

RAPPORT DE STAGE DE FIN D'ETUDES

Pour l'obtention de la
«Licence Appliquée en Sciences et Technologies de l'Information et de
Communication (LASTIC)»

Présenté par :
Amel BACCOURI

Titre

Conception et développement d'une application mobile qui propose un
réseau social de partage d'expérience et de rencontre

Devant le jury :

Encadreur : Mme. Chiraz HOUAIDIA & Mr. Mohamed RHAMNIA

Année Universitaire : 2018 / 2019

Dédicace

Je dédie ce mémoire à mes parents, à mes frères et sœurs

A mes enseignants tous au long de mon cursus scolaire et universitaire,

Je ferai le nécessaire pour être digne de votre respect,

Toute ma gratitude

Remerciements

Je souhaite rendre hommage à tous ceux qui ont contribué à la réalisation de ce travail.

Je tiens, avant tout, à adresser mes plus vifs remerciements à Madame Chiraz HOUADIA, mon encadrante à l'UVT, pour l'encadrement sérieux dont elle a fait preuve ainsi qu'à nos enseignants à qui je dois ma formation.

Je tiens à exprimer ma gratitude à Monsieur Mohammed RHAMNIA, mon encadrant à la société, pour m'avoir, intégré rapidement au sein de l'entreprise et m'avoir accordé toute sa confiance ; pour le temps qu'il m'a consacré tout au long de cette période.

Enfin, j'adresse mes remerciements aux membres du jury pour m'avoir honoré en acceptant d'évaluer ce travail.

Sans oublier tous ceux qui ont participé de près ou de loin à l'accomplissement de ce projet.

1 Table des matières

| | |
|--|----|
| Table des Figures..... | 6 |
| Liste des Tableaux..... | 7 |
| Liste des Acronymes | 8 |
| Introduction générale..... | 9 |
| Chapitre I : Contexte du projet et Etude de l'existant | 11 |
| Introduction | 11 |
| 1. Présentation de l'organisme d'accueil :..... | 11 |
| 2. Cahier des charges :..... | 12 |
| 2.1 Objectif du projet :..... | 12 |
| 2.2 Les différentes tâches à réaliser:..... | 13 |
| 2.3 Méthodologie à Suivre dans le projet :..... | 13 |
| 3. Etude de l'existant : | 14 |
| 3.1 Etude comparative des réseaux sociaux: | 15 |
| 3.2 Choix de la solution :..... | 20 |
| Conclusion..... | 21 |
| Chapitre II : Spécification des besoins | 21 |
| Introduction | 22 |
| 1. Identification des acteurs..... | 22 |
| 2. Besoins Fonctionnels :..... | 22 |
| 2.1 Inscription :..... | 23 |
| 2.2 Authentification :..... | 23 |
| 2.3 Messagerie et discussion instantanée : | 23 |
| 2.4 Notification :..... | 23 |
| 2.5 Participation à une proposition de rencontre :..... | 24 |
| 2.6 Ajouter une proposition de rencontre :..... | 24 |
| 2.7 Gérer le Profil Utilisateur :..... | 24 |
| 2.8 Gestion des Contacts/ Amis: | 25 |
| 3. Diagrammes de cas d'utilisation : | 25 |
| 3.1 Diagramme de cas d'utilisation Général : | 25 |
| 3.2 Diagramme de cas d'utilisation d'Inscription par authentification : | 27 |
| 3.3 Diagramme de cas d'utilisation d'Authentification : | 28 |
| 3.4 Diagramme de cas d'utilisation de Messagerie instantanée :..... | 29 |
| 3.5 Diagramme de cas d'utilisation de gestion de Notification :..... | 30 |
| 3.6 Diagramme de cas d'utilisation de Gestion de propositions de rencontres: | 31 |
| 3.7 Diagramme de cas d'utilisation de Gestion de profil utilisateur : | 34 |

| | | |
|---------------------------------|--|----|
| 3.8 | Diagramme de cas d'utilisation de gestion de contacts/Amis : | 35 |
| 4. | Besoins Non Fonctionnels | 37 |
| 4.1 | La sécurité : | 37 |
| 4.2 | L'ergonomie : | 37 |
| 4.3 | La performance : | 37 |
| 4.4 | L'intégrité : | 37 |
| 4.5 | La maintenabilité: | 38 |
| 4.6 | La disponibilité : | 38 |
| | Conclusion : | 38 |
| Chapitre III : Conception | | 39 |
| | Introduction | 39 |
| 1. | Diagrammes de séquence : | 39 |
| 1.1. | Diagramme de séquence « Inscription par authentification » : | 40 |
| 1.2. | Diagramme de séquence « Authentification » : | 41 |
| 1.3. | Diagramme de séquence « Messagerie instantanée » : | 42 |
| 1.4. | Diagramme de séquence « Gestion de proposition » : | 43 |
| 1.5. | Diagramme de séquence « Gestion de réseau d'amis » : | 44 |
| 2. | Diagrammes de Classe | 45 |
| 3. | Diagrammes de Base de données : | 45 |
| | Conclusion | 46 |
| Chapitre IV : Réalisation | | 47 |
| | Introduction | 47 |
| 1. | Environnement de travail | 47 |
| 1.1 | Architecture Informatique sans serveur ou « Serverless » | 47 |
| 1.2 | Langages et outils utilisés : | 53 |
| 2. | Travail réalisé | 55 |
| 2.1 | Backlog produit et méthodologie de travail : | 55 |
| 2.2 | Développement WebService | 57 |
| 2.3 | Test et validation | 62 |
| | Conclusion | 64 |
| Conclusion générale | | 65 |
| Bibliographie | | 66 |

Table des Figures

| | |
|---|----|
| Figure 1. Organigramme NM-Developpement..... | 12 |
| Figure 2. Cycle de vie d'un projet Scrum | 14 |
| Figure 3. Logo Facebook & Twitter..... | 15 |
| Figure 4. Logo Viadeo & LinkedIn | 16 |
| Figure 5. Logo Copainsd'avant & Trombi | 17 |
| Figure 6. Logo Snapchat & Instagram | 17 |
| Figure 7. Diagramme de Cas d'utilisation Général..... | 26 |
| Figure 8. Diagramme de cas d'utilisation Inscription par authentification | 27 |
| Figure 9. Diagramme de cas d'utilisation Authentification | 28 |
| Figure 10. Diagramme de cas d'utilisation de Messagerie instantanée. | 29 |
| Figure 11. Cas d'utilisation de Gestion de Notification | 30 |
| Figure 12. Diagramme de cas d'utilisation Gestion de proposition de rencontre | 32 |
| Figure 13. Diagramme de cas d'utilisation Gestion de profil utilisateur | 34 |
| Figure 14. Diagramme de cas d'utilisation de gestion de contacts/Amis | 36 |
| Figure 15. Diagramme de séquence « Inscription par authentification» | 40 |
| Figure 16. Diagramme de séquence Authentification | 41 |
| Figure 17. Diagramme de séquence Messagerie instantanée | 42 |
| Figure 18. Diagramme de séquence Gestion de propositions de rencontres | 43 |
| Figure 19. Diagramme de séquence Gestion de réseau d'amis..... | 44 |
| Figure 20. Diagramme de classe général | 45 |
| Figure 21. Diagramme de Base de données | 46 |
| Figure 22. Exemple d'architecture d'une application sans serveur | 48 |
| Figure 23. Amazon S3 | 49 |
| Figure 24. Amazon Cognito | 50 |
| Figure 25. Logo Amazon Cognito..... | 50 |
| Figure 26. Amazon DynamoDB..... | 51 |
| Figure 27. Amazon API Gateway & Lambda..... | 52 |
| Figure 28. Logo NodeJS..... | 54 |
| Figure 29. Logo http/2..... | 54 |
| Figure 30. Logo ReactNative..... | 55 |
| Figure 31. Backlog Produit | 56 |
| Figure 32. Exemple d'US d'ajout d'une proposition de rencontre | 57 |
| Figure 33. WS Proposition de rencontre..... | 58 |
| Figure 34. Schéma WS Proposition de rencontre | 58 |
| Figure 35. Extrait du code JSON du WS proposition de rencontre..... | 59 |
| Figure 36. WS réseau d'amis | 59 |
| Figure 37. Schéma WS Réseau d'amis | 60 |
| Figure 38. Extrait du code JSON du WS réseau d'amis | 60 |
| Figure 39. WS Messagerie instantanée | 61 |
| Figure 40. Schéma WS messagerie instantanée..... | 61 |
| Figure 41. Extrait du code JSON du WS messagerie instantanée | 62 |
| Figure 42. feature_ création de proposition de rencontre privée | 64 |

Liste des Tableaux

| | |
|---|-----------|
| <i>Tableau 1. Tableau comparatif des réseaux sociaux les plus populaires</i> | <i>20</i> |
| <i>Tableau 2. Description Textuelle du Cas d'utilisation Inscription par authentification</i> | <i>28</i> |
| <i>Tableau 3. Description textuelle du cas d'utilisation Authentification</i> | <i>29</i> |
| <i>Tableau 4. Description textuelle du cas d'utilisation Messagerie instantanée</i> | <i>30</i> |
| <i>Tableau 5 . Description textuelle du Cas d'utilisation Notification.....</i> | <i>31</i> |
| <i>Tableau 6. Description textuelle Gestion de Propositions de Rencontres</i> | <i>33</i> |
| <i>Tableau 7. Description textuelle Gestion de profil</i> | <i>35</i> |
| <i>Tableau 8. Description textuelle Gestion d'amis</i> | <i>36</i> |

Liste des Acronymes

R&D: Recherche et développement.

RSPH: Royal Society for Public Health.

IoT: L'Internet des objets (en anglais *Internet of Things*)

AWS: Amazon Web Services

API: Application Programming Interface

REST: Representational State Transfer

HTTP: Hypertext Transfer Protocol

JSON: JavaScript Object Notation – Notation Objet issue de JavaScript.

IOS: iPhone OS

UWP: Universal Windows Platform (Plateforme d'Application Universelle Windows).

US: User Story

Introduction générale

L'être Humain a été conçu pour socialiser quel que soit le média utilisé. [1]

De nos jours, les réseaux sociaux font partie de notre vie quotidienne. Ils nous permettent de communiquer, grâce à de nombreuses sortes de plateformes et applications, et ce dans plusieurs domaines, selon les intérêts de chacun.

Ils sont devenus une véritable addiction chez certaines personnes ; et ils impactent les relations humaines.

Un individu renfermé sur lui-même, derrière son écran, peut avoir 500, 1000 ou même 5000 amis, alors qu'il ne connaît personne dans la vraie vie.

Ce constat se confronte au sens pur de l'amitié qui se construit dans la réalité avec des personnes en qui nous avons confiance et avec qui nous avons eu un contact physique.

D'où vient l'idée de notre projet qui consiste à créer un réseau social ayant pour but de valoriser les contacts physiques et de faire sortir les internautes des relations virtuelles vers les échanges directs dans la vraie vie. Ces internautes pourront partager différents passions et intérêts autour d'un café ou un repas grâce à notre application qui facilite l'organisation de ces rencontres et mémorise tous les évènements. Ce projet effectué pour le compte de la société *NM-DEVELOPPEMENT* vient dans le but de l'obtention du diplôme de Licence Appliquée en Sciences et Techniques de l'Information et de Communications à l'UVT (Université Virtuelle de Tunis).

Ce rapport résume les étapes de réalisation du projet et est réparti sur quatre grands chapitres :

Le premier chapitre présente le contexte du projet ainsi que l'étude de l'existant.

Le deuxième chapitre fait le point sur les spécifications fonctionnelles de l'application, ainsi que les acteurs principaux interagissant avec cette dernière.

Le troisième chapitre, concerne l'étude conceptuelle, il présente les différents diagrammes UML (les diagrammes de séquence, les diagrammes de classe....) ayant pour but de donner

une vue globale sur les scénarios nominaux du système et aidera à bien entamer le chapitre de réalisation.

Le dernier chapitre, contient la partie réalisation de l'application dans laquelle nous allons identifier l'environnement technique utilisé pour le développement de l'application ainsi que les différentes contributions de notre projet.

Chapitre I : Contexte du projet et Etude de l'existant

Introduction

Dans ce chapitre, nous allons commencer par présenter l'organisme d'accueil, puis nous ferons un aperçu sur le cahier de charge.

Ensuite, nous allons présenter l'étude de l'existant et une étude comparative des réseaux sociaux existants sur le marché.

1. Présentation de l'organisme d'accueil :

La Startup NM-Developpement est une Agence Web basée à Tunis spécialisée dans le développement des applications Web et mobiles.

Elle accompagne ses clients nationaux et internationaux dans la réalisation de leurs projets sur multiplateformes. Elle offre aussi des services de mobilité en France pour des clients de renommée internationale.

Les ressources humaines de NM-Developpement sont répartis sur les responsabilités suivantes :

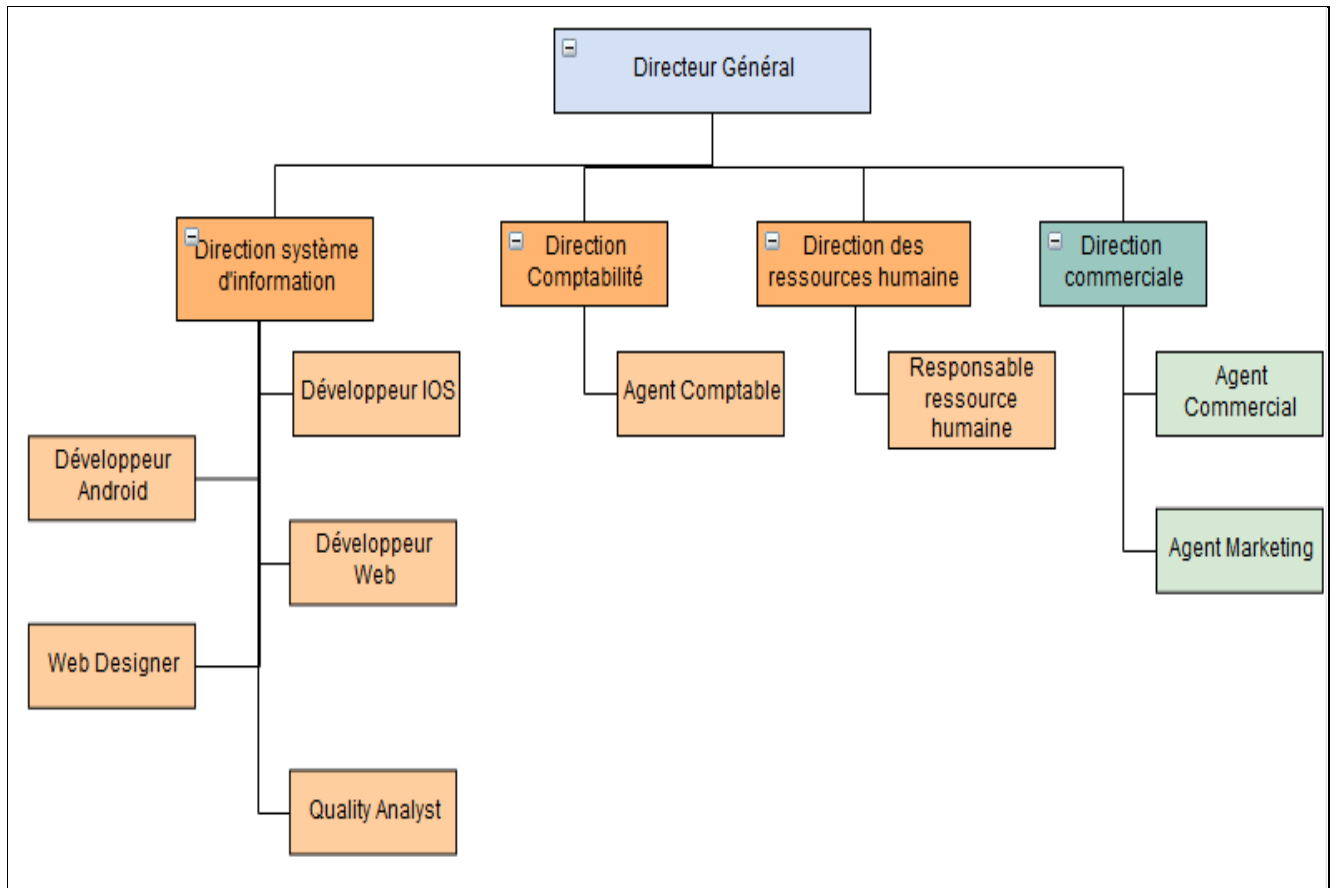


Figure 1. Organigramme NM-Developpement

2. Cahier des charges :

2.1 Objectif du projet :

L'idée du projet vient d'une étude effectuée sur les réseaux sociaux commercialisés dans le marché et aussi d'une proposition de l'entité recherche et développement « R&D » de la société.

Créer un nouveau concept de réseau social qui sort de l'ordinaire et de la vie virtuelle vers le contact direct. Ce réseau social devrait permettre aux internautes de se regrouper avec des nouvelles personnes qui partagent les mêmes intérêts, des amis, des collègues de travail pour des afterwork par exemple,...etc.

2.2 Les différentes tâches à réaliser:

Le travail à réaliser s'articule autour des étapes suivantes :

- ✓ Préparation d'un document de spécification fonctionnelle de l'application demandée.
- ✓ Préparation de la vue conceptuelle du produit.
- ✓ Préparation du Backlog Produit sur lequel nous allons se baser dans le développement.
- ✓ Commencer par le développement de la partie Webservice.
- ✓ Préparation et déroulement des tests en parallèle à chaque « sprint ».

Notons que le travail est partagé avec d'autres intervenants de la même boîte à savoir : Des designers, des intégrateurs, et des développeurs, vu que le projet est consistant et qu'il ne pourra pas être achevé dans une période assez courte que celle du stage.

2.3 Méthodologie à Suivre dans le projet :

La méthodologie appliquée est une méthodologie Agile « Scrum ».

Cette dernière est choisie car elle nous permet de s'adapter aux changements et de livrer des fonctionnalités séparées, afin de donner un aperçu sur le résultat final du produit.

Le travail est fait sous forme d'itération inclut les travaux de conception, de spécifications fonctionnelles, de développement et de test.

Cette méthodologie est constituée principalement sur trois rôles, et quatre réunions.

Les rôles sont:

- Le Product Owner.
- Le Scrum Master qui le garant de l'application de la méthodologie Scrum.
- L'équipe de développement et de test qui réalise le produit.

Les réunions sont ; Planification du Sprint, Stand up meeting, Revue de Sprint, et Rétrospective de Sprint.

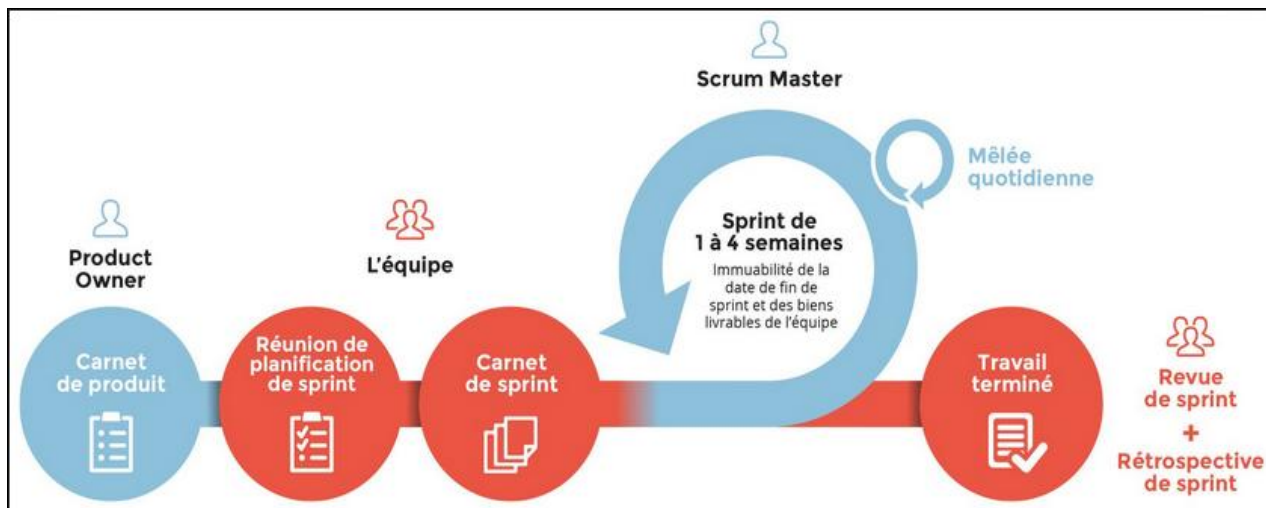


Figure 2. Cycle de vie d'un projet Scrum

3. Etude de l'existant :

Les réseaux sociaux désignent les différentes relations entre les individus et la façon dont celles-ci se structurent. Ils présentent un espace permettant aux personnes inscrites dedans, de communiquer entre elles en mode virtuel, partout dans le monde et d'une manière instantanée.

Les réseaux sociaux sont utilisés principalement par deux catégories d'individus dans des buts différents :

- Par les particuliers, qui utilisent les réseaux sociaux pour partager des informations, des liens, des centres d'intérêts, rechercher des personnes, publier des photos, des vidéos, etc...
- Et par les professionnels qui les utilisent pour générer du business, faire connaître leur entreprise, leurs services, suivre les tendances actuelles, optimiser leurs recrutements, chercher des offres d'emploi, etc...

3.1 Etude comparative des réseaux sociaux:

Aujourd'hui il existe une multitude de réseaux sociaux sur Internet :

Les réseaux sociaux généralistes, professionnels, sur mobile uniquement, payants ou gratuits...

Chacun d'entre eux répond à un besoin bien précis.

Nous pouvons distinguer plusieurs catégories de réseaux sociaux qui sont utilisés pour des fins différentes, citons à titre d'exemples:

a. Les réseaux sociaux généralistes :

Leur objectif est de connecter les individus, de partager des informations, des liens et de créer du contenu qui sera ainsi partagé. Tel que les exemples ci-dessous :

Facebook : C'est le plus large réseau social au monde. Il a été créé en 2004 par Mark Zuckerberg. Il offre une inscription gratuite, et donne la possibilité de partager des photos, des vidéos ou des liens. Il propose également de nombreuses applications telles que des jeux, permet de former des groupes et des pages personnalisées et inclut également un système de messagerie instantanée. [2]

Twitter : Il Permet de suivre librement des utilisateurs et de partager des messages courts limités à 280 caractères (140 caractères auparavant). Il est utilisé comme un espace de « tchat » public pour interagir avec des amis, des entreprises, médias, journalistes ... [3]

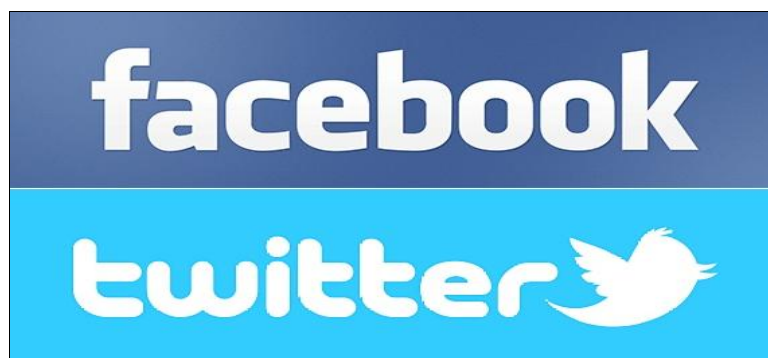


Figure 3. Logo Facebook & Twitter

b. Les réseaux sociaux professionnels :

Ce sont des réseaux sociaux à usage exclusivement professionnel, orientés sur la mise en valeur et les échanges professionnels de ses membres. Les plus connus sont « LinkedIn » et « Viadeo ».

LinkedIn : Lancé en mai 2003 par Reid Hoffman et Allen Blue, il est devenu le leader mondial des réseaux sociaux professionnels. Il permet la mise en relation entre des professionnels, facilite le recrutement de nouveaux collaborateurs et offre aussi un service de recherche de nouveaux clients et partenaires.[4]

Viadeo : Lancé en mai 2004 par Dan Serfaty et Thierry Lunati, c'est un réseau social s'adressant aux entreprises voulant augmenter leurs perspectives d'affaires, gérer et développer leur réseau de contacts professionnels et accroître leurs perspectives de carrière.[5]



Figure 4. Logo Viadeo & LinkedIn

c. Les réseaux d'anciens élèves :

Ils permettent de se reconnecter avec les amis d'enfance et de pouvoir consulter les photos de classe, tel que « Copain d'Avant » et « Trombi ».

Copains d'avant : C'est un réseau social Français, créé par CCM Benchmark Group en 2001, il permet aux participants de retrouver d'anciens camarades, avec qui ils ont partagé leur scolarité, ainsi que leurs activités associatives et professionnelles.[6]

Trombi : C'est un réseau social Français, créé en 2000 par André Pitié. Il aide ses inscrits à trouver et rester en contact avec leurs amis de l'école maternelle jusqu'au lycée et de l'enseignement supérieur. [7]



Figure 5. Logo Copainsd'avant & Trombi

d. Les réseaux sociaux visuels :

Ils permettent de partager des photos et des vidéos en public ou en privé, tel que « Instagram » et « Snapchat ».

Instagram : C'est un réseau social de partage de photos et de vidéos qui appartient à Facebook, lancée par l'Américain Kevin Systrom³ et le Brésilien Michel Mike Krieger en octobre 2010. Il permet de partager des photographies sur un réseau d'amis ou un réseau public, de noter et de commenter sur les clichés déposés par les utilisateurs.[8]

Snapchat : C'est une application gratuite de partage de photos et de vidéos, développé par des étudiants de l'université Stanford en Californie. Cette application permet d'envoyer des messages avec des images ou vidéos qui s'autodétruisent au bout d'une seconde à un temps infini et notamment des selfies.[9]

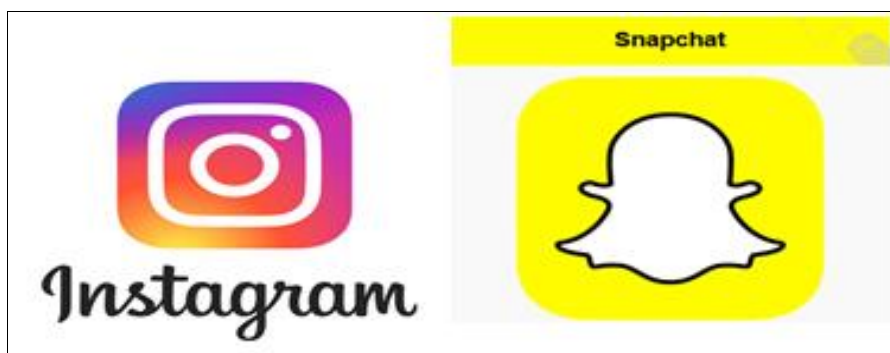


Figure 6. Logo Snapchat & Instagram

e. Les réseaux sociaux vidéo :

Ce sont des plateformes d'hébergement de vidéos où l'on peut regarder et partager des vidéos mises en ligne, tels que « YouTube » et « Dailymotion ».

Dailymotion: C'est un média social créé par Benjamin Bejbaum Et Olivier Poitrey en 2005, il propose un service d'hébergement, de partage et de visionnage de vidéo gratuitement en ligne. [10]

YouTube: C'est un site web d'hébergement de vidéos et un média social sur lequel les internautes peuvent envoyer, regarder, commenter, évaluer et partager des vidéos. Il est créé en février 2005 par Steve Chen, Chad Hurley et Jawed Karim, et finalement racheté par Google en octobre 2006. [11]

f. Tableau comparatif d'une sélection des réseaux sociaux les plus connus sur le marché :

| Réseaux Sociaux | Facebook | Youtub | Instagram | Linkedin |
|------------------------|--|---|--|--|
| Objectifs | Construire et fidéliser une communauté de fans et communiquer auprès d'eux. | Moteur de recherche et propose des vidéos de très haute qualité | Partager des photos personnalisées et construire des relations. | Création de Réseau professionnel |
| Membres | 2.1 milliard | 1,9 milliard | 1 milliard | 294 millions |
| Population | Tout le monde | Tout le monde | Tout le monde | Professionnels |
| Portabilité | Web, Android et IOS | Web, Android et IOS | Web, Android et IOS | Web, Android et IOS |
| Type de contenu | Textes, vidéos, photos, évènements, jeux... | Vidéos | Images et vidéos courts | Textes, images, vidéos |
| Payant | Gratuit | Gratuit | Gratuit / Payant | Gratuit / Payant |
| | -Présente un marché important de clients potentiels -Innovation constante. -Forte croissance | -L'optimisation pour les moteurs de recherche, référencement naturel. -Contenu très varié -Droit d'auteur | -Facile à utiliser, -Impacts sur les ventes -Visuels de très haute qualité Présenter et conserver | -Développer son business -Partage de connaissance et de compétences |

| | | | | |
|-------------------------------|--|--|--|---|
| <p>Points positifs</p> | <ul style="list-style-type: none"> -Création des évènements -Proposition des jeux en ligne. -Permet de garder un lien avec la famille et les amis/connaissances éloignés. -Permet de créer et suivre les groupes que nous aimons. -Permet de diffuser des photos de soi, de ses amis, des vidéo... -Aide phénoménale pour diffuser des informations et se tenir informé de tout. | <ul style="list-style-type: none"> respectés -Qualité vidéo top -Mise en ligne des vidéos -Suivre des chaines spécifiques à intérêt personnel. | <ul style="list-style-type: none"> une trace de sa vie d'une manière plus artistique -S'inspirer par des amis et les influenceurs -Développer une image de marque -Profiter du pouvoir des influenceurs -Relativement facile de promouvoir ses publications avec des #hashtags. | <ul style="list-style-type: none"> professionnels -Accroître l'influence de votre entreprise -Effectuer une veille active sur votre environnement -Chercher des candidats. -Chercher un travail -Facilité d'utilisation. -Créer de nouvelles opportunités. |
| <p>Points négatifs</p> | <ul style="list-style-type: none"> -Manque de visibilité sur le nombre de participant à un évènement crée. -Nombre élevé de faux profils. -Un moyen de faire subir aux autres du harcèlement moral. -Favorise les tendances agressives et les comportements anti-sociaux. -Les personnes utilisant trop sont les plus exposés à présenter des désordres | <ul style="list-style-type: none"> -Exigence de contenu de qualité -Investissement considérable (logiciel, temps, argent...) | <ul style="list-style-type: none"> -Seulement des images et vidéo courts -Accès limité sur ordinateur -Limites du profil -La gestion du compte doit se faire sur appareil mobile. -Très peu de statistiques fournies par l'application -Difficile d'évaluer son audience et sa performance. -L'utilisation d'applications tierces pour retoucher -Possède un effet néfaste | <ul style="list-style-type: none"> -Outil limité -Evoluant peu. -Pas de moteur de recherche. -Publicité trop cher. -Un réseau social chronophage. -Une plateforme orientée B2B. -Peu adapté aux professionnels ayant une clientèle de particuliers. |

| | | |
|---|--|---|
| psychologiques. -Pas de suppression fictive de profil -On passe énormément de temps dessus: perte de notion du temps -On prend beaucoup moins le téléphone pour prendre des nouvelles de nos amis. | | sur la santé mentale des jeunes (selon un sondage de la Royal Society for Public Health.) suite aux photos retouchées, le physiques parfaits et les vies idéales. Ce qui diminue la confiance en soi. |
|---|--|---|

Tableau 1. Tableau comparatif des réseaux sociaux les plus populaires

3.2 Choix de la solution :

Partant de cette étude comparative, nous pouvons noter que les réseaux sociaux apportent beaucoup d'avantages, mais peuvent aussi s'avérer très dangereux pour l'être humain s'il tombe dans l'addiction.

Ces outils peuvent créer une dépendance, des troubles de sommeil et même entraîner une dépression. Certains parmi eux, peuvent nous renvoyer une mauvaise image de nous-même, et aggravent le sentiment d'isolement et de solitude. D'après les études menées par la RSPH (Royal Society for Public Health), les réseaux sociaux participent fortement à affecter la santé des jeunes. Cette organisation prouve que les taux d'anxiété et de dépression ont augmenté de 70% durant les 25 dernières années et que les réseaux sociaux sont directement en cause de cette croissance. [12]

En outre, une autre étude menée par Melissa Hunt, de l'Université de Pennsylvanie aux Etats-Unis, publiée dans le Journal of Social and Clinical Psychology, qui a prouvé que diminuer le temps passé sur les réseaux sociaux pouvait réduire l'anxiété et la dépression. [13]

En plus, le fait d'avoir un nombre énorme d'amis en virtuel, peut ne pas changer la vie réelle surtout si nous les connaissons pas.

D'où la nécessité de renforcer les relations humaines par les contacts directs, et les rencontres face-to-face en utilisant des solutions digitalisées qui facilitent l'organisation de ces rencontres et les rendent plus agréables.

Conclusion

Dans ce chapitre nous avons présenté le contexte général du projet, et une étude sur les réseaux sociaux existants sur le marché, ce qui nous a permis d'avoir une vue synthétique sur les préférences des internautes et de savoir les points faibles de chacun, afin de créer une solution qui répondait au mieux aux besoins utilisateurs et qui favorise les relations humaines, c'est ce que nous allons bien détailler dans le prochain chapitre.

Chapitre II : Spécification des besoins

Introduction

Ce chapitre présentera le volet fonctionnel du projet, les grandes exigences produit à développer ainsi que les acteurs ciblés par cette application avec leurs cas d'utilisation relatifs.

Nous terminons par une petite conclusion qui résumera tous les détails présentés auparavant.

1. Identification des acteurs

Nous avons principalement deux acteurs dans ce projet, dont le deuxième regroupe l'ensemble de staff technique qui travaille sur l'application en amélioration continue:

➤ **L'utilisateur de l'application (l'internaute) :**

C'est l'utilisateur principal de l'application qui bénéficie de ses services en réagissant avec ses fonctionnalités après une inscription.

➤ **Les administrateurs de l'application :**

Ceux qui gèrent l'administration de l'application, implémentent les évolutions, gèrent les sauvegardes et les backups, interviennent pour vérifier les réservations des restaurants et des cafés ...

Les administrateurs représentent toutes les personnes appartenant au staff technique de l'application.

2. Besoins Fonctionnels :

Notre application a plusieurs objectifs à réaliser à court et à moyen terme, tel que :

- Faciliter et organiser des rencontres entre les personnes utilisant l'application.
- Permettre l'interagir en direct entre les abonnés en utilisant une messagerie instantanée
- Fournir une alternative aux environnements de rencontres virtuels.
- Être une plate-forme mondiale de premier plan pour les planifications des rencontres Face to Face.
- Explorer continuellement de nouvelles idées afin de faire de ces rencontres une expérience amusante et plus réaliste.

Ces objectifs seront réalisés à travers les fonctionnalités suivantes, proposées par notre application.

2.1 Inscription :

Comme tous les réseaux sociaux, l'utilisateur doit s'inscrire à l'application pour pouvoir bénéficier de ses services.

L'utilisateur devra donc passer par une authentification en utilisant son compte Facebook, pour pouvoir enregistrer ses informations lors d'une première connexion. Par la suite, et dès que l'inscription est terminée, l'utilisateur aura accès à son espace personnel.

2.2 Authentification :

L'authentification se fait en se connectant avec l'identifiant Facebook de l'utilisateur.

Cette fois, aucune manipulation de données utilisateur ne sera effectuée, car toutes ces informations ont été déjà enregistrées au moment de la première inscription.

Après authentification l'utilisateur sera automatiquement redirigé vers son espace personnel.

2.3 Messagerie et discussion instantanée :

L'application propose à l'utilisateur un service de messagerie instantanée, cette fonctionnalité permet de faciliter la communication entre les membres inscrits.

Ce service permettra à l'utilisateur de saisir, envoyer des messages, des images et des émoticônes à son interlocuteur et éventuellement en recevoir.

2.4 Notification :

L'utilisateur aura un service de notification, qui lui donnera une visibilité sur toutes les actions effectuées sur ses propositions de rencontre, que ce soit une acceptation, un refus, un commentaire déposé sur un déjeuner effectué, des demandes d'amitié, etc...

2.5 Participation à une proposition de rencontre :

Toutes les propositions visibles pour un utilisateur que ce soit privées ou publiques seront affichées dans sa page d'accueil. Il aura la possibilité d'accepter ou refuser cette action.

Le principe est le suivant ; dès que l'utilisateur reçoit une demande de rencontre de la part d'un autre utilisateur de l'application, il aura une notification qui lui sera envoyée pour lui informer que cette personne lui a invité à un café à titre d'exemple.

Lorsque l'utilisateur accepte cette invitation, l'écran de la proposition lui sera affiché avec les détails liés à cette proposition.

Et s'il refuse, l'utilisateur sera redirigé vers son écran d'accueil et une notification sera envoyée au détenteur de la proposition pour l'informer de ce refus.

2.6 Ajouter une proposition de rencontre :

C'est la fonction principale de l'application, l'utilisateur après son inscription, pourra automatiquement créer une proposition de rencontre dans un endroit bien ciblé.

En ajoutant une nouvelle proposition, un écran de localisation sera affiché à l'utilisateur en lui indiquant les endroits disponibles dans un rayon de 1500 m où il pourra planifier sa rencontre. Dès que l'utilisateur sélectionne un endroit de rencontre, cette action lui permet de compléter le reste des informations liées à sa proposition tel que ; la date et l'heure de la rencontre, le nombre maximum de personnes à inviter, le thème de la rencontre avec une courte description du contenu...

Par la suite il devra valider sa saisie pour que sa proposition soit visible aux autres utilisateurs de l'application.

2.7 Gérer le Profil Utilisateur :

L'utilisateur aura la possibilité de gérer son profil, il pourra afficher toutes ses informations, les compléter ou les modifier.

Les informations utilisateurs peuvent être, Nom et prénom de ce dernier, son adresse, ses centres d'intérêts, sa photo de profil, sa date de naissance, etc...

2.8 Gestion des Contacts/ Amis:

L'utilisateur aura la possibilité de gérer son réseau personnel. Le terme 'réseau ' signifie sa liste d'amis/contacts qui compose un réseau privé pour lui.

L'utilisateur pourra chercher, ajouter, ou supprimer quelqu'un et aussi créer une liste de favoris d'amis selon son choix.

3. Diagrammes de cas d'utilisation :

Dans cette section, nous allons présenter les différents diagrammes de cas d'utilisation, qui modélisent le comportement de l'application et qui permettent de capturer ses exigences listées dans le paragraphe précédent. Ils ont pour but aussi d'identifier les interactions entre l'application et ses acteurs.

3.1 Diagramme de cas d'utilisation Général :

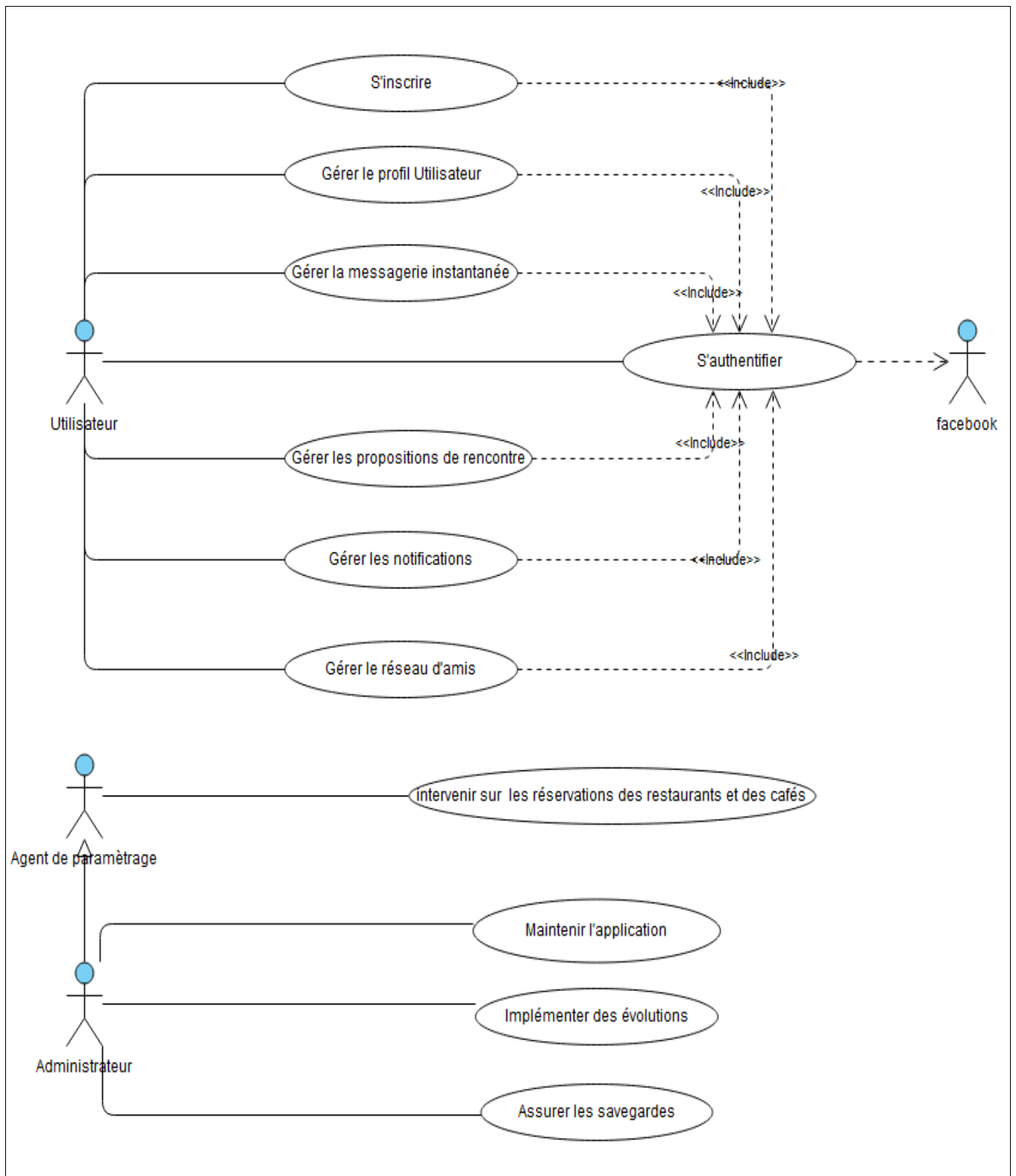


Figure 7. Diagramme de Cas d'utilisation Général

3.2 Diagramme de cas d'utilisation d'Inscription par authentification :

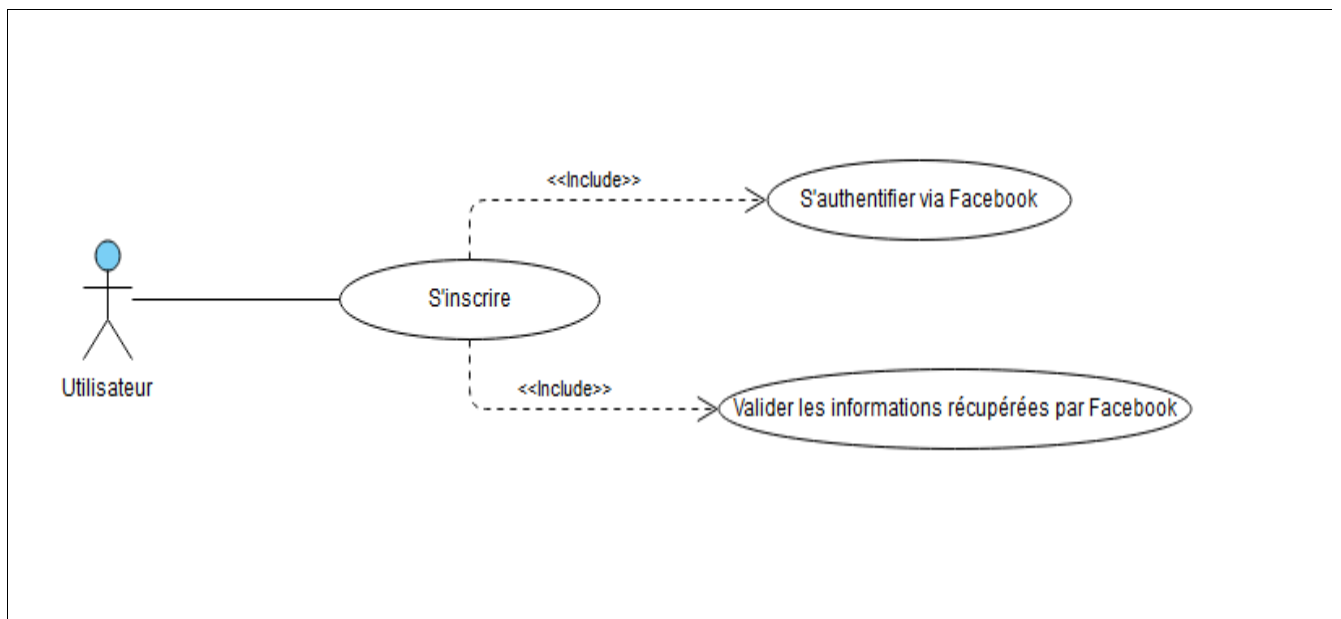


Figure 8. Diagramme de cas d'utilisation Inscription par authentification

Description textuelle 'Inscription par authentification' :

| | |
|--------------------------|---|
| Cas d'utilisation | Inscription par authentification |
| Acteur | Utilisateur/Internaute |
| Description | L'inscription par authentification vient du fait que l'utilisateur sera inscrit en se connectant par son compte Facebook qui permet de récupérer ses informations complètes et nécessaires à une telle action |
| Pré-condition | L'utilisateur doit installer l'application puis ouvrir la page d'authentification |
| Scénario nominal | L'utilisateur clique sur le bouton « Se connecter par Facebook ». Le système va y aller vers Facebook pour vérifier la présence du compte et récupérer ses informations. Le système récupères les informations utilisateur et les affiche |

| | |
|-----------------------------|--|
| | pour validation utilisateur L'utilisateur valide les données récupérées et se connecter à l'application. |
| Post-condition | Le compte utilisateur est inscrit à l'application et pourra voir/créer des propositions de rencontres |
| Scénario d'exception | <u>Scénario1</u> : le compte saisi n'est pas reconnu par Facebook, une erreur est renvoyée et l'utilisateur est amené à choisir un autre compte valide. <u>Scénario2</u> : Une erreur est survenue lors de la récupération des données utilisateur, et l'utilisateur devra essayer encore une fois. |

Tableau 2. Description Textuelle du Cas d'utilisation Inscription par authentification

3.3 Diagramme de cas d'utilisation d'Authentification :

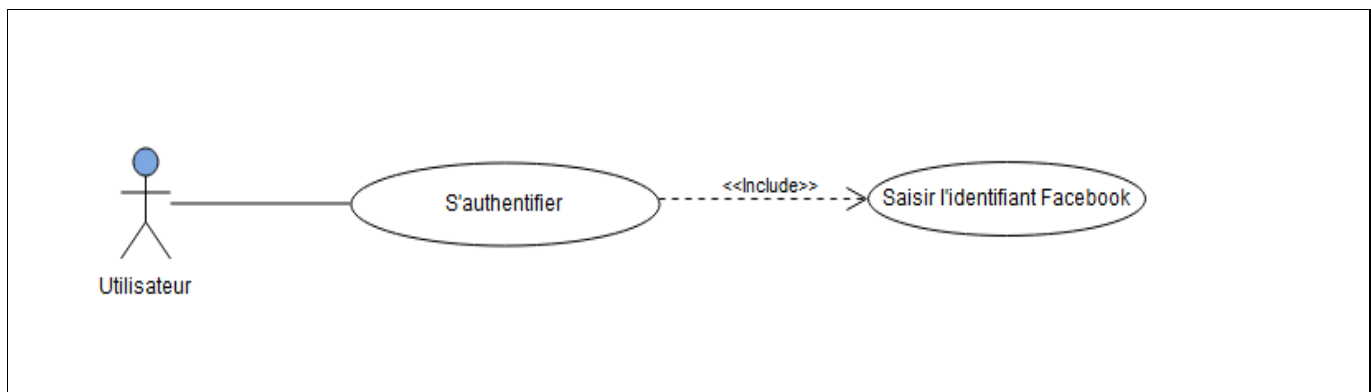


Figure 9. Diagramme de cas d'utilisation Authentification

Description textuelle 'Authentification' :

| | |
|--------------------------|---|
| Cas d'utilisation | Authentification |
| Acteur | Utilisateur/Internaute |
| Description | L'authentification se fait en se connectant avec le l'identifiant Facebook de l'utilisateur |
| Pré-condition | L'utilisateur ouvre l'écran d'authentification |

| | |
|-----------------------------|--|
| Scénario nominal | <p>L'utilisateur clique sur le bouton « Se connecter par Facebook ».</p> <p>Le système va y aller vers Facebook pour vérifier la présence du compte déjà enregistré sans récupération des informations.</p> <p>Une fois la vérification est faite avec succès l'utilisateur est connecté à l'application</p> |
| Post-condition | L'utilisateur accède à l'application. |
| Scénario d'exception | Authentification échouée suite à un compte non reconnu par Facebook. Un message d'erreur doit s'afficher à l'utilisateur pour lui informer du type d'erreur. |

Tableau 3. Description textuelle du cas d'utilisation Authentification

3.4 Diagramme de cas d'utilisation de Messagerie instantanée :

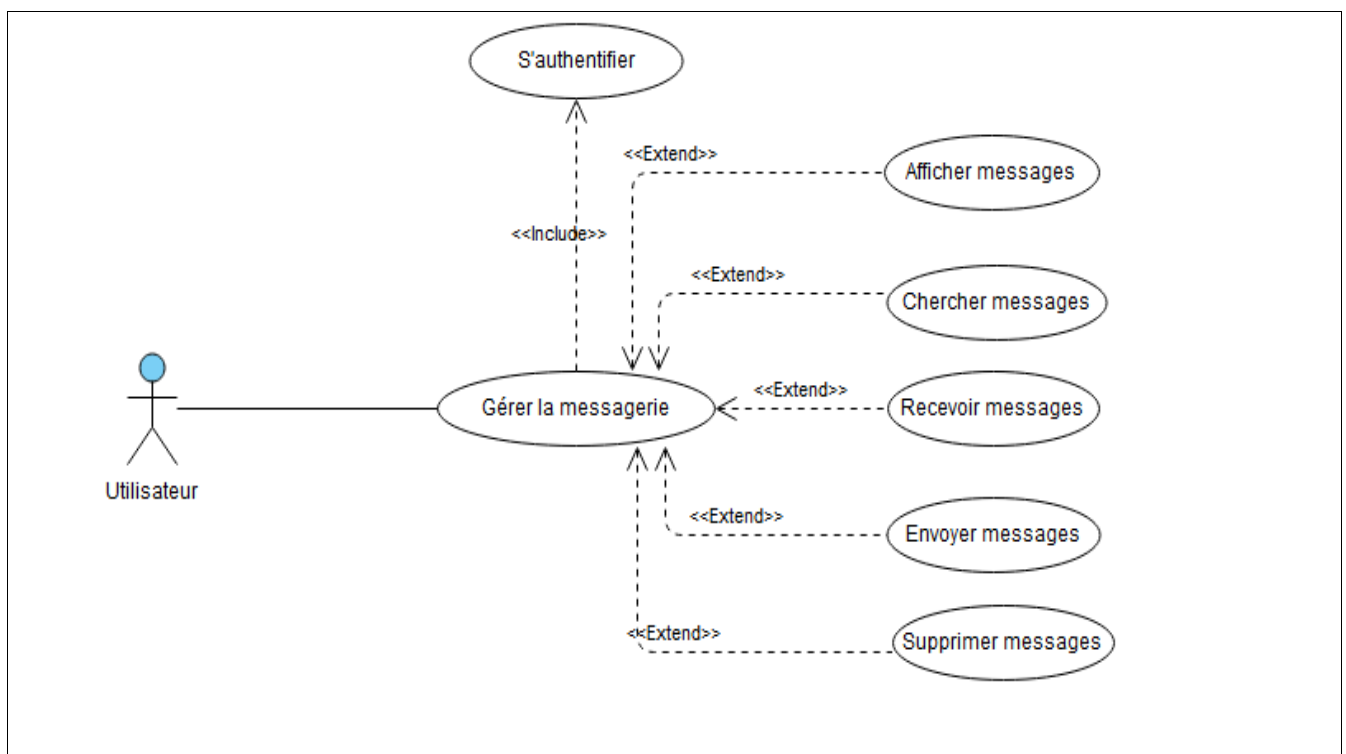


Figure 10. Diagramme de cas d'utilisation de Messagerie instantanée.

Description textuelle 'Messagerie instantanée' :

| | |
|--|---|
| Cas d'utilisation | Messagerie instantanée |
| Acteur | Utilisateur/Internaute |
| Description | La gestion de messagerie permet à l'utilisateur d'envoyer, recevoir, chercher, et supprimer des messages, tout en s'authentifiant à l'application. |
| Pré-condition | L'utilisateur doit s'authentifier à l'application |
| Scénario nominal/ <u>Envoyer un message</u> | L'utilisateur ouvre l'écran de discussion Choisi un ami en cliquant dessus pour ouvrir l'écran de tchat Saisie un message et valide l'envoi Le message est envoyé vers l'utilisateur ciblé |
| Post-condition | Le message est envoyé à la personne concernée. |
| Scénario d'exception | Envoi de message échoué. Un message d'erreur doit s'afficher à l'utilisateur pour lui informer du type d'erreur (erreur de connexion ...). |

Tableau 4. Description textuelle du cas d'utilisation Messagerie instantanée

3.5 Diagramme de cas d'utilisation de gestion de Notification :

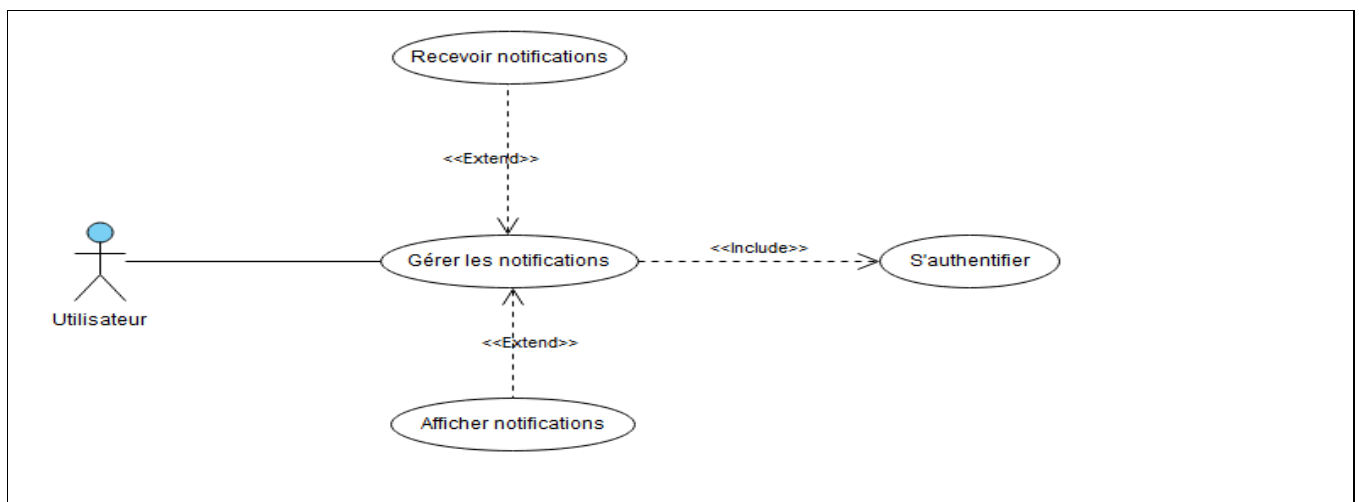


Figure 11. Cas d'utilisation de Gestion de Notification

Description textuelle 'Notification' :

| | |
|---|--|
| Cas d'utilisation | Notification |
| Acteur | Utilisateur/Internaute |
| Description | La gestion de notification permet à l'utilisateur d'afficher et recevoir des notifications. |
| Pré-condition | L'utilisateur doit s'authentifier |
| Scénario nominal/Recevoir notification | L'utilisateur A envoie une invitation de rencontre à un utilisateur B. L'utilisateur B reçoit une notification, pour lui informer de cette invitation. L'utilisateur B accepte la proposition de rencontre, l'utilisateur A détenteur de la proposition reçoit de son côté une notification de l'acceptation de rencontre de l'utilisateur B. |
| Post-condition | La notification est correctement envoyée dans les deux sens. |
| Scénario d'exception | Envoi de notification échoué. L'utilisateur ne reçoit pas une notification de proposition de rencontre qui lui a été envoyée. D'où la possibilité de ne pas pouvoir accepter cette proposition dans le délai. |

Tableau 5 . Description textuelle du Cas d'utilisation Notification

3.6 Diagramme de cas d'utilisation de Gestion de propositions de rencontres:

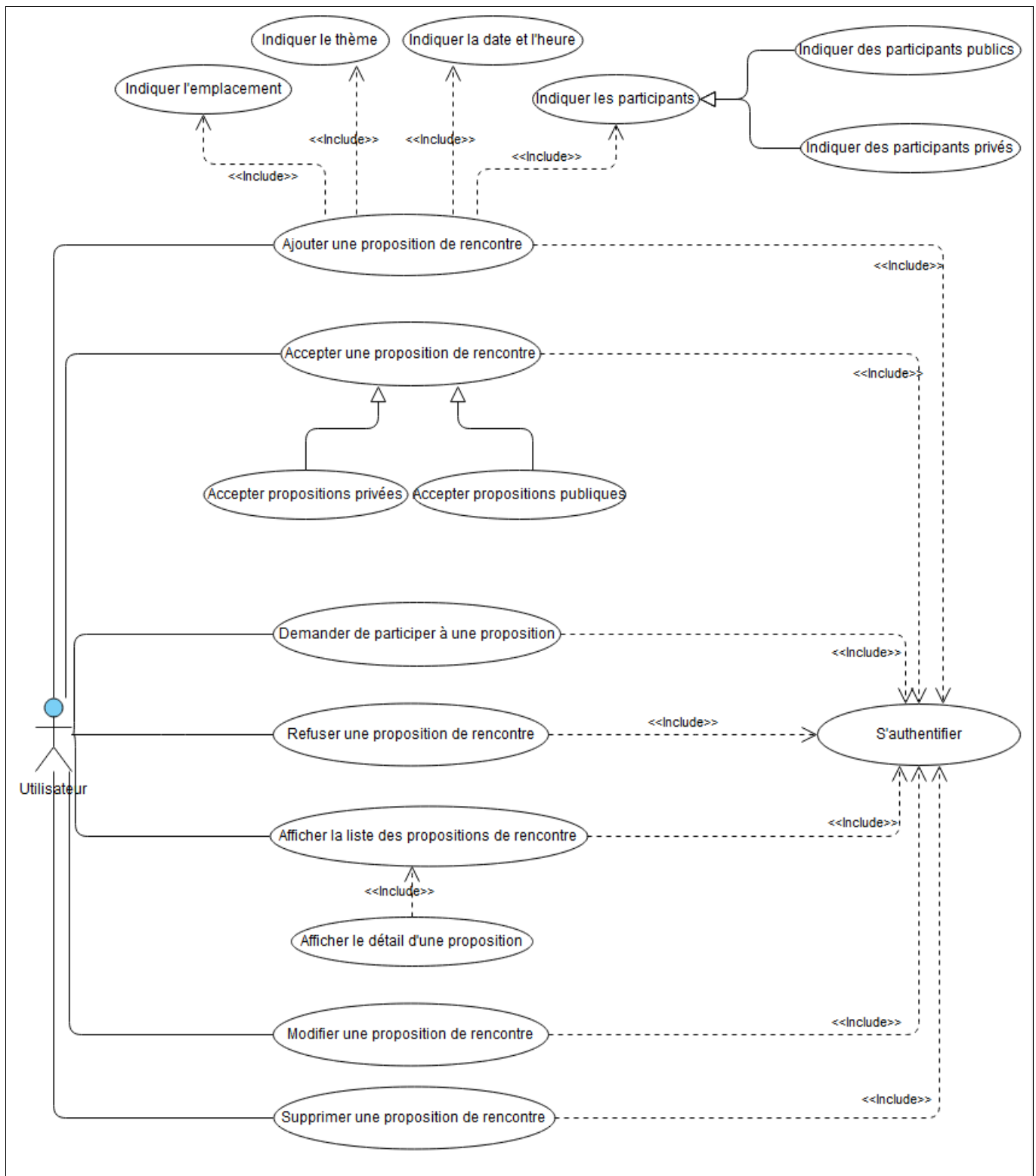


Figure 12. Diagramme de cas d'utilisation Gestion de proposition de rencontre

Description textuelle 'Gestion de propositions de rencontres' :

| | |
|--|--|
| Cas d'utilisation | Gestion de propositions de rencontres |
| Acteur | Utilisateur/Internaute |
| Description | <p>La gestion de proposition est la fonctionnalité principale de l'application, elle permet d'ajouter des propositions de rencontre, accepter ou refuser aussi des propositions des autres utilisateurs.</p> <p>L'utilisateur aura la possibilité de gérer ces propositions en faisant des modifications, affichage, et suppression.</p> |
| Pré-condition | L'utilisateur doit s'authentifier |
| Scénario nominal/ <u>Ajout d'une proposition de rencontre</u> | <p>L'utilisateur doit forcément indiquer les éléments ci-dessous pour pouvoir créer une proposition de rencontre :</p> <ul style="list-style-type: none"> -Choisir l'emplacement de la rencontre. -Préciser le thème de cette rencontre. -Fixer la date et l'heure de la rencontre -et inviter les participants (soit sélectionner des inviter en privé, soit la publier à tout le monde). -Puis valider la création. |
| Post-condition | <p>La proposition de rencontre est créée et elle est visible sur la page d'accueil de l'application</p> <p>Des notifications seront envoyées aux utilisateurs invités.</p> |
| Scénario d'exception | Création échouée. Un message d'erreur doit s'afficher à l'utilisateur pour lui informer du type d'erreur. |

Tableau 6. Description textuelle Gestion de Propositions de Rencontres

3.7 Diagramme de cas d'utilisation de Gestion de profil utilisateur :

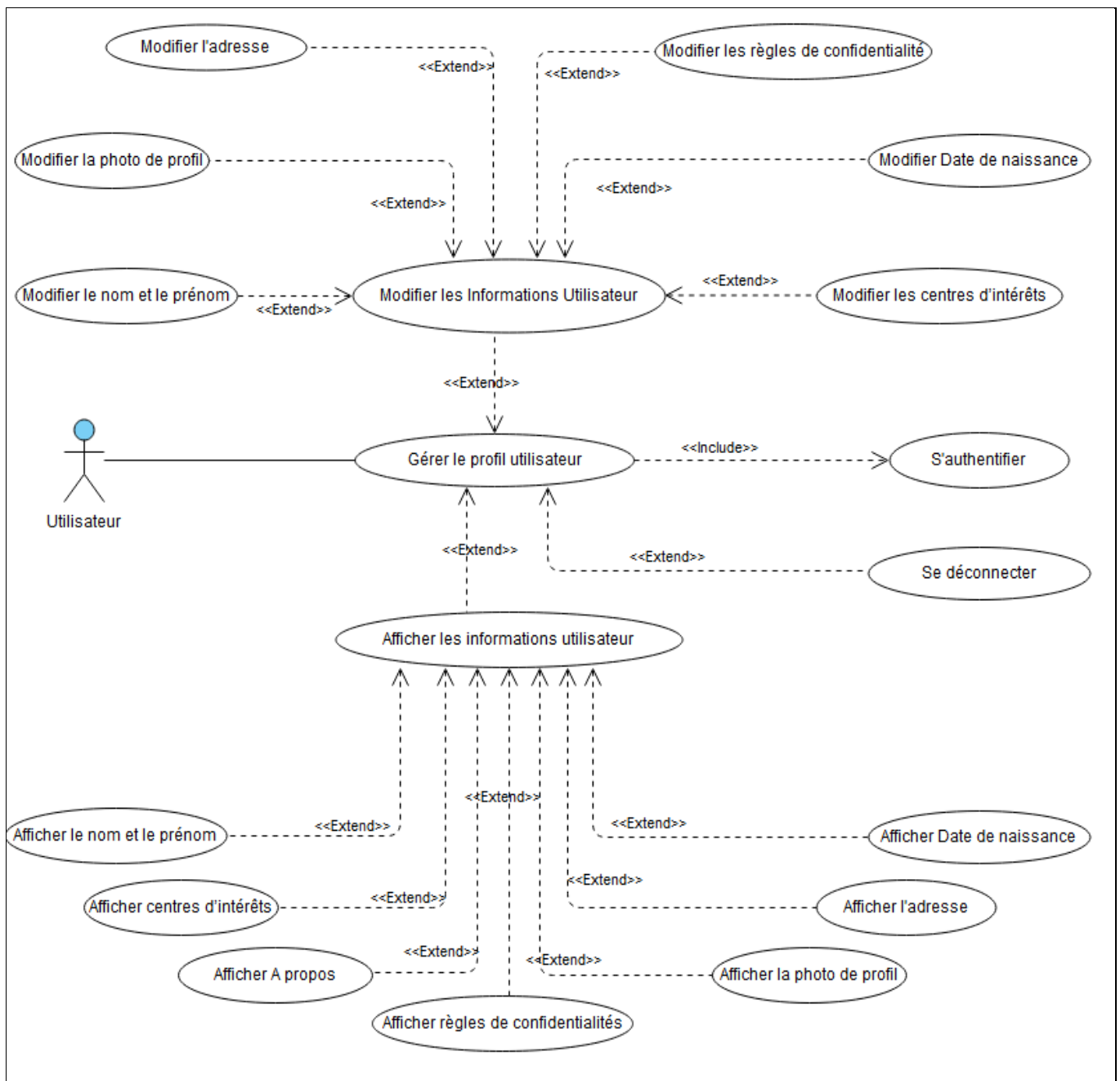


Figure 13. Diagramme de cas d'utilisation Gestion de profil utilisateur

Description textuelle 'Gestion de profil utilisateur' :

| | |
|---|---|
| Cas d'utilisation | Gestion de profil |
| Acteur | Utilisateur/Internaute |
| Description | La gestion de profil, permet à l'utilisateur de gérer son compte sur l'application, afficher ou modifier ses informations. |
| Pré-condition | L'utilisateur doit s'authentifier |
| Scénario nominal/ <u>Modifier Informations</u> | L'utilisateur accès à l'écran profil, Affiche ses informations, Modifie sa photo de profil, ses centres intérêts et son adresse (par exemple) Puis enregistre les modifications apportées. |
| Post-condition | Les informations sont mises à jours et prennent les nouvelles valeurs saisie par l'utilisateur. |
| Scénario d'exception | Modification échouée. Un message d'erreur doit s'afficher à l'utilisateur pour lui informer du type d'erreur (enregistrement échoué ...). |

Tableau 7. Description textuelle Gestion de profil

3.8 Diagramme de cas d'utilisation de gestion de contacts/Amis :

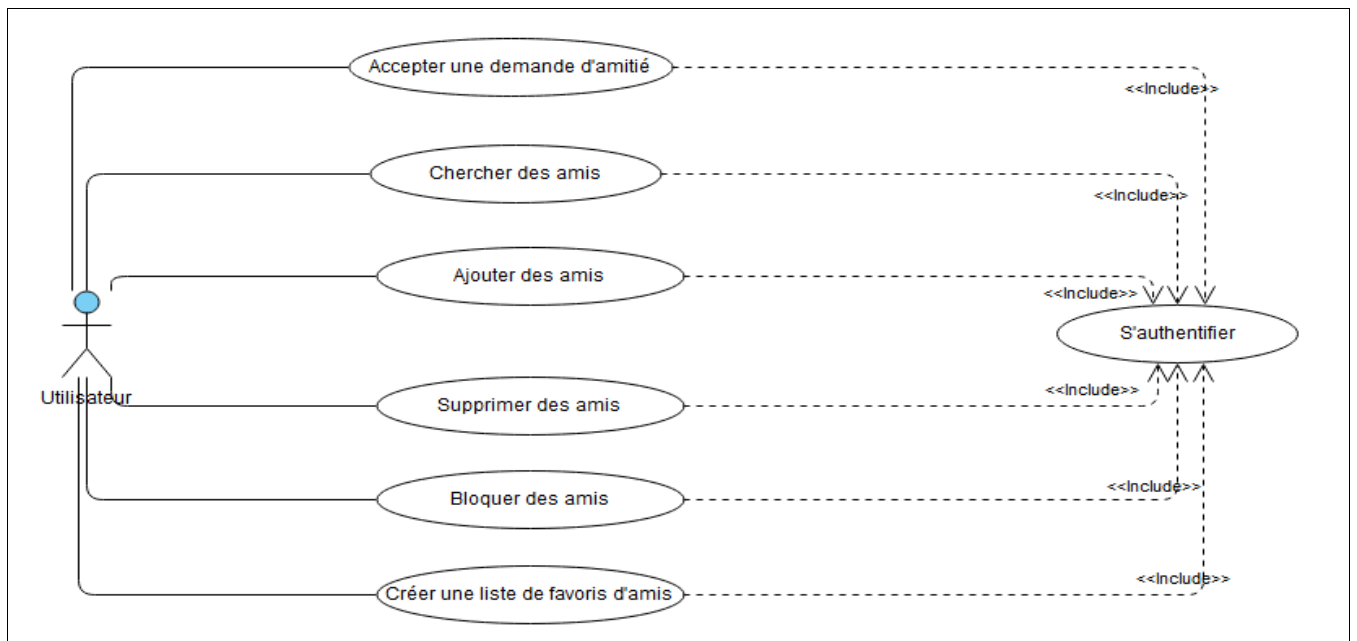


Figure 14. Diagramme de cas d'utilisation de gestion de contacts/Amis

Description textuelle 'Gestion d'amis/contacts' :

| | |
|--|---|
| Cas d'utilisation | Gestion d'amis |
| Acteur | Utilisateur/Internaute |
| Description | La gestion des amis, permet de gérer la liste d'amis de l'utilisateur. Chercher, ajouter, ou supprimer quelqu'un de sa liste et aussi créer une liste de favoris selon son choix. |
| Pré-condition | L'utilisateur doit s'authentifier |
| Scénario nominal/ <u>Ajouter un ami</u> | L'utilisateur clique sur l'icône d'amis Cherche un ami à partir de la barre de recherche qui lui sera proposée, Affiche le profil utilisateur et puis clique sur ajouter. |
| Post-condition | Une demande d'amitié est envoyée à la personne concernée sous forme d'une notification |
| Scénario d'exception | Invitation échouée. Un message d'erreur doit s'afficher à l'utilisateur pour lui informer du type d'erreur (erreur connexion...). |

Tableau 8. Description textuelle Gestion d'amis

4. Besoins Non Fonctionnels

Etant donnée la concurrence du marché, nous allons nous focaliser aussi sur les besoins non fonctionnels, pour pouvoir s'installer dans le marché rapidement et attirer les internautes.

Parmi les besoins non fonctionnels que nous allons travailler dessus, nous notons :

4.1 La sécurité :

Les comptes des utilisateurs doivent être protégés.

La connexion avec login et mot de passe est indispensable.

La visibilité des contenus privés doit être gérée également.

La déconnexion après un temps mort d'inactivité est nécessaire.

4.2 L'ergonomie :

Les écrans doivent être ergonomiques : simples, lisibles, et faciles à utiliser.

Les standards d'ergonomie doivent être appliqués : la densité d'éléments sur les écrans, la disposition et le flux, les couleurs, langages, orthographe, claviers, etc.

4.3 La performance :

Le temps de réponse: le chargement de l'application, ouverture d'écran et des délais de rafraîchissement, etc.

Bien gérer la capacité de l'application à supporter des transactions simultanées par heure et par seconde même.

Capacité de Stockage des données et du contenu des utilisateurs.

4.4 L'intégrité :

L'intégrité et la cohérence des données doivent être assurées à chaque mise à jour, correction ou changement apporter à l'application.

4.5 La maintenabilité:

Le code de l'application doit être compréhensible pour pouvoir assurer son évolution et son extensibilité et répondre aux besoins du marché.

4.6 La disponibilité :

Assurer une très bonne disponibilité de l'application en prenant en compte les heures de maintenance.

Génération de secours: la restauration à l'état précédent lors d'un problème et/ou processus de rétablissement.

Définir la fréquence et les heures des sauvegardes de données

Conclusion :

Dans ce chapitre, nous avons présenté le volet fonctionnel de l'application en précisant les acteurs qui vont interagir avec. Et en élaborant la partie conception dans le chapitre qui suit dans laquelle nous aurons plus de détails sur les interactions et les actions à faire sur notre produit.

Chapitre III : Conception

Introduction

Afin de mieux présenter à quoi sert le système et qui va l'utiliser, et représenter les communications avec et au sein de l'application, nous allons définir les diagrammes suivants:

- Diagramme de séquence
- Diagramme de classe
- Diagramme de base de données

1. Diagrammes de séquence :

Dans cette partie, nous allons identifier les diagrammes de séquences qui représentent graphiquement les interactions entre l'acteur et l'application selon un ordre chronologique bien déterminé.

Ces diagrammes permettent de montrer quelques scénarios choisis des diagrammes des cas d'utilisation mentionnés ci-dessus. (Inscription, Authentification, Messagerie, Gestion de proposition, et Réseau d'amis).

1.1. Diagramme de séquence « Inscription par authentification » :

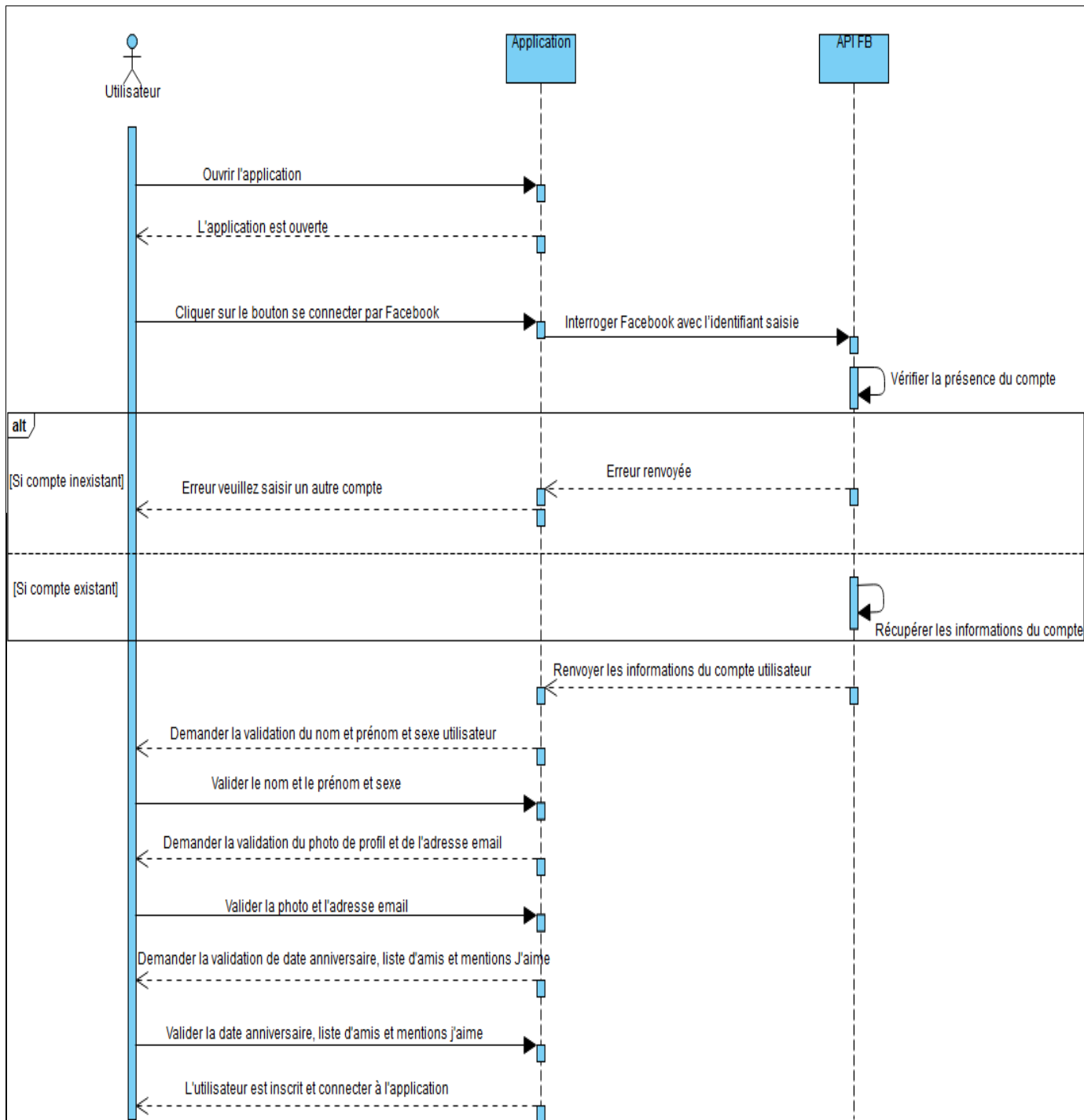


Figure 15. Diagramme de séquence « Inscription par authentification »

1.2. Diagramme de séquence « Authentification » :

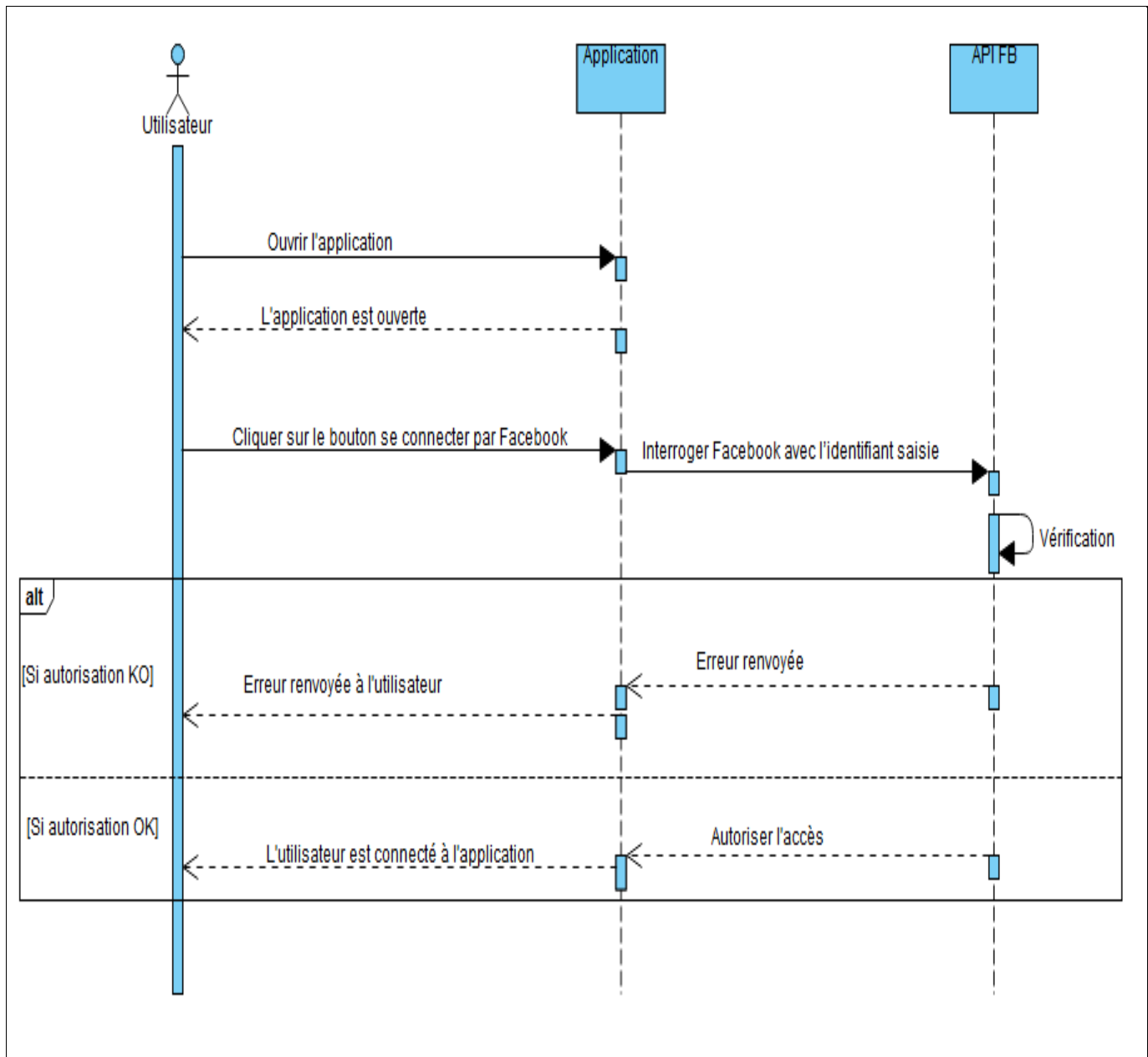


Figure 16. Diagramme de séquence Authentification

1.3. Diagramme de séquence « Messagerie instantanée » :

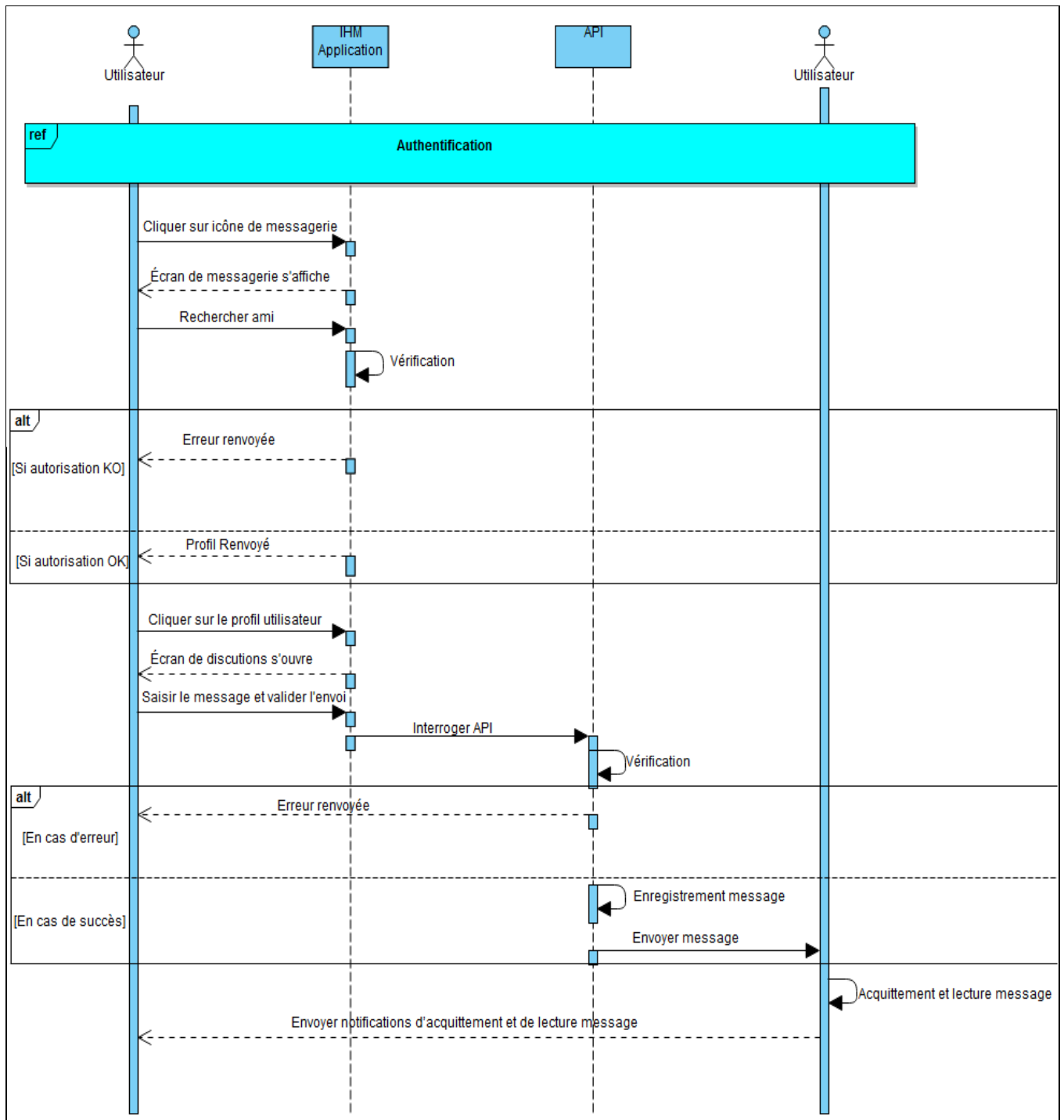


Figure 17. Diagramme de séquence Messagerie instantanée

1.4. Diagramme de séquence « Gestion de proposition » :

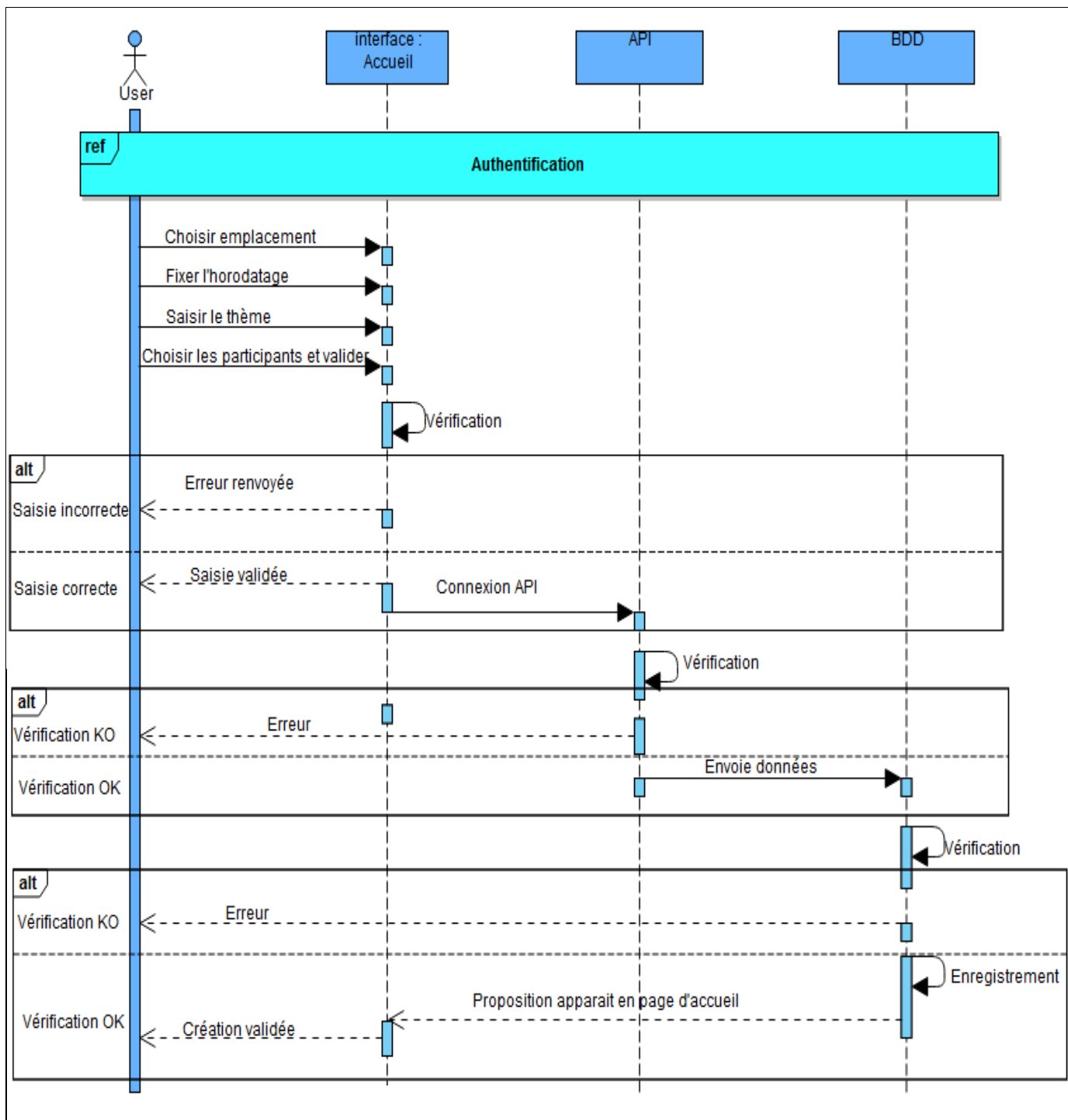


Figure 18. Diagramme de séquence Gestion de propositions de rencontres

1.5. Diagramme de séquence « Gestion de réseau d'amis » :

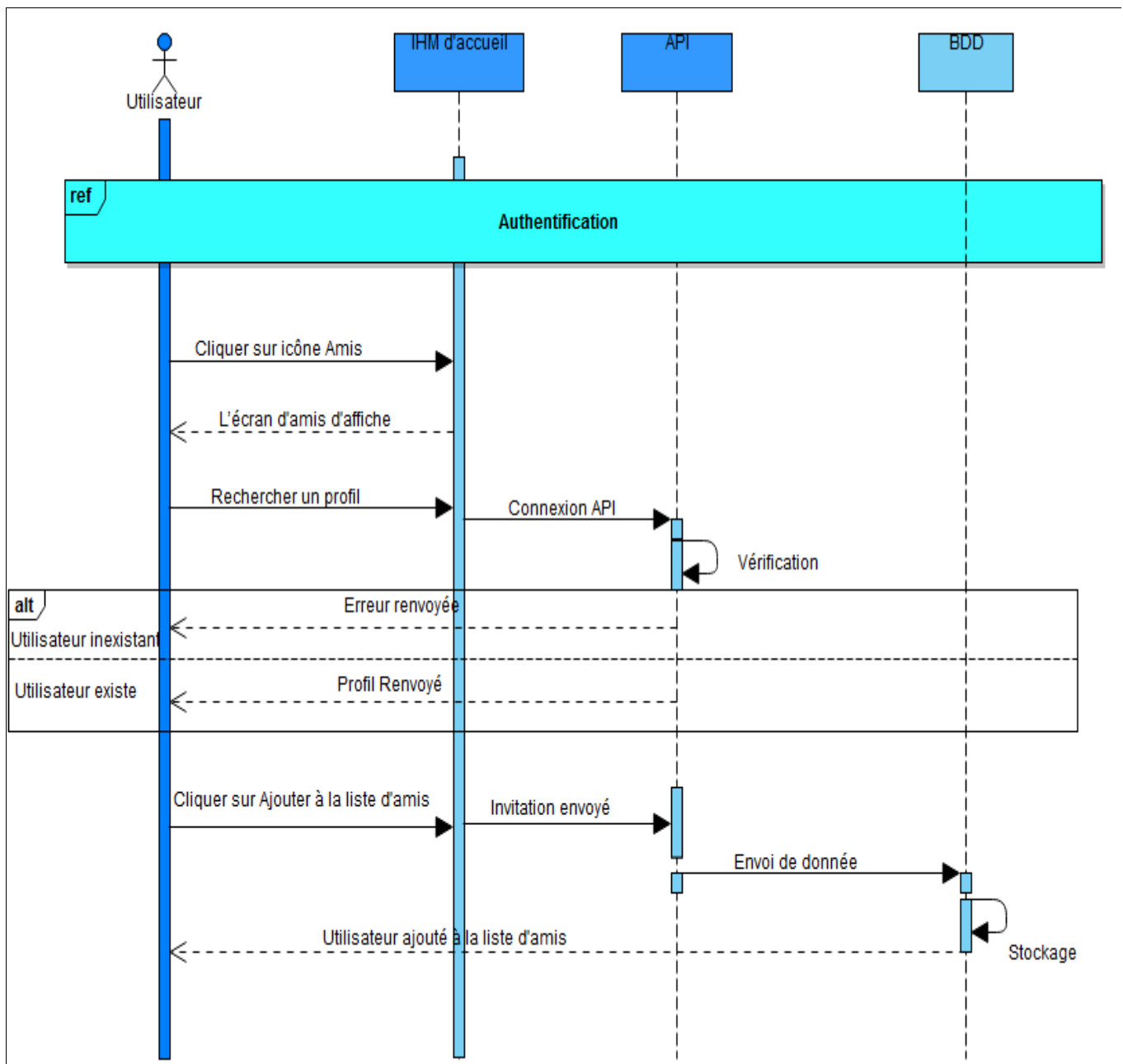


Figure 19. Diagramme de séquence Gestion de réseau d'amis

2. Diagrammes de Classe

Le diagramme de classe est une représentation statique des éléments qui composent le système et de leurs relations.

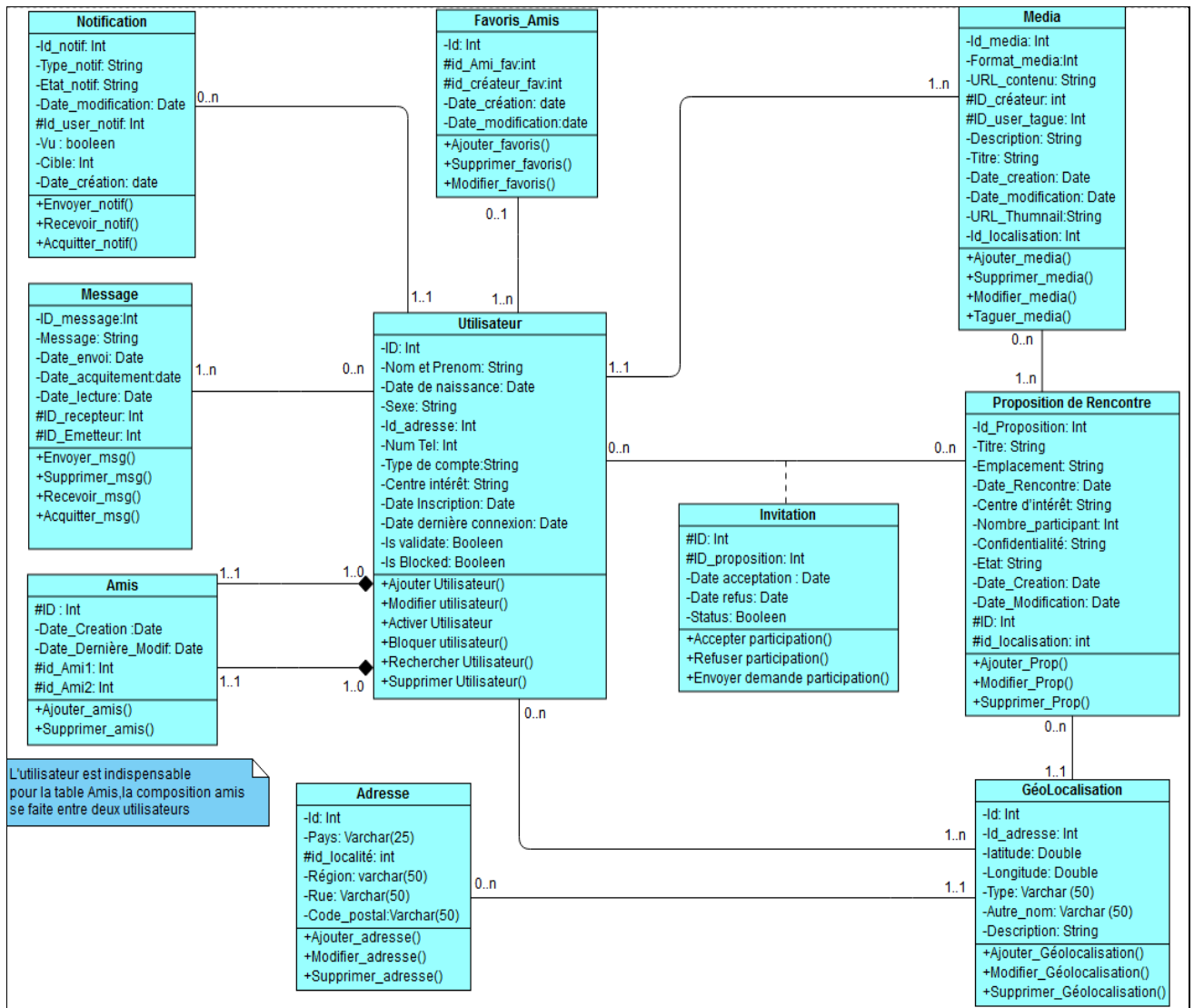


Figure 20. Diagramme de classe général

3. Diagrammes de Base de données :

Le diagramme de Base de données est une représentation des tables de Base de données qui composent le système.

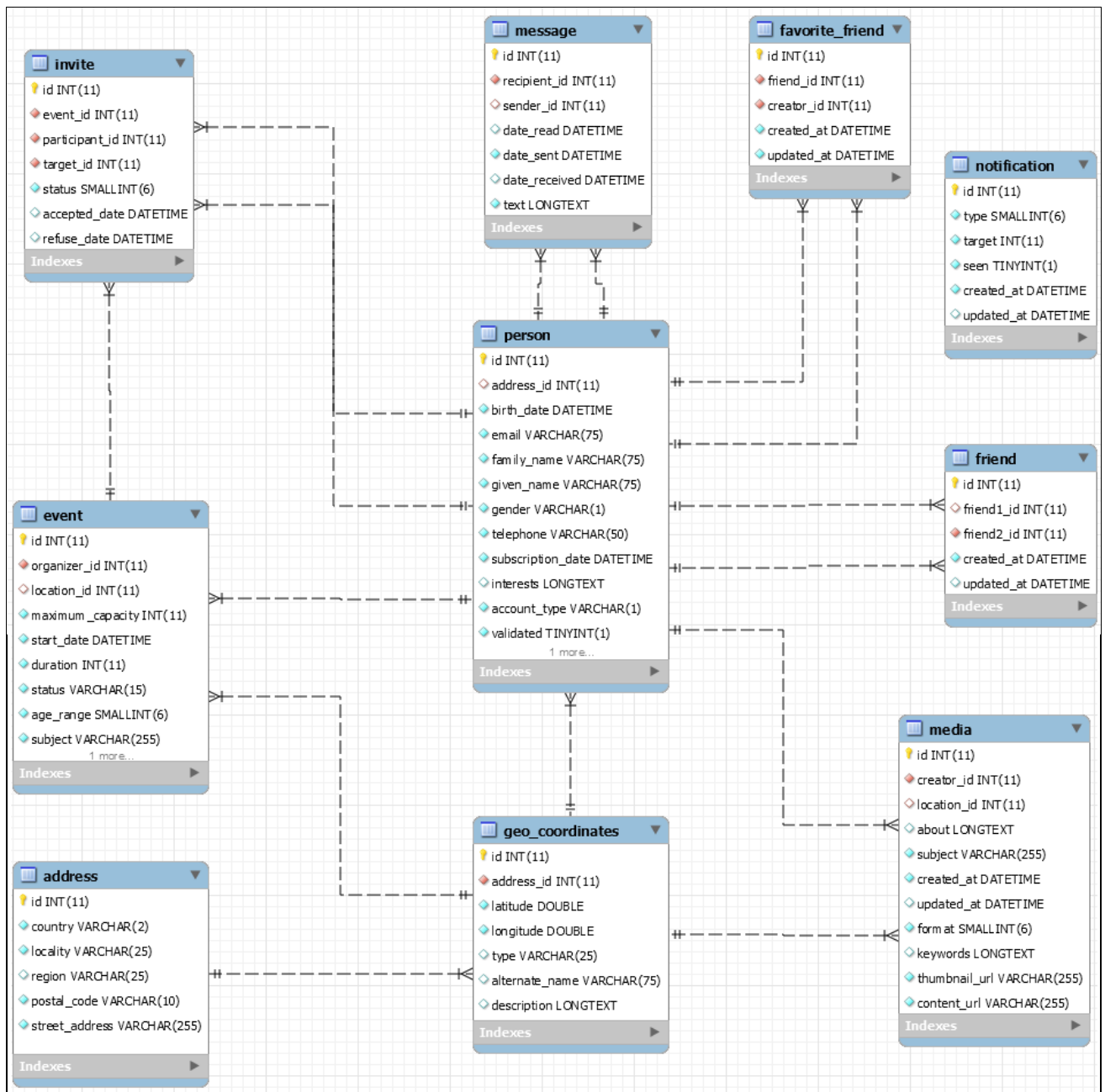


Figure 21. Diagramme de Base de données

Conclusion

Dans ce chapitre nous avons présenté l'étude conceptuelle de l'application en précisant les différents cas d'utilisations possible, ainsi que les scénarios nominaux effectués par les acteurs et en présentant le diagramme de classe général.

Cette étude est l'introduction à la partie réalisation qui sera présentée dans le chapitre suivant.

Chapitre IV : Réalisation

Introduction

Après avoir détaillé la conception de notre application, nous allons consacrer le dernier chapitre de ce rapport pour la partie réalisation. Dans ce but, nous allons présenter dans un premier temps l'environnement de notre travail, par la suite, nous allons détailler le travail réalisé.

1. Environnement de travail

1.1 Architecture Informatique sans serveur ou « Serverless »

Cette architecture permet de créer et exécuter des applications et services sans avoir besoin à gérer de l'infrastructure.

La création d'une application sans serveur nous offre plusieurs avantages, tel que :

- **L'absence de gestion de serveur** : ça nécessite pas d'allouer ou de maintenir le moindre serveur. Aucun logiciel ou moteur d'exécution à installer, à entretenir ou à administrer.
- **La flexibilité du dimensionnement**: Chaque application peut évoluer automatiquement d'où nous aurons besoin seulement d'ajuster sa capacité à travers l'activation des unités de consommation (comme le débit ou la mémoire) plutôt que les unités de serveurs individuels.
- **L'haute disponibilité** : Les applications sans serveur disposent d'une disponibilité et d'une tolérance aux pannes intégrées. Les services exécutant l'application les fournissent les capacités par défaut.
- **Aucune capacité inactive** : Aucun paiement demandé pour les capacités inactives. Il n'est pas nécessaire d'allouer une capacité à l'avance pour calcul et stockage. A titre d'exemple, aucun frais n'est facturé si le code n'est pas en cours d'exécution. [14]

C'est ce qui nous permet de diminuer la charge de gestion d'infrastructure et nous permet surtout de réduire les délais de commercialisation.



Figure 22. Exemple d'architecture d'une application sans serveur

Cette architecture utilise plusieurs composants répartis sur 4 modules principaux :

1.1.1. Hébergement Web statique :

Dans ce module, nous utilisons « Amazon S3 » qui est un service de stockage d'objet, offrant une disponibilité des données, une sécurité et des performances de pointe. Il héberge des ressources Web statiques, notamment HTML, CSS, JavaScript, fichiers image et vidéo, qui sont chargées dans le navigateur de l'utilisateur. Nous pouvons utiliser ce service afin de stocker et protéger n'importe quelle quantité de données : des sites web, des applications mobiles, la sauvegarde et la restauration, l'archivage, des applications d'entreprise, des appareils IoT et des analyses du Big Data.[16]

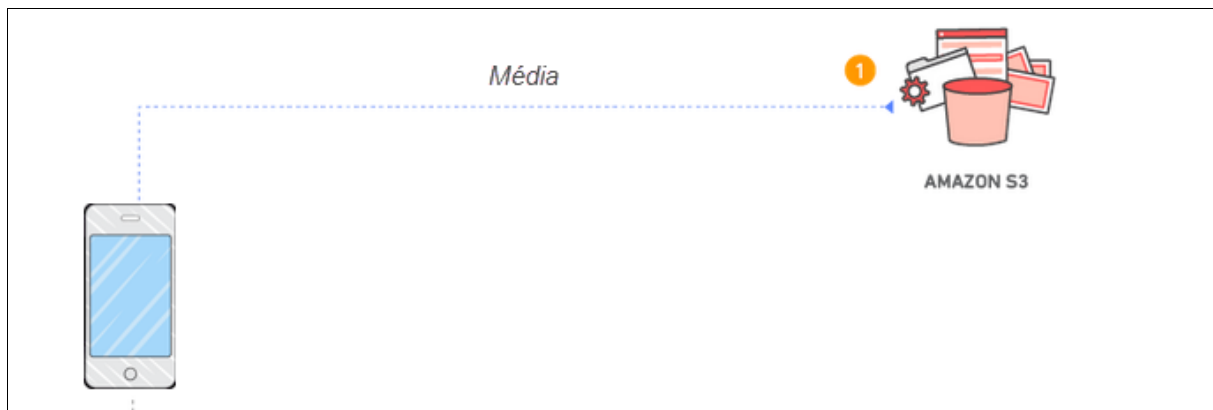


Figure 23. Amazon S3

Amazon S3 offre plusieurs avantages :

- **La performance, la disponibilité et la durabilité de pointe:** Ce service crée et stocke automatiquement des copies de tous les objets sur plusieurs systèmes, il offre une durabilité des données de 99,99%.
- **La capacités de sécurité, de conformité et d'audit sans précédent:** Ce service permet de stocker des données, et les protéger contre tout accès non autorisé avec des fonctions de chiffrement et des outils de gestion d'accès.
- **Le contrôles précis des données :** Ce service permet de classer et gérer les données et générer des rapports sur celles-ci, à l'aide de différentes fonctions tel que: l'analyse de classe de stockage S3, les stratégies de cycle de vie S3...
- **Le service de stockage dans le Cloud le plus pris en charge :** il permet de Stocker et protéger les données dans Amazon S3 en collaborant avec AWS.

1.1.2. Gestion des utilisateurs

Dans ce module, nous utilisons « Amazon Cognito » qui assure les fonctions de gestion et d'authentification des utilisateurs afin de sécuriser l'API backend.



Figure 24. Amazon Cognito

Amazon Cognito permet d'ajouter facilement et rapidement l'inscription et la connexion des utilisateurs ainsi que le contrôle d'accès aux applications Web et mobiles. Il s'adapte à des millions d'utilisateurs et prend en charge la connexion avec les fournisseurs d'identité sociale tels que Facebook, Google et Amazon. [17]

Amazon Cognito offre certains avantages dont :

- **La fédération d'identité sociale et d'entreprise:** Permettre aux utilisateurs de se connecter via des fournisseurs d'identité sociale tels que Google, Facebook et Amazon, et via les fournisseurs d'identité d'entreprise tels que Microsoft Active Directory.
- **La sécurité pour les applications et les utilisateurs:** Il prend en charge l'authentification multi-facteurs et le chiffrement des données.
- **L'intégration facile avec les applications:** Il propose une interface utilisateur intégrée et une simple configuration pour fédérer les fournisseurs d'identité. Il permet d'ajouter une inscription, une connexion et un contrôle d'accès à l'application en quelques minutes...



Figure 25. Logo Amazon Cognito

1.1.3. Backend sans serveur

Dans ce module, nous utilisons « Amazon DynamoDB », qui offre une couche de persistance où les données peuvent être stockées par la fonction Lambda de l'API.



Figure 26. Amazon DynamoDB

Amazon DynamoDB est une base de données qui offre des performances de latence de l'ordre de quelques millisecondes. Il s'agit d'une base de données multi-région et multi-maître gérée avec un système intégré de sécurité, de sauvegarde et de restauration et de mise en cache en mémoire pour les applications à l'échelle d'Internet. Cette base de donnée peut traiter plus de 10 mille milliards de demandes par jour et prend en charge des pics de 20 millions de demandes par seconde. [18]

Amazon DynamoDB offre plusieurs avantages tel que:

- **La performance:** Il prend en charge des applications de très grande envergure dans le monde, elle fournit un temps de réponse constamment inférieurs à quelques millisecondes. Les tables globales DynamoDB existent dans plusieurs régions AWS afin de donner un accès local rapide aux données pour les applications distribuées à travers le monde.
- **Sans serveur:** DynamoDB met automatiquement les tables pour ajuster la capacité et maintenir les performances. La disponibilité et la tolérance aux pannes sont intégrées.

1.1.4. API RESTful

Dans ce module, nous utilisons « AWS Lambda » et « API Gateway », qui permettent d'envoyer et recevoir des données d'une API backend publique.

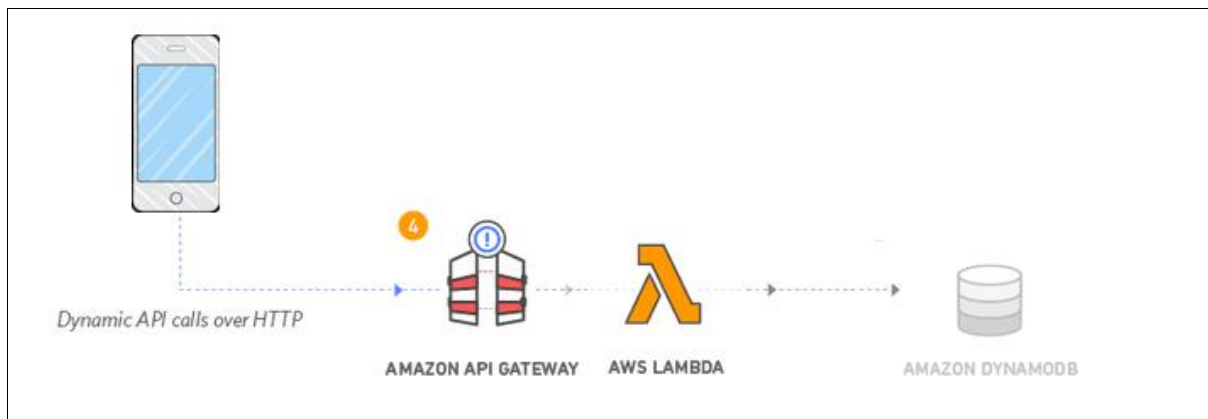


Figure 27. Amazon API Gateway & Lambda

Nous utilisons « Amazon API Gateway » pour exposer la fonction « Lambda » en tant qu'API RESTful. Cette API est accessible sur l'Internet public. Elle sera sécurisée à l'aide du groupe d'utilisateurs Amazon Cognito présenté dans le module précédent.

Amazon API Gateway est un service permettant de créer, de publier, de maintenir, de surveiller et de sécuriser les API REST et WebSocket à n'importe quelle échelle. [19]

Parmi ses avantages:

- **Le développement efficace de l'API:** Exécution de plusieurs versions de la même API simultanément avec API Gateway, ce qui permet d'itérer, de tester et de publier rapidement de nouvelles versions.
- **La surveillance Facile:** Il surveille les mesures de performance et les informations sur les appels API, la latence des données et les taux d'erreur à partir du tableau de bord API Gateway, qui permet de surveiller visuellement les appels vers nos services.
- **La réduction des coûts:** API Gateway fournit un modèle de tarification progressive pour les requêtes API. Avec un prix aussi bas que 1,51 USD par million de requêtes API au niveau le plus élevé.
- **Le contrôle de sécurité flexible:** Il autorise l'accès aux APIs avec AWS Identity and Access Management et Amazon Cognito.
- **L'API sans serveur:** Il crée des API REST à l'aide de l'API Gateway que notre application mobile peut utiliser pour appeler les services AWS disponibles publiquement, via le code s'exécutant dans AWS Lambda.

AWS Lambda est une plate-forme informatique sans serveur, gérée par événements, fournie par Amazon dans le cadre de ses services Web. C'est un service informatique qui exécute du code en réponse à des événements et gère automatiquement les ressources informatiques requises par ce code.[20]

Cette plateforme utilise plusieurs langages, dans notre cas on va opter pour « NodeJS ».

AWS Lambda offre plusieurs avantages tel que:

- **Aucun serveur à Gérer** : AWS Lambda exécute automatiquement le code, sans aucun besoin de le mettre en service ou gérer des serveurs.
- **Le dimensionnement continue** : AWS Lambda dimensionne automatiquement l'application en exécutant le code en réponse à chaque déclencheur. Le code s'exécute en parallèle et traite chaque déclencheur indépendamment.
- **Le calcule au millième de seconde** : Les frais s'appliquent à chaque 100 ms d'exécution de code et selon le nombre de fois où il est déclenché.

1.2 Langages et outils utilisés :

1.2.1 Back-end:

➤ *NodeJS* :

Node.js est une plateforme logicielle libre et événementielle en JavaScript orientée vers les applications réseaux qui doivent être monter en charge. Cette plateforme utilise la machine virtuelle V8 et implémente sous licence MIT, les spécifications CommonJS.

Parmi les modules natifs de Node.js, nous retrouvons http qui permet le développement de serveur HTTP. Node.js est un environnement bas niveau permettant l'exécution de JavaScript côté serveur. Il est utilisé comme plateforme de serveur Web, qui est utilisée par SAP4, LinkedIn , Microsoft , Yahoo!, Rakuten, Sage et PayPal... [21]



Figure 28. Logo NodeJS

1.2.2 Front-end:

➤ *Protocole HTTP 2 :*

HTTP/2 est une évolution du protocole http de la couche application le plus utilisé au monde. Cette évolution permet une utilisation plus efficace des ressources réseau. Ce protocole ne modifie pas les aspects fondamentaux de http et il garde toujours les requêtes, les réponses et les en-têtes. Ces principaux axes amélioration sont ; la rapidité et la sécurité de navigation.

Parmi d'autres améliorations nous citons:

- La compression des headers des requêtes et des réponses, qui permet de réduire la bande passante lorsque les headers et les cookies sont similaires.
- Le multiplexage des requêtes au serveur, qui consiste à faire passer de multiples informations via un seul canal de transmission.
- Le push des ressources du serveur au navigateur permettra désormais à celui-ci d'envoyer l'ensemble des ressources référencées dans une même page, avant que le navigateur n'ait analysé cette page.



Figure 29. Logo http/2

➤ **JSON :**

JavaScript Object Notation est un format léger d'échange de données. Il est basé sur un sous-ensemble du langage de programmation JavaScript. JSON est un format texte indépendant de tous les langages, mais il utilise des conventions qui sont familières à tout programmeur habitué aux langages: C, C++, C#, Java, JavaScript, Perl, Python et d'autres. Ces propriétés font de JSON un langage idéal d'échange de données.

➤ **ReactNative :**

React Native est un framework d'applications mobiles open source créé par Facebook. Il est utilisé dans le développement des applications mobiles Android, iOS et UWP. Il permet aux développeurs d'utiliser React avec les fonctionnalités natives de ces plateformes. Et permet de composer une interface utilisateur mobile riche à l'aide de composants déclaratifs.

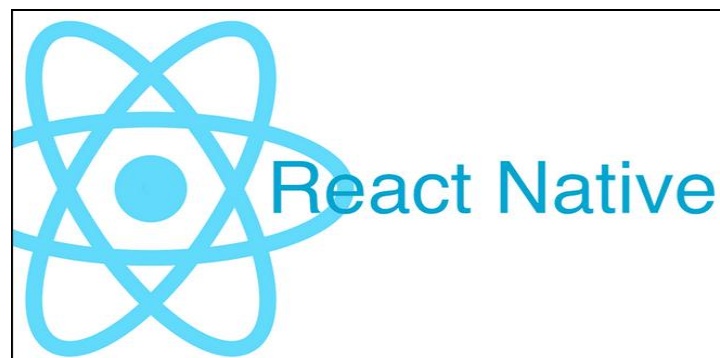


Figure 30. Logo ReactNative

2. Travail réalisé

2.1 Backlog produit et méthodologie de travail :

La méthodologie adoptée sur ce projet est Agile « Scrum », d'où la nécessité de commencer par identifier le Backlog produit de l'application et de définir les User Story de chaque Sprint. Les sprints seront de 3 semaines, chaque Sprint contiendra un nombre d'US selon la capacité de l'équipe de développement.

Les US sont déposés sur l'outil de Gestion de Projet « Trello ».

Trello : C'est un outil de gestion de projet en ligne, multilingue, lancée en septembre 2011 et inspiré de la méthodologie KANBAN de Toyota.

Il est basé sur une organisation des projets en planches listant des cartes, chacune représentant des tâches (des US). Les cartes sont assignables à des utilisateurs et sont mobiles d'une planche à l'autre, traduisant leur avancement.

La version de base est gratuite, avec la présence d'une formule payante permet d'obtenir des services supplémentaires.

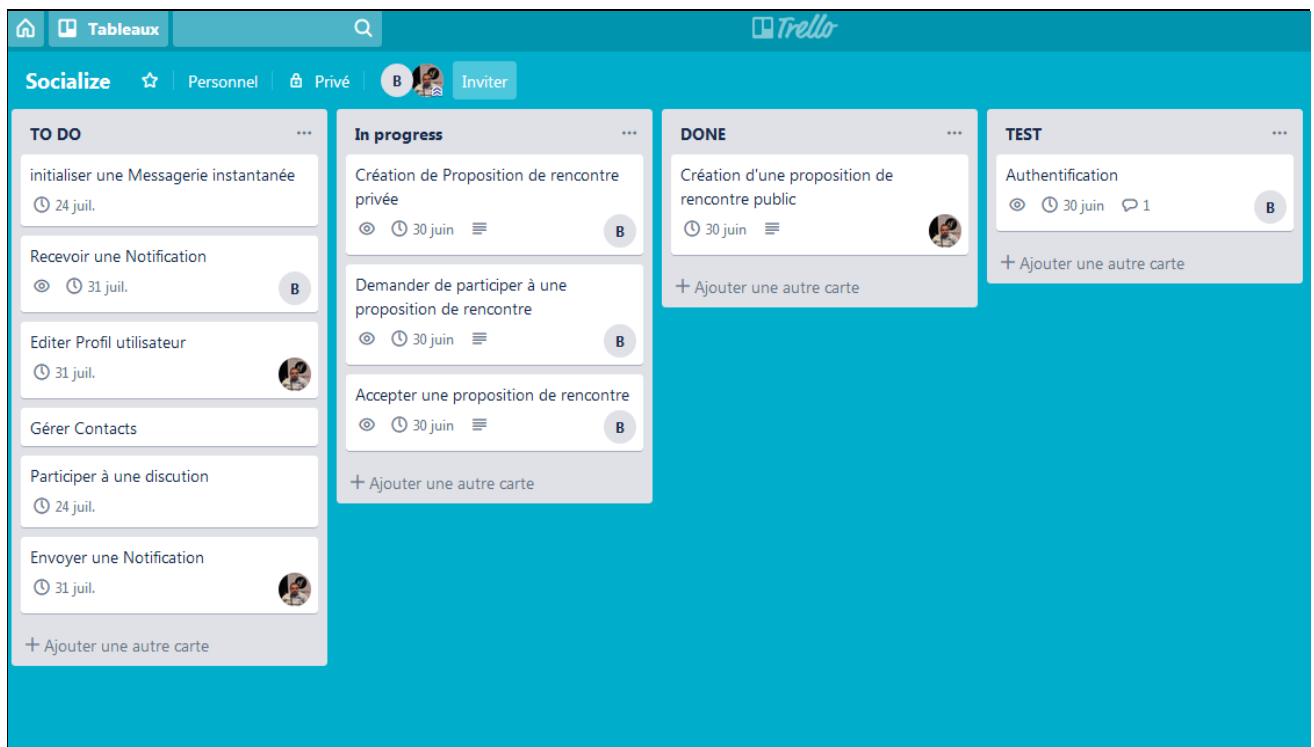


Figure 31. Backlog Produit



Figure 32. Exemple d'US d'ajout d'une proposition de rencontre

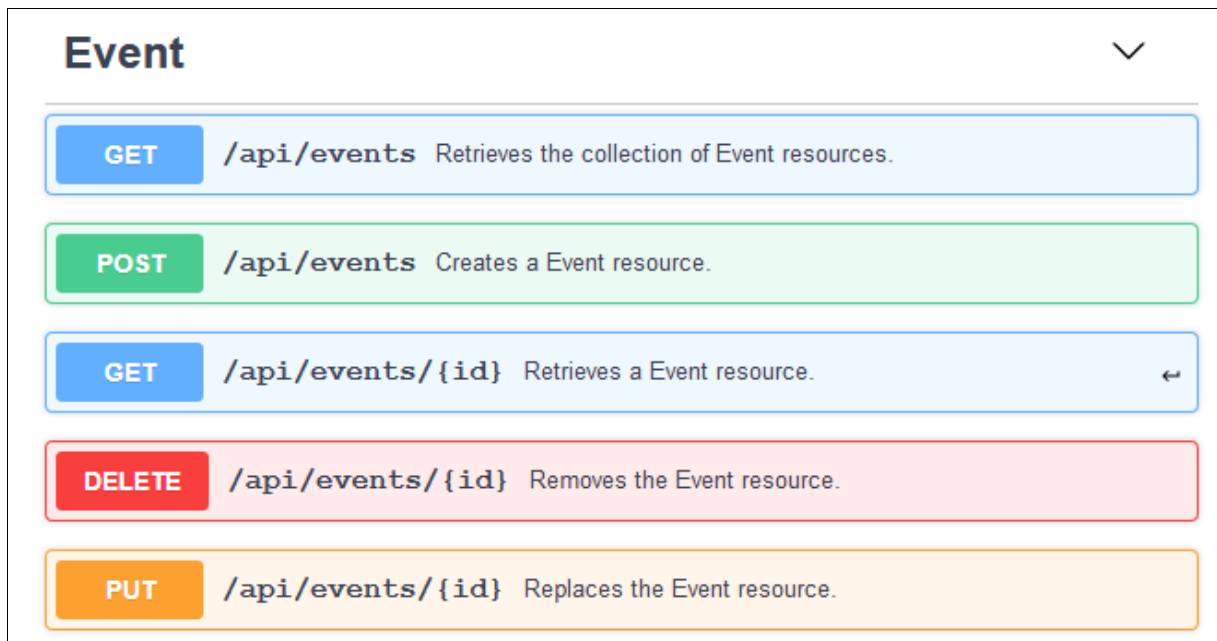
2.2 Développement Webservice

Comme déjà mentionné dans le premier chapitre, dans le travail de développement à réaliser, nous allons commencer par les Webservices.

En premier temps, nous allons commencer à préparer les API, avec le langage « Symfony » en attendant la préparation de l'architecture « Serveless » choisie par la société. Par la suite il y aura une migration de ce travail vers cette architecture.

Ci-dessous des exemples des « End Point » des WS développés, ainsi que leurs schémas relatifs enrichis par des exemples de code « JSON » de quelques requêtes invoquant ces Webservices.

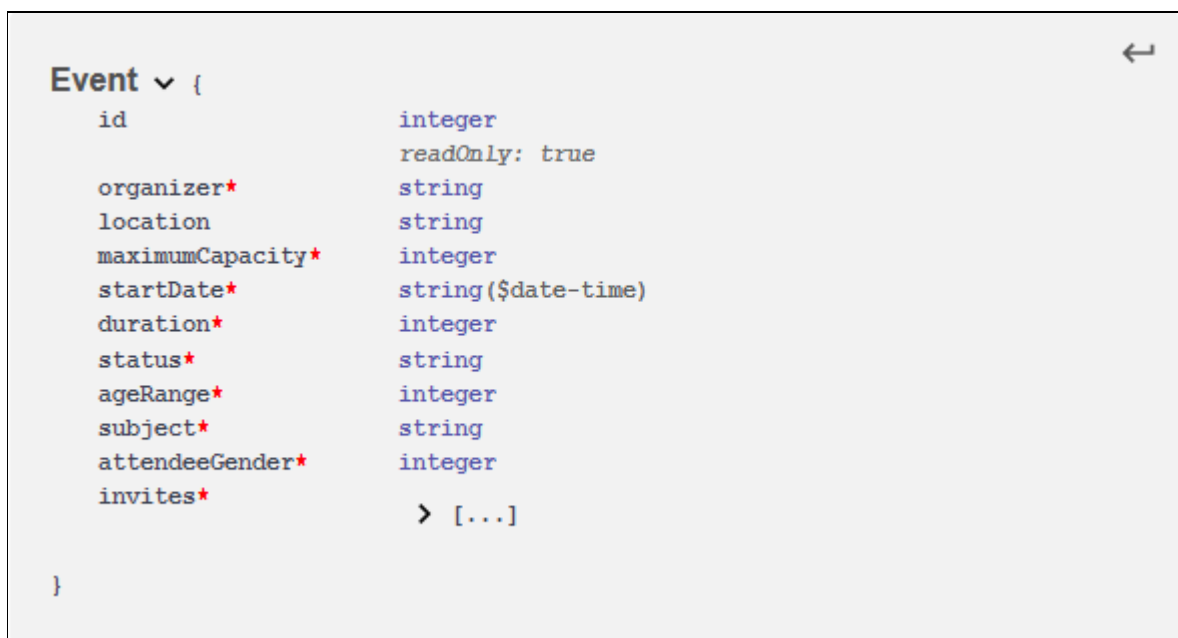
WS « *Event* » : C'est le WS qui représente la fonctionnalité de création de proposition de rencontre.



The image shows a list of API endpoints for the 'Event' resource. Each endpoint is represented by a colored box with a method name, a URL, and a description. The methods are GET, POST, GET, DELETE, and PUT. The endpoints are: /api/events (GET), /api/events (POST), /api/events/{id} (GET), /api/events/{id} (DELETE), and /api/events/{id} (PUT).

| Method | Endpoint | Description |
|--------|------------------|--|
| GET | /api/events | Retrieves the collection of Event resources. |
| POST | /api/events | Creates a Event resource. |
| GET | /api/events/{id} | Retrieves a Event resource. |
| DELETE | /api/events/{id} | Removes the Event resource. |
| PUT | /api/events/{id} | Replaces the Event resource. |

Figure 33. WS Proposition de rencontre



The image shows the JSON schema for the 'Event' resource. The schema is a JSON object with the following properties: id (integer, readOnly: true), organizer* (string), location (string), maximumCapacity* (integer), startDate* (string(\$date-time)), duration* (integer), status* (string), ageRange* (integer), subject* (string), attendeeGender* (integer), and invites* (array of objects, indicated by > [...]).

```
Event {
  id: integer (readOnly: true)
  organizer*: string
  location: string
  maximumCapacity*: integer
  startDate*: string($date-time)
  duration*: integer
  status*: string
  ageRange*: integer
  subject*: string
  attendeeGender*: integer
  invites*: > [...]
```

Figure 34. Schéma WS Proposition de rencontre

```
155 /api/events:
156 get:
157 tags:
158   - Event
159 operationId: getEventCollection
160 summary: Retrieves the collection of Event resources.
161 responses:
162   '200':
163     description: Event collection response
164     content:
165       application/ld+json:
166         schema:
167           type: array
168           items:
169             $ref: '#/components/schemas/Event'
170       application/json:
171         schema:
172           type: array
173           items:
174             $ref: '#/components/schemas/Event'
175       text/html:
176         schema:
177           type: array
178           items:
179             $ref: '#/components/schemas/Event'
180 parameters:
181   - name: page
182     in: query
183     required: false
184     description: The collection page number
185     schema:
186       type: integer
```

Figure 35. Extrait du code JSON du WS proposition de rencontre

WS « *Friend* » : C'est le WS qui représente la fonctionnalité de réseau d'amis.

Friend ▼

GET /api/friends Retrieves the collection of Friend resources.

POST /api/friends Creates a Friend resource.

GET /api/friends/{id} Retrieves a Friend resource.

DELETE /api/friends/{id} Removes the Friend resource.

PUT /api/friends/{id} Replaces the Friend resource.

Figure 36. WS réseau d'amis

| Friend ▾ { | |
|------------|---------------------------|
| id | integer readOnly: true |
| friend1 | string |
| friend2* | string |
| createdAt* | string(\$date-time) |
| updatedAt | string(\$date-time) |
| } | |

Figure 37. Schéma WS Réseau d'amis

```

453 /api/friends:
454   get:
455     tags:
456       - Friend
457     operationId: getFriendCollection
458     summary: Retrieves the collection of Friend resources.
459     responses:
460       '200':
461         description: Friend collection response
462         content:
463           application/ld+json:
464             schema:
465               type: array
466             items:
467               $ref: '#/components/schemas/Friend'
468           application/json:
469             schema:
470               type: array
471             items:
472               $ref: '#/components/schemas/Friend'
473           text/html:
474             schema:
475               type: array
476             items:
477               $ref: '#/components/schemas/Friend'
478     parameters:
479       - name: page
480         in: query
481         required: false
482         description: The collection page number
483         schema:
484           type: integer

```

Figure 38. Extrait du code JSON du WS réseau d'amis

WS « *Message* » : C'est le WS qui représente la fonctionnalité de messagerie instantanée.

| Message | |
|---------|--|
| GET | /api/messages Retrieves the collection of Message resources. |
| POST | /api/messages Creates a Message resource. |
| GET | /api/messages/{id} Retrieves a Message resource. |
| DELETE | /api/messages/{id} Removes the Message resource. |
| PUT | /api/messages/{id} Replaces the Message resource. |

Figure 39. WS Messagerie instantanée

```
Message {
  id integer
  recipient* string
  sender string
  dateRead string($date-time)
  dateSent* string($date-time)
  dateReceived string($date-time)
  text* string
}
```

Figure 40. Schéma WS messagerie instantanée

```

751 ~ /api/messages:
752 ~   get:
753 ~     tags:
754 ~       - Message
755 ~     operationId: getMessageCollection
756 ~     summary: Retrieves the collection of Message resources.
757 ~     responses:
758 ~       '200':
759 ~         description: Message collection response
760 ~         content:
761 ~           application/ld+json:
762 ~             schema:
763 ~               type: array
764 ~               items:
765 ~                 $ref: '#/components/schemas/Message'
766 ~           application/json:
767 ~             schema:
768 ~               type: array
769 ~               items:
770 ~                 $ref: '#/components/schemas/Message'
771 ~           text/html:
772 ~             schema:
773 ~               type: array
774 ~               items:
775 ~                 $ref: '#/components/schemas/Message'
776 ~     parameters:
777 ~       - name: page
778 ~         in: query
779 ~         required: false
780 ~         description: The collection page number
781 ~         schema:
782 ~           type: integer

```

Figure 41.Extrait du code JSON du WS messagerie instantanée

2.3 Test et validation

Le test demandé dans cette partie est le test fonctionnel, c'est un type de test informatique qui permet de vérifier si toutes les US préparés et déposés dans le Backlog, et décrites dans le document de spécifications fonctionnelles, sont respectées.

Les scénarios de test vont être rédigés par le langage « Gherkin » pour pouvoir par la suite les automatiser sur « Cucumber » lors du déroulement des tests de non régression.

Gherkin est un langage inventée par la communauté Ruby et son Framework de BDD (*behaviour driven development*) Cucumber. Il permet d'écrire les cas de test (Acceptance Test) avec la syntaxe suivante :

Étant donné que (une condition de départ)

Et (plus de précision avec d'autres conditions)

Quand (une action / un déclencheur)

Et (des précisions pour d'autres actions faites)

Ci-dessous des exemples des « Features » préparés pour l'automatisation des tests des API avec le langage Gherkin :

```

#langage:fr

Fonctionnalité: Créer une proposition de rencontre publique
  En tant qu'utilisateur
  Je veux créer une proposition de rencontre publique
  dont les tous les utilisateurs de l'application peuvent y participer

Scénario: Se localiser
  Etant donné que je suis connecter à l'application
  Quand je clique sur le bouton Créer une proposition de rencontre
  Alors l'écran de localisation s'ouvre
  Et les cafés et les restaurants à un diamètre de 1500 mètres apparaissent sur le Map

Scénario: Remplir formulaire création et valider
  Etant donné que j'ai sélectionné l'emplacement de la rencontre
  Quand je clique sur valider l'emplacement
  ET que je saisis la date et l'heure de la rencontre
  Et que je saisis le nombre de personnes participants
  Et que je choisis le centre d'interet de cette rencontre
  Et que j'ai choisis public pour le type de la proposition
  Et que je valide ma saisie
  Alors la proposition de rencontre est créée
  Et la proposition de rencontre est publié dans la page d'accueil

Scénario: horodatage <3 heures de l'horodatage actuel
  Etant donné que j'ai sélectionné l'emplacement de la rencontre
  Quand je clique sur valider l'emplacement
  ET que je saisis un horodatage < 3 heures
  Et que je saisis le nombre de personnes participants
  Et que je choisis le centre d'interet de cette rencontre
  Et que j'ai choisis public pour le type de la proposition
  Et que je valide ma saisie
  Alors la proposition de rencontre n'est pas créé
  Et un message d'erreur s'affiche "veuillez saisir un horodatage supérieur ou égal à 3 heures de l'horodatage de création".

Scénario: Emplacement de rencontre non disponible
  Etant donné que j'ai sélectionné l'emplacement de la rencontre
  Quand je clique sur valider l'emplacement
  ET que je saisis la date et l'heure de la rencontre
  Et que je saisis le nombre de personnes participants
  Et que je choisis le centre d'intérêt de cette rencontre
  Et que j'ai choisis public pour le type de la proposition
  Et que l'emplacement n'est pas disponible le jours de la rencontre
  Et que je valide ma saisie
  Alors la proposition de rencontre n'est pas créé
  Et un message d'erreur s'affiche "l'emplacement que vous avez choisie n'est pas disponible, veuillez saisir un autre emplacement"
  Et l'écran de map s'affiche de nouveau

Scénario: Thème de la rencontre non autorisé par rapport à l'emplacement
  Etant donné que j'ai sélectionné un café pour la rencontre
  Quand je clique sur valider l'emplacement
  ET que je saisis la date et l'heure de la rencontre
  Et que je saisis le nombre de personnes participants
  Et que je choisis le centre d'intérêt qui nécessite un repas
  Et que j'ai choisis public pour le type de la proposition
  Et que l'emplacement nest pas disponible le jours de la rencontre
  Et que je valide ma saisie
  Alors la proposition de rencontre n'est pas créé
  Et un message d'erreur s'affiche "le centre d'intérêt que vous avez choisie n'est pas compatible avec l'emplacement"

```

Figure 42.feature_création de proposition de rencontre privée

Conclusion

Dans ce chapitre nous avons détailler l'aspect pratique de notre projet, en présentant les étapes de la mise en place de l'application, l'architecture souhaité et notre participation au developpment des WS ainsi que notre implication dans la phase de test et validation des WS développés.

Conclusion générale

Le domaine de développement mobile est un domaine très important, très varié et en évolution continue. Les solutions qui représentent des réseaux sociaux ont prouvé qu'elles avaient leur place dans la société et qui sont les plus utilisées parmi les applications mobile proposées dans le marché.

Dans ce travail de projet de fin d'études, nous avons présenté les différentes étapes de réalisation de l'application mobile Socialize .

Nous avons également abordé l'importance des contacts humains directs, et nous avons mis le focus sur les problèmes de santé pouvant impacter les internautes suite à l'addiction de quelques réseaux sociaux virtuels.

Pour la planification de ce projet, nous avons adopté la méthode agile "Scrum" qui propose beaucoup de réactivité pour s'adapter aux besoins et aux changements tout au long du cycle de développement. Par rapport à la modélisation, nous avons utilisé le langage UML , ce qui nous a permis de mener correctement la partie d'analyse des besoins grâce aux diagrammes de cas d'utilisations et aussi la partie conception qui contient les diagrammes de séquence, de classe et le diagramme de Base De Données.

Le choix de l'architecture Serverless, ainsi que les langages de programmations utilisés, nous ont permis de commencer à réaliser une application complète avec des nouvelles technologies et de faire la liaison entre nos connaissances académiques et l'intégration dans le monde professionnel. Ce travail nous a permis de développer nos compétences théoriques ainsi que les mettre en pratique en programmation.

A la fin de ce stage, nous pouvons envisager les perspectives suivantes :

- Multilinguisme: Pouvoir utiliser l'application par tout dans le monde.
- Multiplateforme : Prévoir une version IOS.
- Paiement: Ajouter des fonctionnalités payantes.

Bibliographie

- [1] <https://www.drujokweb.fr/blog/reseaux-sociaux/>
- [2] <https://fr.wikipedia.org/wiki/Facebook>
- [3] <https://fr.wikipedia.org/wiki/Twitter>
- [4] <https://www.zdnet.fr/actualites/10-ans-de-linkedin-le-poids-des-chiffres-39790146.htm>
- [5] <https://fr.wikipedia.org/wiki/Viadeo>
- [6] https://fr.wikipedia.org/wiki/Copains_d%27avant
- [7] <https://fr.wikipedia.org/wiki/Trombi.com>
- [8] <https://fr.wikipedia.org/wiki/Instagram>
- [9] <https://fr.wikipedia.org/wiki/Snapchat>
- [10] <https://fr.wikipedia.org/wiki/Dailymotion>
- [11] <https://fr.wikipedia.org/wiki/YouTube>
- [12] https://www.huffpostmaghreb.com/2017/11/30/reseaux-sociaux-effets-in_n_18687414.html
- [13] https://www.rtf.be/tendance/bien-etre/psycho/detail_une-nouvelle-etude-prouve-l-effet- nefaste-des-reseaux-sociaux-sur-notre-bien-etre?id=10068730
- [14] <https://aws.amazon.com/fr/serverless/build-a-web-app/>
- [15] <https://aws.amazon.com/fr/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/>
- [16] <https://aws.amazon.com/fr/s3/>
- [17] <https://aws.amazon.com/fr/cognito/>
- [18] <https://aws.amazon.com/fr/dynamodb/>
- [19] <https://aws.amazon.com/fr/api-gateway/>
- [20] <https://aws.amazon.com/fr/lambda/>
- [21] <https://fr.wikipedia.org/wiki/Node.js>
- [27] <https://fr.wikipedia.org/wiki/Trello>