République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et la recherche scientifique

\*\*\*\*\*\*\*\*\*

Université Saad Dahleb Blida

Faculté des Sciences

Département d'informatique



Rapport de projet de fin d’études

**Sujet : Conception et implémentation d’une application web de gestion des absences.**

Présenté par :

**HAMMOU KACEM AMINE  
SAIDOUN ELYACE**

Licence en informatique

Option : ISIL

Encadré par : DR. CHERFA

2018/2019

SOMMAIRE

[RESUME 3](#_Toc12225427)

[INTRODUCTION  5](#_Toc12225428)

[CHAPITRE 1 : ETUDE D’EXISTANT 7](#_Toc12225431)

[1. Description du contexte du système existant : 7](#_Toc12225432)

[i. Etude de l’existant : 7](#_Toc12225433)

[ii. Problématique : 7](#_Toc12225434)

[2. Objectifs du réseau : 8](#_Toc12225435)

[3. Principe (description) : 8](#_Toc12225436)

[4. Méthode de développement 8](#_Toc12225437)

[CHAPITRE 2 : ANALYSE ET CONCEPTION 9](#_Toc12225438)

[i. Introduction : 9](#_Toc12225439)

[ii. Analyse des besoins : 9](#_Toc12225440)

[i. Identification des acteurs et des cas d’utilisation 12](#_Toc12225441)

[4. Mécanisme opératoire 16](#_Toc12225442)

[5. Description des donnés 23](#_Toc12225443)

[CHAPITRE 3 : IMPLEMENTATION ET REALISATION 25](#_Toc12225444)

[1. Introduction : 25](#_Toc12225445)

[2. Environnement de travail : 25](#_Toc12225446)

[3. Guide D’utilisation du réseau localement : 25](#_Toc12225447)

[4. Implementation  : 25](#_Toc12225446)

[1. L’inscription 26](#_Toc12225448)

[2. L’authentification 27](#_Toc12225449)

[3. Consulter les postes 28](#_Toc12225450)

[4. Consulter les membres 29](#_Toc12225451)

[5. Consulter les profiles 29](#_Toc12225452)

[6. Dashboard 31](#_Toc12225453)

[7. Format mobile 32](#_Toc12225454)

[*CONCLUSION* 33](#_Toc12225455)

[*BIBLIOGRAPHY* 34](#_Toc12225456)

# RESUME

Ce projet de fin d’étude a été réalisé dans le cadre du projet de fin d’études dans le but de l’obtention du diplôme de Licence en ingénierie des systèmes d’information et des logiciels.  
Le but de ce travail est de concevoir et développer un réseau social qui va offrir plus de liens et de communication entre les développeur et les informaticiens,  
et de plus être une source d’information qui va aider à faire des recherches pour bien améliorer leurs services.  
La réalisation de cette solution a été effectuée en utilisant plusieurs outils et technologies parmi lesquels on cite : Visual Studio Code, ReactJs pour le Front-End, NodeJs pour le Back-End, et MongoDB pour la base de donnée (non relationnel).  
Le format d’échange de données JSON, le langage de modélisation UML pour la conception.  
**Mots clés** : MERN-Stack, RestFull API, Mongoose, NPM.

# INTRODUCTION  La tendance actuelle du web, dans l'évolution technologique comme dans l'évolution sociale représente une avancée dans ses utilisations. Illustré par le terme "Web 2.0", les sites de réseaux sociaux, aussi appelés "social networking", sont des exemples parmi ces évolutions sociales. Ils permettent de connecter des internautes selon leurs centres d'intérêt afin de partager des flux d'informations, tels que des vidéos (YouTube), des photos (Flickr), de la musique (Last.fm) ou des marque-pages (del.icio.us), et d'agrandir leurs réseaux d'amis ou de faire partie de communautés. Ces exemples montrent l'ampleur que prend ce phénomène, aussi bien pour les loisirs, que pour des fins professionnels (LinkedIn). Nous avons donc choisi pour notre projet tutoré, de développer notre propre réseau social DEVCONNECTOR en s'appuyant sur des grands sites web comme LinkedIn, Twitter, StackOverFlow (et des forums) et d'autres qui représentent le web 2.0 par leurs technologies utilisées.

# Ainsi l'objectif est de développer un site Internet de réseau social pour les loisirs des internautes, en mettant à leurs dispositions des outils de base pour échanger des informations, des idées, et partager des centres d'intérêts. Nous ne visons pas l'exhaustivité des outils et/ou des informations partagées. Néanmoins nous suivons certaines caractéristiques des réseaux sociaux, comme l'interaction des utilisateurs à travers un groupe d'amis et de communautés.

# Pour mener à bien ce projet, nous avons tout d'abord listé et analysé les exigences requises, afin d’obtenir une expression précise des besoins. Cette analyse a permis de développer plus efficacement les différentes fonctionnalités. Ensuite, nous avons effectué des tests dans le but de comparer les exigences attendues aux résultats obtenus et d'améliorer ces derniers. Et enfin, nous avons procédé à un bilan de notre projet.

# CHAPITRE 1 : ETUDE D’EXISTANT

## Description du contexte du système existant :

### Etude de l’existant :

Le réseau regroupera en effet les fonctionnalités suivantes :  
•Gestion du profil : Il peut ajouter modifier et supprimer ces expériences et son éducation au choix.

•Gestion du Compte : il peut aussi administrer son compte (supprimer, créer)  
•Gestion des commentaires : les commentaires peuvent aussi être ajoutés et supprimés par  
l’utilisateur si c’est nécessaire.  
•Gestion des postes : Cette tâche consiste à ajouter des postes par l’utilisateur et les supprimes si c’est nécessaire.

•Aimer des postes : il peut aimer les postes ou les annuler.

### Problématique :

Parmi ces problèmes :

* Une fiche de suivi, ça se perd.
* Difficulté de gestion des absences.
* Absence de l’automatisation de l’information.
* Erreur de calcul des notes de contrôle continue en cas de perte des fiches de suivi.
* Les étudiants ne peuvent pas voir leurs absences et leurs nombres d’absences justifiées et non justifiées.

## Objectifs du réseau :

Le réseau que nous envisageons de mettre en place permettra d’atteindre les objectifs suivants :

* Automatiser les procédures de recrutement.
* Consulter des questions des domaines d’informatique.
* Faciliter le choix des différents thématiques.
* Exploiter les nouveautés dans l’industrie.

## Principe (description) :

* L’utilisateur accède à son compte sur le réseau grâce à un navigateur (Chrome, Firefox). Avec ses identifiants personnels (Email, MDP).
* L’utilisateur (développeur, designer, étudiant…) pourra sélectionner un thème (Design, web applications, search job) ;
* Une fois le thème choisi, seuls les utilisateurs concernés par ce thème peuvent partager et consulter ces postes ci.
* L’utilisateur à la possibilité de réagir au besoin demander des autres utilisateurs dans le réseau (commenter, aimer...).
* En fin, Il peut consulter les autres utilisateurs inscrit au réseau, les contactés, et voir leurs informations de profil

## Méthode de développement

Afin de réaliser ce projet, nous avons suivi la démarche en cascade (Figure 2.1).

Figure 1 : Modèle en cascade.

Dans le reste de ce rapport, nous allons détailler chaque phase du cycle de vie, en mentionnant les méthodes et les outils utilisés pour réaliser nos objectifs.

# CHAPITRE 2 : ANALYSE ET CONCEPTION

## Introduction :

Pour l’étude conceptuelle de ce projet, nous avons opté pour le Processus Unifié qui est basé sur le langage de modélisation UML (UnifiedModelingLanguage) comme processus de développement. Notre projet vise la mise en place d’une application en architecture trois tiers.

Nous avons choisi UML qui inclut plusieurs activités. En effet, dans ce chapitre nous présentons la conception de notre système de gestion des absences, car elle délimite la portée du système à élaborer, afin de mieux cerner son architecture et identifier les risques qui peuvent altérer son bon fonctionnement.

## Analyse des besoins :

Dans cette partie nous détaillons l’ensemble des fonctionnalités que l’application doit offrir aux utilisateurs.

En effet, le système à réaliser doit répondre aux besoins suivants :

##### Inscription

L’utilisateur doit s’inscrire avant de se connecter à l’application, en  
introduisant son email et mot de passe qui sont obligatoires pour pouvoir se  
connecter, ainsi que ces informations personnelles.

##### Authentification

Afin d’accéder au site et pouvoir accéder aux différentes fonctionnalités (poster, messagerie…), tout membre doit s’authentifier.

##### Postes Tout membre du site possède un journal, auquel il peut ajouter des articles (communément blog). Il peut ainsi poster des messages contenants tous types d’information (ces expériences, questions, remarque, détails, de sa journée, résumé …) que les autres membres pourront visualiser et commenter. De même, un membre a la possibilité de consulter le journal des autres et de le commenter.

##### Messagerie

L’utilisateur contacte les autres membres avec un email.

##### Profil

Un membre a des droits sur ses informations personnelles. Il peut donc modifier à tout moment ses données telles que son pseudo, son mot de passe, expériences, éducations… mais aussi son image personnelle (avatar). Il doit également pouvoir choisir si ses informations (son profil).

##### Membres L’internaute a le droit de consulter les membres du réseau.

**3. Diagramme des cas d’utilisation :**

Nous allons répondre aux questions suivantes : Quels sont les utilisateurs du système ? Quelles sont leurs interactions avec celui-ci ? Il faut donc identifier les différents acteurs ainsi que les cas d’utilisation c’est-à-dire les différentes fonctionnalités du système. Les acteurs pour le site Community sont les suivants :

• **L’internaute / l’invité :** personne qui n’est pas membre du site et qui souhaite le devenir.

• **Le webmaster** : rôle des personnes qui ont en charge la maintenance du site Web.

• **Le membre** : personne qui possède un compte, qui peut rechercher des membres, fait partie de communautés, possède des amis...

Les principaux cas d’utilisation des acteurs précédemment identifiés, ont été bien mis en évidence dans la partie précédente. Voici donc le diagramme de cas d’utilisation.

## Identification des acteurs et des cas d’utilisation

Un cas d’utilisation représente l’ensemble des actions réalisées par le système. Un acteur est un utilisateur du système qui communique et interagit avec les cas d’utilisation à travers une association**.**

1. UC du reseau globale

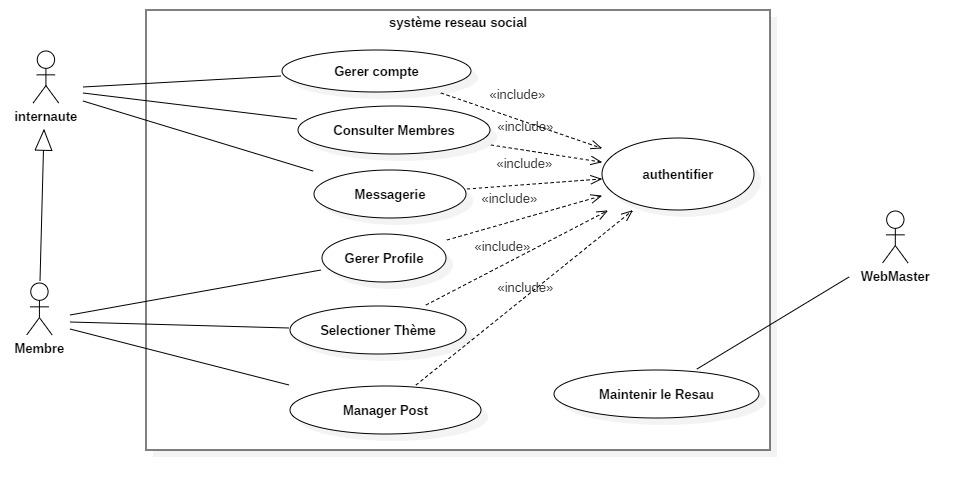


Figure 2.1.1 : digramme des cas d’utilisation globale

Les acteurs et les cas d’utilisation sont résumés dans le tableau suivant :

|  |  |
| --- | --- |
| Les acteurs et les cas d’utilisation sont résumés dans le tableau suivant : Cas  D’utilisation | Acteur |
| Gérer compte | Membre, Internaute |
| Sélectionner thème | Internaute, Membre |
| Gérer profile | Membre |
| Manager post | Membre |
| Messagerie | Membre, Internaute |
| Consulter membres | Membre, Internaute |
| Authentifier | Membre, Internaute |
| Maintenir le réseau | Webmaster |

Tableau 1.1 :les acteurs des cas d’utilisation

1. 2UC gérer compte

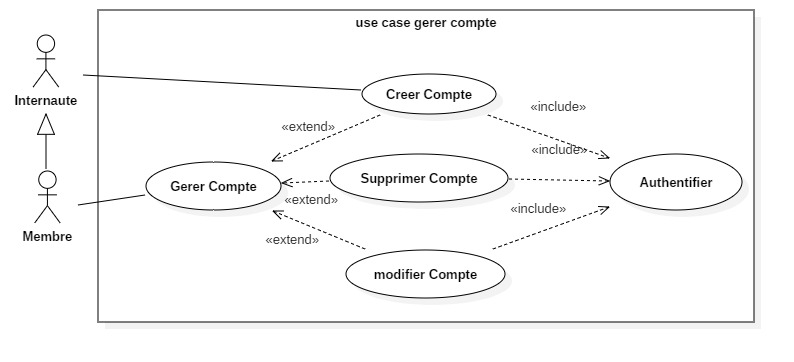
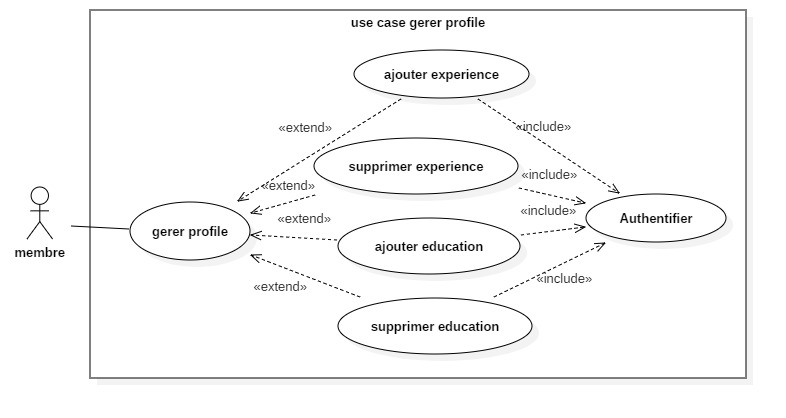
 Figure 2.1.2 : digramme des cas d’utilisation gérer Compte

Tableau 1.2 :les acteurs des cas d’utilisation de gérer compte

|  |  |
| --- | --- |
| Use case | Acteur |
| Gérer Compte | Membre |
| Créer Compte | Internaute |
| Supprimer Compte | Membre |
| Modifier Compte | Membre |
| Authentifier | Membre, Internaute |

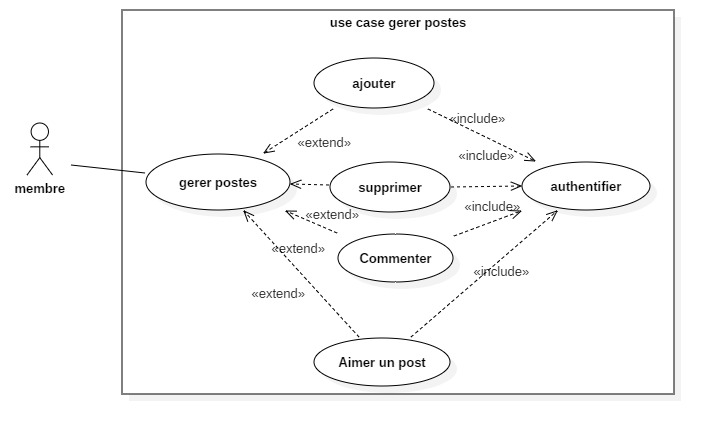
1. UC gérer profile

 **Figure 2.1.3 : digramme des cas d’utilisation gérer profile**

|  |  |
| --- | --- |
| Use case | Acteur |
| Gérer profile | Membre |
| Ajouter Education | Membre |
| Supprimer Education | Membre |
| Ajouter Expérience | Membre |
| Supprimer Expérience | Membre |
| Authentifier | Membre |

Tableau 1.3 :les acteurs des cas d’utilisation de gérer profile

1. UC gérer postes

 **Figure 2.1.4 : digramme des cas d’utilisation gérer postes**

|  |  |
| --- | --- |
| Use case | Acteur |
| Gérer postes | Membre |
| Ajouter | Membre |
| Supprimer | Membre |
| Commenter | Membre |
| Aimer un Post | Membre |
| Authentifier | Membre |

Tableau 1.4 :les acteurs des cas d’utilisation de gérer postes

## Mécanisme opératoire

La conception consiste à affiner les descriptions déjà réalisées dans l’analyse. Elle permet de construire une architecture stable du système à réaliser. Au cours de cette activité, nous allons concevoir les cas d’utilisation à l’aide des diagrammes de séquence.

#### Authentification

Afin d’accéder au site et pouvoir accéder aux différentes fonctionnalités (publier sur le mur, messagerie…), tout membre doit s’authentifier.

Tableau 2 : « S’authentifier »

|  |  |
| --- | --- |
| « S’authentifier » | |
| **Cas d’utilisation** | S’authentifier |
| **Acteur** | Utilisateur |
| **Pré condition** | L’utilisateur a un login et un mot de passe |
| **Post condition** | Acteur authentifié |
| **Description de scenario principale** | 1. L’acteur saisit son login et son mot de passe.  2. IL clique sur le bouton « Log in »  3. Le système vérifie la combinaison login et mot de passe  4. Si le login et le mot de passe sont corrects, le système affiche l’interface correspondante. |
| **Exception** | Si le login et le mot de passe sont incorrects, le système affiche un message d’erreur. |

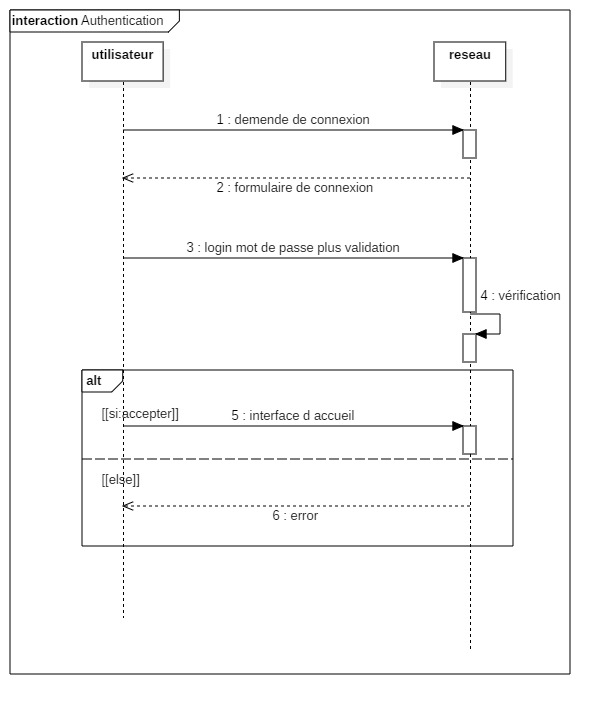


Figure 2.3 : diagramme de séquence « authentification »

#### Inscription

Chaque utilisateur doit s’inscrire avant de s’authentifier dans le réseau en cliquant sur le bouton « inscription » dans l’interface d’authentification pourvoir un accès sécurisé.

Tableau 3 : « S’inscrire »

|  |  |
| --- | --- |
| « S’inscrire » | |
| **Cas d’utilisation** | S’inscrire |
| **Acteur** | Internaute |
| **Pré condition** | L’acteur a un nom, un prénom, un mot de passe, un email, un matricule … |
| **Post condition** | Acteur inscrit |
| **Description de scenario principale** | 1. L’acteur saisit son information (nom, prénom, mot de passe, email, matricule...)  2. IL clique sur le bouton « Singe UP »  3. Le système vérifie la combinaison (nom, prénom, matricule enseignant)  4. Si les informations entrées sont correctes, le système affiche l’interface correspondante |
| **Exception** | Si les informations entrées sont incorrectes, le système affiche un message d’erreur. |

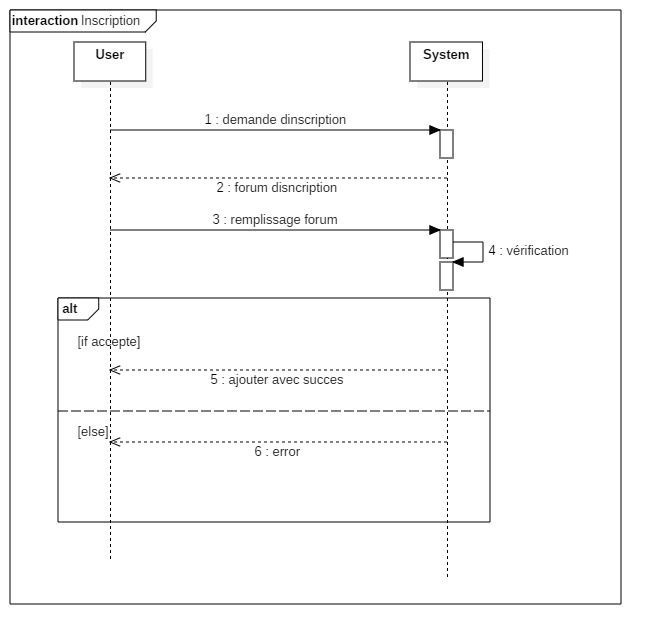


Figure 2. 4 : diagramme de séquence « S’inscrire »

#### Gérer Postes

Tout membre du site possède un journal, auquel il peut ajouter des articles (communément blog). Il peut ainsi poster des messages contenants tous types d’information (ces expériences, questions, remarque, détails, de sa journée, résumé …) que les autres membres pourront visualiser et commenter. De même, un membre a la possibilité de consulter le journal des autres et de commenter.

Tableau 4 : « Gerer poste »

|  |  |
| --- | --- |
| *« Gérer poste »* | |
| **Cas d’utilisation** | *Gérer poste* |
| **Acteur** | membre |
| **Pré condition** | Acteur authentifié |
| **Post condition** | Page des postes |
| **Description de scenario principale** | 1. L’acteur choisit le thème**.**  2. L’acteur clique sur « poster »  3. Le système affiche le chuan de poste  4. L’acteur marque squille veut dire  5. l’acteur click sur ‘Submit’ le poste |

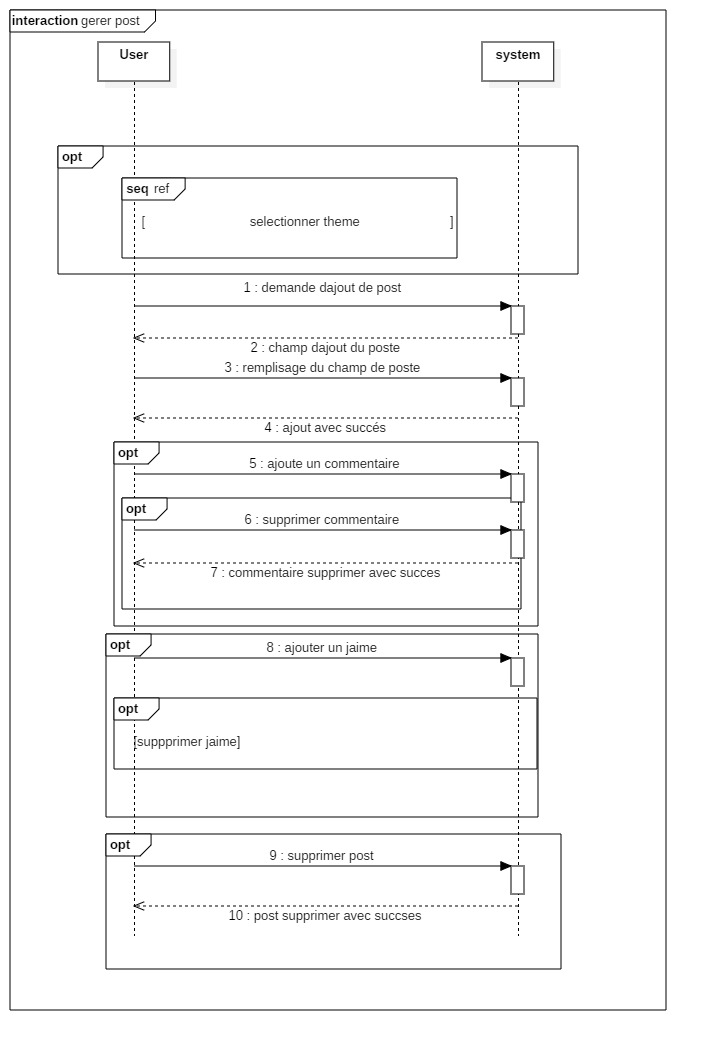


Figure2.5 : diagramme de séquence « Gérer poste »

#### Gérer Profil

1. Un membre a des droits sur ses informations personnelles. Il peut donc modifier à tout moment ses données telles que son pseudo, son mot de passe, expériences, éducations… mais aussi son image personnelle (avatar). Il doit également pouvoir choisir si ses informations (son profil).

Tableau 5 : « Gérer exclusion »

|  |  |
| --- | --- |
| *« Gérer profile »* | |
| **Cas d’utilisation** | *Gérer profile* |
| **Acteur** | membre |
| **Pré condition** | Acteur authentifié |
| **Post condition** | Profile modifier |
| **Description de scenario principale** | 1. L’acteur clique sur le bouton « éditer profile »**.**  3. Le système affiche sont profile.  2. L’acteur choisi les informations qu’il veut modifier.  4. les informations modifier avec succès. |

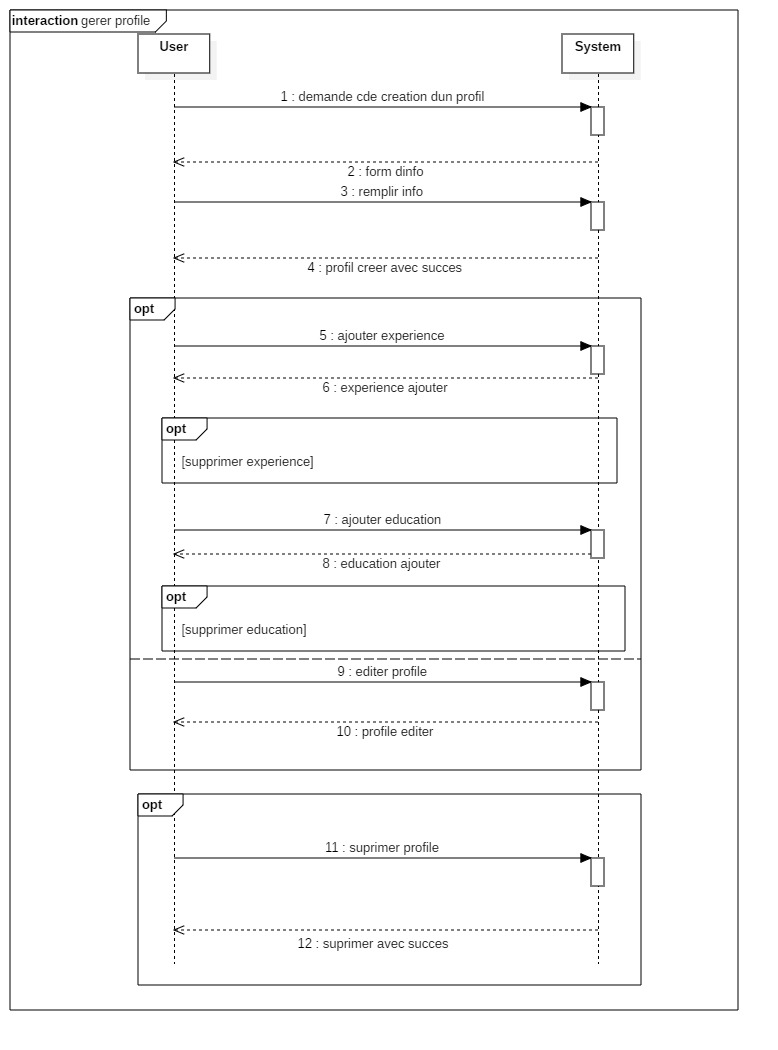


Figure 2.6 : diagramme de séquence « Gérer profile »

## 5. Description des donnés

Dans cette partie, nous étudierons les entités statiques du système. Ceci est illustré par le diagramme de classe suivant :

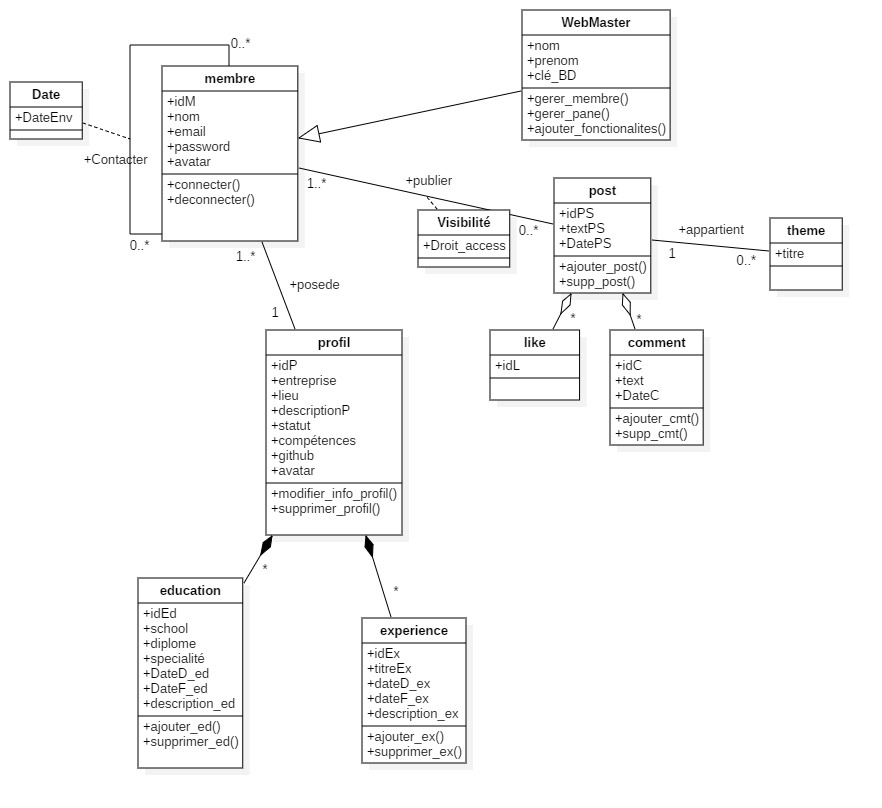


Figure2.2 : diagramme de classes

Remarques sur les différentes associations et cardinalités:

|  |  |  |
| --- | --- | --- |
| Relation | Type d’association | Multiplicité |
| publier | Association | Un membre peut publier 0 ou plusieurs postes. (\*)  Un poste est publié par 1 ou plusieurs membres.. (1 ..\*) |
| appartient | Association | Un theme appartient a 0 ou plusieurs postes. (0 ..\*)  Un post appartient a 1 seule theme. (1) |
| possede | Association | Un ou plusieurs members (1..\*) possede un et un seul profil. (1) |
| Contacter | Association | Un member peut contacter 0 ou plusieurs members (0..\*)  . |

# CHAPITRE 3 : IMPLEMENTATION ET REALISATION

## Introduction :

Dans cette phase nous allons détailles les étapes de développement de notre application web.

Le choix des outils qu’on a utilisé dans notre projet et les environnements de travails

## Environnement de travail :

Afin de pouvoir implémenter notre application web nous avons utilisé principalement les langages suivants :

|  |  |
| --- | --- |
| **Système de gestion de base de données** | * Node server. |
| **Environnement de développement** | * Visual Studio Code |
| **Langage de développement** | * Javascript. * ReactJs * NodeJs * HTML5. * CSS3. * MongoDB. * Bootstrap |

# Guide D’utilisation du réseau localement :

# Installer MongoDB et NodeJs

# changer le fichier « default.json » dans le dossier config

# Ajouter le URI de la connexion du mongodb :

"mongoURI" : "mongodb://localhost/devconnector",

# Installer les dépendances du server

npm install

# Install les dépendances du client

cd client

npm install

# Lancer le server depuis un cmd ou depuis NPM (Node package manager)

npm run dev

# Lancer pour la production

cd client

npm run build

**Visiter notre site :**[**http://devconnector.ga**](http://devconnector.ga/)

1. **Implémentation :**

Nous avons implémenté notre solution comme suit avec des exemples explicatifs :

### L’inscription

Quand l’internaute accède l’URL de l’application web il aura cette interface :



Figure 3.1 : L’interface de login sur PC

Si l’utilisateur n’a pas de compte il doit s’inscrire. Pour cela il doit cliquer sur le bouton « Register » en haut à droite ou sur « Sign Up » au milieu et l’interface suivante :

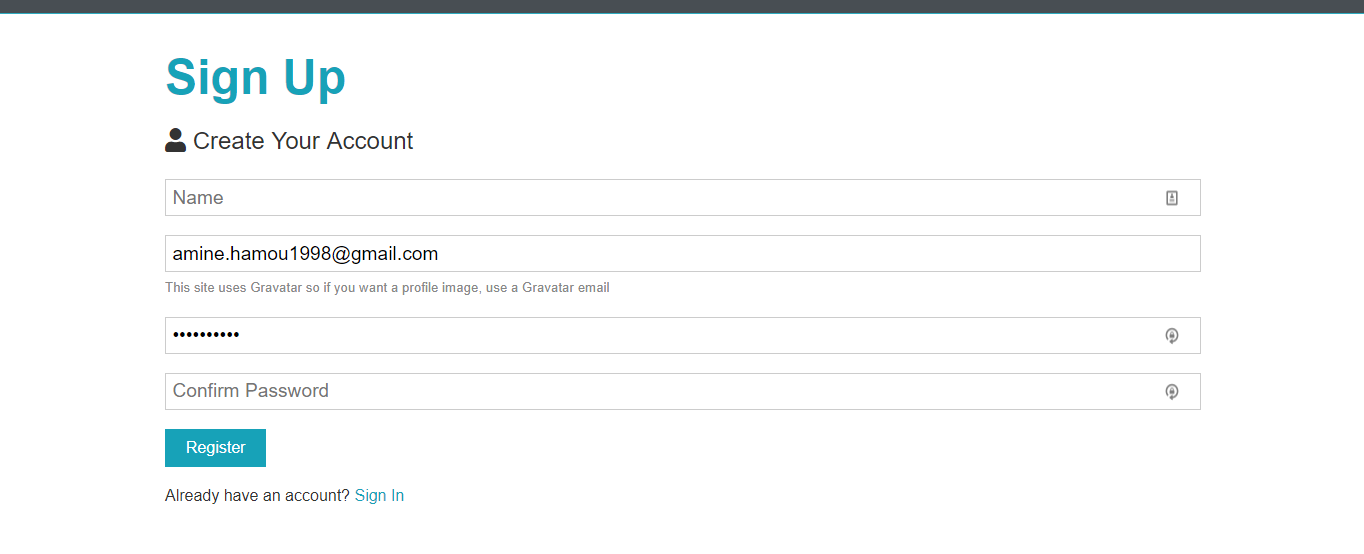


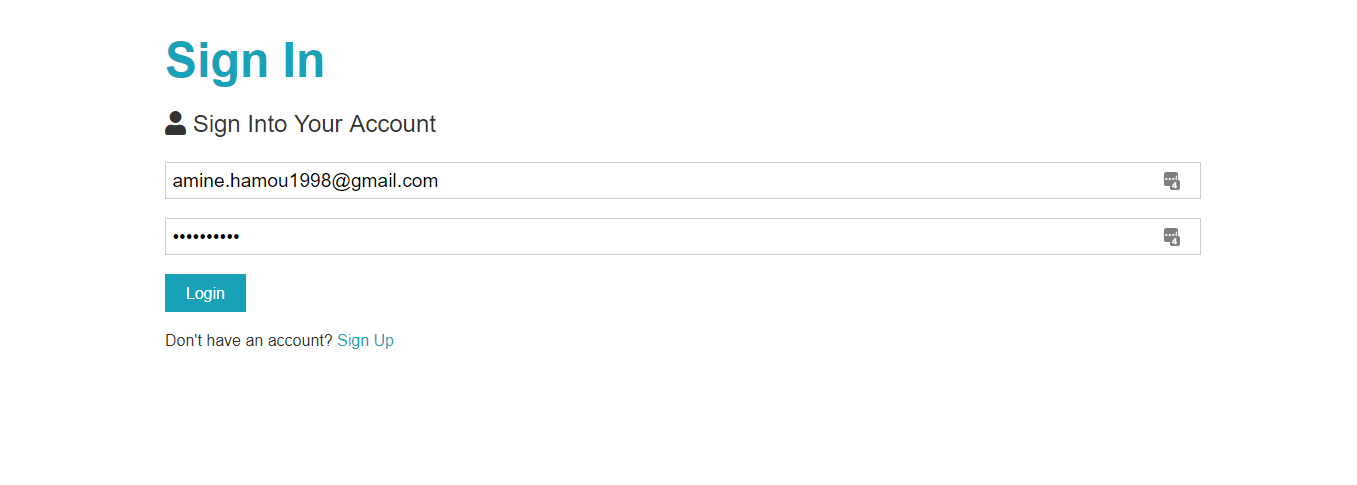
Figure 2.10: L’interface de l’inscription

Il doit alors remplir le formulaire, puis cliquer sur « Register », il est désormais inscrit sur le système.

### L’authentification

L’utilisateur saisie son email et mot de passe qui as était inscrit.

En cas d’erreur un message d’erreur sera affiché, sinon il sera redirigé vers la page principale « Posts »

 Figure 3.2: L’interface de l’authentification

### Consulter les postes

On suppose qu’un utilisateur xxx a déjà fait l’inscription est qu’il existe sur la base de données.

Après s’être correctement authentifié La page « Posts » s’affiche.

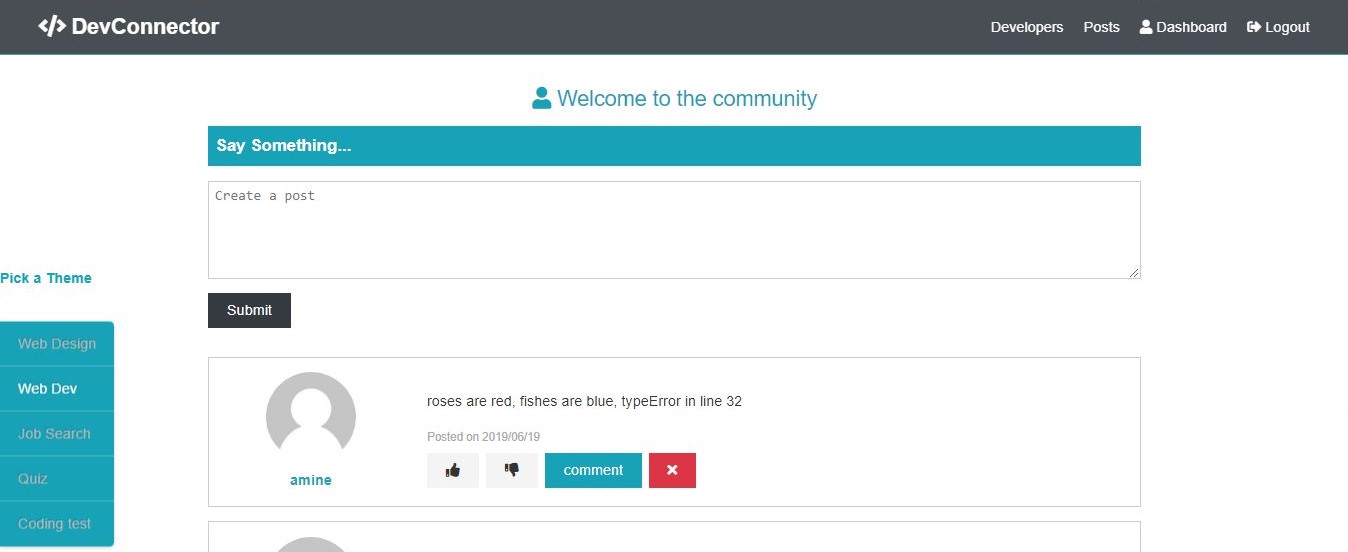


Figure 3.3: L’interface des posts

### Consulter les membres

L’utilisateur peut consulter les membres authentifier dans le réseau.

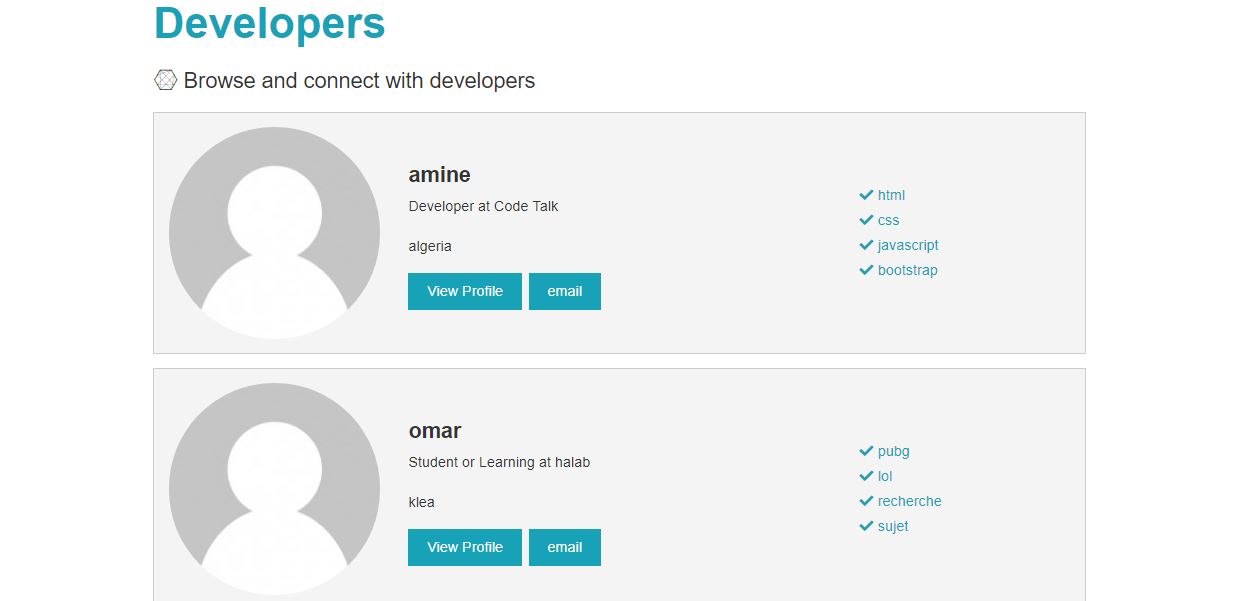


Figure 3.4: L’interface des membres

### Consulter les profiles

L’utilisateur peut aussi consulter les profiles un par un en cliquant sur le bouton « view profile »

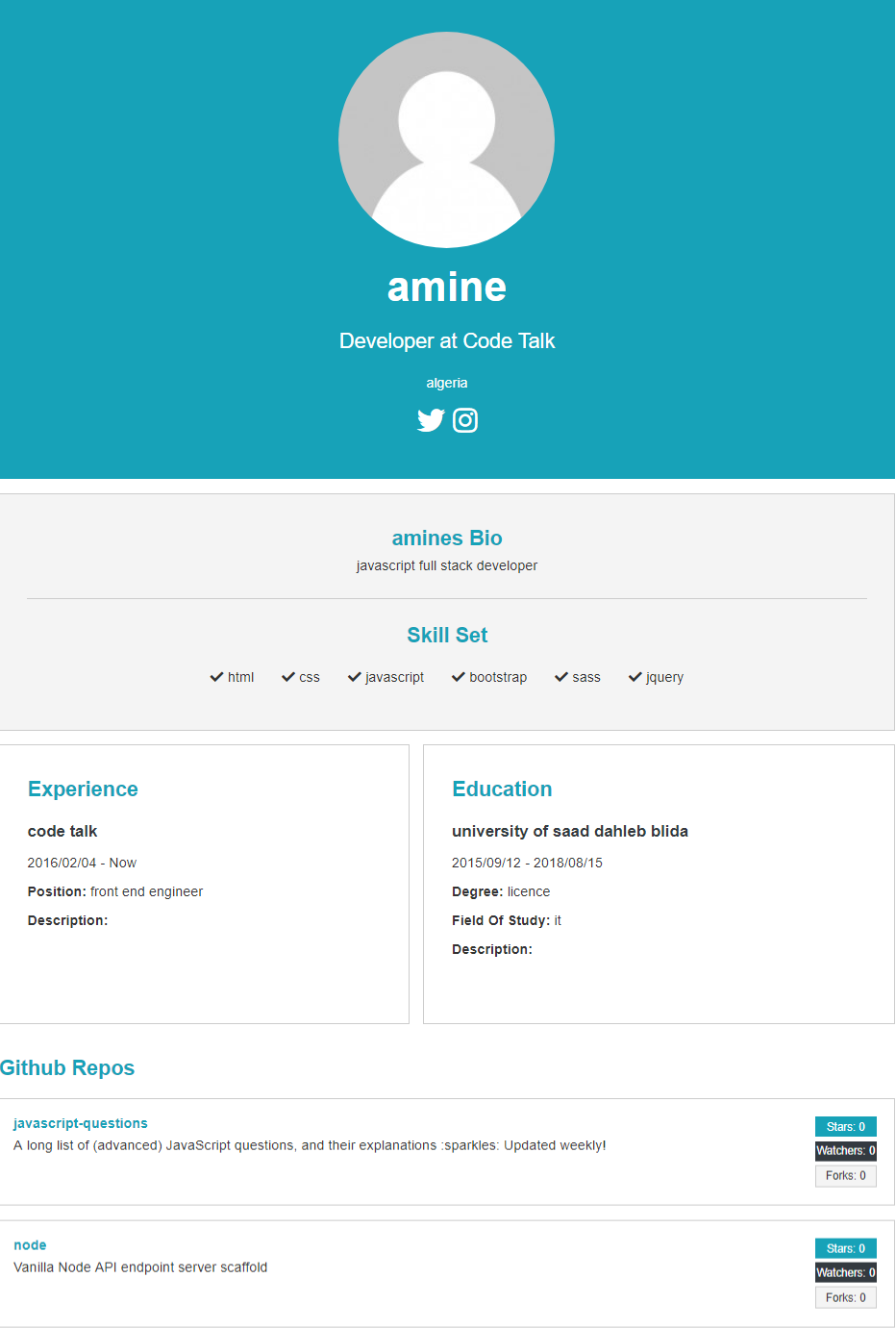


Figure 3.5: L’interface de voir profile

### Dashboard

Les utilisateurs on un espace privé pour modifier leurs informations, ajouter leurs éducations et ajouter aussi leurs expériences dans le domaine.

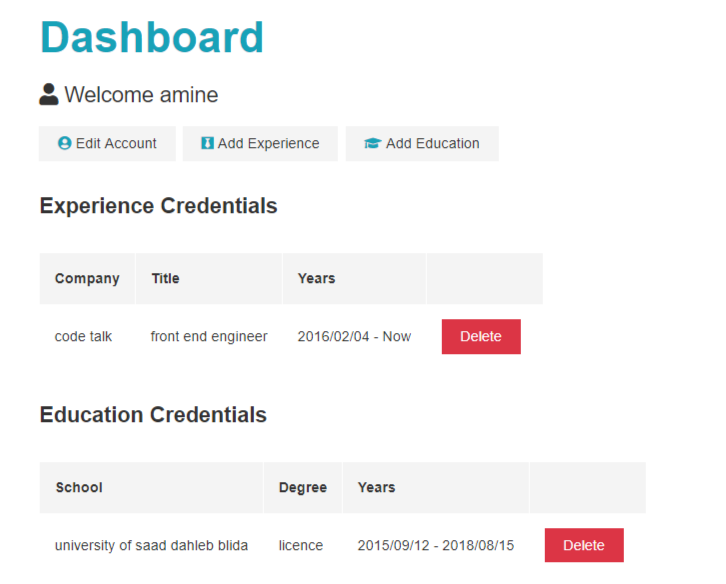


Figure 3.9: L’interface du « Dashboard »

Comme par Example le type de l’utilisateur (Développeur, designer…), le nom de l’entreprise etc…

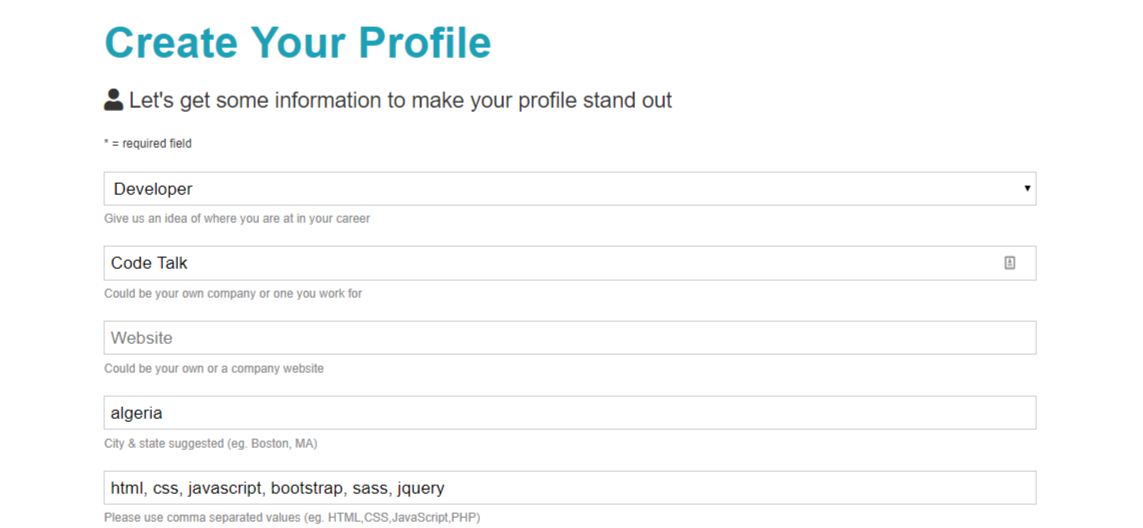


Figure 3.10 L’interface de « edit profile »

### Format mobile

Les 2 interfaces « Consulter Membres » et « Profile » format mobile.

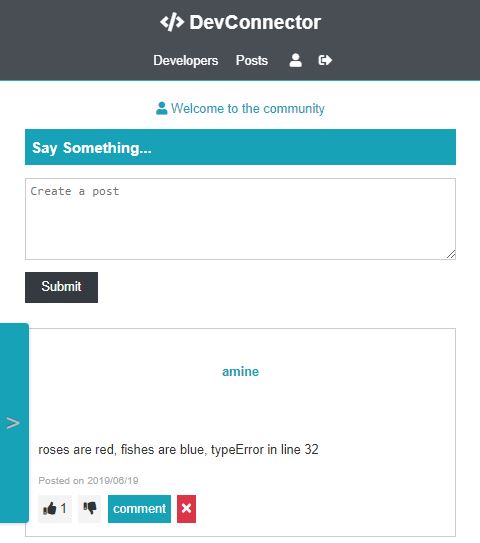
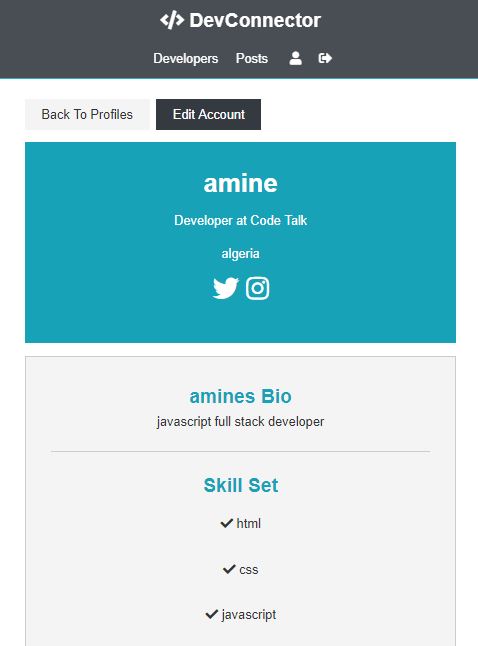
 

Figure 3.19 L’interface de « Login-mobile»

Figure 3.20 L’interface de « inscription-mobile»

# CONCLUSION

Notre projet de fin d’études a constitué une expérience enrichissante qui nous a permis non seulement de mettre en œuvre les connaissances acquises durant les trois années d’études au sein de l’université de Saad Dahleb Blida mais aussi d’améliorer nos compétences au niveau du développement et d’acquérir de nouvelles connaissances informatiques.

Grâce à ce PROJET, nous avons appris à définir les objectifs et mettre en place les moyens de les atteindre tout en bénéficiant de l’aide et de l’évaluation de notre promotrice «M.CHERFA». En fait, notre sujet qui consiste à concevoir et réaliser un réseau social qui permet d’offrir plus de liens et de communication entre les développeurs,  
et de plus être une source d’information qui va aider à faire des recherches.

Durant la réalisation de ce réseau, nous avons pu exploiter l’architecture trois tiers et apprendre à manipuler les langages de développement Web (MERN Stack, HTML, CSS, Bootstrap) et à pratiquer le langage Javascript en utilisant la Framework ReactJs. Nous avons choisi comme méthodologie de développement du site le « Processus Unifié UML ».

Enfin nous avons pu globalement atteindre les objectifs visés et accumuler une masse importante de connaissances aussi bien sur le plan théorique que sur le plan pratique.

# BIBLIOGRAPHY

GitHub repo : (<https://github.com/Zed-M/devconnector3.0>)

Le site : (<http://devconnector.ga>)

Présentation : (<https://prezi.com/jkszfrzwmq29/projet-application-reseaux-social/>)

Des documentations :

( <https://www.etudes-et-analyses.com/theme-economique/reseaux+sociaux> )

(<https://www.electronique-mixte.fr/pfe-rapport-de-projet-de-fin-detude/pfe-25-rapport-de-projet-de-fin-detude-conception-et-mise-en-oeuvre-dun-systeme-de-support-et-dapplication-du-reseau-social/> )

Traversy Media projet : (<https://www.udemy.com/mern-stack-front-to-back/>)

Base de données non relationnel : (<https://www.mongodb.com/cloud/atlas>)

Javascript serveur : (<https://nodejs.org/en/>)

Javascript Client : (<https://reactjs.org/>)