**INSTITUT SUPERIEUR D’INFORMATIQUE**

**ET DES TECHNIQUES DE COMMUNICATION – HAMMAM SOUSSE**

المعهد العالي للإعلامية وتقنيات الاتصال بحمام سوسة

**Rapport de Projet de fin d’études**

Présenté en vue de l’obtention du diplôme de Licence Appliquée en Science Informatique

**Conception et Réalisation d’une Application de Gestion des Formations d’Association ARSII**

***Réalisé par :***

***Akkari Chérifa***

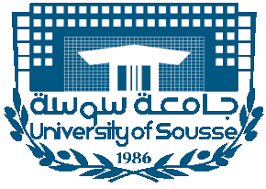
***Département : Multimédia***

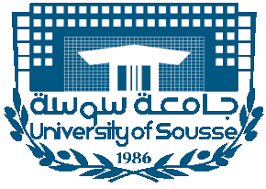
***Encadrant Académique : M. Haythem Saoudi***

***Encadrant Professionnel : M. Ahmed Maalel***

***Société d’accueil***



Année universitaire : 2021/2022****

******MINISTERE DE L’ENSEIGNEMENT SUPERIEUR**

**ET DE LA RECHERCHE SCIENTIFIQUE**

**UNIVERSITE DE SOUSSE**

**المعهد العالي للإعلامية وتقنيات الاتصال بحمام سوسة**

**INSTITUT SUPERIEUR D’INFORMATIQUE**

**ET DES TECHNOLOGIES DE COMMUNICATION – HAMMEM SOUSSE**

**Département Multimédia**

**MEMOIRE DE STAGE FIN D’ETUDES**

**Présenté en vue de l’obtention du diplôme de Licence en science informatique et multimédia**

**Réalisé par :**

*Akkari Chérifa*

**Encadrant :** Mr Haythem Saoudi  **Date :**………………………**Signature :**………….

**Superviseur :** Mr Ahmed Maalel **Date :**………………………**Signature :**………….

Année Universitaire 2021 – 2022

***Dédicaces***

**J’ai Remarqué que toutes les fois où j’ai été Fatiguée, faible**

**La seule personne qui a réussi à trouver les mots justes, pour me redonner Forte : « C’est MOI »**

**J’aimerais me remercier d’avoir travaillé si fort, je veux me remercier de ne pas avoir de journées de repos, je tiens à me remercier de ne jamais m’arrêter et d’avoir tenté de faire plus de bien que de mal et d’être toujours moi-même.**

**Grâce à mon éducation, j'ai toujours cru en moi-même et j'ai travaillé autant que j’ai pu pour arriver ou je voulais être, rien ne m’a jamais été donné.**

**Merci d'être ton propre héros Chérifa, Merci pour le combat merci d'avoir survécu et d'avoir grandi, Merci pour tout ce que tu fais pour toi.**

**Merci de me donner une autre chance de me prouver que je peux faire des choses, que je peux accomplir des choses et que je peux m'améliorer.**

**Merci de m'avoir donné la chance de réécrire mes objectifs, de réévaluer ce que je veux et de faire les plans pour y arriver.**

**Merci de m'avoir rappelé qu'il n'y a pas d'échec, que je peux toujours réessayer et que je peux toujours changer les choses à chaque nouveau souffle et chaque nouveau jour.**

**Merci beaucoup de ne jamais m'avoir abandonné et de m'avoir aidé à trouver la force d'être moi-même, du mieux que je peux, chaque jour !**

**« Très Fière De Moi »**

****

***Dédicaces***

C’est un moment de plaisir de dédier le fruit de mes 16 ans d’études à :

**A Mes chers parents « Karim Akkari » et « Sameh Masri »** Qui m’ont doté d’une éducation digne, leur amour a fait de moi ce que je suis aujourd’hui, qui m’ont toujours poussée et motivé dans mes études. Aucune dédicace, aucun mot ne pourrait exprimer à leur juste valeur la gratitude et l’amour que je vous porte. Chaque ligne de cette mémoire, chaque mot et chaque lettre vous expriment la reconnaissance, le respect, l’amour, l’estime et le remerciement d’être mes parents. Que dieu leur procure bonne santé et longue vie.

**A Mon Cher Frère Karim**

Ta présence, ta joie de vivre et tes petites bêtises sont pour moi une véritable délectation. Que Dieu te protège, te procure santé, réussite et longue vie.

**A Ma grand-mère « Latifa Bazine »**

Que ce modeste travail, soit l’expression des vœux que tu n’as cessé de formuler dans vos prières. Tu t’es toujours sacrifiée pour nous. Ton affection, ta tendresse et ton amour inconditionnel n’ont d’égal que mon amour et mon estime pour toi. Que dieu te procure bonne santé et longue vie.

**A Ma grand-mère « Chérifa Hammouda » et Mon grand-père « Mohamed Masri »**

J’aurais souhaité votre présence en ce moment pour partager ma joie, vous m’avez toujours fait preuve d’amour et d’affection, vous êtes toujours présents dans mon esprit et dans mon cœur. Aussi dans ce moment de joie, vous avez toutes mes pensées. Que vos âmes reposent en paix. J’aimerais citer cette phrase de Victor Hugo, qui prend tout son sens ici : « Tu n’es plus là où tu étais, mais tu es partout là où je suis ».الله يرحمكم و ينعمكم

**A Mes Oncles Paternelles et Maternelles, A Mes Tantes Paternelles et Maternelles**

Avec qui je partage toutes les circonstances de la vie en famille, je vous remercie pour vos

Conseils, aides et encouragements et spécialement Mon Cher Oncle **Fathi Akkari** qui a toujours été avec nous, même s’il est loin de nous.

**A Mon Cher Ami Bédis Hanzouti**

Pour tout effort versé à mon égard, ton aide précieuse lors des difficultés encourues pour mener à terme ce projet. Pour m’avoir répondu à mes questions, et été toujours là pour moi. Tu étais la raison pour laquelle j’ai appris ces nouvelles technologies. Merci énormément mon Cher ami.

A toutes mes **Cousines**, mes **Cousins**, toute la **Famille**, tous les **Amis**, tous les **Collègues** **d’études**, et Mes Chers **Amis d’enfance**, toute personne de près ou de loin qui m’a aidée, avec **qui** j’ai partagé des moments de joie et de bonheur trouvent ici l’expression de mon amour et reconnaissance.

****

******

***Remerciements***

Je tiens tout d’abord à exprimer ma profonde gratitude à

Mon encadrant pédagogique Monsieur **Haythem Saoudi**, Notre Professeur à l’institut supérieur d’informatique et technique de communication HS, pour tous ses précieux conseils, ses remarques et suggestions, pour son écoute active, sa disponibilité et aussi de son suivi permanent de mon travail. Merci beaucoup Cher Monsieur.

Mon Encadrant Technique de L’ARSII,

Monsieur **Ahmed Maalel**

Qui m’a donné cette opportunité de réaliser l’application de son association et pour ses conseils pratiques et directifs.

Merci beaucoup Cher Monsieur.

Mes Vifs remerciements vont également Aux **Membres du Jury**

Qui m’ont fait un grand honneur en acceptant d’évaluer mon modeste travail.

****

**Liste des Figures**

[Figure 1 : Logo de l'association ARSII 16](#_Toc104161578)

[Figure 2 : Logo Eurécia 17](#_Toc104161579)

[Figure 3 : Organisation du travail avec la méthode Kanban 19](#_Toc104161580)

[Figure 4 : Organisation du travail avec la méthode en Cascade 19](#_Toc104161581)

[Figure 5 : Diagramme de cas d'utilisation Globale 26](#_Toc104161582)

[Figure 6 : Diagramme de cas d'utilisation raffiné "Sauthentifier" 27](#_Toc104161583)

[Figure 7 : Diagramme de cas d'utilisation raffiné "gérer les formations" 28](file:///C:\Users\ASUS\Desktop\nouvelle%20version%20rapport%20(2).docx#_Toc104161584)

[Figure 8 : Raffinement de cas d’utilisation "Gérer les candidatures " 31](#_Toc104161585)

[Figure 9 : Interaction entre les couches "MVC" 35](#_Toc104161586)

[Figure 10 : Diagramme de séquence "S’inscrire" 36](#_Toc104161587)

[Figure 11 : Diagramme de séquence "S’authentifier" 37](#_Toc104161588)

[Figure 12 : Diagramme de séquence "Ajouter une Formation" 38](#_Toc104161589)

[Figure 13 : diagramme de séquence "Modifier une formation" 39](#_Toc104161590)

[Figure 14 : Diagramme de classe 40](#_Toc104161591)

[Figure 15 : Logo de Draw.io 46](#_Toc104161592)

[Figure 16 : Logo Visual Studio Code 46](#_Toc104161593)

[Figure 17 : Logo Postman 47](#_Toc104161594)

[Figure 18 : Logo MongoDB 47](#_Toc104161595)

[Figure 19 : Logo Mongoose 47](#_Toc104161596)

[Figure 20 : Interaction entre les composants de la pile "MEAN" 48](#_Toc104161597)

[Figure 21 : Logo Angular 48](#_Toc104161598)

[Figure 22 : Logo Node.js 49](#_Toc104161599)

[Figure 23 : Logo Express.js 50](#_Toc104161600)

[Figure 24 : Logo Git 50](#_Toc104161601)

[Figure 25 : Mappage d'objets entre Node et MongoDB géré via Mongoose 50](#_Toc104161602)

[Figure 26 : Logo HTML 51](#_Toc104161603)

[Figure 27 : Logo CSS 51](#_Toc104161604)

[Figure 28 : Logo Bootstrap 51](#_Toc104161605)

[Figure 29 : Interface D'accueil 52](#_Toc104161606)

[Figure 30 : Interface d’inscription (adhérent) 52](#_Toc104161607)

[Figure 31 : Interface d'inscription (Entreprise) 53](#_Toc104161608)

[Figure 32 : Interface d’authentification 53](#_Toc104161609)

[Figure 33 : Dashboard Admin 54](file:///C:\Users\ASUS\Desktop\nouvelle%20version%20rapport%20(2).docx#_Toc104161610)

[Figure 34 : Calendrier 54](#_Toc104161611)

[Figure 35 : Interface d'ajout d'une Formation 55](#_Toc104161612)

[Figure 36 : Interface de détails d'une formation 55](#_Toc104161613)

[Figure 37 : Interface de modification d'une formation 56](#_Toc104161614)

[Figure 38 : Interface de liste des participants d'une formation 56](#_Toc104161615)

[Figure 39 : interface de Liste des propositions 57](#_Toc104161616)

[Figure 40 : Liste des Candidats 57](#_Toc104161617)

[Figure 41 : Interface de Profil d'un Candidat 58](#_Toc104161618)

**Liste des Tableaux**

[Tableau 1 : Comparatif des Solutions 19](#_Toc104198863)

[Tableau 2 : Comparaison entre les méthodologies 21](#_Toc104198864)

[Tableau 3 : Planification 22](#_Toc104198865)

[Tableau 4 : Diagramme de cas d’utilisation raffiné "S’authentifier" 29](#_Toc104198866)

[Tableau 5 : Description Textuelle de cas d’utilisation "Ajouter une formation" 31](#_Toc104198867)

[Tableau 6 : Description textuelle de cas d’utilisation "Modifier les données d’une formation" 31](#_Toc104198868)

[Tableau 7 : Description textuelle de cas d’utilisation "Supprimer une formation" 32](#_Toc104198869)

[Tableau 8 : Description textuelle de cas d’utilisation "Participer à une formation 33](#_Toc104198870)

[Tableau 9 : Description textuelle de cas d’utilisation "Gérer une candidature" 34](#_Toc104198871)

[Tableau 10 : Admin 42](#_Toc104198872)

[Tableau 11 : Adhérent 43](#_Toc104198873)

[Tableau 12 : Formation 43](#_Toc104198874)

[Tableau 13 : Candidat 44](#_Toc104198875)

[Tableau 14 : Entreprise 44](#_Toc104198876)

[Tableau 15 : Suggestion 44](#_Toc104198877)

[Tableau 16 : Competence 45](#_Toc104198878)

[Tableau 17 : Todo 45](#_Toc104198879)

[Tableau 18 : Configuration du Matériels utilisés 47](#_Toc104198880)

**Liste des acronymes**

**MEAN :** MongoDB, Express, Angular, NodeJS.

**JSON :** JavaScript Object Notation.

**CSS** : Cascading Style Sheets.

**MVC** : Model View Controller.

**UML**: Unified Modeling Language.

**I/O :** input / output

**Introduction générale**

Depuis quelques années que les technologies d'informations et les activités des organisations et plus particulièrement le web ont évolué d'une façon croissante et remarquable. Le monde entier témoigne de l’émergence d’une révolution dans la représentation, l’organisation et le traitement des données. Différents domaines profitent aujourd’hui de cette révolution à travers divers services et applications web, surtout dans le domaine informatique.

Afin de maintenir leur position sur le marché et de croître, les Entreprises d'aujourd'hui doivent réaliser leur plein potentiel face à une concurrence croissante. Elles essaient d'utiliser autant de ces technologies que possible afin d'augmenter leur productivité et faire face à des problèmes compliqués qui peuvent constituer des obstacles au progrès. C’est le même principe pour l’Association d’informatique et innovation en informatique « ARSII » qui occupe un nombre important des ingénieurs et des adhérents. Pour être la plus compétitive possible, cette Association doit s'optimiser, s'adapter, vivre le changement, comprendre ses ressources et les connaissances disponibles, s'organiser avec souplesse et faire travailler ses ressources rapidement dans toutes les situations.

Une gestion des adhérents ainsi que des Formations est une fonction importante pour cette association. Donc bien gérer les formations c’est mettre en œuvre une méthode pour organiser le plan des événements selon le besoin des adhérents sans perte de temps. C’est dans ce contexte que s’inscrit notre projet de fin d’étude.

Bien que nous ne soyons pas ni les premiers ni les derniers à traiter ce sujet. Nous nous proposons de développer une application web simple destinée à toutes les personnes qui font partie de cette Association pour assurer un meilleur service entre l’association et ces adhérents et une meilleure planification des formations.

Pour ce faire, notre rapport sera subdivisé en 4 chapitres :

Le premier intitulé « Etude Préalable » est consacré à la présentation du cadre général du projet et à l’organisme d’accueil ainsi qu’à la présentation du contexte, de l’objectif, du travail demandé.

Le second chapitre nommé « Analyse et spécification des besoins » se focalise sur la détermination des besoins fonctionnels et non fonctionnels ainsi que la présentation du diagramme de cas d’utilisation nécessaire pour chaque processus.

Le troisième chapitre « Conception de la Solution » présente la phase conceptuelle qui sera matérialisée par l’élaboration du diagramme de classes ainsi que celle du diagramme de séquence.

Quant au quatrième « Mise en œuvre et réalisation » nous y exposons les différentes interfaces web développées pour donner une idée sur notre application et une revue sur toutes les technologies utilisées.

Enfin, nous clôturons notre rapport par une conclusion générale et les perspectives possibles pour améliorer dans le futur ce qui a été réalisé.

**Chapitre I : Etude préalable**

**Chapitre I :**

**Etude Préalable**

**Introduction**

Ce premier chapitre aidera à mettre le projet dans son contexte général. En premier lieu, nous allons nous intéresser à la présentation de l’organisme d’accueil « ARSII » au sein de laquelle nous avons effectué ce projet de fin d’étude. Ensuite, nous allons mettre l’accent sur l’étude de l’existant et exposons la solution proposée et les différentes tâches à réaliser.

1. **Présentation de l’organisme d’accueil ARSII**
2. **Historique**

Association de Recherche Scientifique et Innovation en Informatique « ARSII », une jeune association scientifique et apolitique, a été fondée en Mai 2016 par des jeunes enseignants chercheurs en Informatique.

1. **Objectifs de ARSII :**

ARSII a pour Objectif :

* Contribuer à l’appui de la recherche scientifique dans le domaine de l’informatique et de l’innovation ;
* Aide à l’insertion des étudiants dans le milieu professionnel ;
* Proposition, encadrement et suivi de projets innovants.
* Rapprochement de la recherche scientifique et académique au monde de l’industrie.
* Coopération et collaboration avec les partenaires socio-économiques à l’échelle nationale et internationale.

1. **Fiche de Renseignement de l’Association**

* Nom de l’association : Association de recherche scientifique et innovation en informatique « ARSII ».
* Date de création : Mai 2016.
* Fondateurs : des jeunes chercheurs et enseignants en informatique.
* Adresse : Immeuble Louzir, Rue Palestine, Sahloul, Sousse, Tunisie.
* Contact : [contact@arsii.org](mailto:contact@arsii.org) / 73382658.

****

Figure 1 : Logo de l'association ARSII

1. **Cadre du projet :**

Ce projet s’inscrit dans le cadre du projet de fin d'étude en vue de l’obtention du diplôme en licence en science informatique de l'institut supérieure d’informatique et de technique de communication à Hammam Sousse. Il est élaboré au sein de l’association de Recherche scientifique et innovation en informatique « ARSII » Sousse.

1. **Problématique :**

Le responsable a distingué quelques problèmes au sein de l’association, cette dernière planifie les formations sans consulter les étudiants pour avoir idée de leurs besoins, au fait quelques formations ont été réalisées en vain car il n’y a pas eu de public intéressé ou seulement quelques-uns, puisque ça ne répondait pas à leurs besoins, et inversement. En plus la seule source d’information c’est les réseaux sociaux, au fait les demandes de formateurs y sont publiées, sinon il faut se déplacer au local de l’association pour donner les documents nécessaires pour cette offre. Et aussi le responsable n’a pas une idée claire des compétences des adhérents donc généralement les taches qui leur sont attribuées ne correspondent pas à leurs compétences informatiques.

Il est important que le bureau de cette association résoudre ces problèmes pour :

* Investir dans les compétences qu’ils représentent.
* Impliquer les adhérents dans le domaine d’informatique.
* Repérer, faire émerger et cultiver le potentiel de tous.
* Embaucher des profils compétents.
* Fidéliser les adhérents afin qu’ils restent dans l’association et développer leurs compétences afin de les exploiter à leur plein potentiel.
* Placer le nouveau talent au bon endroit et pousser l’intelligence collective.

1. **Présentation du projet**

Ce projet consiste à concevoir une application web simple à manipuler adaptée aux besoins des adhérents et de responsable de l’association. Cette application pourrait aider ces utilisateurs à accéder dans leurs espace interne, assurer une meilleure organisation entre les formations et le besoin des adhérents.

1. **Etude et Critique de l’existant**

La recherche existante est une étape importante de la réalisation d'un projet, en effet elle consiste à comprendre et analyser les solutions existantes et à identifier leurs faiblesses et leurs forces afin de pouvoir identifier les besoins du projet et en tenir compte lors de la conception et de l'exécution de notre projet. Dans ce cadre, nous menons des recherches sur quelques applications qui existent dans le monde avec des thématiques similaires aux nôtres, nous donnons un exemple :

* **Description de l’existant :**

**Eurécia :**

Le SIRH est un Système d'Information sur les Ressources Humaines simple et complet qui, entre autres, gère mieux et automatise la plupart des tâches liées à la gestion des ressources humaines d'une entreprise. C'est un logiciel composé de plusieurs blocs logiciels appelés modules.



Figure 2 : Logo Eurécia

* **Critique de l’existant :**

|  |  |  |
| --- | --- | --- |
|  | Avantages | Inconvénients |
| **Eurécia** | - Plateforme collaborative.  - La sécurité et la confidentialité des données.  - Une solution accessible et collaborative  - Le regroupement des taches  - Vision stratégique des RH  - Améliorez le bien-être au travail | **-** Interface compliquée et très chargé.  - Le design ennuie les utilisateurs.  - concerne Just les entreprises privés    - Mauvaise ergonomie (l’organisation d’un plan de formation nécessite plusieurs clics et affichage de plusieurs interfaces).  - Comporte plusieurs modules (congés, compétences, planning …) |

**Tableau 1 : Comparatif des Solutions**

1. **Solution proposée :**

Dans le cadre décrit précédemment, l’idée de notre projet consiste à développer une application web simple pertinente qui répond aux besoins du publique ciblé avec quelques fonctionnalités permettant à l’association d’offrir un meilleur service à ces adhérents et ses partenaires comme :

* Organisation d’un plan de formation efficace et répond au besoin des adhérents.
* Donner la possibilité de gérer les candidatures pour les différentes offres ce qui permet l’augmentation de la chance pour trouver des emplois.
* Gagner le temps dans la recherche des formations.
* Avoir une idée sur les compétences informatiques de chaque membre.
* Visualiser les compétences de chaque adhérent à travers des courbes.
* Rapidité et efficacité lors de la recherche du meilleur profil.

1. **Travail demandé :**

Dans le cadre de notre stage de fin d’étude effectué dans l’association ARSII, on nous a proposé de concevoir et réaliser une application web qui se focalise sur le module de gestion des formations ainsi que la gestion des adhérents et la gestion des candidatures. Les principales fonctionnalités étant de gérer les adhérents de l’association, gérer les formations ainsi que gérer les candidatures.

1. **Comparaison de Quelques méthodologies**

Les méthodes de développement sont nombreuses, on trouve :

* 1. **La méthodologie Kanban [1] :**

C’est une méthodologie de gestion de projet similaire à bien des égards à Scrum. Mais Kanban est un changement plus évolutif dans le monde Agile. La méthodologie Kanban est légère sur les processus, flexible, n’a pas de rôles prescrits et essaie simplement d’améliorer le débit en augmentant l’attention de l’équipe sur les choses qui comptent vraiment

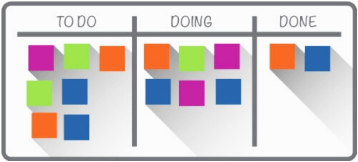


Figure 3 : Organisation du travail avec la méthode Kanban

* 1. **La Méthodologie Cascade [1] :**

La méthode Cascade est divisée en étapes distinctes. Tout d’abord, on commence par collecter et analyser les exigences, concevoir la solution, la mettre en œuvre et résoudre les problèmes, le cas échéant. Chaque étape de ce processus est autonome. Il est nécessaire de terminer une étape avant de passer à une autre.

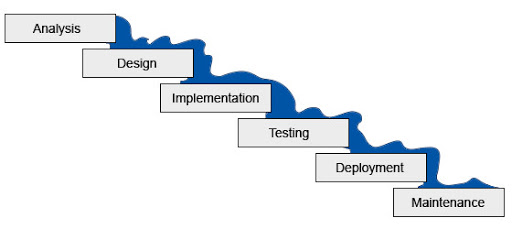


Figure 4 : Organisation du travail avec la méthode en Cascade

|  |  |  |
| --- | --- | --- |
|  | **Avantages** | **Inconvénients** |
| **Kanban** | - Facile à apprendre  - Améliore le flux de travail global  - Minimise le cycle de temps  - Augmente la flexibilité des processus et réduire leur cycle de temps | - Une carte kanban obsolète peut entrainer des problèmes de développement  - Manque de timing  - Pas de délais associés à chaque étape. |
| **Cascade** | - La facilité d’utilisation  - La démarcation claire entre les étapes permet d’organiser et de diviser le travail | -Un risque plus élevé : s’il y a une erreur ou une modification, il faut démarrer le projet dès le début  - Risque d’échec du projet |

Tableau 2 : Comparaison entre les méthodologies

1. **Choix de la Méthodologie**

Dans le cadre de notre projet, nous optons pour la méthodologie Cycle en V :

* 1. **La Méthodologie Cycle En V [2] :**

La méthode du cycle en V est intuitive et facile à mettre en place. Il ne nécessite pas de temps de formation pour l'équipe projet, juste quelques séances de suivi et de coaching. Une phase de validation planifiée dès le début d'un projet peut également limiter les risques lors de la conception

La méthode est divisée en trois phases :

* **Conception :** La conception d'un logiciel selon la méthode du cycle en V implique quatre étapes : l'analyse des besoins, les spécifications, la conception générale et la conception détaillée.
* **Réalisation :** La réalisation comprend le choix du langage de programmation en fonction des attentes fonctionnelles et architecturales établies lors de la phase de conception et le codage du logiciel.
* **Validation :** La phase de validation contient différents tests vérifiant chaque étape de la conception du produit.

1. **Planification**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Février** | | | | **Mars** | | | | **Avril** | | | | **Mai** | | | |
| **Semaine** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** |
| Formation  Angular & Node-JS |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Etude  Préalable |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Modélisation  Conceptuelle |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Réalisation |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Validation &  Tests |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Rédaction du  Rapport |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Tableau 3 : Planification

**Conclusion**

Durant ce Chapitre, nous avons mis le projet dans son contexte général.

Nous avons tout d’abord introduit le cadre d’accueil, après nous avons mis l’accent sur la problématique du projet et les différents objectifs que nous souhaitons atteindre dans notre projet afin de donner une vue générale sur le projet et une étude de l’existant avec ces avantages et ces inconvénients, et nous avons exhibé notre solution proposée.

Finalement, nous avons réalisé une planification périvisuelle à suivre afin de subdiviser l’enchainement du travail.

**Chapitre II** **: Analyse**

**Et Spécification des Besoins**

**Chapitre II :**

**Analyse et spécification des besoins**

**Introduction :**

Le développement de toutes applications consiste à analyser les besoins de base afin de bien adapter la liste des fonctionnalités. Dans ce chapitre, nous nous intéressons à spécifier les différents besoins fonctionnels et non fonctionnels de notre application dans le but d’identifier par la suite les différents acteurs avec une présentation des diagrammes de cas d’utilisation système et leurs descriptions textuelle.

1. **Identification des acteurs**

Un acteur représente une entité externe qui interagit directement avec le système. En réponse à l’action d’un acteur, le système fournit un service qui répond à ses besoins. Pour notre application, il y aura 4 acteurs :

* **L’administrateur :** C’est le gérant de l’application, il a une visibilité totale sur toute la base des données de l’application. Il a pour tâche de gérer le contenu de tout le système.
* **L’adhérent :** c’est un utilisateur de notre application. Il a pour tâche de gérer son profil après une inscription avec une gestion de ses compétences.
* **L’entreprise :** c’est un utilisateur de notre application. Il a pour tâche de consulter la liste des adhérents et faire des recrutements.
* **Le candidat :** il a pour tâche d’envoyer sa candidature.

1. **Spécification des besoins**

L’application considérée doit satisfaire principalement des besoins fonctionnels effectués par le système et des besoins non fonctionnels qui améliorent la qualité logicielle du système.

1. **Besoins fonctionnels**
   1. **Les besoins fonctionnels de l’administrateur**

* **Authentification**
* **Gestion des Adhérents :**
* Le système doit permettre à l’administrateur d’accéder aux comptes des adhérents, supprimer les adhérents.
* Le système doit permettre à l’administrateur de visualiser les compétences de chaque adhérent à travers des courbes.
* **Gestion des formations :**
* L’administrateur visualise les suggestions des formations et organise le plan des formations selon les propositions des adhérents.
* Le système doit permettre à l’administrateur d’ajouter/ supprimer ou modifier les formations sur un calendrier bien organisé.
* **Gestion des Candidatures :**
* Le système doit permettre à l’administrateur d’accéder à la liste des candidats, voir leurs informations.
* L’administrateur répondre à cette candidature que ce soit par Email de confirmation ou de rejection.
* Un email sera envoyé au Candidat à travers l’email de l’association.
* **Gestion des Compétences :**
* L’administrateur peut ajouter
  1. **Les Besoins fonctionnels de l’adhèrent :**

Par rapport à l’adhérent, notre application doit répondre aux exigences suivantes :

* **Inscription et Authentification**
* Pour devenir membre de cette association, il faut que l’adhérent s’inscrire à travers un formulaire d’inscription.
* **Gérer son profil**
* C’est un espace réservé à l’adhérent. Chaque adhérent doit préciser ses coordonnées. L’accès à cette espace nécessite une authentification.
* L’adhérent peut modifier ses propres données.
* L’adhérent peut ajouter ses compétences à travers un formulaire d’ajout.
* L’adhérent peut visualiser la courbe de ses propres compétences.
* **Gérer les suggestions**
* L’adhérent peut proposer une formation, l’a supprimé ou l’a modifié
* Chaque adhérent peut visualiser le plan des formations, Chercher une formation, visualiser les données de chaque formation.
  1. **Les Besoins fonctionnels de l’entreprise :**

Par rapport à l’entreprise, notre application doit répondre aux exigences suivantes :

* **Inscription**
* Cette dernière doit remplir un formulaire d’inscription pour avoir un compte.
* **Gérer son Espace**
* C’est un espace réservé à l’entreprise. Chaque entreprise doit préciser ses coordonnées. L’accès à cet espace interne nécessite une authentification.
* L’entreprise peut consulter la liste des Adhérents.
* L’entreprise peut visualiser les données de chaque adhérent.
* L’entreprise peut contacter l’adhérent souhaité pour un recrutement.
  1. **Les Besoins fonctionnels du Candidat**

Par rapport au candidat, notre application doit répondre aux exigences suivantes :

* Le candidat doit remplir un formulaire de candidature.
* Il attend une réponse concernant sa candidature par Email.

1. **Besoins non fonctionnels**

Les besoins non fonctionnels présentent l’ensemble des règles à suivre et à respecter pour assurer une bonne qualité de l’application à réaliser avec un bon fonctionnement du futur système. Ils se récapitulent en :

* **Extensibilité :** Le code doit être clair pour permettre de futures évolutions ou améliorations.
* **Intégrité :** Garantir la cohérence des données à chaque mise à jour aussi bien d’insertion, de modification et ou de suppression.
* **Ergonomie des interfaces :** Les interfaces doivent être conviviales, facile à utiliser et compréhensibles par tous les utilisateurs (la manipulation de l’interface ne doit pas nécessiter des connaissances poussées). L’application doit respecter les normes de qualités.
* **Sécurité :** Notre plateforme permet la sécurité d’accès et cela est assuré par une authentification des utilisateurs en fonction de leurs droits d’accès, donc chaque utilisateur possède un espace privé.
* **Confidentialité :** Chaque utilisateur peut accéder à son espace sachant qu’il est déjà authentifié à travers son login et mot de passe.
* **Portabilité :** La plate-forme est censée être utilisable sur différents types d’appareils.

1. **Diagramme de cas d’utilisation**
2. **Langage de modélisation UML :**

Le Langage UML (Unified Modeling Language, ou langage de modélisation unifié) a été pensé pour être un langage de modélisation visuelle commun, et riche sémantiquement et syntaxiquement. Il est destiné à l’architecture, la conception et la mise en œuvre de systèmes logiciels complexes par leur structure aussi bien que leur comportement.

1. **Définition d’un diagramme de cas d’utilisation**

Un diagramme de cas d’utilisation est un graphe d’acteur qui capture le comportement fonctionnel du système**.** On va modéliser les besoins des utilisateurs en identifiant les grandes fonctionnalités et les limites du système en respectant les interactions entre le système et ses utilisateurs.

1. **Diagramme de cas d’utilisation global**

La figure suivante présente le diagramme de cas d’utilisation général de notre application.

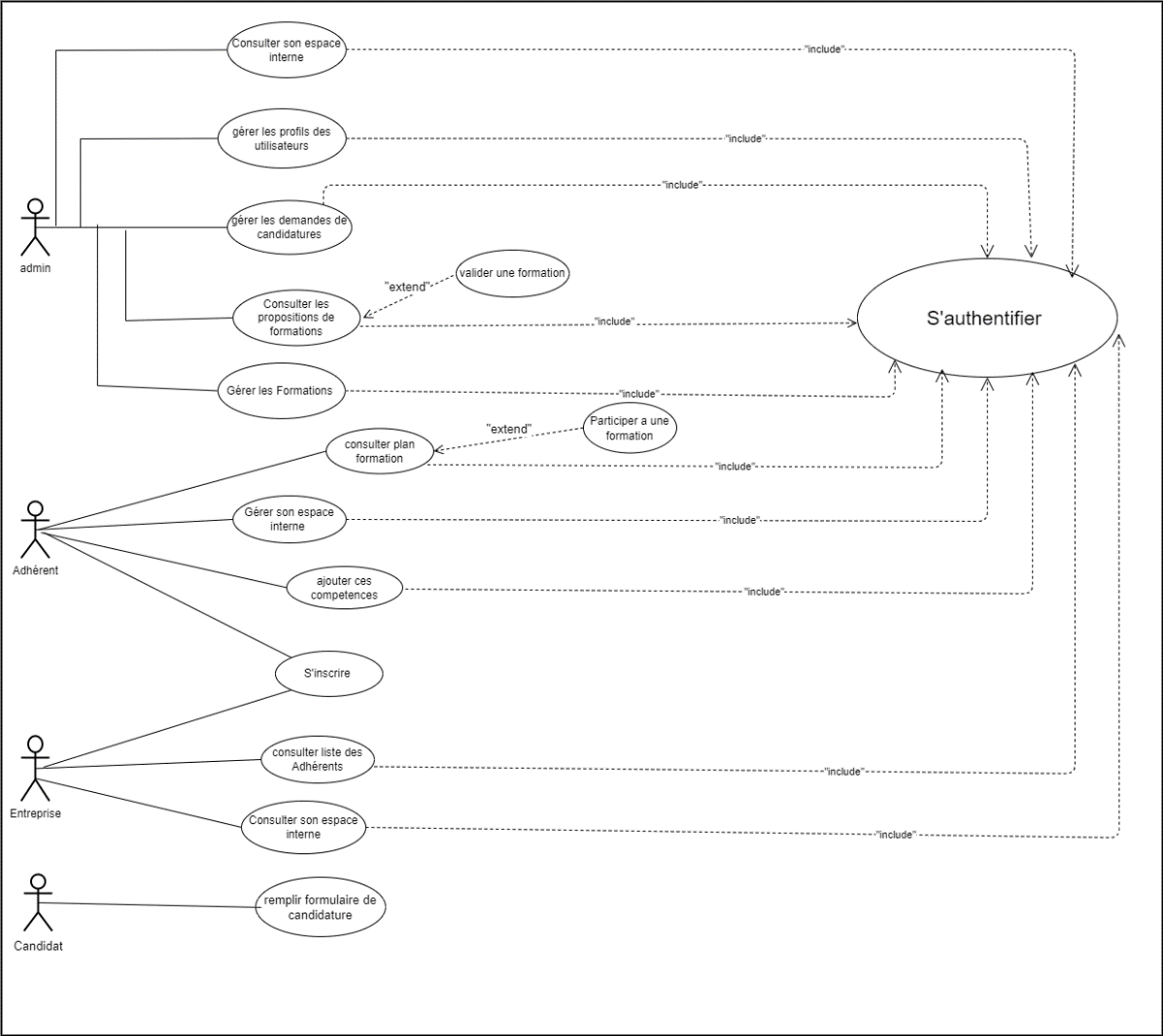


Figure 5 : Diagramme de cas d'utilisation Globale

1. **Raffinements des cas d’utilisation** 
   1. **Raffinement de cas d’utilisation « S’authentifier »**

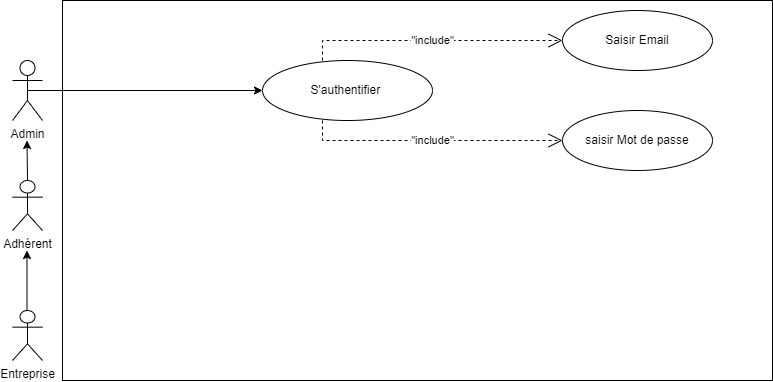
****

Figure 6 : Diagramme de cas d'utilisation raffiné "Sauthentifier"

* + 1. **Description textuelle de diagramme de cas d’utilisation « S’authentifier » :**

|  |  |
| --- | --- |
| **Cas d’utilisation :** | S’authentifier |
| **Acteurs :** | Administrateur – Adhérent – Entreprise |
| **Post-Condition :** | L’utilisateur doit avoir un compte |
| **Pré-condition :** | L’utilisateur est redirigé vers son espace privé. |
| **Scénario nominal :** | * L’utilisateur demande le formulaire d’identification * Le système affiche le formulaire d’identification * L’utilisateur saisit son email et son mot de passe. * Le système vérifie les informations saisies par l’utilisateur pour déterminer son rôle. |
| **Scénario alternatif :** | * Un champ lui manque la saisie * Le bouton se connecter reste désactiver * Le formulaire d’authentification s’affiche de nouveau * L’utilisateur n’existe pas dans la BD * Un message d’erreur s’affiche. * Le formulaire d’authentification s’affiche de nouveau * Le mot de passe ou Email est incorrect * Un message d’erreur s’affiche * Le formulaire d’authentification s’affiche de nouveau |

Tableau 4 : Diagramme de cas d’utilisation raffiné "S’authentifier"

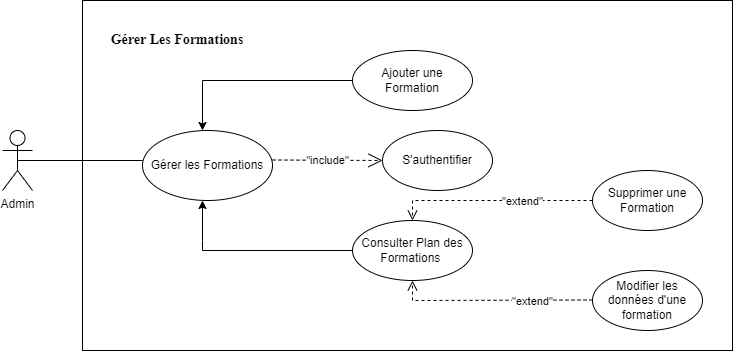
* 1. **Raffinement de cas d’utilisation « Gérer les formations »**

Figure 7 : Diagramme de cas d'utilisation raffiné "gérer les formations"

* + 1. **Description textuelle de diagramme de cas d’utilisation « Gérer les formations » :**

|  |  |
| --- | --- |
| **Cas d’utilisation du système :** | Ajouter une nouvelle formation |
| **Description du cas d’utilisation :** | On crée dans le système une nouvelle Formation |
| **Acteurs :** | Administrateur |
| **Pré-condition :** | Authentification |
| **Post-condition :** | Une nouvelle Formation est ajoutée à la base de données. |
| **Scénario normal :** | - L’Administrateur clique sur la date souhaitée dans le calendrier  - Le système affiche le formulaire d’ajout  -L’administrateur remplit les champs par les informations nécessaires et clique sur le bouton « Ajouter ».  -Le Système vérifie les informations saisies.  - Un message de succès d’ajout d’une nouvelle formation s’affiche.  -La formation ajoutée sera affiché dans le calendrier. |
| **Scénario alternatif :** | **-** Un champ lui manque la saisie  - Le bouton se connecter reste désactiver dès que tous les champs seront remplis.  - La date de fin inférieur aux dates de début => le système affiche une alerte de vérification des dates |

Tableau 5 : Description Textuelle de cas d’utilisation "Ajouter une formation"

**Description textuelle de diagramme de cas d’utilisation « Modifier une formation ».**

|  |  |
| --- | --- |
| **Cas d’utilisation du système :** | Modifier les données d’une formation |
| **Description du cas d’utilisation :** | On modifie dans le système une Formation |
| **Acteurs :** | Administrateur |
| **Pré-condition :** | Authentification |
| **Post-condition :** | Données de formation modifiées |
| **Scénario normal :** | - L’administrateur clique sur la formation dans le calendrier  - Une interface, qui contient les informations de cette formation s’affiche  - L’administrateur clique sur le bouton « Modifier », modifie le formulaire et clique sur le bouton Valider  - Le système vérifie les informations saisies par l’utilisateur.  - Un message de succès de modification des données de la formation s’affiche |
| **Scénario alternatif :** | **1-** l’administrateur sort avant d’avoir enregistré la modification  - le formulaire reste affiché  **2-** l’administrateurremplit le formulaire de modification avec des informations non valides.  - un message d’erreur s’affiche  - le formulaire de modification reste affiché |

Tableau 6 : Description textuelle de cas d’utilisation "Modifier les données d’une formation"

* + 1. **Description Textuelle de diagramme de cas d’utilisation « Supprimer une formation ».**

|  |  |
| --- | --- |
| **Cas d’utilisation du système :** | Supprimer une formation |
| **Acteurs :** | Administrateur |
| **Pré-condition :** | Authentification |
| **Post-condition :** | Formation supprimée |
| **Scénario normal :** | - L’administrateur clique sur la formation dans le calendrier  - Une interface, qui contient les informations de cette formation s’affiche  - L’administrateur clique sur le bouton « Supprimer »  -Une alerte de confirmation s’affiche  -L’administrateur confirme la suppression.  - Un message de succès de suppression de la formation s’affiche |
| **Scénario alternatif :** | -L’administrateur ne confirme pas la suppression  -la formation ne sera pas supprimé |

Tableau 7 : Description textuelle de cas d’utilisation "Supprimer une formation"

1. **Description textuelle de cas d’utilisation « Participer à une formation »**

|  |  |
| --- | --- |
| **Cas d’utilisation du système :** | Participer à une formation |
| **Acteurs :** | Adhérent |
| **Pré-condition :** | Authentification |
| **Post-condition :** | Votre participation est enregistrée |
| **Scénario normal :** | - L’adhérent clique sur la formation souhaitée dans le calendrier  - Une interface, qui contient les informations de cette formation s’affiche  - L’administrateur clique sur le bouton « participer »  -Une alerte de confirmation s’affiche |

Tableau 8 : Description textuelle de cas d’utilisation "Participer à une formation

1. **Raffinement de cas d’utilisation « Gérer les candidatures »**

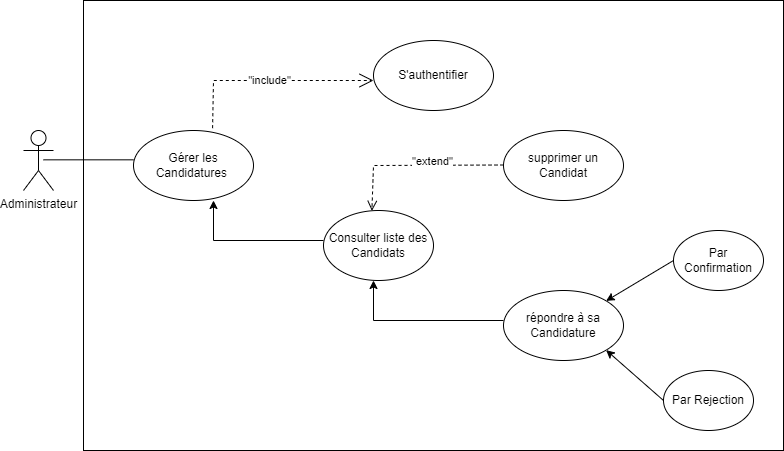
****

Figure 8 : Raffinement de cas d’utilisation "Gérer les candidatures "

* 1. **Description textuelle de cas d’utilisation « Gérer une candidature »**

|  |  |
| --- | --- |
| **Cas d’utilisation :** | Gérer les Candidatures |
| **Acteurs :** | Administrateur |
| **Post-Condition :** | Authentification |
| **Pré-condition :** | Répondre à une Candidature |
| **Scénario nominal :** | * L’administrateur consulte la liste des Candidats * Le système affiche la liste de tous les candidats * L’administrateur clique sur le candidat * Le système affiche les informations personnelles de ce candidat * L’administrateur répondre à cette candidature à l’aide d’un bouton que ce soit de confirmation ou rejection * Un Email sera envoyé au candidat sur son compte Gmail |
| **Scénario alternatif** | * L’email du candidat n’existe pas/n’est pas correcte * L’email ne sera pas envoyé. |

Tableau 9 : Description textuelle de cas d’utilisation "Gérer une candidature"

**Conclusion :**

Ce chapitre a été consacré à la spécification des besoins fonctionnels et non fonctionnels du système en indiquant les rôles de chaque acteur ainsi la présentation des diagrammes des cas d’utilisation. Ces derniers indiquent les différentes interactions entre les utilisateurs et le système résultant. Dans le chapitre suivant, nous entamerons la phase conceptuelle.

**Chapitre III : Conception de la Solution**

**Chapitre III :**

**Conception de la Solution**

**Introduction :**

La Conception est une phase très importante pour la réalisation de tout projet informatique. C’est une démarche basée sur de différentes étapes qui permet de répondre à l’ensemble des besoins, puis d’imaginer des contextes d’utilisation dont le but est d’obtenir une satisfaction maximale des utilisateurs. Elle permet de donner une architecture au projet en découpant en briques. Nous construisons une vue globale de la solution sous forme des diagrammes.

1. **Mise en place de L’architecture Logicielle MVC**

Dans cette partie, nous allons entamer une étude théorique de l’architecture logicielle adoptée pour le développement de notre solution. L’architecture adoptée pour la couche présentation de notre application est l’architecture Mdel-View-Controller. Le MVC est un design pattern (patron de conception). Il permet d’avoir le code plus structuré, plus évolutif, plus maintenable permettant de profiter de plusieurs mécanismes, d’avoir de la persistance de données, et bien d’autres choses encore **[3].**

Le modèle MVC est basé sur trois éléments :

* **Modèle** : rassemble des données du domaine, des connaissances du système. Contient les classes dont les interfaces doivent être vues et manipulées.
* **Vue** : contient les fonctionnalités nécessaires pour gérer et contrôler les interactions de l’utilisateur avec la vue et le modèle.
* **Contrôleur** : contient les fonctionnalités nécessaires pour gérer et contrôler les interactions de l’utilisateur avec la vue et le modèle.

Le principal avantage de choisir cette architecture c’est la séparation de la couche interface utilisateur avec les autres parties du système (car les interfaces utilisateurs sont beaucoup plus susceptibles de changer que la base de connaissances du système).

Les différentes interactions entre le modèle, la vue et le contrôleur sont résumées par le schéma de la figure suivante :

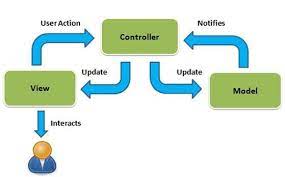


Figure 9 : Interaction entre les couches "MVC"

1. **Diagramme de Séquence**
2. **Définition :**

Les diagrammes de séquences ont pour but de présenter des collaborations entre objets selon un point de vue temporel. Il s’agit d’une séquence d’événements ou de messages échangés pendant le déroulement d’un scénario particulier d’un cas d’utilisation.

1. **Diagramme de séquence « S’inscrire »**

Chaque utilisateur doit s’inscrire à la plate-forme à travers un formulaire d’inscription sur notre site. Chaque utilisateur a un formulaire spécifique à lui. S’il y a des données manquantes, le système affiche un message d’erreur. Cette figure décrit le diagramme de séquence « S’inscrire ».

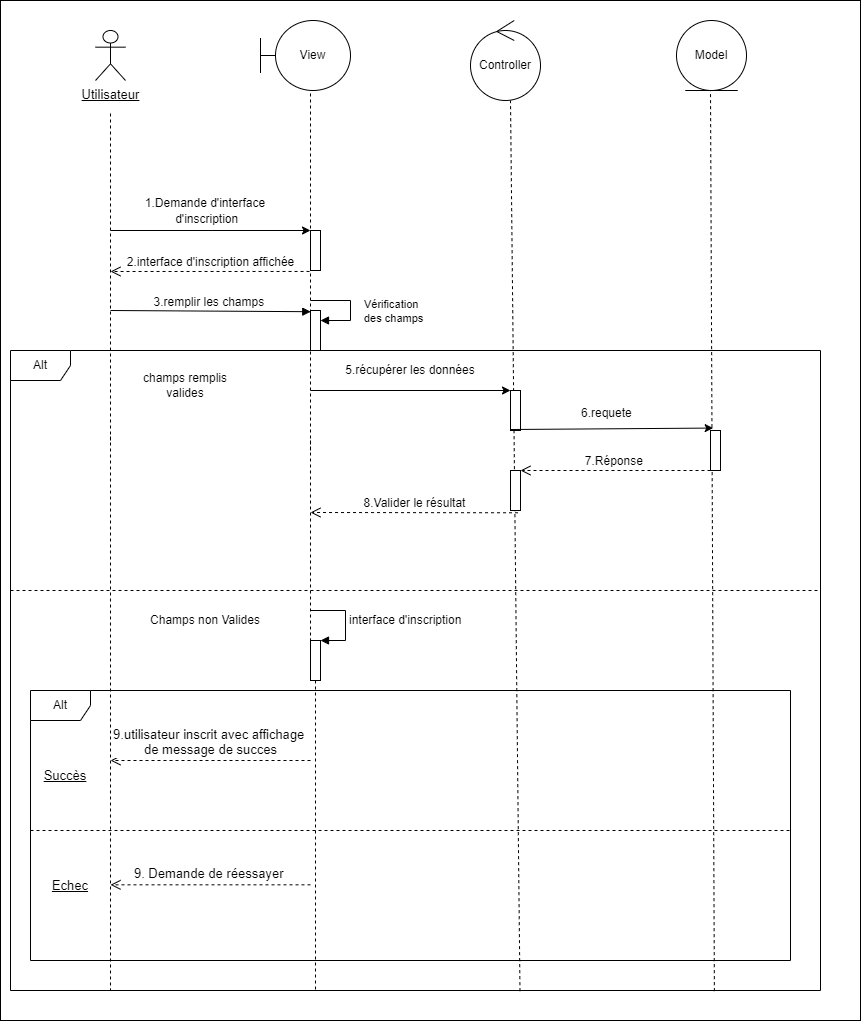


Figure 10 : Diagramme de séquence "S’inscrire"

1. **Diagramme de Séquence « S’authentifier »**

Pour pouvoir accéder à son espace privé, chaque utilisateur doit tout d’abord s’identifier par son email et son mot de passe à travers le système de l’application. Celui-ci prend en charge de vérifier et consulter la base de données :

* S’il est accepté, donc il y’aura l’accès à l’application et à son espace approprié.
* Sinon, le système lui affiche un message d’erreur pour rectifier ses données.

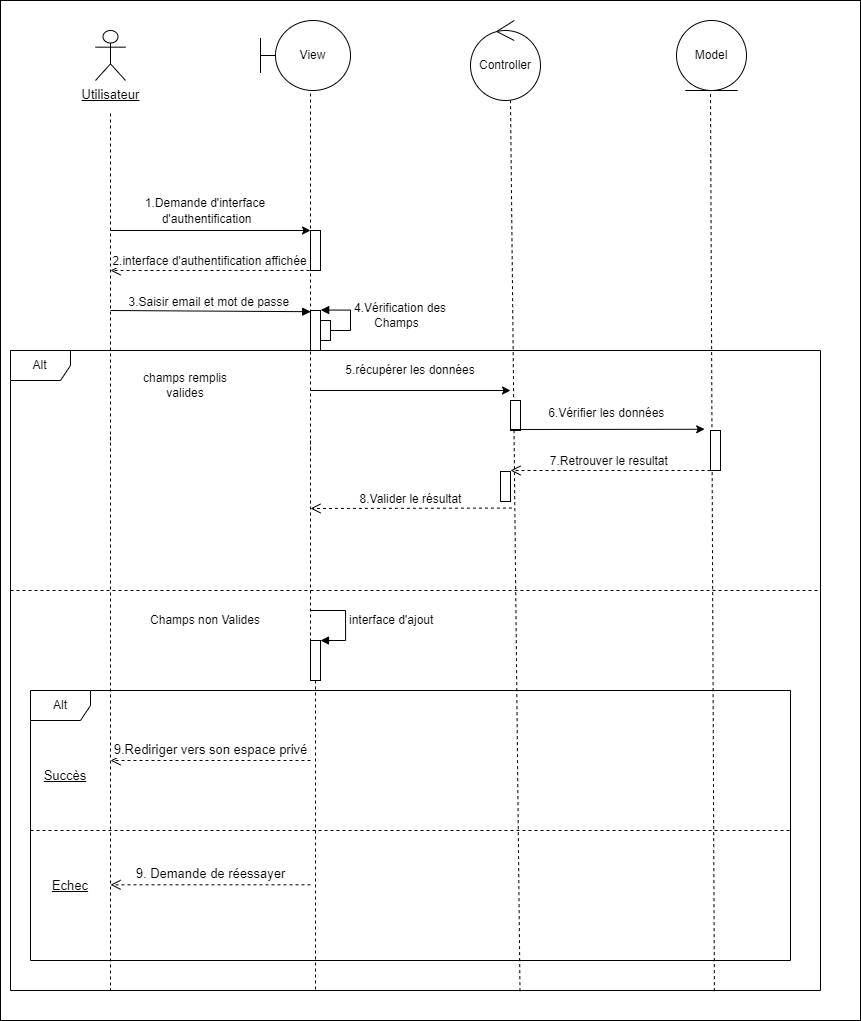


Figure 11 : Diagramme de séquence "S’authentifier"

1. **Diagramme de séquence de « Ajouter une Formation »**

Pour pouvoir ajouter une nouvelle formation, l’administrateur doit tout d’abord s’authentifier, accède au plan de formation, cliquer sur la date souhaitée et un formulaire d’ajout s’affiche, remplir le formulaire, il faut que tous les champs soient valides pour pouvoir cliquer sur le bouton « Ajouter ».

****

Figure 12 : Diagramme de séquence "Ajouter une Formation"

1. **Diagramme de séquence « Modifier une formation »**

Pour pouvoir modifier les détails d’une formation, l’admin doit tout d’abord cliquer sur la formation souhaitée pour afficher les détails, après cliquer sur le bouton modifier et une interface s’affiche.

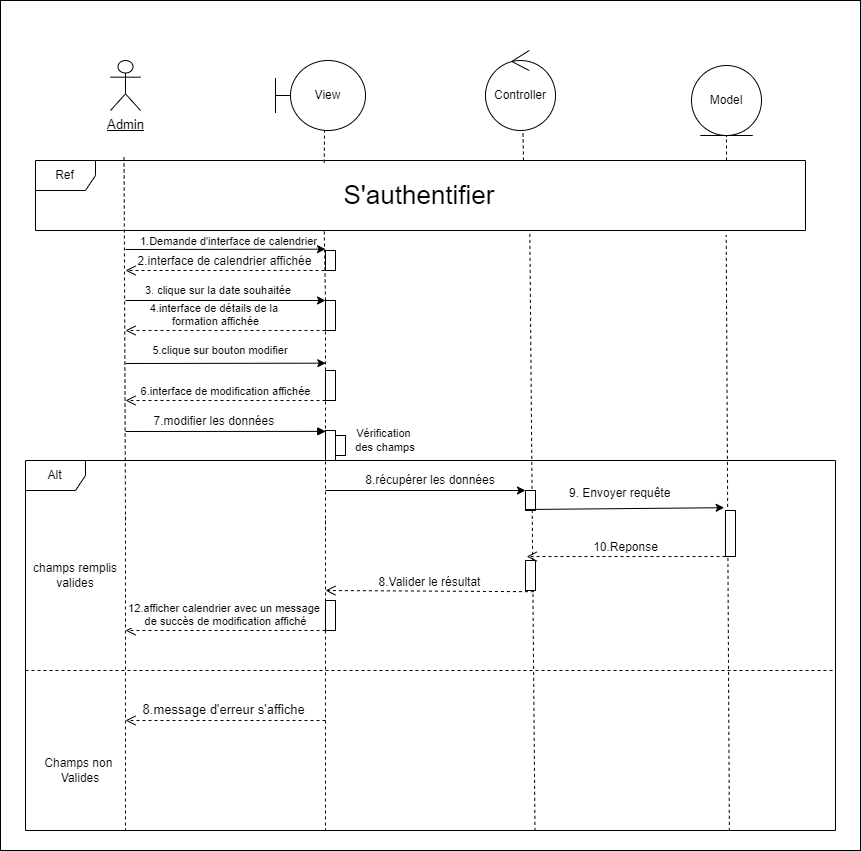
****

Figure 13 : diagramme de séquence "Modifier une formation"

1. **Diagramme de Classes**
2. **Définition :**

Le diagramme de classes fait partie des types de diagrammes UML. Cette représentation est axée sur le concept des classes et des associations entre elles indépendamment de langage de programmation utilisé. Nous élaborons un diagramme de classes pour simplifier l’interaction des objets d’un système que nous sommes en train de modéliser.

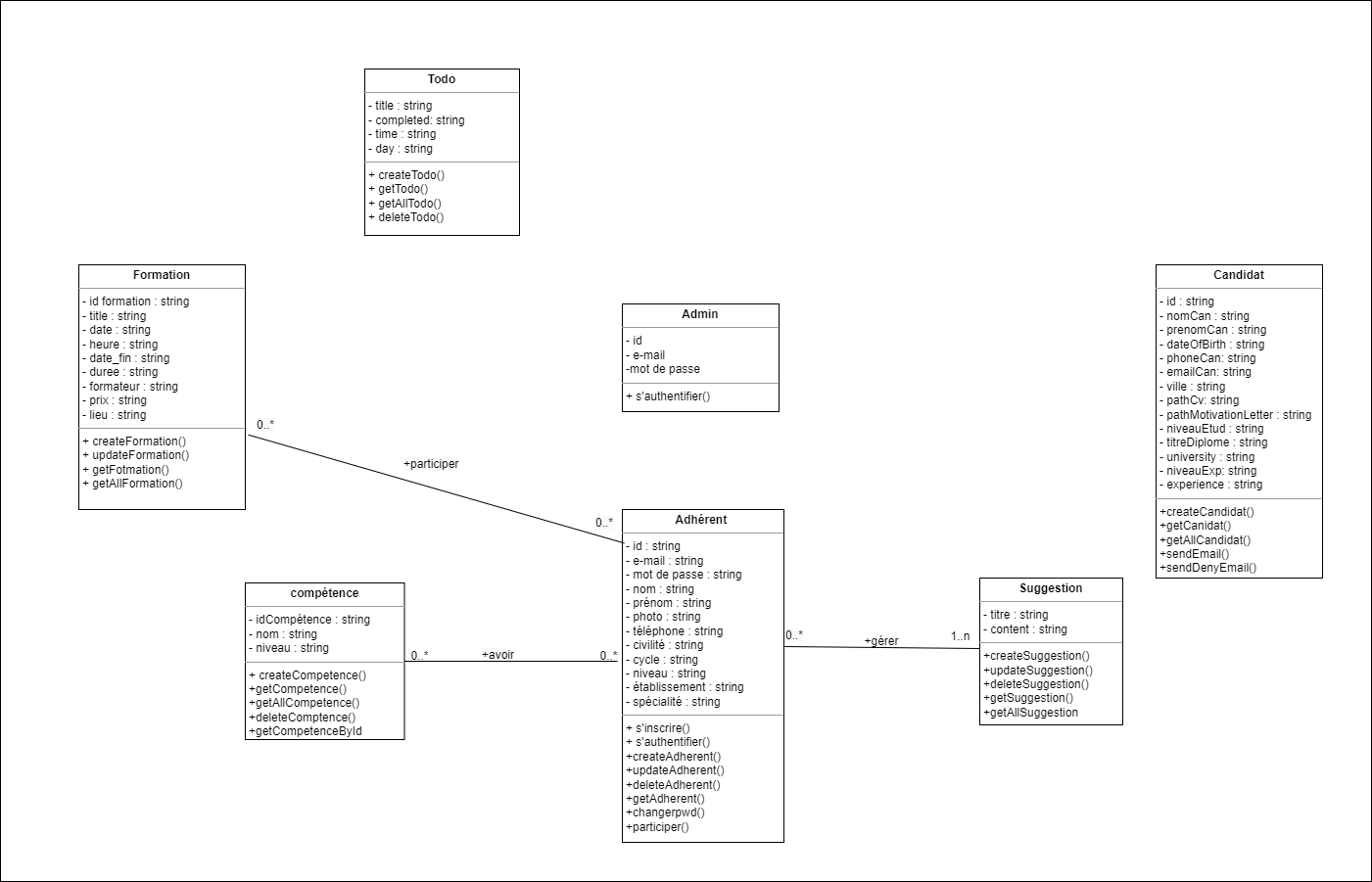


Figure 14 : Diagramme de classe

1. **Dictionnaire des données**

* **Admin** (Email, Password, Nom, Prenom)

|  |  |  |
| --- | --- | --- |
| Propriétés | | Type |
| Attribut | Désignation |
| Email | L’email est l’identifiant de l’admin | String |
| Nom et prénom | Le nom de l’admin | String |
| Password | Le mot de passe de l’admin | String |

Tableau 10 : Admin

* **Adhérent** (Email, Password, CIN, Civilité, Nom et Prénom, Téléphone,

Profession, Cycle, Etablissement, Niveau, Spécialité).

|  |  |  |
| --- | --- | --- |
| Propriétés | | Type |
| Attribut | Désignation |
| Email | L’email et l’identifiant de l’adhèrent | String |
| Password | Le mot de passe de l’adhèrent | String |
| CIN | L’identifiant de l’adhèrent | String |
| Civilité | La civilité de l’adhèrent | String |
| Nom et Prénom | Le nom et le prénom de l’adhérent | String |
| Téléphone | Le numéro du téléphone de l’adhérent | String |
| Profession | La situation professionnelle de l’adhérent | String |
| Cycle | Le cycle d’étude de l’adhérent | String |
| Etablissement | L’université de l’adhérent | String |
| Niveau | Le niveau d’étude de l’adhérent | String |
| Spécialité | La spécialité d’étude de l’adhérent | String |
| Photo | La photo de l’adhérent | String |

Tableau 11 : Adhérent

* **Formation** (Email, Password, CIN, Civilité, Nom et Prénom, Téléphone,

Profession, Cycle, Etablissement, Niveau, Spécialité).

|  |  |  |
| --- | --- | --- |
| Propriétés | | Type |
| Attribut | Désignation |
| Title | Le nom de la formation | String |
| Date | Date début de la formation | String |
| Heure | L’heure de la formation | String |
| Date\_fin | Date fine de la formation | String |
| Duree | La durée de la formation | String |
| Formateur | Le formateur qui va assurer la formation | String |
| Prix | Le prix de la formation | String |
| Lieu | Le lieu où va dérouler la formation | String |

Tableau 12 : Formation

* **Candidat** (nomCan, prenomCan, dateOfBirth, phoneCan, emailCan, ville,

pathCv, pathMotivationLetter, niveauEtud, titreDiplome, university,

niveauExp, experience)

|  |  |  |
| --- | --- | --- |
| Propriétés | | Type |
| Attribut | Désignation |
| nomCan | Le nom du candidat | String |
| prenomCan | Le prénom du candidat | String |
| dateOfBirth | La date de naissance du candidat | String |
| phoneCan | Le numéro du téléphone du candidat | String |
| emailCan | L’email du candidat | String |
| ville | La ville du candidat | String |
| pathCv | Le CV du candidat | String |
| pathMotivationLetter | La lettre de motivation du candidat | String |
| niveauEtud | Niveau d’étude du candidat | String |
| titreDiplome | Titre du diplôme du candidat | String |
| university | L’université du candidat | String |
| niveauExp | Le niveau d’experience du candidat | String |
| experience | L’experience du candidat | String |

Tableau 13 : Candidat

* **Entreprise** (emailEnt, password, nomEnt, nomResp, telEnt, adresse).

|  |  |  |
| --- | --- | --- |
| Propriétés | | Type |
| Attribut | Désignation |
| emailEnt | L’email de l’entreprise | String |
| password | Le mot de passe de l’entreprise | String |
| nomEnt | Le nom de l’entreprise | String |
| nomResp | Le nom du responsable du l’entreprise | String |
| telEnt | Le numéro de téléphone de l’entreprise | String |
| adresse | L’adresse du l’entreprise | String |

Tableau 14 : Entreprise

* **Suggestion** (titre, content).

|  |  |  |
| --- | --- | --- |
| Propriétés | | Type |
| Attribut | Désignation |
| titre | Le titre de la formation proposée | String |
| content | Le contenu de la formation proposé | String |

Tableau 15 : Suggestion

* **Competence** (nom, niveau).

|  |  |  |
| --- | --- | --- |
| Propriétés | | Type |
| Attribut | Désignation |
| nom | Le nom de compétence | String |
| niveau | Le niveau de compétence | String |

Tableau 16 : Competence

* **Todo** (title, time, day).

|  |  |  |
| --- | --- | --- |
| Propriétés | | Type |
| Attribut | Désignation |
| title | Titre de tache à faire | String |
| Time | L’heure de cette tache | String |
| Day |  | String |

Tableau 17 : Todo

**Conclusion :**

Dans ce chapitre, nous avons présenté l’étude conceptuelle de notre système, ce qui nous permet de nous rendre compte que l’activité de conception a explicité l'exécution de notre système. L'activité de conception nous emporte à l’activité d’implémentation.

Dans le chapitre suivant, nous entamerons la phase de l’implémentation.

**Chapitre IV : Mise en œuvre et réalisation**

**Chapitre IV :**

**Mise en Œuvre et Réalisation**

**Introduction :**

Ce Chapitre présente le dernier volet de notre rapport, à savoir l’implémentation de l’application. Nous commençons par une description de notre environnement du travail logiciel et matériel ainsi que les technologies utilisées pour réussir le développement de notre application. Ensuite nous expliquons les différentes interfaces de notre application.

1. **Environnement de Travail**

Dans cette partie, nous citons les outils matériels et logiciels nécessaires pour le développement de notre application web.

1. **Environnement Matériel**

L’environnement matériel sur lequel nous avons travaillé est constitué d’un ordinateur portable ASUS avec les caractéristiques suivantes :

|  |  |
| --- | --- |
| PC-Portable / RAM : | ASUS / 8Go |
| Disque Dur : | SSD (512 Go) |
| Processeur : | Intel Core i5- 7th Gen |
| Système d’exploitation : | Windows 10 Pro |

Tableau 18 : Configuration du Matériels utilisés

1. **Environnement Logiciel**

Cette partie présente les différents logiciels, environnement de conception, environnements de base de données utilisés dans l’élaboration de notre projet.

* 1. **Environnement de conception UML**
* **Draw.io :**

Draw.io **[4]** : ​​est un outil de conception UML innovant dédié à rendre la modélisation de système simple et rentable en traduisant les conditions et les détails du système en solutions graphiques concrètes et détaillées. Cet outil fournit un ensemble complet d'outils logiciels pour la modélisation de tous les types de diagrammes, la planification, la modélisation de classe, etc. Nous avons utilisé cet outil dans le développement de diagrammes de cas d'utilisation, de diagrammes de classes et de diagrammes de séquence.

****

Figure 15 : Logo de Draw.io

* 1. **Environnement de développement intégré**
* **Visual Studio Code**

Visual Studio Code est un éditeur de code extensible, open source et gratuit développé par Microsoft pour Windows, Linux et MacOs. Les fonctionnalités incluent la prise en charge du débogage, la coloration syntaxique, la complétion intelligente du code, les extraits de code, la refactorisation du code et Git intégré. Il peut être utilisé avec plusieurs langages de programmation, comme Java, JavaScript, Go, NodeJS et C++ **[5].** Visual Studio Code nous fournit tous les outils dont nous avons besoin pour développer le frontend et le backend de notre application web. De plus, cela nous permet de mettre notre code source sur Git.

****

Figure 16 : Logo Visual Studio Code

* 1. **Outil d’administration de la base de données**
* **Postman**

Postman est un excellent environnement pour développer des API et des services Web RESTFUL. Il permet de construire et d'exécuter des requêtes http, de les stocker dans l'historique directement depuis l'interface graphique pour pouvoir les rejouer, mais surtout de les organiser en collections **[6].**



Figure 17 : Logo Postman

* **MongoDbCompass**

MongoDB est un système de gestion de base de données de documents NoSQL sans schéma. Cela signifie que vous pouvez y stocker des documents JSON et que la structure de ces documents peut varier car elle n'est pas appliquée comme les bases de données SQL. C'est l'un des avantages de l'utilisation de NoSQL car il accélère le développement des applications et réduit la complexité des déploiements **[7].**

****

Figure 18 : Logo MongoDB

* **Mongoose**

Mongoose est une bibliothèque de modélisation de données d'objets (ODM) pour MongoDB et Node.js. Il gère les relations entre les données, fournit une validation de schéma et est utilisé pour traduire entre les objets dans le code et la représentation de ces objets dans MongoDB **[8].**

****

Figure 19 : Logo Mongoose

1. **Développement**
2. **Présentation des Technologies**

On **a** développé notre application avec **MEAN** qui est une pile logicielle JavaScript gratuite et à code source libre. L’acronyme « MEAN » signifie MongoDB, Express.js, Angular, et Node.js. L’un de principaux avantages de la pile MEAN est l’utilisation d’un langage unique java script, qui s’exécutent à tous les niveaux de l’application qui est une approche efficace et moderne de développement web. La figure montre l’interaction entre les différentes technologies de « **MEAN STACK** ». **[9]**

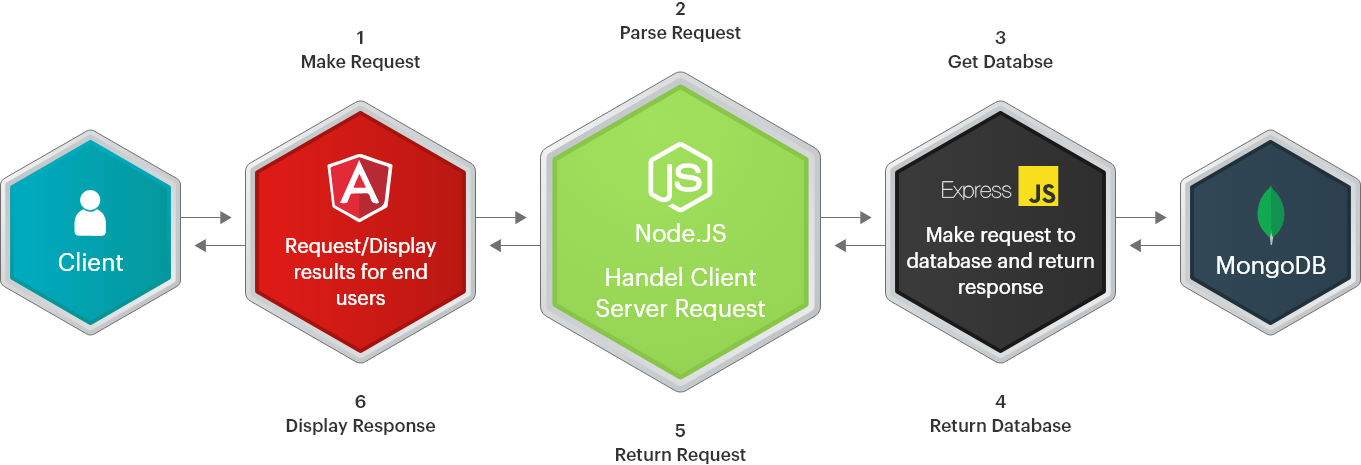


Figure 20 : Interaction entre les composants de la pile "MEAN"

1. **Les technologies utilisées** 
   1. **Définition d’un Framework**

Un Framework (ou infrastructure logicielle en français) désigne en programmation informatique un ensemble d'outils et de composants logiciels à la base d'un logiciel ou d'une application.

* **Couche Présentation : La technologie Angular**
* **Angular**

Angular est un Framework JavaScript développé par Google et reconnu comme l'un des Framework les plus complets. En effet, il embarque de nombreuses fonctionnalités en lui-même, permettant de développer des applications web, même les plus complexes. Cependant, en raison de son évolution, Angular convient également à tout projet d'application de bureau ou mobile. Rapide à déployer et puissante, elle sait convaincre les grandes et les petites entreprises **[10].**

****

Figure 21 : Logo Angular

* **Vue & Contrôleur** : où la vue est un fichier HTML et un fichier CSS, et le contrôleur est une classe Type Script qui ouvre les traitements à faire, c’est le lieu entre la vue et les services.
* **Service** : c’est composant qui gère les requêtes http, afin d’assurer l’échange des données JSON entre les services Web et le contrôleur.
* **Pipe** : c’est un composant qui peut être utilisé dans les balises html personnaliser l’affichage des données il prend en entrée des données, les transforme et retourne en sortie dans un format désiré.
* **Couche métier d’accès aux données : La technologie NodeJS**
* **Node.JS**

Node.js est une plate-forme open source basée sur le moteur JavaScript V8 de Chrome, permettant de développer des applications à l'aide de JavaScript. En tant que tel, Node.js est synonyme de JavaScript omniprésent, un paradigme qui unifie le développement d'applications Web autour d'un seul langage de programmation, plutôt que des langages séparés pour les scripts côté serveur et côté client. Il se distingue des autres plates-formes par une approche non bloquante qui permet l'exécution asynchrone des entrées/sorties (I/O) **[11].**

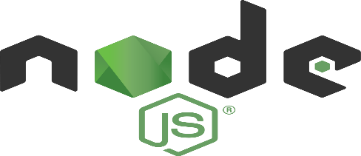


Figure 22 : Logo Node.js

* **Express.js :**

Express.jsest un Framework pour construire des applications web basées sur Node.js. C'est de fait le Framework standard pour le développement de serveur en Node.js. L'auteur original, TJ Holowaychuck, le décrit comme un serveur inspiré de Sinatra 4 dans le sens qu'il est relativement minimaliste tout en permettant d'étendre ses fonctionnalités via des plugins. L’utilisation du Node JS impose automatiquement l’utilisation de l’express.JS. Express.JS est un Framework pour construire des applications web basées sur Node.js, c'est en fait le Framework standard pour le développement du serveur en Node.js **[12].**

****

Figure 23 : Logo Express.js

* **Git :**

Git est un logiciel de gestion de sources et de contrôle de versions. Ce type de programmes a plusieurs fonctions, notamment :

* Permettre le retour à une version ultérieure quelconque.
* Garder un historique des modifications avec leur nature, leur date, leur auteur etc.
* Permettre à des utilisateurs distincts et souvent distants de travailler ensemble sur les mêmes fichiers **[13].**



Figure 24 : Logo Git

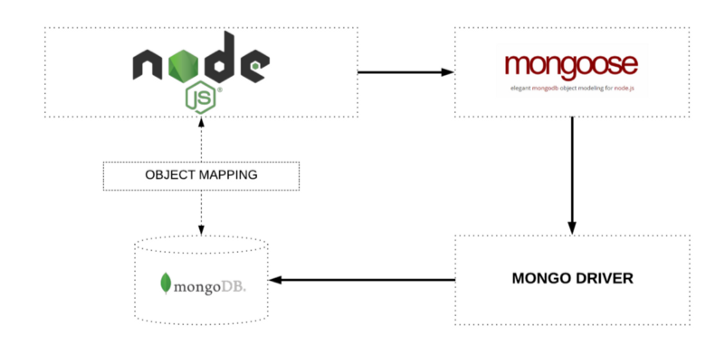


Figure 25 : Mappage d'objets entre Node et MongoDB géré via Mongoose

1. **Les langages utilisés**

* **HTML 5**

HTML est l’acronyme de « *Hypertexte Markup Language 5* » est la dernière révision majeure du HTML (format de données conçu pour représenter les pages web) **[14].**

****

Figure 26 : Logo HTML

* **CSS 3**

CSS est l’acronyme de « Cascading Style Sheets » ce qui signifie « feuille de style en cascade ». C’est un langage informatique qui décrit la présentation des documents HTML et XML **[15].**

****

Figure 27 : Logo CSS

* **Bootstrap**

Bootstrap est une collection d'outils utiles à la création du design (graphisme, animation et interactions avec la page dans le navigateur, etc.) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. C'est l'un des projets les plus populaires sur la plate-forme de gestion de développement GitHub **[16].**

****

Figure 28 : Logo Bootstrap

1. **Scénario Complet d’utilisation de la Solution** 
   1. **Les interfaces Graphiques**

La Conception des interfaces de l’application est considérée comme une étape fondamentale puisque toutes les interactions avec le cœur de l’application passent à travers ces interfaces. L’ensemble de ces interfaces représentera un scénario complet pour la solution

* 1. **Page d’accueil**

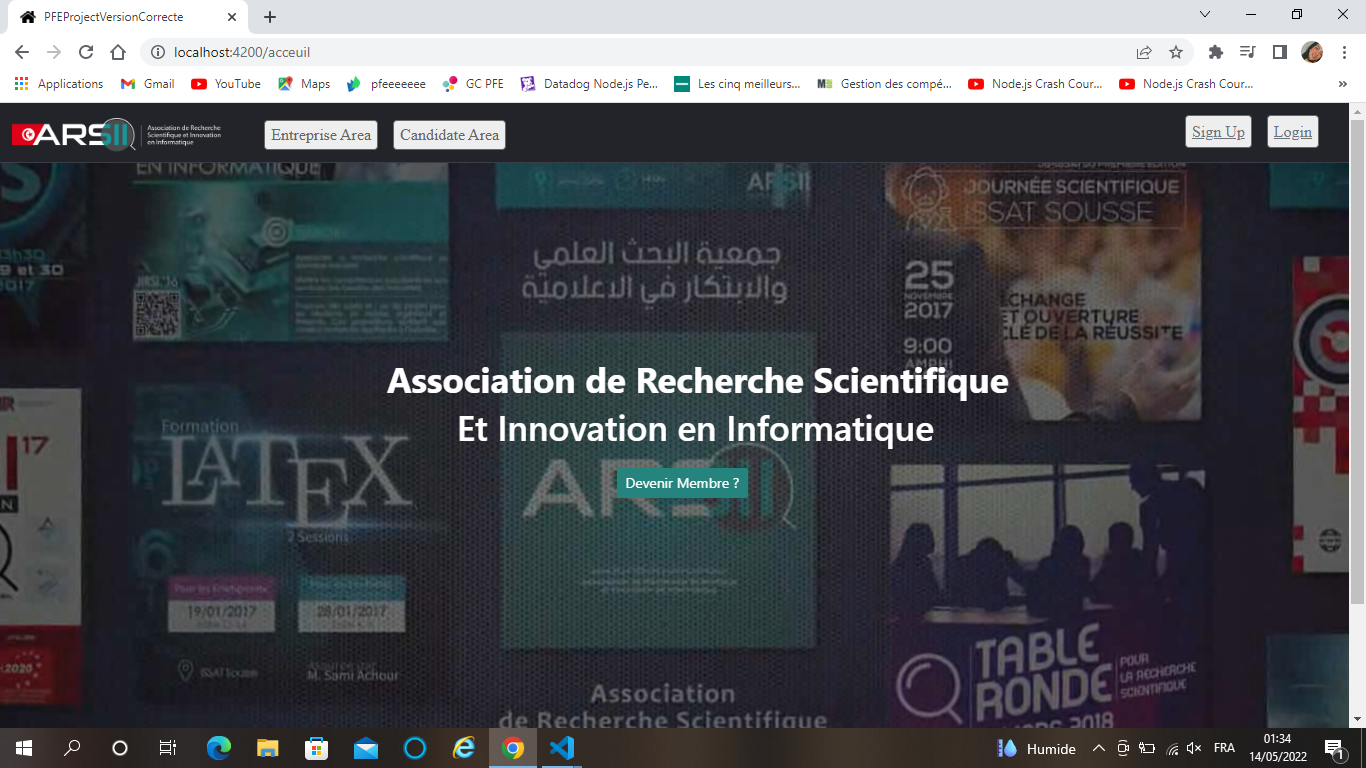
****

Figure 29 : Interface D'accueil

* 1. **Page D’inscription :**

Pour un premier accès à notre application, l’adhérent doit s’inscrire à travers ce formulaire

****

Figure 30 : Interface d’inscription (adhérent)

Et voici la page d’inscription spécifique de l’entreprise :

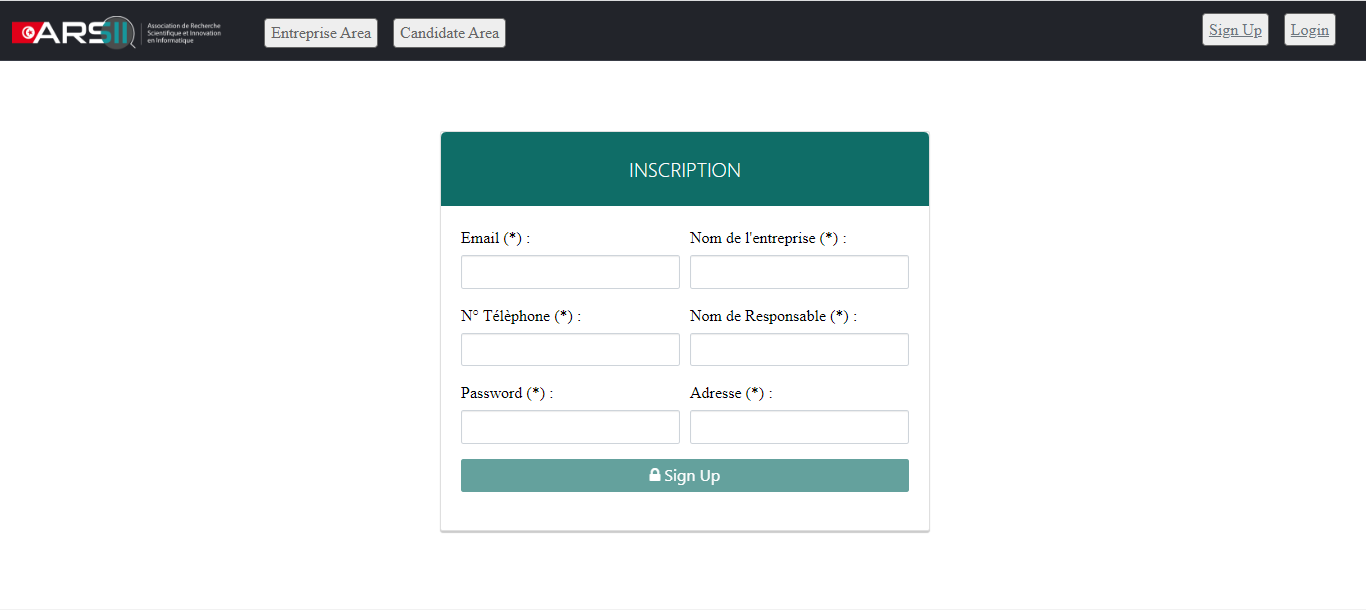


Figure 31 : Interface d'inscription (Entreprise)

* 1. **Page de Connexion :**

Cette interface montre la page de connexion. Si l’utilisateur a créé un compte, il se connecte en mettant son e-mail et son mot de passe.

**L’admin,L’adhérent et l’entreprise se connecte a travers la seule page de connexion.**

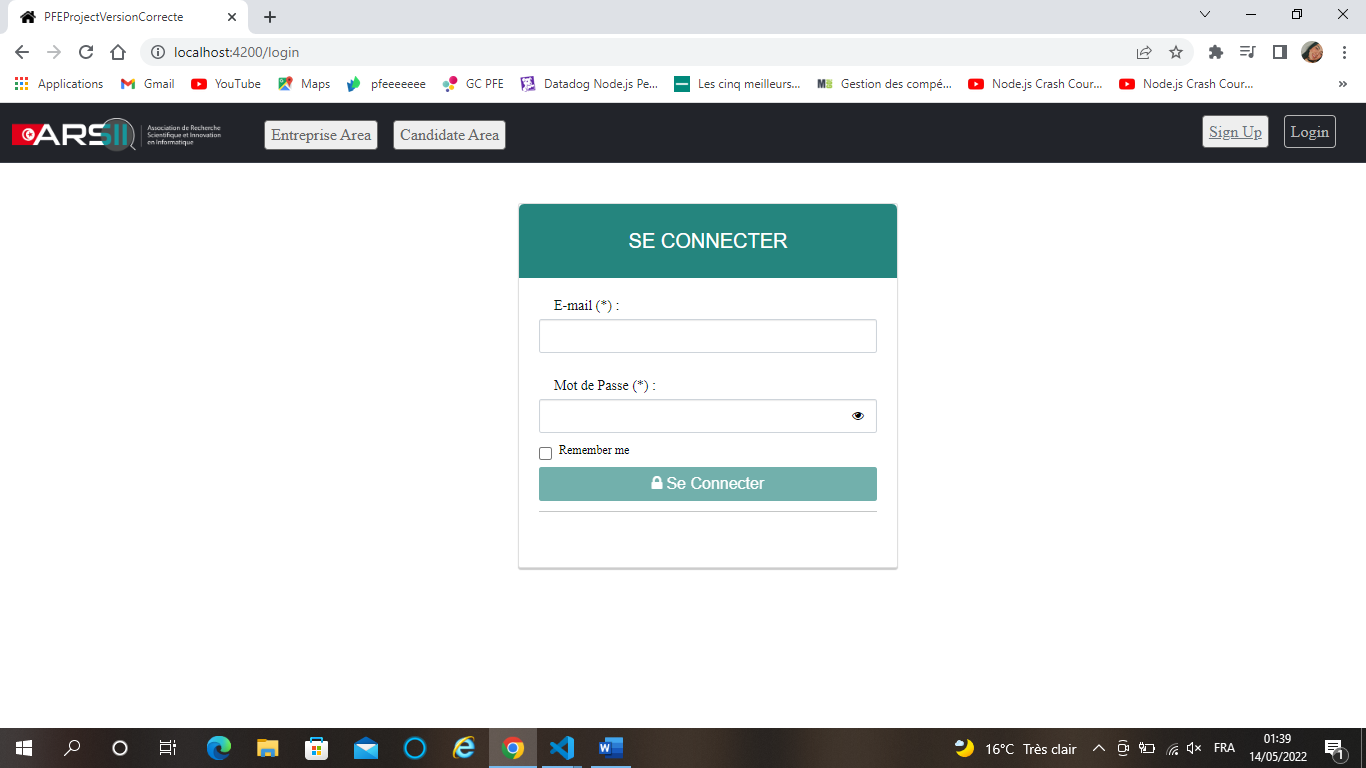
****

Figure 32 : Interface d’authentification

* 1. **Espace Admin :**

Une fois l'administrateur est authentifié, il aura accès aux différentes fonctionnalités de l'application. Il possède alors tous les droits d'accès.

Cette interface est l’espace interne de l’admin, à travers elle l’admin peut visualiser les courbes, gérer les adhérents, les formations etc.

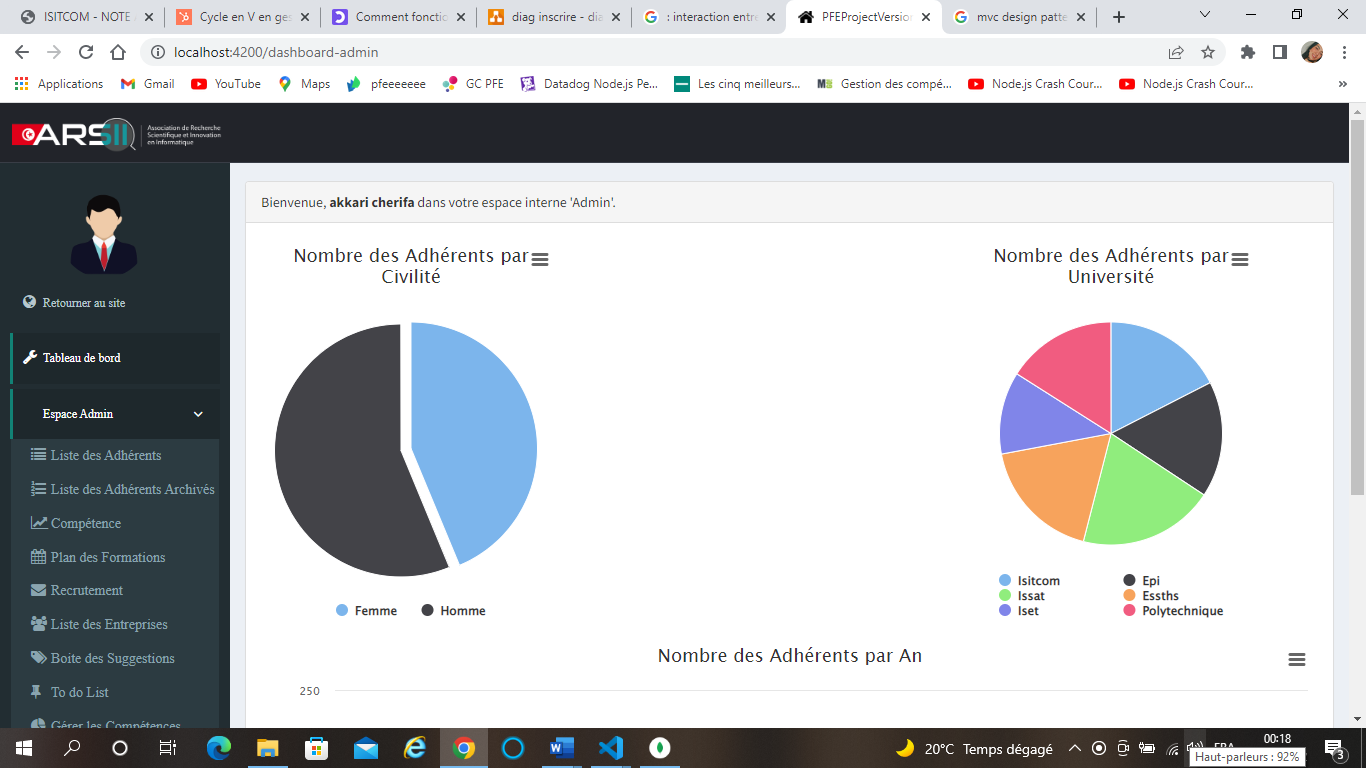
 L’interface suivante permet à l’admin la gestion des formations :

Figure 33 : Dashboard Admin

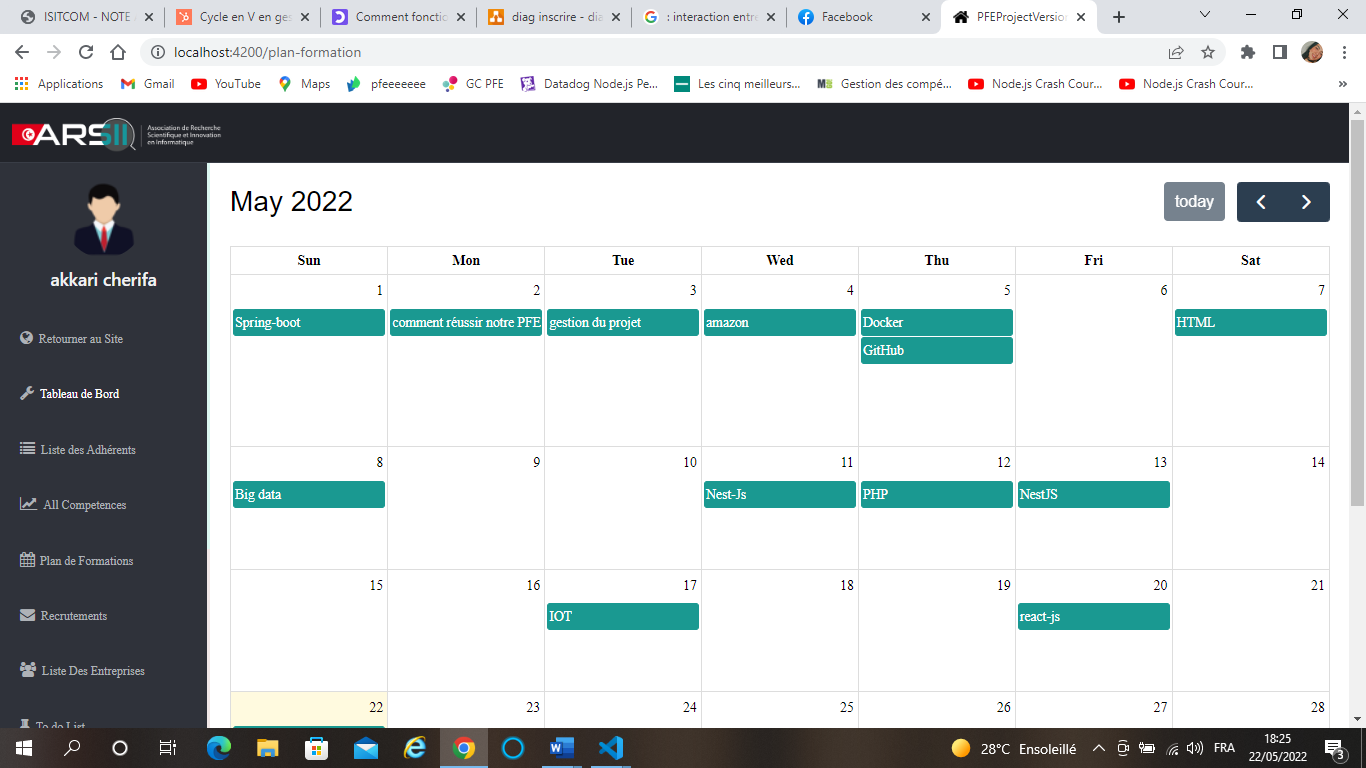


Figure 34 : Calendrier

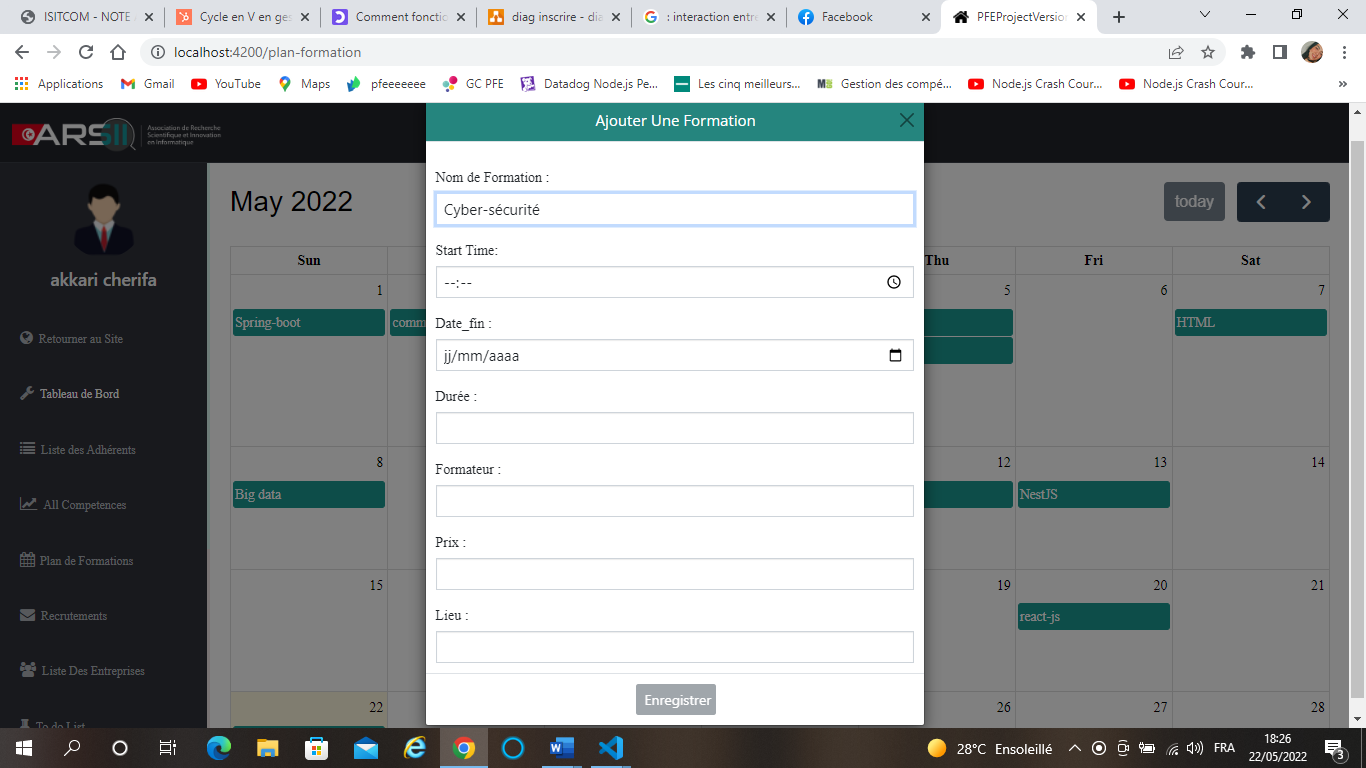


Figure 35 : Interface d'ajout d'une Formation

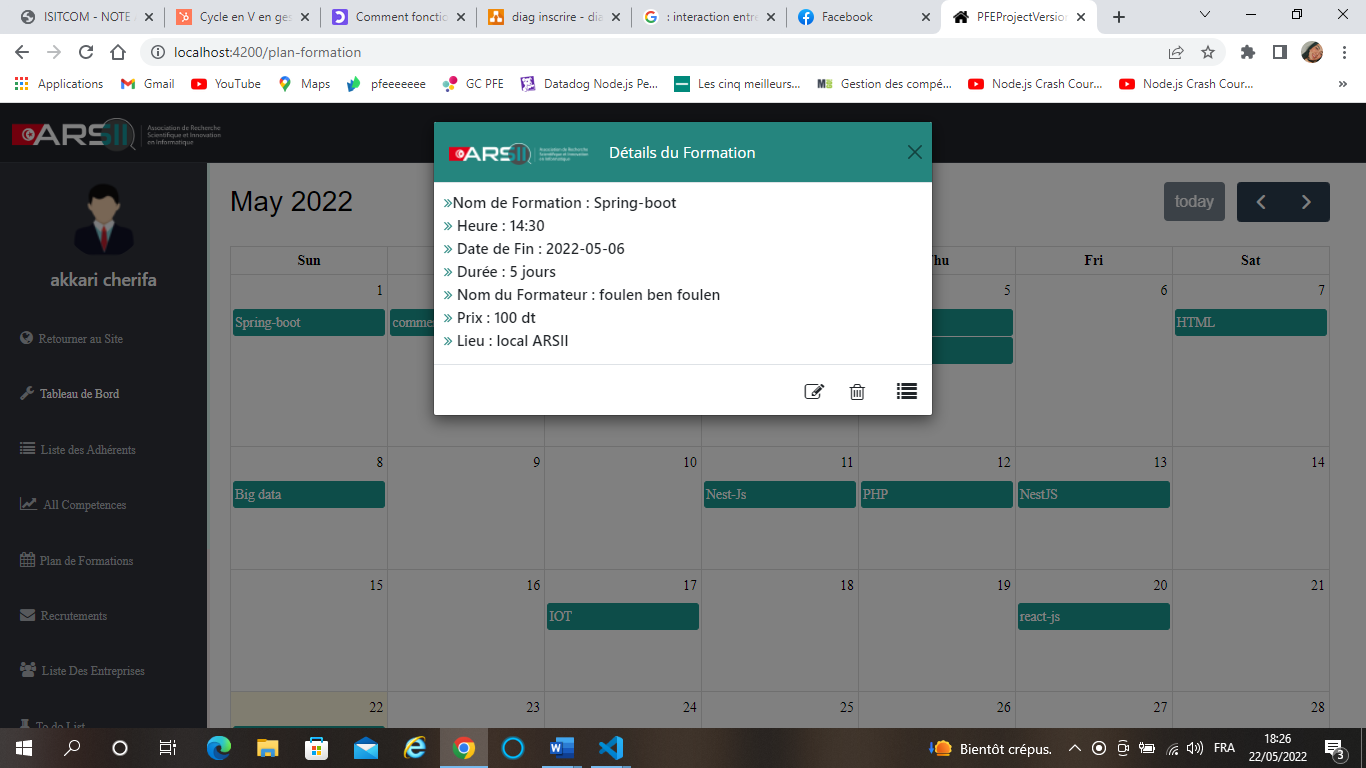


Figure 36 : Interface de détails d'une formation

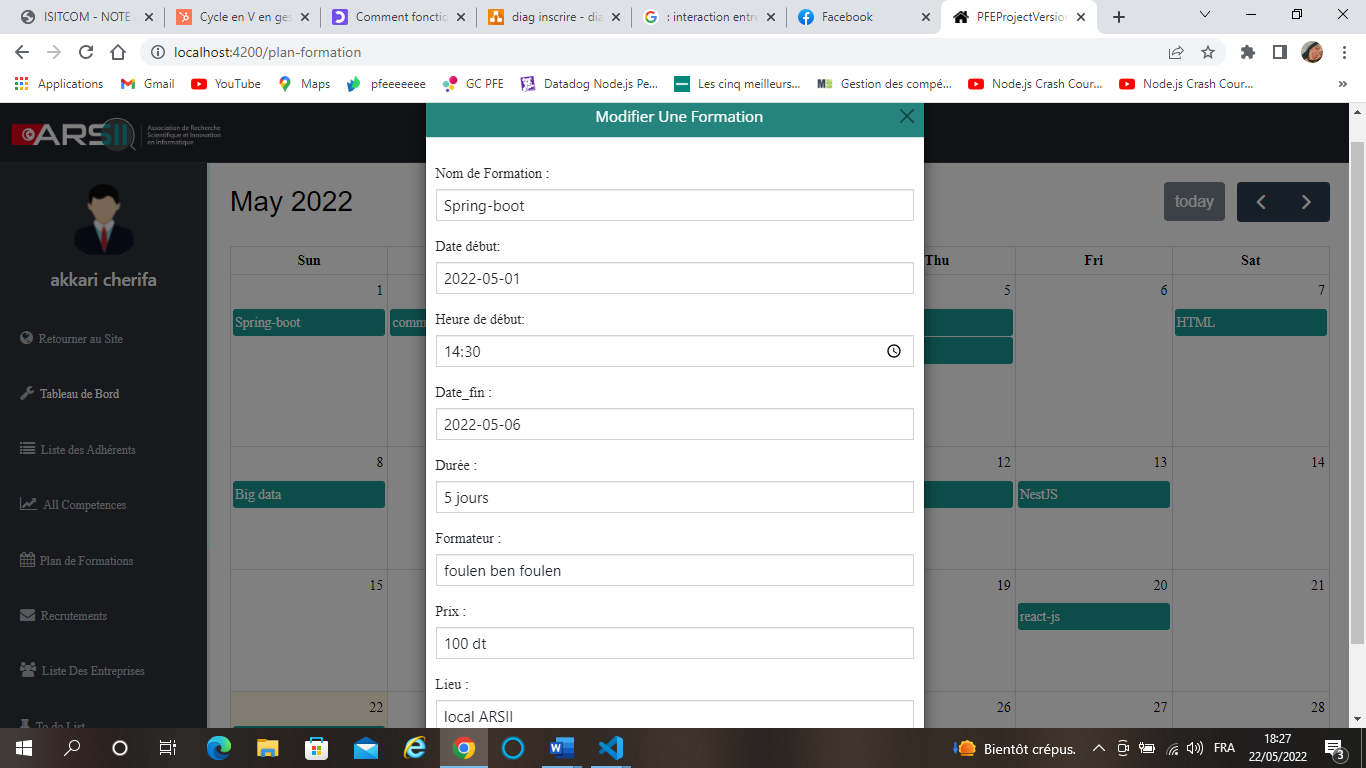


Figure 37 : Interface de modification d'une formation

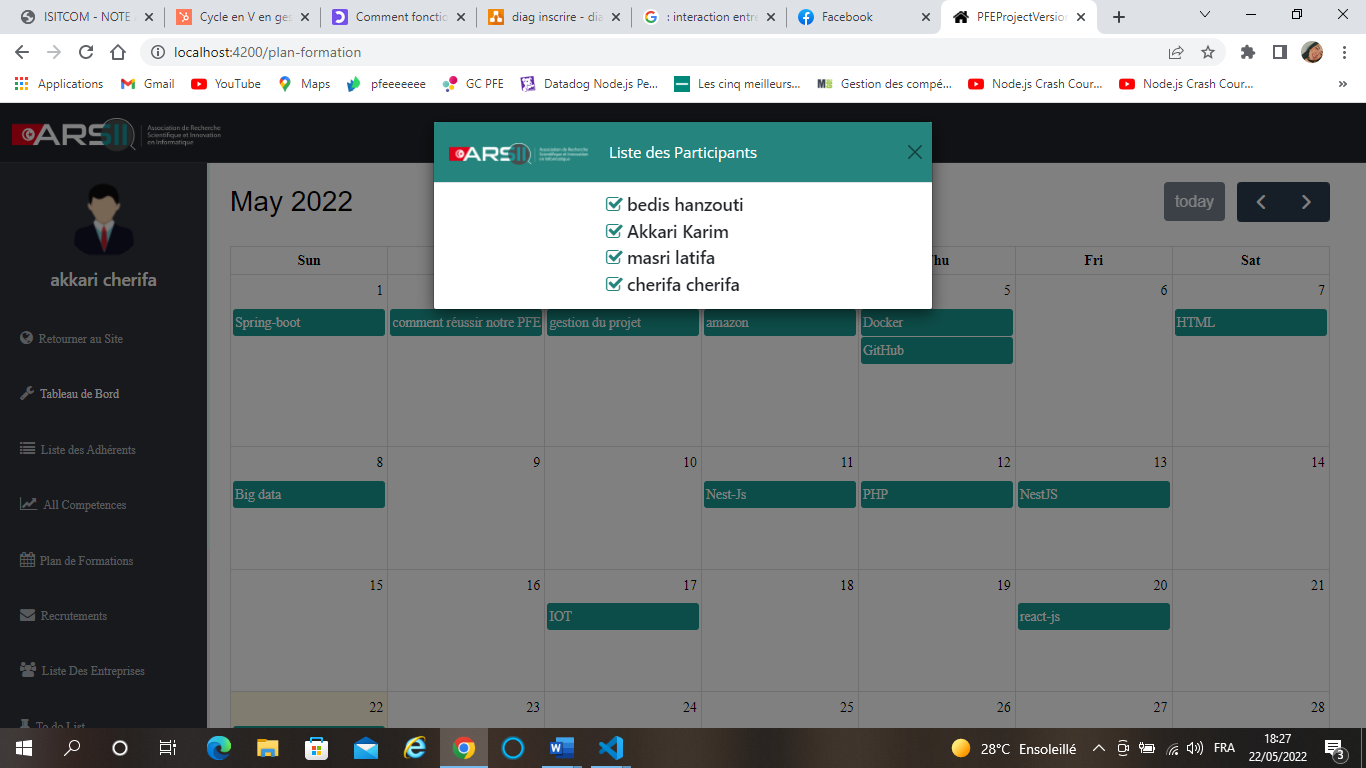


Figure 38 : Interface de liste des participants d'une formation

L’administrateur peut visualiser la liste des formations proposés par les adhérents pour organiser un plan de formations selon leur besoin.

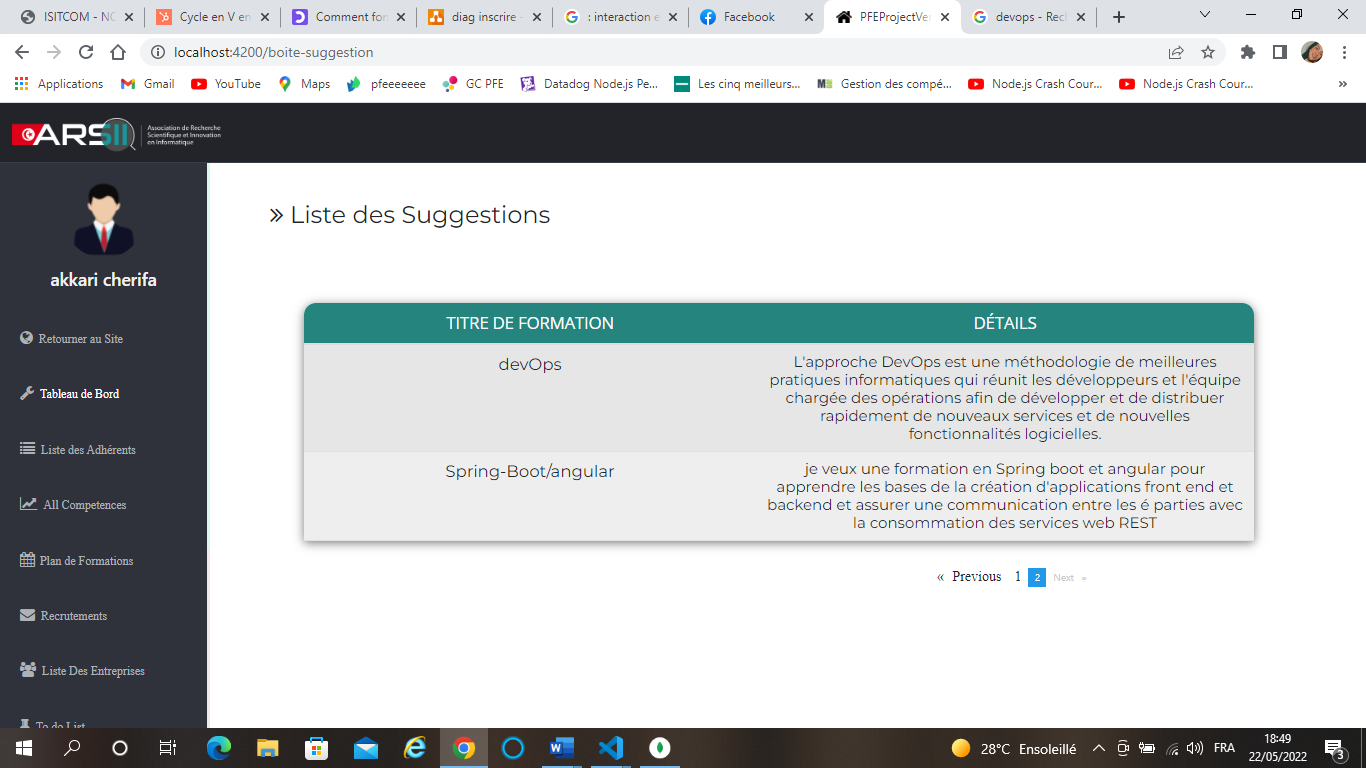


Figure 39 : interface de Liste des propositions

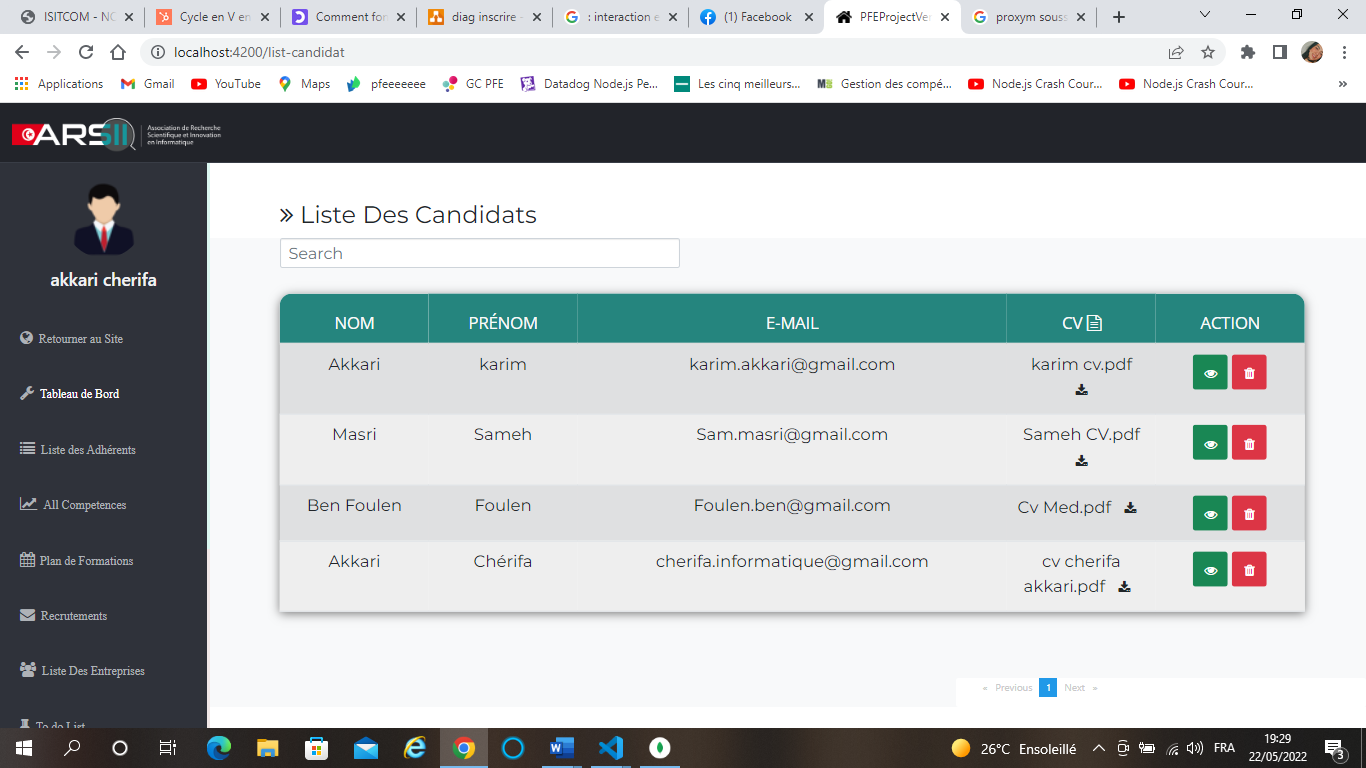
Depuis cette interface, l’administrateur peut consulter une liste de candidats ayant postulé à des postes de formateur.

Figure 40 : Liste des Candidats

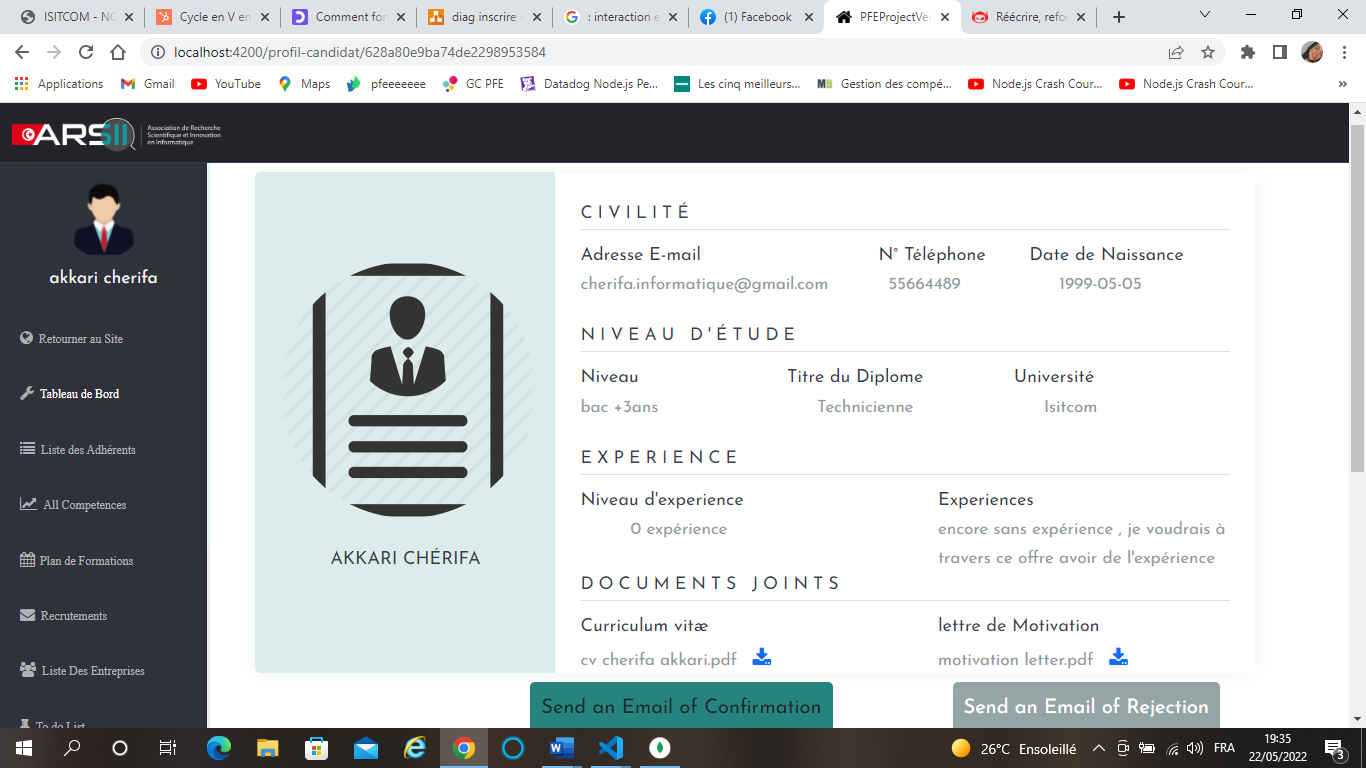
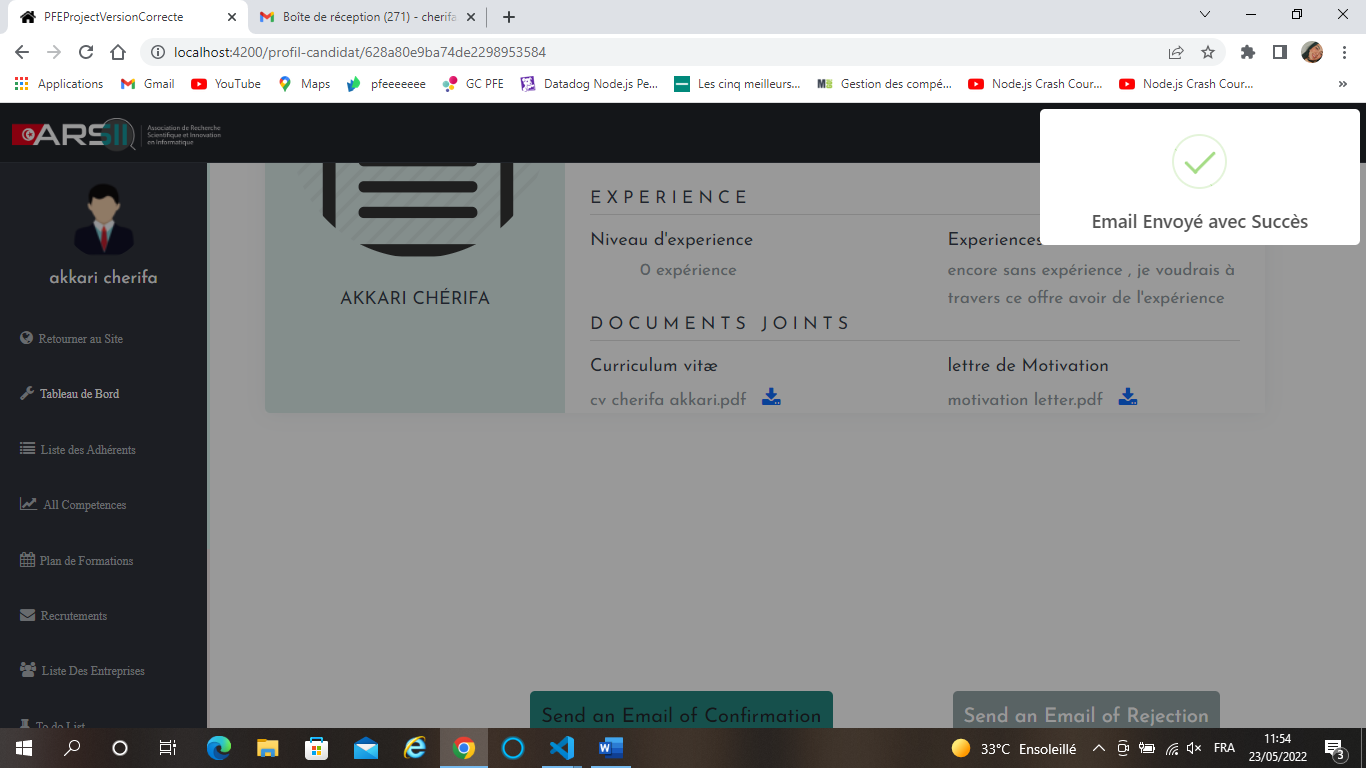
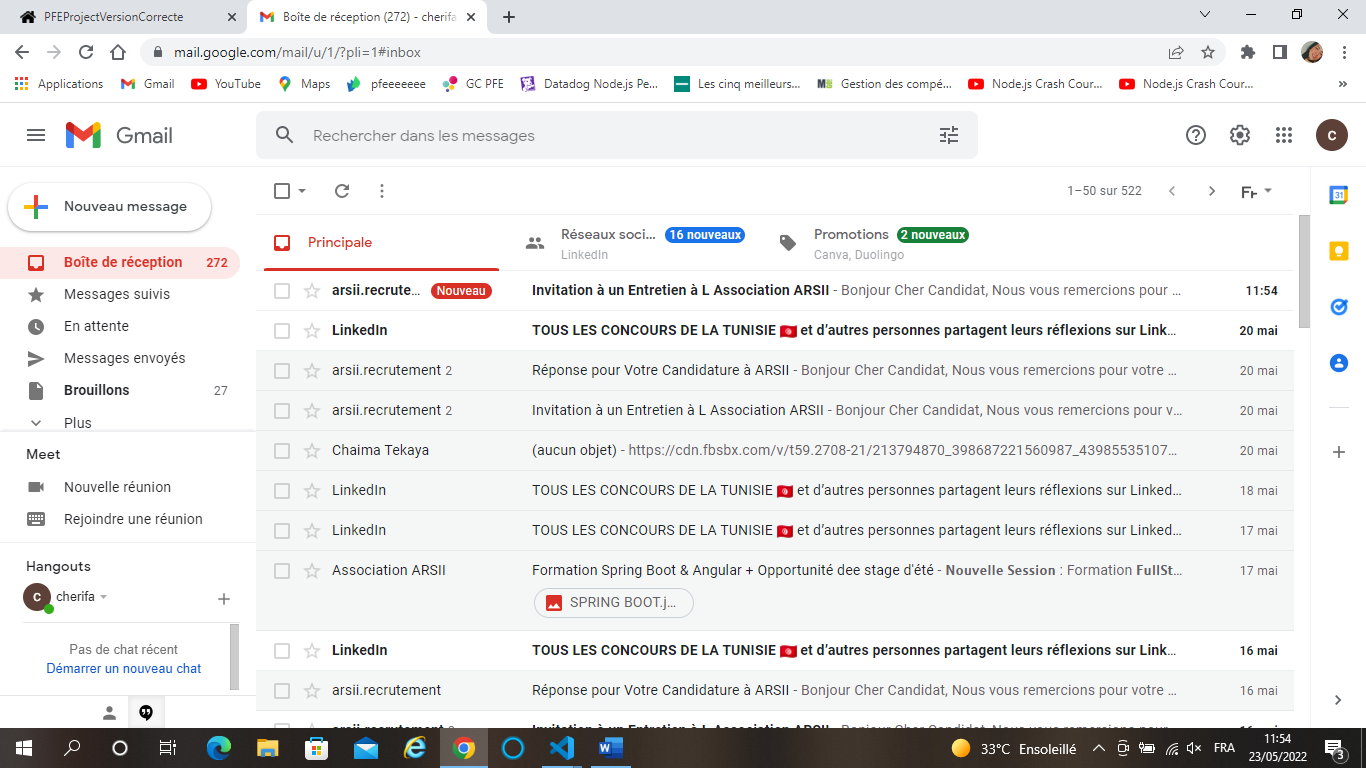
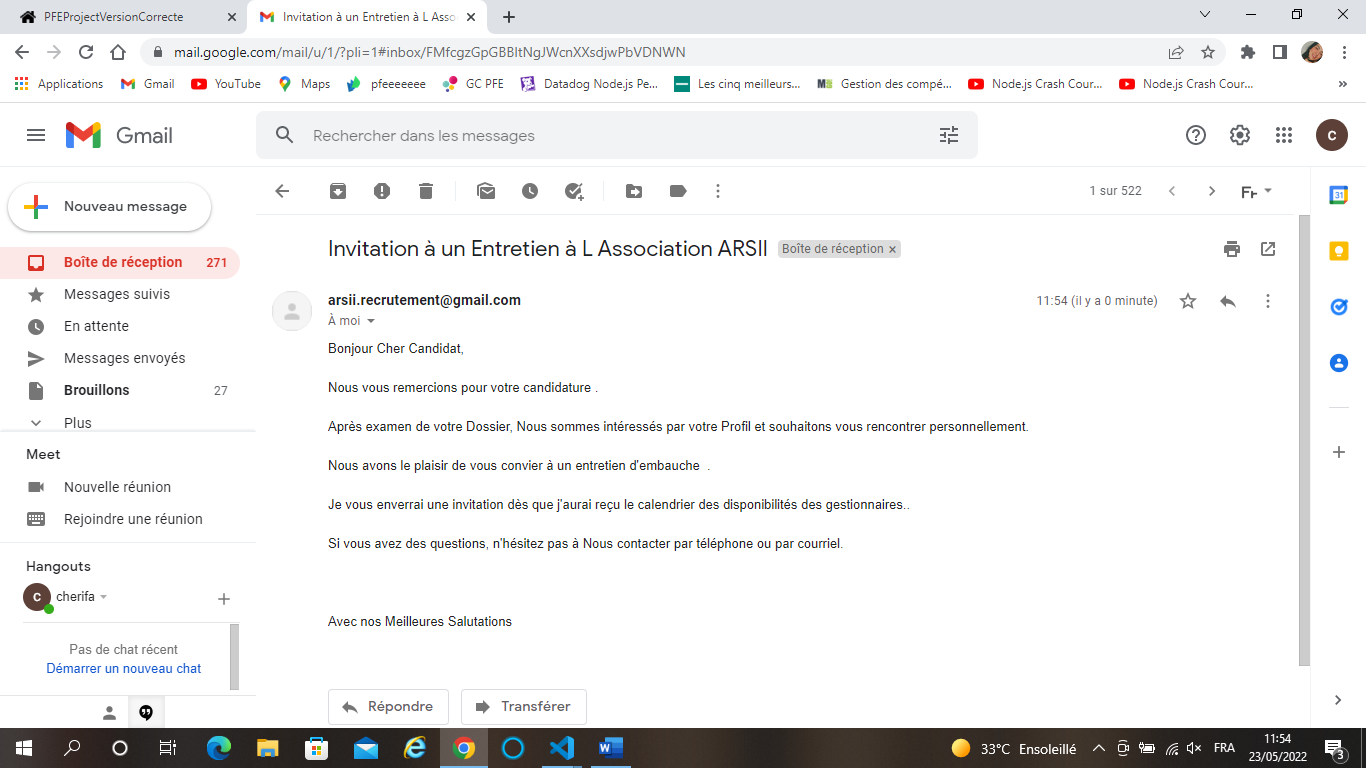


Figure 41 : Interface de Profil d'un Candidat







**Conclusion générale et Perspectives**

Il est à rappeler que notre projet a été réalisé au sein de l’association ARSII dans le cadre de notre stage de fin d’étude. Il nous a permis de revoir et de suivre avec beaucoup d’attention, les méthodes fondamentales de modélisation et de conception des systèmes d’informations.

Notre projet consiste à réaliser une application web qui aura pour mission de faciliter l’accès des adhérents à leur espace interne de l’association à travers des tableaux de bords et d’assurer une organisation optimale des formations.

Notre application a été implémentée également dans le cadre de l’architecture Model-View-Controller (MVC). Notre application nous a permis de toucher les technologies Node.JS et Angular qui sont utilisées actuellement sur le marché.

Pour ce faire, nous avons commencé en situant le sujet dans son contexte général en réalisant une étude des solutions existantes et ensuite en détaillant les objectifs, les besoins fonctionnels et non fonctionnels de la solution proposée, la méthodologie adoptée, la planification du projet et le diagramme de cas d’utilisation global. Le deuxième chapitre a été dédié à la conception de notre solution en élaborant la modélisation UML sous forme de diagrammes de séquences et le diagramme de classe des entités. En ce qui concerne le troisième chapitre, il a été consacré à la présentation des architectures physique et logique appliquées, les environnements de travail, les Framework et technologies adoptées ainsi que les interfaces du scénario complet de la solution.

Comme perspectives à notre travail, nous pouvons continuer à améliorer le système en implémentant des tests d’évaluation de chaque compétence qui aide à donner le niveau réel de l’adhérent pour donner une meilleure crédibilité. Des Quiz en ligne avec des questions bien déterminé peuvent aider l’adhérent à avoir le réel niveau de ces compétences.

Sur le plan personnel, ce stage été une véritable occasion pour épanouir nos connaissances dans le domaine de développement web. Le fait que nous ayons pu développer notre indépendance, travailler en toute autonomie, développer notre savoir être, assurer de plus en plus de responsabilité et travailler sur une application basée sur des technologies web récentes a été une expérience vraiment enrichissante et motivante.

**Webographie**

**[1] Kanban + Cascade : [21/04]**

Disponible sur : <https://www.maestroprojet.fr/index.php/2020/11/03/gestion-de-projet-methodologie/?fbclid=IwAR1CkCO5LIeO57jW-3R6kQIpZI5RtT1tgWbg8MDejE4SLD1ZAnq4XysmkUo>

**[2] cycle en V : [02/03]**

Disponible sur : <https://blog.hubspot.fr/marketing/cycle-en-v#:~:text=La%20méthode%20du%20cycle%20en%20V%20implique%20toutes%20les%20étapes,la%20réalisation%20et%20la%20validation>.

**[3] le modèle MVC : [12/03]**

Disponible sur :

<https://www.irif.fr/~carton/Enseignement/InterfacesGraphiques/Cours/Swing/mvc.html>

**[4] draw.io : [05/05]**

Disponible sur : <https://www.tice-education.fr/tous-les-articles-er-ressources/articles-internet/819-draw-io-un-outil-pour-dessiner-des-diagrammes-en-ligne>

**[5] Visual Studio Code : [05/05]**

Disponible sur : <https://blog.sodifrance.fr/visual-studio-code-quesaco>

**[6] Postman : [05/05]**

Disponible sur : <https://latavernedutesteur.fr/2020/03/06/outil-de-test-tester-vos-api-avec-postman/>

**[7] Mongo DB : [05/05]**

Disponible sur : <https://www.mongodb.com/what-is-mongodb>

**[8] Mongoose : [05/05]**

Disponible sur : <https://fr.accentsconagua.com/articles/code/an-introduction-to-mongoose-for-mongodb-and-node-js.html>

**[9] MEAN stack : [05/05]**

Disponible sur : <http://lerjen.me/mean-stack/>

**[10] Angular 12 : [06/05]**

Disponible sur : <https://monpetitdev.fr/cest-quoi-angular-definition/>

**[11] Node-JS : [06/05]**

Disponible sur : <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1445306-node-js-definition-simple-et-utilisation-pratique/>

**[12] Express.js : [06/05]**

Disponible sur : <https://developer.mozilla.org/fr/docs/Learn/Server-side/Express_Nodejs>

**[13] Git : [06/05]**

Disponible sur : <https://git-scm.com/book/fr/v1/D%C3%A9marrage-rapide-A-propos-de-la-gestion-de-versionv>

**[14] HTML 5 : [06/05]**

Disponible sur : <https://www.ionos.fr/digitalguide/sites-internet/developpement-web/html5-cest-quoi/>

**[15] CSS 3 : [06/05]**

Disponible sur : <https://www.futura-sciences.com/tech/definitions/internet-css-4050/>

**[16] Bootstrap : [06/05]**

Disponible sur : <https://www.hostinger.fr/tutoriels/cest-quoi-bootstrap/>

**Résumé**

Ce rapport présente le compte rendu de trois mois de stage de fin d’étude effectué au sein de l’association ARSII. Il s’inscrit dans le cadre du stage de fin d’étude en vue de l’obtention du diplôme de licence appliquée en Sciences informatiques et multimédia.

Ce projet a fait l’objet d’une expérience intéressante, très bénéfique sur le plan professionnel comme sur le plan humain.

Le travail dont nous étions chargés consistait de concevoir et réaliser une application web simple à manipuler adaptée aux besoins des adhérents de l’association pour donner un meilleur service au public ciblées.

**Mots Clés** : Gestion des formations, gestion des compétences, angular, nodeJS,

**Abstract**

This report presents the report of three months of training course of end of study carried out within the association ARSII. It is part of the end-of-study internship for the diploma of applied license in Computer Science and Multimedia.

This project was an interesting experience, very beneficial on the professional and human levels.

The work we were in charge of consisted in designing and realizing a simple to handle web application adapted to the needs of the members of the association to give a better service to the targeted public.

**Keyword:** skills management, training management, item, Angular, NodeJS, MongoDB