



---

Institut National des Sciences Appliquées et de Technologie

UNIVERSITÉ DE CARTHAGE

## **Rapport du stage d'été**

Filière : GL

---

### **InChat**

---

Effectué par

**ELKAMEL Marouane**

Encadrant ProxymIT : **Mr Ketteni Seif-Eddine**

Effectué à



Année Universitaire : 2016/2017



---

# Remerciements

Avant tout développement sur cette expérience professionnelle, il apparaît opportun de commencer ce rapport de stage par des remerciements, à ceux qui m'ont beaucoup appris au cours de ce stage, et même à ceux qui ont eu la gentillesse de faire de ce stage un moment très profitable.

Aussi, je remercie Mr Seif-Eddine Ketteni, mon maître de stage qui m'a formé et accompagné tout au long de cette expérience professionnelle avec beaucoup de patience et de pédagogie. Enfin, je remercie l'ensemble des employés de ProxymIT pour les conseils qu'ils ont pu me prodiguer au cours de ce mois.

Enfin, je remercie et exprime toute ma reconnaissance à l'entreprise ProxymIT qui a accepté de m'accueillir.

---

# Table des Matières

Liste des Figures	iv
Résumé	v
Abstract	vi
Introduction Générale	1
<b>I Étude Théorique</b>	<b>2</b>
1 État de l'art . . . . .	2
<b>II Spécifications du système</b>	<b>3</b>
1 Spécification des besoins . . . . .	3
1.1 Identification des besoins . . . . .	3
1.1.1 Les besoins fonctionnels . . . . .	3
1.1.2 Les besoins non fonctionnels . . . . .	4
1.1.3 Diagramme des cas d'utilisation Général . . . . .	5
1.2 Contexte du projet . . . . .	6
1.2.1 Environnement matériel . . . . .	6
1.2.2 Environnement logiciel . . . . .	6
1.2.3 Environnement humain . . . . .	6
<b>III Conception</b>	<b>7</b>
1 Méthodologie suivie . . . . .	7
1.1 La méthodologie Agile . . . . .	8
1.2 L'approche Scrum . . . . .	8
2 Analyse . . . . .	9
2.1 Analyse de l'application . . . . .	10
2.1.1 Diagrammes de séquence système . . . . .	10
3 Conception . . . . .	14
3.1 Domaine du projet . . . . .	15
<b>IV Réalisation</b>	<b>16</b>
1 Outils et langages utilisés . . . . .	16

1.1	Java . . . . .	17
1.2	Enterprise Architect . . . . .	17
1.3	SQL Server Express 2016 Express . . . . .	17
2	Présentation de l'application . . . . .	17
2.1	Itérations . . . . .	17
2.1.1	Sprint 3 : Implement SignIn and LogIn . . . . .	18
2.1.2	Sprint 3 : Implémentation de la gestion des amitiés . . . . .	19
2.1.3	Sprint 4 : implémentation de socket server . . . . .	20
<b>Conclusion Générale et Perspectives</b>		<b>22</b>

---

# Liste des Figures

II.1 Diagramme des cas d'utilisation . . . . .	5
III.1 Diagramme de séquence système « créer compte » . . . . .	10
III.2 Diagrammes de séquence système « LogIn » . . . . .	11
III.3 Diagrammes de séquence système « rechercher ami » . . . . .	12
III.4 Diagrammes de séquence système « ajouter ami » . . . . .	13
III.5 Diagrammes de séquence système « envoyer message » . . . . .	14
III.6 Modele de domaine . . . . .	15
IV.1 Product backlog . . . . .	18
IV.2 SignUp . . . . .	18
IV.3 LogIn . . . . .	19
IV.4 ajout ami . . . . .	19
IV.5 request envoyé . . . . .	19
IV.6 invitation reçu . . . . .	20
IV.7 invitation accepté . . . . .	20
IV.8 lancement serveur . . . . .	20
IV.9 serveur en marche . . . . .	21

---

# Résumé

L'application InChat a été mise en œuvre dans le cadre d'un stage d'été au sein de l'entreprise ProxymIT . L'étude et la réalisation de ce projet ont été menés par moi-même et sous la supervision du mon encadrant Mr Seif-Eddine Ketteni. L'élaboration du projet a débuté le 1 juillet et s'est terminée en 30 juillet 2017.

InChat est une application Java de messagerie. Elle a été pensée pour être un outil complet, intégrant un maximum de fonctionnalités et automatisant les procédures utilisée actuellement dans le domaine du networking sociale.

Ainsi, l'étude conceptuelle de l'application InChat a porté sur les 3 axes suivants :

- la gestion du profil.
- la gestion des amitiés.
- la gestion des messages.

Le présent rapport est une synthèse des différentes étapes développées dans la conception et l'élaboration ce projet. Ce travail a été effectué en appliquant les principes de la méthodologie Agile et en s'appuyant sur le langage de modélisation UML.

---

# Abstract

The InChat application was implemented as part of a summer internship at ProxymIT. The study and realization of this project were carried out by myself and under the supervision of my supervisor Mr Seif-Eddine Ketteni. Project development began on July 1 and ended on July 30, 2017.

InChat is a Java messaging application. It was designed to be a complete tool, integrating a maximum of functionality and automating the procedures currently used in the field of social networking.

Thus, the conceptual study of the InChat application focused on the following 3 axes :

- profile management.
- management of friendships.
- management of messages.

This report is a synthesis of the different stages developed in the design and development of this project. This work was carried out by applying the principles of the Agile methodology and based on the UML modeling language.



---

# Introduction Générale

Comme son nom l'indique, InChat est un logiciel de messagerie, destiné à être introduit dans le domaine du social networking. Plus précisément, InChat est destinée à être utilisée dans la communication à travers les sockets. Le sujet de notre application a été proposé dans le cadre de la réalisation d'un stage d'été au sein de l'entreprise ProxymIT. L'idée derrière ce travail est de faciliter le networking sociale :

- la gestion d'un profil personnel.
- la gestion des amitiés.
- l'envoi et la réception des messages.

InChat se présente donc comme une application Java qui offre à tout utilisateur la possibilité de créer un compte et l'utiliser pour gérer des amitiés et les messages.

Dans l'élaboration de ce travail, l'équipe a choisi de suivre la méthodologie Agile. L'objectif étant d'accélérer le développement de l'application en générant des versions successives et en les améliorant par un processus itératif.

La répartition des tâches à chaque étape a été décidée conjointement par les membres de l'équipe à la suite de réunions de revue et planification. La réalisation de ce projet a nécessité des efforts conjugués et une implication effective des membres de l'équipe. Le présent rapport, scindé en 5 principales sections citées ci-dessous est une synthèse du travail de recherche, de spécification, de conception et d'implémentation qui a été réalisé tout au long du projet.

Le présent document est organisé comme suit :

1. Une étude théorique du projet : C'est le premier chapitre du rapport. Il contient une étude de l'art.
2. Spécifications du système : Dans cette partie nous nous sommes intéressés à l'étude des fonctionnalités essentielles, des exigences requises de notre système, ainsi qu'à l'environnement destiné à accueillir notre application.
3. Conception : C'est la phase la plus importante dans la réalisation du projet. Elle comporte deux grandes lignes :
  - l'analyse : qui modélise les fonctionnalités de l'application
  - la conception : qui est à une fois une réalisation de l'analyse et une représentation de la structure du code à générer.
4. Réalisation : Ce chapitre spécifie l'environnement technologique de l'application et présente les résultats de l'implémentation de celle-ci.
5. Conclusion : Ce chapitre résume le travail décrit dans le rapport et dresse les perspectives du projet.

---

---

# Chapitre I

---

## Étude Théorique

### Plan

1	État de l'art . . . . .	2
---	-------------------------	---

### 1 État de l'art

Dans notre collecte d'informations et dans la mise en place de notre système, nous sommes partis sur la base des solutions déjà existantes et les avons étudiées.

L'application a été principalement inspiré du Facebook le leader dans le monde de networking sociale.

On a adopté le concept des invitations d'amitiés pour pouvoir communiquer entre deux utilisateurs avec le droit d'accepter ou refuser l'invitation.

Aussi on a créer deux modes de communication : un mode connecté et un autre mode déconnecté.

### Conclusion

Tout au long de ce chapitre, nous avons pu donner une présentation claire du cadre de travail. En outre, nous avons projeté l'objectif du système à développer. Pour cela, nous avons commencé par les solutions existantes. Puis, nous avons pu introduire le projet que l'on vise à développer. Par ailleurs, nous allons nous intéresser, dans le chapitre suivant, aux besoins de l'application pour répondre aux attentes du système.

---

---

# Chapitre II

---

## Spécifications du système

### Plan

<b>1</b>	<b>Spécification des besoins . . . . .</b>	<b>3</b>
1.1	Identification des besoins . . . . .	3
1.1.1	Les besoins fonctionnels . . . . .	3
1.1.2	Les besoins non fonctionnels . . . . .	4
1.1.3	Diagramme des cas d'utilisation Général . . . . .	5
1.2	Contexte du projet . . . . .	6
1.2.1	Environnement matériel . . . . .	6
1.2.2	Environnement logiciel . . . . .	6
1.2.3	Environnement humain . . . . .	6

### Introduction

Dans ce chapitre nous allons présenter le cahier des charges du projet. En premier lieu, nous préciserons les besoins fonctionnels et non fonctionnels. Ensuite, nous procéderons à la présentation du diagramme de cas d'utilisation général. Enfin, nous allons aborder le contexte matériel, logiciel et humain du projet.

## 1 Spécification des besoins

### 1.1 Identification des besoins

#### 1.1.1 Les besoins fonctionnels

Notre application doit comporter les volets suivants :

— **Profil :**

- Sauvegarder toutes les informations-utilisateur dans un seul endroit . Ces information peuvent être : Nom, prénom, pseudo, âge, adresse...
- Joindre la photo de profil du l'utilisateur (importer des images)

- Effectuer des mise a jour de profil.
- **Amitié :**
  - Trouver un utilisateur pour demander l'ajout.
  - Envoyer une invitation.
  - Gérer une invitation (accepter ou refuser).
  - Supprimer un ami...
- **Messages :**
  - Pouvoir créer des conversations.
  - Envoyer et recevoir des messages.

### 1.1.2 Les besoins non fonctionnels

#### Exigences de qualité

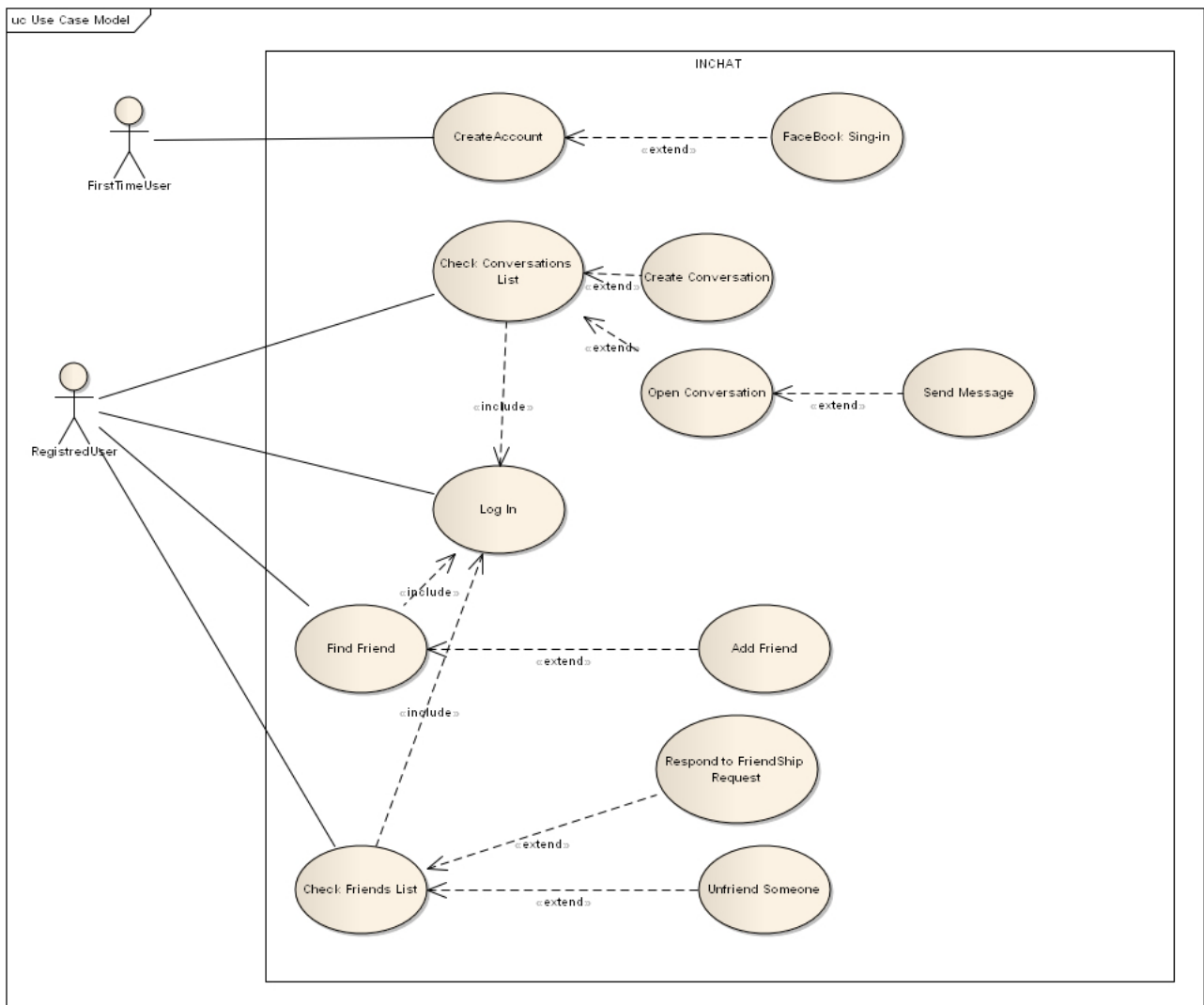
- **Disponibilité 24h/24 :** La totalité des infrastructures à déployer doivent permettre que l'application soit disponible de manière continue 24h/24 et 7j/7.
- **Convivialité :** Le futur logiciel doit être facile à utiliser. En effet, les interfaces utilisateurs doivent être conviviales c'est-à-dire simples, ergonomiques et adaptées à l'utilisateur quel que soit son appareil.
- **Portabilité**
- **Rapidité**

#### Contraintes

- **Sécurité :** Vu que les données manipulées par notre application sont critiques, nous devons garantir une sécurité optimale. Ainsi, les droits d'accès au système doivent être bien attribués, afin d'assurer la sécurité des données.
- **Fiabilité :** L'application doit être tolérante aux éventuelles pannes pouvant affecter le réseau.

### 1.1.3 Diagramme des cas d'utilisation Général

Chaque usage que l'utilisateur fait du système est représenté par un cas d'utilisation. Chaque cas d'utilisation représente une fonctionnalité qui lui est offerte afin de produire le résultat attendu. Ainsi, « le diagramme de cas d'utilisation décrit l'interaction entre le système et l'acteur en déterminant les besoins de l'utilisateur et tout ce que doit faire le système pour l'acteur ». Ci-dessous nous présentons le diagramme de cas d'utilisation général de notre système :



**Figure II.1** – Diagramme des cas d'utilisation

### 1.2 Contexte du projet

#### 1.2.1 Environnement matériel

Du point de vue matériel, la plateforme à mettre en œuvre est simple : mis à part le serveur socket les appareils électroniques (ordinateur, tablette ou même Smartphone) ayant accès à internet, aucun autre équipement n'est requis.

#### 1.2.2 Environnement logiciel

Du point de vue logiciel, notre application informatique est une application java portable. L'application utilise une base de donnée hébergé sur Amazone, et un serveur de socket qui peut fonctionner sur n'importe quelle machine java ayant accès à internet. Le fonctionnement de « InChat » est entièrement automatisé et aucune ressource humaine n'est requise pour l'administration de la plateforme.

#### 1.2.3 Environnement humain

Toute personnes peut créer un compte et utiliser l'application.

## Conclusion

Dans ce chapitre, nous nous sommes posé les questions fondamentales pour spécifier les tâches. Cette spécification va servir dans le prochain chapitre à trouver la réponse la plus convenable à ces questions et ainsi résoudre le problème en question.

---

# Chapitre III

---

## Conception

### Plan

<b>1</b>	<b>Méthodologie suivie . . . . .</b>	<b>7</b>
1.1	La méthodologie Agile . . . . .	8
1.2	L'approche Scrum . . . . .	8
<b>2</b>	<b>Analyse . . . . .</b>	<b>9</b>
2.1	Analyse de l'application . . . . .	10
2.1.1	Diagrammes de séquence système . . . . .	10
<b>3</b>	<b>Conception . . . . .</b>	<b>14</b>
3.1	Domaine du projet . . . . .	15

### Introduction

Dans un projet informatique, et plus généralement, pour tous les types de projets, l'étape de conception est cruciale. Le développement du produit final est en effet conditionné par la vision et les décisions entreprises dans la phase de conception. Ainsi, il est difficile de garantir la bonne qualité d'une application sans pour autant, avoir une bonne conception ni même négliger son importance puisqu'il s'agit là de présenter formellement une réponse au problème posé.

Dans ce chapitre, nous vous présentons notre conception vis-à-vis du problème posé. Ce dernier n'est autre que la réalisation d'une application de messagerie JAVA .

Nous allons procéder en deux temps, avec une première section détaillant l'analyse de notre application « InChat », et une seconde qui constitue la conception proprement dite de celle-ci.

## 1 Méthodologie suivie

Nous avons choisi de travailler avec la méthode SCRUM. D'après nos recherches, c'est la méthode agile la plus utilisée, la plus éprouvée, documentée et supportée parmi toutes celles existantes.

### 1.1 La méthodologie Agile

Le terme « agile » renvoie vers une approche particulière de gestion de projets informatiques. Celle-ci est définie par opposition aux approches traditionnelles prédictives et séquentielles (de type cycle en V ou en cascade).

L'approche agile vise à réduire le cycle de vie du logiciel, c'est-à-dire à accélérer son développement en développant une version minimale, puis en intégrant les fonctionnalités une à une par un processus itératif basé sur une écoute client et sur des tests tout au long du cycle de développement. La visibilité ainsi offerte du côté du client et des développeurs est clef.

### 1.2 L'approche Scrum

La vie d'un projet Scrum est rythmée par un ensemble de réunions clairement définies et strictement limitées dans le temps :

- Planification du sprint : l'équipe de développement sélectionne les éléments prioritaires du « Product Backlog »
- Revue de Sprint (à la fin du sprint) : l'équipe de développement présente les fonctionnalités terminées au cours du sprint et recueille les feedbacks du Product Owner et des utilisateurs finaux.
- Rétrospective de sprint (après la revue de sprint) : c'est l'occasion de s'améliorer (productivité, qualité, efficacité, conditions de travail, etc) à la lueur du "vécu" sur le sprint écoulé.
- Mêlées quotidiennes : il s'agit d'une réunion de synchronisation de l'équipe de développement qui se fait en 15 minutes.

L'approche Scrum propose de commencer par lister les exigences fonctionnelles et non fonctionnelles du client afin de produire le « Product Backlog ».

À partir du Product Backlog seront ensuite définis les Sprints du projet. Dans chaque sprint (ou itération), l'équipe sélectionne une portion des exigences du client à réaliser. Chaque sprint inclut des travaux de conception, de spécification fonctionnelle et technique, de développement et de test. À la fin de chacune de ces itérations, le produit partiel mais utilisable est montré au client. Ce dernier peut alors se rendre compte par lui même très tôt du travail réalisé, de l'alignement sur le besoin avant de conclure sur cette itération et de passer à la suivante.



## 2 Analyse

### Introduction

L'analyse a pour but de se doter d'une vision claire et rigoureuse du problème posé et du système à réaliser en déterminant ses éléments et leurs interactions.

Dans ce qui suit, nous allons commencer par une analyse du domaine (diag package) puis une analyse de l'application où nous allons modéliser les fonctionnalités les plus importantes en présentant pour chaque cas d'utilisation le diagramme des classes participantes correspondant.

## 2.1 Analyse de l'application

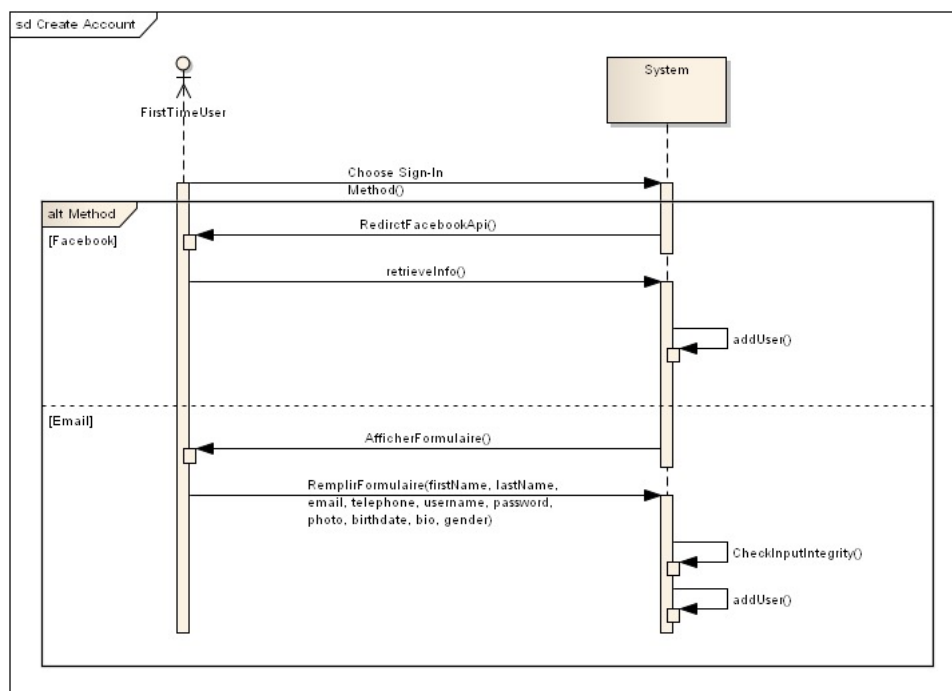
### 2.1.1 Diagrammes de séquence système

Nous allons dans cette étape isoler les messages propres aux cas d'utilisation et les représenter graphiquement sur des diagrammes de séquence UML.

Pour chaque message, le diagramme de séquence système montre les événements système déclenchés par les acteurs.

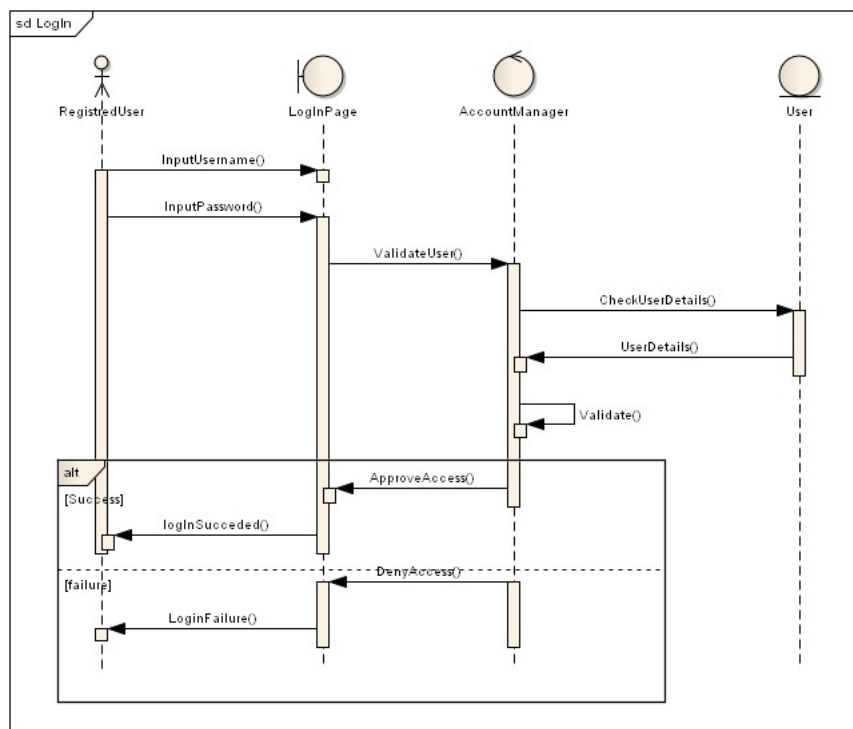
Nous allons présenter le diagramme de séquence système correspondant à chaque cas d'utilisation développée.

Ci-dessous le diagramme de séquence correspondant au l'utilisation « creation du compte » :



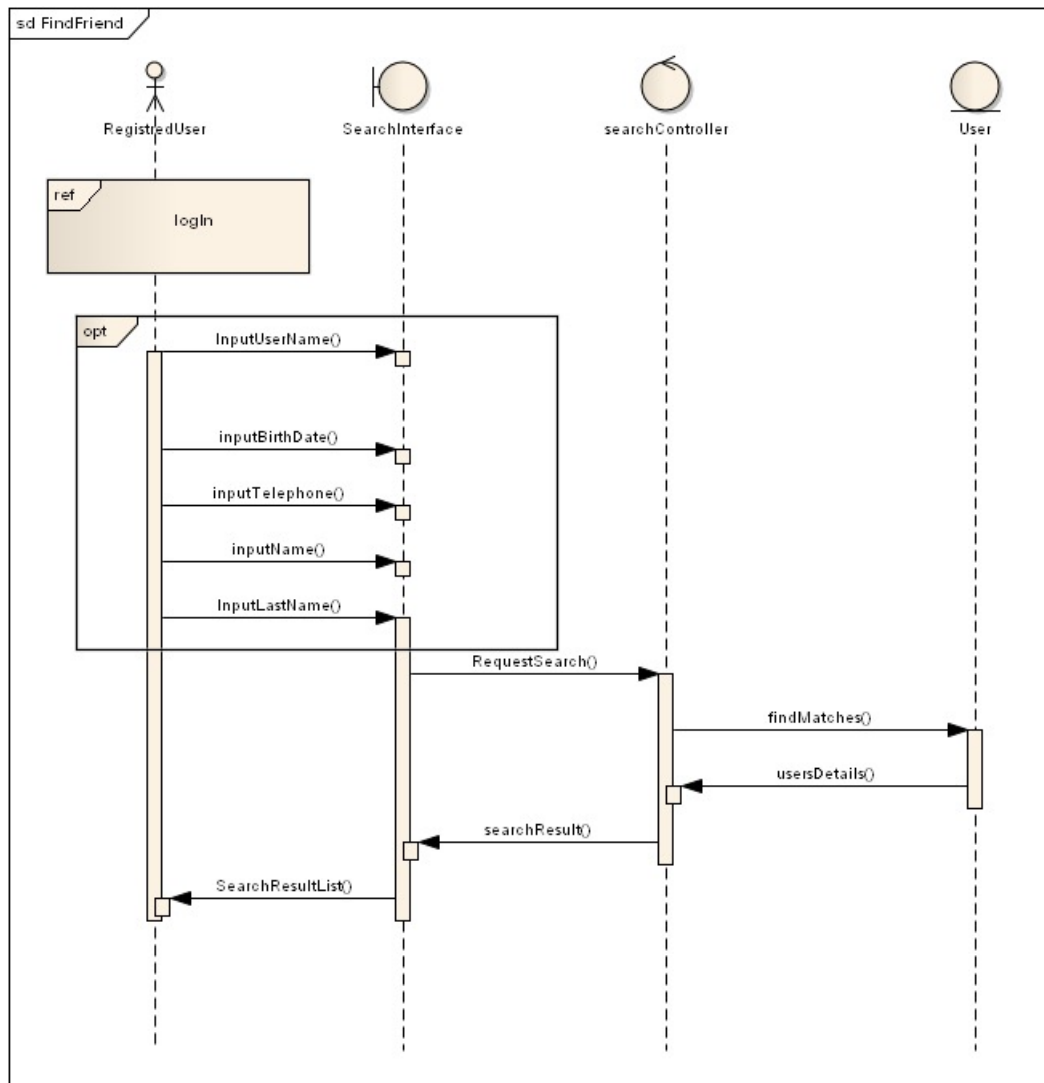
**Figure III.1** – Diagramme de séquence système « créer compte »

Ci-dessous le diagramme de séquence correspondant au d'utilisation « LogIn » :



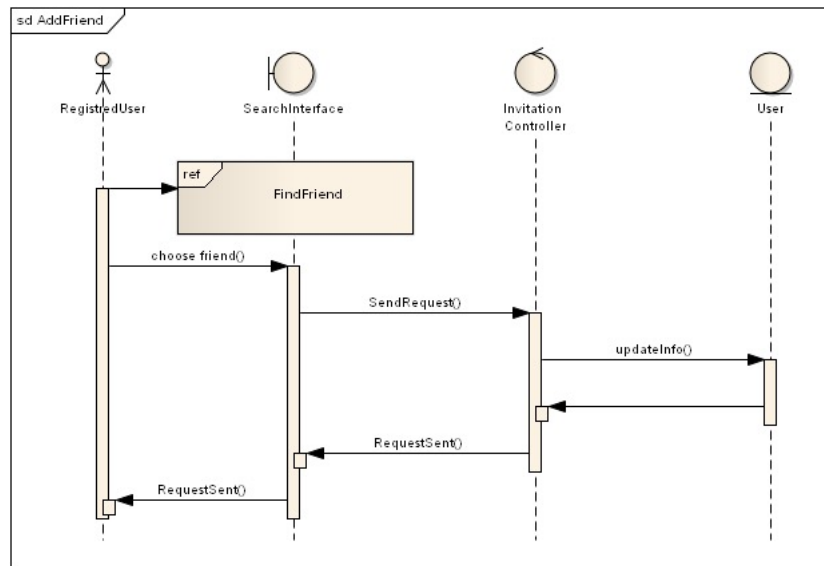
**Figure III.2** – Diagrammes de séquence système « LogIn »

Ci-dessous le diagrammes de séquence système « rechercher ami »



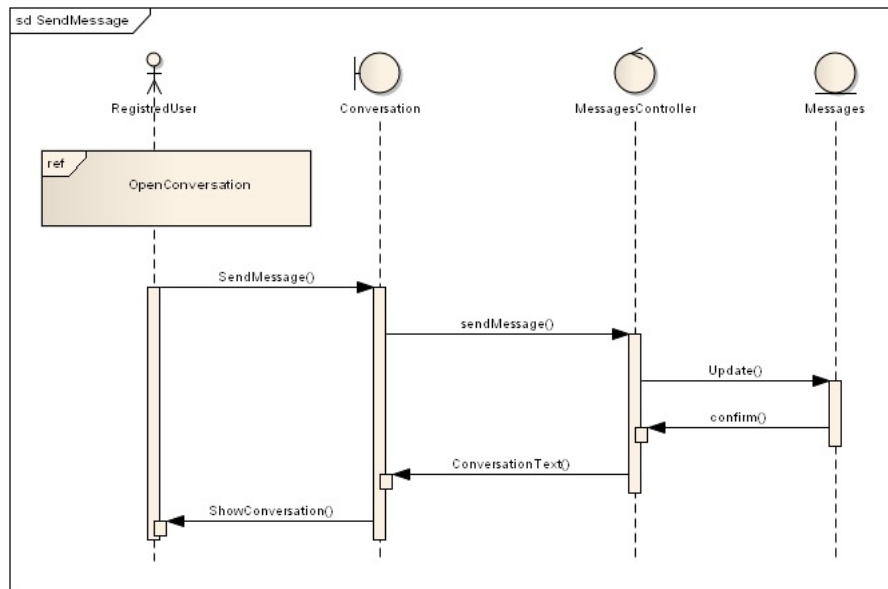
**Figure III.3** – Diagrammes de séquence système « rechercher ami »

Ci-dessous le diagrammes de séquence système « ajouter ami » :



**Figure III.4** – Diagrammes de séquence système « ajouter ami »

Ci-dessous le diagrammes de séquence système « envoyer message » :



**Figure III.5** – Diagrammes de séquence système « envoyer message »

## 3 Conception

### Introduction

Cette phase propose une réalisation de l'analyse et des cas d'utilisation en prenant en compte toutes leurs exigences. Nous allons présenter une conception des classes ainsi que de l'architecture de notre application

### 3.1 Domaine du projet

L'élaboration du modèle des classes du domaine permet d'opérer une transition vers une véritable modélisation objet. L'analyse du domaine est une étape totalement dissociée de l'analyse des besoins. Elle peut être menée avant, en parallèle ou après cette dernière.

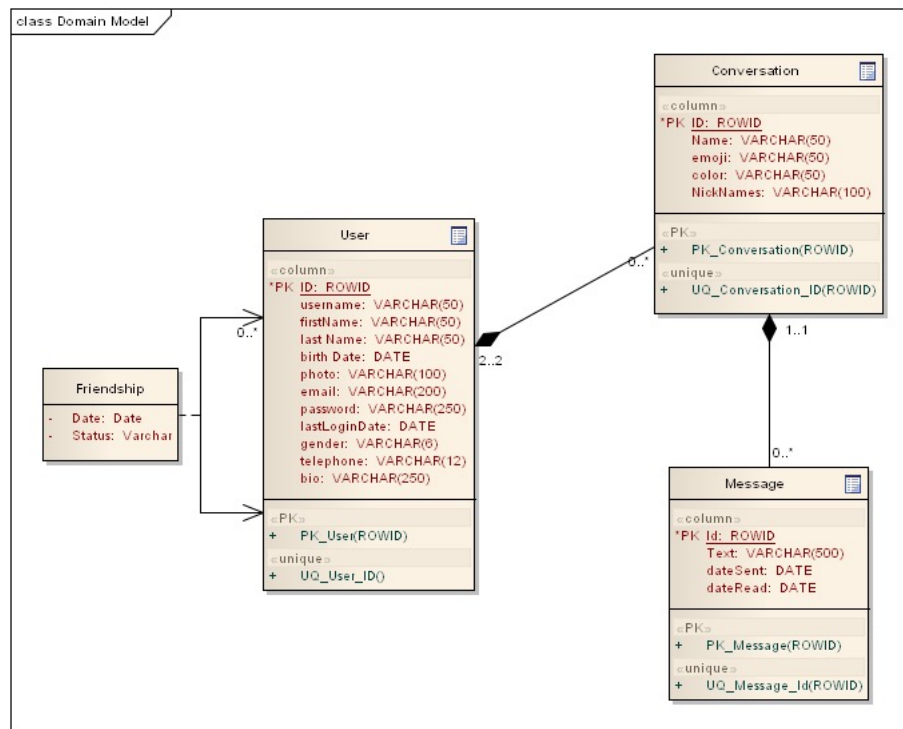


Figure III.6 – Modele de domaine

## Conclusion

Ce chapitre nous a permis de présenter la démarche que l'on a suivie pour résoudre le problème en question, que ce soit par la description de la hiérarchie du site ou même par le diagramme des classes présentant le squelette des classes du modèle MVC. Mais, il reste tout de même la partie concrète ; la réalisation qui sera le sujet du chapitre suivant.

---

# Chapitre IV

---

## Réalisation

### Plan

<b>1</b>	<b>Outils et langages utilisés . . . . .</b>	<b>16</b>
1.1	Java . . . . .	17
1.2	Enterprise Architect . . . . .	17
1.3	SQL Server Express 2016 Express . . . . .	17
<b>2</b>	<b>Présentation de l'application . . . . .</b>	<b>17</b>
2.1	Itérations . . . . .	17
2.1.1	Sprint 3 : Implement SignIn and LogIn . . . . .	18
2.1.2	Sprint 3 : Implémentation de la gestion des amitiés . . . . .	19
2.1.3	Sprint 4 : implémentation de socket server . . . . .	20

### Introduction

La réalisation vient couronner le travail de l'étude préalable et de l'étude conceptuelle. Elle présente la dernière étape et elle est très importante puisque grâce à elle le projet informatique va exister réellement, sa réussite est conditionnée par une multitude de choix essentiellement d'ordre technique concernant l'exécution de ce qui a été conçu et proposé comme solution afin de répondre aux besoins des utilisateurs et de remédier aux insuffisances perçues. Ce chapitre est donc consacré à la présentation de l'environnement matériel et logiciel nécessaire pour implémenter cette structure tout en veillant à garantir les spécificités de sécurité et les services de gestion énoncés au cahier des charges.

## 1 Outils et langages utilisés

Notre choix de la technologie s'est porté sur :



### 1.1 Java

VJava est un langage de programmation orienté objet développé par Sun Microsystems. Java est un environnement de programmation multi-thread indépendant de la plate-forme, conçu pour la création de programmes et d'applications destinées à être utilisées sur Internet et dans des intranets.

### 1.2 Enterprise Architect

Nous avons choisi de travailler sur le logiciel « Enterprise Architect » pour la modélisation de notre application. C'est un outil de modélisation graphique à base de diagrammes UML. Le langage UML (ou le Langage de Modélisation Unifié) étant centré sur la notion de « conception orientée objet », nous avons pu modéliser notre application d'une manière simple et claire.

### 1.3 SQL Server Express 2016 Express

C'est un système de gestion de données proposé gratuitement, qui offre un magasin de données complet et fiable pour les applications bureautiques et les sites Web légers.

## 2 Présentation de l'application

### 2.1 Itérations

#### Introduction

Dans le développement de notre application « InChat », nous avons commencé par la sélection des exigences les plus prioritaires à réaliser à partir du product backlog. Nous avons pu par la suite entamer le premier sprint. Nous avons ainsi pu répartir notre travail en 8 Sprints que montre la figure suivante.

Dans ce qui suit, allons détailler les sprints 3,4 et 6 obtenus :

## IV.2 Présentation de l'application

I5-8	Implement sending any update through socket	
I5-7	Implement sending message through Socket	
I5-6	Implement Socket server	
I5-5	Implement Message sending offline	
I5-4	Implement friendship Mecanism	
I5-3	Implment Sign in and logIn	
I5-2	implement CRUD USER	
I5-1	Implement DataBase	

**Figure IV.1** – Product backlog

### 2.1.1 Sprint 3 : Implement SignIn and LogIn

La principale tâche menée au cours de cette itération consiste en la mise en oeuvre de l'enregistrement dans l'application et la connexion.

Signup

# Sign UP

Username

First Name

Last Name

Email

Password

Confirm password

Gender

Male

Telephone Number

Photo

Choose File

Birthdate

Bio

Back Clear Sign Up

**Figure IV.2** – SignUp

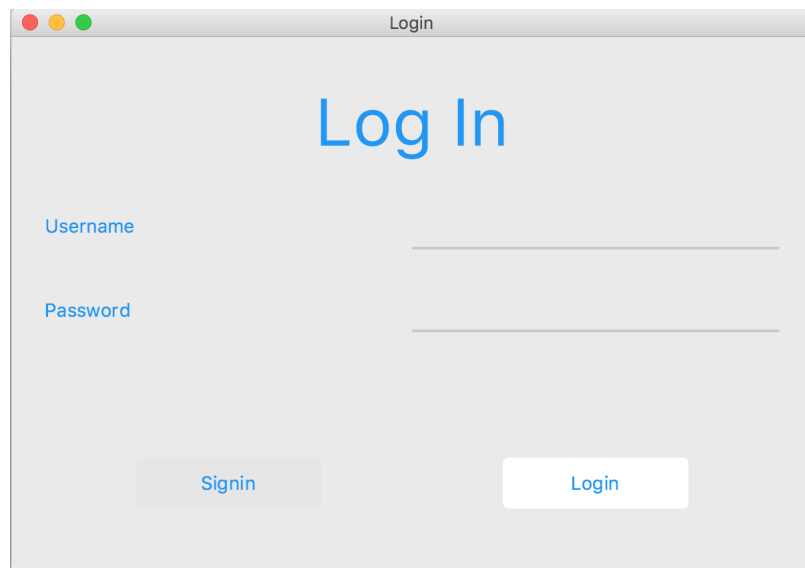


Figure IV.3 – LogIn

### 2.1.2 Sprint 3 : Implémentation de la gestion des amitiés

Ce sprint concerne le volet « gestion des amitiés » de notre application. Il implémente l'intégralité du processus de :

- Envoie de l'invitation.
- Réponse de l'invitation.

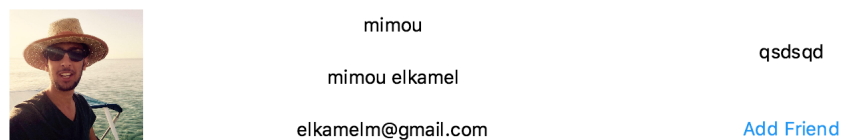


Figure IV.4 – ajout ami



Figure IV.5 – request envoyé

## IV.2 Présentation de l'application

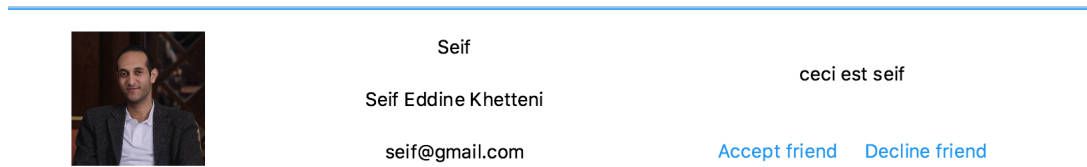


Figure IV.6 – invitation reçu

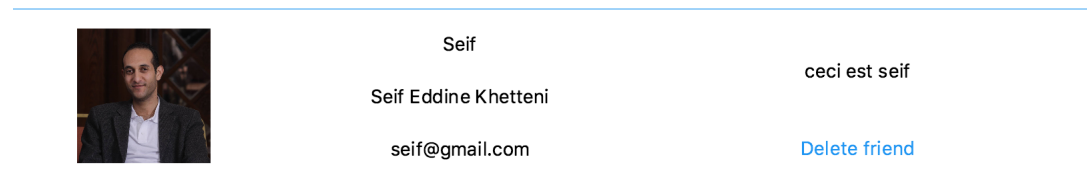


Figure IV.7 – invitation accepté

### 2.1.3 Sprint 4 : implémentation de socket server

Dans cette itération, nous avons implémenté la logique de gestion du notification instantané a travers le serveur de socket :

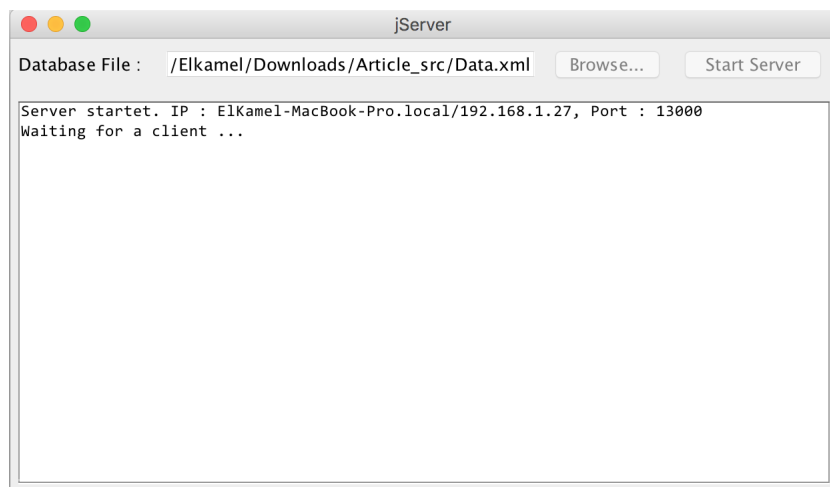
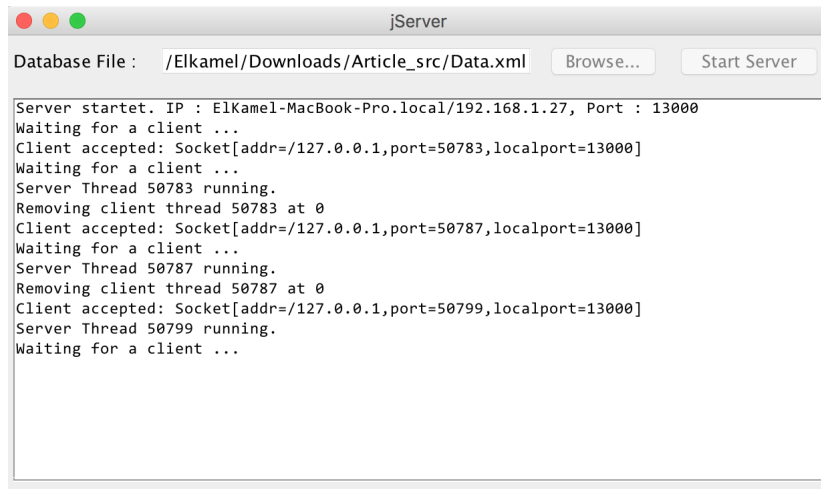


Figure IV.8 – lancement serveur

## IV.2 Présentation de l'application

---



**Figure IV.9** – serveur en marche

## Conclusion

Dans ce chapitre nous avons listé les différents sprints qu'on a suivi tout au long de notre projet en application des principes de la méthode "SCRUM".

---

# Conclusion Générale et Perspectives

Nous avons développé l'application InChat pour offrir une nouvelle plateforme de messagerie.

Dans la conception et le développement de notre application, nous avons utilisé la méthode SCRUM qui est la méthodologie agile la plus répandue et la mieux adaptée aux petits projets travaillés en un temps limité. Elle permet notamment d'impliquer l'ensemble des collaborateurs, de gagner ainsi en productivité et en temps.

Cette dernière contrainte a considérablement limité la réalisation de notre projet. Nous avons focalisé en conséquence sur les fonctionnalités essentielles et n'avons pu explorer toutes les pistes pour notre application.

Ainsi, comme perspective pour l'état actuel de notre projet, nous prévoyons :

- utilisation des emojis.
- fil d'actualité.

---