

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.

Résumé

Le présent rapport est la synthèse de mon stage de fin d'études effectué au sein de la société Appartoo Paris une startup dans le secteur de l'immobilier.

Au cours de ce stage, j'ai rejoint une équipe de jeune développeur, dans le cadre d'un projet interne. Il s'agit de la mise en place d'une application de matching entre colocataires qui se compose d'un service web Restfull en Symfony et un FrontEnd en AngularJS.

Le but de cette application est de garantir aux utilisateurs qui cherchent une colocation, d'habiter avec des personnes avec qui ils sont compatibles en retrouvant ceux qui possèdent le même caractère et habitudes.

Cela implique de mettre en œuvre un algorithme de comparaison et de classement des résultats par compatibilité, cet algorithme a été fait grâce au moteur de recherche elasticsearch. Afin de réaliser ce projet, mon équipe a adopté une méthodologie Agile, qui est scrum, qui repose sur un cycle itératif et incrémental.

Cette méthode nous a apporté une grande organisation et un processus de développement optimal.

Ce projet réalisé dans une durée de cinq mois, clôture ma dernière année d'études au sein de l'ENSA de Fès.

Abstract

This report is a synthesis of my end of study internship carried out within the company Appartoo Paris, which is a startup in real estate.

During this internship, I had the pleasure of joining a team of young developers as part of an internal project. This project consists of implementing a web application which purpose is matching people with their best possible roommate. This application is divided in two parts, the first one is a Restful web service with Symfony, and the second is a FrontEnd with AngularJS.

The purpose of this application is to guarantee its users who are looking to share a house, that they are going to live with people who are compatible with their personality.

This functionality is made by comparing the nature of each one of its users, and executing an algorithm that will retrieve the best person to be with, based on habits and traits. This algorithm is executed by the search engine Elasticsearch.

To achieve this project, my team and I followed and Agile method, which is Scrum, which rely on having an iterative and incremental cycle. This method gave us a lot of benefits in organization and an optimal development process.

This project was achieved for five months, closes my last year of studies in the ENSA of FES.

Liste des tableaux

Tableau 1 Fiche Technique d'Appartoo	3
Tableau 2 : rôles scrum du projet	9
Tableau 3 Définitions	15
Tableau 4 Acteurs du projet	17
Tableau 5 : Descriptif du cas :s'inscrire	20
Tableau 6 Descriptif du cas : Bannir un compte	21
Tableau 7 Descriptif du cas : Générer la liste des utilisateurs compatibles	22
Tableau 8 Descriptif du cas : Discuter avec un utilisateur	25
Tableau 9 Descriptif du cas : postuler à une annonce	26
Tableau 10 Descriptif du cas : Accepter Candidat	27
Tableau 11 Descriptif du cas : Envoi de mail	29
Tableau 12 Descriptif du cas : paiement	31
Tableau 13 Description des classes	37

Listes des figures

Figure 1 Logo d'Appartoo	3
Figure 2 Organigramme d'Appartoo	4
Figure 3 Architecture Scrum	9
Figure 4 Logo de Trello	10
Figure 5 : Colonnes Trello du projet	10
Figure 6 : Colonne Tâche du Sprint	11
Figure 7 : Colonne en cours Trello.....	11
Figure 8 : Colonne prêt à tester Trello	11
Figure 9 Logo de Slack	12
Figure 10 Diagramme de Gantt	13
Figure 11 Logo de Gantt Project	13
Figure 12 Diagramme de paquetage.....	18
Figure 13 Diagramme de cas d'utilisation : Gestion de compte	19
Figure 14 Diagramme de séquence : inscription	20
Figure 15 Diagramme de séquence : Bannir	21
Figure 16 Diagramme de cas d'utilisation : Matching.....	21
Figure 17 Diagramme de séquence : Matching.....	23
Figure 18 Diagramme de cas d'utilisation : gestion d'annonces.....	23
Figure 19 Diagramme de cas d'utilisation : Messagerie.....	24
Figure 20 Diagramme de séquence : Messagerie	25
Figure 21 Diagramme de cas d'utilisation : Gestion candidats	26
Figure 22 Diagramme de séquence : Postuler	28
Figure 23 Diagramme de cas d'utilisation : Mailing	29
Figure 24 Diagramme de séquence : Mailing	30
Figure 25 Diagramme de cas d'utilisation : Paiement	30
Figure 26 Diagramme de séquence : "Paiement.....	32
Figure 27 Diagramme d'activités : chercheur.....	33
Figure 28 Diagramme d'activités : proposeur	34
Figure 29 Diagramme d'activités : Administrateur	35
Figure 30 : Diagramme de Classe	36
Figure 31 Architecture MVC	39
Figure 32: Architecture du projet	41
Figure 33 : Routes du service Web	51
Figure 34 : Page d'accueil.....	55
Figure 35 Page d'inscription.....	55
Figure 36 : Page d'informations du profil.....	56
Figure 37 Page de création de colocation virtuelle	57
Figure 38 Page de recherche	57
Figure 39 Page du profil.....	58
Figure 40 Page d'invitation.....	58

Figure 41 Interface de messagerie.....	59
Figure 42 Interface de paiement.....	60
Figure 43 Interface ADmin Profil	60
Figure 44 Interface admin annonce	61
Figure 45Admin Statistique	61

Table des matières

Introduction générale.....	1
Chapitre 1	2
I- Présentation de l'organisme d'accueil.....	3
1.1 - Fiche technique Appartoo :.....	3
1.2 - Services :.....	4
1.3 - L'équipe :.....	4
II- Présentation du projet	4
2.1 - Context général.....	4
2.2 - Etude de l'existant	5
2.3 - Problématique.....	6
2.4 - Solutions	6
2.5 - Objectif du projet.....	7
III- Conduite de projet	8
3.1 - Méthodologie de développement.....	8
3.2 - Application de Scrum sur notre projet.....	9
3.3 - Planning prévisionnel du projet.....	12
3.4 - Outils de Gestion de projet :	13
Chapitre 2	14
I- Spécifications fonctionnelles du système.....	15
1.1 Définitions.....	15
1.2 Fonctionnalités globales du site	16
II- Identification des acteurs	17
III- Décomposition du système	18
3.1 Diagramme de paquetage des cas d'utilisation.....	18
3.2 Diagrammes de séquence et de cas d'utilisation.....	18

3.3 Diagrammes d'activités.....	33
3.4 Diagramme de classe métier	36
Chapitre 3	38
I- Structure du projet.....	39
1- Back end : Service Web.....	39
2- Front End : Site Web	40
3- Front End Mobile.....	40
4- Communication entre le service WEB et le front End	41
Chapitre 4	42
I- Technologies et outils de développement	43
1- Outils de développement	43
2- Outils serveurs	44
3- Web Service (Back end)	45
4- Front End	48
II- Répartition des tâches	50
III- Déroulement de Mon stage.....	50
1- Utilisation de Git	50
2- Service Web.....	51
3- Front End	52
IV- Démonstration	54
Interface Administrateur	60
Conclusion.....	62
Références	63

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.



Figure 1 Logo d'Appartoo

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.

1.1 - Fiche technique Appartoo :

Raison sociale	Appartoo
Forme juridique	SASU Société par actions simplifiée à associé unique
Année de création	2015
Siege social	APPARTOO, 176 AVENUE CHARLES DE GAULLE 92200 NEUILLY SUR SEINE
Siret	81263134900014
Téléphone	+33 638963607
FAX	+33 73233607
Président	M Jimmy CROMBEZ

Tableau 1 Fiche Technique d'Appartoo

1.2 - 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

1.3 - 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.

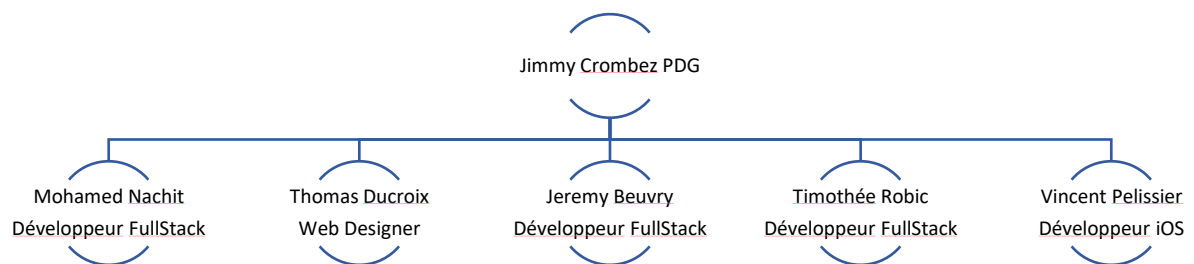


Figure 2 Organigramme d'Appartoo

II- Présentation du projet

2.1 - Context général

On vise dans cette partie à présenter le cadre logique du projet pour mieux structurer et formuler ensuite les objectifs attendus.

QUI ?

Nachit Mohamed, élève ingénieur à l'Ecole Nationale des Sciences Appliquées de Fès, actuellement en troisième année du cycle ingénieur, filière génie informatique.

QUOI ?

Il s'agit d'un projet de fin d'études intitulé « **DEVELOPPEMENT D'UNE APPLICATION WEB DE MATCHING ENTRE COLOCATAIRES** », que j'effectue sous l'encadrement académique du Professeur Kenzi Adil, et l'encadrement professionnel de M. Crombez Jimmy au sein d'Appartoo.

QUAND ?

Le stage que j'ai effectué est étendu sur une durée de 5 mois, il s'est déroulé du 13 février jusqu'au 13 Juillet 2017.

OÙ ?

Ce projet a eu lieu au sein de l'organisme Appartoo, qui est une société spécialisée dans le secteur immobilier.

POURQUOI ?

Ce projet de fin d'études avait pour cibles principales l'obtention du diplôme d'ingénieur d'état et l'acquisition des bases de ce métier en mettant en pratique le savoir et le savoir-faire qui m'ont dument été enseignés tout au long des trois dernières années à fin que je puisse être prêt à intégrer le marché du travail.

COMMENT ?

En mettant en œuvre mes connaissances dans le domaine d'informatique, ce que j'ai acquis personnellement grâce à mes professeurs ainsi que ce que j'ai appris seul et durant mes stages antérieurs.

2.2 - Etude de l'existant

En France, les personnes en recherche de logement n'ont pas beaucoup de choix sur les méthodes à suivre pour trouver un logement, leurs seuls moyens sont :

Passer par une agence :

Un agent immobilier est mandaté par un ou des bailleurs qui souhaite(nt) louer leur bien : il doit mettre en valeur un logement et trouver le bon locataire.

Si une personne conclus un contrat de location (bail) avec cet agent immobilier, elle sera amenée à payer des frais dits d'agence qui inclus

Les frais des visites, la constitution du dossier, la rédaction du bail, l'établissement d'état des lieux d'entrée.

Consulter les Sites d'annonce

De nombreux sites d'annonces immobilières existent sur Internet. Ces sites contiennent des annonces de logement publiés par des propriétaires, auxquelles les gens peuvent postuler. Les propriétaires et les chercheurs de logement se dirigent vers ces sites afin d'éviter de payer les frais d'agences.

Ces deux méthodes contiennent une étape commune, c'est l'étude du dossier.

Avant la signature du bail, le propriétaire (ou l'agent immobilier) peut exiger certains documents au candidat locataire, voire à la personne qui se porte caution pour lui. Ces documents permettent de prouver l'identité des personnes concernées et de s'assurer de leur niveau de ressources.

En effet, Avant de louer un logement, le propriétaire a le droit de s'informer sur l'identité et la solvabilité de son futur locataire.

À ce titre, il peut réclamer plusieurs documents comme les 3 dernières quittances de loyer, des bulletins de salaires. Et exiger assez de critères, comme le fait d'avoir un salaire supérieur ou égal à 3 fois le prix du loyer, ou bien exiger une caution importante comme garantie.

2.3 - Problématique

La majorité des chercheurs de locations et colocations sont des étudiants et des jeunes qui n'ont pas forcément les moyens pour remplir les exigences des propriétaire, ce qui rend la mission de trouver un logement très difficile pour eux, et peut même les pousser à chercher des logements illégalement (sous location) ou bien falsifier leurs informations pour faire croire aux propriétaire qu'ils sont éligibles.

En plus, l'un des plus grands problèmes des agences et des sites d'annonces c'est qu'ils ne garantissent pas l'entente des colocataires, les personnes qui réussit à débaucher un logement se retrouve partageant un toit avec des colocataires avec qui ils ne s'entendent pas.

En effet, la différence entre les personnalités et les habitudes peuvent créer une atmosphère de vie difficile, par exemple : fumeur/ non-fumeur, quelqu'un qui aime le bruit et un autre qui préfère le calme, quelqu'un qui aime parler et un autre plus réservé.

Par conséquent, plusieurs colocations ont été séparé et le chercheur se retrouve dans la position de départ, sans savoir est ce que son prochain colocataire va être compatible avec lui, ce qui provoque un sentiment de peur chez les chercheurs comme les propriétaires, car ils ne sont pas sûr que le contrat va durer.

2.4 - Solutions

La solution proposée par Appartoo compte résoudre les deux grandes problématiques en même temps.

Afin de remédier aux problèmes du manque de moyen pour les chercheurs, et offrir aux propriétaires des candidats éligibles et garantir le paiement de leur loyer, Appartoo va offrir une plateforme qui va aider les gens à former des colocations avant même de chercher un appartement, par conséquent, le fait de chercher en groupe veut dire une force financière plus importantes que quand une personne cherche seul, ce qui va encourager le propriétaire à accepter la candidature.

D'un autre côté, cette plateforme contiendra un système de matching, qui va calculer automatiquement le degré de compatibilité entre les personnes et proposer les candidats les plus adaptés et par conséquent, garantir une colocation idéale sans problème et sans dispute, ce qui va résoudre le problème de la rupture des colocs et ainsi mettre en aise les propriétaires ainsi que les chercheurs d'appartement.

2.5 - Objectif du projet

Ce projet a comme premier objectif la mise en place d'une plateforme qui propose aux chercheurs de logements des colocataires potentiels compatible avec leur personnalité, ainsi qu'un logement adapté à leur moyen ; et aux propriétaires des candidats qui correspondent à leurs critères et exigences, tout en évitant les problèmes liés à l'entente dans la vie quotidienne des colocataires.

L'application proposera aux chercheurs de logement les fonctionnalités suivantes :

1. Trouver les personnes les plus compatibles avec leurs personnalités pour devenir colocataire
2. Discuter avec leurs colocataires potentiels
3. Consulter une liste d'annonces d'appartement dans une région choisie
4. Postuler à ces annonces seul ou bien avec les personnes avec qui ils sont compatibles

L'application proposera également des fonctionnalités pour les propriétaires :

1. Déposer des annonces gratuitement
2. Consulter une liste des profils compatibles avec leurs critères
3. Gérer les candidatures de leurs annonces

III- Conduite de projet

3.1 - Méthodologie de développement

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, notre équipe a décidé d'adopter l'approche Agile.

a. 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.)

b. 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.

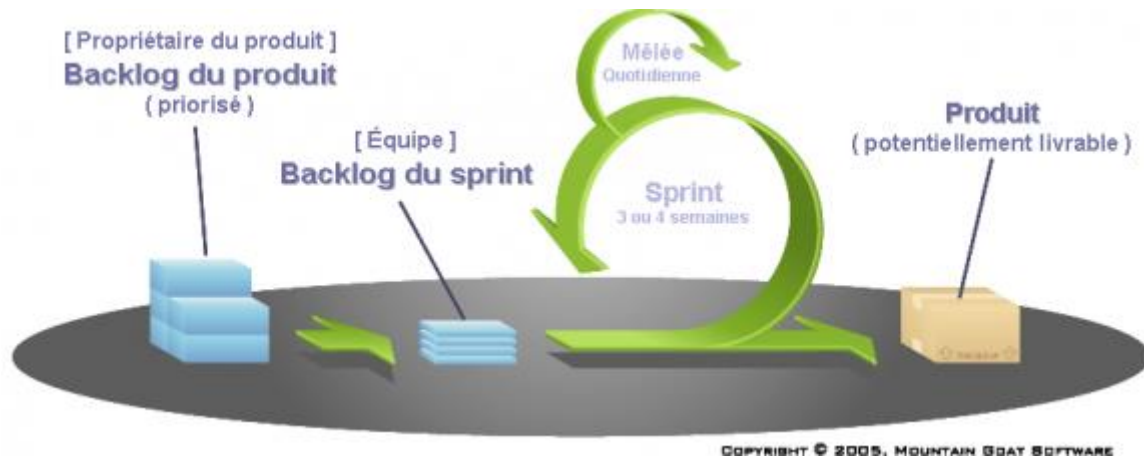


Figure 3 Architecture Scrum

3.2 - Application de Scrum sur notre projet

La durée d'un sprint a été décidée comme étant égale à comme **1 semaine**. Cette décision a été prise en tenant compte de la taille du projet, de l'équipe et du temps attribué aux projets.

Les rôles scrum des différents acteurs sont comme suit :

Product Owner (Client)	M Jimmy Crombez
Scrum Master	Vincent Pélissier
Equipe de développement	Mohamed Nachit Jeremy Beuvry Timothée Robic Thomas Ducroix

Tableau 2 : rôles scrum du projet

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



Figure 4 Logo de Trello

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.

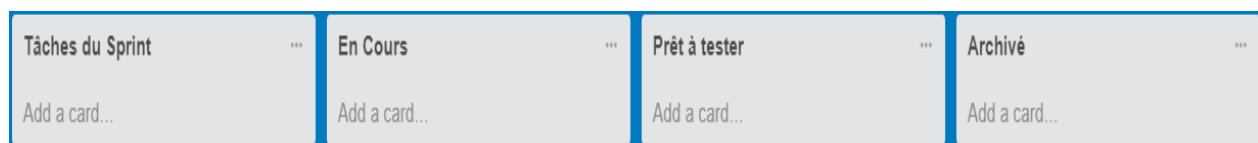


Figure 5 : Colonnes Trello du projet

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

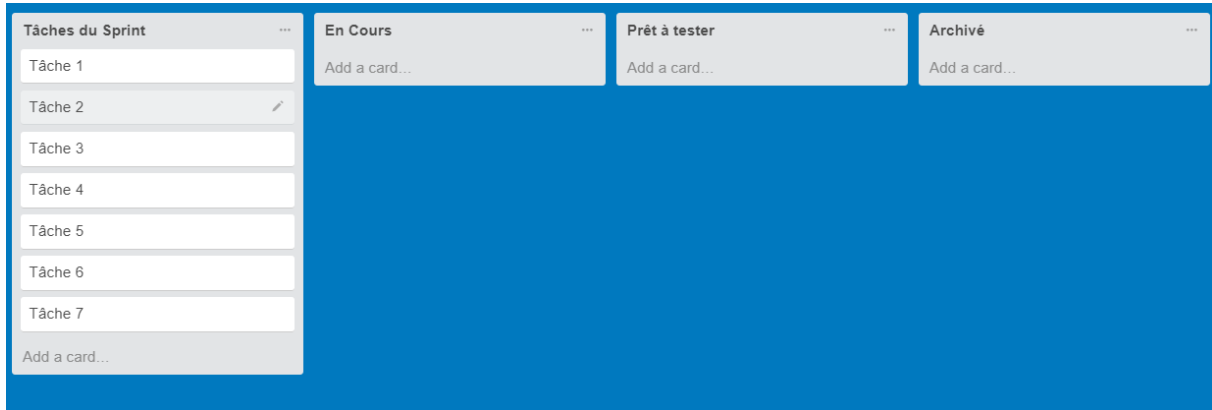


Figure 6 : Colonne Tâche du Sprint

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

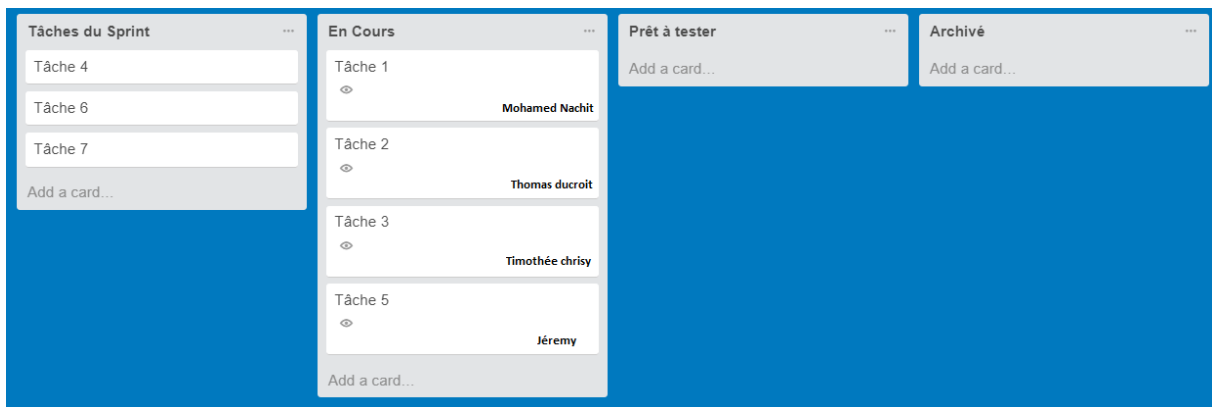


Figure 7 : Colonne en cours Trello

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

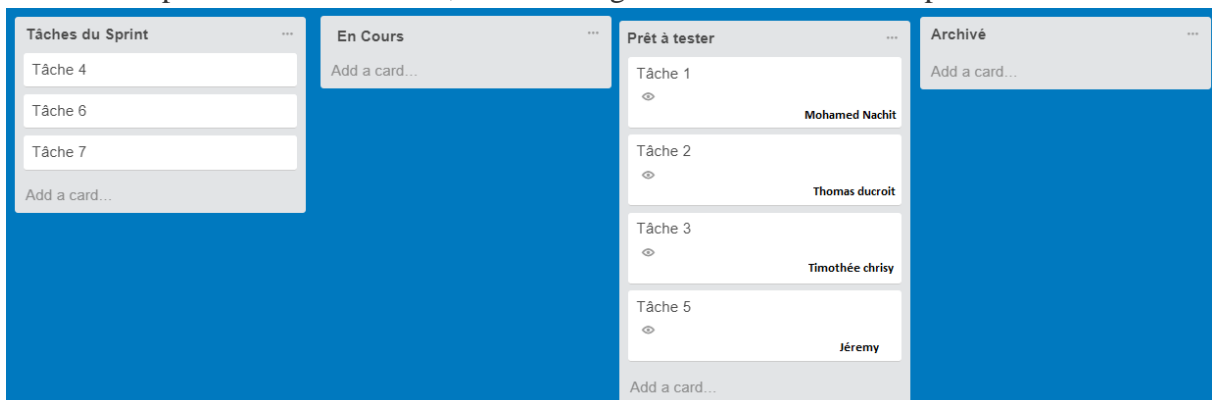


Figure 8 : Colonne prêt à tester Trello

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



Figure 9 Logo de Slack

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.)

3.3 - Planning prévisionnel du projet

L'objectif du planning est de déterminer les étapes du projet et le timing. Ce planning joue un rôle primordial pour la réalisation et le suivi du projet, il est établi dans le début de chaque projet afin de suivre le bon déroulement de chaque tâche. L'élaboration du planning s'est basée sur l'hypothèse suivante :

- Respect des périodes de validation
- Le planning peut subir des modifications suivant les impératifs du projet

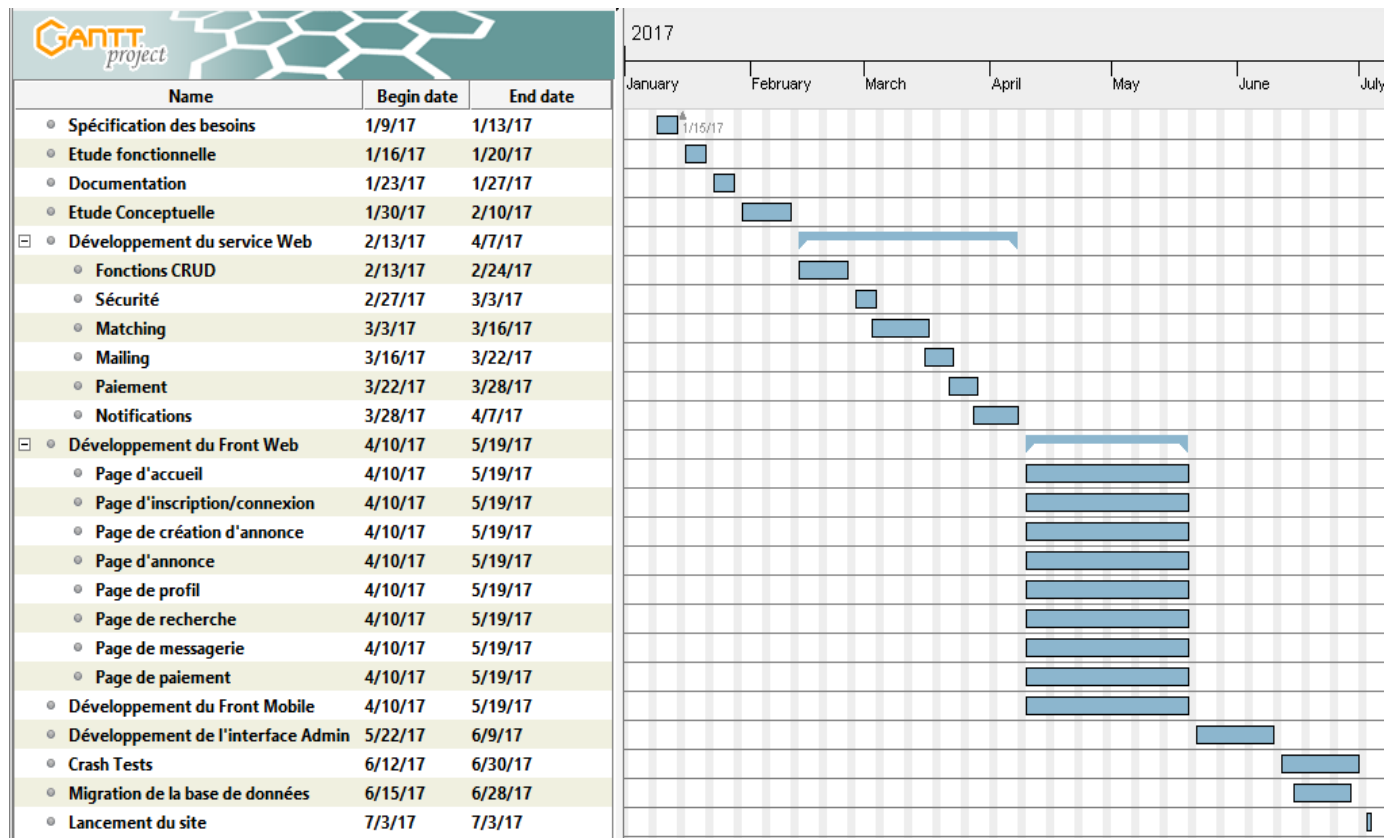


Figure 10 Diagramme de Gantt

3.4 - Outils de Gestion de projet :



Figure 11 Logo de Gantt Project

Gantt Project est un logiciel de gestion de projet, il permet de planifier les projets et les ressources, et d'assurer le suivi des projets durant leur réalisation. Cet outil permet ainsi au chef de projet d'assurer une gestion de projet professionnelle, conforme à l'état de l'art, et ainsi garantir le respect des délais.

Ces principales fonctionnalités :

- Création des tâches d'un projet
- Répartition des tâches sur des membres
- Gestion des priorités et des couts

Chapitre 2

Analyse et conception

Ce chapitre expliquera plus en détail les parties de notre projet ainsi que les fonctionnalités qu'il doit contenir, en se basant sur les acteurs possibles comme référence, en plus de la conception en utilisant le langage UML

I- Spécifications fonctionnelles du système

1.1 Définitions

Utilisateur	Personne physique qui utilise le site Appartoo afin de profiter de ces services
Chercheur	Utilisateur qui cherche un logement, ou bien des colocataires potentiels avec qui il aimerait chercher où habiter, un chercheur possède plusieurs badges.
Badge	Entité qui représente un caractère de personnalité d'un chercheur
Matching	L'application d'un processus de calcul afin de retrouver le colocataire idéal pour un utilisateur en se basant sur ces badges
Propriétaire	Utilisateur qui a publié une annonce d'un bien immobilier afin de retrouver une personne pour le lui louer.
Annonce	Désigne une offre consultable par tous les utilisateurs, une annonce peut désigner un appartement ou bien une colocation virtuelle. Une annonce peut avoir des membres qui habite dans le logement ou bien qui font partie d'une colocation virtuelle.
Logement	Entité immobilière qui regroupe plusieurs personnes vivant sur le même toit.
Colocation virtuelle	Entité qui regroupe plusieurs personnes qui cherche à louer ensemble et qui n'ont pas encore un appartement
Postuler	Envoyer un message aux propriétaires d'une annonce pour lui montrer l'intérêt sur celle ci
Conversation débloquent	Une conversation est débloquée quand l'un des participants de cette conversation est un membre premium
Membre Basique	Utilisateur qui n'a pas encore payé pour débloquent toutes les fonctionnalités du site
Membre Premium	Un utilisateur qui débloquent les fonctionnalités du site payante

Tableau 3 Définitions

1.2 Fonctionnalités globales du site

a. Inscription et connexion

Un utilisateur doit avoir la possibilité de s'inscrire sur l'application : soit avec son email ou bien son Facebook. En plus, il doit pouvoir modifier ses informations n'importe quel moment, ainsi que de désactiver son compte. Après son inscription, l'utilisateur peut vérifier son email par biais d'un email de validation qui lui sera envoyé.

Ce compte est utilisé afin de pouvoir créer des annonces, postuler aux annonces des autres utilisateurs ainsi que recevoir leurs messages.

b. Matching

Pour les utilisateurs qui cherchent des colocataires avec qui habiter, la plateforme offre une interface de recherche des profils qui correspondent à la personnalité de l'utilisateur courant, l'utilisateur peut ajouter d'autres critères s'il en décide, cette fonctionnalité permet de comparer les badges de tous les utilisateurs du site et les afficher les profils qui ont le plus de badge en commun par ordre de pertinence.

En plus de chercher des profils compatibles, le matching permet aussi de rechercher des annonces de logement qui correspondent aux critères de l'utilisateurs.

Finalement, cette interface sert aussi a aider les propriétaires à retrouver des candidats éligibles à leur logement.

c. Gestion des annonces

Chaque utilisateur connecté pourra publier son annonce sur le site, celle-ci sera visible pour tous les utilisateurs, le propriétaire de l'annonce pourrait alors choisir de la modifier ou bien la désactiver pour la rendre invisible.

d. Gestion des candidats

Un utilisateur peut postuler pour une annonce de logement ou bien de colocation virtuelle afin de la rejoindre.

Le propriétaire de l'annonce a la possibilité d'accepter ou bien refuser les candidats.

Les candidats acceptés seront ajoutés à l'annonce en tant que membre et peuvent parler entre eux en privé.

e. Messagerie

L'application offre aux utilisateurs une interface de messagerie qui leur permettrai de discuter en temps réel entre eux, afin de donner l'opportunités aux détenteurs d'annonces de mieux connaitre les candidats et leurs moyens pour faciliter leurs choix de les accepter à les rejoindre.

f. Paiement

Chaque utilisateur aura la possibilité de promouvoir son compte en un compte premium,

Le site offrira une variété de forfait auquel l'utilisateur pourrait choisir selon sa situation,

Le paiement sera effectué grâce à une interface de paiement en ligne, où l'utilisateur renseignera ces informations bancaires et sera débité selon le forfait qui a choisi.

Un compte premium offrira à cet utilisateur de nouvelles fonctionnalités dans le site et plus de liberté.

g. Notifications

Chaque utilisateur sera notifié par les changements sur le site qui le concerne, par des alertes

Qui lui sera visible sur le site, celui-ci offrira une interface pour pouvoir consulter ces notifications afin de rester toujours à jour et réagir le plus vite possible aux événements.

h. Mailing

L'application va offrir un système de mailing qui se chargera d'envoyer des mails dans le courriel des utilisateurs abonné, ces mails se déclencheront dans des événements précis, comme l'ajout d'une annonce qui correspond aux critères de l'utilisateur ou bien le fait de recevoir un message. Ce qui permettrait à l'utilisateur de recevoir des informations importantes même s'il ne consulte pas le site.

II- Identification des acteurs

Dans cette partie nous allons identifier les acteurs du système informatique, ce sont les entités qui vont interagir avec notre application. On retrouve les acteurs suivants :

Acteur	Description
Administrateur	L'administrateur est chargé de la gestion des utilisateurs et des annonces, des demandes. Il observe toute les activités qui se produit sur l'application
Propriétaire	Utilisateur qui possède un appartement et qui aimerait le louer
Chercheur	Utilisateur qui cherche un logement, ou bien des personnes avec qui il aimerait habiter
Système	On signifie par système la source de toutes les actions qui proviennent de notre application vers les utilisateurs

Tableau 4 Acteurs du projet

III- Décomposition du système

Dans un système complexe, il est mieux de le diviser par module de familles des fonctions afin de simplifier l'analyse de celui-ci

3.1 Diagramme de paquetage des cas d'utilisation

Un diagramme de packages est un diagramme UML qui fournit une représentation graphique de haut niveau de l'organisation de l'application, et aide à identifier les liens de dépendance entre les packages. Le diagramme de paquetage de la figure représente l'ensemble des packages constituant notre application, les flèches signifie les dépendances entre les modules :

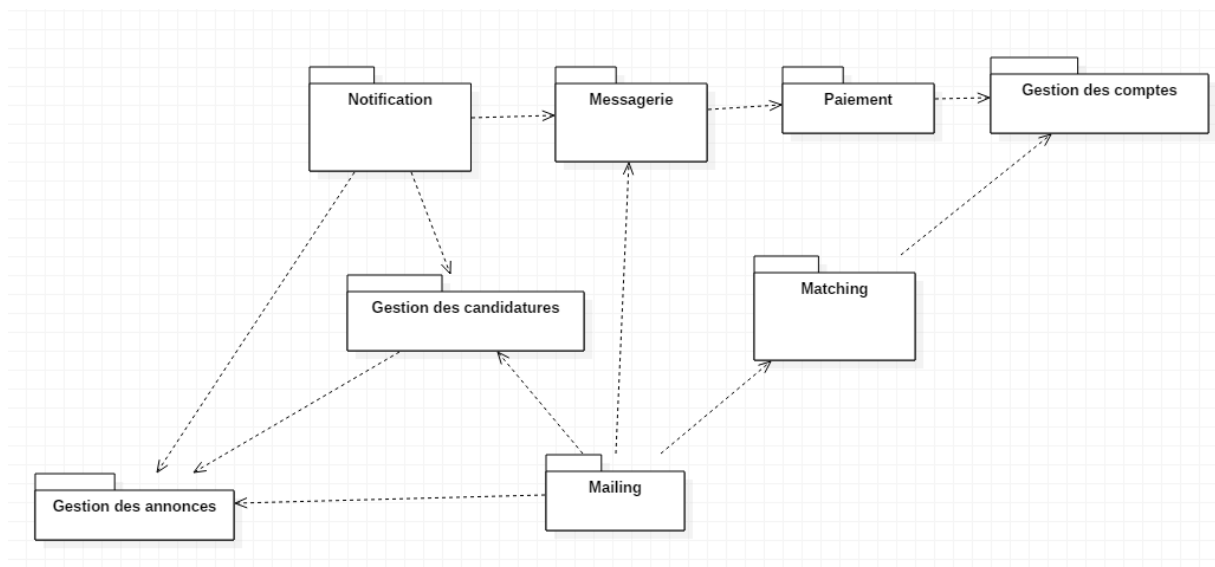


Figure 12 Diagramme de paquetage

3.2 Diagrammes de séquence et de cas d'utilisation

a. Module de Gestion de compte :

Ce module a pour objectif la gestion de toutes les informations du compte d'un utilisateur, la modification et la suppression des informations, la désactivation du compte ainsi le bannissement d'un compte.

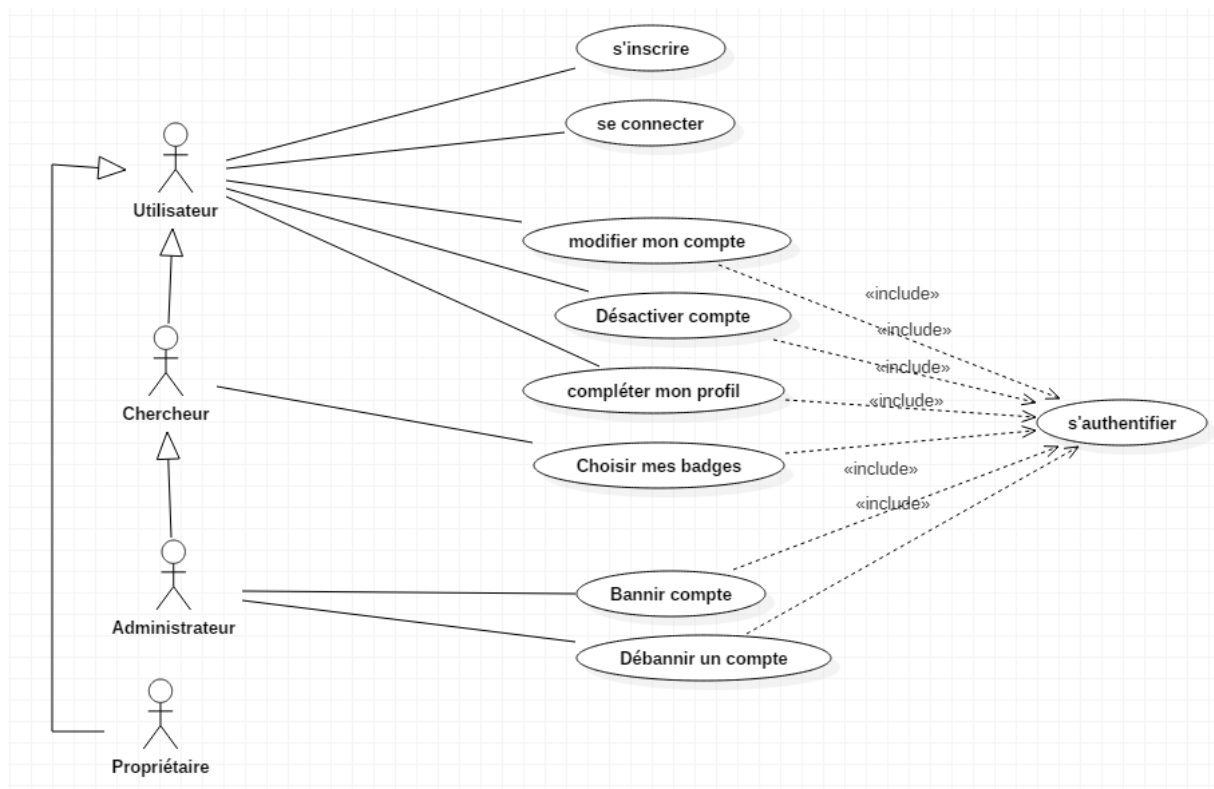


Figure 13 Diagramme de cas d'utilisation : Gestion de compte

Descriptif du cas d'utilisation « s'inscrire »

Cas d'utilisation	S'inscrire
Acteurs	Utilisateur
Description	Ce cas d'utilisation permet aux utilisateurs de créer un nouveau compte pour pouvoir bénéficier des services offerts par l'application
Précondition	Visiter le site ou l'application mobile.
Scénario nominal	1- Arrive sur une page du site 2- Le système lui envoie le formulaire d'inscription 3- L'utilisateur saisit son nom, prénom, email et mot de passe 4- Le système vérifie les données saisies ; 5- Le système connecte l'utilisateur automatiquement au site 6- L'utilisateur accède à son espace personnel.
Scénario alternatif	a- Le système envoie un message d'erreur exprimant qu'il existe un compte qui utilise le même mail saisi par l'utilisateur ; b- Le système renvoie le formulaire d'inscription pour modifier l'adresse mail ;

Post-condition	Les informations du nouveau compte utilisateur sont enregistrées dans la base de données du système.
----------------	--

Tableau 5 : Descriptif du cas :s'inscrire

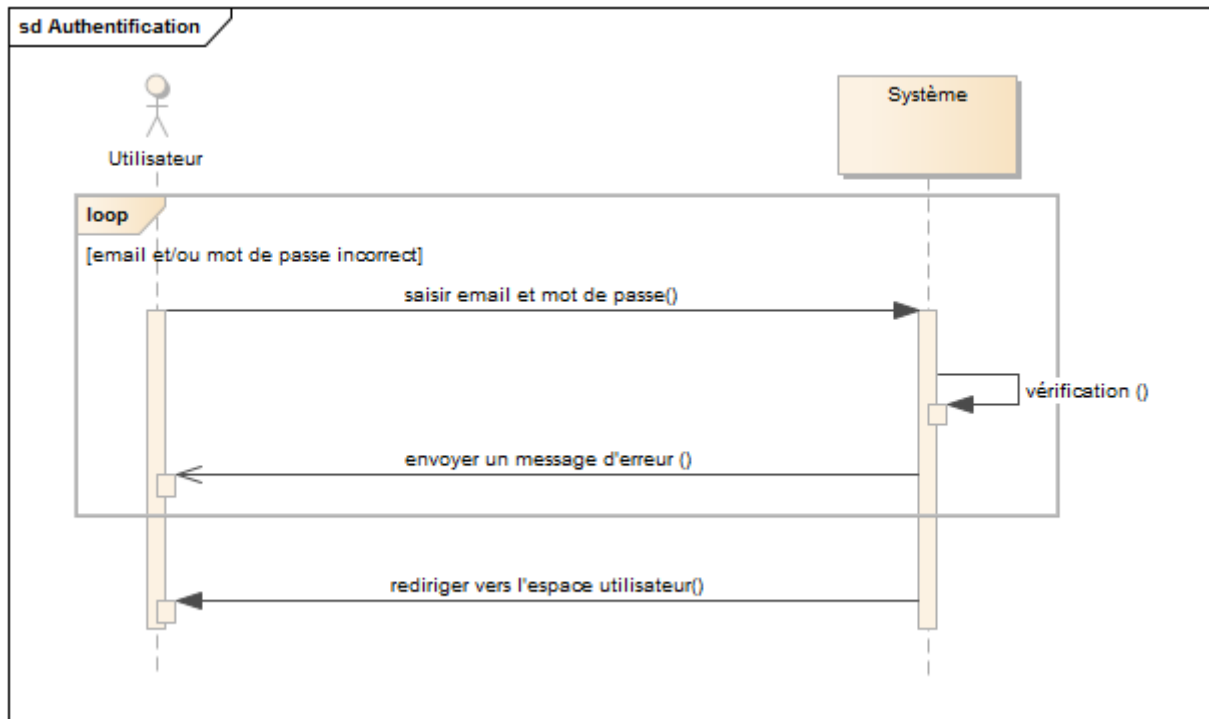


Figure 14 Diagramme de séquence : inscription

Descriptif du cas d'utilisation « Bannir compte »

cas d'utilisation	Bannir compte
Acteurs	Administrateur
Description	Ce cas d'utilisation permet à l'administrateur du site de bannir un utilisateur, généralement pour infractions des règles ou de fraudes.
Précondition	Se connecter avec un compte Admin
Scénario nominal	1- L'administrateur remarque une annonce frauduleuse 2- L'administrateur clique sur le bouton bannir le propriétaire de l'annonce 3- Le système change le statut du compte de l'utilisateur et détruit sa session 4- Le système désactive toutes les annonces de cet utilisateur 5- System informe l'administrateur que l'utilisateur est banni
Scénario alternatif	L'utilisateur est déjà banni Le système propose à l'administrateur de lui pardonner

Post-condition	L'utilisateur est déconnecté du site et ne peut plus y accéder.
----------------	---

Tableau 6 Descriptif du cas : Bannir un compte

Diagramme de séquence

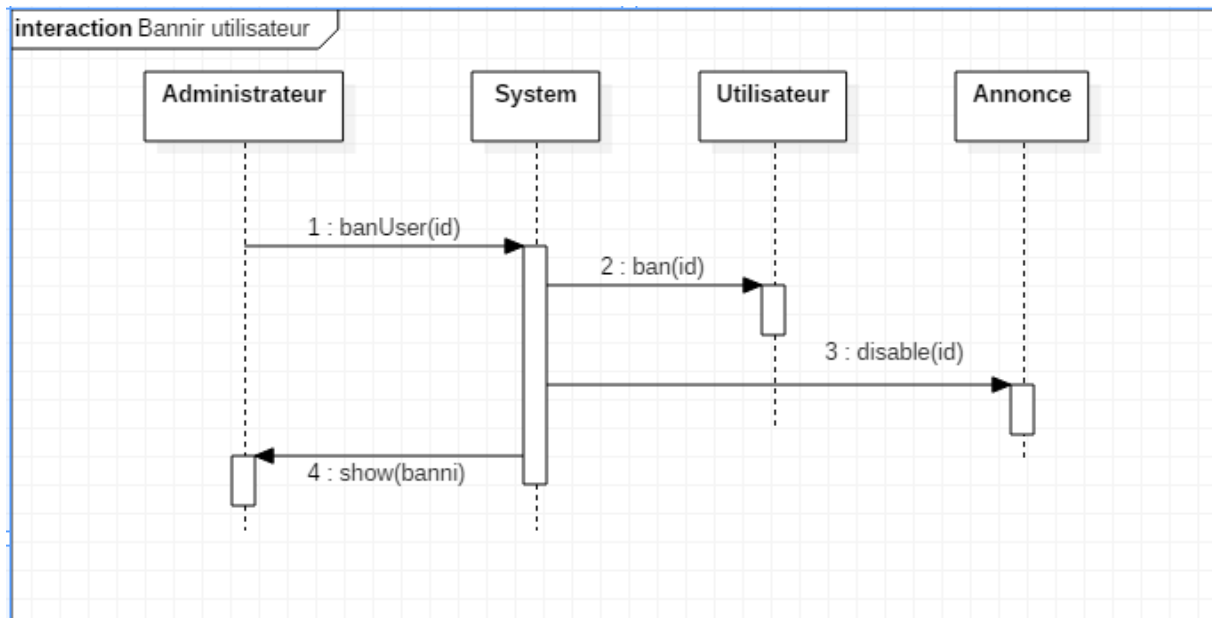


Figure 15 Diagramme de séquence : Bannir

b. Module de Matching :

Ce module permet la génération des profils les plus adaptés selon la nature du profil et de ces informations

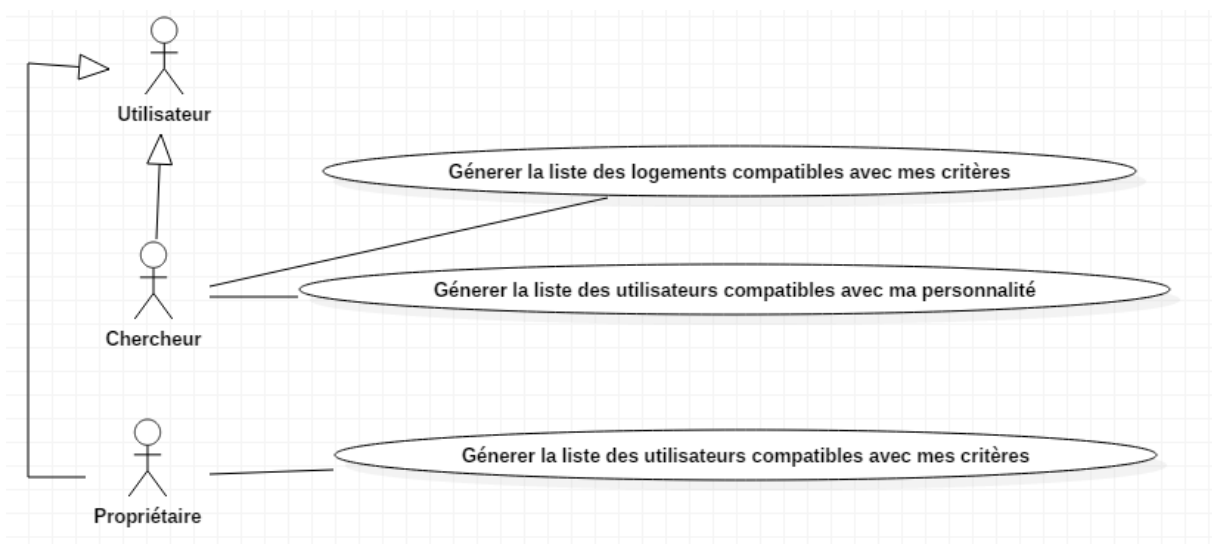


Figure 16 Diagramme de cas d'utilisation : Matching

Cas d'utilisation	Générer la liste des utilisateurs compatibles
Acteurs	Chercheur
Description	Ce cas d'utilisation permet aux utilisateurs de générer une liste de profils qui correspondent à leurs personnalités
Précondition	Etre connecté avec un compte Avoir choisis mes badges
Scenario nominal	<ol style="list-style-type: none"> 1- L'utilisateur accède à la page du matching 2- L'utilisateur clique sur générer les profils 3- Le système reçoit les traits de personnalité de l'utilisateur et les envoie aux systèmes de recherche qui compare les badges avec ceux figurant dans la base de données 4- Le système de recherche renvoie les résultats classés au système 5- Le système affiche les résultats sur l'écran 6- L'utilisateur consulte les profils compatibles avec lui
Scenario alternatif	<ol style="list-style-type: none"> a- L'utilisateur n'avait pas choisis ces badges aux préalables b- Le système ne trouve pas de résultats c- Le système demande à l'utilisateur de renseigner plus de critères
Post-condition	Demander à l'utilisateur s'il est satisfait des résultats

Tableau 7 Descriptif du cas : Générer la liste des utilisateurs compatibles

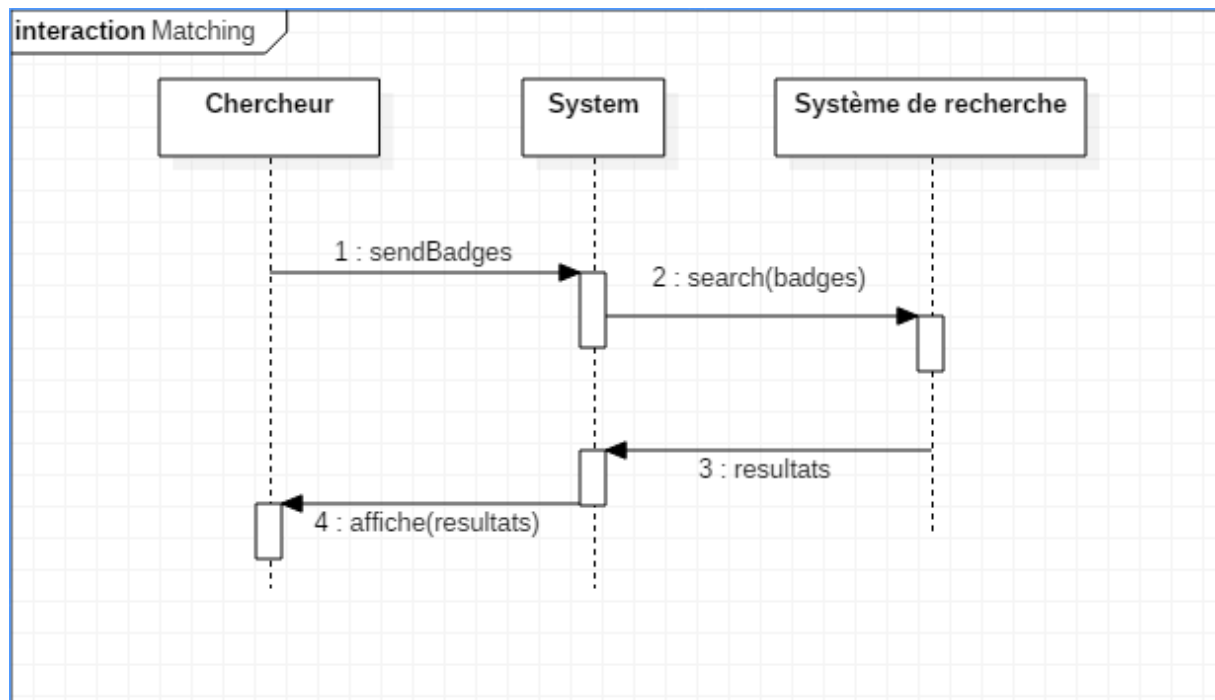


Figure 17 Diagramme de séquence : Matching

Module de gestion d'annonces :

Permet aux utilisateurs de mettre en ligne une annonce dans le site, ainsi que de la créer, modifier et de la désactiver.

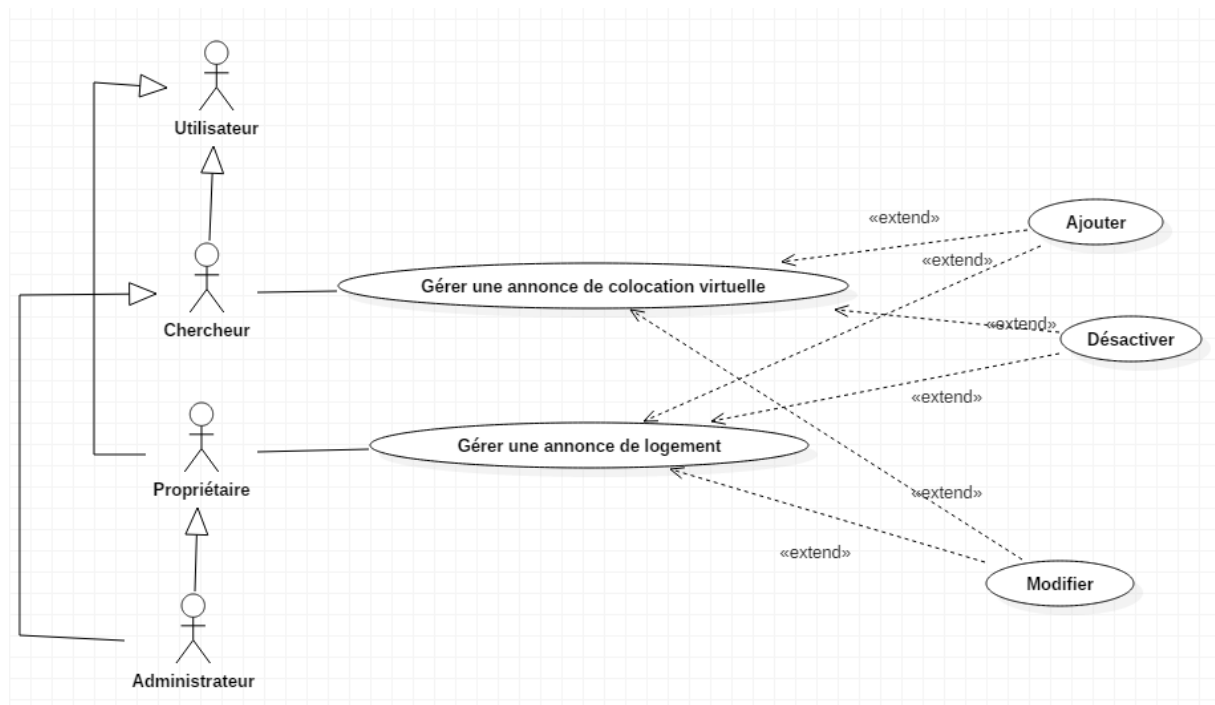


Figure 18 Diagramme de cas d'utilisation : gestion d'annonces

c. Module de messagerie :

Ce module permet aux propriétaires des annonces et les candidats de discuter en temps réel via une interface graphique. La dépendance avec le module du paiement est dû au fait que deux personnes ne peuvent pas discuter sauf si l'un d'eux est un membre premium.

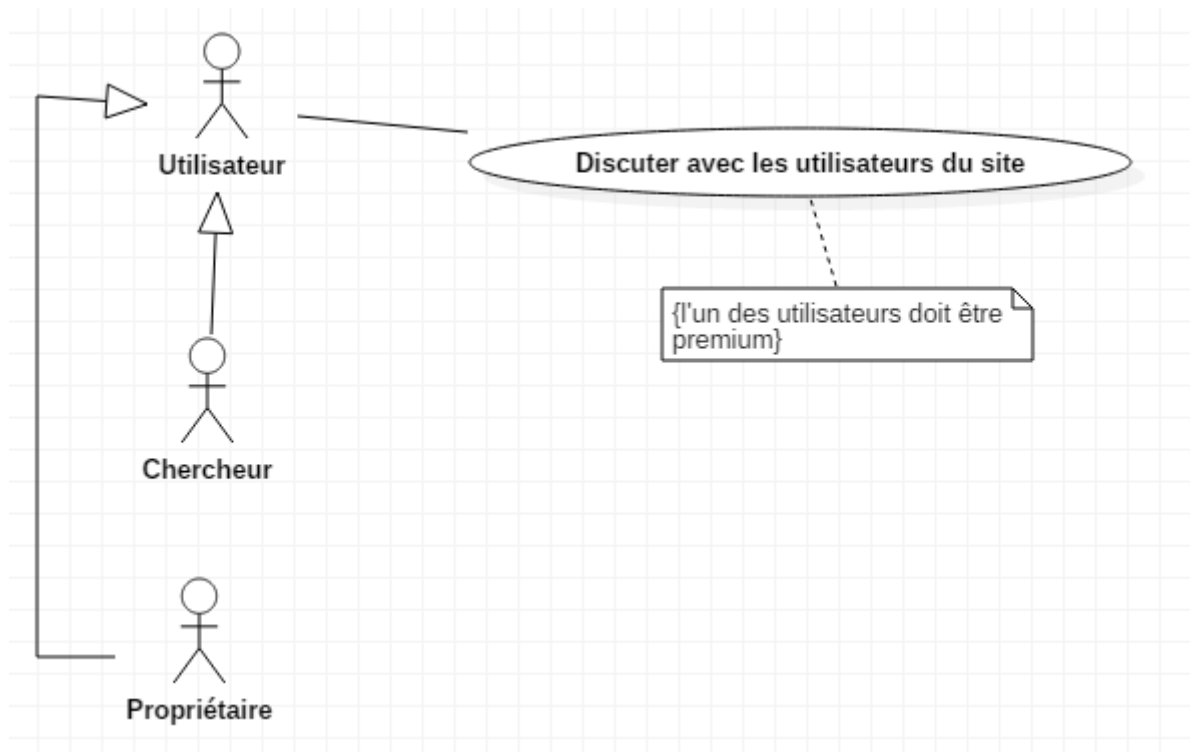


Figure 19 Diagramme de cas d'utilisation : Messagerie

Cas d'utilisation	Discuter avec un utilisateur
Acteurs	Utilisateur
Description	Ce cas d'utilisation décrit les discussions entre deux utilisateurs du site
Précondition	Etre premium
Scénario nominal	1- L'utilisateur1 accède à la page du profil de l'utilisateur 2 2- L'utilisateur1 clique sur le bouton envoyer message 3- L'utilisateur1 rédige son message et l'envoie 4- Le système insère le message dans la base de données et l'affiche à l'utilisateur 2 5- L'utilisateur 2 accède à la page de messagerie 6- L'utilisateur2 consulte le message 7- L'utilisateur2 répond au message

Scenario alternatif	L'utilisateur1 essaie d'envoyer un message vide Le système renvoie l'utilisateur vers l'interface de messagerie pour réécrire son message
Post-condition	Les deux utilisateurs continuent de discuter

Tableau 8 Descriptif du cas : Discuter avec un utilisateur

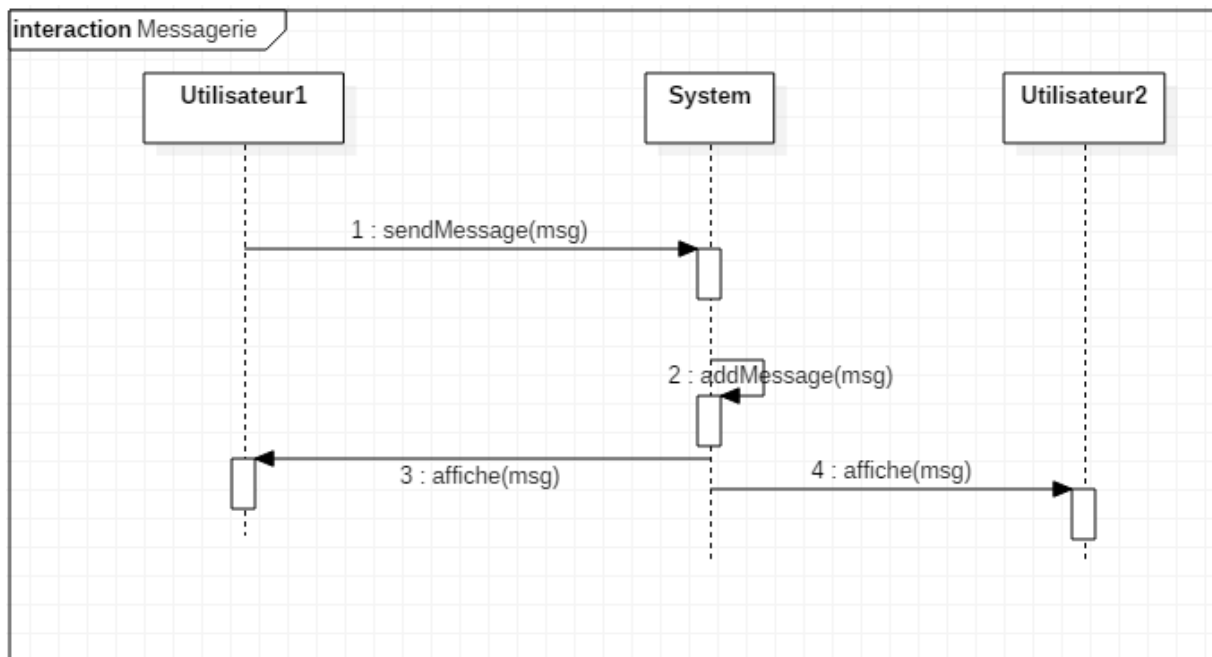


Figure 20 Diagramme de séquence : Messagerie

d. Module de gestion des candidats :

Ce module permet aux chercheurs de postuler pour des annonces d'un côté, et d'un autre de permettre aux propriétaires d'annonces d'exécuter des actions qui concerne les candidats qui postule pour leurs annonces, ces actions peuvent être de les accepter et les faire rejoindre en tant que membre, ou bien de les refuser.

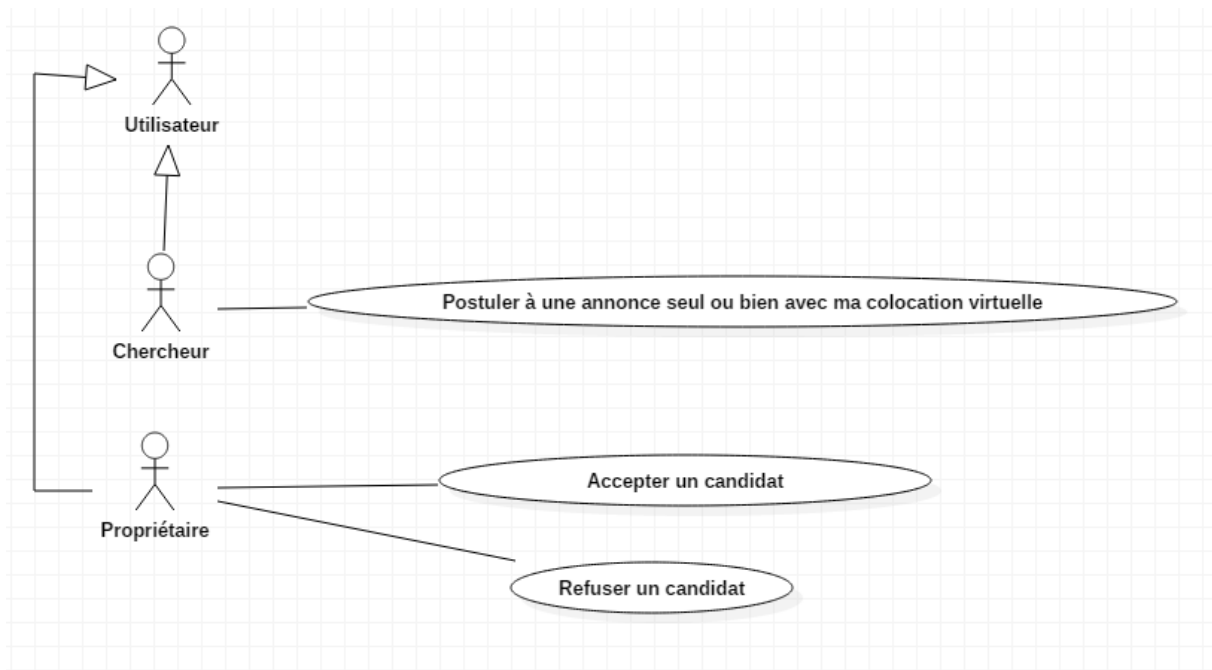


Figure 21 Diagramme de cas d'utilisation : Gestion candidats

cas d'utilisation	Postuler à une annonce
Acteurs	Utilisateur
Description	Ce cas d'utilisation décrit le processus de postulation d'un chercheur à un propriétaire
Précondition	Etre connecté
Scénario nominal	1- L'utilisateur1 accède à la page d'une annonce 2- L'utilisateur1 clique sur le bouton postuler 3- L'utilisateur1 rédige un message pour se décrire et l'envoie 4- Le système insère le message dans la base de données et l'affiche à l'utilisateur 2 5- L'utilisateur 2 accède à la page de messagerie 6- L'utilisateur2 consulte le message 7- L'utilisateur2 répond au message
Scénario alternatif	L'utilisateur1 essaie d'envoyer un message vide Le système renvoie l'utilisateur vers l'interface de messagerie pour réécrire son message
Post-condition	Les deux utilisateurs continuent de discuter

Tableau 9 Descriptif du cas : postuler à une annonce

cas d'utilisation	Accepter Candidat
Acteurs	Propriétaire
Description	Ce cas d'utilisation décrit le processus auquel un propriétaire accepte un candidat pour rejoindre son logement
Précondition	Etre connecté, être le propriétaire de l'annonce
Scénario nominal	Après avoir discuté avec le candidat dans l'interface de messagerie et être intéressé par son profil, l'interface propose un bouton accepter candidat, le propriétaire clique sur ce bouton. Le système cache le bouton accepter, ajoute le candidat à la liste des membres du logement dans la base de donnée, et l'ajoute directement dans la conversation privé de l'annonce et affiche un message de succès
Scénario alternatif	L'utilisateur n'est pas connecté, le système affiche l'interface de connexion
Post-condition	Le candidat est membre de l'annonce

Tableau 10 Descriptif du cas : Accepter Candidat

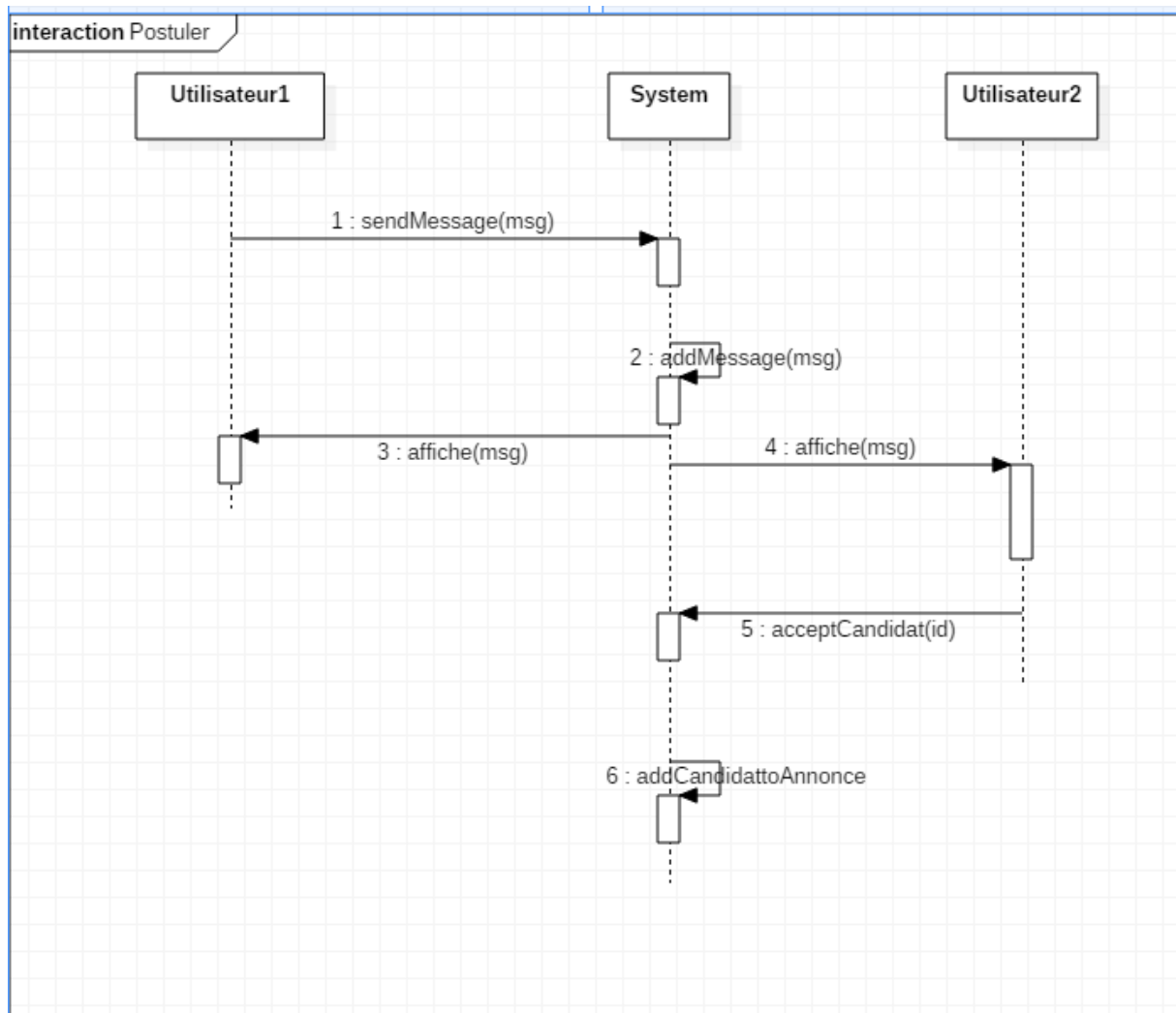


Figure 22 Diagramme de séquence : Postuler

e. Module de notification :

Ce module permet au système d'envoyer des notifications en temps réel aux utilisateurs dès le déclenchement d'un évènement par rapport à eux :

- Annonce ou profil qui correspond à leurs critères
- Nouveau Message
- Candidature acceptée / refusée
- Nouveau candidat pour leur annonce

f. Module de Mailing

Ce module permet au système d'envoyer des mails aux utilisateurs dès le déclenchement d'un évènement par rapport à eux, afin de garder l'utilisateur au courant même s'il n'est pas sur le site : les événements peuvent être :

- Annonce ou profil qui correspond à leurs critères
- Demande de validation de mail
- Nouveau Message
- Candidature acceptée / refusée

- Nouveau candidat pour leur annonce
- Utilisateur a été banni
- Mail de bienvenue sur le site

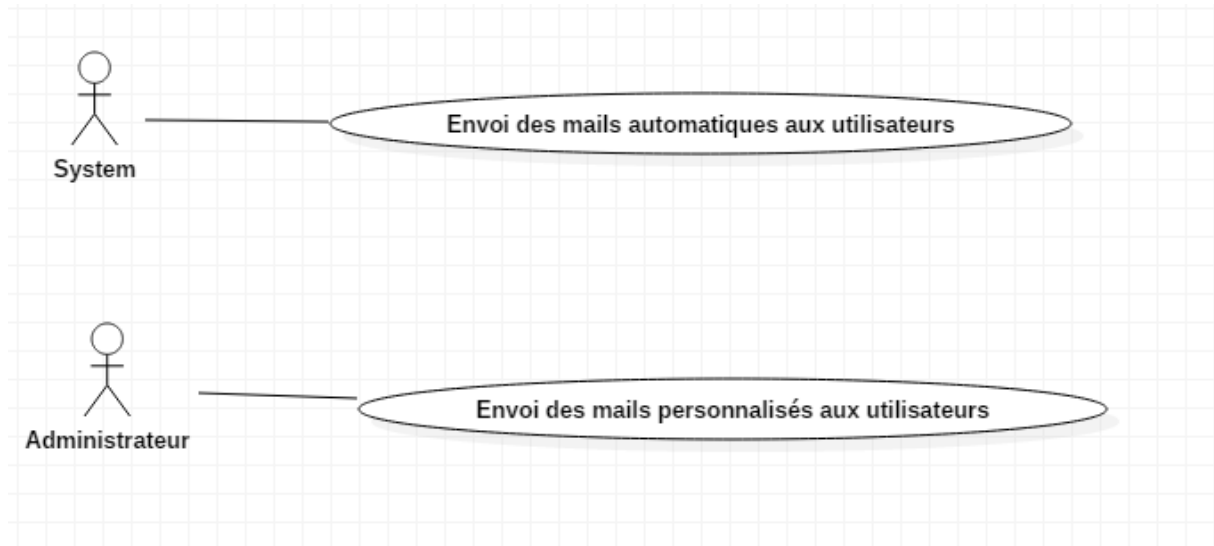


Figure 23 Diagramme de cas d'utilisation : Mailing

cas d'utilisation	Envoi de mail
Acteurs	System
Description	Ce cas d'utilisation permet au système d'envoyer un email à un utilisateur spécifique lors d'un événement qui le concerne
Précondition	Avoir une adresse mail valide
Scénario nominal	Un événement sur le site arrive lorsque l'utilisateur n'est pas connecté Système observe cet événement et déclenche une fonction La fonction envoie un mail qui correspond à l'événement déclenché
Scénario alternatif	Le mail entré par l'utilisateur est invalide, système lui envoie un message sur le site
Post-condition	L'utilisateur reçoit un mail qui contient un lien de redirection vers le site

Tableau 11 Descriptif du cas : Envoi de mail

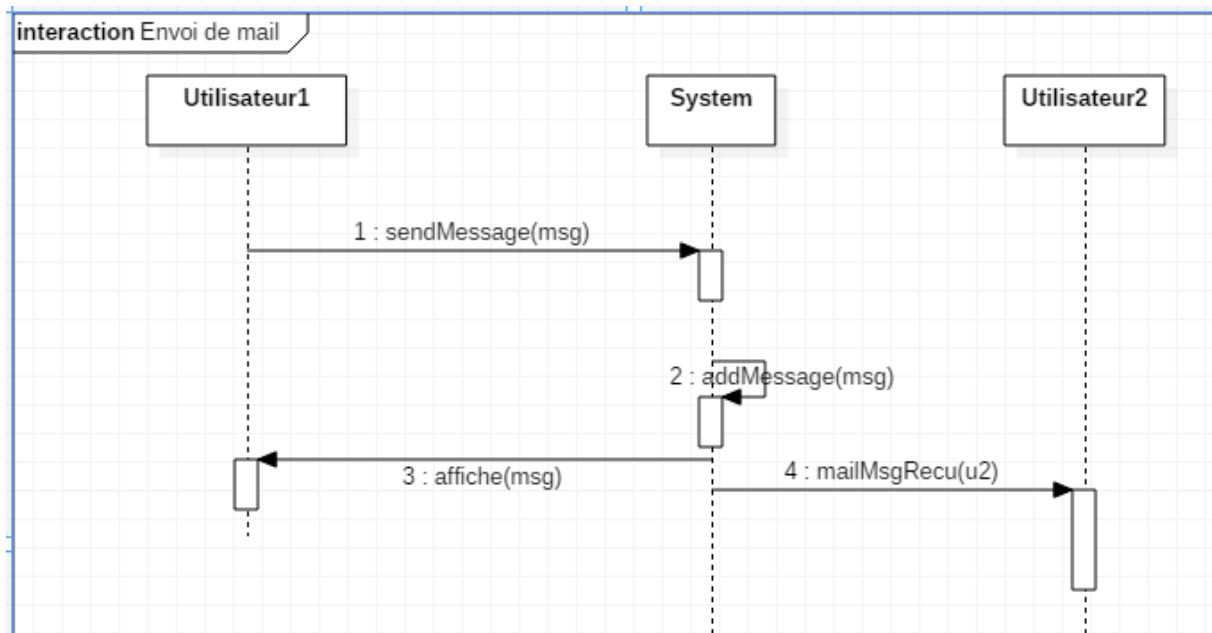


Figure 24 Diagramme de séquence : Mailing

g. Module de paiement

Ce module permet de promouvoir un compte basique vers un compte premium, ainsi que gérer les délais d'abonnements et les forfaits.

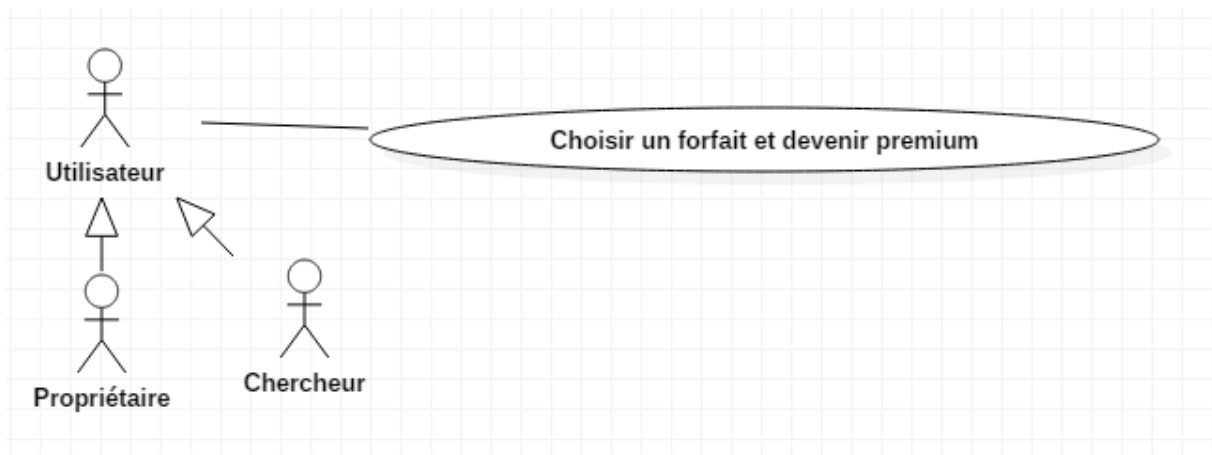


Figure 25 Diagramme de cas d'utilisation : Paiement

Cas d'utilisation	Paielement
Acteurs	Utilisateur
Description	Ce cas d'utilisation permet à l'utilisateur de passer d'un compte basique à un compte premium
Précondition	Etre connecté
Scenario nominal	<ol style="list-style-type: none"> 1. L'utilisateur accède à la page des forfaits 2. Le système affiche les tarifs depuis la base de données 3. L'utilisateur choisit le forfait qui lui convient 4. Le système le redirige vers la page de paiement 5. L'utilisateur insère ses coordonnées bancaires 6. L'utilisateur valide ses coordonnées 7. Le système vérifie la transaction 8. Le système change le type du compte de l'utilisateur 9. Le système envoie un mail qui contient la facture
Scenario alternatif	Coordonnées bancaires erronées Système redirige l'utilisateur vers la page de paiement Et redemande l'insertion des informations
Post-condition	L'utilisateur reçoit un mail qui contient la facture de son paiement

Tableau 12 Descriptif du cas : paiement

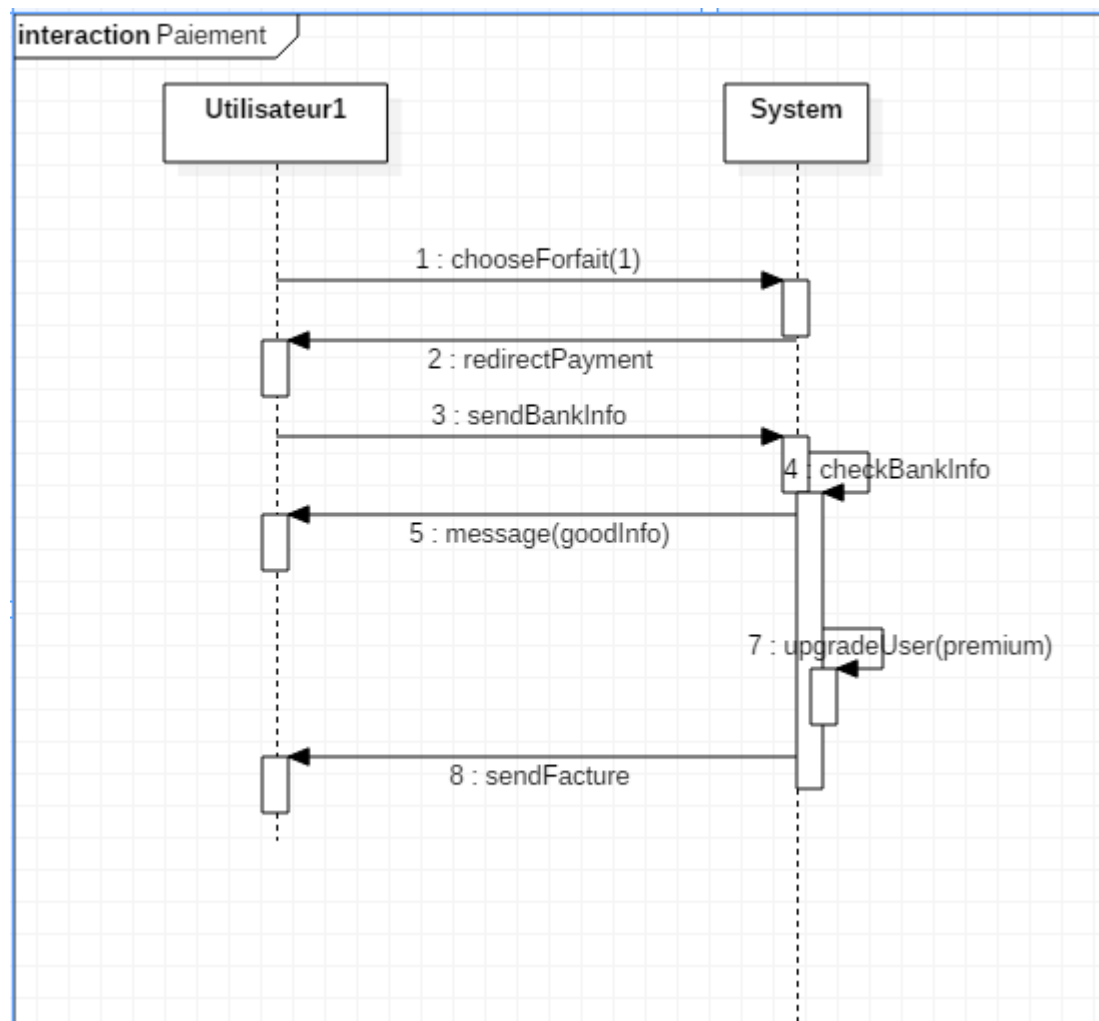


Figure 26 Diagramme de séquence : "Paiement"

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.

3.3 Diagrammes d'activités

Illustrations des scénarios possibles d'utilisation du site :

Scénario d'une personne qui cherche où habiter :

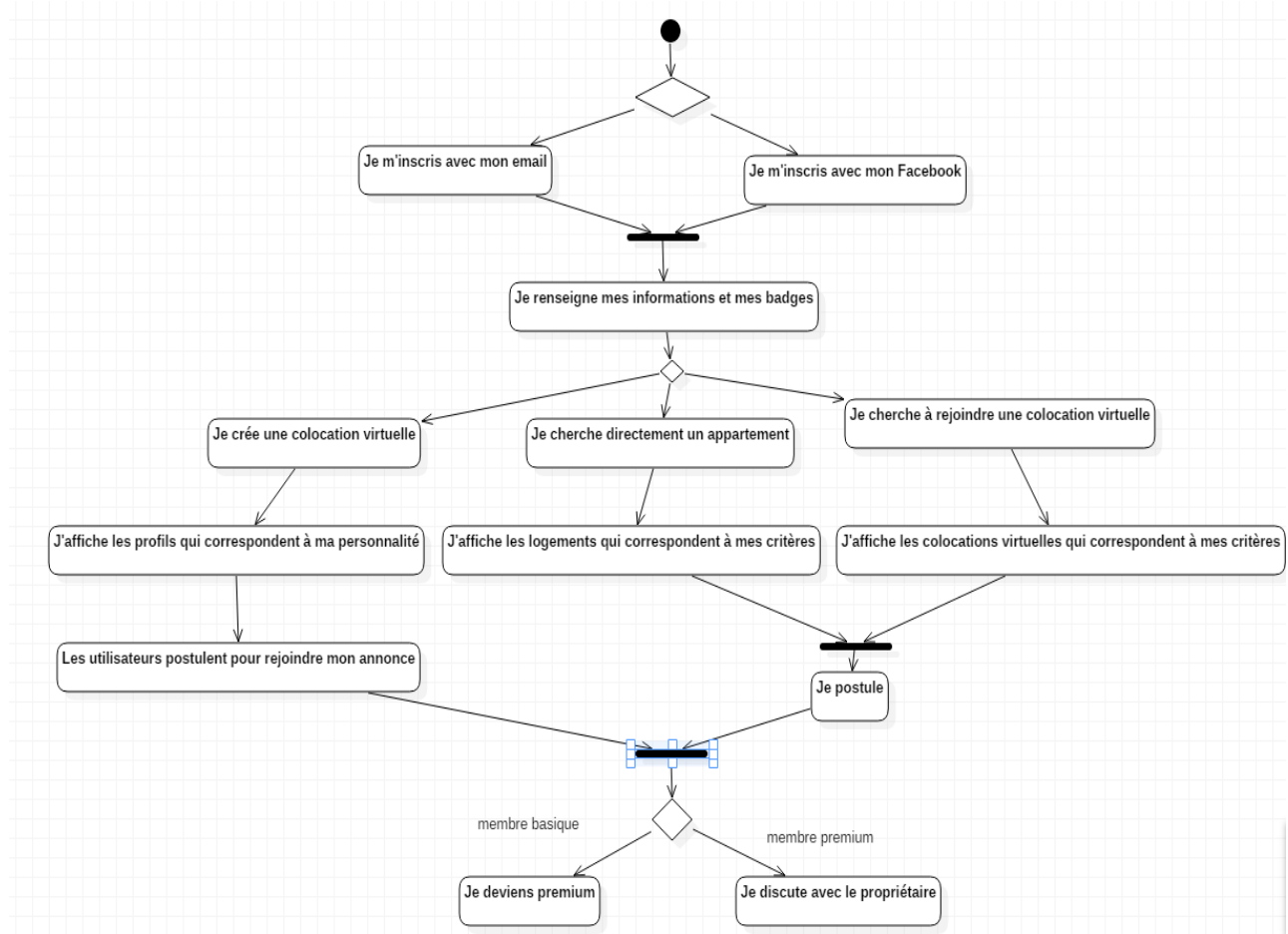
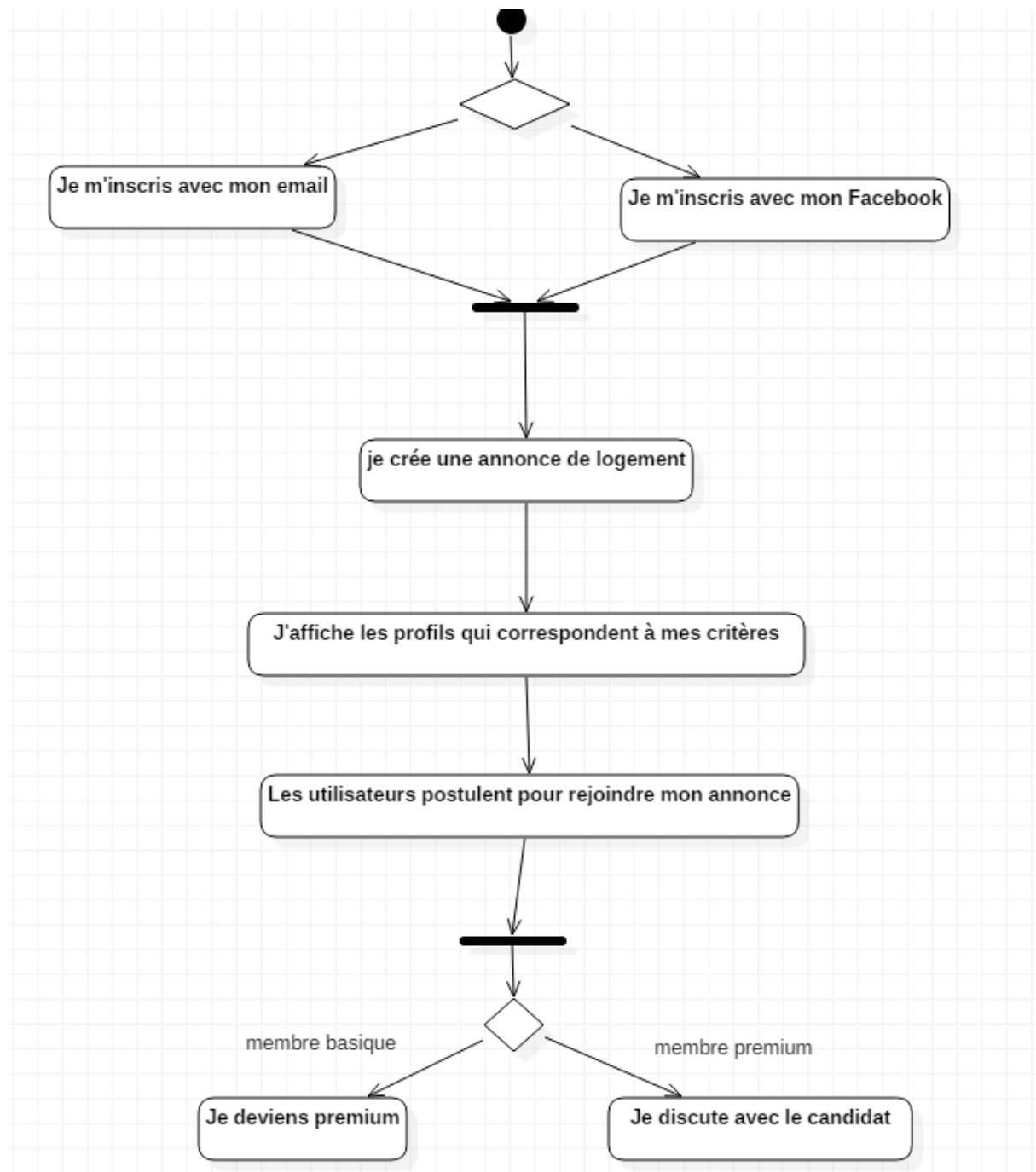
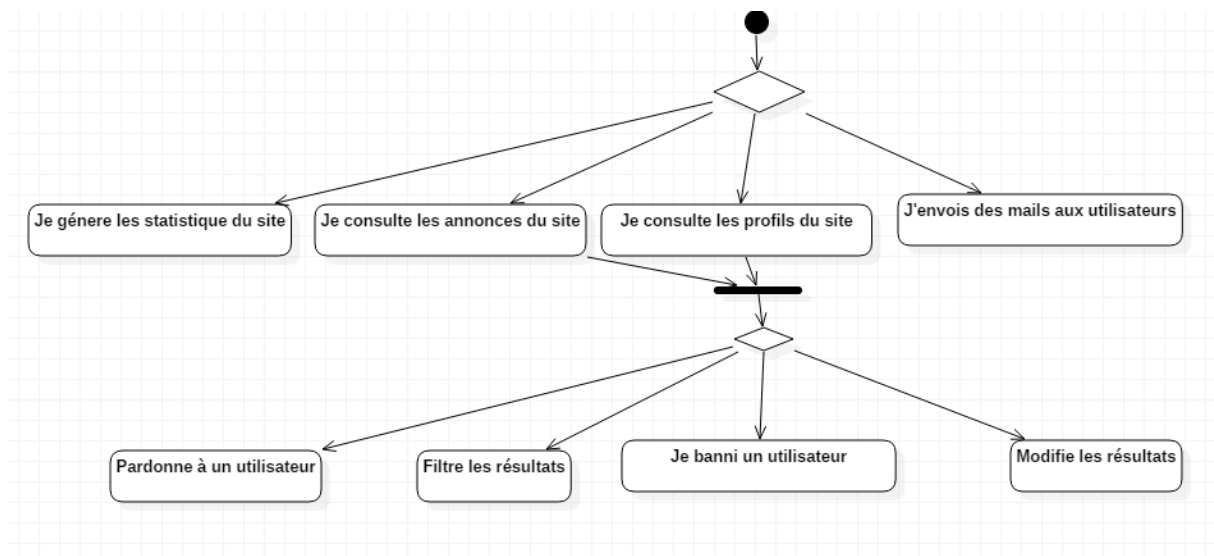


Figure 27 Diagramme d'activités : chercheur

Scénario d'un utilisateur qui propose un appartement*Figure 28 Diagramme d'activités : proposeur*

Administrateur*Figure 29 Diagramme d'activités : Administrateur*

3.4 Diagramme de classe métier

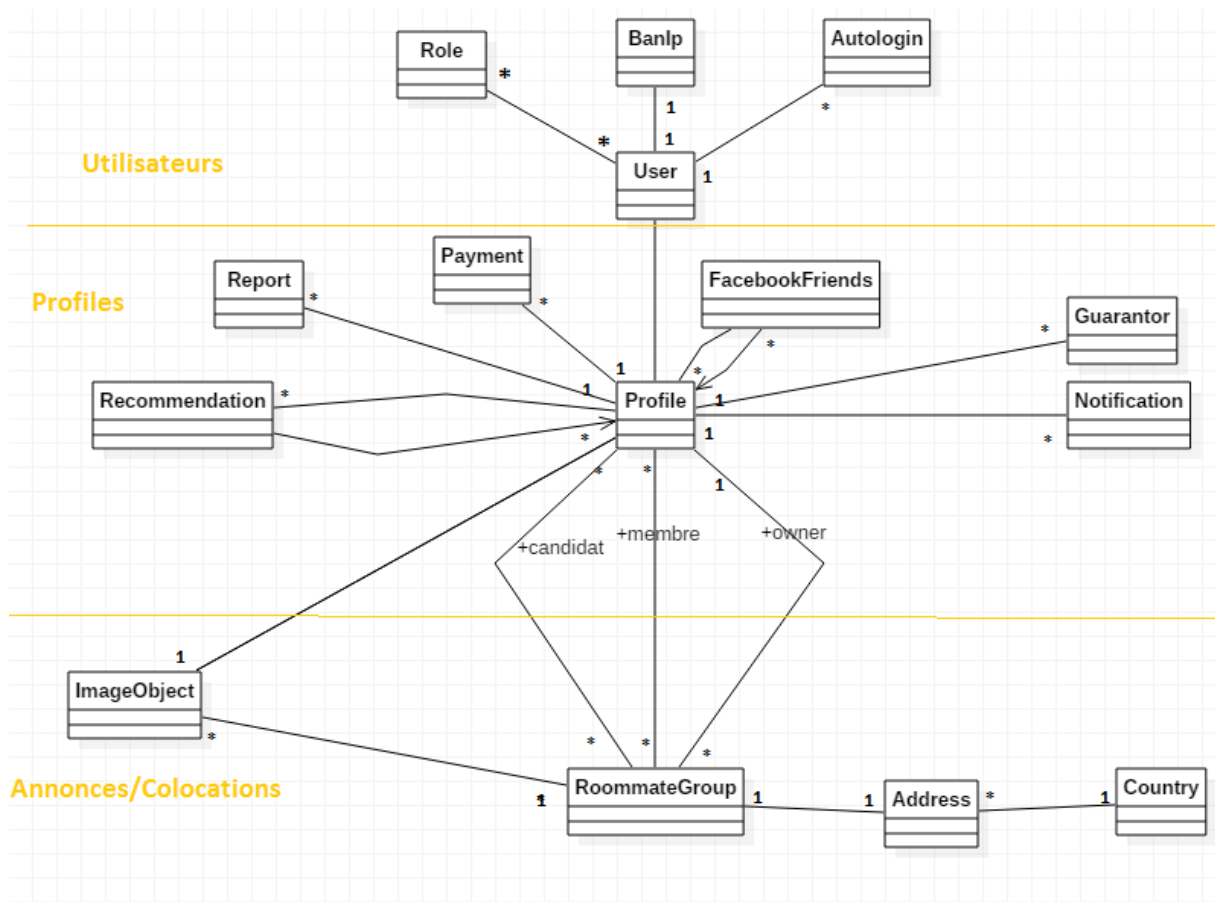


Figure 30 : Diagramme de Classe

RoommateGroup	<p>Cette classe représente les deux cas d'annonces : les colocations virtuelles ainsi que les appartements, vu que les deux ont les mêmes attributs.</p> <p>Une annonce doit absolument avoir un propriétaire de type profil.</p> <p>Elle peut contenir des candidats et des membres de ce même type</p>
Address	Cette classe représente les informations sur l'adresse d'une annonce, tel que la latitude, longitude , ville, pays...
Country	Contient le nom et le code d'un pays
Profile	Cette classe représente un profil sur le site, un profil est une entité qui contient toute les informations d'un utilisateur ainsi que ses badges
ImageObject	Représente une image, soit d'un profil ou bien d'une annonce

Recommendation	Représente une recommandation entre deux profils, celui qui recommande et un autre qui reçoit la recommandation
Report	Représente le report d'un utilisateur suite à un comportement frauduleux
Payment	Cette classe contient toutes les informations des Payments d'un utilisateur, les forfaits qui l'a acheté et leurs date de fins, ainsi que son statut (premium ou basique)
FacebookFriends	Cette classe représente le lien d'amitié Facebook entre deux profils
Guarantor	Cette classe contient les informations sur le garant d'un profil afin de garantir son éligibilité
Notification	Cette classe contient les notifications d'un profil, leur type et leur date
User	La classe User contient les informations d'un compte utilisateur, son nom de compte et son mot de passe, son ID Facebook et la date de la dernière connexion
AutoLogin	La classe auto login contient les jetons qui permettent à l'utilisateur de s'authentifier automatiquement lorsqu'il clique sur un lien depuis son mail
BanIP	Contient l'adresse IP banni d'un utilisateur
Role	Représente le rôle d'un utilisateur, qui peut être administrateur ou bien normal

Tableau 13 Description des classes

Chapitre 3

Etude technique

L'étude technique a pour objectif de définir l'architecture physique et celle logique de la solution. Dans ce sens, ce chapitre décrit la structure de notre projet

I- 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)

1- 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, par conséquent, il va suivre une architecture MVC

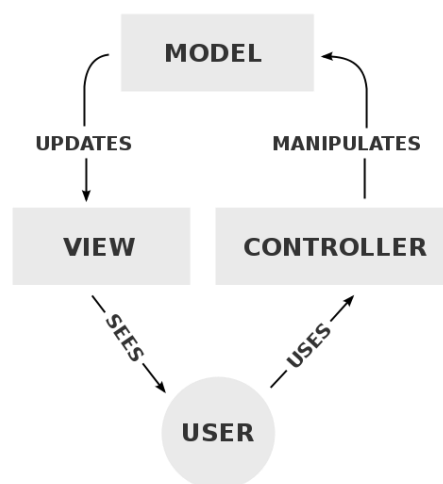


Figure 31 Architecture MVC

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.

2- 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.

3- 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.

4- 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.

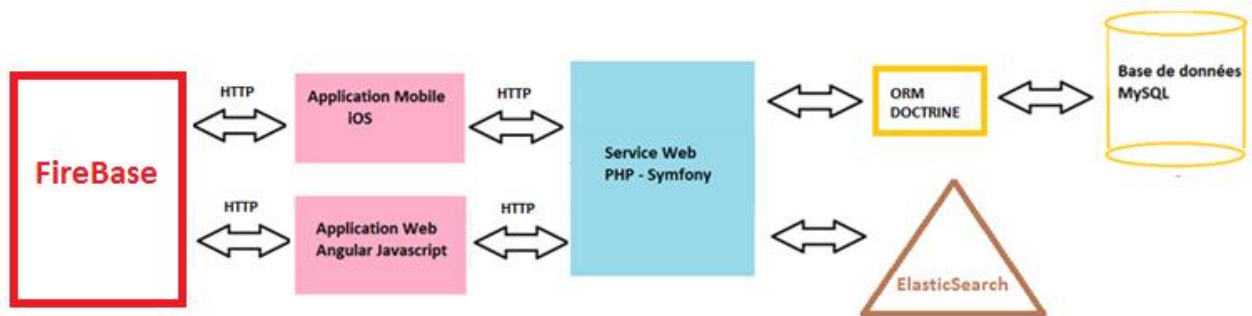


Figure 32: Architecture du projet

Chapitre 4

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

1- 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 personnalisées.

Justification du choix : tester le service web avant d'avoir développé le front end.

2- 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.

Justification du choix : Augmenter les performances du serveur, rediriger l'url vers les bons ports car notre service web et notre site fonctionnent 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.

3- 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.

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/plugins tel que Sonata
- Forte capacité d'évolution
- Communauté toujours plus grande (plus de 300 000 développeurs dans 120 pays)

Justification du choix :

Symfony fournit plusieurs bundles simple et optimal qui vont nous aider à développer facilement plusieurs fonctionnalités attendu de notre service web :

- FOSRESTBUNDLE : Fournit plusieurs outils pour développer rapidement des API REST
- JWTAuthenticationBundle : Permet l'authentification JWT (Json Web Token) pour les applications Symfony
- 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'Envolv, 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 6455 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.

4- 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 apis 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- Répartition des tâches

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

Vincent Péliissier : 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.

III- 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) .

1- 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.

2- 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 33 : Routes du service Web

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

a. Etape 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.

b. Etape 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.

c. 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.

d. Etape du Mailing

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

e. Etape Paiement

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

f. 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.

g. 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.

3- 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.

a. 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.

b. 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.

c. 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.

d. 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

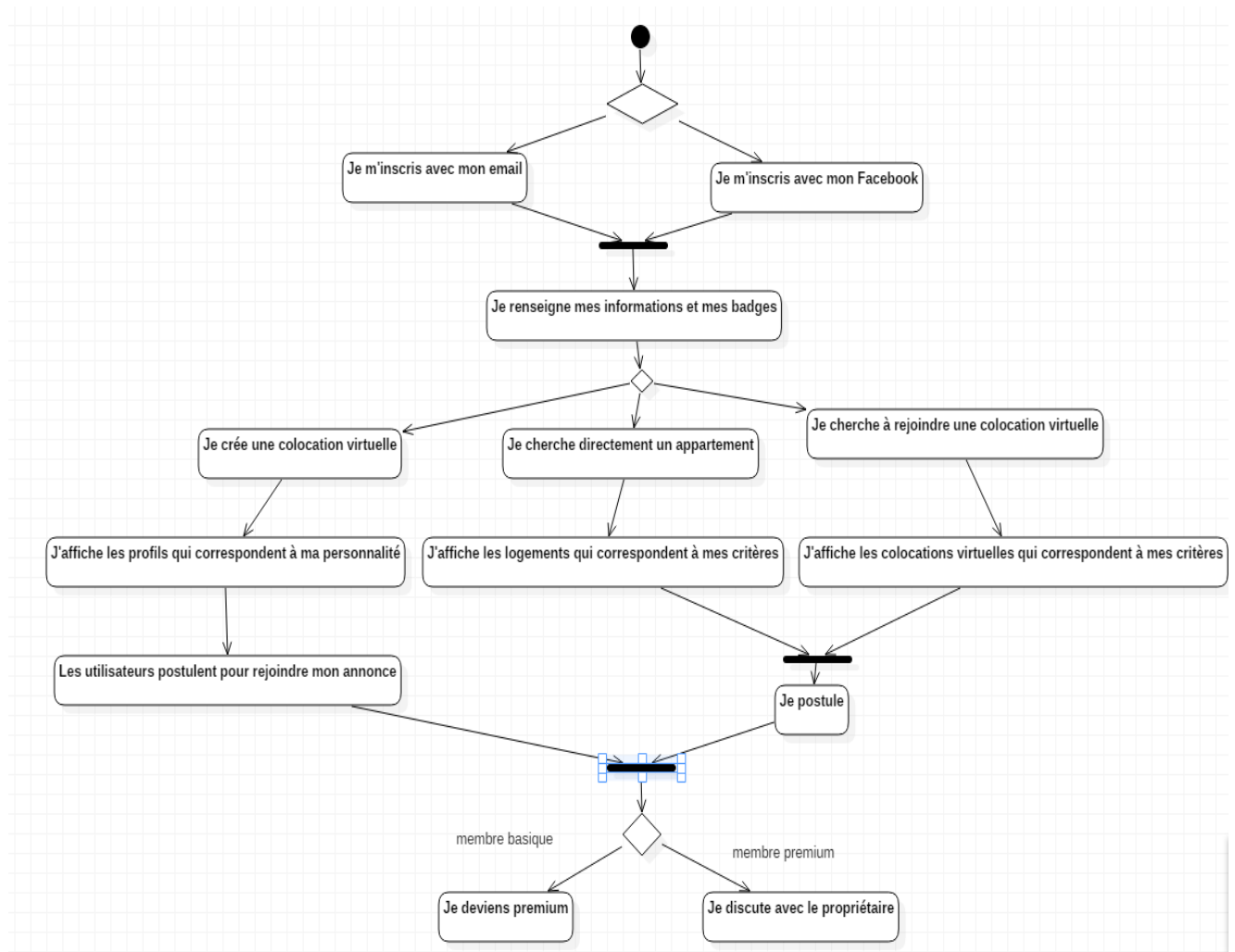
Envoyer des invitations aux amis pour les inciter à rejoindre le site

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.

IV- 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 34 : Page d'accueil

Inscription

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

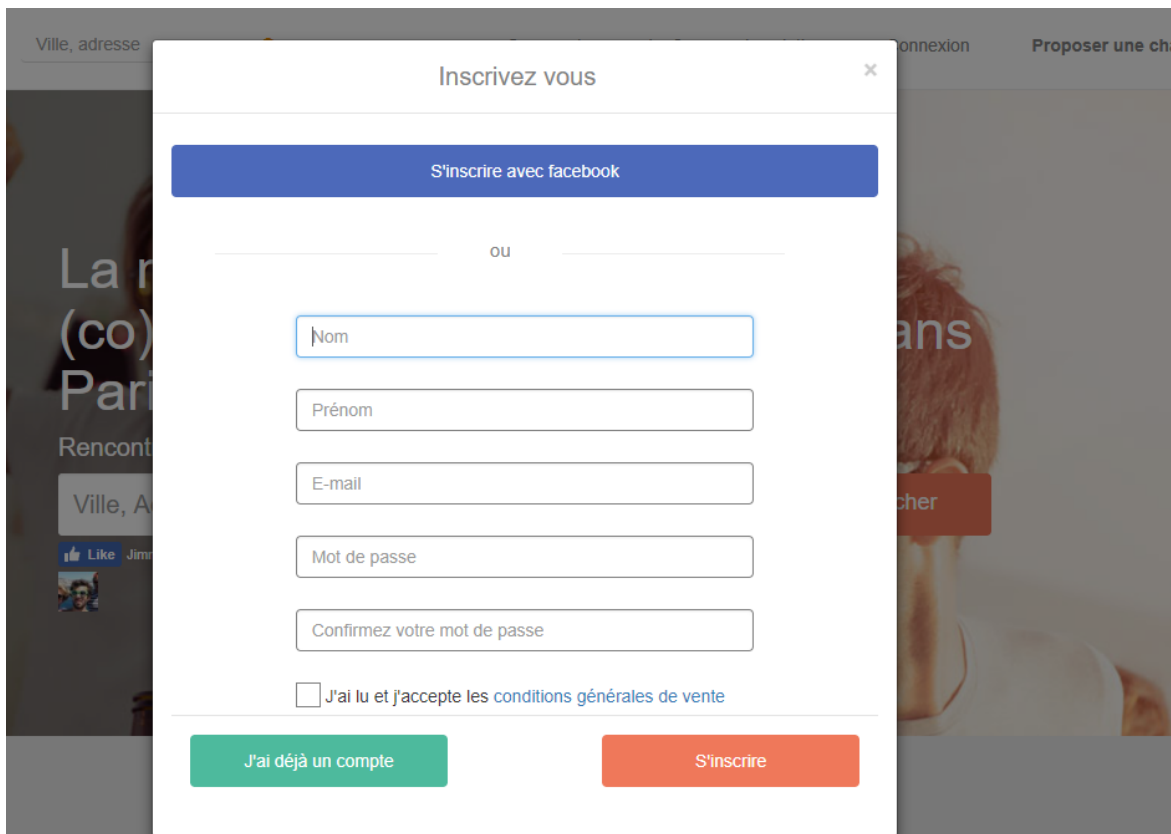


Figure 35 Page d'inscription

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 checked options are 'Non fumeur' and 'Fumeur'.

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 36 : 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 37 Page de création de colocation virtuelle

Chercher des profils qui correspondent à ma personnalité

Appart Ville, adresse

Critères pour les profils

Genre

Fumeur

Célibataire

Statut

Moins de filtres Je souhaite être alerté Rechercher

Béra Ariane Angélique

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

Figure 38 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'logis profile page for a user named Angélique. The header includes the Appart'logis logo, a search bar with the placeholder 'Ville, adresse', and navigation links: 'Comment ça marche?', a mail icon, a profile icon with a dropdown arrow, and 'Proposer une chambre'. The profile section on the left features a large photo of Angélique, a green 'Inviter à me rejoindre' button, a yellow box stating 'Actif il y a 2 heure(s)', a green box showing '100 % de réponses', and a red box indicating 'Membre Basique'. Below these are three verification status boxes: 'Non vérifié' with a Facebook icon, 'Non vérifié' with an email icon, and 'Non vérifié' with a phone icon. The main content area on the right has 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'. At the bottom right of this section is a green button 'Recommander : 0'.

Figure 39 Page du profil

Inviter à rejoindre ma colocation

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

This screenshot shows the same Appart'logis profile page as Figure 39, but with a modal window open for inviting the user to a specific listing. The modal has a title bar 'Pour quelle annonce voulez-vous inviter ce profil à postuler ?' and a close button. Inside the modal is a small image of a room and the text 'Coloc Gare du Nord'. The background profile page is dimmed, showing the same bio and buttons as before.

Figure 40 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.

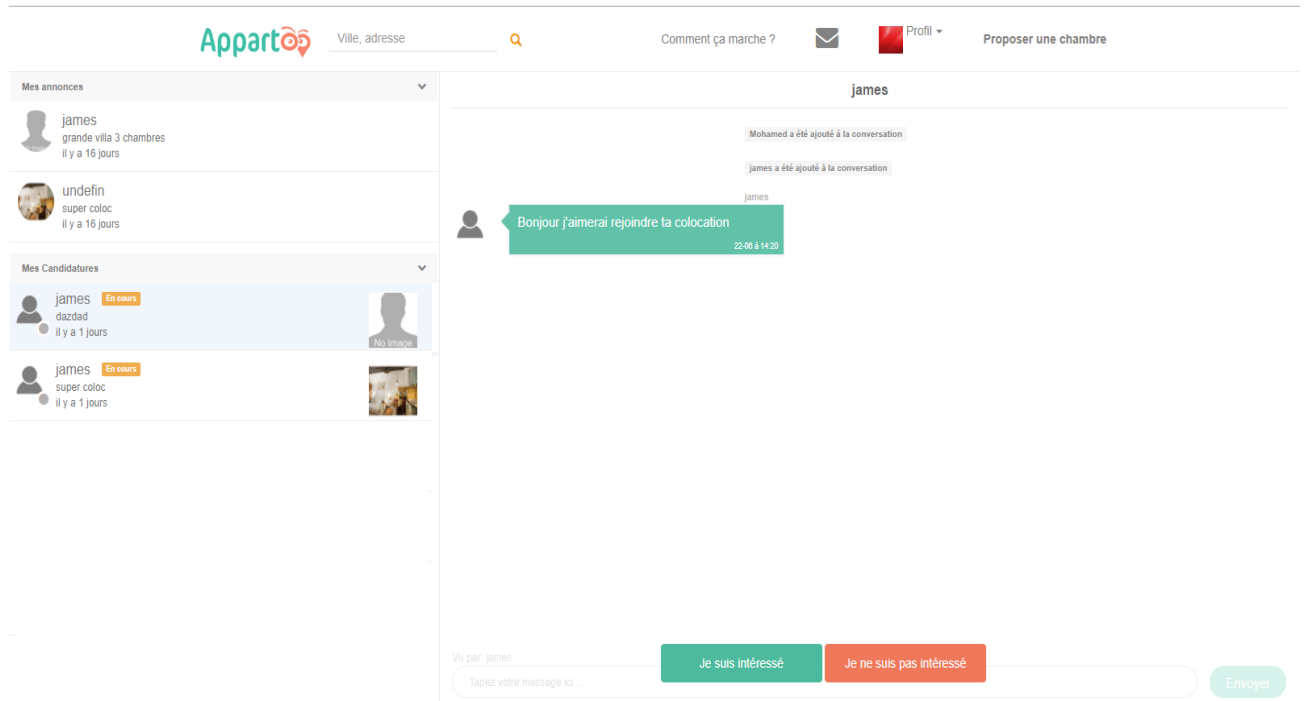





Figure 41 Interface de messagerie


Page Premium

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




[Comment ça marche ?](#)

 Profil ▾
 [Proposer une chambre](#)

Choisissez la formule qui vous correspond :




7 jours

€ par jour
Soit €



30 jours


€ par jour
Soit €





3 mois

€ par jour
Soit €

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 42 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

Filtres

Ajouter









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

Figure 43 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	  		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	  		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	  		non
<input type="checkbox"/>		Colocation 2pers. Denfert	Colocation	brasse marie 0681722194	Paris 75014 France	24/06/17 08:54	600	19/06/2017 Indéterminée	0 / 0	0	  		non

Figure 44 Interface admin annonce

Ou bien visualiser les statistiques d'utilisations du site

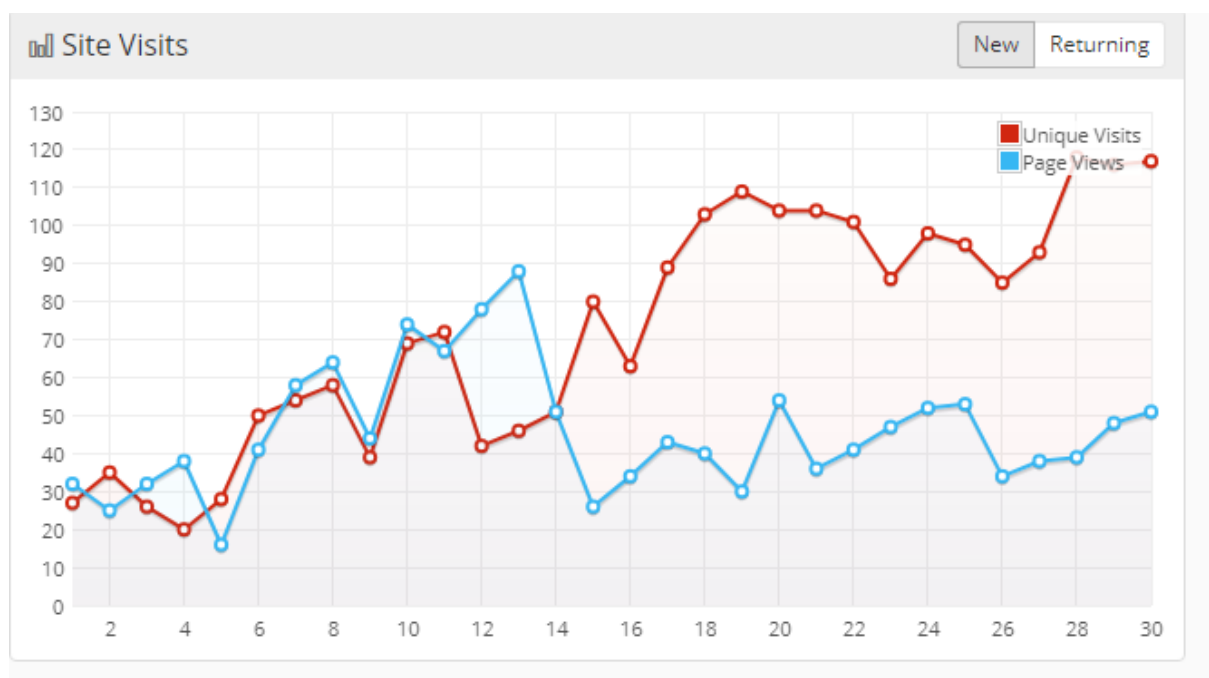


Figure 45 Admin Statistique

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.

Références

Introduction aux méthodes agiles et Scrum. (s.d.). Récupéré sur [http://www.agiliste.fr:](http://www.agiliste.fr:80/http://www.agiliste.fr/introduction-methodes-agiles/)
<http://www.agiliste.fr/introduction-methodes-agiles/>

M Kenzi Adil - la technologie des services web. (s.d.).

Wikipédia - Slack. (s.d.). Récupéré sur Wikipedia:
[https://fr.wikipedia.org/wiki/Slack_\(plateforme\)](https://fr.wikipedia.org/wiki/Slack_(plateforme))

Wikipédia - Trello. (s.d.). Récupéré sur Wikipédia: <https://fr.wikipedia.org/wiki/Trello>