# INTRODUCTION GENERALE

Le stage est un moyen qui permet à l’étudiant de s’intégrer dans la vie professionnelle en vue d’avoir une idée générale sur le fonctionnement de l’entreprise. A ce niveau, l’objectif des instituts supérieurs des études technologies est permet à l’étudiant d’enrichir ses connaissances et de s’adapter à l’environnement de l’entreprise.

L’objectif de ce stage est de faire une intervention pratique qui me permet d’applique mes connaissances scientifiques et technique relatives au domaine de l’informatique, et de tester mes facultés d’analyse et de synthèse.

Etant donné la forte croissance du marché des applications web, aujourd’hui, le développement d’application intéresse énormément d’utilisateurs et il est reconnu dans la plupart des domaines y compris les domaines de l’éducation en ligne.

Puisque l'apprentissage en ligne permet aux étudiants d'apprendre de leur propre façon et de [manière autorégulée](https://www.easy-lms.com/fr/base-connaissances/apprentissage-formation/apprentissage-autoregule/item10384), il aide à absorber plus d'informations. Si une personne a besoin de passer plus de temps sur un point en particulier, elle peut l'étudier aussi longtemps et autant de fois qu'elle le souhaite. Alors Nous pouvons conclure que l’éducation en ligne augmente la productivité.

En effet, les logiciels et les applications web dans le domaine l’éducation en ligne connaissent actuellement un essor important. Leurs utilisations se multiplient et ces produits peuvent être très bien les diversifier. C’est dans ce contexte, que s’intègre notre projet de fin d’étude effectué au sein de la société « SDI Kairouan » et qui consiste à concevoir et à réaliser une application web d’éducation en ligne « E-Learning ».

Le présent rapport se résume en cinq chapitres comme la suite :

Le premier chapitre « **Cadre général du projet** » permet de placer le projet dans son contexte général. Dans ce chapitre introductif nous présentons le cadre du projet incluant la présentation de l’organisme d’accueil, la problématique, la solution proposée ainsi que la méthodologie adoptée en se basant sur la méthodologie Scrum.

Le second chapitre «**Préparation de projet**» pour expliquer les besoins du système, identifier les acteurs, le diagramme de cas d’utilisation général, le pilotage du projet ainsi que la planification des sprints.

Par la suite, nous aborderons le troisième chapitre «**Sprint 0**» qui montre la préparation de l’environnement de travail puis l’architecture utilisé.

Le quatrième chapitre «**Release 1**» qui consiste à présenter le «**Sprint 1**» et le «**Sprint 2**».

Le cinquième chapitre «**Release 2**» qui s’intéresse de présenter le «**Sprint 3**», «**Sprint 4**», «**Sprint 5**».

Finalement, nous allons conclure nos travaux avec une conclusion générale et une présentation des perspectives qui résument tout le travail et mènent notre projet à terme.

# Chapitre 1 : Cadre générale du projet

## Introduction

Dans ce chapitre, nous présentons le contexte de travail et les objectifs de notre projet. Nous allons commencer par la présentation de l’organisme d’accueil ainsi que son organisation interne. Ensuite, nous passons à l’étude de l’existant. Nous finirons par la définition du langage de modélisation et la méthodologie de travail choisie.

## Présentation du cadre du stage

### Présentation de la société

#### Fiche entreprise

La fiche d'entreprise est un document qui permet de présenter les détails de la société :

|  |  |
| --- | --- |
| **Secteur d’activité** | Informatique |
| **Taille de l’entreprise** | Entre 20 et 100 employés |
| **Catégorie** | Société privée locale |
| **Année de fondation** | 2015 |
| **Adresse** | Cyber Park Kairouan 3100 |

**Tableau 1 : Fiche d'identité de la société**

#### Présentation

**ETC Tunisie**, est une société OFFSHORE de développement informatique située à Mansoura, Gouvernorat Kairouan 3199, Tunisie, qui travaille en collaboration avec une Entreprise de Service Numérique (ESN) française, créée depuis plus de 07 ans, leader dans son domaine et qui met à disposition de sa clientèle son savoir-faire, ses services et ses produits afin de réaliser et réussir leurs projets digitaux.

**ETC Tunisie** fournit non seulement des solutions logicielles standards, capables de répondre aux besoins des clients, mais aussi des solutions sur mesure, s’appuyant sur ses propres outils, ses compétences projet, et son expertise métier sur le marché européen.

Spécialises-en :

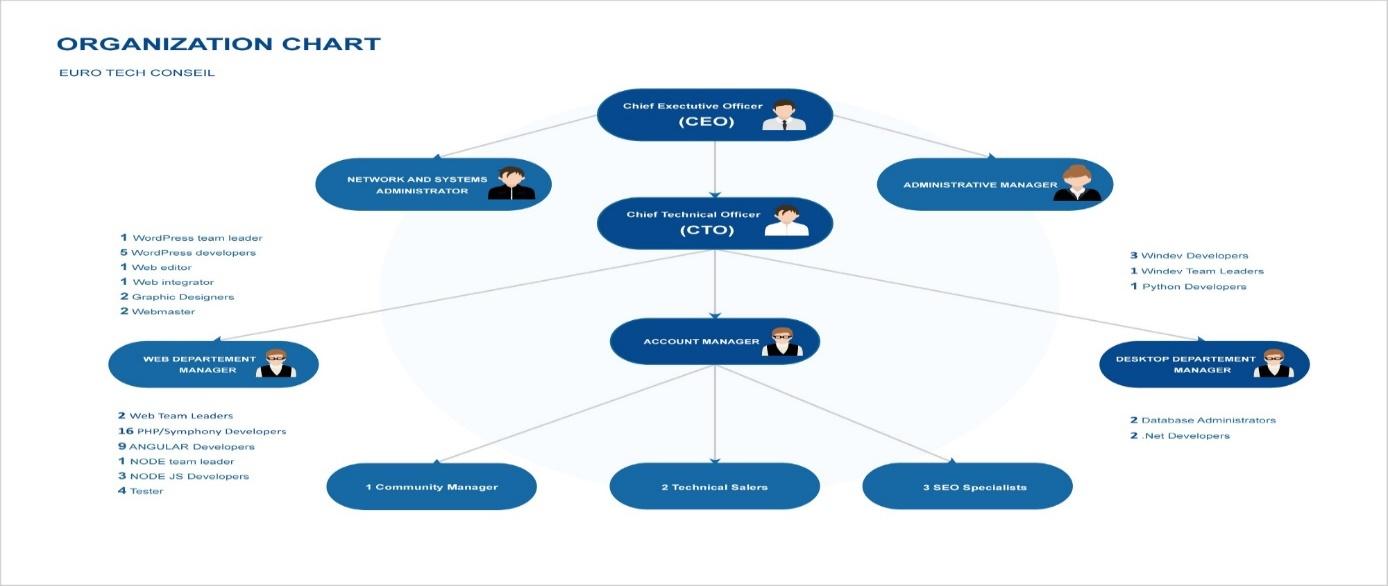
* Développement sur-mesure des logiciels.
* Développement sur-mesure des applications web et mobile.
* La Digitalisation des entreprises.
* Infogérance et web Marketing.
* Hébergement et Référencement
* Cloud Computing



**Figure 1 : Logo de la société ETC Tunisie**

#### Organigramme

Ci-dessous la figure 1 expose l’organigramme de la société. C’est une représentation schématique des liens fonctionnels, organisationnels et hiérarchiques d’une entreprise.



**Figure 2 : Organigramme d la société**

Dans le cadre de notre projet de fin d’études, On nous a affectés avec l’équipe WEB DEPARTEMENT MANAGER.

### Etude du Projet

Cette section est composée de trois parties, nous commençons par l’analyse de l’existant, puis nous introduisons les problématiques et nous finissons par la solution à proposer. L’étude du projet est une phase déterminante pour bien comprendre les défis mis en jeu et pour dégager les points forts et les points faibles de chacune des solutions. Elle est une étape très importante vu qu’elle procure une vision plus claire du sujet et une compréhension plus profonde des tâches à réaliser.

#### Analyse de l’existant

L’étude de l’existant est la phase la plus importante dans l’analyse d’un projet. Cette partie est primordiale pour la mise en route des projets informatiques, et qui consiste à expliquer le contexte général et qui permet de débloquer les différentes imperfections qui se trouvent dans le système actuel. Il existe plusieurs applications basées sur le spectre de fréquence.

En effet un existant riche et stable nous permettra d’avoir un aperçu concret qui servira de fondement pour le développement de notre application. Cette étude nous a permis de localiser les applications web que nous vous présentons dans la suite.

##### **Présentation du sujet**

Étant donné que **ETC Tunisie** est intéressée à fournir des programmes dans plusieurs domaines d’application qui peuvent contribuer à résoudre de nombreux problèmes et il répond aux besoins de l’utilisateur, et vue que l’apprentissage en ligne est devenu l’un des domaines les plus populaires d’aujourd’hui, alors **ETC Tunisie** décidé de lancer une plateforme d’apprentissage en ligne.

Cette solution consiste à appliquer les méthodes techniques, et outils de gestion spécifiques aux différentes étapes du projet, de l'évaluation de l'opportunité jusqu'à l'achèvement du projet.

##### **Problématique**

**ETC Tunisie** nous a proposé l'idée du projet, et elle nous donnant la permission d'ajouter des fonctionnalités qui pourraient répondre aux besoins de l'utilisateur, Donc, nous avons fait beaucoup de recherches et d'examens de nombreuses applications web tunisiennes existantes, pour analyser les problèmes et les défauts qui existent.

Après notre étude qu’on a faite nous n’avons constaté que la plupart des applications ont les problèmes suivants:

* Concernant les fonctionnalités nécessaires : le système s’appuie sur des fonctionnalités spécifiques et ils sont très basiques.
* Concernant la convivialité des interfaces graphiques : les interfaces du système sont très simples et basiques. Elles ne sont pas très ergonomiques.
* Concernant l’expérience d’utilisateur : il y a beaucoup des boutons, plusieurs champs à remplir pour qu’un utilisateur s’inscrire dans l’application, l’utilisateur n’a l’accès de consulter les formations ou les cours existent qu’après l’inscription, le paiement et main en main et il n’y a pas un système de panier.
* Concernant la performance : l’application est trop lente.
* Absence des parties d’évaluations dans l’application.
* Les formations sont déjà enregistrées et il n’y a pas des formations en ligne et un chat en temps réelle avec le formateur.

##### **Solution proposée**

À la suite de ces constatations, il paraît clair qu’il faut se pencher dès à présent sur le sujet et de commencer à réaliser une application avec les caractéristique suivantes :

* Enrichir les fonctionnalités de l’application.
* L’étudiant peut évaluer une formation ou un formateur.
* Une interface ergonomique qui assure la facilite de l’utilisation pour tous. elle est caractérisée par sa convivialité.
* Une expérience d’utilisateurs de haute qualité : un Etudiant non inscrit dans la plateforme peut consulter les formations, les titres des chapitres les notes d’une formation, il peut ajouter des formations au panier, valider le panier avec un système de paiement en ligne.
* Fournir des formations en ligne : Le formateur et ses étudiants peuvent s’interagissent dans une vidéo de conférence et un chat en temps réels.

Ce travail rentre dans le cadre de notre projet de fin d’études qui vient conclure notre formation en licence en technologies de l’informatique à l’Institut Supérieur des études technologique de Kairouan (ISET Kairouan).

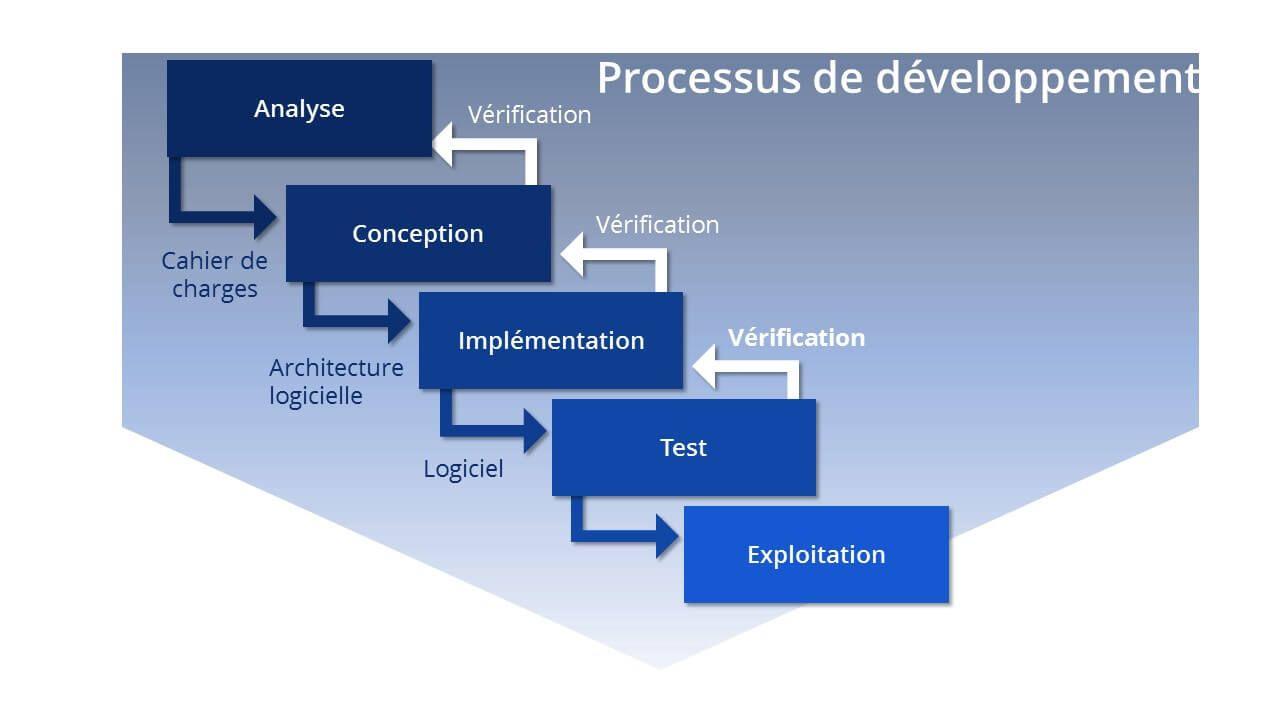
## Méthodologie de travail

Cette section est consacrée à la définition de notre approche méthodologique. Cette approche est essentielle afin de pouvoir comprendre les démarches suivies dans la conception et la réalisation de l’application.

### Méthodes classiques

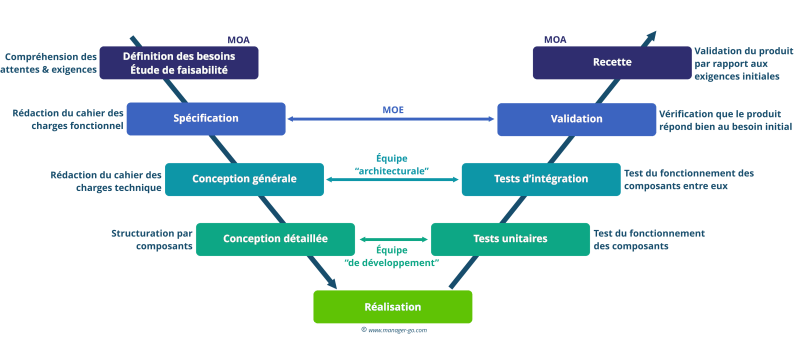
La méthode classique de gestion de projet, également appelée méthode traditionnelle ou méthode prédictive, est un modèle linéaire, où les différentes phases de développement suivent un ordre précis. Chaque tâche doit être terminée avant de passer à la tâche suivante. [1]

❃ **Cascade**: Le modèle en cascade ou "Waterfall" est une méthode de développement de produits qui consiste à diviser un projet en phases qui se succèdent de manière séquentielle suivant un flux descendant régulier. La séquence classique est composée de 5 grandes phases : Analyse (ou collecte des besoins), conception, mise en œuvre, test et maintenance, chacune se terminant par un contrôle et des livrables. [2]



**Figure 3 : cycle en cascade**

❃ **Le cycle en V**: en gestion de projet découle du modèle en cascade théorisé dans les années 1970, qui permet de représenter des processus de développement de manière linéaire et en phases successives. La lettre V fait référence à la vision schématique de ce cycle, qui prend la forme d’un V : une phase descendante suivie d’une phase ascendante. Le cycle en V associe à chaque phase de réalisation une phase de validation [3], comme l’illustre le schéma ci-dessous :



**Figure 4 : cycle en V**

### Les Méthodes Agiles

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. [4]

❃ **Extrême Programming (XP):** C’est une méthode de gestion de projet qui applique à l’extrême les principes du développement agile, c’est-à-dire se concentrer sur les besoins du clients, mettre en place un développement itératif et l’intégration continue. [5]

❃ **Processus unifié (UP):** C’est un processus de développement logiciel itératif, centré sur l’architecture, piloté par des cas d’utilisation et orienté vers la diminution des risques. [6]

❃ **Scrum:** La méthode SCRUM est une méthode agile créée en 2002, dont le nom est un terme emprunté au rugby qui signifie la mêlée. Elle s’appuie sur le découpage de projet en itérations nommées Sprint. Un Sprint peut avoir une durée qui varie généralement entre deux semaines et un mois. Avant chaque Sprint les taches sont estimées en temps et en complexité. Ces estimations permettent à la fois de planifier les livraisons mais aussi d’estimer le cout de ces taches auprès du client. [7]

Après les études effectuées sur les différentes méthodologies et pour assurer le bon déroulement des différentes phases de notre projet, nous avons choisi Scrum comme une méthodologie de conception et de développement.

## SCRUM

Ci-dessous la figure 5 qui présente l’architecture du **SCRUM** de la société.

C:\Users\zaidb\OneDrive\Bureau\scrumGrafique.png

**Figure 5 : Architecture SCRUM**

### Pourquoi SCRUM

Nous avons adopté Scrum comme méthodologie de gestion pour notre projet car elle offre plus d’avantages par rapport aux autres méthodes :

* + Scrum rassemble les deux côtés théorique et pratique et se rapproche beaucoup de la réalité.
  + Meilleur contrôle de qualité.
  + Le travail d’équipe est fortement impliqué.
  + Une excellente répartition des tâches.
  + Forte capacité d’adaptation aux changements et ajustements grâce à des itérations courtes.
  + Implication de client à la fin de chaque sprint.

### L’équipe SCRUM

La méthodologie SCRUM définit trois rôles :

* Product Owner

C’est un chef de projet en mode agile. Il est en charge de satisfaire les besoins des clients en menant à bien la livraison d’un produit de qualité. Il sert d’interface entre l’équipe technique, l’équipe marketing et les clients. Au sein de son groupe de travail, il est le premier responsable de la conception du produit. [8]

* Le SCRUM Master

Le Scrum Master est souvent un membre à part entière de l’équipe technique et/ou de l’équipe chargée de la gestion de projet. Il a pour mission de coacher les développeurs et tous les acteurs dans la création d’un produit, pour faciliter sa réussite. Pour cela, il met en pratique la méthodologie Scrum, au travers de conseils, de planning et de pistes d’itération. En somme, il est le garant du respect de la mise en pratique de cette méthode de travail. [9]

* SCRUM Team Members

Ces gens ont la responsabilité de réaliser le sprint et les livrables. Les participants doivent collaborer pour satisfaire aux exigences des clients.

### Les artéfacts dans SCRUM

Les principaux artéfacts dans SCRUM sont :

* Product Backlog

Il contient la liste des fonctions que le client veut en tant que User Stories. Il est visible à toutes les parties prenantes qui peuvent ajouter des éléments. Les priorités des éléments sont constamment mises à jour par le product Owner. Le backlog du produit de notre projet est présenté dans le chapitre spécification des besoins.

* Sprint Backlog

Il comprend les besoins à réaliser, négociés entre le chef de produit et l’équipe projet lors de la planification du sprint. Il est évident pour toute l’équipe qu’elle utilise comme référence dans le scrum quotidien.

### Les activités du sprint

Les principales activités dans SCRUM sont :

* Sprint planning meeting

La planification du sprint est une réunion critique et importante dans le cycle de vie du projet, et a lieu au début de Sprint. Elle permet à l’équipe projet et au responsable produit de définir l’ensemble des buts à atteindre au cours du Sprint en se basant sur le « backlog du produit » priorisé et d’estimer la capacité de production nécessaire pour atteindre ce but.

* Daily SCRUM

Chaque jour, les membres de l’équipe de projet passent en moyenne une quinzaine de minutes à discuter entre eux. Chaque participant rapporte ce qu’il a fait la veille et les obstacles qu’il a rencontrés. Nous discutons aussi des tâches à réaliser le jour de la réunion.

* Sprint Review Meeting

À la fin du sprint, l’équipe projet présente les fonctionnalités réalisées au cours du Sprint et recueille les feedbacks du product Owner et des utilisateurs finaux. C’est également le moment d’anticiper la portée des prochains Sprints et d’ajuster la planification de la release (nombre de Sprints restants).

* Sprint Retrospective Meeting

Une autre réunion se tient à la fin de chaque sprint. Le SCRUM master anime cette réunion. L’équipe de projet évalue leur comportement et leur rendement dans le processus de développement afin de prendre des mesures pour mieux gérer les prochains Sprints.

## Conclusion

Dans ce chapitre introductif, nous avons présenté plusieurs points, en commençant par la présentation de l’organisme d’accueil, ainsi que l’étude de l’existant, en finissant avec le langage et méthodologie de travail. Le chapitre suivant sera destiné pour la planification du projet.

# Chapitre 2 : Préparation de projet

## Introduction

Comme indiqué dans le chapitre précédent, nous avons choisi d’adopter la méthodologie Scrum pour la conception de notre plateforme. Dans ce chapitre

« **Préparation de projet** » Les travaux à faire donc dans ce chapitre ce sont : définir les différentes fonctionnalités en identifiant les besoins, les rôles des utilisateurs, ensuite identifier l’équipe SCRUM et leurs rôles et enfin élaborer le Product Backlog. À la fin de ce chapitre, nous allons planifier la répartition initiale des sprints.

## Identification des acteurs

### Les acteurs

Les acteurs sont des entités externes au système. Il représente une personne ou un autre système informatique qui souhaite accéder à un ou plusieurs des services fournis par l’interface. Il interagit avec le système en envoyant ou en recevant des messages. Notre application implique les différents acteurs suivants :

* Administrateur

C’est la personne qui possède le privilège du plus haut niveau.

* Formateur

Consiste à Présenter les formations et gérer ses contenus.

* Candidat

C’est lui qui peut acheter la formation et profiter de son contenu.

## Spécification des besoins

La phase de spécification des besoins, est la première étape du processus de développement que nous avons adopté. En effet, elle formalise et détaille ce a été ébauche au cours de l’étude préliminaire, et permet de dégager l’étude fonctionnelle du système. Elle permet ainsi d’obtenir une idée sur ce que va réaliser le système en termes de métier (comportement du système).

### Les besoins fonctionnels

L’analyse fonctionnelle est une approche qui s’implique pour caractériser les fonctionnalités d’un produit afin de répondre aux besoins des utilisateurs. En effet, chacune de ces exigences tient compte des attentes des différents utilisateurs du système conçu. Par conséquent, nous définissons toutes les exigences fonctionnelles à mettre en œuvre dans le projet.

Le système à concevoir doit permettre à l’utilisateur d’effectuer les opérations suivantes:

* Besoins fonctionnelles communs entre tous les utilisateurs de l’application
* Authentification
* L’Administrateur doit s’authentifier avant de passer à son dashboard.
* Le formateur doit s’authentifier avant de passer à son dashboard.
* Le Candidat doit s’authentifier avant de valider un panier ou d’accéder à son profil.
* Visiter et Modifier le profile
* Il s’agit d’un outil permettant à l’utilisateur de la plateforme de Visiter et de modifier son profil.
* Besoins fonctionnelles n’intéressent que l’administrateur
* Gestion des formations, des catégories, et des groupes (les groupes sont spécifiquement pour les formations en ligne et on va expliquer ce concept au plus tard)
* Il s’agit d’un outil permettant d’effectuer les opérations de gestion telles que l’ajout, la suppression, la modification et la consultation.
* Le contrôle de la session d’une formation (ouvrir ou fermer).
* Gestion des formateurs
* L’administrateur peut ajouter, supprimer, désactiver le compte d’un formateur et visiter la liste de tous les formateurs.
* Créer un sondage
* Pour s’assurer que lors de la création d’une formation en ligne il y a un nombre approprié des candidats qui y participeront, l’administrateur doit créer un sondage pour les utilisateurs.
* Consulter quelques statistiques de la plateforme
* Il s’agit d’un outil permettant à l'administrateur de consulter des statistiques de la plateforme telles que le nombre d’inscription des candidats dans une durée spécifique, un courbe qui affiche les meilleurs formateurs selon l'évaluation des candidats et un courbe qui affiche les meilleures formations.
* Besoins fonctionnelles n’intéressent que le formateur
* Consulter ses formations courantes
* Il s’agit d’un outil permettant au formateur de consulter les formations qu’ils présentent actuellement et voire ses détails.
* Contrôler la visibilité publique d’une formation
* Lorsque le contenu de la formation est complètement rempli, le formateur a le droit de mettre la formation en publique ou lorsque le formateur découvre une faute dans un chapitre d’une formation spécifique alors il met cette formation invisible pour les consultants de la plateforme puis corrige la faute et finalement met cette formation en publique.
* Consulter quelques statistiques
* L’application fournit une vue pour qu’un formateur puisse voir le nombre de toutes les formations qui a présenté, son évaluation, l’évaluation d’une formation spécifique et le taux de croissance de son expérience.
* Ajouter des tâches à fait
* L’application fournit un formulaire pour qu’un formateur peut ajouter ses tâches a fait d’une formation qui a charge de la présenter, aussi il fournit un calendrier pour afficher l’ensemble des tâches de chaque formation.
* Gérer les Chapitres et ses contenues (vidéos, fichiers, documents, quizz)
* Il s’agit d’un outil permettant d’effectuer les opérations de gestion telles que l’ajout, la suppression, la modification et la consultation.
* Besoins fonctionnelles n’intéressent que le Candidat
* S’inscrire
* Pour bénéficier des avantages spécifiques de l'application, le Candidat doit créer son compte.
* Consulter et filtrer la liste des formations
* Le candidat peut consulter les formations.
* Pour offrir la meilleure expérience utilisateur, le Candidat peut effectuer des filtres pour obtenir des formations.
* Voire les meilleures formations
* L’application fournit une vue pour afficher au candidat les meilleures formations notées.
* Consulter les détails d’une formation
* Le Candidat peut consulter les avis des autres candidats, les titres des chapitres, et tous les détails d’une formation sans s’authentifier.
* Gérer un panier
* Le Candidat peut ajouter une formation au panier ou la supprimer, peut aussi vider tout le panier.
* Valider un panier par un système de paiement en ligne
* Après l’ajout des formations au panier le Candidat a le choix de valider son panier à l'aide d'un système de paiement en ligne.
* Répondre à un sondage
* Un candidat peut répondre à un sondage créé par l’administrateur.
* Consulter les contenues des chapitres
* Un Candidat qui a acheté une formation peut consulter le contenu de cette formation.
* Répondre aux quiz
* Un Candidat qui a s’abonner à une formation spécifique, peut alors répondre au quiz de ces chapitres.
* Evaluer un formateur et une formation
* Un Candidat qui a s’abonner à une formation spécifique, peut alors évaluer cette formation et son formateur.
* Publier des commentaires dans un chapitre
* Dans chaque chapitre de la formation les candidats peuvent laisser des commentaires ou des questions.
* Télécharger des fichiers et des vidéos d’une formation
* Un Candidat à la droite de télécharger les fichiers et les vidéos d’une formation.
* Participer aux groups d’une formation en ligne
* Pour éviter le grand nombre des étudiants qui présentent dans la même séance d’une formation en ligne alors il y a chaque formation un nombre des groupes limite et chaque groupe à un maximum de nombre des candidats.
* Lorsque les groupes de la formation sont pleins alors cette formation ne sera visible que pour ces abonnés.
* Besoins fonctionnelles intéressent le Candidat et le formateur
* Chat en ligne
* Le formateur et ses étudiants peuvent discuter en temps réel.
* Joindre des séances en ligne
* Il s’agit d’un outil pour assurer des vidéos conférences dans notre application.

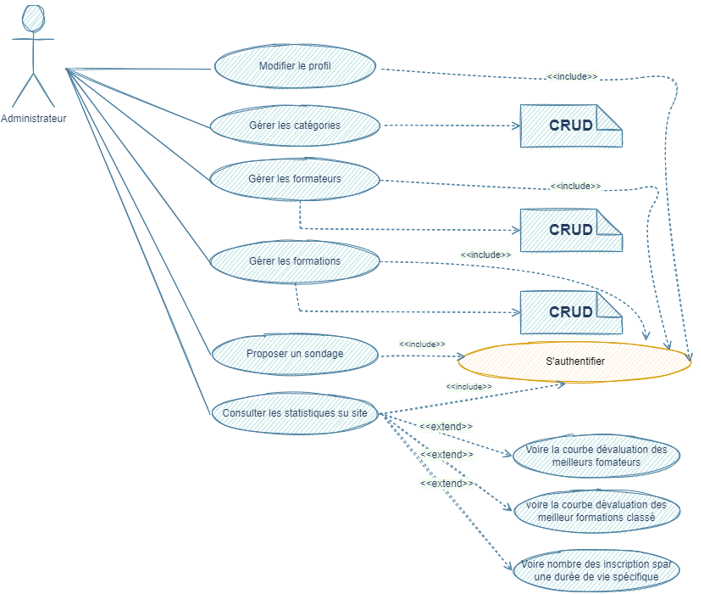
### Les besoins non fonctionnelles

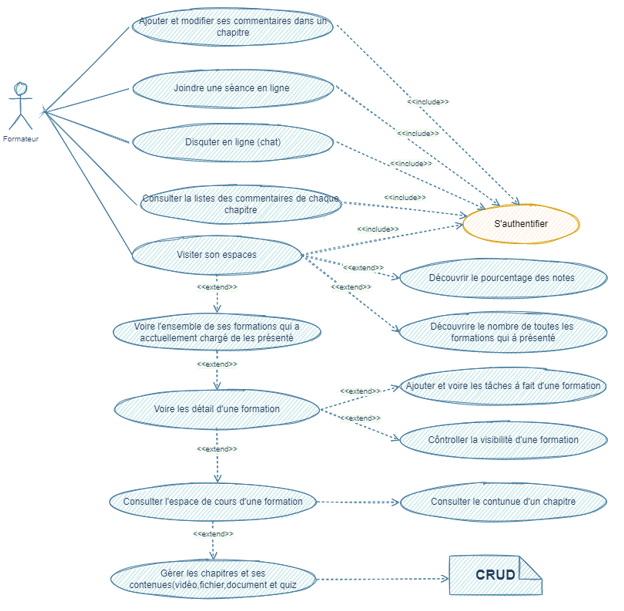
Les exigences non fonctionnelles sont importantes car elles affectent indirectement les résultats et les performances du système, ce qui signifie qu’elles ne peuvent être ignorées, pour lesquelles les exigences suivantes doivent être satisfaites :

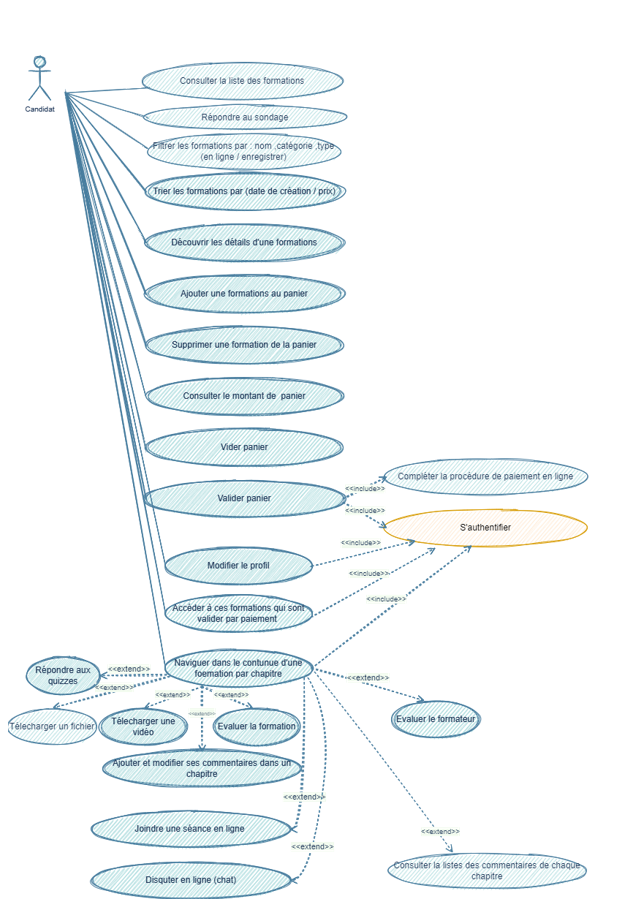
* Sécurité : Le système doit être sécurisé au niveau des données : authentification et contrôle d’accès.
* Ergonomie : c'est-à-dire que les interfaces de l’application sont claires, bien organisées et faciles à utiliser, les couleurs doivent être en harmonie.
* Performance : Le système à développer doit avoir un temps de réponse raisonnable au chargement de l’application et des différents écrans liés à sa fonctionnalité.
* La portabilité : Elle fait référence à l’efficacité avec laquelle un système fonctionne dans un environnement par rapport à un autre.
* Maintenabilité : le système à développer doit avoir un code qui est clair, lisible, est bien structuré pour qu’il soit facile à modifier.

### Diagrammes des cas d’utilisation générale

Le diagramme de cas d’utilisation nous permet de modéliser le comportement d’un système et enregistrer les exigences du système. Ces diagrammes permettent également de déterminer les interactions entre le système et ses intervenants. Les cas d’utilisation et les acteurs dans les diagrammes de cas d’utilisation décrivent ce que le système fait et comment les acteurs l’utilisent.







## Pilotage du projet avec SCRUM

Nous commençons par identifier l’équipe SCRUM, puis nous établissons le backlog de produits.

### Identification de l’équipe SCRUM

L’équipe de SCRUM intègre la coopération de plusieurs acteurs. Dans notre solution, l’équipe est identifiée par :



**SCRUM Master: Mlayah Nejemeddin**

**Product Owner: Hind Développeur**: **Bouraoui Zaid**

**Figure 6 : Identification de l'équipe SCRU**

### Le backlog du produit

Comme nous avons interprété dans le premier chapitre, le backlog du produit est une liste de tâches exprimées sous forme de besoins (users stories).

|  |  |  |  |
| --- | --- | --- | --- |
| Id story | **Priorité** | **Activité thème** | **Description** |
| 1 | High | **Authentication** | En tant qu'utilisateur de la plateforme je veux m’authentifier afin d’accéder à mon espace en toute sécurité. |
| 2 | Medium | **Modifier profile** | En tant qu'utilisateur de la plateforme je veux consulter mon profile afin de le modifier. |
| 3 | Medium | **Gérer les Categories** | En tant qu' Administrateur, je veux gérer des catégories afin d’affecter chaque formation dans une catégorie. |
| 4 | Medium | **Gérer les Formations** | En tant que Administrateur, |
| 5 | Medium | **Gérer les Formateurs** | En tant qu' Administrateur, je veux gérer des formateurs afin de donner à chaque formateur les formations qui a chargé de les présenter. |
| 7 | Medium | **Gérer les Groupes** | En tant qu' Administrateur, je veux gérer des groupes afin de  Donner à chaque formation en ligne ses groupes pour éviter le grand nombre des candidats dans une même séance en ligne. |
| 8 | Medium | **Publier des Sondages** | En tant qu' Administrateur, je veux publier un sondage afin de connaître les avis des candidats pour une formation en ligne avant de sa création. |
|  | mudium | **Répondre à un sondage** | En tant que candidat, je souhaite répondre à un questionnaire de formation qui sera créé pour inciter l'administrateur à le créer. |
|  |  |  |  |
| 8 | high | **Consulter les Statistiques** | En tant qu' Administrateur, je veux consulter les statistiques de la plateforme afin d’explorer d’une manière simple des informations telles que les nombre d’inscription des candidats dans chaque (mois, semaine, jour), les populaires formations et formateurs. |
|  |  | **Visiter espace du formateur et Consulter des Statistiques** | En tant que Formateur, je veux consulter mes statistique afin de prendre les informations d'une manière simple telles que le pourcentage de mes évaluations par les candidats, le nombre de toutes mes formation, mon grade (expérience) selon le nombre des formations qui j’ai chargé de les présenter, pourcentage d’évaluations de mes courantes formations. |
| 10 | high | **Gérer les taches a fait de chaque formation** | En tant que Formateur, je veux Rechercher les projets. Afin de connaître les tâches a fait de chaque formation d’une manière simple. |
| 11 | high | **Gérer les chapitres d’une formation et son contenues (Vidéos, fichiers, document et quizz)** | En tant que Formateur, je veux gérer les chapitres et ses contenues afin de remplir le contenue d’une formation qui va être affichée pour les abonnés. |
| 12 | low | **Contrôler la visibilité d’une Formation** | En tant que Formateur, je veux contrôler la visibilité publique d’une formation afin de s’assurer que seulement les formations qui j’ai fini ses tâches seront affichés au consultant de la plateforme et pour mettre une formation invisible lorsqu’il y a des fautes. |
| 13 | High | **Discuter en ligne** | En tant que Formateur et Candidat, nous voulons discuter en temps réel dans une formation en ligne afin d’assurer une bonne communication entre nous. |
| 14 | High | **Joindre une séance en ligne** | En tant que Formateur et Candidat, nous voulons rejoindre une séance en ligne afin de faciliter la compréhension de la leçon. |
|  | Medium | **Publier des commentaires dans un chapitre** | En tant que Formateur et Candidat, nous voulons publier des commentaires dans un chapitre afin de poser des questions ou d’ajouter quelques remarques dans un chapitre. |
| 15 | Medium | **Consulter la liste des formations disponibles** | En tant que Candidat, je veux consulter la liste des formations afin de voir les détails d’une formation ou de l’acheté. |
| 16 | Medium | **Filterer les formations** | En tant que Candidat, je veux des outils de filtrages dans la liste des formations afin de faciliter l’obtention rapide de ce j’ai besoin. |
| 17 | high | **Gérer le panier** | En tant que Candidat, je veux gérer mon panier sans créer un compte et sans s’authentifier afin de sauvegarder les formations qui m’intéressent dans mon panier d’une manière simple et légère. |
| 19 | high | **Valider un Panier avec un système de paiement en ligne** | En tant que Candidat, je veux une procédure de paiement en ligne pour la validation de mon panier à fin d'abonnés dans des formations en toute simplicité. |
| 20 | high | **Naviguer entre les chapitres** | En tant que Candidat abonné dans une formation je souhaite naviguer entre les chapitres afin de découvrir le contenu de chaque chapitre. |
|  | medium | **Consulter mes formations** | En tant que Candidat, je souhaite consulter mes formations afin de les avoir rapidement. |
| 20 | medium | **Evaluer le formateur et la formation** | En tant que candidat, je souhaite évaluer le formateur et sa formation afin d'exprimer mon opinion et d'en faire profiter d'autres qui ne se sont pas encore s'abonner à cette formation de mon point de vue. |
|  | high | **Répondre à un quiz** | En tant que candidat, je souhaite répondre à unquiz afin de tester mes connaissances. |
| 21 | low | **importer un**  **fichier et un vidéo** | En tant que candidat, je souhaite un outil qui me permettra de télécharger un fichier et une vidéo afin de les visionner lorsqu'il n'y a pas de connexion internet. |

**Tableau 2 : Product Backlog**

### Structure des sprints

La phase de planification des sprints est une importance primordiale pour la réussite d’un projet. Cette étape permet de préparer un planning de travail et d’avoir une démarche optimisée pour le découpage et l’affectation des tâches selon les priorités que nous avons affectées dans notre backlog de produit.

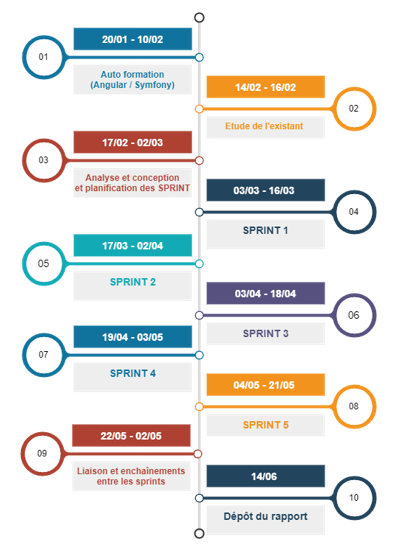
### Planification des sprints du projet

La planification de sprints de notre projet est présentée dans la figure ci-dessous



## Planification de réalisation du projet

Pour gérer ce projet, tout au long de la période de stage, nous avons montre notre planning la figure ci-dessous.



## Conclusion

Dans ce chapitre, nous avons commencé par l’identification des besoins ainsi que l’identification des acteurs, puis nous avons présenté le diagramme des cas d’utilisation globale. Ensuite, on a fixé l’équipe SCRUM et le backlog de produit, enfin nous avons décomposé la planification de nos sprints. Dans le chapitre suivant, nous allons présenter la préparation de l’environnement de travail.

# SPRINT 0 – PREPARATION De L’environnement De Travail

## Introduction

Dans ce chapitre nous allons expliciter la phase ultime d’un cycle de développement avec SCUM. En premier lieu nous débutons par l’environnement matériel et logiciel, et en deuxième lieu nous présenterons l’architecture logicielle et opérationnelle ainsi que le diagramme de déploiement.

## Environnement de travail

Pour assurer le bon développement de l’application, il faut utiliser des outils adéquats à la réalisation du projet. Ces outils représentent l’environnement de travail, nous présentons en premier lieu l’environnement matériel. En second lieu, nous présentons l’environnement logiciel.

### Environnement matériel

Nous motionnons les caractéristiques de l’ordinateur sur lesquelles nous avons développé l’application parce qu’elles peuvent donner une idée sur les conditions du travail.

L’application a été développée sur un ordinateur portable ASUS qui se caractérise par :

* **Processus** : Intel(R) Core (TM) i7-7500U CPU @ 2.70GHz 2.90 GHz.
* **Mémoire Ram installée** : 8,00 Go.
* **Type de système** : Système d’exploitation Windows 10 64 bits.
* **Disque dur** : 1To.
* **Ecran** : 15.6 pouces.

### Environnement logiciel

Le développement des applications informatiques ne nécessite pas seulement des outils matériels, mais aussi des outils logiciels.

* Outils de développement

— XAMPP

XAMPP est un ensemble de logiciels servant à mettre en place aisément un serveur Web, un serveur FTP et un serveur de messagerie électronique. C’est une distribution de logiciels libres (X Apache MySQL Perl PHP) offrant une bonne souplesse d’utilisation, reconnue pour son installation simple et rapide. [10]



**Figure 7 : Logo XAMPP**

— Visual Studio Code

C’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 possède un riche écosystème d’extensions pour d’autres langages (tels que C ++, C#, Java, Python, PHP, Go) et des runtimes (tels que .NET et Unity). [11]



**Figure 8 : Logo Visual Studio Code**

— phpMyAdmin

phpMyAdmin est un logiciel libre écrit en PHP qui a pour mission de s’occuper de l’administration d’un serveur de base de données MySQL ou MariaDB.

Vous pouvez utiliser phpMyAdmin pour réaliser la plupart des tâches d’administration, ceci incluant la création de base de données, l’exécution de demandes, et l’ajout de comptes utilisateur. [12]

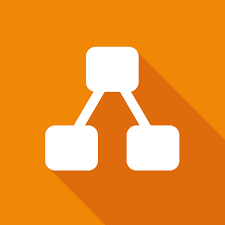


**Figure 9 : phpMyAdmin**

* Outil de conception

— draw.io

diagrams.net (draw.io) est une solution accessible en ligne et gratuite qui permet de créer des diagrammes et des organigrammes. Vous avez accès à toutes les fonctionnalités d’un outil de création de diagrammes professionnel, dans un espace de travail en anglais. [15]



**Figure 10 : Logo Draw.io**

* Choix technologiques

— Angular 12

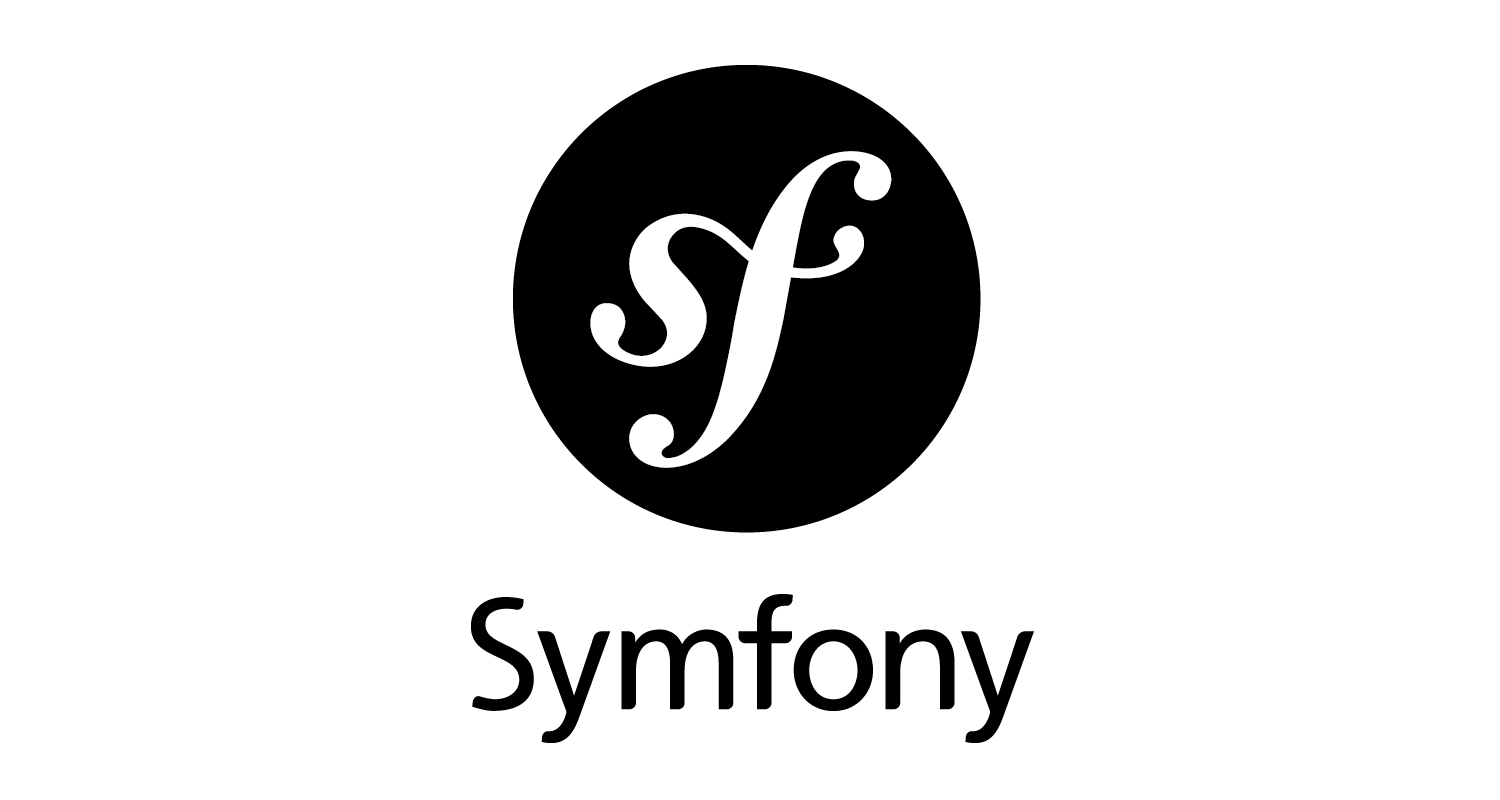
Angular est un framework JavaScript développé par Google et reconnu pour être l’un des frameworks les plus complets. Il embarque en effet nativement nombre de fonctionnalités le qualifiant pour le développement d’applications web, même les plus complexes. Grâce à son évolution, Angular est cependant également adapté à tout projet d’application de bureau ou mobile. Rapide à déployer et puissant, il sait convaincre petites et grandes entreprises.[citation]



**Figure 11 : Logo Angular**

* Symfony 6

Symfony est un ensemble de composants PHP ainsi qu'un [framework](https://fr.wikipedia.org/wiki/Framework) [MVC](https://fr.wikipedia.org/wiki/Mod%C3%A8le-vue-contr%C3%B4leur) [libre](https://fr.wikipedia.org/wiki/Logiciel_libre) écrit en [PHP](https://fr.wikipedia.org/wiki/PHP). Il fournit des fonctionnalités modulables et adaptables qui permettent de faciliter et d’accélérer le développement d'un [site web](https://fr.wikipedia.org/wiki/Site_web).[citation]



**Figure 12 : Logo Symfony**

## Choix de l’architecture

Afin de mieux comprendre le fonctionnement de l’application, la notion d’architecture est nécessaire. Dans cette section, nous détaillons l’architecture logiciel et l’architecture opérationnelle de notre application ainsi que le diagramme de déploiement.

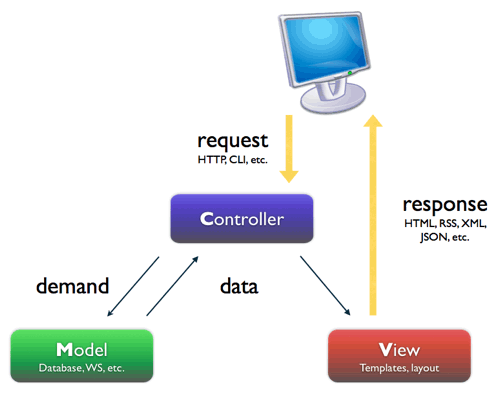
### Architecture logiciel

* **Back-end : L’architecture MVC**

Pour le développement web, les solutions les plus populaires pour organiser notre code de nos jours sont la mise en place d’une architecture MVC. Elle définit un cadre d’organisation de votre code en accord avec sa nature. Cette architecture a trois couches:

* **Le modelé :** interagit avec la base de données, les regroupes, traite et gère les données. Le model contient les données et leur logique.
* **La vue** : s’occupe principalement de faire afficher ce que le modelé renvoie. Ensuite, elle s’occupe de recevoir toute interaction de l’utilisateur. La vue contient la présentation graphique à renvoyer à l’utilisateur.
* **Le contrôleur** : prend en charge de synchroniser le modèle et la vue. Il capte toutes les activités de l’utilisateur et en fonction de de ses activités, il actionne les changements à effectuer sur l’application. Le contrôleur traite les actions utilisateur (via des requêtes), demande au modelé d’effectuer les changements, puis passe les données à la vue.

La figure ci-dessous montre le concept de l’architecture MVC



**Figure 13 : L'Architecture MVC**

La figure de l’architecture MVC, montre le flux d’information entre le différent élément du modèle :

1- L’utilisateur envoie une requête HTTP

2- Le contrôleur appelle le modèle, celui-ci va récupérer les données

3- Le modèle retourne les données au contrôleur

4- Le contrôleur décide de la vue à afficher, va l’appeler

5- Le code HTML de la vue est envoyé à l’utilisateur pour qu’il puisse

Naviguer normalement

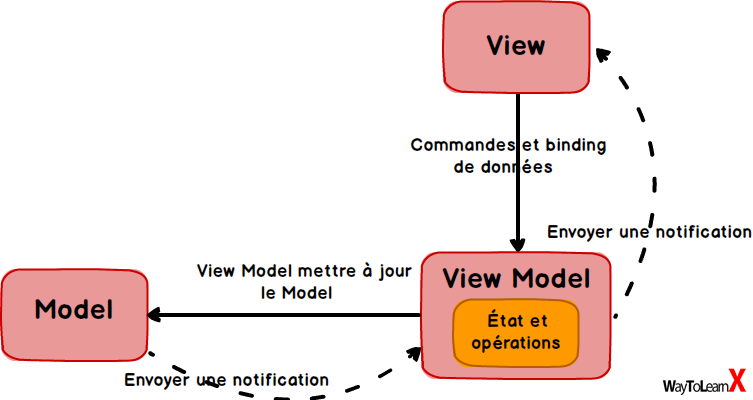
L’approche MVC apporte de réels avantages :

* Une conception claire et efficace grâce à la séparation des taches de la vue et du contrôleur
* Un gain de temps de maintenance et d’évolution du site
* Une plus grande souplesse pour organiser le développement du site entre différents développeurs (indépendance des données, de l’affichage (webdesign) et des actions)

Nous allons travailler notre future solution avec cette architecture logique de modèle MVC

* Front-end : L’architecture MVVM

Le modèle MVVM (Model-View-ViewModel) est un modèle de design architectural d’interface utilisateur qui découple l’interface utilisateur du code associé. Il est supporté la liaison de données bidirectionnelle entre View et ViewModel. Ceci permet de propager automatiquement les changements dans l’état ViewModel à la vue.

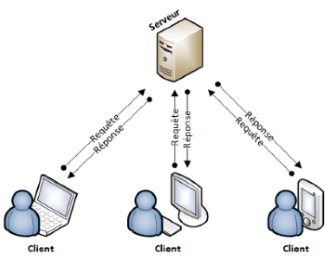
****

**Figure 14 : L'architecture MVVM**

* Le modèle représente un ensemble de classes qui expliquent la logique opérationnelle, c’est-àdire. le modèle d’objet opérationnel et le modèle de données (opérations d’accès aux données).Il définit aussi des règles d’affaires pour les moyens de données comme la façon dont les données peuvent être modifiées et manipulées.
* La vue représente les composants de l’interface utilisateur comme CSS, jQuery, HTML, etc. La vue montre les données reçues par le contrôleur en conséquence. Cela change aussi le(s) modèle(s) dans l’interface utilisateur.
* Le ViewModel est responsable de l’affichage des méthodes, commandes et autres fonctionnalités. qui permettent de maintenir l’état de la vue, de manipuler le modèle en fonction des actions sur la vue et de déclencher les événements dans la vue elle-même.

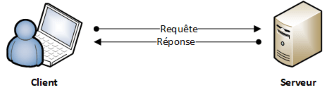
### Architecture de serveur

Une architecture client-serveur représente l’environnement dans lequel des applications de machines clientes communiquent avec des applications de machines de type serveurs. L’exemple classique est le navigateur Web d’un client qui demande (on parle de “requête”) le contenu d’une page Web à un serveur Web qui lui renvoie le résultat (on parle de “réponse”).



**Figure 15 : L'architecture client/serveur**

* Si toutes les ressources nécessaires sont présentes sur un seul serveur, on parle d’architecture à deux niveaux ou 2 tiers (1 client + 1 serveur).



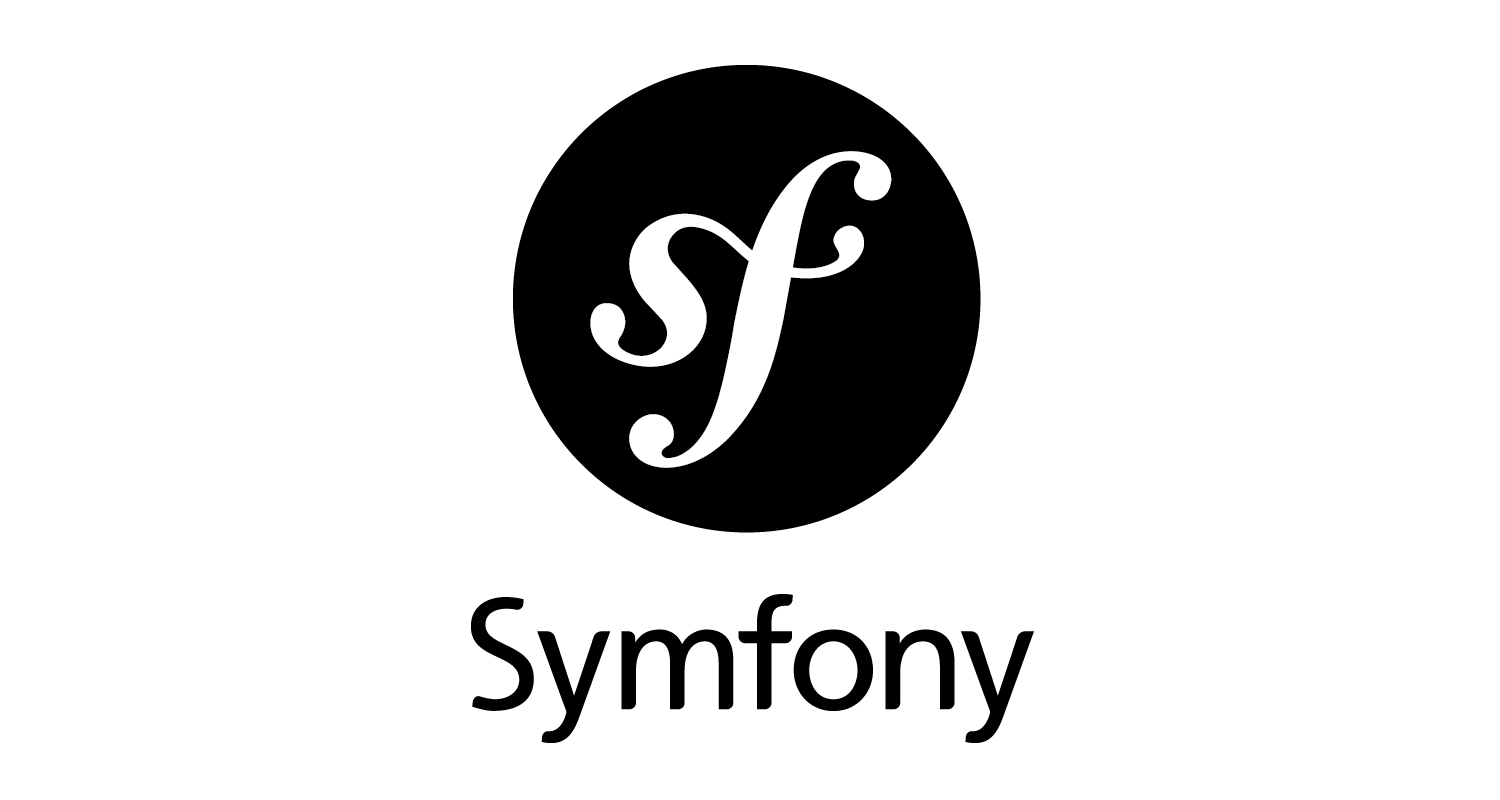
**Figure 16 : Architecture 2-tiers**

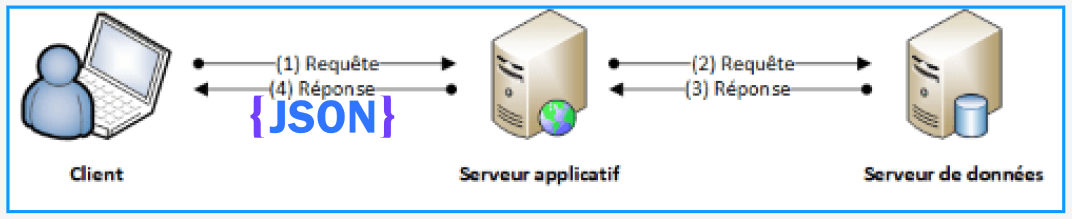
* Si certaines ressources sont présentes sur un deuxième serveur (par exemple des bases de données), on parle d’architecture à trois niveaux ou 3 tiers (1 client interroge le premier serveur qui lui-même interroge le deuxième serveur).



**Figure 17 : Architecture 3-tiers**

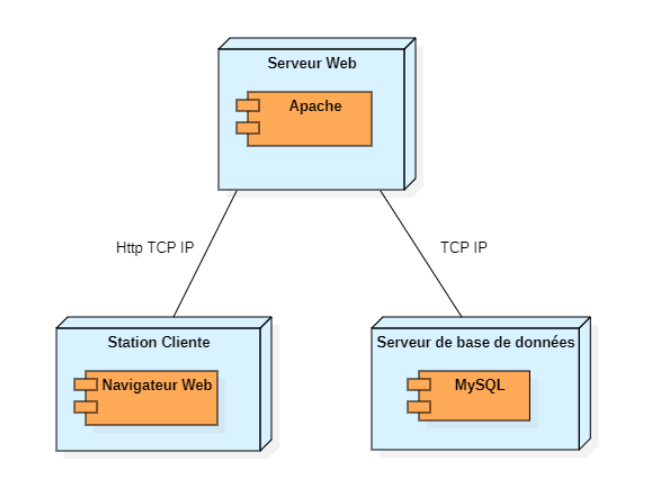
Nous résumons alors les choix architecturaux que nous avons retenus dans la figure suivante :



### Diagramme de déploiement

Un diagramme de déploiement fait partie de la catégorie des diagrammes structurels, car il décrit un aspect du système même. Dans le cas présent, le diagramme de déploiement décrit le déploiement physique des informations générées par le logiciel sur des composants matériels.



**Figure 18 : Diagramme de déploiement**

## Conclusion

Dans ce chapitre, nous avons présenté l’environnement de développement et le choix des techniques ainsi que l’architecture adoptée. Dans le chapitre suivant, nous allons détailler le premier sprint.

# Chapitre 4 : RELEASE 1

## Introduction

Durant ce chapitre, nous allons traiter les deux premiers sprints de notre projet. Dans chaque sprint nous allons présenter en premier lieu le backlog sprint, en deuxième lieu nous allons définir la capture des besoins. En troisième lieu nous allons déterminer l’analyse, nous passons par la suite à déterminer la conception. Enfin, nous clôturerons ce chapitre par l’implémentation et les différentes phases de tests.

## Le développement du sprint 1

### Le backlog du Sprint

Il s’agit de l’ensemble des fonctionnalités caractérisées par l’équipe scrum à partir du backlog de produit et qui seront accomplis au cours de notre premier sprint. Le tableau ci-dessous résume le backlog du premier sprint.

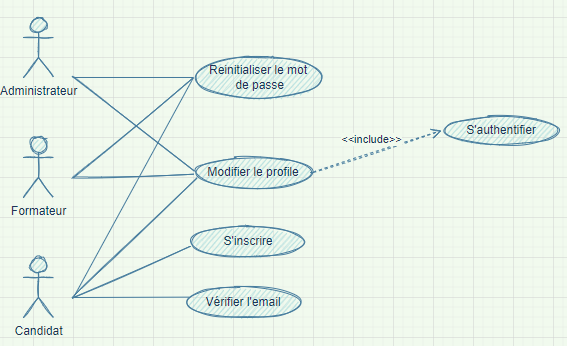
|  |  |  |
| --- | --- | --- |
| Id story | **Activité thème** | **Tâches** |
| 1 | **Authentication** | -Développer une fonction pour l’inscription d’un utilisateur.  -A l’aide de lexik\_jwt bundle assurer un système d’authentification (se connecter avec JSON WEB TOKEN).  -A l’aide de la bundle [SymfonyCastsResetPasswordBundle](https://github.com/symfonycasts/reset-password-bundle) ajouter l’option ‘mot de passe oublier ‘.  -A l’aide de la bundle verify-email-bundle assurer un système de vérification des emails des utilisateurs après l’inscription.  -Finalement en utilisant Postman faire un test pour valider les REST’s API. |
| -Générer deux Composant :  - Composant  pour l’inscription.  - Composant  pour la connexion.  -Gérer les routes nécessaires.  -Appeler les API’s dans un service ‘auth-service’.  -Créer un intercepteur pour intercepter la requête qui contient un mot (‘api’) sauf (‘api/login’ ou ‘api/register’ ou ‘api/public’) du client et le modifier en ajoutant un jeton a l’entête (‘Authorization’). Ce jeton est générer lors de la connexion d’utilisateur et s’enregistrer dans le stockage local de navigateur.  -Dans la composant de connexion, lorsque le jeton s’enregistre dans le stockage local de navigateur, alors décoder ce jeton pour récupérer le rôle de l’authentifiant et selon son rôle on rediriger a un route spécifique (Interface d’administrateur ou Interface d’un Candidat ou Interface de formateur).  -Dans le ‘auth-service’ développer une fonction qui fait un test si le mail de l’authentifiant a été vérifiée ou non. |
| 2 | **Modifier profile** | -Développer une fonction qui modifier les données d’un utilisateur (le profile récupérer ses donnes de la même table utilisateur).  -faire un test. |
| -Dans le service ‘auth-service’ ajouter une nouvelle fonction qui appelle le REST API de la modification de profile puis générer un nouveau composant ‘profile’, ajouter une méthode qui appelle la nouvelle fonction du service ‘auth-service’ et créer nouveau formulaire puis ajouter le nom de la FormGroud dans l’entête du formulaire et donne à chaque champs un formControlName qui est déjà déclarer sous la formGroup dans le composant profile.  -faire un test. |

**Tableau 3 Backlog de sprint 1**

Pour avoir une vision globale du comportement fonctionnel des user stories de ce sprint, nous avons choisi le diagramme de cas d’utilisation donné la figure suivante.

### Diagramme de cas d’utilisation

La figure ci-dessous présente le diagramme de cas d’utilisation de premier sprint.



**Figure 19 : Diagramme de cas d'utilisation de sprint 1**

### Capture des besoins

Lors de cette partie, nous allons présenter les raffinements des cas d’utilisation les plus délicats ainsi que leurs descriptions textuelles.

#### Description textuelle du cas d’utilisation : s’authentifier

|  |  |
| --- | --- |
| Cas d’utilisation | S’authentifier |
| Acteur | Utilisateur de l’application (administrateur, formateur, candidat) |
| Description brève | Les utilisateurs de l’application doit s’identifier Avant de profiter de  certaines fonctionnalités de l'application. |
| Pré-condition | Les utilisateurs doivent être crées dans la base de données |
| Scénario Nominal | 1. L’utilisateur demande la page d’authentification ou une page sécurisé par un système d’authentification. 2. Le système vérifie si l’utilisateur est déjà connecté, si oui vérifié son rôle 3. Le système affiche la page d’authentification. 4. L’utilisateur saisit ses droits d’accès. 5. Le système vérifie l’existence des données. 6. Le système reconnait le rôle d’utilisateur. 7. Le système vérifie si l’email d’utilisateur a été vérifié ou non 8. Le système affiche la page correspondante. |
| Scénario alternatif | 2.1 : Utilisateur été déjà authentifier et il demande une page sécurisé mais avec son rôle il n’a pas l’autorisation d’accéder à la page demandé.  2.1.1 : Le système redirige l’utilisateur vers une page qui affiche un message d’erreur ‘401’.  5.1 : Login ou mot de passe non valide.  5.1.1 : Le système affiche un message d’erreur.  5.1.2 : Reprise de la troisième étape de l’enchainement nominal.  6.1 : Rôle d’utilisateur n’a pas l’accès à accéder à la page demandé (en cas ou l’utilisateur demande une page sécurisé non la page d’authentification dans l’étape 1).  6.1.1 : Le système redirige l’utilisateur vers une page qui affiche un message d’erreur ‘401’.  7.1 : Email non vérifié  7.1.1 : Le système affiche un message d’erreur.  7.1.2 : Reprise de la troisième étape de l’enchainement nominal. |
| Post-condition | -Utilisateur authentifié.  -Affichage de l'interface graphique correspondant au rôle d’utilisateur ou affichage de la page sécurisé au cas où l’utilisateur demande une page sécurisé dans l’étape 1. |

**Tableau 4 : Description textuelle du cas d'utilisation : s'authentifier**

#### Description textuelle du cas d’utilisation : s’inscrire

|  |  |
| --- | --- |
| Cas d’utilisation | S’inscrire |
| Acteur | Candidat |
| Description brève | Le Candidat doit inscrire dans l’application Avant de profiter de  certaines fonctionnalités de l'application. |
| Pré-condition | Le Candidat demande la page d’inscription. |
| Scénario Nominal | 1. Le système affiche la page d’inscription. 2. Le Candidat remplie le formulaire d’inscription. 3. Le système vérifie l’existence du mail saisit. 4. Le système enregistre l’utilisateur. 5. Le système envoie un mail de vérification au Candidat pour vérifier son email. 6. Le système redirige le Candidat au page de connexion. |
| Scénario alternatif | 3.1 : email déjà inscrit.  3.1.1 : Le système affiche un message d’erreur.  3.1.2 : Reprise de la première étape de l’enchainement nominal. |
| Post-condition | Candidat inscrit dans la plateforme. |

**Tableau 5 : Description textuelle du cas d'utilisation : s'inscrire**

#### Description textuelle du cas d’utilisation : modifier profile

|  |  |
| --- | --- |
| Cas d’utilisation | Modifier profile |
| Acteur | Utilisateur de l’application (administrateur, formateur, candidat) |
| Description brève | L’utilisateur peut modifier son profil. |
| Pré-condition | Utilisateur authentifié |
| Scénario Nominal | 1. L’utilisateur demande la page de modification du profil. 2. Le système affiche la page avec un formulaire qui contient les informations courantes d’utilisateur. 3. L’utilisateur saisit ses nouvelles informations. 4. Le système vérifie la saisie d’utilisateur. 5. Le système modifie les informations d’utilisateur. 6. Le système affiche un message de succès |
| Scénario alternatif | 4.1 : le formulaire non valide.  4.1.1 : Le système affiche un message d’erreur.  4.1.2 : Reprise de la deuxième étape de l'enchaînement nominal.  4.2 : En cas ou l’utilisateur peut changer son mot de passe la saisit de son mot de passe actuel et fausse.  4.2.1 : Le système affiche un message d’erreur.  4.2.1 : Reprise de la deuxième étape de l'enchaînement nominal. |
| Post-condition | -Les informations d’utilisateur sont modifiées avec succès. |

**Tableau 6 : Description textuelle du cas d'utilisation : modifier profile**

### Analyse

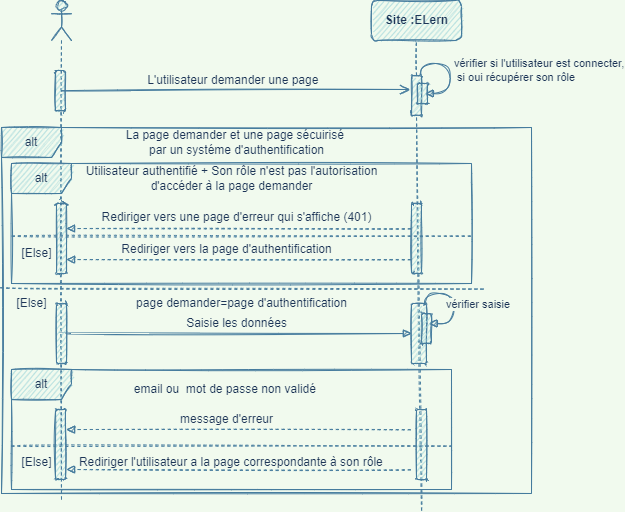
Après la partie de capture des besoins, nous allons faire l’analyse sur les différents scénarios. Cette étape est consacrée pour l’élaboration des diagrammes de séquences systèmes et des diagrammes de classes participants

#### Diagrammes de séquences système

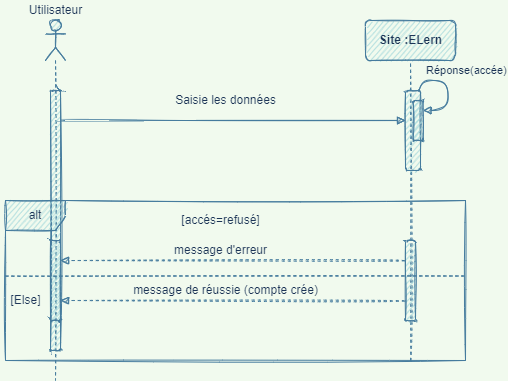
Les cas d’utilisation décrivent les interactions des acteurs avec le système que nous voulons spécifier et concevoir. Lors de ces interactions, les acteurs produisent des messages qui affectent le système informatique et appellent généralement une réponse de celui-ci. Nous allons isoler ces messages et les représenter graphiquement sur des diagrammes de séquence UML.

Nous allons représenter le d’un scénario représentatif des cas d’utilisation décrits précédemment.

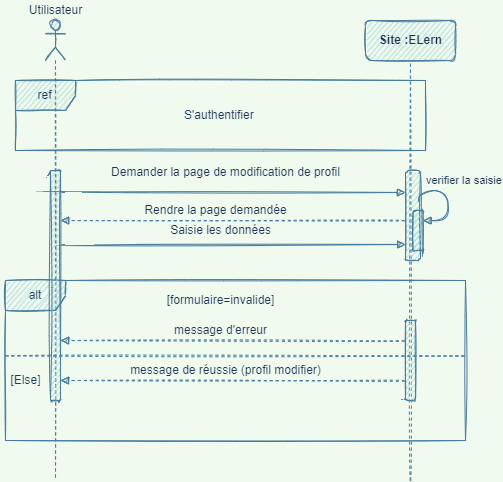
1. **Diagramme de séquences système du cas d’utilisation : s’authentifier**

****

1. **Diagramme de séquences système du cas d’utilisation : s’inscrire**



1. **Diagramme de séquences système du cas d’utilisation : modifier profile**



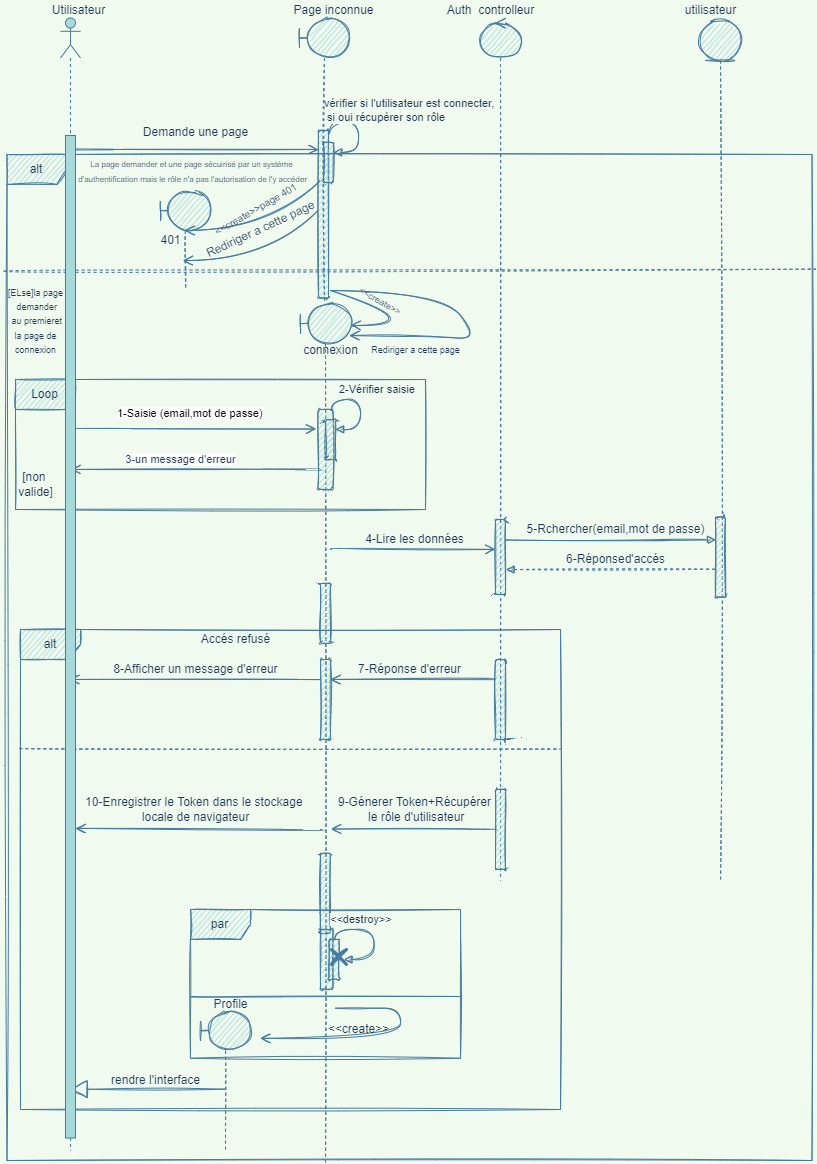
### Conception

Dans cette section nous présentons le diagramme de séquence détaillé des cas d’utilisation les plus délicats ainsi que le diagramme de classe de conception.

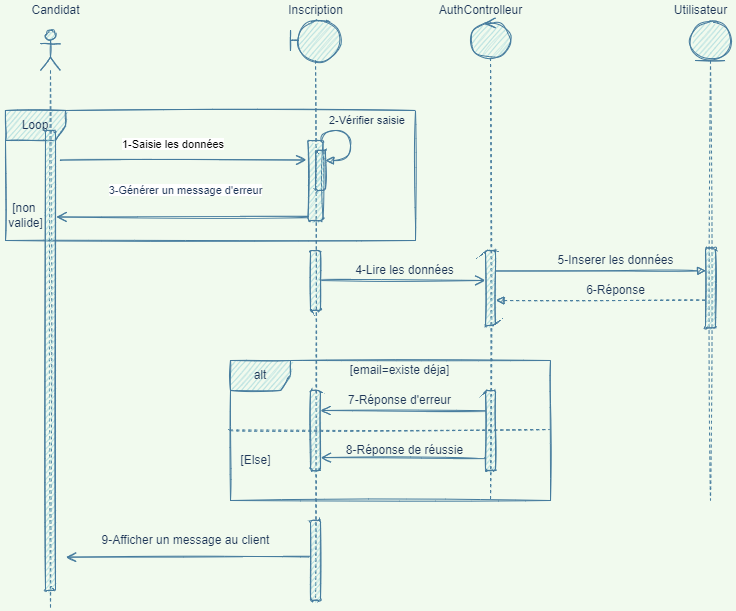
#### Diagramme de séquences détaillée

Le diagramme séquentiel sert à illustrer les interactions entre les objets dans un scénario d’utilisation. Afin de simplifier, nous représentons l’acteur principal à gauche du diagramme, ainsi que les éventuels acteurs secondaires à droite du système. L’objectif est de décrire le déroulement des interactions entre les acteurs ou les objets.

1. Diagramme de séquence détaillée du cas d’utilisation : s’authentifier



1. Diagramme de séquence détaillée du cas d’utilisation : s’inscrire



1. Diagramme de séquence détaillée du cas d’utilisation : modifier profile

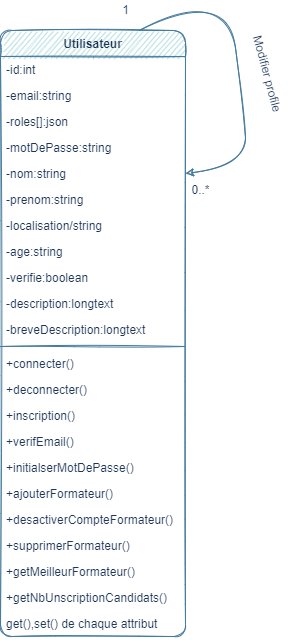
#### Diagramme de classes de ce sprint

Le diagramme de classe de conception permet de mieux comprendre le fonctionnement de l’application en démontrant la liaison entre chaque classe participante dans le sprint.

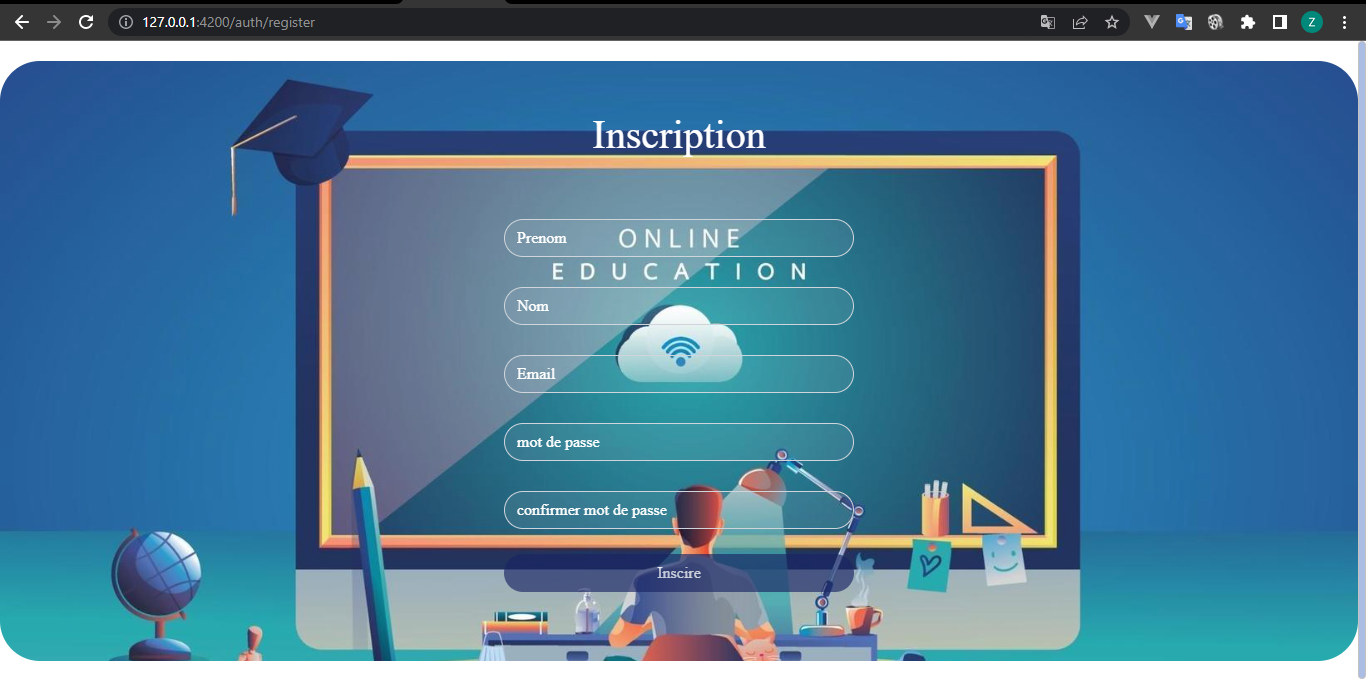
##### Dictionnaire de données

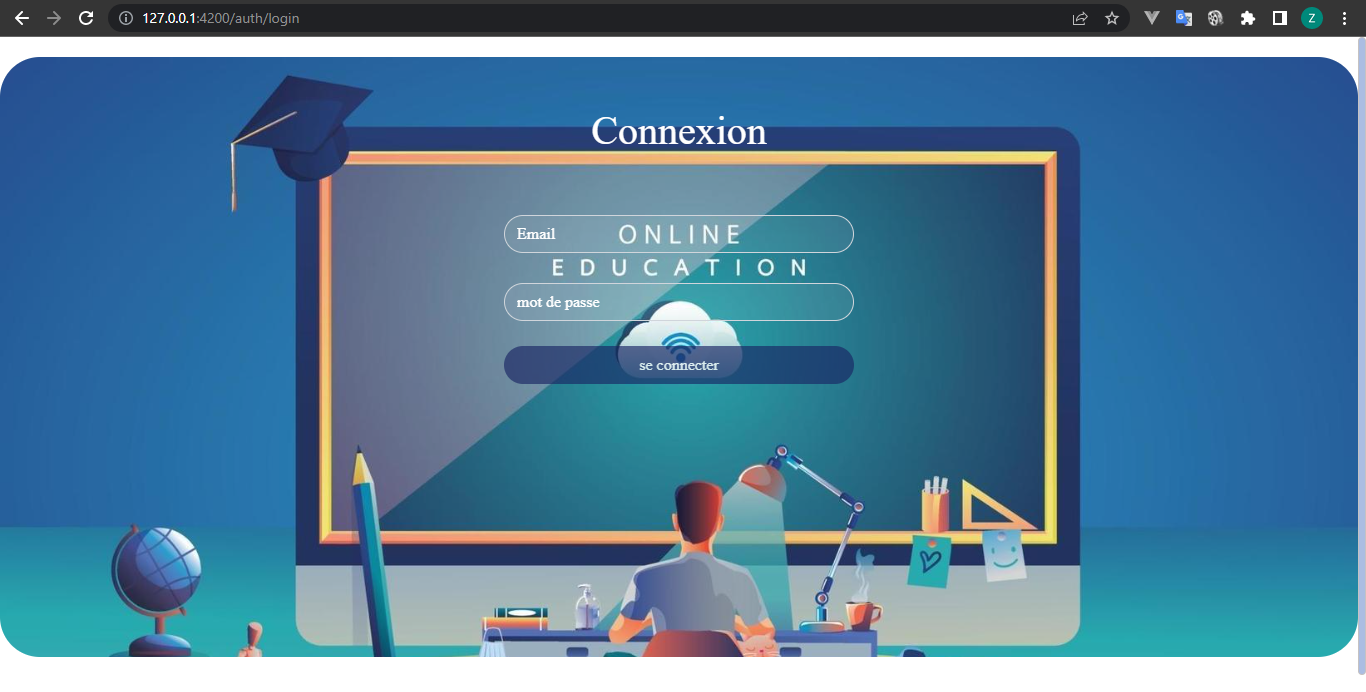
|  |  |  |
| --- | --- | --- |
| ***Nom*** | ***Constraints*** | ***Type*** |
| id | PRIMARY KEY | INT |
| email | NOT NULL | VARCHAR(100) |
| roles | NOT NULL | json |
| motDePasse | NOT NULL | VARCHAR(100) |
| nom | NOT NULL | VARCHAR(100) |
| prenom | NOT NULL | VARCHAR(100) |
| localisation | NULL | VARCHAR(100) |
| age | NULL | int |
| verifie | NULL | boolean |
| description | NULL | longtext |
| breveDescription | NULL | longtext |

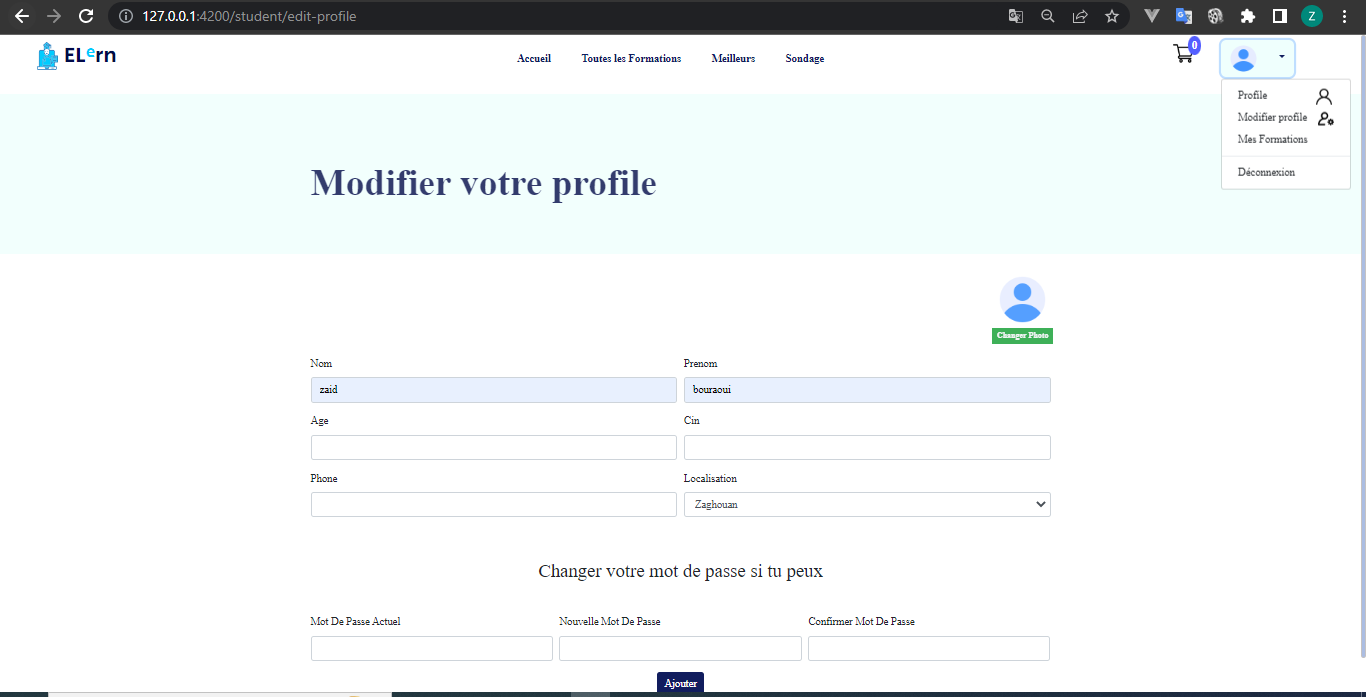
##### Le modèle



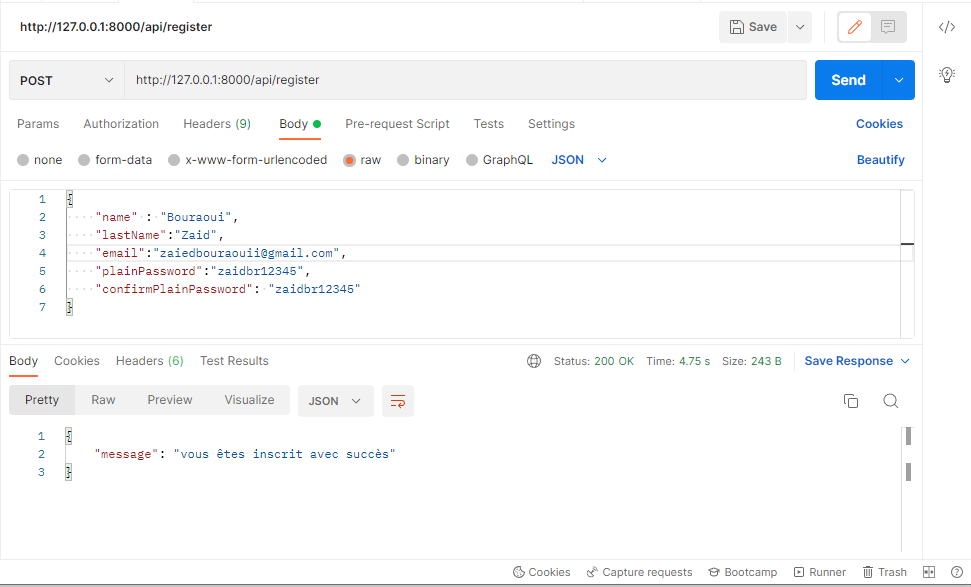
### Revue de sprint

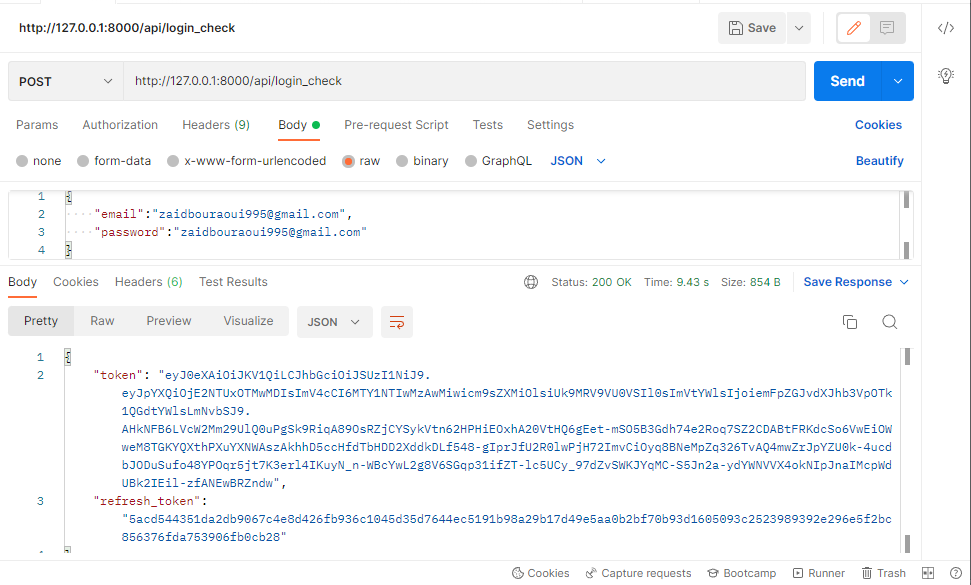


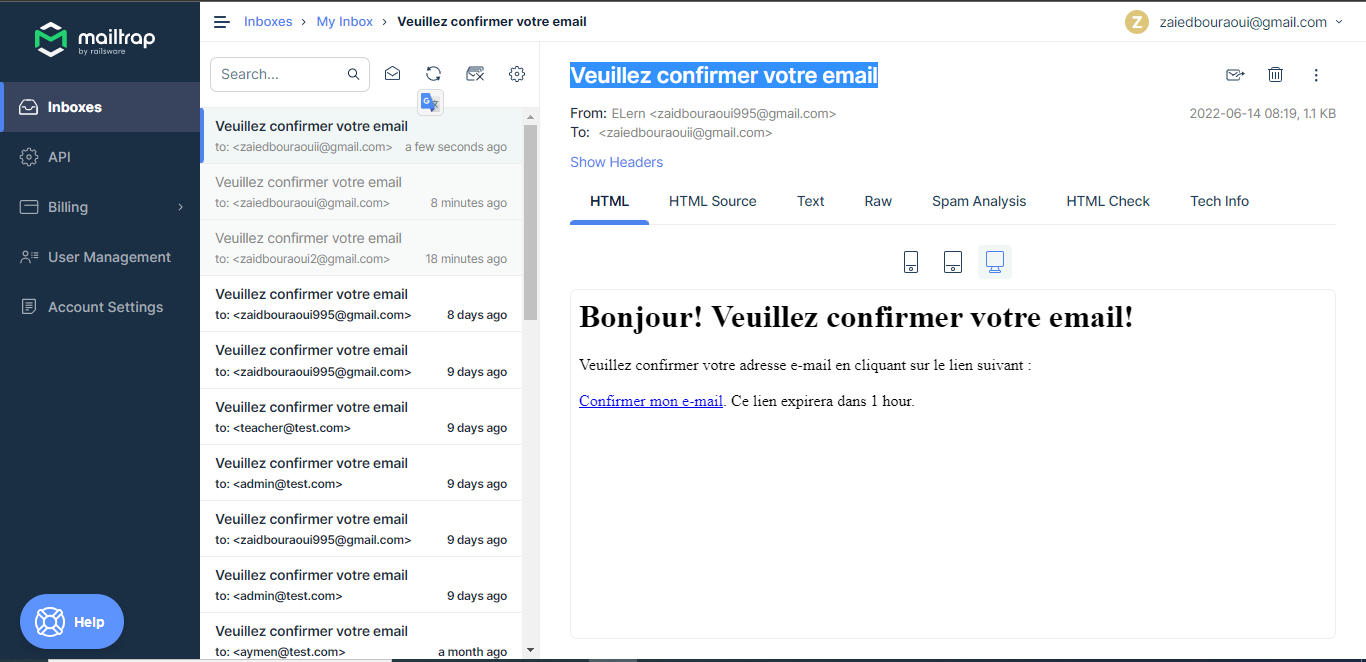


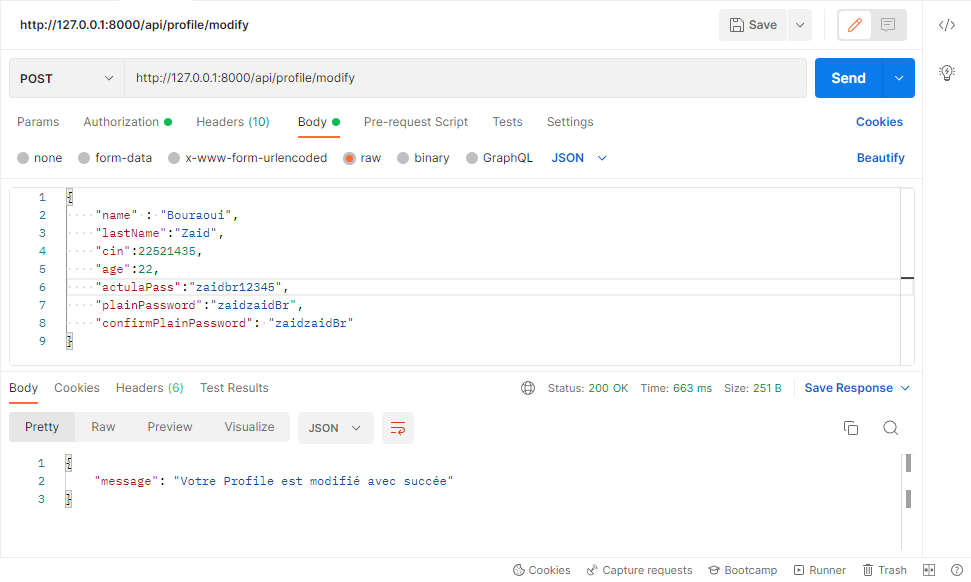


### Tests









***Conclusion***

Dans ce sprint, nous analysons et spécifions d'abord toutes les exigences liées à ce sprint, puis nous continuons à concevoir des graphiques afin que nous puissions mettre en œuvre des services Web et leur consommation.

## Le Développement du sprint 2

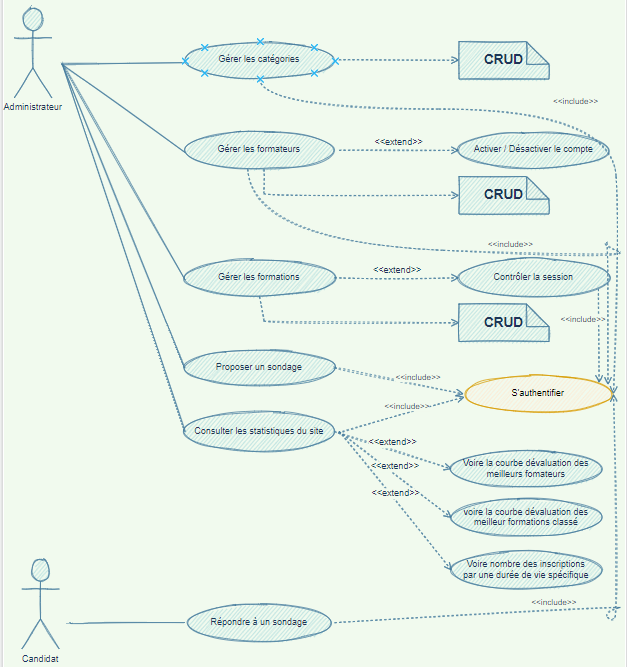
En se basant sur le même principe que le sprint précédent, cette partie sera consacrée à la réalisation du deuxième sprint. Ce sprint est consacré à la réalisation de la partie qui comprend l’administration effectuée par l’administrateur et la réponse du candidat aux sondages.

### Le backlog du sprint 2 :

Comme pour le premier sprint, nous allons présenter l’ensemble de fonctionnalités (Users Stories) extraites à partir du backlog du produit et qui vont être réalisées durant ce sprint.

|  |  |  |  |
| --- | --- | --- | --- |
| Id story | **Priorité** | **Activité thème** | **Description** |
| 3 | Medium | **Gérer les Categories** | Dans un nouveau contrôleur ajouter les 4 fonctions suivantes :  create () : pour la création d’un nouveau objet et le persister dans la base de donnes.  update (id) : pour la modification d’un objet déjà existe.  delete (id) : supprimé l’objet.  getData () : Utiliser SerializableInterface pour coder l’objet au format JSON puis utiliser l’annotation Groups pour revenir seulement les propriétés annotés dans la classe au bute d’envoyer seulement les données nécessaires dans la réponse d’API.  -Finalement faire un test. |
| -Ajouter un nouveau service ‘service-manager’ et faire toutes les appels des API’s dans ce service a l’aide des méthodes de HttpClient (get, post, put, delete) puis l’injecter dans les composant nécessaires et faire l’appelle de ces méthodes.  -Générer un nouveau module ‘admin-dashboard’, puis créer 3 nouveaux composants :  -add-component : utiliser une méthode du service ‘service-manager’ qui fait appelle à la REST API qui est responsable d’enregistrer les données d’objet actuel dans la base de données. Ce composant contient d’un formulaire avec les champs nécessaires qui sont respectivement compatible aux clés de la requête qui l’API le reçue.  - edit-component : même travaille de add-component mais dans ce composant on ’a une utilisation des Observables pour s’abonner au route actuel pour permettre de modifier l’objet.  -display-component : utiliser une méthode du service ‘service-manager’ qui fait appelle à REST API qui est responsable de récupérer les données de la base de données. Dans ce composant on a encore l’appelle de la méthode qui fait l’appelle de l’API responsable de supprimer l’objet identifier.  En utilisant le ReactiveFormModule assurer l’envoie et la validation des formulaires.  -Préparer un design agréable.  -Implémenter les routes nécessaires  -protéger le module admin-dashboard par un guard ‘admin-guard’ et configurer le route de ce module comme un route asynchrone (Lazy loaded).  -Finalement faire un test. |
| 4 | Medium | **Gérer les Formations** | -Même travaille de la tache ‘**Gérer les Catégories’.** |
| -Même travaille de la tache ‘**Gérer les Catégories’** en utilisant le même service **‘service-manager’** et des nouvelles composant sous le module ‘admin-dashboard’. |
| 5 | Medium | **Gérer les Formateurs** | -Même travaille de la tache ‘**Gérer les Catégories’.** |
| -Même travaille de la tache ‘**Gérer les Catégories’** en utilisant le même service **‘service-manager’** et des nouvelles composant sous le module ‘admin-dashboard’. |
| 7 | Medium | **Gérer les Groupes** | Même travaille de la tache ‘**Gérer les Catégories’.** |
| -Même travaille de la tache ‘**Gérer les Catégories’** en utilisant le même service **‘service-manager’** et des nouvelles composant sous le module ‘admin-dashboard’. |
| 8 | Medium | **Publier des Sondages** | -Même travaille de la tache ‘**Gérer les Catégories’.** |
| -Même travaille de la tache ‘**Gérer les Catégories’** en utilisant le même service **‘service-manager’** et des nouvelles composant sous le module ‘admin-dashboard’. |
|  | mudium | **Répondre à un sondage** |  |
|  |
| 8 | high | **Consulter les Statistiques** | -Dans la classe userRepository on créer une méthode qui contient une requête SQL pour récupérer les nombre de registration par jour, par semaine et par mois. Dans la même classe ajouter une autre méthode qui récupérer les 10 plus formateurs noté par des étoiles.  -Dans la classe FormationRepository en créer une méthode qui contient une requête SQL pour récupérer les 10 meilleurs formation noté par des étoiles.  -On appelle ces méthodes dans un contrôleur et  -Ajouter les routes de chaque fonction.  -Finalement faire un test. |
| -Générer 3 nouvelles composant :  ‘chart-teacher’, registration-user,  ‘chart-cours’  -Comme toujours fait l’appelle des API’s dans  Le service ‘service-manager’.  -Installer la bibliothèque ‘chartJs’ puis utiliser ses composant pour générer des courbes. Voire la documentation de ‘chartJs’ pour voir comment intégrer son composant pour afficher des dynamiques données.  -Faire un test. |

## Diagramme des cas d’utilisation du deuxième sprint

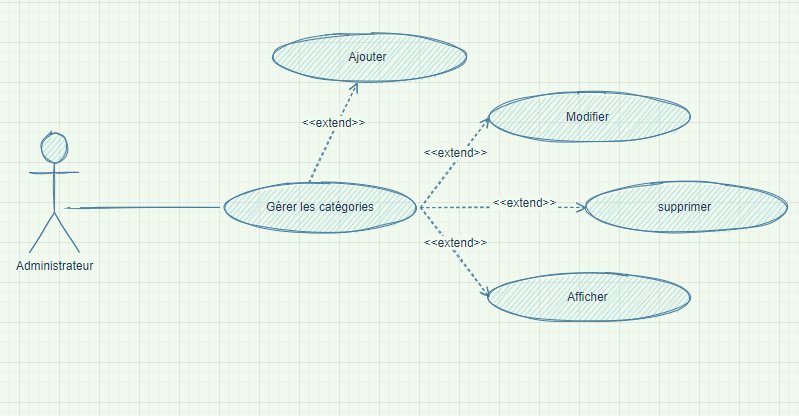


## Capture des besoins

Lors de cette partie, nous allons présenter les raffinements des cas d’utilisation de sprint 2 les plus délicats ainsi que leurs descriptions textuelles. Comme les cas (gestion des catégories, gestion des formations, gestion des formateurs, proposer un sondage) sont les mêmes que le processus de travail, nous choisissons de travailler sur un cas d'entre eux, qui est le cas de gestions des catégories.

#### Analyse du cas d’utilisation: Gérer les catégories

1. le raffinement du cas d’utilisation : Gérer les catégories



1. Description textuelles du cas d’utilisation : Ajouter une catégorie

|  |  |
| --- | --- |
| Cas d’utilisation | Ajouter objet |
| Acteur | L’administrateur |
| Description brève | L’administrateur peut ajouter une catégorie (formation ou catégorie ou formateur ou sondage) à la base de données. |
| Pré-condition | Administrateur authentifié. |
| Scénario Nominal | 1. L’administrateur demande la page d’ajout. 2. Le système affiche la page demandé. 3. L’administrateur remplie le formulaire. 4. Le système vérifie la saisit 5. Le système ajouter l’objet a la base de données. 6. Le système reset le formulaire et affiche un message de succès. |
| Scénario alternatif | 4.1 : formulaire invalide.  4.1.1 : Le système reset le formulaire et affiche un message d’erreur. |
| Post-condition | -Un objet est ajouté à la base de données. |

|  |  |  |  |
| --- | --- | --- | --- |
| Id story | **Priorité** | **Activité thème** | **Description** |
| 9 | Medium | **Visiter espace du formateur et Consulter des Statistiques** | -Dans la classe userRepository on créer une méthode qui contient une requête SQL pour récupérer les nombre étoiles de formateur courant. Dans la même classe ajouter une méthode pour récupérer le nombre des formations associé par le formateur courant  -Dans la classe FormationRepository on créer une méthode qui contient une requête SQL pour récupérer les nombre étoiles d’une formation identifié.  -On appelle ces méthodes dans un contrôleur et  ajouter les routes de chaque fonction.  -Finalement faire un test. |
| -Installer la bibliothèque ‘ng-apexcharts’ et ajouter le module ‘NgApexChartsModule’ dans un nouvelle module ‘teacher-dashboard’ qui contient toutes les composants associé par le formateur.  -Obtenir les données de la réponse du serveur, mettre ses donnes dans un tableau d’un nouveau composant ‘analytic’ puis à l’aide de l’implémentation de module ‘NgApexChartsModule’ afficher un graphique avec ces donnees.  -Faire un test. |
| 10 | high | **Gérer les taches a fait de chaque formation** | -Même travaille de la tache ‘**Gérer les Catégories’.** |
| -Générer un nouveau composant  ‘my-calender’  -Installer la bibliothèque JavaScript ‘fullcalender’. Voire la documentation principale de cette bibliothèque pour comprendre comment afficher le tableau des objets de type Calendriers de chaque formation dans un calendrier UI.  -Construire un petit formulaire pour l’ajout d’une tâche.  -Faire un test |
| 11 | high | **Gérer les chapitres d’une formation et son contenues (Vidéos, fichiers, document et quizz)** | -Des opérations CRUD pour les objets (chapitre, vidéos, fichier, documentation, quizz) : utiliser le même principe de la tache ‘**Gérer les Catégories’.**  -Faire un test. |
| -Installer la bibliothèque ‘ng-apexcharts’ et ajouter le module ‘NgApexChartsModule’ dans un nouvelle module ‘teacher-dashboard’ qui contient toutes les composants associé par le formateur.  -Obtenir les données de la réponse du serveur, mettre ses donnes dans un tableau d’un nouveau composant ‘analytic’ puis à l’aide de l’implémentation de module ‘NgApexChartsModule’ afficher un graphique avec ces données.  -Faire un test. |
|  |  | **Contrôler la visibilité d’une Formation** | -Simple méthode pour changer l’état d’une formation (visible ou non).  -Ajouter les routes de la fonction.  -Finalement faire un test. |
| Fait l’appelle d’API dans le ‘manager-service’  Et ajouter une méthode dans le composant ‘détails’ qui utilise l’appelle d’API et le relié par un évènement click d’un bouton. |
|  | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Id story | **Priorité** | **Activité thème** | **Description** |
| 15 | Medium | **Consulter la liste des formations disponibles** | -Utiliser l’annotation Groupe dans des attributs spécifiques dans la classe formation et les classes qui ont une relation avec cette classe pour sélectionné le contexte des données sérialiser qui sont revenir lors d’une requête clientèle.  -Ajouter les routes de la fonction.  -Finalement faire un test. |
| Génère un nouveau module ‘student’.  Générer une nouvelle interface Cours qui prend le même nom et type des attributs du contexte revenir du serveur. Puis utilisé cette interface comme un type des données revenir du serveur (‘généricité’).  Ajouter deux composants ‘list-cours’ qui prend le tableau d’objet et un composant fils ‘cours’ qui affiche un objet du tableau.  -Assurer que seulement les formations avec l’attribut ‘visible = true’ sont affichées aux visiteurs de l’application.  -Assurer que seulement les formations avec des groupes pas pleines sont affichées aux visiteurs de l’application.  -Assurer que seulement  Note : Faire une boucle et passer chaque objet du tableau au composant fils et recevoir cet objet à l’aide de l’annotation input puis l’afficher.  -L’affichage des détails de chaque formation sera dans le même composant ‘cours’ mais dans un modal.  Avoir un design agréable.  Faire un test. |
| 16 | Medium | **Filterer les formations** | -A l’aide de la Doctrine bundle utiliser la fonction findBy () et filtrer puis récupérer les données du la base (par nom, par catégorie, par type et par prix). Assure que chaque filtre est dans une méthode.  -Ajouter les routes.  -Faire un test. |
| -Dans cette partie on a deux choix a essayé et finalement en prend le choix le plus performant  -Choix 1 : utiliser les REST API’s pour faire la filtre à chaque requête clientèle sortant à partir les champs de la composants ‘list-cours’ (balise input).  -Choix 2 : Développer un pipe qui filtrer le tableau sans utiliser les API’s de filtre (utiliser seulement l’API qui appelé toute les objets sans filtre) : Cette possibilité est plus attirante car le nombre des requêtes sur le serveur va diminuer  et toutes les filtre génèrent coté client (Navigateurs).  -Faire un test. |
| 17 | high | **Gérer le panier** | -Dans cette partie il est préférable d’utiliser la session pour stocker un panier avec ces éléments (id formation) pour qu’un utilisateur  gérer son panier sans s’authentifier.  - Fonction 1 : créer une nouvelle session qui enregistre l’id d’une ou plusieurs formations.  - Fonction 2 : récupérer toutes les id de la session et revenir la formation de chaque id.  - Fonction 3 : supprimer un seule id stocké dans la session.  - Fonction 4 : vider toutes la session.  -Ajouter les routes de chaque fonction.  - Faire un test avec Postman. |
| -Créer un nouveau service ‘cart-service’ et fait l’appelle des REST API’s dans des méthodes.  -Vérifier que le navigateur enregistre un cookie  dans le stockage des cookies lors de l’appelle de la méthode qui est responsable d’envoyer une requête qui contient l’id d’une formation pour enregistrer dans la session.  -Créer un nouveau composant ‘cart-manage’  et injecter le ‘cart-service’ puis fais l’appelle de ses méthodes dans des différents fonctions et finalement préparé le design et associé ces fonction avec des évènements qui génèrent par boutons et afficher les formations.  -Faire un test. |
|  | high | **Valider un Panier avec un système de paiement en ligne** | -Ajouter une fonction qui récupère toute les id existes dans la session puis persister ses id vers la table commande avec l’id d’utilisateur courant et un statut ‘nonValid’ et renvoie l’id de la commande comme réponse.  -Développer une fonction pour récupérer le résultat de la procédure de paiement.  -Si le résultat est valide alors change le statut de la commande en statut ‘valid’, si non ne fait rien.  -Si la formation qui l’utilisateur veut l’acheter est une formation en ligne et le résultat de paiement est valide alors ajouter l’utilisateur automatiquement a un groupe de la formation.  -Faire un test |
| -Ajouter une fonction qui Récupère l’id de la dernière commande a été créer de l’utilisateur courant, puis envoyer cet id avec le payload de l’API Konnect et Récupéré la réponse de l’API Konnect qui contient l’url du formulaire de paiement avec la référence de paiement courante et rediriger l’utilisateur à cet url.  -Ajouter une fonction qui Récupère la réponse du résultat d’API Konnect et appelle l’API de back-end qui est responsable de changer le statut de la commande et envoyer le résultat de paiement au back-end |

|  |  |  |  |
| --- | --- | --- | --- |
| Id story | **Priorité** | **Activité thème** | **Description** |
| 13 | High | **Discuter en ligne** | -Pour cette tache on utilise la bundle ‘mercure’ de symfony  (Voire la documentation principle de symfony).  -Faire un test. |
| -Développer une fonction qui fait l’appelle à l’API d’ajout des messages.  -Développer une fonction qui fait l’appelle à l’API de récupérations des messages dans une conversation.  -Abonnées les deux fonctions au protocole ‘mercure’ pour assurer la récupération des messages en temps réelle.  -Avoir un design agréable.  -Faire un test. |
| 14 | High | **Joindre une séance en ligne** | -Développer une fonction qui permettre à enregistre l’url d’une séance en ligne dans la base des données.  -Faire un test. |
| -Intégrer le SDK de l’application ZOOM dans Angular pour générer des url des séances en ligne. Puis appeler l’API pour sauvegarder l’url ZOOM de chaque formation en ligne dans la base de données.  -Préparer un design agréable.  -Faire un test. |
|  | Medium | **Publier des commentaires dans un chapitre** | -Développer des fonctions qui assurent les fonctionnalités suivantes :  -Ajouter un commentaire.  -Modifier un commentaire.  -Récupérer toutes les commentaires.  -Faire un test. |
| -Appeler les API’s du back-end puis préparer les interfaces nécessaires.  -Faire un test si l’utilisateur courant est l’écrivain d’un commentaire, alors affiche un bouton modifier pour la modification de son commentaire.  -Faire un test. |
| 20 | high | **Naviguer entre les chapitres** | - Développer des fonctions qui récupère les chapitres de chaque formation et ses continues. |
| -Appeler les API’s du back-end puis préparer les interfaces nécessaires.  -Faire un test. |
|  | medium | **Consulter mes formations** | - Développer une fonction qui récupère seulement les formations qui existent dans la table commande avec un ‘statut= valide’ pour l’utilisateur courant.  -Faire un test. |
| -Appeler l’API du back-end puis préparer les interfaces nécessaires pour l’affichage des formations.  -Faire un test. |
| 20 | medium | **Evaluer le formateur et la formation** | - Développer une fonction qui fait l’ajout ou la modification d’une évaluation en fonction du cas existant, si l’utilisateur n’a jamais évalué la formation, alors on a une création d’une nouvelle évaluation lorsque l’API réserve une requête si non l’évaluation d’utilisateur sera modifier.  -Même travail pour l’évaluation du formateur. |
| -Appeler les API’s du back-end puis préparer les interfaces nécessaires.  -Faire un test. |
|  | high | **Répondre à un quiz** | - Même travail de la tâche évaluer le formateur et la formation.  -Faire un test. |
| -Appeler les API’s du back-end puis préparer les interfaces nécessaires.  -Faire un test. |
| 21 | low | **importer un**  **fichier et un vidéo** | -Ajouter une fonction qui qui Récupère les fichiers et les vidéos d’un chapitre. |
| -Appeler les API’s du back-end puis préparer les interfaces nécessaires.  -Utiliser HTMLL 5 pour assurer le téléchargement des vidéos ou les fichiers.  -Faire un test. |

|  |  |
| --- | --- |
| Cas d’utilisation | Ajouter objet |
| Acteur | L’administrateur |
| Description brève | L’administrateur peut ajouter un objet (formation ou catégorie ou formateur ou sondage) à la base de données. |
| Pré-condition | Administrateur authentifié. |
| Scénario Nominal | 1. L’administrateur demande la page d’ajout. 2. Le système affiche la page demandé. 3. L’administrateur remplie le formulaire. 4. Le système vérifie la saisit 5. Le système ajouter l’objet a la base de données. 6. Le système reset le formulaire et affiche un message de succès. |
| Scénario alternatif | 4.1 : formulaire invalide.  4.1.1 : Le système reset le formulaire et affiche un message d’erreur. |
| Post-condition | -Un objet est ajouté à la base de données. |

|  |  |
| --- | --- |
| Cas d’utilisation | modifier objet |
| Acteur | L’administrateur |
| Description brève | L’administrateur peut modifier un objet (formation ou catégorie ou formateur) existe dans la base de données. |
| Pré-condition | Administrateur authentifié. |
| Scénario Nominal | 1. L’administrateur demande la page de la modification. 2. Le système affiche la page demandé avec un formulaire remplit par les informations actuel de l’objet. 3. L’administrateur modifie le formulaire. 4. Le système vérifie la saisit 5. Le système modifie l’objet dans la base de données. 6. Le système affiche un message de succès. |
| Scénario alternatif | 4.1 : formulaire invalide.  4.1.1 : Le système reset le formulaire et affiche un message d’erreur. |
| Post-condition | -Un objet est modifié dans la base de données. |

|  |  |
| --- | --- |
| Cas d’utilisation | consulter les statistiques |
| Acteur | L’administrateur |
| Description brève | L’administrateur peut consulter les statistiques de la plateforme. |
| Pré-condition | Administrateur authentifié. |
| Scénario Nominal | 1. L’administrateur demande la page des statistiques. 2. Le système récupère les données nécessaires puis les traduit sous forme des figures graphiques. 3. Le système affiche la page demandé. |
| Post-condition | - Les statistiques sont présentées à l’administrateur. |

|  |  |
| --- | --- |
| Cas d’utilisation | Répondre au sondage |
| Acteur | Le Candidat |
| Description brève | Le Candidat peut réagit a un sondage. |
| Pré-condition | Candidat authentifié et un sondage existe. |
| Scénario Nominal | 1. Le candidat click sur un bouton pour répond à un sondage. 2. Le système enregistre le réagit du candidat. |
| Scénario alternatif | 1.1 : Le candidat click plusieurs fois sur le bouton  1.1.1 : le système enregistre seulement la réponse du premier click sur le même bouton. |
| Post-condition | -Une réponse est ajoutée à la base de données. |

|  |  |
| --- | --- |
| Cas d’utilisation | Répondre au quiz |
| Acteur | Le Candidat |
| Description brève | Un Candidat peut répondre à un quiz. |
| Pré-condition | Un Candidat authentifié, Un Candidat abonne à la formation qui contient le quiz. |
| Scénario Nominal | 1. Le candidat répondre à une question de quiz. 2. Le système écoute à la réponse du candidat et affiche le nombre des points ajouté ou retiré selon la réponse du candidat. 3. Le candidat complète toutes les questions du quiz. 4. Le système affiche le résultat final du candidat. |
| Scénario alternatif | 2.1 : Lorsque le candidat n’a aucune réponse juste ou il est actuellement à la première question.  2.1.1 : Le système ne retire pas un nombre des points au candidat lorsque son réponse est fausse. |
| Post-condition | Le Candidat a réponde au quiz. |

|  |  |
| --- | --- |
| Cas d’utilisation | Evaluer formation et formateur |
| Acteur | Le Candidat |
| Description brève | Un Candidat abonné dans une formation alors il a le droit d’évaluée cette formation et son formateur. |
| Pré-condition | Un candidat abonné dans une formation et authentifiée. |
| Scénario Nominal | 1. Le candidat click sur un bouton. 2. Le système affiche un modal qui contient un champ de commentaire et un champ des étoiles, ces champs sont remplie par l’évaluation du candidat si il est déjà fait une évaluation. 3. Le candidat met un nombre des étoiles et laisse un commentaire pour la formation ou le formateur ou tous les deux. 4. L’évaluation du candidat sauvegarde dans la base de données. |
| Scénario alternatif | 3.1 : le candidat est déjà fait une évaluation.  3.1.1 : son évaluation est modifiée par la nouvelle. |
| Post-condition | Le Candidat a réponde au quiz. |

|  |  |
| --- | --- |
| Cas d’utilisation | Ajouter des commentaires dans un chapitre. |
| Acteur | Le Candidat et le Formateur |
| Description brève | Un Candidat abonné dans une formation et le présentateur de la formation peut ajouter des commentaires dans chaque chapitre. |
| Pré-condition | Un candidat abonné dans une formation et le présentateur de cette formation sont authentifié |
| Scénario Nominal | 1. L’acteur ajoute un commentaire 2. Le système ajoute ce commentaire à la base de données et l’affiche à la liste des commentaires de chapitre courant. |
| Post-condition | Un commentaire a été ajouté à la base de données. |

|  |  |
| --- | --- |
| Cas d’utilisation | Consulter la liste des formations |
| Acteur | Le Candidat |
| Description brève | Un candidat peut consulter la liste des formations pour explorer les formations existes dans l’application. |
| Pré-condition | Le candidat demande la liste des formations |
| Scénario Nominal | 1. Le système vérifié la visibilité de la formation (et sont nombre des places vides si elle est en ligne). 2. Le système rendre seulement les formations visibles (et contient des nombre des places vides dans ses groupes si elle est en ligne). |
| Post-condition | Le Candidat a réponde au quiz. |

|  |  |
| --- | --- |
| Cas d’utilisation | Filtrer les formations |
| Acteur | Le Candidat |
| Description brève | Un candidat qui visite l’application il peut fait des filtres pour récupérer des formations spécifiques. |
| Pré-condition | Le candidat demande la liste des formations |
| Scénario Nominal | 1. Le candidat choisir le type de filtre. 2. Le système renvoie seulement les formations remplissant la condition du filtre. |
| Scénario Alternatif | 1.1 : Le type de filtre ne correspondant avec aucune formation.  1.1.1 : Le système rendre une page vide avec un message. |
| Post-condition | Le nombre des formations afficher diminue ou sera zéro. |

|  |  |
| --- | --- |
| Cas d’utilisation | Ajouter formation au panier |
| Acteur | Le Candidat |
| Description brève | Un candidat qui visite l’application il peut ajouter une formation au panier.. |
| Pré-condition | Le candidat est dans la page de la liste des formations. |
| Scénario Nominal | 1. -Le candidat click sur un bouton ajouter au panier. 2. Le système vérifié si la formation est déjà dans le panier ou non. 3. Le système ajoute la formation au panier. 4. L’icône du panier dans le nave-bar ajoute 1 a son nombre actuel. |
| Scénario Alternatif | 2.1 : La formation déjà existe dans le panier.  2.1.1 : Le système n’ajoute pas la formation au panier. |
| Post-condition | Une formation est ajoutée dans le panier. |

|  |  |
| --- | --- |
| Cas d’utilisation | Valider le panier |
| Acteur | Le Candidat |
| Description brève | Un candidat peut valider son panier. |
| Pré-condition | Le candidat est dans la page de panier. |
| Scénario Nominal | 1. Le candidat s’authentifier 2. Le candidat click sur le bouton ‘valider panier’ 3. Le système vérifié la dernière commande de courant utlisateur 4. Le système crée une nouvelle commande et rendre son id. 5. Le système redirige l’utlisateur à la page de paiement 6. Le candidat remplir le formulaire du paiement et confirmer le paiement 7. Le système redirige l’utilisateur à la page du résultat du paiement avec la référence de paiement et ses détails. 8. Le système vérifié le résultat de paiement. 9. Le système affiche le résultat de paiement. 10. Le système modifie le statut de la commande en statut valide. |
| Scénario Alternatif | 3.1 : la commande non nulle.  3.1.1 : Le système vérifié si la commande est déjà validée, si oui il affiche un message au candidat ‘votre panier est déjà validé’.  3.1.2 : Le système vérifié si la commande |
| Post-condition | Un panier est validé, alors le candidat peut accède au formations qui sont validé dans ce panier. |