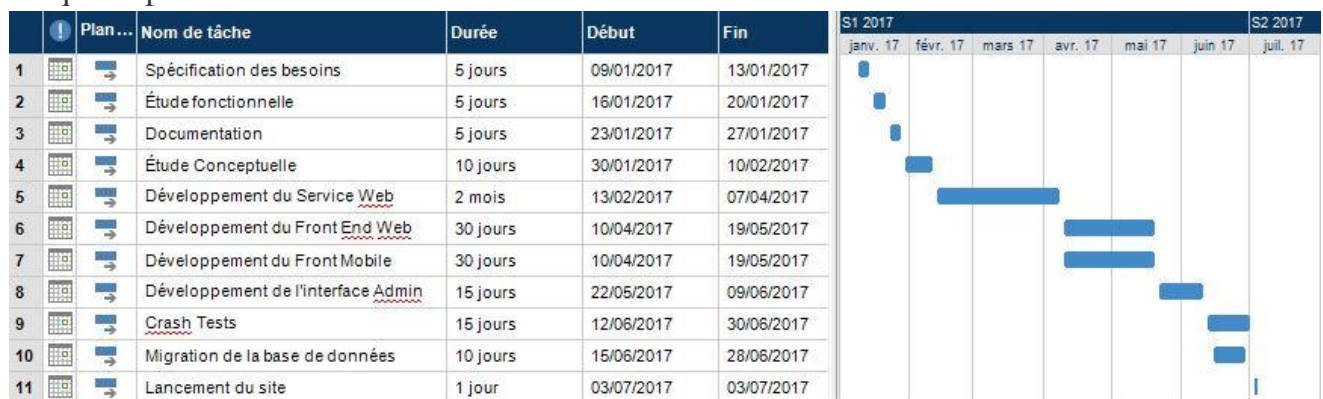


Figure 3 : Diagramme Gantt du projet

IV- Déroulement de l'application de la méthode

Réunion de planification d'un Sprint

Chaque Lundi, Le client, Scrum master et l'équipe du développement dont je fais partie organisent une réunion afin de définir les tâches du sprint.

Réunion à la fin du sprint : Revue de sprint

Une réunion est alors tenue chaque vendredi (Fin du sprint) pour faire des tests sur les tâches effectuées afin de s'assurer qu'elles marchent comme prévu, en consultant l'avis des développeurs.

Afin d'avoir une meilleure visibilité sur nos tâches et notre avancement, l'équipe a opté pour l'utilisation de l'outil Trello. Et pour avoir une bonne communication entre l'équipe lors des sprints, on a utilisé le logiciel slack.

Stand up meeting quotidien (daily scrum)

Chaque matin, nous faisons une réunion rapide, où chacun des développeurs a 5 minutes de parole, où il décrit ce qu'il a fait, ce qu'il est en train de faire, et ce qu'il compte faire après, ainsi que les difficultés qu'il a eues.



Trello est un outil de gestion de projet en ligne, lancé en septembre 2011, et inspiré par la méthode Kanban de Toyota. Il est basé sur une organisation des projets en planches listant des cartes, chacune représentant des tâches. Les cartes sont assignables à des utilisateurs et sont mobiles d'une planche à l'autre, traduisant leur avancement (Wikipédia - Trello, s.d.).

Afin d'aligner l'outil avec notre méthodologie Scrum, nous avons créé 4 colonnes / planches qui serviront à présenter nos tâches et les membres qui travaillent dessus.

Tâches du Sprint	En Cours	Prêt à tester	Archivé
Add a card...	Add a card...	Add a card...	Add a card...

Figure 4 : Colonnes Trello du projet

Tâches du Sprint : Cette colonne contiendra toutes les tâches prévues pour le sprint actuel

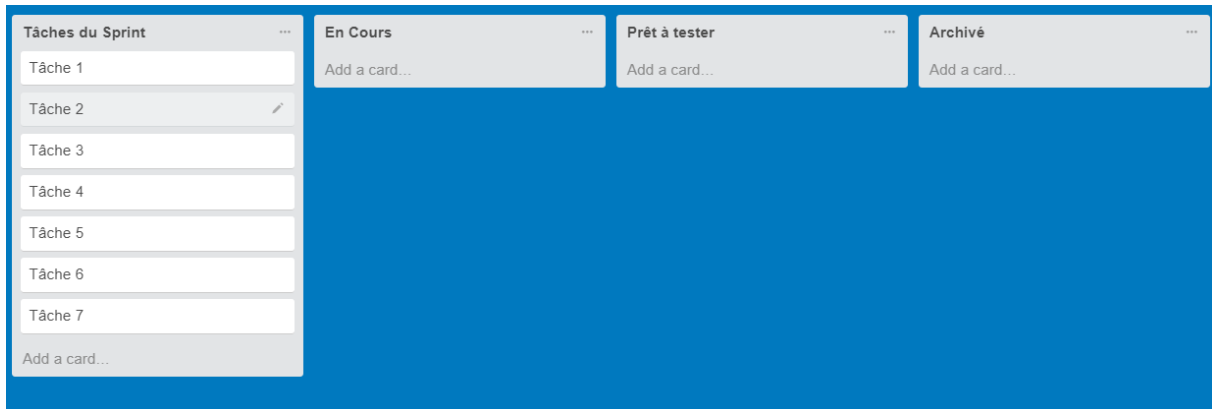


Figure 5 : Colonne Tâche du Sprint

En Cours : Chaque membre de l'équipe de développement s'attribuera une tâche et la mettra dans la colonne en cours

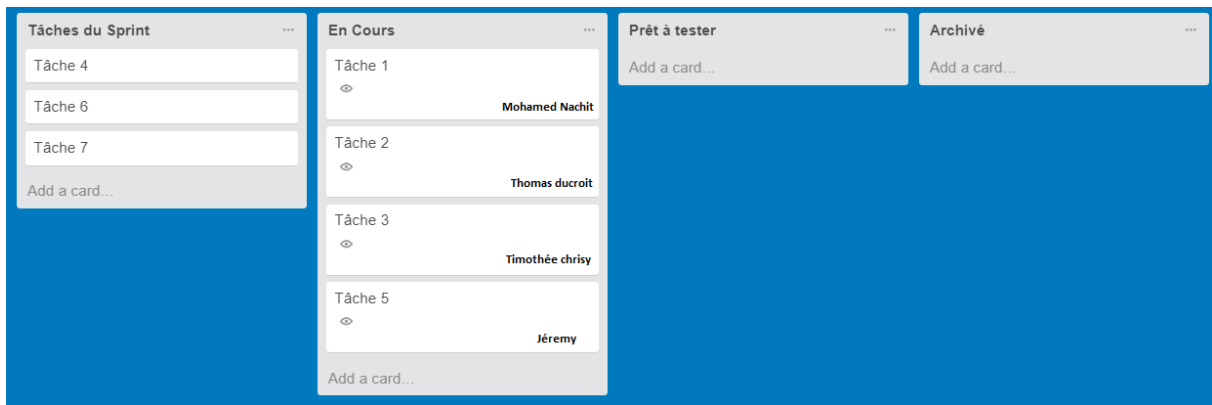


Figure 6 : Colonne en cours Trello

A tester : Après avoir fini la tâche, celle-ci est glissée dans la colonnes prêt à tester

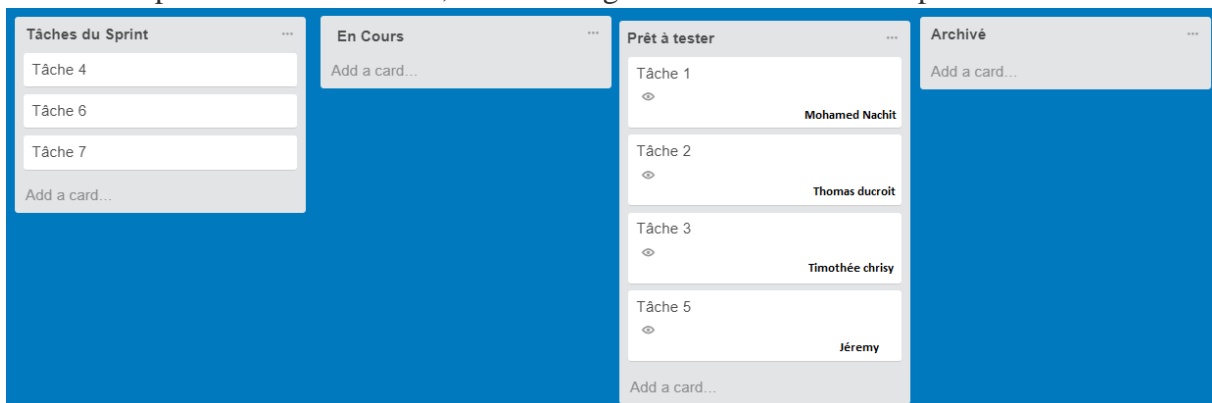


Figure 7 : Colonne prêt à tester Trello

Archivé : Lors de la réunion des tests, si la tâche fonctionne comme prévu, on l'archive, sinon on la remet dans la colonne des tâches du sprint



Slack est une plate-forme de communication collaborative propriétaire (SaaS) ainsi qu'un logiciel de gestion de projets .

Slack fonctionne à la manière d'un chat IRC organisé en canaux correspondant à autant de sujets de discussion. La plateforme permet également de conserver une trace de tous les échanges (« Slack » est l'acronyme de « Searchable Log of All Conversation and Knowledge »³), permet le partage de fichiers au sein des conversations et intègre en leur sein des services externes comme GitHub, Dropbox, Google Drive ou encore Heroku pour centraliser le suivi et la gestion d'un projet (Wikipédia - Slack, s.d.)

V- Conception

La conception de l'application était déjà faite lorsque j'ai rejoint l'équipe. Néanmoins j'ai dû faire quelques modifications durant le développement afin de m'adapter aux nouvelles tâches demandées.

Diagramme de classe métier

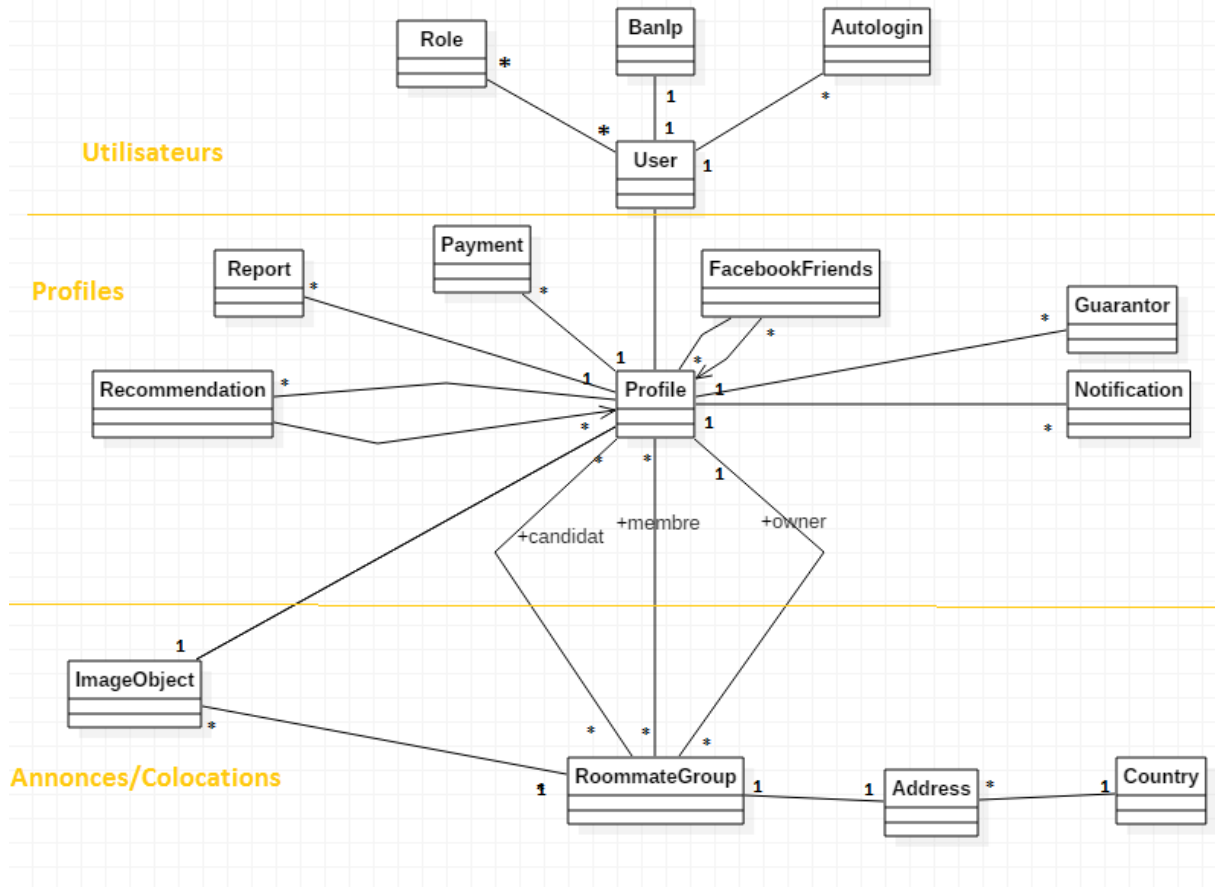


Figure 8 : Diagramme de Classe

La classe RoommateGroup concerne les deux cas d'annonces : les colocations virtuelles ainsi que les appartements, vu que les deux ont les mêmes attributs.

La classe profile contient toutes les informations sur le profil et ses caractéristiques, elle va être primordiale afin de faire le matching.

Illustrations des scénarios possibles d'utilisation du site :

Scénario d'une personne qui cherche où habiter :

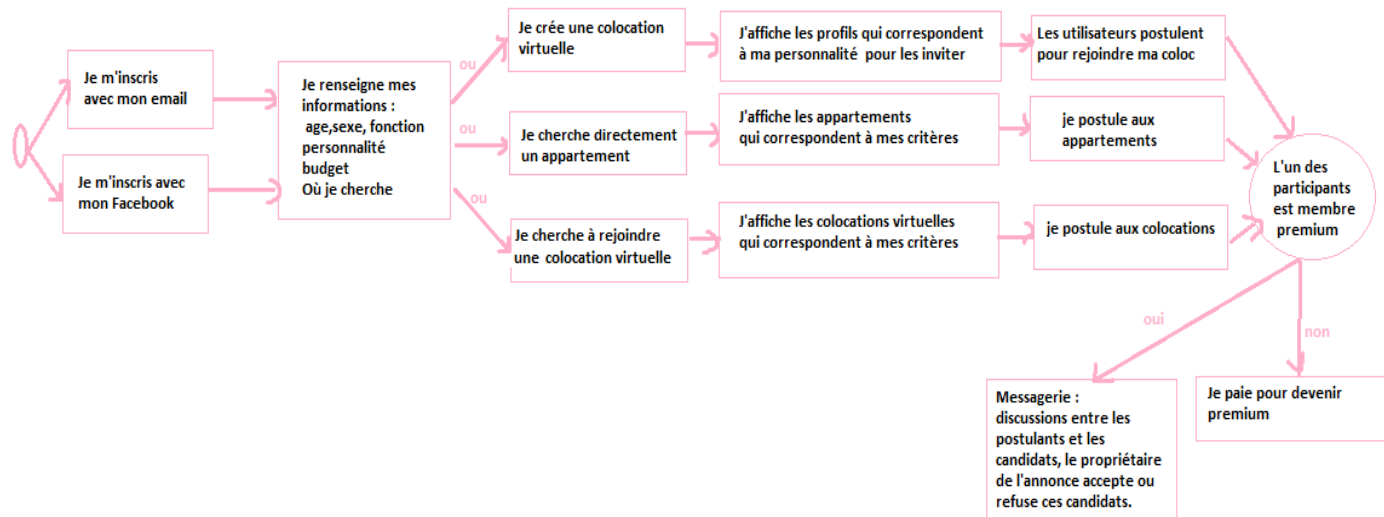


Figure 9 : Schéma explicative du scénario d'un chercheur

Scénario d'un utilisateur qui propose un appartement

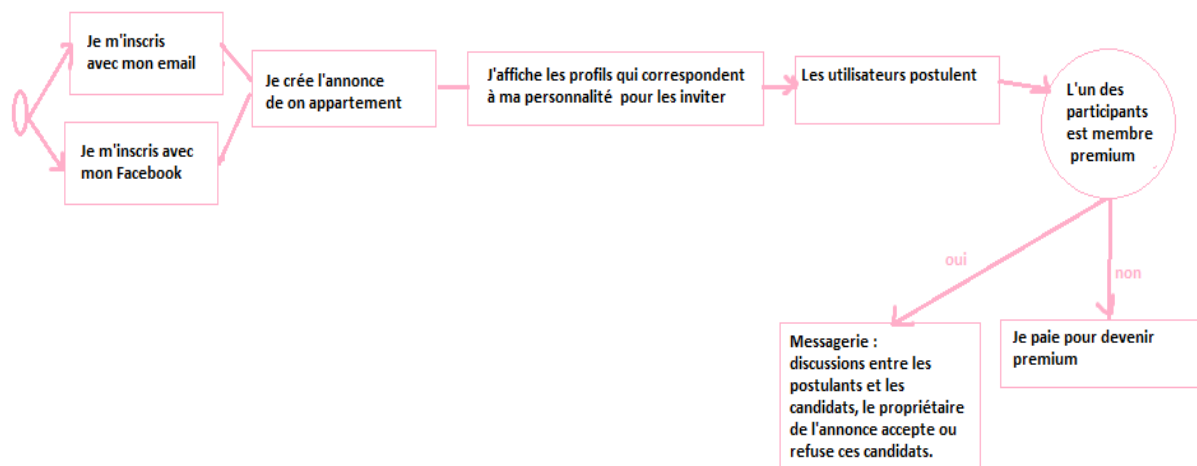


Figure 10 : Schéma explicative du scénario d'un Proposeur

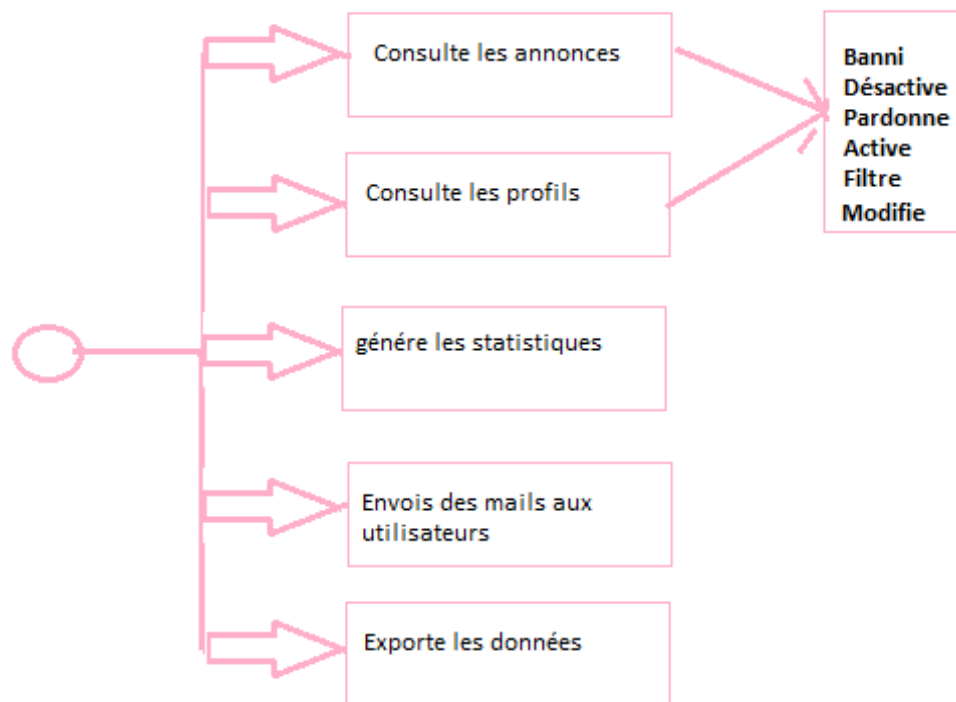
Administrateur

Figure 11 : Schéma explicative des rôles d'un administrateur

Chapitre 3

Déroulement du projet

Ce chapitre présente les technologies utilisées durant notre projet, ainsi que les étapes suivies afin de le réaliser et ma contribution dans ces étapes.

I- Technologies et outils de développement

Outils de développement



git est un logiciel de gestion de versions décentralisé.

Un gestionnaire de version est un système qui enregistre l'évolution d'un fichier ou d'un ensemble de fichiers au cours du temps de manière à ce qu'on puisse rappeler une version antérieure d'un fichier à tout moment. Dans les exemples de ce livre, nous utiliserons des fichiers sources de logiciel comme fichiers sous gestion de version, bien qu'en réalité on puisse l'utiliser avec pratiquement tous les types de fichiers d'un ordinateur.



Bitbucket est un service web d'hébergement et de gestion de développement logiciel utilisant les logiciels de gestion de versions Git et Mercurial.

Il sera utilisé afin d'héberger notre projet.



POSTMAN

POSTMAN est un client http qui permet d'envoyer des requêtes personnalisés.

Justification du choix : tester le service web

Outils serveurs



Nginx est un serveur HTTP et reverse proxy utilisé par de nombreux sites. Il utilise les changements d'état pour gérer plusieurs connexions en même temps, le traitement de chaque requête est découpé en de nombreuses mini-tâches et permet ainsi de réaliser un multiplexage efficace entre les connexions. Afin de tirer parti des ordinateurs multiprocesseurs, plusieurs processus peuvent être démarrés. Ce choix d'architecture se traduit par des performances très élevées, mais également par une charge et une consommation de mémoire particulièrement faibles

comparativement aux serveurs HTTP classiques, tels qu'Apache.

Raison d'utilisation : Augmenter les performances du serveur, rediriger l'url vers les bon ports car notre service web et notre site fonctionne sur des ports différents (8080 et 8000).



Secure Shell (SSH) est à la fois un programme informatique et un protocole de communication sécurisé. Le protocole de connexion impose un échange de clés de chiffrement en début de connexion. Par la suite, tous les segments TCP sont authentifiés et chiffrés. Il devient donc impossible d'utiliser un sniffer pour voir ce que fait l'utilisateur.

Le protocole SSH a été conçu avec l'objectif de remplacer les différents programmes rlogin, telnet, rcp, ftp et rsh.

Avec SSH, l'authentification peut se faire sans l'utilisation de mot de passe ou de phrase secrète en utilisant la cryptographie asymétrique. La clé publique est distribuée sur les systèmes auxquels on souhaite se connecter. La clé privée, qu'on prendra le soin de protéger par un mot de passe, reste uniquement sur le poste à partir duquel on se connecte. L'utilisation d'un « agent ssh (en) » permet de stocker le mot de passe de la clé privée pendant la durée de la session utilisateur.

Justification du choix : garantir une communication sécurisée avec le serveur



rsync remote synchronization, (en français : « synchronisation distante ») est un logiciel libre de synchronisation de fichiers, distribué sous licence GNU GPL. La synchronisation est unidirectionnelle, c'est-à-dire qu'elle copie les fichiers de la source en direction de la destination. rsync est donc utilisé pour réaliser des sauvegardes incrémentielles ou différentielles ou pour diffuser le contenu d'un répertoire de référence.

Il utilise un protocole de mise à jour à distance plus efficace afin d'accélérer significativement le transfert de fichiers lorsque le fichier de destination existe déjà.

Le protocole de mise à jour à distance rsync lui permet de ne transférer que la différence entre deux jeux de fichiers à travers le réseau, en utilisant un algorithme de recherche de somme de contrôle. Ce protocole établit ses connexions sur le port TCP 873 par défaut

Justification du choix : Déploiement des applications développée d'une manière rapide vu qu'il fait une mise à jour au lieu de copier tout le contenu.

Web Service (Back end)



Symfony est un ensemble de composants PHP ainsi qu'un framework MVC libre écrit en PHP. Il fournit des fonctionnalités modulables et adaptables qui permettent de faciliter et d'accélérer le développement d'un site web.

Justification du choix :

Le framework Symfony est très modulaire et il est possible d'en choisir uniquement certains morceaux.

Symfony a été conçu pour s'intégrer facilement avec un ORM tel que Doctrine permettant de faciliter la persistance d'objets en base de données.

Gestion multi-langues simplifiée

Système intégré d'améliorations des performances

Existence de nombreux Bundles/pluggins tel que Sonata

Forte capacité d'évolution

Communauté toujours plus grande (plus de 300 000 développeurs dans 120 pays)¹²

Bundles : FOSRESTBUNDLE : Fournit plusieurs outils pour développer rapidement des API REST

JWTAuthenticationBundle : Permet l'authentification JWT (Json Web Token) pour les applications symphony

GosWebSocketBundle : Permet l'utilisation du protocole Web Sockets.

FOSELASTICABundle : Permet l'intégration d'elasticsearch dans notre projet

Doctrine : ORM qui fait le mapping entre le modèle objet et le modèle relationnel



Firebase est un ensemble de services d'hébergement pour n'importe quel type d'application (Android, iOS, Javascript, Node.js, Java, Unity, PHP, C++ ...). Il propose d'héberger en NoSQL et en temps réel des bases de données, du contenu, de l'authentification sociale (Google, Facebook, Twitter et Github), et des notifications, ou encore des services, tel que par exemple un serveur de communication temps réel. Lancé en 2011 sous le nom d'Envolve, par Andrew Lee et par James Templin, le service est racheté par Google en octobre 2014. Il appartient aujourd'hui à la maison mère de Google : Alphabet.

Toute l'implémentation et la gestion serveur de Firebase est à la charge exclusive de la société Alphabet. Les applications qui utilisent Firebase intègrent une bibliothèque qui permet les diverses interactions possibles.

Justification du choix : Stockage des conversations des utilisateurs de manière sécurisée, avec un format JSON facile à parser, et permet de les retrouver facilement. En plus de fournir une communication en temps réel qui nous aidera à implémenter une messagerie instantanée. Ce stockage nous permettra d'économiser de la place sur notre serveur.



WebSocket est un standard du Web désignant un protocole réseau de la couche application et une interface de programmation du World Wide Web visant à créer des canaux de communication full-duplex par-dessus une connexion TCP. Le protocole a été normalisé par l'IETF dans la RFC 64552 en 2011 et l'interface de programmation est en cours de standardisation par le W3C.

Le WebSocket proposera à terme une implémentation native et unifiée dans les navigateurs et serveurs web d'un canal bidirectionnel permettant :

- La notification au client d'un changement d'état du serveur
- L'envoi de données en mode « pousser » (méthode Push) du serveur vers le client (sans que ce dernier ait à effectuer une requête)

Justification du choix : meilleure façon d'implémenter les notifications en temps réel.



Elasticsearch est un serveur utilisant Lucene pour l'indexation et la recherche des données. Il fournit un moteur de recherche distribué et multi-entité à travers une interface REST. C'est un logiciel libre écrit en Java et publié en open source sous licence Apache.

Elasticsearch est le serveur de recherche d'entreprise le plus populaire, suivi par Apache Solr qui utilise aussi Lucene2. Il est associé à deux autres produits libres, Kibana et Logstash, qui sont respectivement un visualiseur de données et un ETL (initialement destiné aux logs).

L'indexation des données s'effectue à partir d'une requête HTTP PUT. La recherche des données s'effectue avec la requête HTTP GET. Les données échangées sont au format JSON.

Justification du choix: Nous avons utilisé elasticsearch afin d'optimiser les recherches dans la base de données. Et augmenter les performances de notre service web, en plus d'offrir des fonctions intégrées de matching.



MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde³, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, Informix et Microsoft SQL Server. MySQL est un serveur de bases de données relationnelles SQL développé dans un souci de performances élevées en lecture, ce qui signifie qu'il est davantage orienté vers le service de données déjà en place que vers celui de mises à jour fréquentes et fortement sécurisées. Il est multi-thread et multi-utilisateur.

Justification du choix: simple, open source.

Front End



AngularJS3 est un framework JavaScript libre et open source4 développé par Google.

AngularJS est fondé sur l'extension du langage HTML par de nouvelles balises (tags) et attributs pour aboutir à une définition déclarative des pages web, par opposition à l'utilisation systématique de l'élément div et à la définition des éléments de présentation en JavaScript.

Le code HTML étendu représente alors la partie « vue » du patron d'architecture MVC (modèle-vue-contrôleur) auquel AngularJS correspond, via des modèles en couche appelés « scopes » et des contrôleurs permettant de prototyper des actions en code JavaScript natif. AngularJS utilise une boucle de dirty-checking (qui consiste à surveiller et à détecter des modifications sur un objet JavaScript) pour réaliser un data-binding bidirectionnel permettant la synchronisation automatique des modèles et des vues.

AngularJS embarque un sous-ensemble de la bibliothèque open source jQuery appelé jqLite, mais peut aussi utiliser jQuery si elle est chargée

Justification du choix : Facile à utiliser, augmente les performances du site et modulable, contient des modules sur les API qu'on veut utiliser (AngularFire pour firebase, ezfb pour facebook).



Sass (Syntactically Awesome Stylesheets) est un langage de génération de feuilles de style initialement développé par Hampton Catlin et Nathalie Weizenbaum.

Sass est un langage de feuilles de style en cascade (CSS). C'est un langage de description qui est compilé en CSS. SassScript est un langage de script pouvant être utilisé à l'intérieur du code Sass. Deux syntaxes existent. La syntaxe originale, nommée « syntaxe indentée », est proche de Haml. La nouvelle syntaxe se nomme « SCSS ». Elle a un formalisme proche de CSS.

Il ajoute à css les mécanismes suivants : variables, imbrication, mixins, opérateurs et fonctions.

Justification du choix : préférence personnelle.

Bootstrap



Bootstrap est une collection d'outils utile à la création du design (graphisme, animation et interactions avec la page dans le navigateur ... etc. ...) 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 en option. C'est l'un des projets les plus populaires sur la plate-forme de gestion de développement GitHub.

Justification du choix: Garantit la responsivité des sites web, bonne documentation.

Les api utilisées



API Google Maps : API qui permet d'intégrer et de manipuler la carte de google.



API Facebook : permet d'utiliser les fonctionnalités de facebook, authentification , partage , envoi de message...



Stripe : API qui permet de collecter les informations des cartes bancaire d'une façon sécurisé Et effectuer des paiements sur celles ci.



Nexmo : API qui permet d'envoyer des sms aux numéros de téléphones.



MailGun : Api puissante qui permet d'envoyer, recevoir et traquer les emails.

II- Déroulement de Mon stage

Dans ce chapitre, je vais décrire les tâches que j'ai pu réaliser durant mon stage ainsi que ma progression.

Lors de la première semaine de mon stage, j'ai eu la chance de rencontrer plusieurs mentors qui m'ont donné des conseils très précieux. En effet, Appartoo a fait appel à des ingénieurs expérimentés qui ont pu nous transmettre leur savoir-faire sur les bonnes pratiques du développement, du travail en équipe et sur l'UI (user interface) et l'UX (user experience) .

Utilisation de Git

Nous avons commencé le projet concrètement par l'installation de Git pour organiser le travail en équipe. Notre utilisation de Git était comme suit :

Le produit a été divisé en deux projets : le frontend et le backend, chaque développeur de l'équipe possède une branche en son nom sur chaque projet, où il travaille sans affecter l'avancement des autres.

En plus, il existe une branche appelée 'dev' qui va regrouper l'avancement de chacun de nous. A chaque fois que quelqu'un finit une tâche importante, il va fusionner (merge) sa branche avec la branche dev.

Après avoir fini le développement du site et du service web, la branche dev sera alors fusionnée avec la branche master qui contiendra le produit final.

Service Web

Nous avons décidé de commencer par le développement des fonctionnalités de base du web service vu que c'est la base de nos applications.

J'ai consacré la deuxième semaine à l'auto-formation, j'ai recherché les meilleures solutions afin d'implémenter un service web sur Symfony, je me suis formé sur elasticsearch, websockets et l'authentification avec des jetons json afin de bien les maîtriser avant de les utiliser.

Ensuite, on a commencé le développement du service web.

Dans un premier temps nous avons défini les différentes routes dont nous aurions besoin pour notre service.

Voici un exemple de routes que contient notre web service :

search_manager	GET	ANY	ANY	/api/searchGeneral
api_email_message_send	POST	ANY	ANY	/api/message/send
api_alert_post	POST	ANY	ANY	/api/alert
api_report_post	POST	ANY	ANY	/api/report
api_contact_appartoo	POST	ANY	ANY	/api/contact
api_ban_ip	POST	ANY	ANY	/api/ban/ip
app_autologin	POST	ANY	ANY	/api/autologin/{email}/{ref}
api_send_new_msg	POST	ANY	ANY	/api/conversations/{conversationId}/message/new/
api_contact_profile	POST	ANY	ANY	/api/conversations/new
api_msg_desactivate	POST	ANY	ANY	/api/conversation/{conversationId}/message/desactivate/
api_update_push_token	POST	ANY	ANY	/api/pushToken
api_update_facebookId	PUT	ANY	ANY	/api/profiles/update/facebookId
api_update_search	PUT	ANY	ANY	/api/profiles/update/search
api_roommate_group_create	POST	ANY	ANY	/api/group
api_roommate_group_get	GET	ANY	ANY	/api/group/{groupId}
api_roommate_group_put	PUT	ANY	ANY	/api/group/{groupId}
api_roommate_group_status	PUT	ANY	ANY	/api/groups/{groupId}/isActive
api_group_find_with_appartoo	PUT	ANY	ANY	/api/groups/{groupId}/findwith
api_profile_find_with_appartoo	PUT	ANY	ANY	/api/profile/{profileId}/findwith

Figure 12 : Routes du service Web

A l'aide de trello, nous avons divisé les tâches de développement des fonctions de chacune de ces routes.

Lors de l'étape CRUD, j'ai développé la classe profil et annonce, les fonctions du CRUD sont automatiquement fournies par symfony grâce à l'orm doctrine, il fallait juste créer les classes, définir les routes :

GET api/profil/10

DELETE api/annonce/200,

Ainsi que développer les contrôleurs pour communiquer avec les modèles et nous renvoyer une réponse au format JSON.

Lors de l'étape authentification : j'ai utilisé l'api JWT, qui me renvoie un jeton lorsque je m'identifie avec le bon username/mot de passe, celui-ci sera utilisé sur chaque requête.

Etape Matching : grâce à l'api riche d'Elasticsearch, j'ai pu développer un algorithme qui se base sur les informations de l'utilisateur courant, et qui renvoie la liste des profils qui correspondent à sa personnalité.

Elasticsearch classe les résultats par score qui reflète le degré de compatibilité entre deux utilisateurs.

Mes collègues se sont chargés de développer la partie du matching propriétaire/locataire.

Etape du Mailing : l'un de mes collègues s'est chargé de cette fonctionnalité en utilisant l'api MailGun.

Etape Paiement : un autre collègue s'est chargé de cette fonctionnalité en utilisant l'api Stripe.

Etape Notifications : cette étape était très intéressante pour moi car grâce à elle j'ai appris une nouvelle technologie, celle des WEB Sockets. J'ai pu comprendre leur fonctionnement et comment les utiliser afin de garantir une communication full duplex entre le client et le serveur.

Interface Administrateur : Symfony offre un bundle SonataAdmin qui facilite la création d'une interface administrateur complète, avec toutes les fonctions de gestion, statistique et exportation. L'inconvénient de ce module est le fait que ses fonctions communiquent directement avec une seule base de données. Or, notre application utilise aussi firebase pour la messagerie, et nous aurions besoin d'avoir ces informations afin de calculer les taux de réponses d'utilisateurs, et leur statuts (en ligne / hors ligne).

C'est pour cela que j'ai personnalisé l'interface en ajoutant mes propres services et fonctionnalités afin de m'adapter à cette situation.

Grâce au bundle GOSWebSocket j'ai pu implémenter un système de notification qui utilise le mécanisme pubsub (publish-subscribe) afin de permettre aux clients de s'abonner à une chaîne pour recevoir les nouvelles notifications.

Front End

Notre collègue designer s'est chargé de faire la maquette du site, ainsi que de définir les palettes des couleurs et développer les composants statiques du site. Après nous avons défini les standards des composants du site (boutons, liste, panel) de tout le site afin d'avoir un design uniforme.

Ma mission était de prendre en charge les fonctionnements dynamiques des pages en utilisant le Framework AngularJS, afin de communiquer avec les deux services web (firebase et le nôtre), et ainsi recevoir des données à afficher sur nos pages et effectuer des actions.

La communication entre firebase se fait via le module angularfire, et la communication avec notre service web se fait grâce au module restangular qui permet d'envoyer des requêtes http plus facilement.

On a commencé par créer les routes du site et les lier à nos pages HTML.

Ensuite chacun de nous s'est chargé de développer le fonctionnement d'une page qui correspondent aux routes qu'on a définies.

Authentification : J'ai pu développer une page d'authentification, qui envoie les identifiants de l'utilisateur au service web, celui-ci nous renvoie un jeton qu'on doit inclure sur chaque requête http, dans le header Authorization.

Page de la messagerie : J'ai réalisé cette étape seul car elle était intéressante à mon avis, j'ai réussi à développer un système de messagerie en temps réel, grâce aux fonctions de détection des événements qu'offre le module angularfire, En plus, j'ai implémenté les actions (accepter , refuser quelqu'un) , gérer les statuts de chaque conversation, savoir si l'utilisateur est en ligne, ainsi que la détection des nouveaux

Messages pour afficher leur notification.

Page de recherche : J'ai pu grâce au module angular-google-maps, implémenter une carte google afin de refléter les résultats de nos recherches selon leurs localisation en utilisant les markers, afin de faciliter leurs visibilité et permettre une recherche plus précise. La page de recherche envoie une requête à notre service web, qui envoie une requête à elasticsearch afin de lui renvoyer les résultats adaptés à notre utilisateur, et ainsi les renvoyer à notre frontend pour qu'ils soient affichés correctement à notre utilisateur.

Actions Facebook : Je me suis également chargé du développement des actions possibles de Facebook, grâce au Facebook graph api. J'ai pu utiliser les fonctionnalités de connexion via un compte Facebook dans notre site, les fonctionnalités suivantes :

Partager une annonce

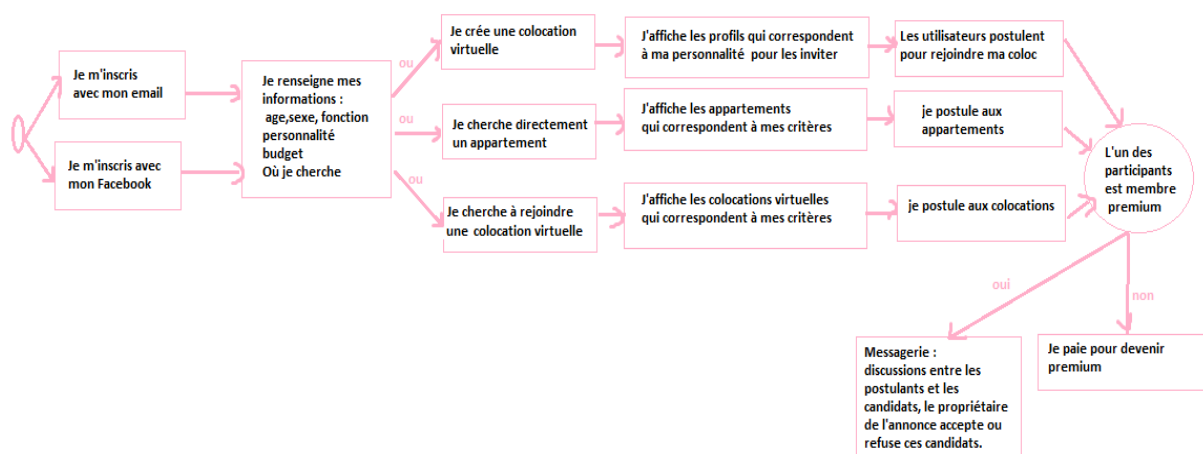
Envoyer des invitations aux amis pour les inciter à rejoindre le site

Possibilité d'ajouter des amis Facebook dans les colocations virtuelles, celle-ci se charge de lui créer un compte automatiquement grâce aux informations que l'api nous renvoie, et lui envoyer une notification pour l'inciter à se connecter au site.

Les pages de gestion de comptes, page des profils et d'annonces ont été gérées par mes collègues.

III- Démonstration

Nous allons suivre le schéma suivant afin de visualiser les étapes de notre démonstration, c'est le scénario d'une personne qui cherche un logement/ une colocation.



Page d'accueil

L'utilisateur arrivera en premier sur la page d'accueil



Figure 13 : Page d'accueil

Inscription

Il devrait ensuite s'inscrire pour profiter de toutes les fonctionnalités du site

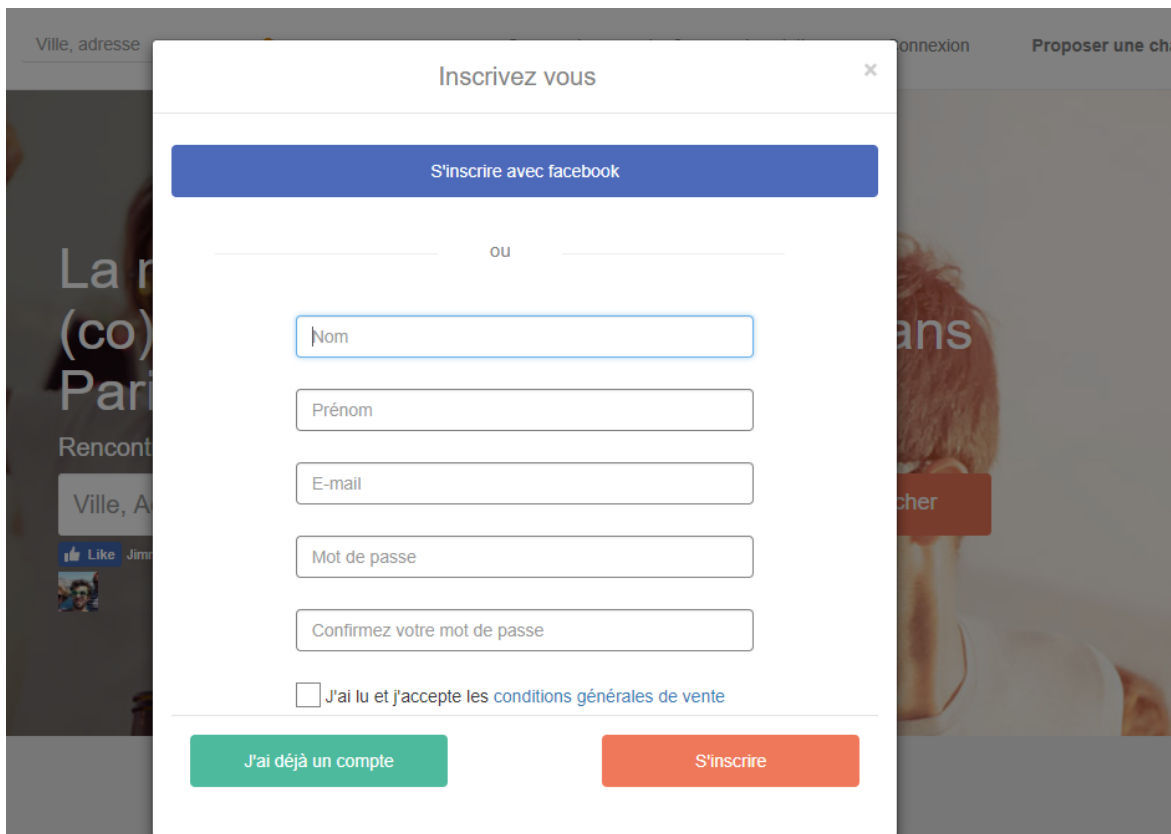
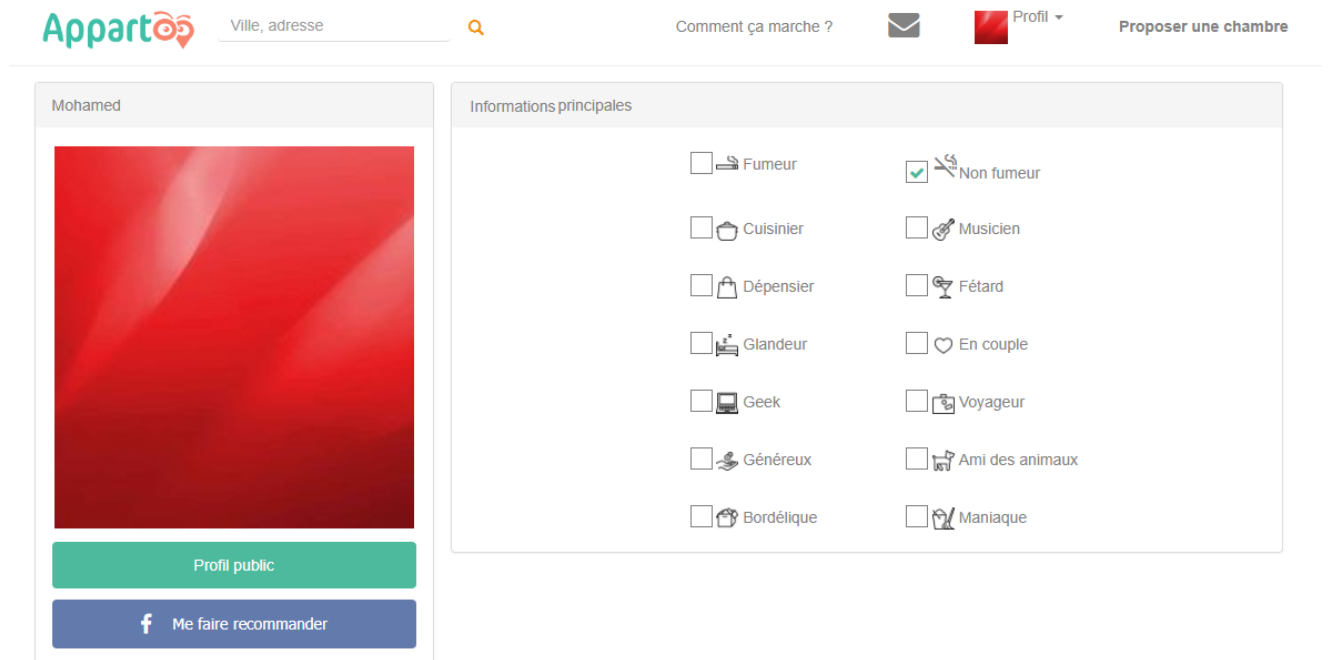


Figure 14 Page d'inscription

Renseignement des informations

Ensuite, il devra renseigner les informations qui correspondent à sa personnalité



The screenshot shows the Appart'logis website interface. At the top, there is a search bar with the text 'Ville, adresse' and a magnifying glass icon. To the right of the search bar are links for 'Comment ça marche ?', a mail icon, a 'Profil' dropdown menu, and a 'Proposer une chambre' button. Below the header, the profile of 'Mohamed' is displayed. On the left, there is a large red placeholder for a profile picture. Below the picture are two buttons: 'Profil public' (green) and 'Me faire recommander' (blue with a Facebook icon). On the right, under the heading 'Informations principales', there is a list of interests with checkboxes. The 'Non fumeur' option is checked.

Informations principales	
<input type="checkbox"/> Fumeur	<input checked="" type="checkbox"/> Non fumeur
<input type="checkbox"/> Cuisinier	<input type="checkbox"/> Musicien
<input type="checkbox"/> Dépensier	<input type="checkbox"/> Fétard
<input type="checkbox"/> Glandeur	<input type="checkbox"/> En couple
<input type="checkbox"/> Geek	<input type="checkbox"/> Voyageur
<input type="checkbox"/> Généreux	<input type="checkbox"/> Ami des animaux
<input type="checkbox"/> Bordélique	<input type="checkbox"/> Maniaque

Figure 15 : Page d'informations du profil

Il aura après le choix entre créer une colocation virtuelle ou bien rechercher des profils directement.

Création d'une colocation virtuelle

L'utilisateur doit suivre les étapes d'un wizard afin de créer la colocation.

75 %

Qui sont les autres colocataires ?

Ajouter des Colocs

OU

Ajouter un coloc

Précédent Suivant

Date de naissance 30/7/1993

Figure 16 Page de création de colocation virtuelle

Chercher des profils qui correspondent à ma personnalité

Appart Ville, adresse Comment ça marche ? Inscription Connexion Proposer une chambre

Critères pour les profils

Genre ☐ Femme ☐ Homme

Fumeur ☒ Oui ☐ Non

Célibataire ☒ Oui ☐ Non

Statut ☒ Etudiant ☐ Salarié

Moins de filtres je souhaite être alerté Rechercher

Béa Ariane Angélique

1000€ il y a 2 heure(s) 1000€ il y a 2 heure(s) 1000€ il y a 2 heure(s)

Figure 17 Page de recherche

Consultation des profils

Après avoir choisi le profil qui l'intéresse, l'utilisateur peut consulter les informations concernant celui-ci.

The screenshot shows the Appart'logement website interface. At the top, there's a navigation bar with the Appart'logement logo, a search bar labeled 'Ville, adresse', a magnifying glass icon, a link 'Comment ça marche ?', an envelope icon, a user profile icon labeled 'Profil', and a button 'Proposer une chambre'. Below this, the profile of Angélique is displayed. On the left, there's a large photo of her. To the right of the photo, there's a green button 'Inviter à me rejoindre', a yellow box indicating 'Actif il y a 2 heure(s)', a green box showing '100 % de réponses', and a red box labeled 'Membre Basique'. At the bottom of this section are three icons: Facebook (labeled 'Non vérifié'), Email (labeled 'Non vérifié'), and Phone (labeled 'Non vérifié'). To the right of the photo, there's a text box with the heading 'Bonjour, je suis Angélique !' followed by a bio: 'Professeure d'Histoire-Géographie parachutée sur l'académie de Créteil, je recherche une colocation sur la région parisienne. Mes 5 années d'expérience de la vie en coloc' m'ont donné une image plus que positive de celle-ci. Indépendante mais décontractée et bonne enfant, voilà l'ambiance que je recherche pour les années parisiennes qui se profilent ... Ma zone géographique; Paris 11, 12, 20 et RER A'. Below the bio is a green button 'Recommander : 0'.

Figure 18 Page du profil

Inviter à rejoindre ma colocation

Il pourrait alors l'inviter à rejoindre sa colocation virtuelle.

This screenshot shows the same profile page as Figure 18, but with a modal window open over it. The modal has a title bar that says 'Pour quelle annonce voulez-vous inviter ce profil à postuler ?'. Inside the modal, there's a small image of a room and the text 'Coloc Gare du Nord'. The background profile page is dimmed, but the 'Inviter à me rejoindre' button and other profile details are still visible.

Figure 19 Page d'invitation

Interface de messagerie

Si quelqu'un postule pour la colocation virtuelle, l'utilisateur peut, via l'interface de messagerie, accepter ou refuser le candidat, en plus de lui parler avant de prendre sa décision.

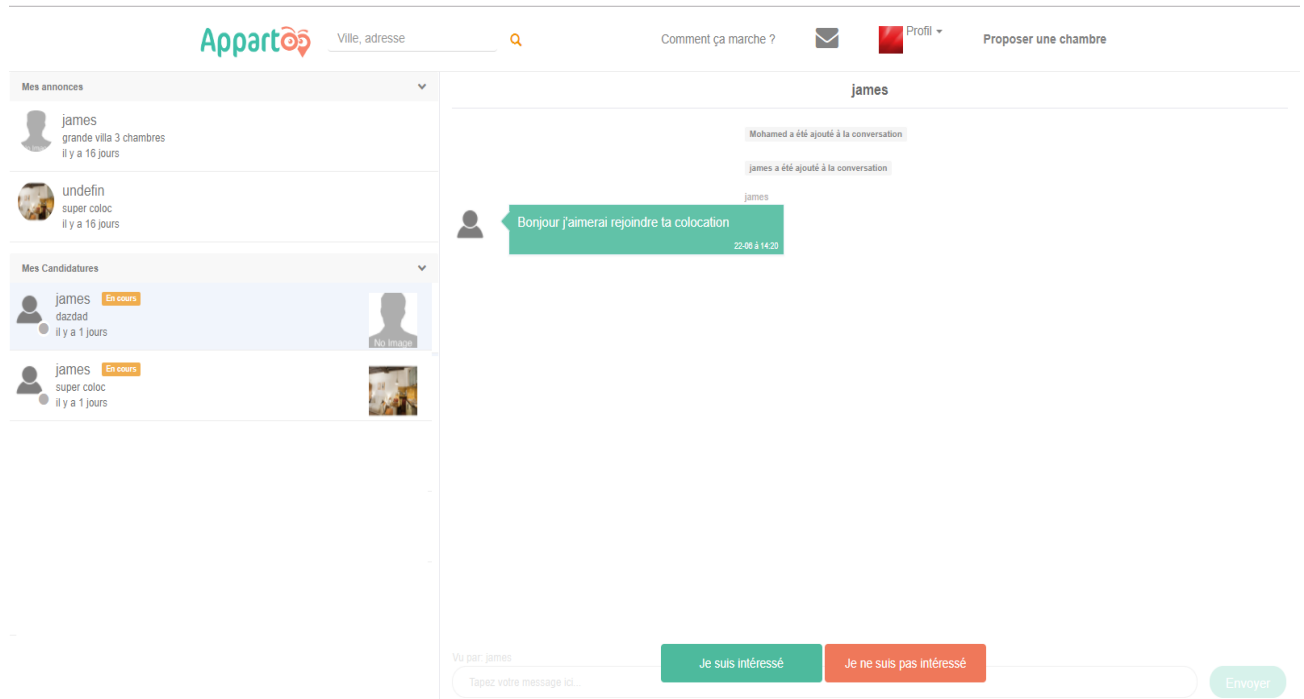


Figure 20 Interface de messagerie

Page Premium

Pour que les deux utilisateurs peuvent parler, l'un d'eux doit être premium

Appart'OS Ville, adresse Comment ça marche ? Profil [Proposer une chambre](#)

Choisissez la formule qui vous correspond :

Je déménage dans une semaine !!

7 jours

€ par jour
Soit €

Choisir

Choisissez 1 30 jours pour trouver une colocation !

30 jours

€ par jour
Soit €

Choisir

J'ai 90 jours pour trouver youpi !!!!!

3 mois

€ par jour
Soit €

Choisir

En devenant membre premium, vous pouvez :

Echangez avec tous les colocataires
 Consultez et répondez à tous vos messages
 Accédez aux coordonnées de tous les membres
 Augmentez vos chances d'être contactés

Figure 21 Interface de paiement

Interface Administrateur

Profils

L'administrateur peut consulter la liste des profils du site qui contiendra toutes les informations sur eux, ainsi que des actions.

Sonata Admin / Profile List											
<div> Filtres Ajouter </div>											
<input type="checkbox"/>	Photo	Nom et Prénom	Age	Telephone	Fb	Cherche	Dernière connexion ▲	Likes	En ligne	% Réponses	Actions
<input type="checkbox"/>		Ervitmonslip Jean-Philippe 5949227fb538f2@default.fr				oui	24/06/2017 10:57 37.165.219.216	0	Non	0 msg	Activé
<input type="checkbox"/>		Lambeets Chloé (F) 594e0cdb5b1c282@default.fr		0630507510		non	24/06/2017 10:55 37.165.219.216	0	Non		Activé
<input type="checkbox"/>		ym 594922b3c1cb812@default.fr				oui	24/06/2017 10:48 37.165.219.216	0	Non	0 msg	Activé
<input type="checkbox"/>		beslay 594922a5544527@default.fr				oui	24/06/2017 10:34 37.165.219.216	0	Non	0 msg	Activé
<input type="checkbox"/>		Buboix Marie 5949228b7701e4@default.fr				non	24/06/2017 10:32 37.165.219.216	0	Non		Activé
<input type="checkbox"/>		Denelain Paul (H) 594e0ce9dde5483@default.fr	26	0676278372		non	24/06/2017 08:55	0	Non		Activé

Figure 22 Interface Admin Profil

Annonces

L'administration pourrait opter de voir aussi les annonces

Sonata Admin / Roommate Group List

Filtres Ajouter

<input type="checkbox"/>	Image	Titre	Type	Créateur	Adresse	Créé le	Prix	Availability Starts	Réponses	Vues	Actions	Email Vérifié	Partagé
<input type="checkbox"/>		Amazing flat 5mn Champs Elysées	Appartement	Denelain Paul 0676278372	Paris 8e Arrondissement France	24/06/17 08:55	600	13/07/2015 Indéterminée	0 / 0	0	Activer Désactiver		non
<input type="checkbox"/>		Coloc Gare du Nord	Appartement	Lambeets Chloé 0630507510	Paris France	24/06/17 08:55	750	27/11/2016 Indéterminée	0 / 0	0	Activer Désactiver		non
<input type="checkbox"/>		Chambre Paris V	Appartement	VARDANIAN Tamara 066739130	Paris France	24/06/17 08:55	730	17/06/2017 Indéterminée	0 / 0	0	Activer Désactiver		non
<input type="checkbox"/>		Colocation 2pers. Denfert	Colocation	brasme marie 0681722194	Paris 75014 France	24/06/17 08:54	600	19/06/2017 Indéterminée	0 / 0	0	Activer Désactiver		non

Figure 23 Interface admin annonce

Ou bien visualiser les statistiques d'utilisations du site

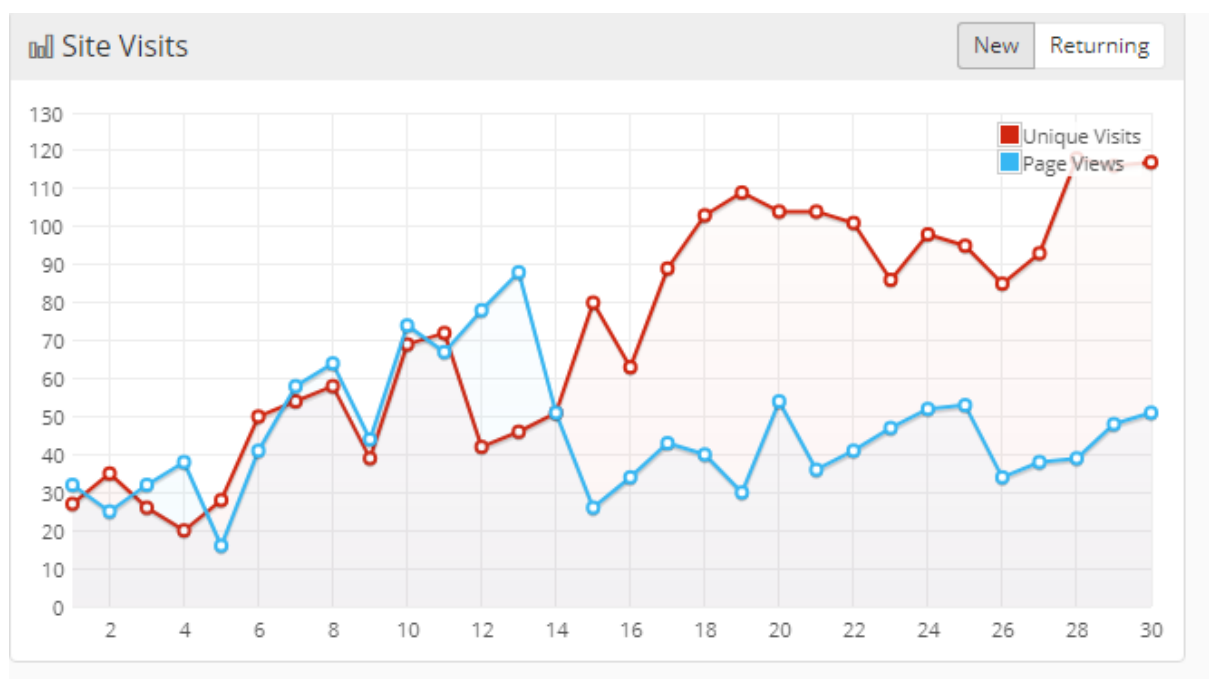


Figure 24Admin Statistique

Conclusion

Le présent rapport représente la synthèse du projet de fin d'études que j'ai effectué au sein de l'organisme Appartoo, et qui consistait à mettre en œuvre une application DE MATCHING entre colocataires.

Ce projet de fin d'études m'a donné l'opportunité de mettre en pratique différentes connaissances acquises durant notre cursus à l'Ecole Nationale des Sciences Appliquées de Fès, notamment l'architecture orientée service et les services web enseigné par mon encadrant M. Kenzi Adil.

En plus, c'était une expérience très enrichissante pour moi étant donné que j'ai pu maîtriser deux grands framework : AngularJS et Symfony, ce qui me sera d'une grande utilité pour ma carrière professionnelle. De plus, j'ai appris à utiliser plusieurs API reconnus tel que Facebook, Google Maps et FireBase, chose qui m'a permis d'améliorer mes compétences dans mon domaine. Aussi, l'utilisation de certains outils comme Trello et Slack pour la répartition des tâches et la communication m'a donné une nouvelle perspective sur l'organisation et son importance quant au gain du temps d'un projet.

Ceci a été possible suite à la formation dont j'ai bénéficié mais aussi grâce à la présence, l'encadrement et le dévouement des personnes qui m'ont entouré tout au long de cette période.

J'ai eu l'occasion de travailler dans un milieu professionnel et organisé, et l'adoption de la méthodologie agile m'a donné une nouvelle perspective sur le déroulement d'un projet.

Par ailleurs, j'ai amélioré mes capacités à travailler en équipe, communiquer et m'organiser avec mes collègues, et ce, en m'adaptant à un environnement qui m'est nouveau, dans un pays étranger et dont la culture diffère de la mienne.

L'un des avantages de faire un stage dans une startup, malgré la taille de l'équipe, c'est le fait de participer à tous les processus de prise de décision et brainstorming, tout en faisant entendre son avis et, par conséquent, avoir un grand poids dans l'entreprise.