 République Tunisienne Ministère

De l’Enseignement Supérieur et de Recherche

Scientifique Université

|  |
| --- |
| MEMOIRE DE STAGE FIN D’ETUDES |
| Environnement Numérique d’enseignement |
| Présente pour obtenir le titre : Diplôme national de licence en science de l’informatique |

|  |
| --- |
| Année scolaire 2022 / 2023 |

**Réalisé par :** Mgannem Saber

**SOUTENU LE**.... /….. /…….

LE JURY D’EXAMEN :

**Rapporteur :** Dr. Aymen Louati

**Encadreur-Entrepris :** Mme Garma Abir

**Président :**

**Encadreur-ISI :**

**Dédicace**

Du fond du cœur, je dédie ce modeste travail à tous ceux qui me sont chers.

À mes chers parents

Qui n’ont jamais cessé de m’assister et m’encourager aucune dédicace ne saurait exprimer mon amour et mes reconnaissances.

Puisse dieu vous prêter bonne santé et longue vie.

À mon sérieux encadreur

Qui m’a guidée avec ses précieux conseils.

À mes sœurs et à mon frère

Merci pour le soutien, pour tous les bons moments que nous passons ensemble.

Je vous aime.

À tous les membres de familles, à mes amis et toute personne qui je l’ai connu au Kef ravis de vous connaitre.

**Mgannem saber**

**Remerciement**

Je tiens tout d’abord à remercier « IT NEXT SOLUTION » de m’avoir accueilli durant ces 4 mois et de m’avoir donné l’opportunité, à travers mon stage.

De même, Je remercie sincèrement Mr. « Louati Aymen » mon directeur de recherche qui n’a pas cessé de mon encourager tout au long de mon projet ainsi que par sa générosité en matière de formation et d’encadrement.

Je remercie également mon encadrante de stage de la société  « Garma Abir »qui m’a beaucoup aidé et soutenu pour que ce travail soit bien organisé en me mettant sur la bonne voie sur tous les plans.

Tous mes remerciements les plus sincères à toutes les personnes avec lesquelles j’ai eu le plaisir de collaborer et qui ont pu par la même occasion m’aider durant toute la durée de mon stage.

Enfin , je remercie infiniment les membres de jury qui ont accepté de ma faire l’honneur de juger mon travail.

À tous merci.

Introduction générale

Les TIC facilitent l’accès à une culture générale riche et étendue. Elles permettent aussi le développement et l’acquisition d’un grand nombre de compétences, dont la capacité de synthèse et d'analyse. Les différentes sources d’information disponibles sur Internet aident les élèves à aiguiser leur rigueur intellectuelle et morale. Les TIC insufflent également une motivation, une détermination et un engouement sans précédent chez les élèves. Marier les TIC à la pédagogie favorise aussi le développement d’habiletés de communication, tant sur le plan de l’oral que de l’écrit. L’usage des TIC accroît aussi les contributions originales et significatives de la part des élèves. Les TIC sont, enfin, d'un commerce agréable pour la pédagogie, ce qui ne gâte rien. Pour toutes ces raisons, nous sommes entièrement convaincus que l’arrimage des TIC à la pédagogie, ce n’est pas l’école sans livre ni cartable pour l’élève. Ce n’est pas non plus la machine à apprendre de Skinner qui remplace – à nouveau – l’enseignant. L’intégration des TIC dans la pédagogie c’est l'épanouissement réel et virtuel des pédagogies humanistes et socioconstructivistes, de la pédagogie du projet, de l’apprentissage coopératif, de l’école du goût d’apprendre et de l’école transfrontalière, ouverte sur le reste du monde. Comme au moment où Gutenberg redéfinissait l’accès à la connaissance avec l’imprimerie, l’éducation est aujourd'hui en mesure de faire un gigantesque bond en avant avec l’alliance TIC et pédagogie.[[1]](#footnote-1)

***CHAPITRE 1 :***

***PRESENTATION***

***GENERALE***

***DU***

***PROJET***

***Plan***

1 Présentation de l’organisme d’accueil . . . . . . . . . . . . . . . . . . . . .

2 Étude de l’existant . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

3 Choix méthodologique . . . . . . . . . . . . . . . . . . . . . . . . . . ..

4 Langage de Modélisation UML . . . . . . . . . . . . . . . . . . . . . . . .

Introduction :

L’étude préliminaire est la première phase de tout projet réussi dans ce premier chapitre nous présentons brièvement notre structure d’accueil. Nous aborderons l’analyse de l’existant après l’exposition de la problématique, et nous exposerons ensuite la solution proposée. Enfin, nous terminerons ce chapitre en mettant l’accent sur la méthodologie de travail adoptée et nous énoncerons les raisons que nous ont poussés à ce choix.

Présentation de la société :

It Next Solutions est une entreprise technologique internationale qui conçoit des solutions web, mobile et désigne, innovantes et personnalisées. Créé en 2019, It Next Solutions est présent aujourd’hui à Teboulba, Monastir. Elle se positionne comme le partenaire de confiance des grandes entreprises tunisiennes et Française.

● Organisme d’accueil :

• Adresse : Teboulba, 5080, Monastir

• Numéro Tel° : (+216) 95 261 644

• Email : itnextsolution @gmail.com

● Services :

It Next Solution offre à ces clients :

 Services informatiques  Solutions digitales

 Application & Site Web  Logiciel de gestion d’entreprise

 Marketing digital  Solutions PDV - Logiciel de Caisse

 Développement web  Design graphique

Cadre général de projet :

Contexte de projet :

Contexte de Projet Notre projet ayant comme objectifs la conception et la réalisation d’une plateforme de « Environnement Numérique d’Enseignement » qui consiste à mettre en place une plateforme web destinée à l’apprentissage en ligne ou le « e-formation » pour les écoliers l'école primaire. L’E-formation est considérée comme une simple alternative à la formation traditionnelle, en général qualifiée de présentielle. Or l’e-formation n’est pas réduite à la formation à distance, mais elle couvre un spectre plus large de situations associées au développement et à la gestion des connaissances.

Analyse de l’existant :

Description de l’existant :

Il existe plusieurs plateformes en ligne qui permettent l ‘apprentissage mais seulement quelques solutions sont largement utilisées.

« Madrassatti » est un exemple de plateforme tunisienne spécifique dans le domaine de l'éducation. Dans le cadre de ses efforts visant à rapprocher le tuteur de l'école et à suivre l'évolution technologique, le projet "Madrassatti" propose aux élèves, notamment aux parents, une nouvelle application - dans sa version actualisée - simple d'utilisation et téléchargeables sur leur smartphone, leur permettant de suivre toutes les évolutions, actualités et activités des deux perspectives, ainsi que les taux de nombres, les horaires et les devoirs des élèves.

La figure suivante montre la page principale de cette plateforme :



Figure 1.1 : Site Internet de Madrassatti.

« MonProf.tn » est une autre plateforme qui offre l'opportunité de l'éducation à distance dès notre écolier destiné à les intervenants du système éducatif : élèves et enseignants, mais avec absences les cotes parentes

La figure suivante présente la page d’accueil de cette plateforme :

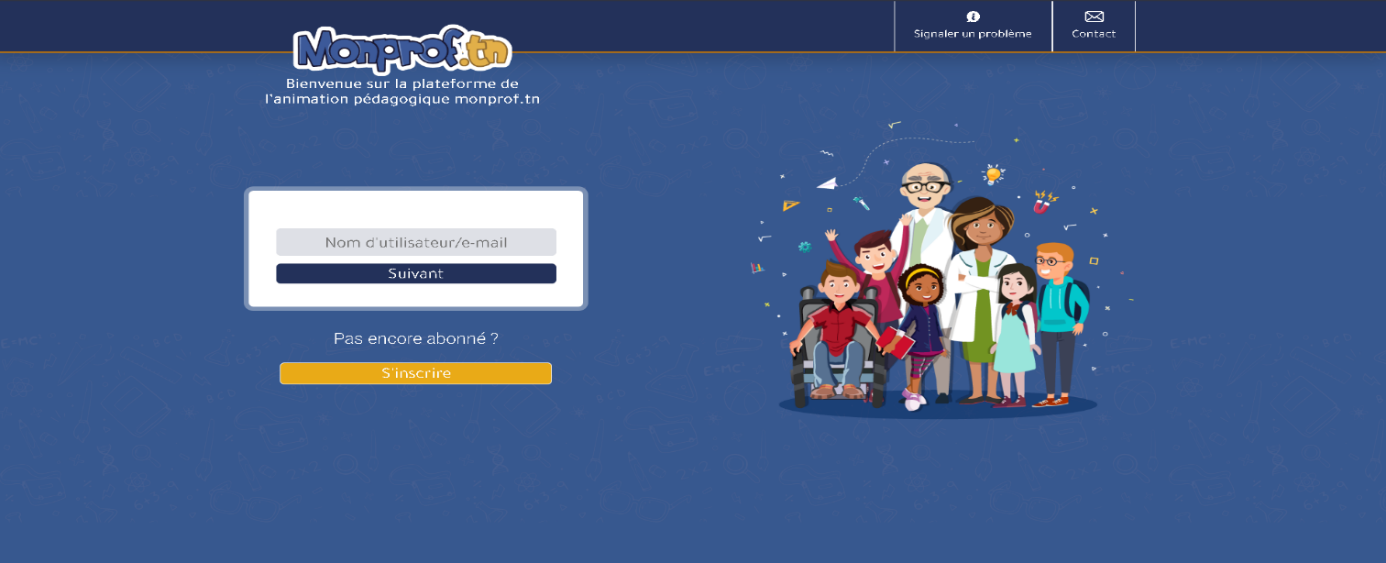


Figure 1.2 : Site Internet de MonProf.

**Critiques de l’Existant**

- La majorité des plateformes existantes ont un nombre limité dès la fonctionnalité de l’étudiant.

- Les parents n’ont pas, généralement, un espace spécifié pour consulter liste de notes de leur enfants, l’emploi de temps et contacter les enseignants

- Les participants n'ont pas un espace pour gérer leurs inscriptions

- Absence d’un espace dédiée pour les réunions en ligne

**Solution Proposée**

Pour remédier aux lacunes déjà mentionnées dans le paragraphe précédent, nous proposons comme solution la conception et le développement d’une plateforme d'étude a distance pour les écoles primaires qui permet de faciliter la communication entre l’institutrice, les apprenants et les parents. De plus, elle permet de mieux assimiler l'apprentissage des cours à distance (qui offre un service aux apprenants d’écoles primaires)

.

**La démarche adoptée**

La complexité croissante des systèmes informatiques a conduit les concepteurs à s’intéresser aux méthodes de développement. Ces dernières ont toujours essayé d’apporter un contrôle continu sur un projet tout au long de son processus de vie pour aboutir à la fin à un logiciel fiable et de qualité et pour garantir de rester fidèle aux besoins du candidat.

**Choix du cadre méthodologique**

Étant à la mode, les démarches agiles prouvent de jour en jour leur efficacité par rapport aux méthodes classiques caractérisées par leur rigidité en définissant un cadre strict en termes du coût, du cadre et du périmètre. En se focalisant sur la satisfaction des clients, en autorisant le changement, en offrant une meilleure visibilité aux développeurs et en se basant sur des cycles courts, ces approches sont de plus en plus adoptées par de nombreuses entreprises, notamment de développement informatique. Alors que le développement d’une application avec la méthode « traditionnelle » est très différent. Le choix entre une méthode et une autre dépend de la nature et de l'échelle du projet.

Lors de l’utilisation d’une approche « traditionnelle », le projet, ses fonctionnalités et sa finalité sont clairement définies à l’avance. Cette méthode repose sur l’utilisation d’un processus strict, la rédaction d’une documentation détaillée et une implication plus faible du client. Elle consiste à définir l’ensemble des fonctionnalités du projet, les spécifier de façon détaillée, les développer puis les tester avant validation et mise en service. Il existe de nombreuses méthodes AGILE et il ne s’agit pas de choisir la meilleure parmi les méthodes existantes. Il s’agit plutôt de choisir la méthode là mieux adaptée à notre projet. Afin de fixer notre méthode, nous avons effectué une étude comparative entre les méthodes classiques et la méthode AGILE. Nous présentons dans la figure suivante 1.3 un comparatif entre l’approche classique et l’approche AGILE.



Figure 1.3 : Figure comparatif entre l’approche classique et l’approche AGILE

Agile

Définition :

Les méthodes agiles sont des méthodologies essentiellement dédiées à la gestion de projets informatiques. Elles reposent sur des cycles de développement itératifs et adaptatifs en fonction des besoins évolutifs du client. Elles permettent notamment d'impliquer l'ensemble des collaborateurs ainsi que le client dans le développement du projet.

Ces méthodes permettent généralement de mieux répondre aux attentes du client en un temps limité (en partie grâce à l'implication de celui-ci) tout en faisant monter les collaborateurs en compétences. Ces méthodes constituent donc un gain en productivité ainsi qu'un avantage compétitif tant du côté client que du côté du fournisseur.[[2]](#footnote-2)

Caractéristiques

Les valeurs communes de ces méthodes :

Les méthodes agiles se reconnaissent toutes dans les valeurs suivantes.

**L'équipe et la communication avant les outils et processus :** dans la vision agile, l'équipe est bien plus importante que les outils ou les procédures de fonctionnement. Il est préférable d'avoir une équipe soudée et dont les membres communiquent entre eux, composée de développeurs de niveaux différents, plutôt qu'une équipe composée d'experts qui travaillent de manière isolée. La communication est donc une notion fondamentale dans un contexte de développement agile.

**L'application avant la documentation :** il est primordial que le projet fonctionne, c'est la priorité avant toute chose. La documentation technique et les autres outils (de tests, de reporting) constituent une aide précieuse, mais ne sont pas une fin en soi. Une documentation précise est utile comme moyen de communication. Il est parfois préférable de simplement commenter abondamment le code lui-même, et surtout de transférer la totalité des compétences et connaissances du métier à l'ensemble des collaborateurs de l'équipe.

**La collaboration avant la négociation :** le client doit être impliqué dans le développement. Le fournisseur ne doit pas se contenter de négocier un contrat au début du projet, puis de refuser l'évolution des besoins du client. Le client doit collaborer avec l'équipe et fournir des comptes rendus réguliers sur l'adaptation du logiciel à ses attentes.

**L'acceptation du changement et la flexibilité avant la planification :** la planification initiale et la structure du projet doivent être flexibles afin de permettre les évolutions attendues par le client. En effet, les premières livraisons du projet donnent très souvent suite à des demandes d'évolution.[[3]](#footnote-3)

Les Principes méthodes Agiles :

Sous le parapluie de l’agilité, on trouve plusieurs noms dont les plus connus sont :

* SCRUM
* Extreme Programming (XP)
* Rapid Application Development (RAD)
* Crystal clear

Pour notre projet, nous nous sommes orientés vers SCRUM.

SCRUM

Définition :

Scrum est un cadre de travail léger qui aide les personnes, les équipes et les organisations à générer de la valeur grâce à des solutions adaptatives pour des problèmes complexes.[[4]](#footnote-4)

Ken Schwaber et Jeff Sutherland les créateurs de SCRUM le définissent dans son guide comme suit : « SCRUM est un cadre de travail itératif permettant de répondre à des problèmes complexes et changeants, tout en livrant de manière productive et créative des produits de la plus grande valeur possible. ». [[5]](#footnote-5)

Scrum est simple. Essayez‐le tel qu'il est et, déterminez si sa philosophie, sa théorie et sa structure aident à atteindre les objectifs et à créer de la valeur. Le cadre de travail Scrum est volontairement incomplet, ne définissant que les parties nécessaires pour mettre en œuvre la théorie Scrum. Scrum repose sur l'intelligence collective des personnes qui l'utilisent. Plutôt que de fournir aux gens des instructions détaillées, les règles de Scrum guident leurs relations et leurs interactions

. La figure ci-dessous décrit le « processus » SCRUM en mettant l’accent sur ses principaux artefacts, rôles et évènements :

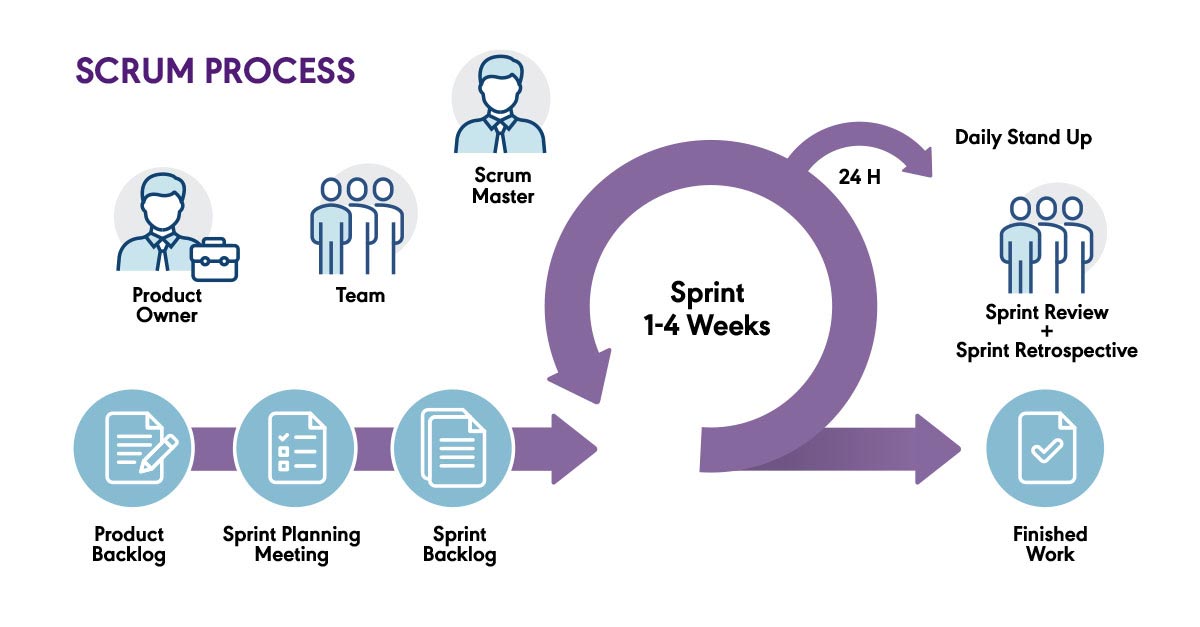


Figure 1.4 : Processus SCRUM

Pourquoi Scrum ??

Pourquoi la méthodologie Scrum plutôt qu’une autre ? Si vous développez un produit, une application par exemple, les besoins de vos clients, ou de votre employeur, ne sont pas fixer. Vos propositions vont même leur donner de nouvelles idées, sur lesquelles vous allez devoir rebondir.

Cette grande flexibilité qu’on vous demande, c’est ce qu’on appelle l’agilité. Finis les cahiers des charges détaillés sous toutes les coutures ! Finie la planification rigide et contraignante !

Le principe de base de Scrum est le suivant :

* Tout d’abord, déterminez le maximum des fonctionnalités pour former le carnet du produit.
* Deuxièmement, définissez la priorité de la fonction et sélectionnez la priorité qui sera exécutée à chaque itération.
* Par la suite focaliser l’équipe de façon itérative sur l’ensemble de fonctionnalités à réaliser, dans des itérations appelées Sprints. Un Sprint aboutit toujours sur la livraison d’un produit partiel fonctionnel appelé incrément.

Scrum est :

* Léger
* Simple à comprendre
* Difficile à maitriser
* Facilite également la communication entre l’équipe de projet durant un développement d’un projet avec la méthode Scrum il y a plusieurs étapes à suivre avec une démarche spécifique et une interaction avec plusieurs intervenants.

Langage de modélisation UML :

**La modélisation est la conception de applications logicielles** avant le codage. La modélisation est une partie essentielle de grands projets logiciels et projets utiles aux projets moyens et même petits bien. Un modèle joue le rôle analogue dans le développement de logiciels qui plans et autres plans (cartes de site, élévations, modèles physiques) jouer dans la construction d'un gratte-ciel.[[6]](#footnote-6) Pour concevoir notre application, nous avons choisi UML comme langage de modélisation et ce grâce à ses points forts et ses divers diagrammes qu’il propose.

Conclusion :

Dans ce chapitre, nous avons commencé par présenter notre organisme d’accueil. It Next Solutions Ensuite, nous avons dressé la problématique de notre projet Aussi, nous avons établi des solutions qui rendent l’application plus performante. Nous avons dévoilé le langage et la méthode de conception qui seront utilisés dans les prochains chapitres de ce rapport.

Le chapitre suivant sera consacré à l’étude des besoins fonctionnels et non fonctionnels, la spécification du Backlog de produit final et la préparation du planning de notre travail.

***CHAPITRE 2 :***

***PLANIFICATION***

***&***

***ARCHITECTURE***

***Plan***

1. Identification des acteurs . . . . . . . . . . . . . . . . . . . . . . . . . . . 14
2. Analyse et Spécifications des besoins. . . . . . . . . . . . . . . . . . . . 15
3. Diagramme de cas d’utilisation . . . . . . . . . . . . . . . . . . . . . . . . 16
4. Diagramme de classe global (métiers) . . . . . . . . . . . . . . . . . . . . 17
5. Planning du traitement des cas d’utilisation . . . . . . . . . . . . . . . . 18
6. Pilotage du projet avec Scrum……... . . . . . . . . . . . . . . . . . . . . . . ..22
7. Architecture de la solution . . . . . . . . . . . . . . . . . . . . . . . . . . 25
8. Environnement de travail . . . . . . . . . . . . . . . . . . . . . . . . . . . 27
9. Choix technologiques . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29

Introduction :

Après avoir mis en cadre ce projet, ce chapitre présentera la planification et l’architecture de notre projet. Nous commencerons par l’identification des acteurs de notre plateforme, lister les besoins fonctionnels et non fonctionnels tout en appliquant notre cadre méthodologique présenté dans le chapitre précédent. Enfin, nous donnerons un bref aperçu sur le matériel utilisé, les technologies et les langages de programmation utilisés pour la mise en place de notre plateforme.

Identification des acteurs :

Un acteur représente l’abstraction d’un rôle joué par des entités externes (utilisateur, dispositif matériel ou un autre système) qui interagit directement avec le système étudié. Notre plateforme aura quatre acteurs :

Enseignant.

Elève. (Étudiant, élève, apprenti …)

Parent.

Administration.

* Les élèves : sont inscrits en fonction du niveau d'éducation. Ils peuvent, consulter les leçons, les enregistrements et télécharger les cours, les exercices aussi participer en des formations et des compétitions.
* Les Parents : peuvent créer des comptes, consulter les notes, l’emplois de temps et contacter les enseignants.
* Les enseignants : peuvent ajouter des cours et des enregistrements sous la forme des vidéos à chaque matière aussi des documents et des examens chaque un de ses cours, vidéos et documents appartiennent à des offres ou peuvent être gratuits. Ils peuvent créer des formations, des compétitions, contacter les parents et postuler les notes.
* L’administrateur de la plateforme pédagogique assure la maîtrise d’œuvre de tous les systèmes d’informations de l’établissement destinés aux activités pédagogiques des enseignants. Il installe, fait évoluer et exploite techniquement et fonctionnellement la plateforme. Il est l’interlocuteur privilégié des enseignants pour la prise en main et l’exploitation de ces outils numériques.

Analyse et Spécifications des besoins :

La spécification de besoins constitue la phase de départ de toute application à développer dans laquelle nous identifions les besoins de notre application. Ensuite, nous distinguons les besoins fonctionnels qui présentent les fonctionnalités attendues de notre application et les besoins non fonctionnels pour ´éviter le développement d’une application insatisfaite et de trouver un accord commun entre les spécialistes et les utilisateurs pour la réussite de notre projet.

Les besoins fonctionnels :

Les besoins fonctionnels c’est ce que l’utilisateur attend en matière de fonctionnalités. Ces besoins présentent une description abstraite de ce que l’application doit faire pour les utilisateurs et qui convient à leurs attentes et satisfaire leurs exigences.

Les besoins fonctionnels de l’administrateur :

* Authentification : Chaque administrateur doit s’authentifier en tapant login et mot de passe pour accéder à son espace privé.
* Gestion des utilisateurs : Après son authentification, l’administrateur consulte son espace privé dont il peut : consulter, ajouter, modifier et supprimer des utilisateurs.
* Gérer les dates de fête : Après son authentification, l’administrateur consulte son espace privé dont il peut : consulter, ajouter, modifier et supprimer les dates de fête
* Gérer les emplois de temps : Après son authentification, l’administrateur consulte son espace privé dont il peut : consulter, ajouter, modifier les emplois de temps de l’enseignant et l’étudiant.
* Gérer les cours et les exercices : Après son authentification, l’administrateur consulte son espace privé dont il peut : consulter, ajouter, modifier les cours et les exercices de chaque matière.
* Gérer les documents et des fichiers intéressant les utilisateurs.
* Changer des textes ou des images de plateforme.
* Gérer les droits d’accès.

Les besoins fonctionnels de l’enseignant :

* Authentification : Chaque enseignant doit s’authentifier en tapant login et mot de passe pour accéder à son espace privé.
* Les fonctions de gestion de compétition :
* Consulter la liste de compétition
* Ajouter, Modifier, Supprimer compétition
* Les fonctions de gestion des cours :
* Consulter la liste de cours déjà crée
* Ajouter, Modifier, Supprimer cours
* Les fonctions de gestion des notes :
* Consulter la liste des notes déjà crée
* Ajouter, Modifier, Supprimer notes
* Les fonctions de gestion des exercices :
* Consulter la liste de exercices déjà crée
* Ajouter, Modifier, Supprimer exercices
* Envoie un messager

Les besoins fonctionnels de l’écoliers :

* Authentification : Chaque étudiant doit s’authentifier en tapant login et mot de passe pour accéder à son espace privé
* Participer la compétition
* Consulter liste de note
* Consulter les cours et les exercices

Les besoins fonctionnels du parent :

* Authentification : Chaque parent doit s’authentifier en tapant login et mot de passe pour accéder à son espace privé.
* Contacter les enseignant
* Consulter liste de notes
* Consulter emplois de temps

Besoins non Fonctionnels :

Une fois les besoins fonctionnels sont bien définis, les besoins non fonctionnels doivent être pris en

compte tout au long du processus de développement de notre plateforme

Les besoins non fonctionnels sont des besoins qui ont un aspect visible pour l’utilisateur, mais qui ne sont pas reliés directement au comportement du système.

* Besoin d’utilisation : La plateforme devrait être cohérente au point de vue de l’ergonomie. Tous les standards d’ergonomies doivent être présents : interface utilisateur bien claire, conviviale et simple dans l’utilisation
* Besoin de « Responsive design » : les IHMS de notre plateforme doivent se comporter de la même façon sur des terminaux de types différents et de caractéristique différente.
* Besoins de sécurité : l’accès aux services de la plateforme sera strictement réservé aux utilisateurs inscrits, un accès avec login et mot de passe sera exigé.
* Besoin de performance : il s’agit d’optimiser le temps de chargements des pages web de la plateforme et ceci par l’utilisation des données et nouvelles pratiques du développement web.
* Besoin de portabilité et de comptabilité : notre application doit être portable sur tous les environnements logiciels (Windows, Mac OS, Linux).
* Besoin d’extensibilité : la plateforme doit permettre à tout moment d’ajout d’autres modules.

Diagramme de cas d’utilisation

Définition :

Les diagrammes de cas d’utilisation (DCU) permettent de recueillir, d’analyser, d’organiser les besoins et de recenser les grandes fonctionnalités d’un système (étape analyse du système). Ses principaux éléments sont :

— Les acteurs : qui présentent une personne externe, un processus ou une chose qui interagit avec un système.

— Les cas d’utilisations qui représentent une fonctionnalité visible de l’extérieur.

Diagramme de cas d’utilisation général :

Le diagramme de cas d’utilisation global, c’est un formalisme qui permet de décrire d’une manière générale l’utilisation des différentes fonctionnalités de notre application pour tous les acteurs.

Voici le diagramme de cas d’utilisation globale pour notre système présenté par la figure 2.1 :

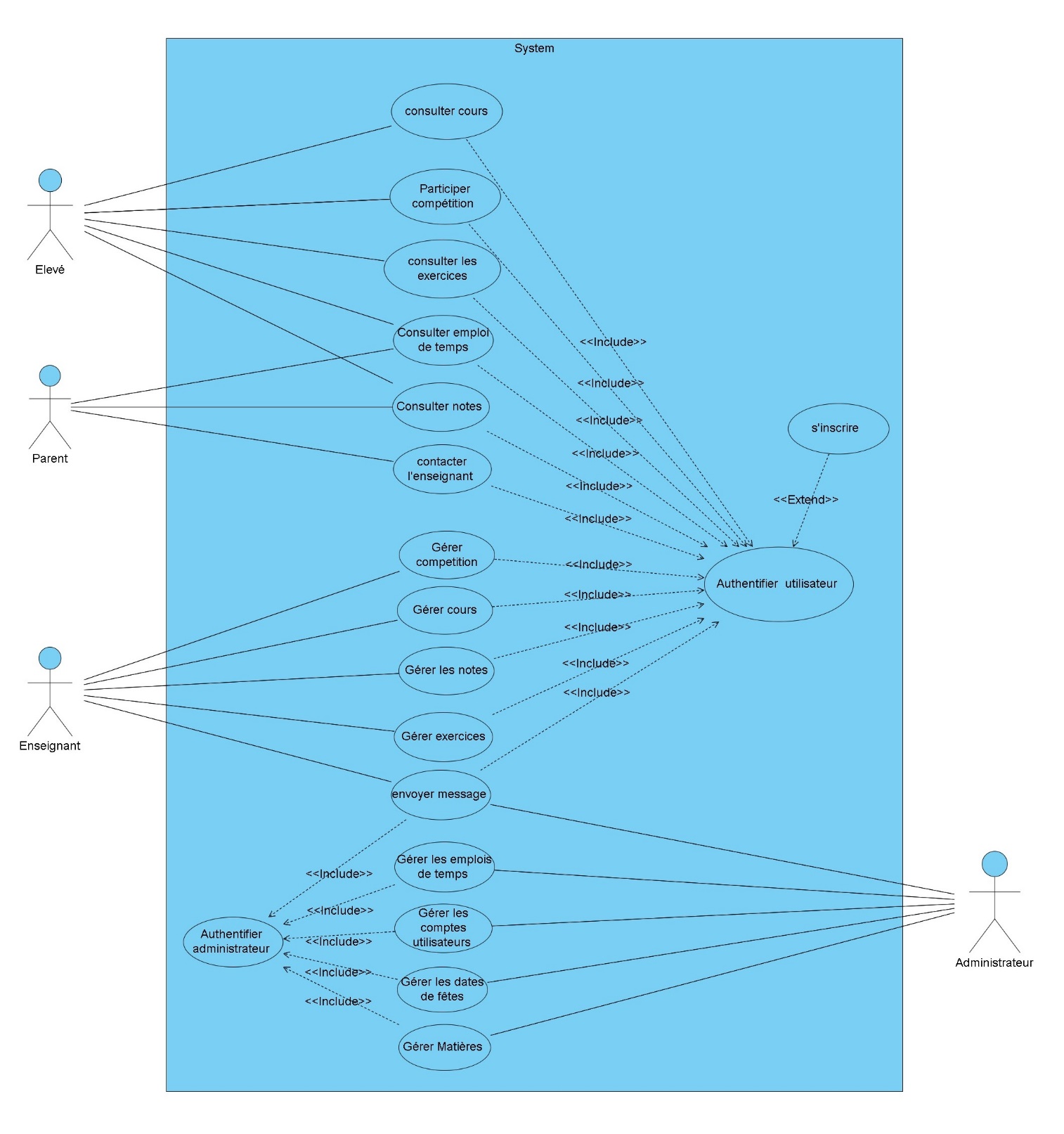


Figure 2.1 : Diagramme de cas d’utilisation global

Diagramme de classe global (métiers) :

Le [langage de modélisation unifié](https://www.lucidchart.com/pages/fr/langage-uml) (UML) peut vous aider à modéliser des systèmes de plusieurs façons. Le diagramme de classes est l'un des types les plus populaires en langage UML. Très utilisé par les ingénieurs logiciels pour documenter l'architecture des logiciels. Le diagramme de classe est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations. Ce diagramme fait partie de la statique d’UML car fait abstraction des aspects temporels et dynamiques. Cela permet d’unifier le travail de plusieurs programmeurs au sein d’une même équipe, en plus de sauver du temps. Le diagramme de classes se base sur les concepts suivants :

- Classe : description formelle d’un ensemble d’objets ayant une sémantique, des propriétés et un

Comportement commun.

- Association : relation sémantique entre deux ou plusieurs classes.

-Propriété (attribut) : élément permettant de décrire une classe ou une association.

-Méthode : opération réalisée par les objets d’une classe.

(mizelet ma 5demtouch)

Planning du traitement des cas d’utilisation :

Le classement des cas d’utilisation doit prendre en considération deux facteurs principaux : la priorité et le risque.

**Priorités**

Le carnet du produit est toujours rempli de divers éléments que vous devez traiter. Le principal problème est qu’il n’y a qu’un nombre limité d’heures dans une journée et de dollars dans un budget. Vous devez clarifier vos priorités et décider sur quelles fonctionnalités travailler en émier.

La meilleure façon de s’y prendre est d’adopter une approche systématique de la priorisation Scrum. Pour que cela fonctionne, les équipes doivent partager la responsabilité de déterminer les éléments et les tâches hautement prioritaires. Par exemple, tant que la gestion des utilisateurs n’est pas terminée, nous ne pouvons pas gérer un modèle d’extraction des données. Par conséquent, nous avons déterminé trois priorités : haute, moyenne et faible priorité.

**Risques**

Le risque est défini comme un événement incertain ou un ensemble d’événements qui peuvent affecter les objectifs d’un projet et peuvent contribuer à son succès ou à son échec. Les menaces sont des risques qui pourraient affecter le projet de manière négative, c’est pour ça la gestion des risques est une étape nécessaire à sa réussite. Il s’agit se représente la possibilité qu’une tâche donnée ait des impacts négatifs sur le projet si elle est mal définie ou calibrée.

Tableau 2.1 : Risque du projet

|  |  |  |  |
| --- | --- | --- | --- |
| les risques | type | Impact sur le  projet | Action correctives |
| Panne au niveau de base de  données | Suppression de la  base de données | Problème d’extraire des  données | Sauvegarder des  données |

Pilotage du projet avec Scrum

Équipe et rôle :

Scrum définit un modèle d’équipe qui maximise la flexibilité, la créativité et la productivité.

L’équipe Scrum comprend :

* + Concepteur/Développeur : les développeurs sont les personnes de l’équipe Scrum qui s’engagent à créer n’importe quel aspect d’un incrément utilisable à chaque sprint.[[7]](#footnote-7)
  + SCRUM MASTER : le Scrum Master est responsable de l’établissement de Scrum. Ils le font en aidant tout le monde à comprendre la théorie et la pratique Scrum, à la fois au sein de l’équipe Scrum et de l’organisation.[[8]](#footnote-8)
  + Product Owner : un Product Owner Scrum est responsable de maximiser la valeur du produit résultant du travail de l’équipe Scrum. La façon dont cela est fait peut varier considérablement d’une organisation à l’autre, d’une équipe Scrum ou d’un individu à l’autre. Le Product Owner est également responsable de la gestion efficace du Product Backlog.[[9]](#footnote-9)

Tableau 2.2 : Equipes de réalisation de projet

|  |  |
| --- | --- |
| Nom et prénom | Type Fonction/Rôles dans le projet |
| Saber Mgannem | Stagiaire/concepteur et développeur |
| D .Aymen LOUATI | Encadrant/Scrum Master |
| Abir Garma | Responsable société/Product Owner |

Le carnet du produit (Product Backlog) :

Le Product Backlog est une liste ordonnée de tout ce qui pourrait être requis dans le produit et est l’unique source des besoins pour tous les changements à effectuer sur le produit. Le Product Owner est responsable du Product Backlog dans son contenu, sa disponibilité et son ordonnancement.

Le tableau suivant représente le Product Backlog de notre plateforme. Chaque histoire utilisateur constitue un item du backlog et elle est caractérisée par un rang déduit à partir de sa priorité. Nous détaillons ici la signification des différents termes utilisés.

✔ Id qui représente l’identifiant d’histoires utilisateur ;

✔ Thème pour mieux ordonnez les histoires utilisateur comme exemple gestion de profil ;

✔ User Story comporte la description des histoires utilisateurs suivant la forme « En tant

que...Je veux...Afin que » ;

✔ Conditions de satisfaction :

✔ La priorité des histoires utilisateur selon la valeur métier et l’ordre de réalisation.

Tableau 2.3 : Le Backlog (Carnet du produit)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Thème | En tant que .. | Je veux | .. Fin que | Condition de satisfaction | Priorité |
| 1 | Inscription | Utilisateur | M’inscrire  sur la plateforme  mon compte. | Je devienne  Membre | - L’adresse email est  unique pour chacun  -Appui sur le bouton  « Registre », je me retrouve  sur la page de connexion.  - Le champ E-mail sera du  type adresse email.  - Le mot de passe sera caché  avec une confirmation de mot de passe.  - Appui sur le bouton  « Connection » », je me retrouve  sur la page de profile de mon espace prive. | 1 |
| 2 | Authentification | Utilisateur  Admin | Me  connecter | J’accède à  mon espace | Après authentification je me  retrouve sur mon tableau de  bord, | 2 |
| 4 | Gestion Profils | Utilisateur  (Enseignant, Parent, Etudiant)  Admin | -Consulter  Mon profil  -modifier  mon profil | - Vérifie mes données  - Je mettre à jour mes  informations | -Consulter Mon Compte  -Remplir les champs avec des  Informations à modifier | 2 |
| 5 | Envoyer message | Enseignant  Parent  Admin | Envoyer et consulter message | - Je voie la  Liste de conversation  -je voie la liste d’amie  -envoyer message | -Accéder sur « Message » aux Sidebar, puis remplir le champ avec le message et appui sur l’icône envoyer | 1 |
| 6 | Gérer des  utilisateurs | Admin | Consulter, modifier  Et supprimer | - Vérifie mes données  - Je mettre à jour mes  informations | En tant que administrateur, je consulte mon espace privée dont je peux : consulter , ajouter, modifier et supprimer des utilisateurs. | 2 |
| 7 |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |

Architecture de la solution :

Nous avons présenté l’architecture logique et logicielle

Architecture logique :

La figure 2.2 représente l’architecture technique que nous avons adoptée pour notre solution.

Il s’agit d’une application web qui se connecte à un serveur de base de données distant, via internet, afin de récupérer les données, ce qui nécessite l’intégration d’un serveur web entre l’application client et le serveur de base de données. D’où l’architecture de notre application qui est partagée entre :

• Le client : conteneur de l’application et demandeur des ressources.

• Le serveur d’applications : vu que les données seront communiquées entre deux environnements hétérogènes.

Le rôle principal de serveur web est de gérer la communication entre le client et le serveur de base de données.

• Le serveur de base de données qui fournit les données au serveur web.

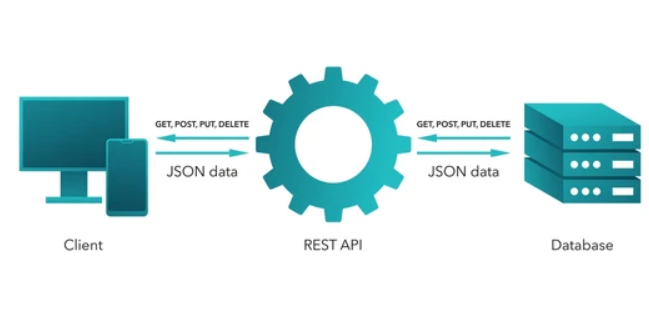


Figure 2.2 : Architecture logique

Architecture logicielle :

Afin de bien concevoir notre site web et bien organiser notre code source, nous avons opté pour l’architecture MVC (Modèle - Vue – Contrôleur) qui a pour but de séparer la logique du code en Trois parties que l’on retrouve dans des fichiers distincts, comme l’explique la description qui suit :

• Modèle :

Cette partie gère les données de notre site. Son rôle est d’aller récupérer les Informations « brutes » dans la base de données, de les organiser et de les assembler pour Qu’elles puissent ensuite être traitées par le contrôleur.

• Vue : Partie visible d'une [interface graphique](https://fr.wikipedia.org/wiki/Interface_graphique). La vue se sert du modèle, et peut être un diagramme, un formulaire, des boutons, etc. Une vue contient des éléments visuels ainsi que la logique nécessaire pour afficher les données provenant du modèle[3](https://fr.wikipedia.org/wiki/Mod%C3%A8le-vue-contr%C3%B4leur#cite_note-lott-3). Dans une application de bureau classique, la vue obtient les données nécessaires à la présentation du modèle en posant des questions. Elle peut également mettre à jour le modèle en envoyant des messages appropriés[4](https://fr.wikipedia.org/wiki/Mod%C3%A8le-vue-contr%C3%B4leur#cite_note-:0-4). Dans une application web une vue contient des balises [HTML](https://fr.wikipedia.org/wiki/Hypertext_Markup_Language)

• Contrôleur : Module qui traite les actions de l'utilisateur, modifie les données du modèle et de la vue

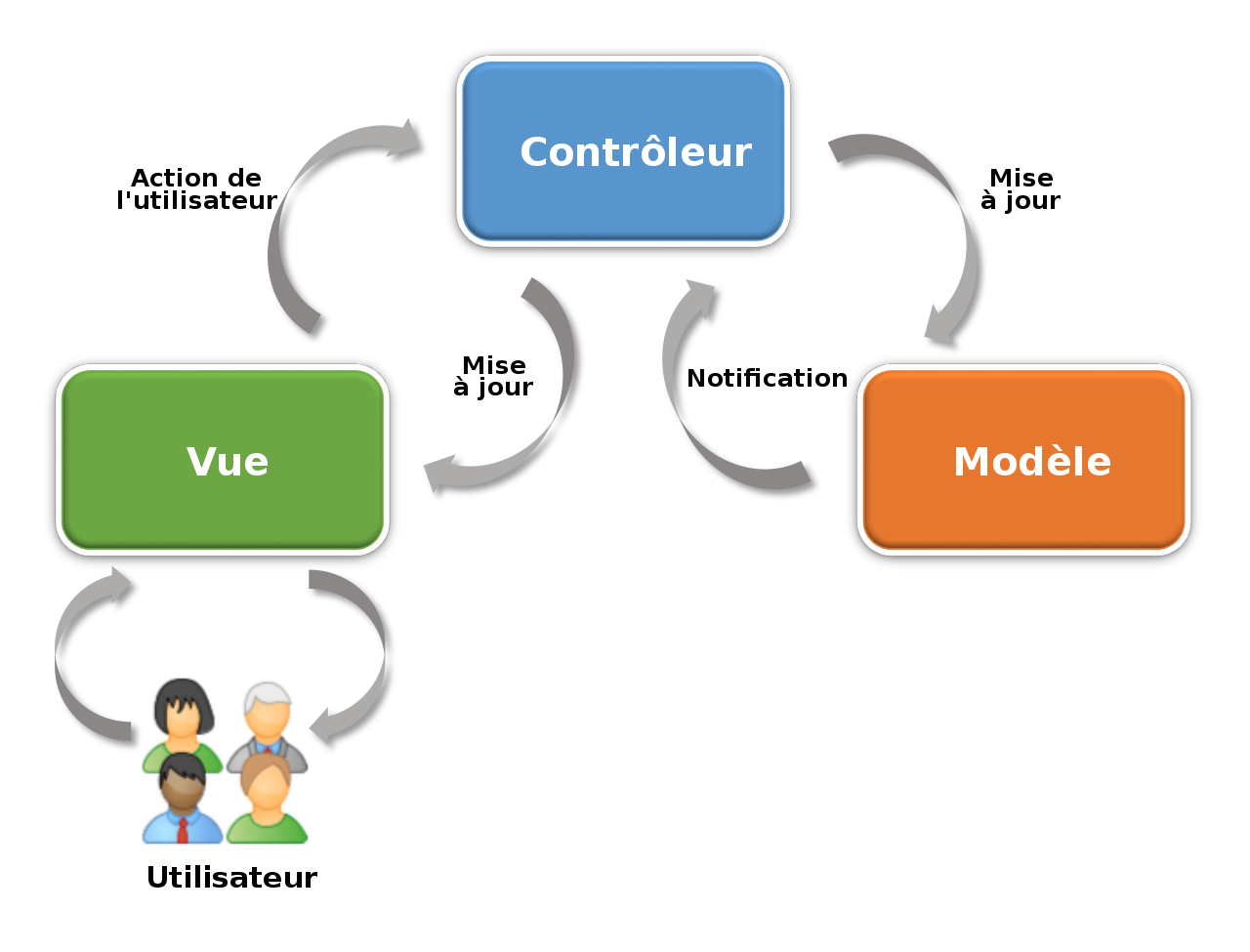


Figure 2.4 : Architecture MVC

Environnement de travail :

L’environnement de travail désigne l’ensemble des outils matériels et logiciels qui compose le cadre de travail, Pour réaliser ce travail, nous avons eu recours aux environnements suivants :

Environnement logiciel :

Dans cette partie, nous présentons les différents logiciels utilisés pour la réalisation de notre système.

• Outil de développement :

\* Visual Studio Code :

Visual Studio Code est un éditeur de code source léger mais puissant qui s’exécute sur votre bureau et est disponible pour Windows, macOS et Linux. Il est livré avec un support intégré pour JavaScript, TypeScript et Node.js et dispose d’un riche écosystème d’extensions pour d’autres langages (tels que, Java, PHP).

• Serveur d’application :

\* XamppServer :

XAMPP est l’environnement de développement PHP le plus populaire. Elle est une distribution Apache entièrement gratuite et facile à installer contenant MySQL, PHP et Perl. Le paquetage open source XAMPP a été mis au point pour être incroyablement facile à installer et à utiliser.

X pour cross-plateforme (LAMPP pour Linux, WAMPP pour Windows...)

- A pour Apache

- M pour MySQL

- P pour PHP

- P pour Perl

• Outils de rédaction de rapport :

\* Draw.io :

Est un logiciel propriétaire pour la création de diagrammes et de graphiques. Le logiciel vous permet de choisir parmi une fonction de mise en page automatique ou de créer une mise en page personnalisée. Ils ont un grand choix de formes et des centaines d’éléments visuels pour rendre votre diagramme ou graphique unique en son genre.

• Outils de gestion de base de donnée :

\* MySQL :

MySQL est un serveur de bases de données relationnelles Open Source. Un serveur de bases de données stocke les données dans des tables séparées plutôt que de tout rassembler dans une seule table. Cela améliore la rapidité et la souplesse de l’ensemble. Les tables sont reliées par des relations définies, qui rendent possible la combinaison de données entre plusieurs tables durant une requête. Le SQL dans "MySQL" signifie

"Structured Query Language" : le langage standard pour les traitements de bases de données.

Choix technologiques**:**

•Langage de développement :

* PHP : HyperText Préprocesseur, PHP est un langage de script généraliste populaire qui est particulièrement adapté au développement Web est inclus dans une page HTML et sera exécuté chaque fois qu’un visiteur affichera la page.[[10]](#footnote-10)
* HTML : Le HTML (HyperText Markup Language) est le langage utilisé pour construire les pages Web. Logique et élémentaire, inspirée du langage naturel, sa syntaxe est à la portée du programmeur débutant, à tel point qu'on hésite à parler encore de programmation. Cette simplicité est à l'origine du développement extraordinaire du Web. Mais simplicité ne signifie pas limitation : des greffons modulaires (prêts à l'emploi) en JavaScript, en VBScript ou en d'autres langages complexes donnent au HTML la puissance d'un langage fédérateur universel. Le HTML sert aussi à produire de puissantes appLications multi-plates-formes sur CD ou sur intranet. S'initier au HTML, cette deuxième révolution du texte, après l'imprimerie, deviendra bientôt aussi indispensable qu'apprendre à lire et à écrire.[[11]](#footnote-11)
* JavaScript est un puissant langage de programmation pour le Web. Ce langage de script permet d'insérer du contenu interactif dans des applications web côté client. Les pages web ne sont plus statiques, mais contiennent des programmes interactifs, contrôlent le navigateur et, combiné avec DOM, créent des pages DHTML (Dynamic HTML).[[12]](#footnote-12)
* CSS : une nouvelle fonctionnalité ajoutée au HTML qui donne aux développeurs de sites Web et aux utilisateurs plus de contrôle sur la façon dont les pages sont affichées. Avec CSS, les concepteurs et les utilisateurs peuvent créer des feuilles de style qui définissent l’apparence des différents éléments, tels que les en-têtes et les liens. Ces feuilles de style peuvent ensuite être appliquées à n’importe quelle page Web.
* JSON : Signifie « JavaScript Object Notation » est un format d’échange de données léger. Il est facile pour les humains de lire et d’écrire. Il est facile pour les machines d’analyser et de générer. Il est basé sur un sous-ensemble de la norme de langage de programmation JavaScript ECMA-262 3rd Edition - décembre 1999. JSON est un format de texte qui est complètement indépendant du langage mais utilise des conventions familières aux programmeurs de la famille C de langages, y compris C, C ++, C, Java, JavaScript, Perl, Python et bien d’autres. Ces propriétés font de JSON un langage d’échange de données idéal.
* API : API est l’acronyme d’Application Programming Interface, que l’on traduit en français par interface de programmation applicative ou interface de programmation d’application. L’API est une solution informatique qui permet à des applications de communiquer entre elles et de s’échanger mutuellement des services ou des données
* Framework :
* Laravel : Qu'est-ce qui distingue Laravel des autres Framework Web PHP ? Rapidité et simplicité, pour commencer. Ce cadre de développement d'applications rapide et son écosystème d'outils vous permettent de créer rapidement de nouveaux sites et applications avec un code propre et lisible. Entièrement mise à jour pour couvrir Laravel 5.8, la deuxième édition de ce guide pratique fournit l'introduction définitive à l'un des Framework Web les plus populaires d'aujourd'hui.[[13]](#footnote-13)
* Bootstrap : Bootstrap est un Framework de développement frontal gratuit et open source pour la création de sites Web et d’applications Web. Le Framework Bootstrap est basé sur HTML, CSS et JavaScript (JS) pour faciliter le développement de sites et d’applications réactifs et mobiles.
* React : est un Framework JavaScript. React a été créé à l'origine par des ingénieurs de Facebook pour résoudre les problèmes liés au développement d'interfaces utilisateur complexes avec des ensembles de données qui changent au fil du temps. Ce n'est pas une entreprise anodine et doit non seulement être maintenable, mais aussi évolutive pour fonctionner à l'échelle de Facebook[[14]](#footnote-14)

Conclusion :

Dans le cadre de ce chapitre, nous avons détaillé les besoins fonctionnels et non fonctionnels de notre application. Ainsi, nous avons présenté le diagramme de cas d’utilisation général et le diagramme de classe global. Ensuite, l’architecture de notre application, les technologies utilisées et la méthodologie de conception Par la suite, nous avons dressée le carnet de produit et la répartition des sprints. Dans la partie suivante, nous allons détailler les différents sprints.

***CHAPITRE 3 :***

***INSCRIPTION*** ***&***

***AUTHENTIFICATION***

***& GESTION PROFILS***

***Plan***

1. Carnet du produit (Backlog du sprint 01) . . . . . . . . . . . . . . . .
2. Analyse . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
3. Conception . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
4. Réalisation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

Introduction :

Dans ce chapitre, nous allons détailler le travail réalisé durant le premier sprint, il dure généralement entre 1 et 4 semaines c’est une suite de tâches que l’équipe Scrum doit bien effectuer pour un objectif préalablement fixé. En effet, chaque itération (Sprint) représente une vision distribuée de la période de la production livrable.

Le développement de chaque sprint passe par les étapes d’analyse, de conception et de réalisation.

Le premier sprint vise la réalisation des items « Authentification, Inscription, et Gérer Profils ».

Carnet du produit (Backlog du sprint 01) :

L’équipe Sacrum doit obligatoirement se réunir pour définir le but du sprint avant de le commencer. Ce but doit être défini en terme métier pour qu’il soit compréhensible par les membres en dehors de l’équipe.

Il s’agit de répondre à une question fondamentale « pourquoi faisons-nous ce sprint ?»

Le but du présent sprint est de réaliser les trois Items :

* + S’inscrire
  + S’authentifier
  + Gestion Profils

Il est le temps maintenant de décider quelles histoires de notre backlog seront incluses dans le backlog du sprint. Les estimations des histoires sont définies en jour.

Le but de ce sprint est de réaliser des fonctionnalités basiques de la plateforme

|  |  |
| --- | --- |
| **ID U. S** | **User Stories** |
| **1.1** | En tant qu’un visiteur, je veux m’inscrire sur la plateforme en créant mon compte |
| **2.1** | En tant que Admin ou Utilisateurs (Enseignant, Parent, Etudiant), je peux me connecter pour accéder à mon espace. |
| **3.1** | En tant que Admin ou Utilisateurs (Enseignant, Parent, Etudiant), je peux consulter mon profil pour vérifier mes données |
| **3.2** | En tant que Admin ou Utilisateurs (Enseignant, Parent, Etudiant), je peux modifier mon profil pour mettre à jour mes informations |

Tableau 3.1 : user stories du sprint1

Le tableau suivant représente le Backlog du premier Sprint :

|  |  |  |  |
| --- | --- | --- | --- |
| **Items** | **User Stories** | **Tâches** | **Estimation** |
| **S’inscrire** | M’inscrire sur la  Plateforme | Créer les désigne | 2 jours |
| Contrôle sur les champs | 1 jours |
| Rendre l’interface « Responsive » | 2 jours |
| Relier le front end & back end avec API | 1 |
| Créer le base de donnée | 3 jours |
| **S’authentifier** | Me connecter à  Mon espace | Créer les désigne | 2 jours |
| Contrôle sur les champs | 1 jours |
| Créer le base de donnée | 2 jours |
| Relier le front end & back end avec API | 1 jours |
| Rendre l’interface « Responsive » | 2 jours |
| **Gestion Profil** | Consulter mon  Profil | Créer la page et les désigne(jsx,css,js) | 3 jours |
| Rendre l’interface « Responsive » | 1 jour |
| Afficher les données (API) | 1 jour |
| Modifier mon  Profil | Créer la fonctionnalité « Modifier profil » | 3 jours |

Tableau 3.2 : Le back log du 1er sprint

Notre premier sprint s’étalera donc sur 25 jours.

Analyse :

Dans cette section, nous allons présenter les activités principales, présentons le diagramme de cas d’utilisation raffiné de sprint 01 avec une description textuelle.

Diagramme de cas d’utilisation raffiné de sprint 01 :

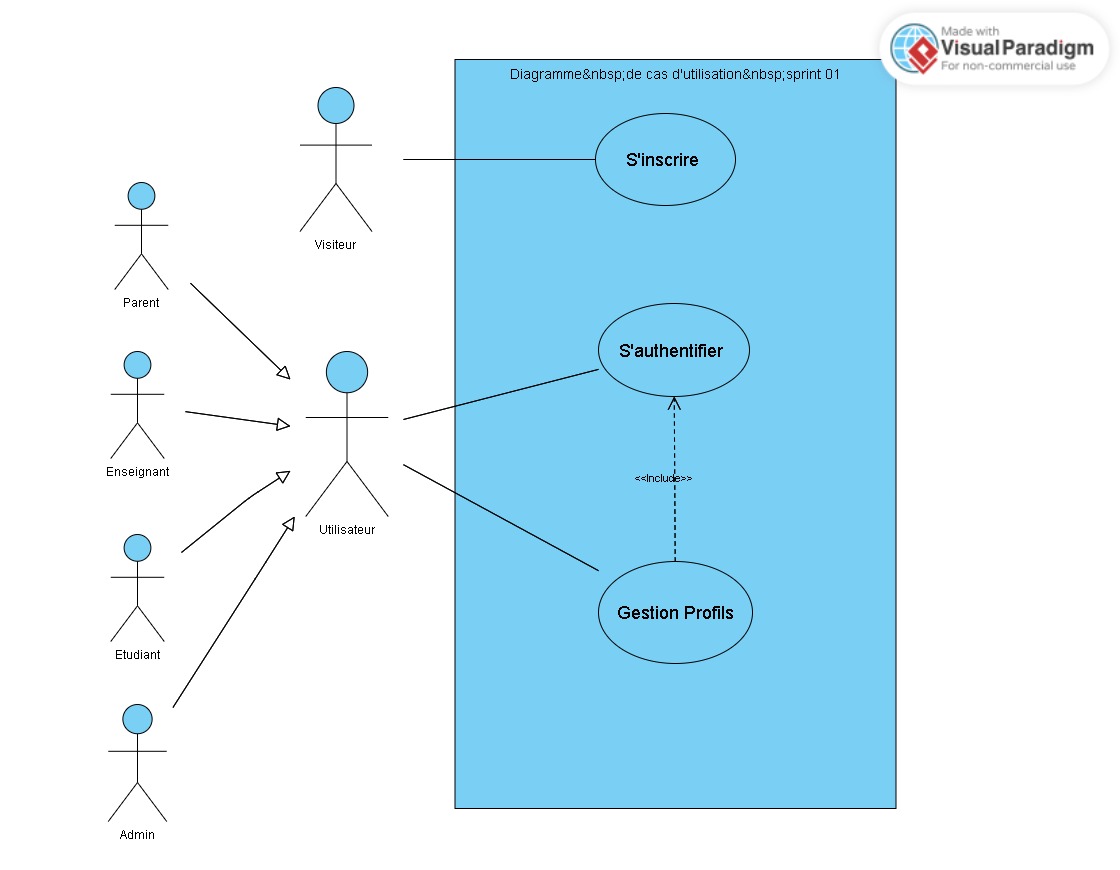


Figure 3.1 : Diagramme des cas d’Utilisation sprint 1.

Item « S’inscrire » :

|  |  |
| --- | --- |
| **Description textuelle de cas d’utilisation « S’inscrire »** | |
| **Titre** | S’inscrire |
| **Acteur** | Visiteur |
| **Prés condition** | Visiteur non inscrit |
| **Post condition** | Visiteur inscrit et possède un compte soit Parent ou Etudiant |
| **Description du scenario nominal** | 1. Le visiteur remplit le formulaire d’inscription 2. Le visiteur confirme la saisie de ses données 3. Le système enregistre la demande et affiche un message de confirmation 4. Le système effectue une redirection vers la page d’authentification. |
| **Scenarios alternatifs** | **les informations sont incorrectes :**le système affiche un message d’erreur et l’enchainement se démarre à l’action1 |

Tableau 3.3: Description Textuelle du cas d’Utilisation « S’inscrire »

Item « S’authentifier » :

|  |  |
| --- | --- |
| **Description textuelle de cas d’utilisation « S’authentifier »** | |
| **Titre** | S’authentifier |
| **Acteur** | Administrateur  Enseignant  Parent  Etudiant |
| **Prés condition** | Utilisateur non authentifié |
| **Post condition** | Utilisateur authentifié |
| **Description du scénario nominal** | 1. L’utilisateur saisie son nom d’utilisateur et mot de passe. 2. L’utilisateur confirme la saisie des données. 3. Le système vérifie la saisie des données. 4. Le système affiche l’interface d’accueil de l’utilisateur avec un message de succès |
| **Scénarios alternatif** | **Les informations sont incorrectes :** l’enchainement démarre à l’action 3, le système affiche un message d’erreur pour que l’utilisateur vérifie ses données. |

Tableau 3.4 : Description Textuelle du cas d’Utilisation « S’authentifier »

Item « Gestion Profil » :

Raffinement du cas d’utilisation « Gestion Profil » :

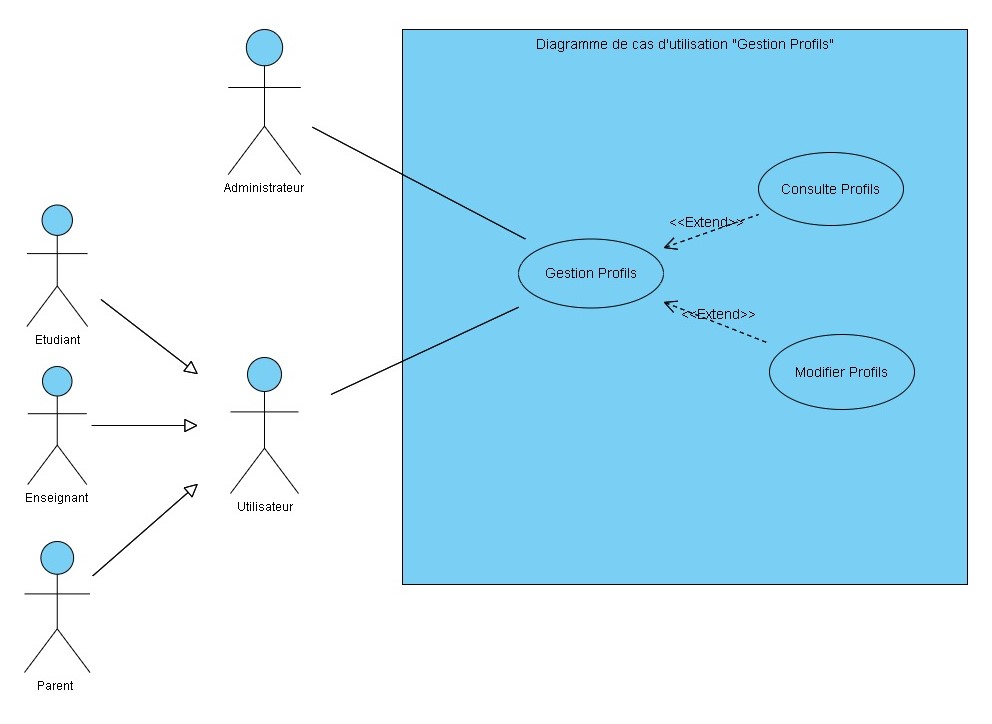


Figure 3.2 : Diagramme des cas d’Utilisation « Gestion Profils ».

Descriptions Textuelles du cas d’Utilisation « Gérer Profil » :

|  |  |
| --- | --- |
|  | **Description textuelle de cas d’utilisation « Gérer Profil »** |
| **Titre** | Gérer profil |
| **Acteur** | Utilisateur (Enseignant, Parent, Etudiant) ou Admin |
| **Prés condition** | Utilisateur ou Admin authentifié |
| **Post condition** | Profil modifié |
| **Description du scénario nominal** | 1. L’utilisateur ou l’admin demande la page de modification de profil 2. Le système affiche le formulaire et valide 3. L’utilisateur ou l’admin modifie les informations et valide 4. Le système vérifie les données saisies 5. Le système enregistre les données et affiche un message de réussite |
| **Scénarios alternatif** | **L’information est incorrecte :** L’enchainement démarre à l’action 2, les système affiche un message d’erreur pour que l’utilisateur vérifie ses donnés |

Tableau 3.5 : Description Textuelle du cas d’Utilisation « Gestion Profils »

Conception :

La conception est l’activité suivante pour ce sprint, elle lui donne une architecture et une forme bien définit. Dans cette partie nous commençons par la conception statique des différent cas d’utilisation déjà détaillées dans la partie précédente par le diagramme de séquence et le diagramme des classes participantes.

Diagramme de séquence :

Un diagramme de séquence est un type de diagramme d'interaction, car il décrit comment et dans quel ordre plusieurs objets fonctionnent ensemble. Ces diagrammes sont utilisés à la fois par les développeurs logiciels et les managers d'entreprises pour analyser les besoins d'un nouveau système ou documenter un processus existant. Les diagrammes de séquence sont parfois appelés diagrammes d'événements ou scénarios d'événements.[[15]](#footnote-15)

En se référant aux descriptions textuelles et le diagramme de cas d’utilisation de la section président nous présentons les diagrammes de séquences adéquats dans ce qui suit.

Diagramme de Séquence S’inscrit :



Figure 3.3 : Diagramme de Séquence « S’inscrit »

Diagramme de Séquence Connexion :

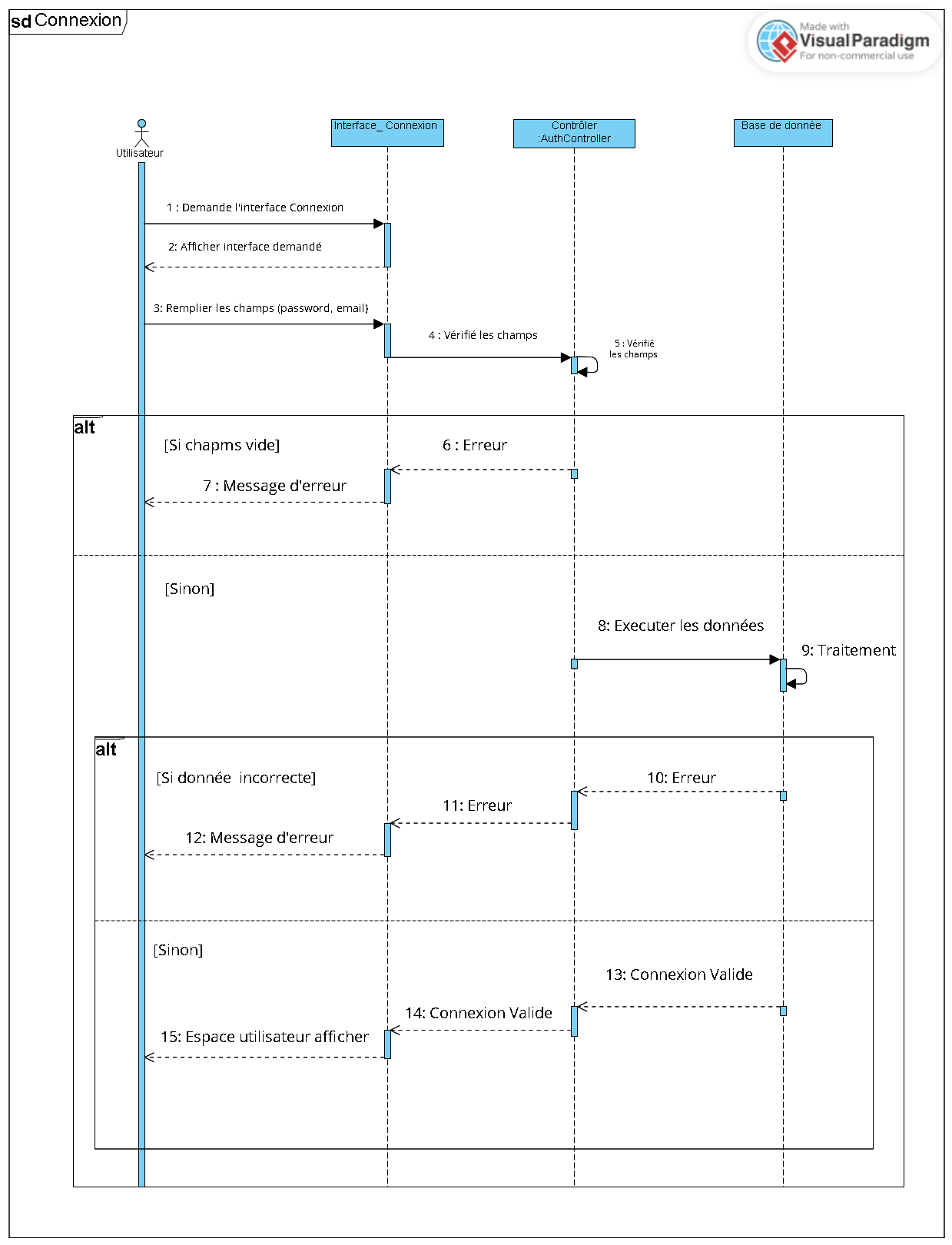


Figure 3.4 : Diagramme de Séquence « Connexion »

Diagramme de Séquence Gestion Profil :

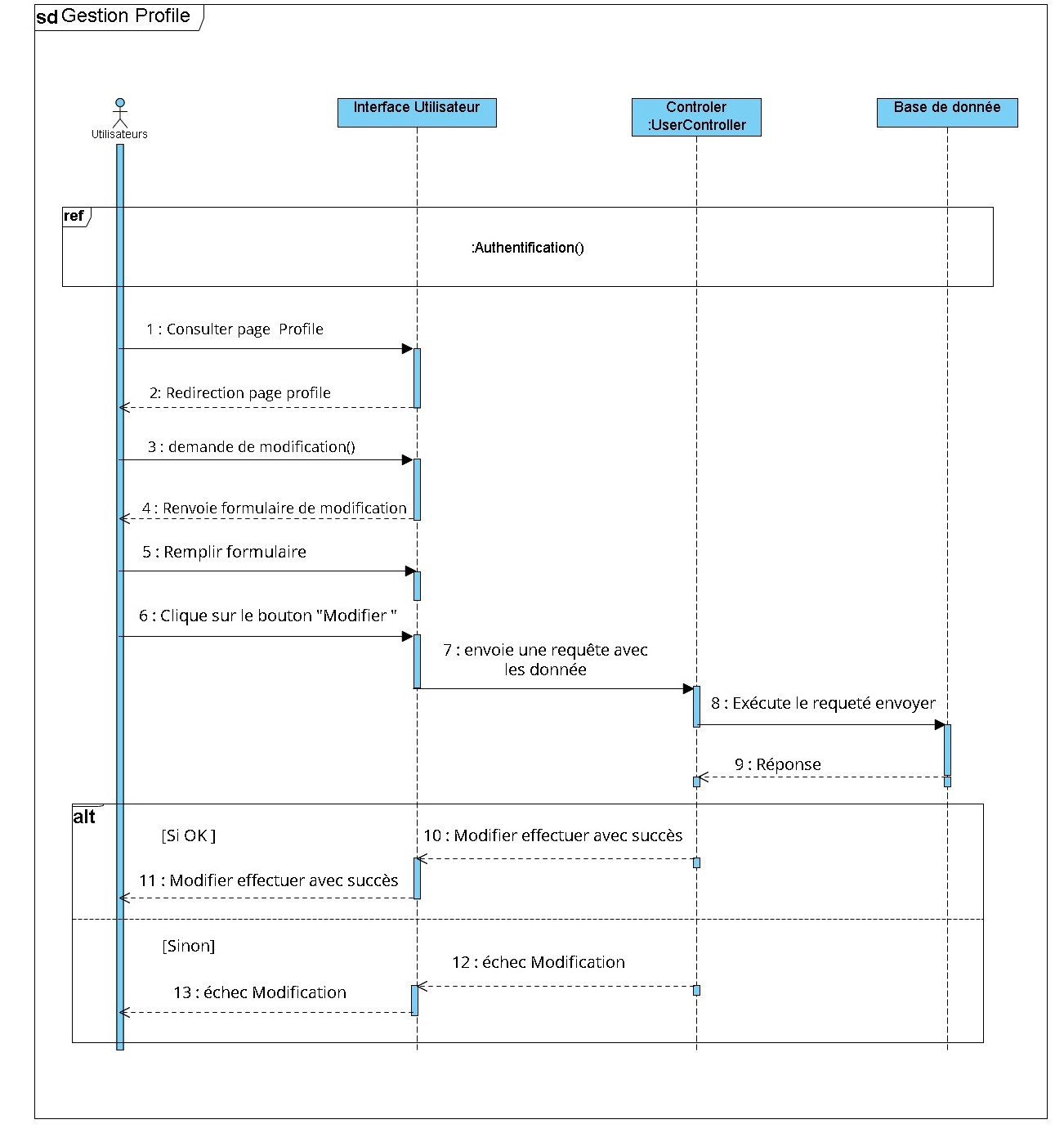


Figure 3.5 : Diagramme de Séquence « Gestion Profils »

Diagramme Des Classes Participantes De « Sprint 1 » :

Les diagrammes de classes sont les collection d’élément de modélisation statique qui montre la structure d’un modèle. Vous pouvez utiliser des diagrammes de classes pour modéliser les objets qui constituent le système, pour afficher les relations entre les objets et pour décrire ce que ces objets font et les services qu'ils fournissent.

Tout au long de nos sprints, nous essayerons de construire ces diagrammes au fur et à mesure en ajoutant les différentes classes déduites.

Les figures suivantes représentent les diagrammes de classe relatif à ce sprint

Diagramme de classe de conception « Inscription » :

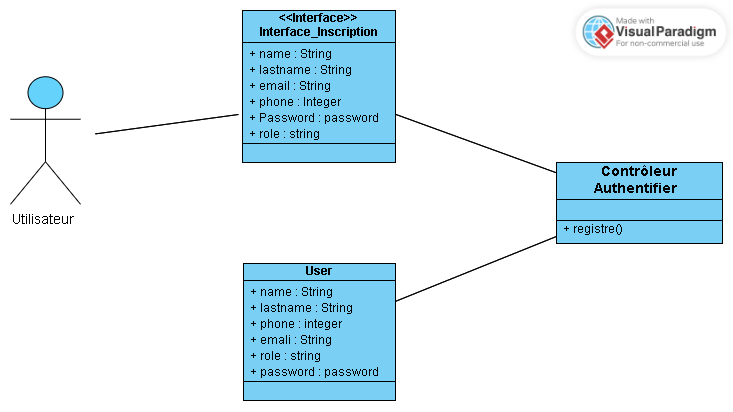


Figure 3.6 : Diagramme de classe au niveau analyse « Connexion »

Diagramme de classe de conception « Connexion » :

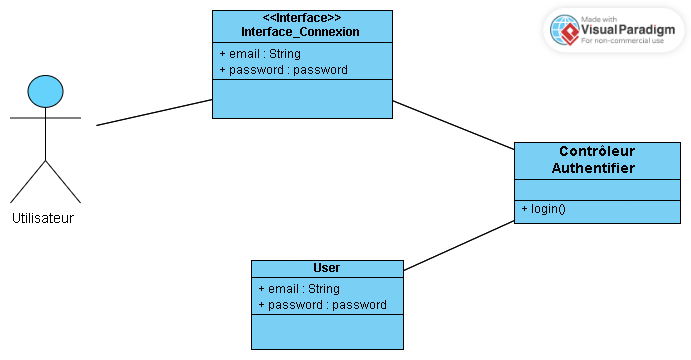


Figure 3.7 : Diagramme de classe au niveau analyse « Connexion »

Réalisation Du Sprint 1 :

La conception des interfaces de l'application est la première étape et le très importante puisque toutes les interactions entre les utilisateurs et le cœur de l'application passent à travers ces interfaces, on doit alors guider l'utilisateur avec les messages d'erreurs et de notification, ainsi présenter un système complet. Dans cette partie, nous allons présenter quelques interfaces de l'application avec des captures d’écran.

La page principale est accessible à tous les visiteurs qui n'a pas des comptes ou ne pas fait l'étape de l’inscription. Il s’agit de la première interface que vous trouverez lors du l’ouverture de la plateforme contenant des informations générales.

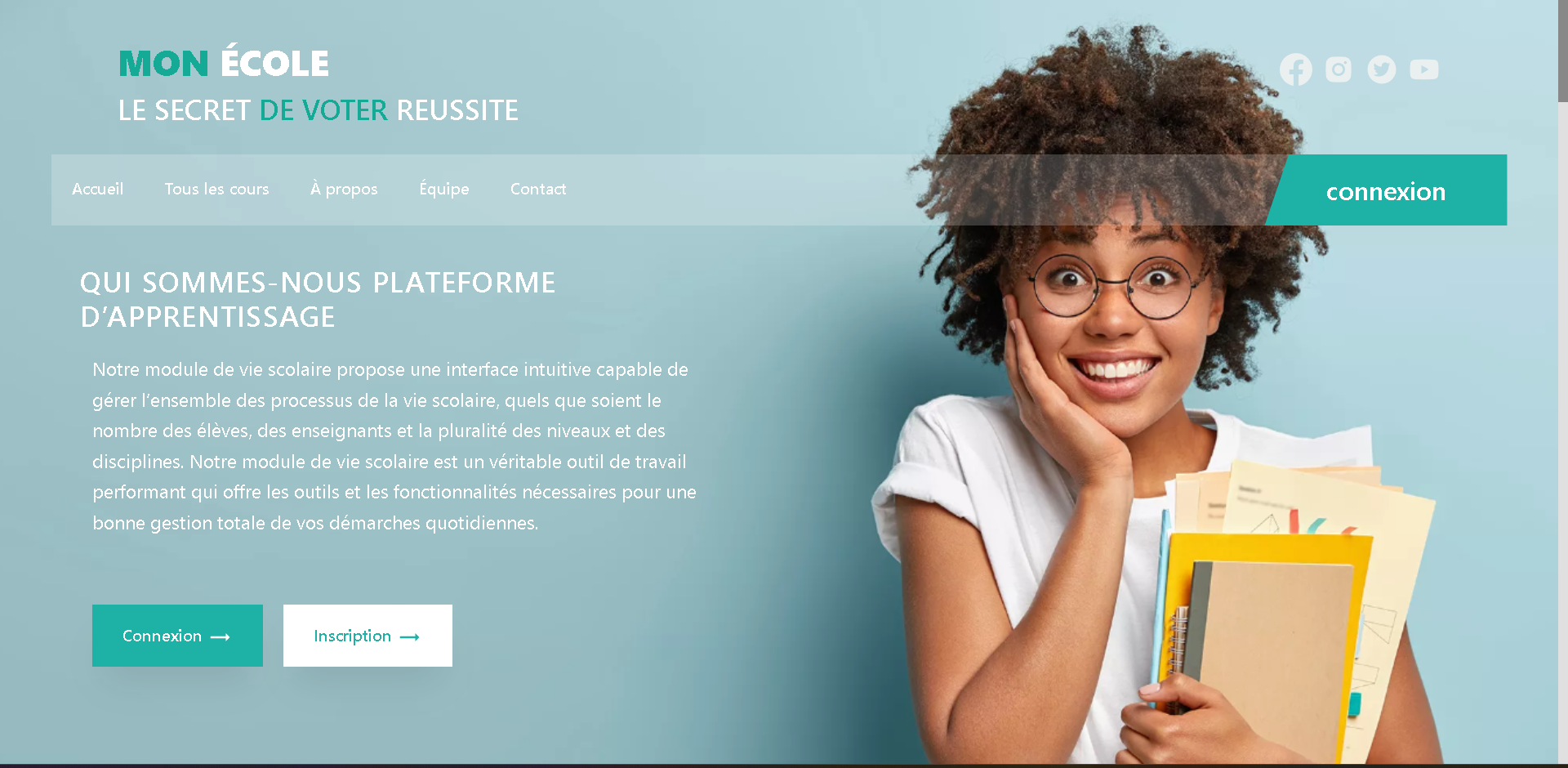


Figure 3.7 : Page D’accueil Visiteur.



Figure 3.8 : Page à propos Visiteur.



Figure 3.9 : Page à propos Visiteur.

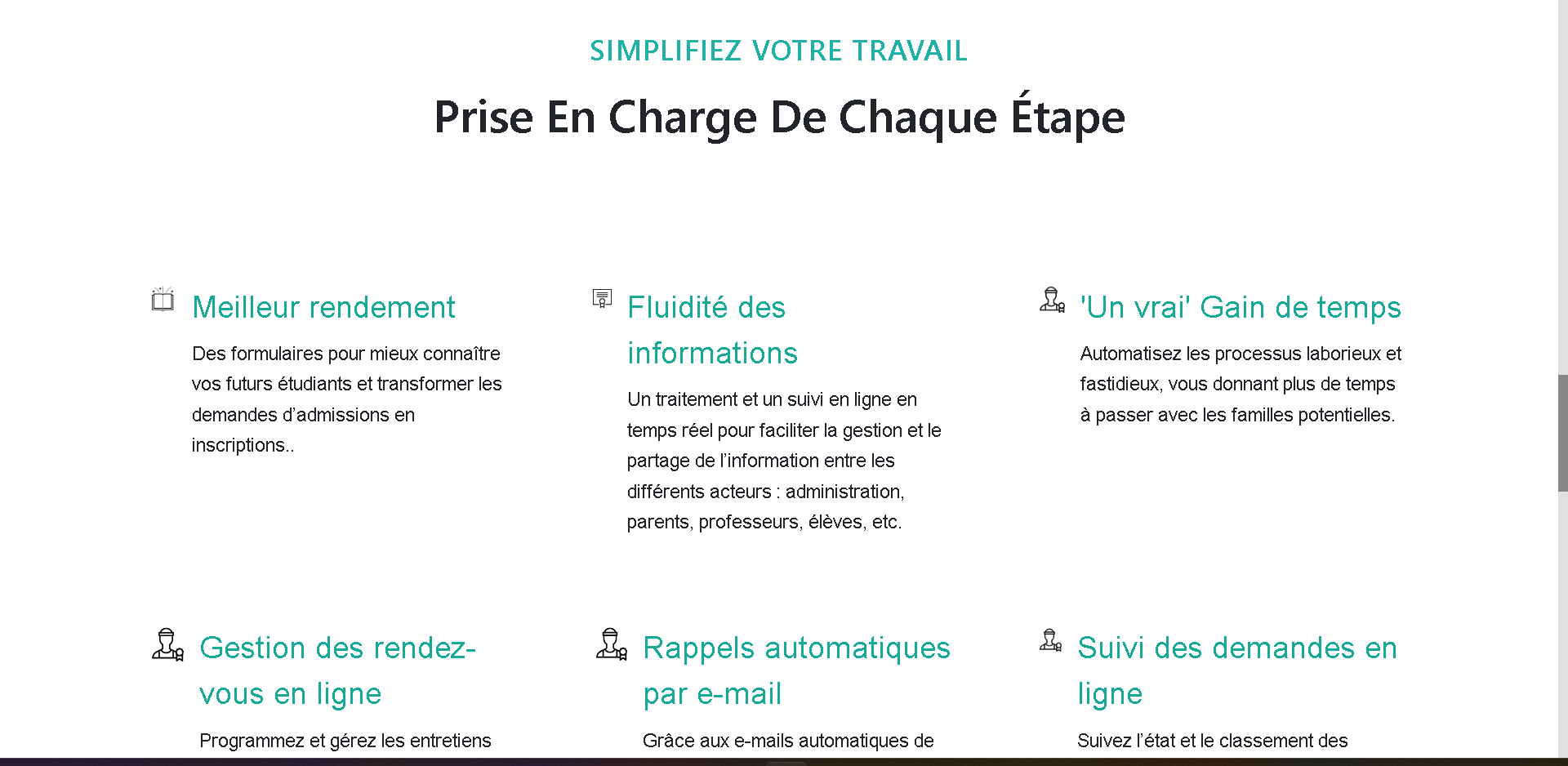


Figure 3.10 : Page à propos Visiteur.

Un visiteur doit créer un compte, pour s’inscrit il doit remplir le formulaire suivant :

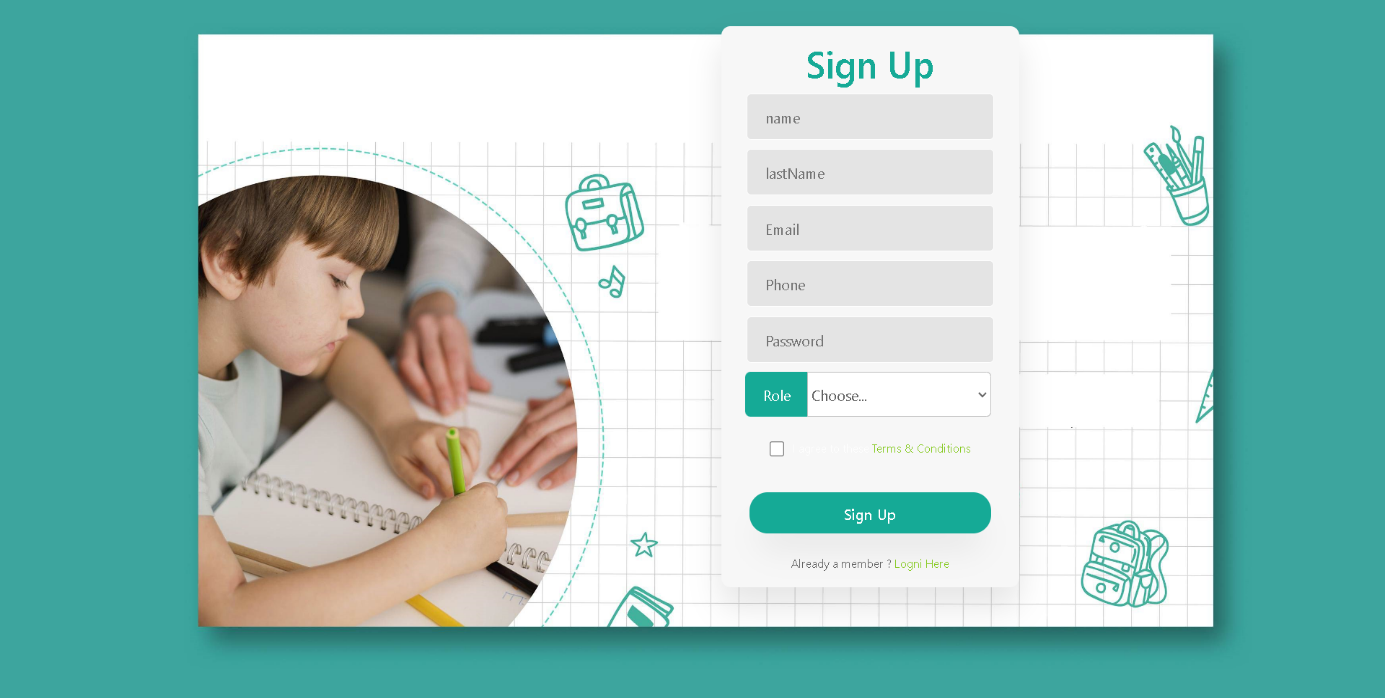


Figure 3.11 : Interface D’inscription

Une fois l’utilisateur inscrit, il est redirigé vers une page de Connexion :

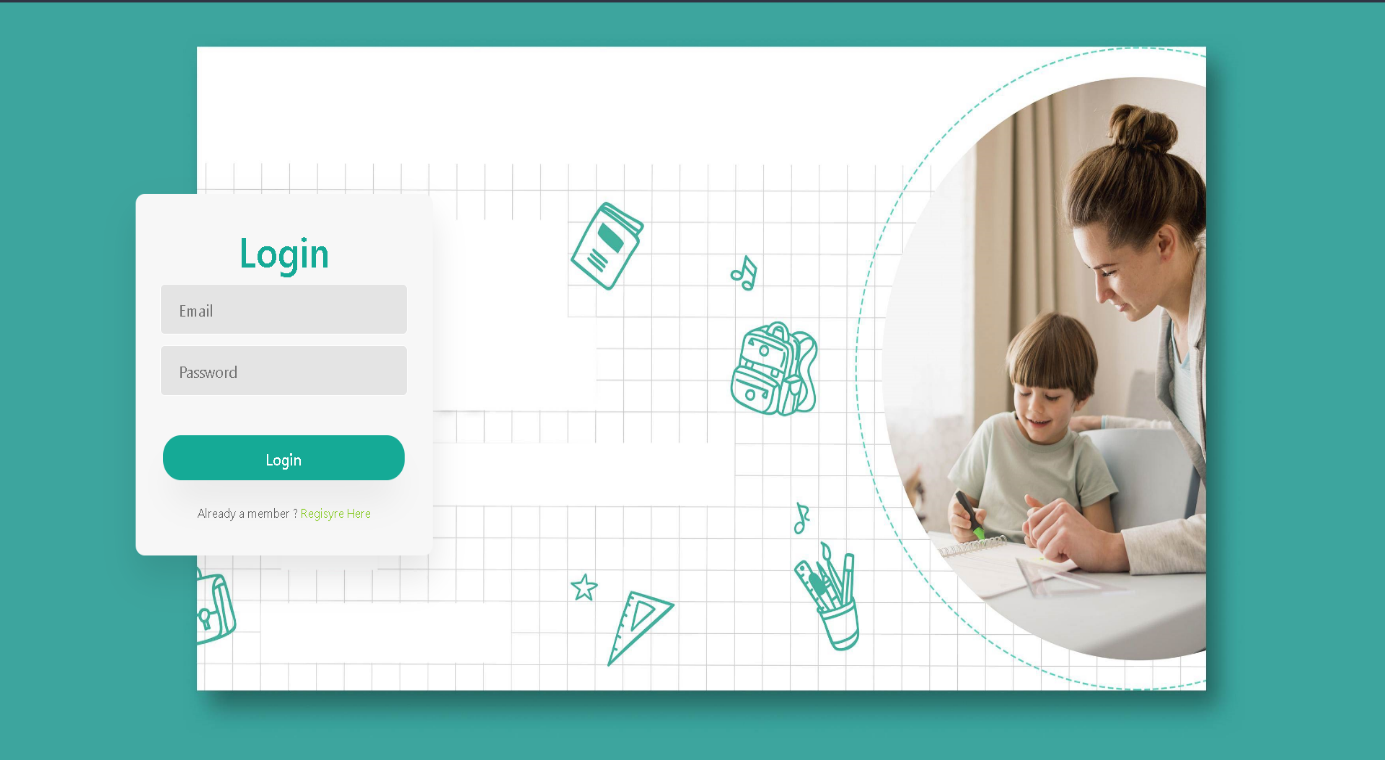


Figure 3.12 : Interface De Connexion

Avec un nom d’utilisateur et un mot de passe valide, le système redirige l’utilisateur vers son compte à l’interface Profil il peut afficher les informations de son compte et faire des modifications.

Exemple Compte Admin

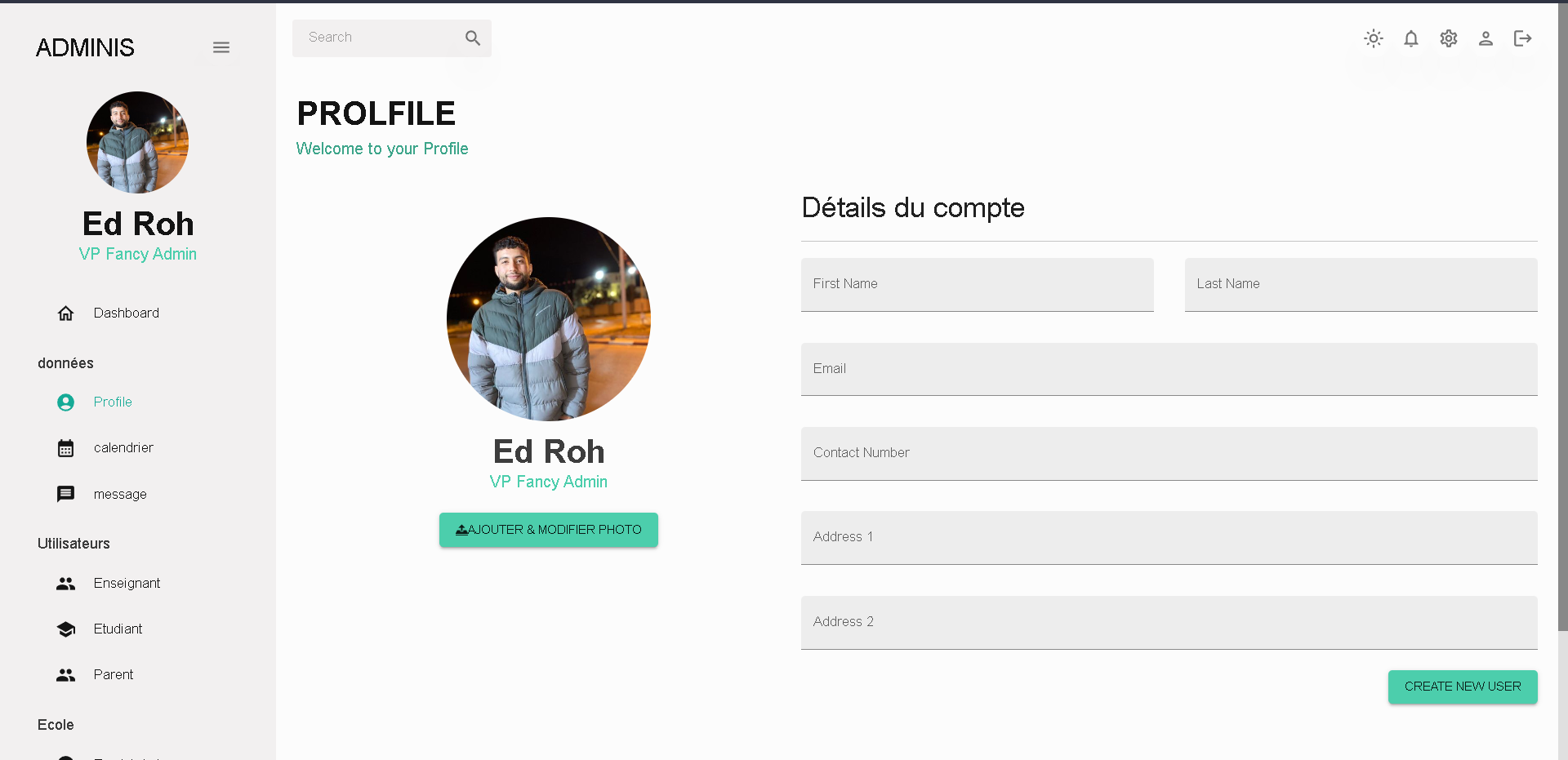


Figure 3.13 : Interface Du Paramètre Du Profile « Light » mode.

Il Ya deux mode pour les compte Admin « darck » & « Light » mode exemple pour « darck » mode :

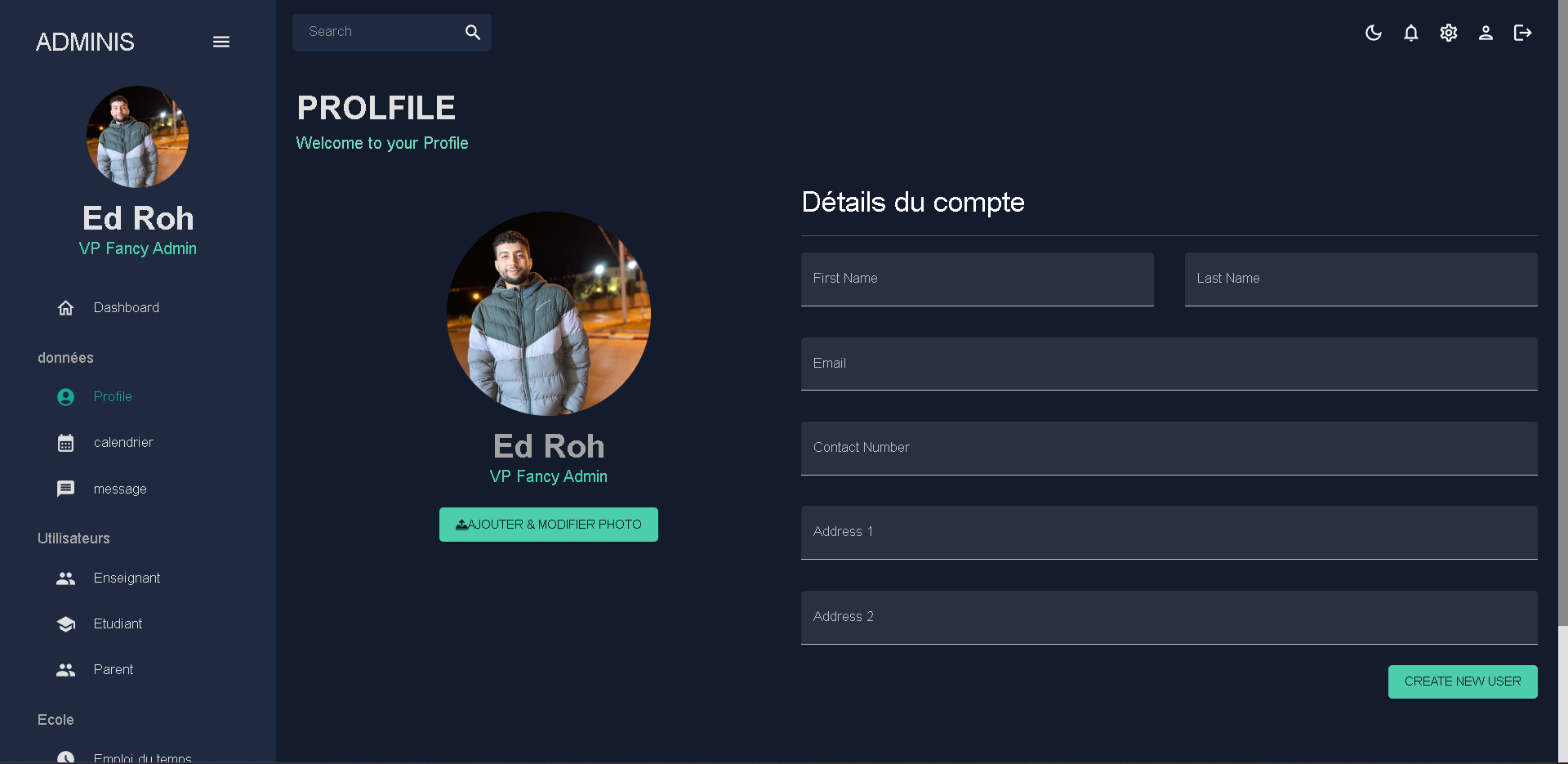


Figure 3.14 : Interface Du Paramètre Du Profile « Darck » mode.

CONCLUSION :

Tout au long de ce chapitre, nous avons présenté le premier sprint en passant par l’analyse, la conception, la réalisation.

Dans le chapitre suivant, nous dévoilons les fonctionnalités du deuxième Sprint.

1. De l’importance de l’intégration pédagogique des technologies de l’information et de la communication (TIC) par les enseignants du primaire, du secondaire et du tertiaire [↑](#footnote-ref-1)
2. https://ineumann.developpez.com/tutoriels/alm/agile\_scrum/ [↑](#footnote-ref-2)
3. https://ineumann.developpez.com/tutoriels/alm/agile\_scrum/ [↑](#footnote-ref-3)
4. https://scrumguides.org/scrum-guide.html [↑](#footnote-ref-4)
5. https://scrumguides.org/index.html [↑](#footnote-ref-5)
6. https://www.uml.org/what-is-uml.htm [↑](#footnote-ref-6)
7. "https ://www.scrum.org/" [↑](#footnote-ref-7)
8. "https ://www.scrum.org/" [↑](#footnote-ref-8)
9. "https ://www.scrum.org/" [↑](#footnote-ref-9)
10. https://urlz.fr/lc6j [↑](#footnote-ref-10)
11. https://www.google.tn/books/edition/Programmer\_en\_HTML/I8E8PQAACAAJ?hl=fr [↑](#footnote-ref-11)
12. https://www.google.tn/books/edition/JavaScript/7ZzQwAEACAAJ?hl=fr [↑](#footnote-ref-12)
13. https://www.google.tn/books/edition/Laravel\_Up\_Running/HcqPDwAAQBAJ?hl=fr&gbpv=0 [↑](#footnote-ref-13)
14. <https://books.google.tn/books?hl=fr&lr=&id=NZCKCgAAQBAJ&oi=fnd&pg=PR6&dq=react+js+definition&ots=KBzpSkFD3b&sig=BcatYKq7uJn94h3kNJz->

    Q9U28P0&redir\_esc=y#v=onepage&q=react%20js%20definition&f=false [↑](#footnote-ref-14)
15. https://www.lucidchart.com/pages/fr/diagramme-de-sequence-uml [↑](#footnote-ref-15)