

# RAPPORT DE STAGE DE FIN D'ÉTUDES

Présenté en vue de l'obtention du  
Diplôme national de licence en science de l'informatique  
Spécialité : Computer science

Par :

**Rayen BEN HASSINE**

## Conception et développement d'une plateforme e-learning

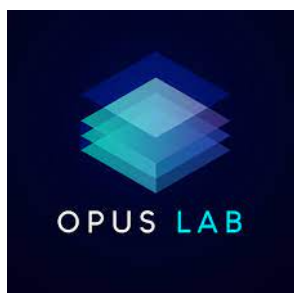
Encadrant professionnel :

**Mr. Sayed HAMDİ**

Encadrante académique :

**Mme. Nadia BRIDAA**

Réalisé au sein de Opuskills lab



Année universitaire : 2021/2022

J'autorise l'étudiant à faire le dépôt de son rapport de stage en vue d'une soutenance.

Encadrant professionnel, **Sayed Hamdi**

**Signature et cachet**

  
**STE opuskills Lab**  
N°7, Rue Kairouan, 1002 Tunis  
MF : 1716681 Y/A/M/000

J'autorise l'étudiant à faire le dépôt de son rapport de stage en vue d'une soutenance.

Encadrante académique, **Nadia Bridaa**

**Signature**



---

## DÉDICACE

“ Avec l'expression de ma reconnaissance, Je dédie ce travail à tous ceux qui me sont chers,

***À mes chers parents,***

*Que ce travail soit l'expression de ma reconnaissance pour vos sacrifices consentis, votre soutien moral et matériel que vous n'avez cessé de prodiguer tout au long de mes années d'études. Vous avez tout fait pour mon bonheur et ma réussite. Que dieu vous préserve en bonne santé et vous accorde une longue vie.*

***À ma famille et mes amis,***

*Je vous dédie ce travail pour vos encouragements et le soutien que vous m'avez apporté.*

***À l'équipe d'OPUS LAB,***

*Pour leurs efforts continus, les moments que nous avons vécu ensemble et les souvenirs qu'on a eu au cours de ce stage.*

”

**- Ben Hassine Rayen**

---

# REMERCIEMENTS

Tout d'abord, je souhaite adresser mes plus sincères remerciements à tous les membres du jury d'avoir accepté d'évaluer ce travail.

Je tiens à remercier madame **Nadia BRIDAA**, mon encadrante académique au sein de l'ISI pour son aide et pour les conseils fructueux qu'elle n'a cessé de me prodiguer tout au long de ce stage.

Je tiens à adresser mes sincères remerciements à monsieur **Ahmed BEN BOUZID** pour m'avoir accepté comme stagiaire et de m'avoir si bien accueilli et permis de développer mes connaissances au domaine du développement web.

Ainsi, j'adresse mes profondes gratitudes et respectueuse reconnaissance à mon encadrant professionnel monsieur **Sayed HAMD**I pour sa bonne volonté d'accepter de m'encadrer, pour tout le temps qu'il m'a octroyé et pour tous les conseils qu'il m'a prodigué.

Mes remerciements s'adressent aussi à notre CMO et designer graphique **Siwar GHARBI** pour son professionnalisme, le temps qu'elle nous a consacré, les conseils et pour tous les bons moments qu'on a partagé ensemble.

Enfin, je tiens à remercier profondément tous les enseignants de l'ISI pour la qualité de la formation qu'ils nous ont fourni tout au long de notre cursus universitaire.

---

# TABLE DES MATIÈRES

<b>Dédicace</b>	<b>ii</b>
<b>Remerciements</b>	<b>iii</b>
<b>Introduction générale</b>	<b>1</b>
<b>1 Étude préalable</b>	<b>3</b>
Introduction . . . . .	4
1 Organisme d'accueil . . . . .	4
1.1 Présentation de l'organisme d'accueil . . . . .	4
1.2 Organigramme . . . . .	5
2 Problématique . . . . .	5
3 Etude et critique de l'existant . . . . .	6
3.1 Etude de l'existant . . . . .	6
3.2 Critique de l'existant . . . . .	8
4 Solution proposée . . . . .	9
4.1 Contexte de la solution . . . . .	9
4.2 Présentation de la solution . . . . .	9
4.3 Apports de la solution . . . . .	10
5 Méthodologie adoptée . . . . .	10
5.1 Pourquoi SCRUM? . . . . .	10
5.2 La Méthodologie SCRUM . . . . .	11

5.3	Rôles définis par SCRUM . . . . .	12
5.4	Artéfacts de SCRUM . . . . .	12
5.5	Événements de SCRUM . . . . .	13
6	Environnement de développement . . . . .	14
6.1	Environnement matériel . . . . .	14
6.2	Environnement logiciel . . . . .	14
7	Architecture . . . . .	16
7.1	Architecture logique . . . . .	16
7.2	Architecture physique . . . . .	18
8	Choix technique . . . . .	19
	Conclusion . . . . .	22
<b>2</b>	<b>Planification et capture des besoins</b>	<b>23</b>
	Introduction . . . . .	24
1	Capture des besoins . . . . .	24
1.1	Besoins fonctionnels . . . . .	24
1.2	Besoins non fonctionnels . . . . .	25
1.3	Identification des acteurs . . . . .	25
2	Diagramme de cas d'utilisation . . . . .	26
3	Réalisation des prototypes des interfaces . . . . .	28
4	Pilotage du projet avec SCRUM . . . . .	29
4.1	Équipes et rôles . . . . .	29
4.2	Backlog du produit . . . . .	30
4.3	Planification des releases . . . . .	33
4.4	Découpage en sprints . . . . .	34
4.5	Planning de réalisation du projet . . . . .	35
	Conclusion . . . . .	36

---

## TABLE DES FIGURES

1.1	Logo d'Opuskills Lab . . . . .	4
1.2	Organigramme d'Opus lab . . . . .	5
1.3	Logo de Moodle . . . . .	6
1.4	Interface de DataCamp . . . . .	7
1.5	Interface de Taki Academy . . . . .	7
1.6	Les événements de SCRUM . . . . .	11
1.7	Logo Visual Studio Code . . . . .	14
1.8	Logo Google chrome . . . . .	14
1.9	Logo Git . . . . .	15
1.10	Logo Gitlab . . . . .	15
1.11	Logo Postman . . . . .	15
1.12	Logo Draw io . . . . .	15
1.13	Logo Adobe XD . . . . .	15
1.14	Logo Overleaf . . . . .	16
1.15	Logo Google sheets . . . . .	16
1.16	Logo Microsoft teams . . . . .	16
1.17	L'architecture MVVM . . . . .	17
1.18	L'architecture multi-couches . . . . .	18
1.19	Architecture 3-tières . . . . .	18
1.20	Logo de React JS . . . . .	19
1.21	Logo de Javascript . . . . .	19

1.22	Logo de Typescript . . . . .	19
1.23	Logo de Python . . . . .	19
1.24	Logo de Redis . . . . .	20
1.25	Logo de docker . . . . .	20
1.26	Logo de Amazon web services . . . . .	20
1.27	Logo de Redux . . . . .	20
1.28	Logo Tailwind CSS . . . . .	21
1.29	Logo Node JS . . . . .	21
1.30	Logo MongoDB . . . . .	21
1.31	Logo Express JS . . . . .	21
1.32	Logo Socket.io . . . . .	21
1.33	Fonctionnement des API Rest . . . . .	22
2.1	Identification des acteurs . . . . .	26
2.2	Diagramme de cas d'utilisation global . . . . .	27
2.3	Prototype de l'interface d'authentification . . . . .	28
2.4	Prototype de l'interface de création des parcours . . . . .	28
2.5	Prototype de l'interface du parcours de l'apprenant . . . . .	29
2.6	Découpage en sprints . . . . .	34
2.7	Besoins fonctionnels de chaque sprint . . . . .	35
2.8	Planning de réalisation du projet . . . . .	35



---

## LISTE DES TABLEAUX

1.1	Points forts et points faibles de Moodle . . . . .	8
1.2	Points forts et points faibles de Taki Academy . . . . .	8
1.3	Points forts et points faibles de DataCamp . . . . .	9
1.4	Rôles définis par SCRUM . . . . .	12
1.5	Environnement matériel . . . . .	14
2.1	L'équipe scrum . . . . .	29
2.2	Backlog du produit . . . . .	30

---

# LISTE DES ABRÉVIATIONS

- **API** = **Application Programming Interface**
- **AWS** = **Amazon Web Services**
- **B2B** = **Business to business**
- **B2C** = **Business to consumer**
- **CSS** = **Cascading Style Sheets**
- **HTML** = **HyperText Markup Language**
- **HTTP** = **HyperText Transfer Protocol**
- **JWT** = **JSON Web Tokens**
- **LMS** = **Learning management system**
- **LXC** = **Linux Containers**

- **MOOC** = **Massive Open Online Course**
- **MVVM** = **Model View ViewModel**
- **PHP** = **Hypertext Preprocessor**
- **QCM** = **Question à choix multiples**
- **QCU** = **Question à choix unique**
- **RBAC** = **Role-Based Access Control**
- **REST** = **Representational State Transfer**
- **S3** = **Simple Storage Service**
- **SCV** = **Système de contrôle de versions**
- **SPA** = **Single-page application**
- **TIC** = **Technologies de l'information et de la communication**
- **TICE** = **Technologies de l'information et de la communication pour l'enseignement**
- **UI/UX** = **User Interface/User experience**
- **UML** = **Unified Modeling Language**
- **V/F** = **Vrai/Faux**

---

# INTRODUCTION GÉNÉRALE

**A**l'aube de l'ère industrielle, l'homme fournit le grand espoir de devenir le maître absolu de toute chose. En effet, grâce au progrès technique qui a touché plusieurs domaines, l'humanité a réellement franchi un pas gigantesque vers la modernité et le bien-être.

Le secteur de l'enseignement n'échappe pas de cette transformation. En effet, nul ne peut nier que ce noble métier a beaucoup évolué dans la dernière décennie grâce à l'apparition des nouvelles technologies et techniques de télécommunication mais sa définition de base reste constante. Pour une approche simple du terme, il peut se définir comme une pratique, mise en œuvre par un enseignant et permettant de transmettre des connaissances à un élève, étudiant ou toute autre personne par le biais de la communication verbale et écrite.

Cependant, les avis sur la manière d'enseigner ne cessent de se multiplier. Certains estiment que les manières traditionnelles d'enseignement, qui sont basées sur le cours magistral, les travaux pratiques, le « tableau noir » et le support papier, sont beaucoup plus efficaces que celles basés sur les TICE. D'autres sont portés à croire que l'enseignement moderne facilite l'apprentissage, car il utilise les nouvelles technologies et surtout l'internet.

Avec le progrès scientifique, la pédagogie digitale est devenue l'une des formes d'apprentissage les plus adéquates pour la génération actuelle. L'enseignement a intégré les nouvelles technologies et fait appel à d'autres canaux de communication pour transmettre les connaissances, tels que le contenu audio-visuel, les animations 3D etc. La technologie favorise également l'apprentissage à distance et l'interaction au-delà du fait de partager un même espace physique en introduisant un nouveau concept : Le e-learning.

Ce terme peut être traduit par l'apprentissage électronique ou l'apprentissage virtuel. Il en-

globe toutes formes d'acquisition de connaissances et de compétences au moyen de technologies de l'information et de la communication (TIC) dans le but de favoriser les interactions en matière d'apprentissage, tant sur le plan du contenu, des activités et des outils d'apprentissage, qu'avec d'autres utilisateurs.

C'est dans ce cadre, la société OPUSKILLS LAB, nous a confié de mettre en place une plateforme e-learning ou un LMS dont l'objectif est de sortir du cadre d'enseignement classique et d'enseigner les skills digitales en ligne et d'une façon facile et moderne.

Notre rapport est structuré en six chapitres :

Le premier chapitre intitulé « **Étude préalable** » présentera l'organisme d'accueil et introduira la problématique, l'étude de l'existant et la solution proposée.

Le second chapitre « **Planification et capture des besoins** » sera la première partie dans l'application du cadre méthodologique SCRUM. Il exposera les fonctionnalités de l'application avec la spécification des besoins fonctionnels et non fonctionnels ainsi que la présentation des différents intervenants au sein du système. Il illustrera également le découpage du projet en sprints.

Le troisième chapitre représentera le premier sprint intitulé « **Authentification et gestion des comptes utilisateurs** ». Nous enchaînerons ensuite avec le quatrième chapitre qui illustre le deuxième sprint « **Gestion du contenu de la plateforme** ». Le cinquième chapitre contiendra le troisième sprint intitulé « **Exploitation du contenu de la plateforme** ». Enfin, dans le dernier chapitre on présentera notre quatrième sprint « Statistiques et forum de discussion ».

Ce rapport sera clôturé par une conclusion générale afin de récapituler tout le travail que l'on a réalisé, tout en énumérant les compétences, et l'expérience que nous avons acquis durant ces quatre mois de stage.

---

# CHAPITRE 1

---

## ÉTUDE PRÉALABLE

<b>Introduction . . . . .</b>	<b>4</b>
<b>1    Organisme d'accueil . . . . .</b>	<b>4</b>
<b>2    Problématique . . . . .</b>	<b>5</b>
<b>3    Etude et critique de l'existant . . . . .</b>	<b>6</b>
<b>4    Solution proposée . . . . .</b>	<b>9</b>
<b>5    Méthodologie adoptée . . . . .</b>	<b>10</b>
<b>6    Environnement de développement . . . . .</b>	<b>14</b>
<b>7    Architecture . . . . .</b>	<b>16</b>
<b>8    Choix technique . . . . .</b>	<b>19</b>
<b>Conclusion . . . . .</b>	<b>22</b>

# Introduction

Le premier chapitre sera consacré à la présentation du projet dans son contexte général. Dans un premier temps, nous allons présenter l'organisme d'accueil. Ensuite, nous allons dégager notre problématique et procéder à l'étude des solutions existantes pour formuler notre solution à adopter. Enfin, nous présentons le choix méthodologique et l'architecture de notre application.

## 1 Organisme d'accueil

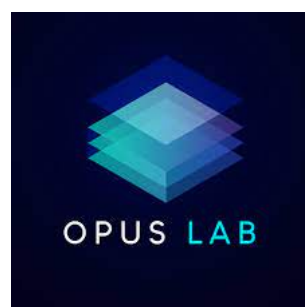
### 1.1 Présentation de l'organisme d'accueil

Opuskills lab est une startup fondée en Mars 2021 par ces trois co-fondateurs Ahmed Ben Bouzid, Siwar Gharbi et Sayed Hamdi.

C'est un cabinet de formation agile qui offre des formations en compétences digitales en B2C (pour les étudiants) et en B2B (pour les entreprises, les organismes, les associations...) dans les domaines suivants : Le développement web, le design graphique et le marketing digital

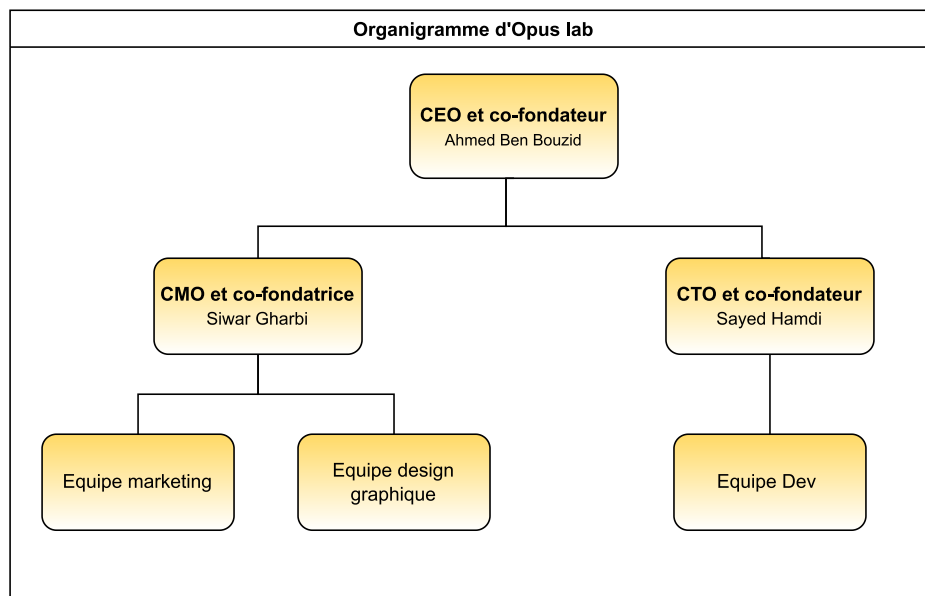
Opuskills lab travaille aussi avec des clients et des partenaires sur des projets dans le développement web, le design graphique et le marketing digital.

Son but est de délivrer le meilleur contenu et de fournir le cadre idéal qui favorise l'apprentissage et le partage afin d'aider les apprenants à développer leurs compétences digitales.



**FIGURE 1.1 :** Logo d'Opuskills Lab

### 1.2 Organigramme



**FIGURE 1.2 :** Organigramme d'Opus lab

## 2 Problématique

Nous avons constaté que le système éducatif, en particulier les étudiants en Tunisie, n'ont pas les compétences nécessaires pour décrocher des stages ou des emplois. Il y a un écart entre le marché et le système éducatif. Nous avons constaté aussi que pour améliorer leurs compétences plusieurs étudiants optent pour l'e-learning grâce à sa flexibilité en termes de temps et de lieu. Cependant, cette flexibilité peut conduire parfois à un manque d'action dans le sens où l'apprenant n'est plus motivé pour poursuivre son apprentissage. De plus, la mise en pratique des connaissances est très importante pour mémoriser plus facilement les nouvelles informations. Or ce critère, n'est pas toujours présent lorsqu'il s'agit d'enseignement en ligne. Enfin, lorsque les étudiants passent à ce type d'apprentissage, ils perdent un rituel vital de connexion avec des personnes partageant les mêmes idées et cet aspect psychologique nuit à la motivation et aux performances.

**Alors, que peut-on faire pour résoudre ces problèmes afin de permettre aux apprenants de réaliser leurs objectifs d'apprentissage et trouver facilement des emplois?**



### 3 Etude et critique de l'existant

L'étude de l'existant est une phase déterminante pour bien comprendre le fonctionnement des systèmes existants, dégager leurs points forts et leurs points faibles pour enfin proposer notre propre solution.

A ce stade, nous présenterons les solutions existantes et nous procéderons à une étude comparative de celles-ci.

#### 3.1 Etude de l'existant

Après une recherche approfondie, nous avons fini par trouver quelques applications qui peuvent nous aider à élaborer les différentes fonctionnalités de notre projet.

##### ◆ Moodle

« Moodle est une plateforme éducative en ligne. Aussi appelée Learning Content Management system ou LCMS. elle est distribuée gratuitement, écrite en PHP. Elle a été créée pour aider les enseignants à développer leurs cours en ligne et à fournir le matériel à leurs élèves rapidement et efficacement. elle est utilisée dans les universités, les cours en ligne, différents projets et même dans l'apprentissage mixte.»[?]



**FIGURE 1.3 :** Logo de Moodle

##### ◆ Datacamp

DataCamp est une plateforme MOOC « Massive Open Online Course », créée en 2014. L'entreprise offre des cours en ligne à plus de 350 milles apprenants à travers le monde. Les cours disponibles sont R, Python et la science des données sous forme de courtes vidéos.

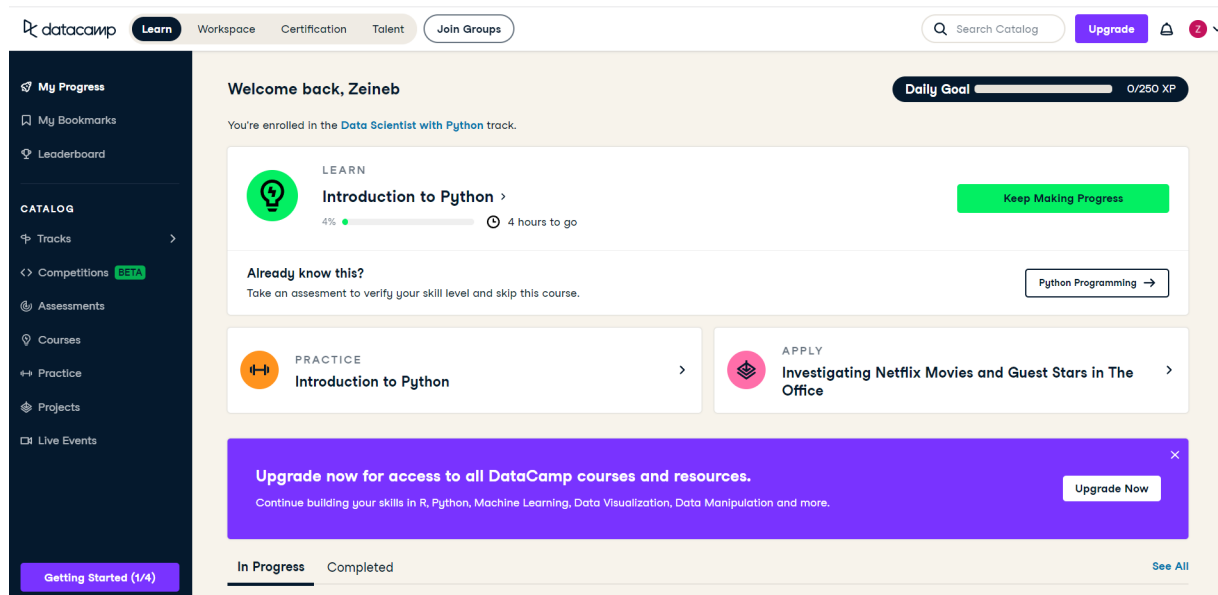


FIGURE 1.4 : Interface de DataCamp

### ◆ Taki Academy

Taki academy est une plateforme tunisienne d'enseignement en ligne destinée aux élèves de 4ème année primaire jusqu'au BAC et pour les étudiants en cycle préparatoire aux études d'ingénieur. L'enseignement sur la plateforme est basé des vidéos et des séances en direct encadrés par un groupe de professeurs.

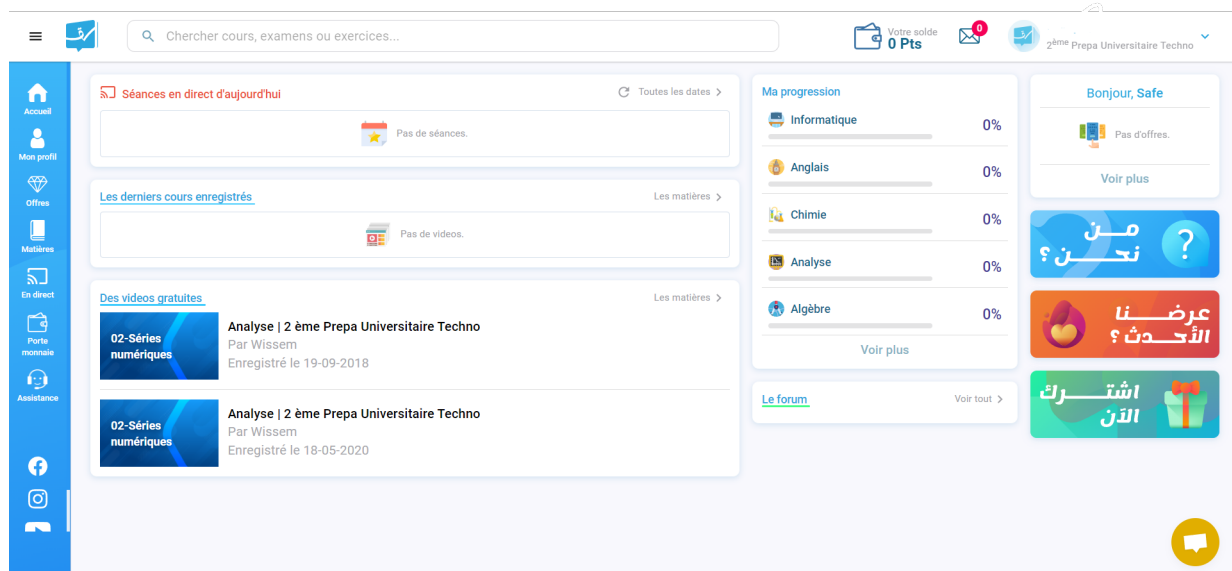


FIGURE 1.5 : Interface de Taki Academy

### 3.2 Critique de l'existant

**TABLEAU 1.1 :** Points forts et points faibles de Moodle

Points forts	Points faibles
Différents types de ressources : pages de textes, pages web, livres, questionnaires, tâches, forums, ateliers, débats, etc.	Complexité d'utilisation (pas intuitif) : multitude de paramètres spécifiques, étapes de configuration fastidieuses et difficultés à choisir les bons paramétrages.
Interactivité : favorise la communication enseignant-apprenant à l'aide de discussions, forums et wikis.	Absence du concept de ludification : pas de sources d'attraction pour l'apprenant.
Suivi pédagogique et examens : évaluation des travaux des apprenants par des enseignants.	Plateforme destinée seulement pour l'enseignement : on ne peut pas avoir des opportunités professionnelles.

**TABLEAU 1.2 :** Points forts et points faibles de Taki Academy

Points forts	Points faibles
Possibilité de suivre des cours en direct.	La plateforme est limitée à un seul style d'enseignement avec les vidéos. Ceci peut ne pas être la meilleure façon d'apprendre pour certains étudiants.
Interface ergonomique et facile à utiliser.	Pas de forum de discussion pour échanger des idées à propos d'un sujet.

**TABLEAU 1.3 :** Points forts et points faibles de DataCamp

Points forts	Points faibles
DataCamp propose de suivre une carrière conçue avec une combinaison de différents cours pour donner aux apprenants les connaissances nécessaires à cette carrière.	Certaines vidéos sont un peu courtes et manquent de détails.
Offre des questions de pratique et d'évaluation.	Les utilisateurs doivent vraiment creuser pour trouver des informations sur le plan d'essai gratuit.
Communauté active.	Il n'y a pas de tests formels pour évaluer vos progrès à la fin des cours.

## 4 Solution proposée

### 4.1 Contexte de la solution

Les formateurs d'Opus Lab ont commencé à enseigner des programmes à temps partiel le week-end dans la conception graphique, le développement Web et le marketing digital. Ces programmes ont rencontré le succès, plus d'étudiants voulaient s'inscrire et la vision s'est élargie.

L'équipe d'Opus Lab veut enseigner ses parcours disponibles sur une plateforme d'apprentissage en ligne qui met l'accent sur de l'apprentissage actif et qui propose des programmes approfondis dans les digital skills contenant des exercices et des mini-projets permettant aux étudiants de mettre en pratique leurs connaissances.

### 4.2 Présentation de la solution

« Opus Lab » est une plateforme e-learning spécialisée dans les compétences digitales (design graphique, développement web, marketing digital et intelligence artificielle).

La plateforme présente des formations approfondies et offre un système de suivi spécifique qui facilite l'auto-formation des apprenants.

### 4.3 Apports de la solution

La plateforme vise à favoriser une véritable expérience d'apprentissage pour les étudiants en offrant un contenu de qualité et un système de suivi continu. Elle se base sur l'apprentissage actif où l'apprenant met en pratique ses connaissances au fur et à mesure lors de sa formation.

Pour motiver les apprenants et les engager à bien assimiler le contenu des parcours disponibles, on va introduire le concept de la gamification (ou la ludification) qui est « une méthode qui consiste à appliquer les codes et mécanismes attachés au monde des jeux vidéo à des secteurs auxquels ils n'étaient pas destinés »[?]. En effet, au fur et à mesure de son apprentissage, l'apprenant collecte des points et obtient des récompenses. Notre but est de sortir du cadre d'enseignement classique et de rendre l'étudiant autonome en focalisant sur le concept d'apprendre en jouant.

Notre plateforme offre aussi la possibilité de participer à des missions professionnelles avec les partenaires d'Opus Lab. Ceci est une opportunité pour l'étudiant qui va lui permettre de s'ouvrir au monde professionnel et de concrétiser ses connaissances acquises sur des projets réels.

L'interaction entre l'apprenant et l'instructeur n'étant pas présente dans plusieurs environnements d'apprentissage en ligne, nous essayons dans notre plateforme d'établir et maintenir cette interaction à travers un forum de discussion où les participants peuvent échanger des idées et les apprenants peuvent poser leurs questions.

Enfin, notre plateforme vise aussi à rendre le travail de l'instructeur aussi simple que possible et lui permet de suivre l'avancement des étudiants et donner un feedback.

## 5 Méthodologie adoptée

Tout projet ayant un niveau de complexité considérable rend l'adoption d'une méthodologie de développement une nécessité pour garantir une qualité acceptable et éviter tout retard au niveau des délais. Suite à une réunion qu'on a fait avant le démarrage du projet, on a décidé de travailler avec SCRUM.

### 5.1 Pourquoi SCRUM?

SCRUM est la méthodologie la plus adaptée à la réalisation de notre projet, car elle se base sur l'implication du client dans le processus du développement, la répartition des tâches, la communication et la collaboration entre les différents membres du projet. Le principe de base est

simple : L'équipe avance ensemble et reste prête à réorienter le projet au fur et à mesure de sa progression.

### 5.2 La Méthodologie SCRUM

La méthodologie SCRUM est une méthodologie de gestion de projet agile qui définit le cadre des projets complexes. Cette approche permet à l'équipe de s'adapter rapidement aux changements que les clients effectuent régulièrement.



**FIGURE 1.6 :** Les événements de SCRUM

### 5.3 Rôles définis par SCRUM

**TABLEAU 1.4 :** Rôles définis par SCRUM

Rôle	Fonction
Product Owner	<p>Définit à quoi le produit doit ressembler et exprime clairement qu'elles sont les fonctionnalités qu'il doit contenir. C'est la seule personne responsable à :</p> <ul style="list-style-type: none"><li>• Développer et communiquer explicitement le but du produit.</li><li>• Créer et communiquer clairement les éléments du backlog produit.</li><li>• Ordonner les éléments du backlog produit par priorité.</li><li>• S'assurer que le backlog produit est transparent, visible et compris.</li></ul>
Scrum Master	<p>Assure le bon déroulement de la méthodologie de SCRUM et gère la coordination entre le product owner et l'équipe de développement.</p>
L'équipe de développement	<p>Constituée de 3 à 5 personnes qui réalisent le produit. Elles comptent satisfaire toutes les exigences techniques requises. Chacun doit être autonome et doit assurer le bon déroulement des tâches tout en respectant les délais prescrits.</p>

### 5.4 Artéfacts de SCRUM

Les artéfacts SCRUM sont basés sur un ensemble de valeurs, principes et pratiques qui fournissent la base de la philosophie agile. Ces artéfacts ont été spécialement conçus pour maximiser la transparence des informations afin que tout le monde ait la même compréhension de leurs définitions et de leurs utilités.

### ♦ **Le backlog du produit :**

Le backlog du produit contient l'ensemble des fonctionnalités à réaliser par l'équipe de développement sous forme de user stories classées par priorité.

### ♦ **Le backlog du sprint :**

Le backlog du sprint est une liste de user stories à réaliser dans un sprint qui dure de 2 à 4 semaines.

### ♦ **L'incrément produit :**

L'incrément en scrum correspond à l'ensemble des user stories du backlog du produit qui ont été accomplies pendant le sprint en cours.

## 5.5 Événements de SCRUM

### ♦ **La planification de sprint**

Avant de commencer chaque sprint, il faut faire une réunion de planification où on répond aux questions suivantes :

Quel est l'objectif de ce sprint?

Quels éléments prioritaires du product backlog peuvent être convertis en un incrément potentiellement livrable à la fin du sprint?

### ♦ **La mêlée quotidienne**

C'est une réunion quotidienne d'une durée maximale de 15 minutes. Elle permet à l'équipe de développement de mesurer son avancement chaque jour.

### ♦ **Le revue de sprint**

C'est une démonstration à la fin de chaque sprint où le product owner invite les parties prenantes pour recueillir auprès d'eux des feedbacks et renforcer la collaboration.

### ♦ **Le rétrospective de sprint**

Évaluer la qualité de travail réalisé pour tirer des leçons et les mettre au profit du sprint suivant, à travers l'élaboration d'un plan d'actions d'amélioration.



## 6 Environnement de développement

### 6.1 Environnement matériel

Pour la réalisation de notre travail nous avons utilisé deux ordinateurs portables avec les caractéristiques suivantes.

**TABEAU 1.5 :** Environnement matériel

Caractéristiques	Description
Processeur	Intel Core i5 11ème génération
RAM	16 Go
Disque dur	512 Go SSD
Système d'exploitation	Windows 11 professionnel

### 6.2 Environnement logiciel

#### ◆ Visual Studio Code



**FIGURE 1.7 :** Logo Visual Studio Code

C'est un environnement de développement intégré développé par Microsoft. il est extensible et supporte plusieurs langages.

#### ◆ Google chrome



**FIGURE 1.8 :** Logo Google chrome

Google Chrome est un navigateur Web multiplateforme développé par Google.

### ◆ Git



**FIGURE 1.9 :** Logo Git

C'est un système de contrôle de version (VCS) qui permet de suivre toutes les modifications faites sur le code. Il facilite aussi le travail d'une équipe de développeurs grâce à un système de branches.

### ◆ Gitlab



**FIGURE 1.10 :** Logo Gitlab

« Se basant sur les fonctionnalités du logiciel Git, elle permet de piloter des dépôts de code source et de gérer leurs différentes versions. »[?]

### ◆ Postman



**FIGURE 1.11 :** Logo Postman

C'est un outil gratuit permettant de tester des API.

### ◆ Draw io



**FIGURE 1.12 :** Logo Draw io

C'est un logiciel de dessin graphique multiplateforme gratuit et open source utilisé pour créer des diagrammes UML, des organigrammes, des diagrammes de réseau etc.

### ◆ Adobe XD



**FIGURE 1.13 :** Logo Adobe XD

C'est une solution d'UI/UX design permettant la conception des maquettes pour les interfaces utilisateur.

### ◆ Overleaf



**FIGURE 1.14 :** Logo Overleaf

C'est un éditeur LaTeX en ligne qui offre la possibilité de rédiger des documents d'une manière collaborative

### ◆ Google sheets



**FIGURE 1.15 :** Logo Google sheets

C'est un logiciel tableur propre à Google

### ◆ Microsoft teams



**FIGURE 1.16 :** Logo Microsoft teams

Outil de Microsoft permettant d'organiser des réunions, des conversations et des appels.

## 7 Architecture

### 7.1 Architecture logique

L'architecture logique est « ...la structure des composants d'un programme/système, leurs interrelations et les principes et lignes directrices gouvernant leur conception et leur évolution au fil du temps »[?]. Elle sert à décomposer notre système en des parties spécialisées ce qui facilite la réutilisation et permet aux développeurs de travailler sur des parties individuelles du système en isolation.

Le choix du style architectural est donc l'une des premières étapes du développement logiciel et intervient lors de la phase de conception.

### ♦ Architecture du frontend

Nous avons séparé la logique de présentation de notre application en trois couches selon l'architecture MVVM :

- **View** (la vue en français) : C'est l'interface qui contient les différents éléments graphiques (boutons, liens, listes...).
- **Model** (le modèle en français) : Il contient les données manipulées par le programme.
- **ViewModel** (le modèle de vue en français) : Il fait le binding (une liaison dynamique) entre la vue et le modèle de vue. C'est à dire le fait de modifier la vue va être répercuté sur le modèle et inversement.

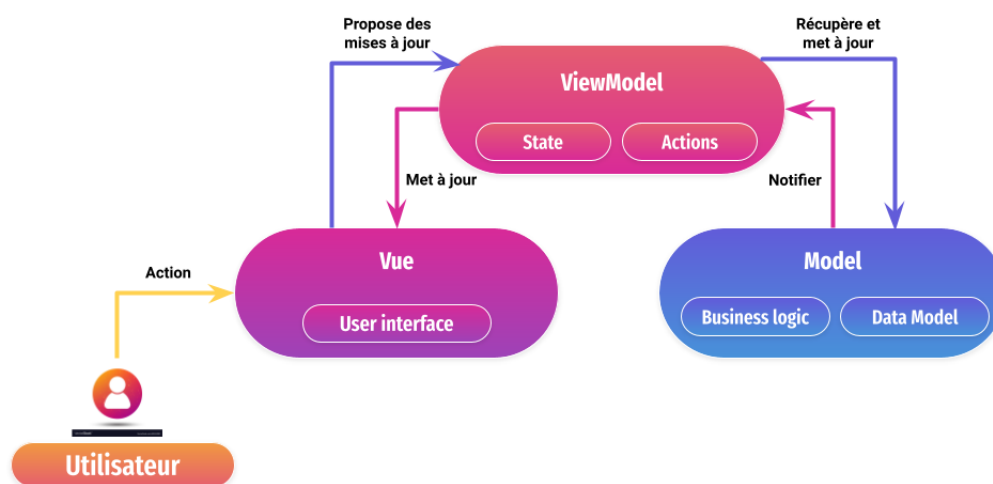


FIGURE 1.17 : L'architecture MVVM

### ♦ Architecture du backend

Pour le backend, on a utilisé une architecture multi-couches qui consiste à séparer les routes HTTP, la logique métier et les données ce qui facilite la réutilisabilité et la maintenabilité du code.

La figure ci-dessous explique l'architecture adoptée pour le backend.

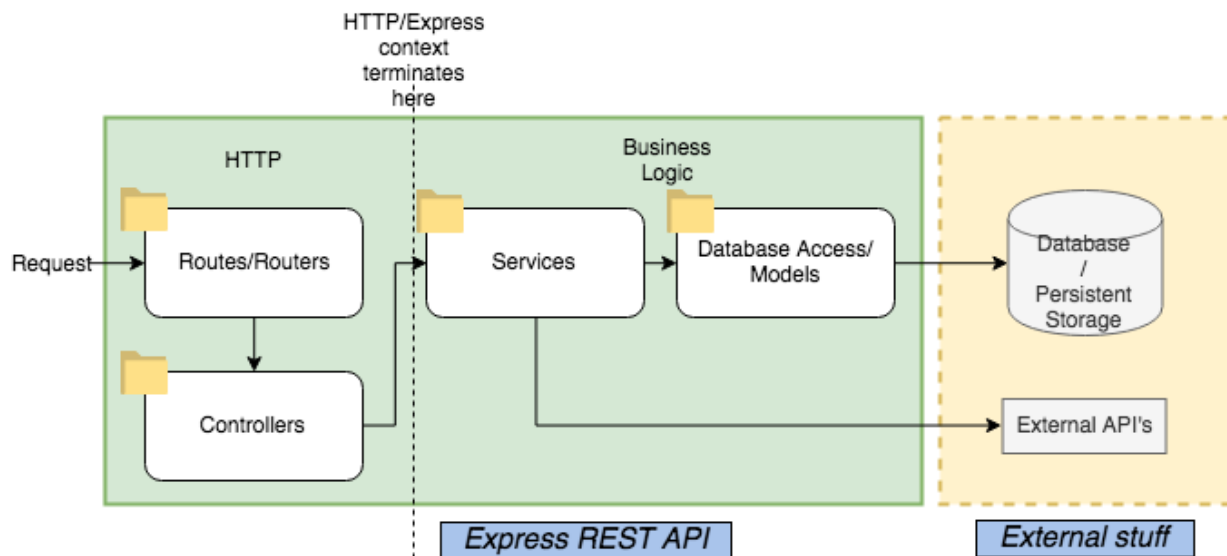


FIGURE 1.18 : L'architecture multi-couches

[?]

## 7.2 Architecture physique

L'architecture physique décrit tous les composants matériels qui supportent une application. Ces composants peuvent être des ordinateurs, des postes de travail, des périphériques de stockage etc.

Notre application utilise une architecture 3-tiers qui applique les principes suivants :

- **La présentation** : est toujours prise en charge par le poste client.
- **La logique applicative** : est prise en charge par un serveur intermédiaire.
- **Les données** : sont toujours gérées par un serveur de données.

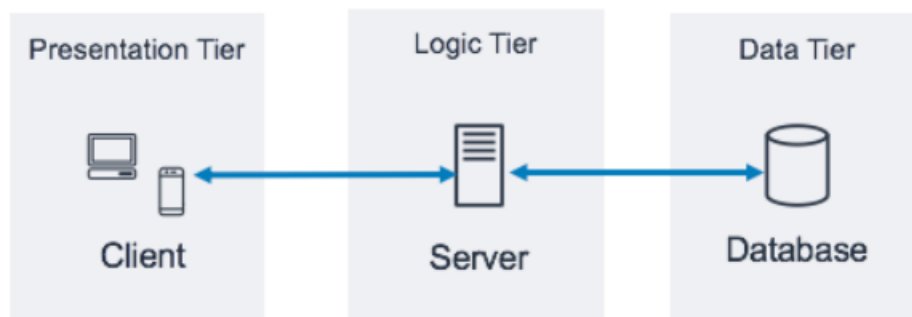


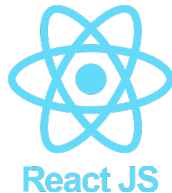
FIGURE 1.19 : Architecture 3-tières

[?]

### 8 Choix technique

Dans cette section, nous présentons la liste des technologies et langages de programmation utilisés pour le développement de ce projet.

#### ◆ React JS



**FIGURE 1.20 :** Logo de React JS

C'est une bibliothèque JavaScript développée par Facebook qui permet de développer des applications web monopage (SPa) en utilisant des composants réutilisables..

#### ◆ Javascript



**FIGURE 1.21 :** Logo de Javascript

JavaScript est un langage de programmation interprété, dynamique, compilé juste à temps et doté de fonctionnalités de premier ordre. Bien qu'il soit surtout connu comme un langage de script pour les pages web, de nombreux environnements non-browser tels que Node.js.

#### ◆ Typescript



**FIGURE 1.22 :** Logo de Typescript

TypeScript est un langage de programmation développé et maintenu par Microsoft. Il s'agit d'un sur ensemble de syntaxe stricte de JavaScript et ajoute un typage statique optionnel au langage. TypeScript est conçu pour le développement de grandes applications et se compile en JavaScript.

#### ◆ Python



**FIGURE 1.23 :** Logo de Python

Python est un langage de programmation de haut niveau. Sa philosophie de conception met l'accent sur la lisibilité du code avec l'utilisation d'une indentation importante. Son approche orientée objet vise à aider les programmeurs à écrire un code clair et logique pour des projets de petite et grande envergure.

### ◆ Redis



FIGURE 1.24 : Logo de Redis

C'est un système de stockage de données clé valeur en mémoire qui permet de réduire le temps de réponse et de traiter plusieurs requêtes.

### ◆ Docker

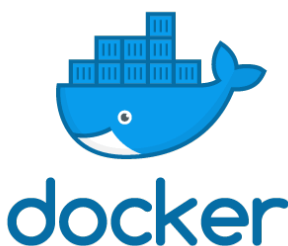


FIGURE 1.25 : Logo de docker

C'est une technologie virtualisation par conteneurs reposant sur LXC qui permet de créer des conteneurs qui contiennent des applications avec leurs dépendances.

### ◆ AWS



FIGURE 1.26 : Logo de Amazon web services

C'est un fournisseur de services cloud pour les entreprises et les particuliers.

### ◆ Redux



FIGURE 1.27 : Logo de Redux

Redux est une bibliothèque JavaScript open source qui permet la gestion d'états des applications web.

### ◆ Tailwind CSS



**FIGURE 1.28 :** Logo Tailwind CSS

Framework CSS facile à utiliser et permet de personnaliser le design d'une page web.

### ◆ Node JS



**FIGURE 1.29 :** Logo Node JS

NodeJS est un environnement d'exécution permettant d'utiliser le JavaScript côté serveur.

### ◆ Mongo DB



**FIGURE 1.30 :** Logo MongoDB

C'est un système de gestion de base de données NoSQL basé sur les documents.

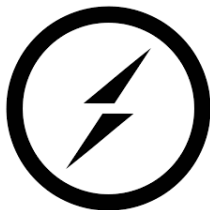
### ◆ Express JS



**FIGURE 1.31 :** Logo Express JS

Framework pour construire des applications web basées sur NodeJS.

### ◆ Socket.io



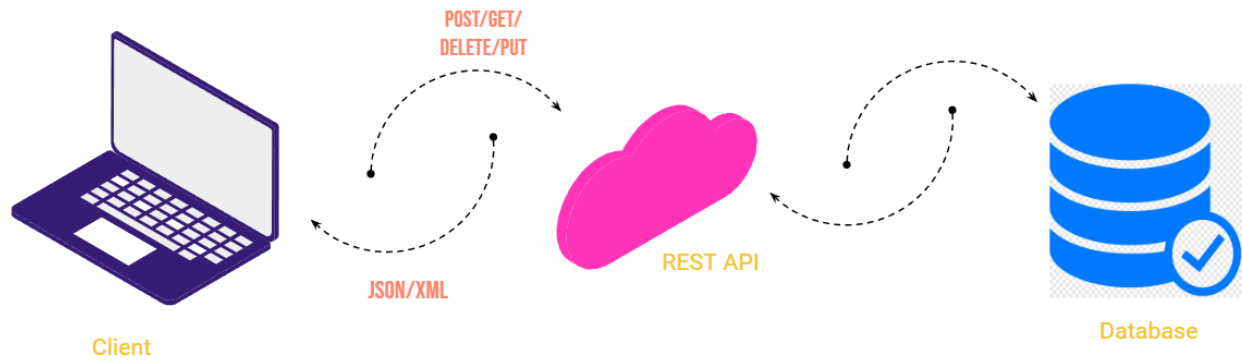
**FIGURE 1.32 :** Logo Socket.io

C'est une bibliothèque Javascript qui permet d'établir une communication bidirectionnelle entre les clients Web et les serveurs pour développer des applications en temps réel.



### ◆ Rest API

REST est un style architectural reposant sur le protocole HTTP et permet de créer des services web.



**FIGURE 1.33 :** Fonctionnement des API Rest

## Conclusion

Dans ce chapitre, nous avons présenté le contexte du projet. Puis, nous avons procédé à une étude des solutions existantes tout en fournissant les critiques nécessaires. Nous avons également justifié le choix de la méthodologie de travail utilisée ainsi que l'architecture de notre application. a ce niveau, nous pouvons désormais passer au prochain chapitre, qui contiendra une planification du projet et une identification des fonctionnalités de notre application.

---

## CHAPITRE 2

---

# PLANIFICATION ET CAPTURE DES BESOINS

<b>Introduction . . . . .</b>	<b>24</b>
<b>1    Capture des besoins . . . . .</b>	<b>24</b>
<b>2    Diagramme de cas d'utilisation . . . . .</b>	<b>26</b>
<b>3    Réalisation des prototypes des interfaces . . . . .</b>	<b>28</b>
<b>4    Pilotage du projet avec SCRUM . . . . .</b>	<b>29</b>
<b>Conclusion . . . . .</b>	<b>36</b>

# Introduction

Dans ce chapitre, nous allons élaborer la première phase de la méthodologie scrum. En premier lieu, nous allons déterminer les exigences fonctionnelles et non fonctionnelles du projet. Ensuite, on enchaînera par l'identification des acteurs ainsi que l'élaboration du backlog du produit. a la fin de ce chapitre, nous aurons comme finalité le découpage du projet en releases et sprints afin de bien planifier notre travail.

## 1 Capture des besoins

### 1.1 Besoins fonctionnels

La spécification des besoins fonctionnels est une phase indispensable qui détermine les principaux objectifs de l'application.

Notre application doit satisfaire les besoins fonctionnels suivant :

- ◆ **Authentification** : Chaque utilisateur doit s'authentifier pour pouvoir accéder à son compte.
- ◆ **Gestion des comptes utilisateurs** : la création des compte des apprenants, instructeurs, créateurs de cours et administrateurs ainsi que la modification des informations du profil.
- ◆ **Gestion des parcours à enseigner** : création des parcours formés par des super skills, des skills, des quizzes, des checkpoints et un projet final.
- ◆ **Gestion des challenges** : dépôts des challenges avec leurs points requis sur la plateforme.
- ◆ **Affectation des apprenants et des instructeurs aux parcours disponibles.**
- ◆ **Suivi du parcours par l'apprenant** : l'apprenant doit suivre son parcours étape par étape.
- ◆ **Passage des tests** : l'apprenant peut passer des quizzes, des checkpoints ou un projet final.
- ◆ **Évaluation des tests des apprenants** : l'instructeur doit pouvoir noter les travaux des apprenants dans les checkpoints et les projets finaux.
- ◆ **Calcul du score de l'apprenant** : notre application doit permettre à l'apprenant d'avoir un score calculé en fonction de son avancement dans le parcours.
- ◆ **Statistiques** : notre application doit permettre à chaque acteur de visualiser des courbes et des statistiques utiles.
- ◆ **Forum de discussion** : les apprenants et les instructeurs peuvent participer à un forum.

### 1.2 Besoins non fonctionnels

Les besoins non fonctionnels sont importants car ils agissent d'une façon indirecte sur le résultat et ils ont un aspect visible pour l'utilisateur, ce qui fait qu'ils ne doivent pas être négligés.

Notre application doit satisfaire les exigences non fonctionnelles suivantes :

- ◆ **L'ergonomie** : l'interface homme-machine de notre application doit satisfaire les critères de l'ergonomie (l'adaptabilité, le guidage, la cohérence, la gestion des erreurs...)
- ◆ **La sécurité** : notre application doit être protégée contre les attaques car elle contient des données confidentielles.
- ◆ **La maintenabilité** : le code de l'application doit respecter les bonnes pratiques pour faciliter la maintenance.
- ◆ **La rapidité** : notre application doit minimiser le temps de réponse et être rapide et performante.
- ◆ **La compatibilité** : l'application doit être compatible avec tous les navigateurs web.

### 1.3 Identification des acteurs

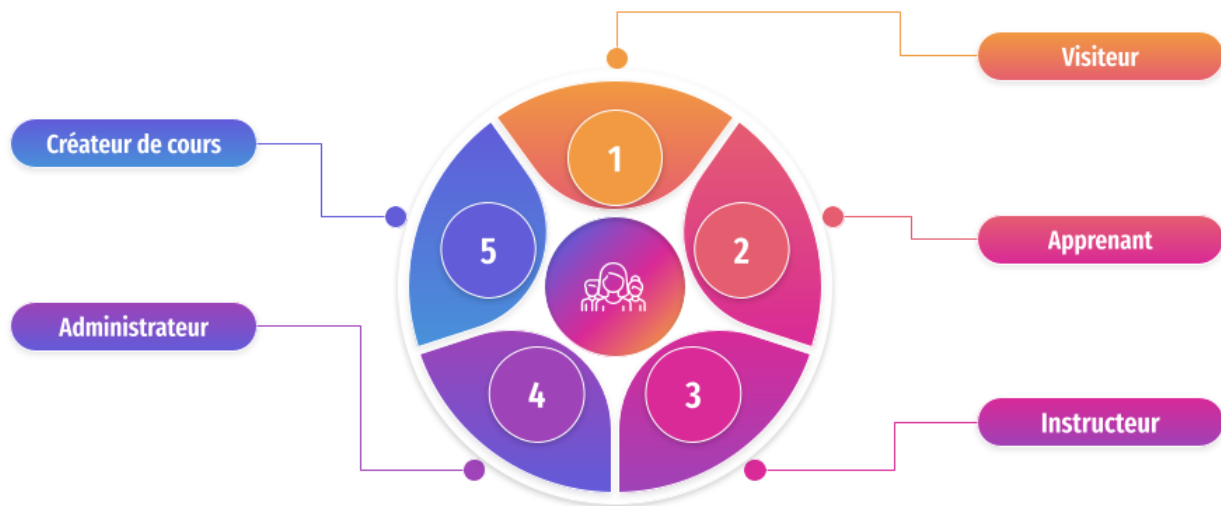
Dans cette étape, nous allons identifier les acteurs qui vont interagir avec le système.

Pour chaque acteur, nous définissons les différents objectifs qu'il tente d'atteindre en utilisant la plateforme.

Les acteurs principaux sont les suivants :

- ◆ **Le visiteur** : cet acteur peut créer un compte pour qu'il puisse s'authentifier en tant qu'un nouvel apprenant.
- ◆ **L'apprenant** : cet acteur doit s'authentifier pour pouvoir s'inscrire à un parcours à l'aide d'un code voucher qu'il doit l'acheter à l'avance. Une fois inscrit, il peut suivre son parcours, passer des tests et participer à des challenges.
- ◆ **Le créateur de cours** : c'est le responsable de la gestion du contenu de la plateforme. Il gère le contenu de chaque parcours et dépose les nouveaux challenges.
- ◆ **L'instructeur** : affecté à un parcours, il peut consulter son contenu, voir la liste des apprenants inscrits à ce dernier, suivre l'avancement de chaque apprenant et noter les check-points, les challenges et les projets finaux.

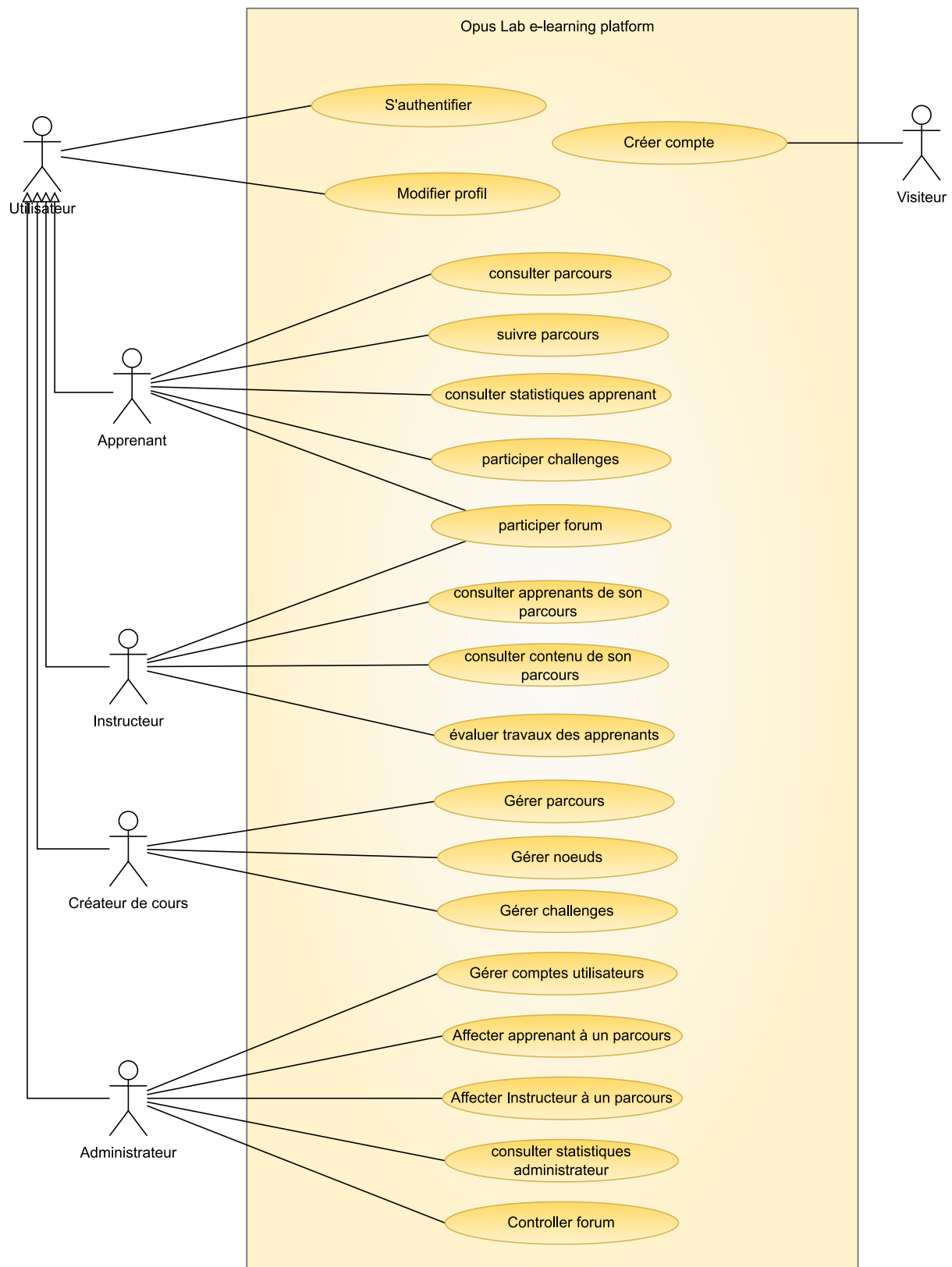
- ♦ **L'administrateur** : il doit s'authentifier pour pouvoir gérer les différents utilisateurs de la plateforme. Il peut ajouter ou modifier les informations d'un compte utilisateur. Il peut aussi affecter un apprenant ou un instructeur à un parcours.



**FIGURE 2.1** : Identification des acteurs

## 2 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation sert à définir le système, les acteurs, les cas d'utilisations et les liens entre acteurs et cas d'utilisation. La figure suivante présente le diagramme de cas d'utilisation global de notre application.

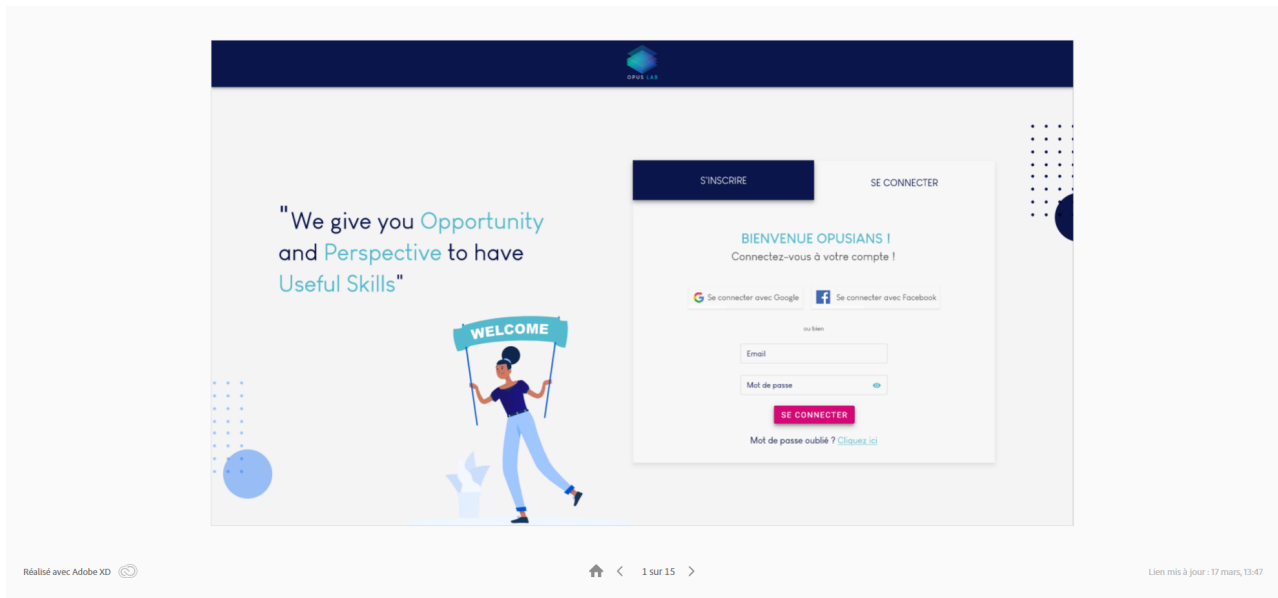


**FIGURE 2.2 :** Diagramme de cas d'utilisation global

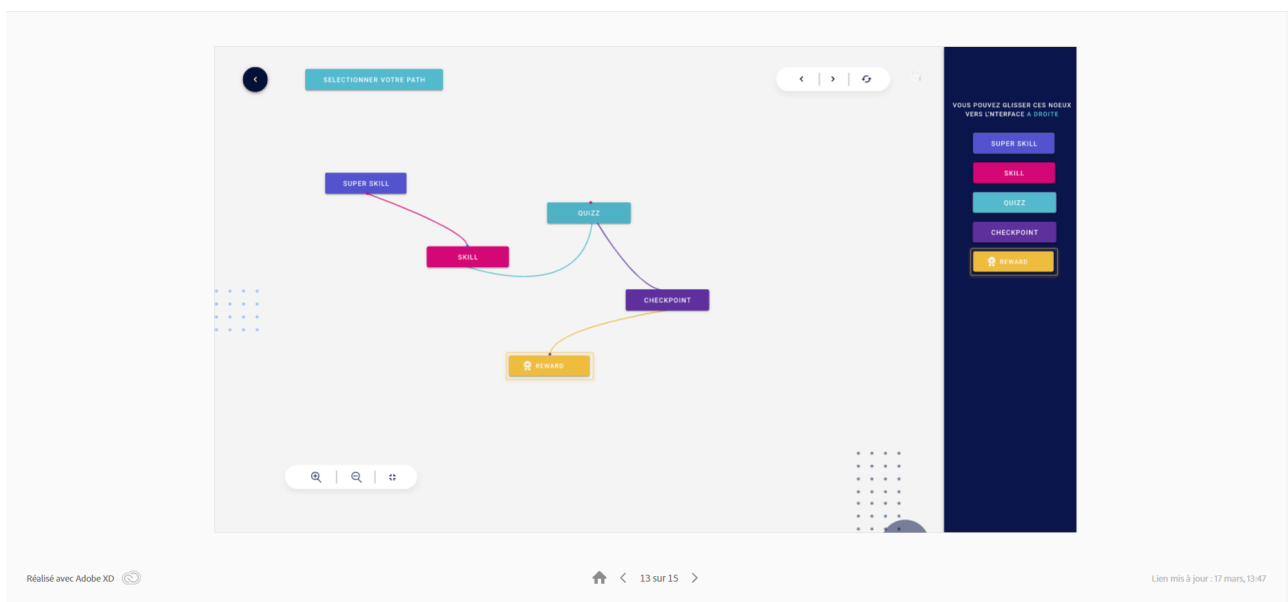
### 3 Réalisation des prototypes des interfaces

Cette étape est réalisée par notre designer graphique qui nous a élaboré les prototypes des interfaces de l'application tout en respectant les règles de l'ergonomie et la charte graphique de la société.

Nous présentons ci-dessous les prototypes de quelques interfaces :



**FIGURE 2.3 :** Prototype de l'interface d'authentification



**FIGURE 2.4 :** Prototype de l'interface de création des parcours

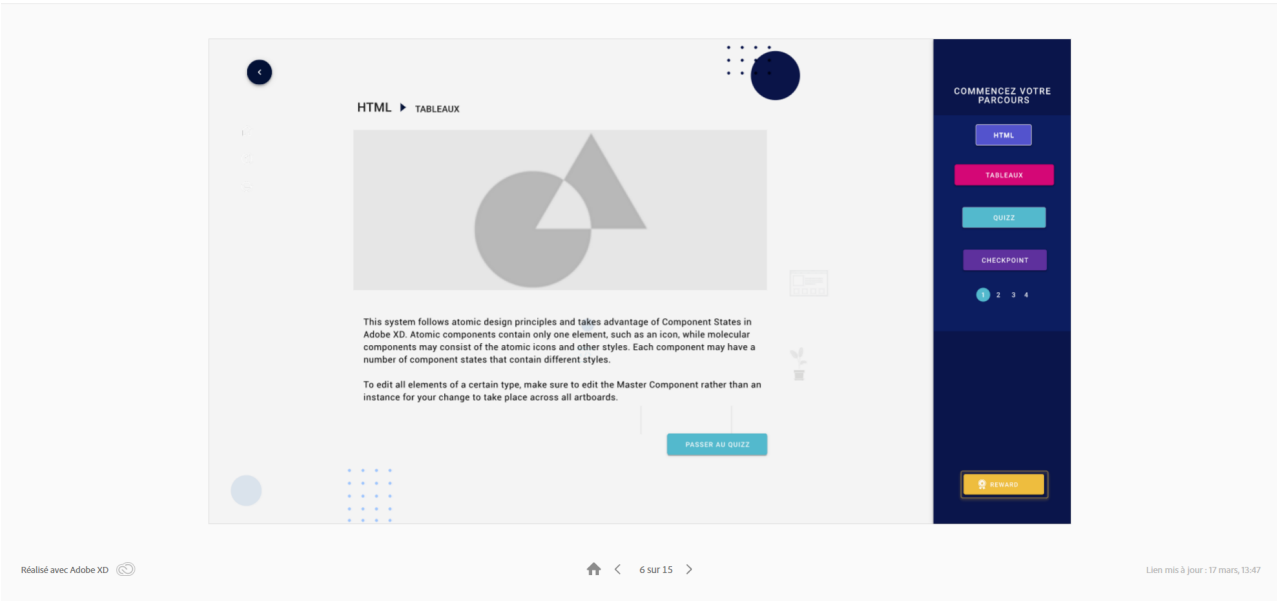


FIGURE 2.5 : Prototypage de l'interface du parcours de l'apprenant




4 Pilotage du projet avec SCRUM

4.1 Équipes et rôles

«L'unité fondamentale de scrum est une petite équipe de personnes, une équipe scrum. L'équipe scrum est composée d'un scrum Master, d'un Product Owner et de développeurs. Au sein d'une équipe scrum, il n'y a pas de sous-équipes ou d'hiérarchies. Il s'agit d'une unité cohérente de professionnels concentrés sur un objectif à la fois, l'objectif du produit.»[?]

La méthode SCRUM intègre généralement la participation de plusieurs acteurs. Dans notre contexte il s'agit de :

TABLEAU 2.1 : L'équipe scrum

		
<b>Product owner</b>	<b>Scrum master</b>	<b>Equipe de développement</b>
Ahmed BOUZID	Sayed HAMDI	Rayen BEN HASSINE Fedi ABDOULI



### 4.2 Backlog du produit

Le tableau ci-dessous est le backlog du produit qui contient la liste des fonctionnalités sous forme de user stories toute en précisant la priorité et le temps estimé pour développer chaque fonctionnalité.

C'est le Product Owner qui se charge de sa tenue, de son évolution et de son enrichissement. C'est également à lui de le rendre disponible pour les équipes et les parties prenantes.

**TABLEAU 2.2 :** Backlog du produit

ID	User Story	Priorités	Estimation
1	En tant que visiteur, je souhaite créer un compte afin de s'inscrire à la plateforme.	élevée	24 heures
2	En tant qu'utilisateur, je souhaite m'authentifier à mon compte	élevée.	20 heures
3	En tant qu'utilisateur, je souhaite changer mon mot de passe en cas d'oubli.	moyenne.	8 heures
4	En tant qu'utilisateur je souhaite gérer mon profil afin de paramétrer mes coordonnées selon mes préférences.	faible	12 heures
5	En tant qu'administrateur, je souhaite gérer les comptes des utilisateurs.	moyenne	24 heures
6	En tant que créateur de cours, je peux gérer les super skills.	élevée	32 heures
7	En tant que créateur de cours, je peux gérer les skills.	élevée	24 heures
8	En tant que créateur de cours, je peux gérer les quizzes.	élevée	24 heures
9	En tant que créateur de cours, je peux gérer les check-points.	élevée	20 heures
10	En tant que créateur de cours, je peux gérer les projets finaux.	élevée	20 heures
11	En tant que créateur de cours, je peux gérer les parcours.	élevée	24 heures
12	En tant que créateur de cours, je peux gérer les challenges.	faible	24 heures

13	En tant que créateur de cours, je peux gérer les special quizzes.	faible	16 heures
14	En tant qu'administrateur, je souhaite affecter un apprenant à un parcours	élevée	4 heures
15	En tant qu'administrateur, je souhaite affecter un instructeur à un parcours	élevée	4 heures
16	En tant qu'apprenant, je souhaite consulter la liste des parcours disponibles afin de choisir le parcours qui me convient	élevée	4 heures
17	En tant qu'apprenant, je souhaite consulter la liste des super skills et des skills dans un parcours	élevée	4 heures
18	En tant qu'apprenant, je souhaite m'inscrire à un parcours	élevée	4 heures
19	En tant qu'apprenant, je souhaite consulter la liste des parcours dans lesquelles je suis inscrit	élevée	4 heures
20	En tant qu'apprenant, je souhaite consulter le contenu d'un super skill	élevée	8 heures
21	En tant qu'apprenant, je souhaite consulter le contenu d'un skill	élevée	8 heures
22	En tant qu'apprenant, je souhaite passer un quiz afin d'évaluer les compétences acquises	élevée	8 heures
23	En tant qu'apprenant, je souhaite passer un checkpoint afin d'appliquer l'ensemble des compétences acquises dans un mini projet	élevée	16 heures
24	En tant qu'apprenant, je souhaite réaliser un projet final afin d'appliquer toutes les compétences acquises durant le parcours	élevée	16 heures
25	En tant qu'apprenant, je souhaite gagner des points	moyenne	16 heures

26	En tant qu'apprenant, je souhaite passer des special quizzes	faible	16 heures
27	En tant qu'apprenant, je souhaite participer à un challenge	faible	16 heures
28	En tant qu'instructeur, je souhaite consulter la liste des parcours dans lesquelles je suis affecté	faible	8 heures
29	En tant qu'instructeur, je souhaite consulter le contenu de mon parcours	faible	8 heures
30	En tant qu'instructeur, je souhaite consulter la liste des apprenants dans mon parcours	élevée	8 heures
31	En tant qu'instructeur, je souhaite évaluer les checkpoints, les projets finaux et les challenges effectuées par les apprenants	élevée	16 heures
32	En tant qu'apprenant, je souhaite consulter le feedback et le score attribué par l'instructeur	élevée	4 heures
33	En tant qu'administrateur, je souhaite consulter le nombre d'apprenants, le nombre d'instructeurs et nombre de créateurs de cours dans la plateforme	élevée	2 heures
34	En tant qu'administrateur, je souhaite consulter le pourcentage des apprenants inscrits dans un parcours	élevée	2 heures
35	En tant qu'administrateur, je souhaite consulter le nombre d'apprenants inscrits dans un parcours chaque mois	élevée	2 heures
36	En tant qu'apprenant, je souhaite consulter le pourcentage de mon avancement dans le parcours	élevée	2 heures
37	En tant qu'apprenant, je souhaite consulter le nombre de noeuds accomplis par jour	élevée	2 heures
38	En tant qu'apprenant je souhaite consulter mon classement parmi les apprenants inscrits dans le même parcours	élevée	2 heures

39	En tant qu'apprenant, je souhaite créer une discussion dans un forum	élevée	14 heures
40	En tant qu'apprenant, je souhaite modifier une discussion que j'ai créé	moyenne	1 heures
41	En tant qu'apprenant, je souhaite supprimer une discussion que j'ai créé	moyenne	1 heures
42	En tant qu'apprenant, je souhaite rejoindre une discussion dans un forum	élevée	14 heures
43	En tant qu'apprenant, je souhaite envoyer et recevoir un message dans une discussion	élevée	2 heures
44	En tant qu'instructeur, je souhaite créer une discussion dans un forum	élevée	14 heures
45	En tant qu'instructeur, je souhaite modifier une discussion que j'ai créé	moyenne	1 heures
46	En tant qu'instructeur, je souhaite supprimer une discussion que j'ai créé	moyenne	1 heures
47	En tant qu'instructeur, je souhaite rejoindre une discussion dans un forum	élevée	14 heures
48	En tant qu'instructeur, je souhaite envoyer et recevoir un message dans une discussion	élevée	2 heures
49	En tant qu'administrateur, je souhaite supprimer une discussion publique	faible	1 heures
5	En tant qu'administrateur, je souhaite supprimer un membre dans une discussion publique	faible	1 heures
17	En tant qu'administrateur, je souhaite supprimer un message dans une discussion publique	faible	1 heures

### **4.3 Planification des releases**

« Une release peut être définie comme une période de temps à l'issue de laquelle une version du livrable est proposée. Elle est constituée d'un ensemble de sprints, leur nombre étant dépendant

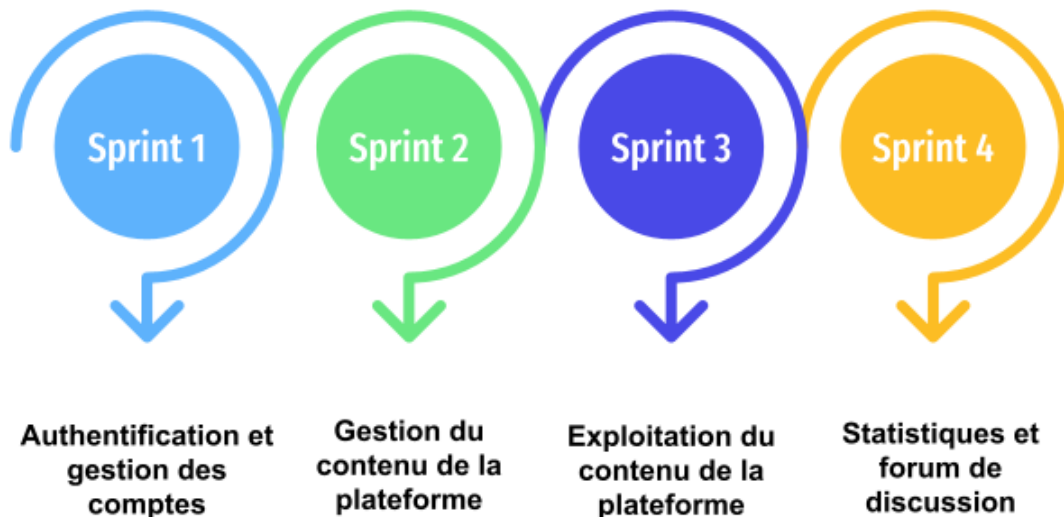
de leur durée et de celle fixée par la release. »[?]

Décomposer le projet en releases nous permet d'avoir à la fin de chaque release un produit livrable et indépendant des autres versions du même projet.

Notre projet peut être décomposé en une seule release puisque les fonctionnalités de notre application sont étroitement liées.

### 4.4 Découpage en sprints

Le découpage du projet en sprints est un élément fondamental de la méthodologie scrum. Pour se faire, nous avons classé nos besoins fonctionnels par priorité et par degré de dépendance. Nous avons décidé enfin de décomposer le projet en quatre sprints comme illustré dans la figure ci-dessous :



**FIGURE 2.6 :** Découpage en sprints

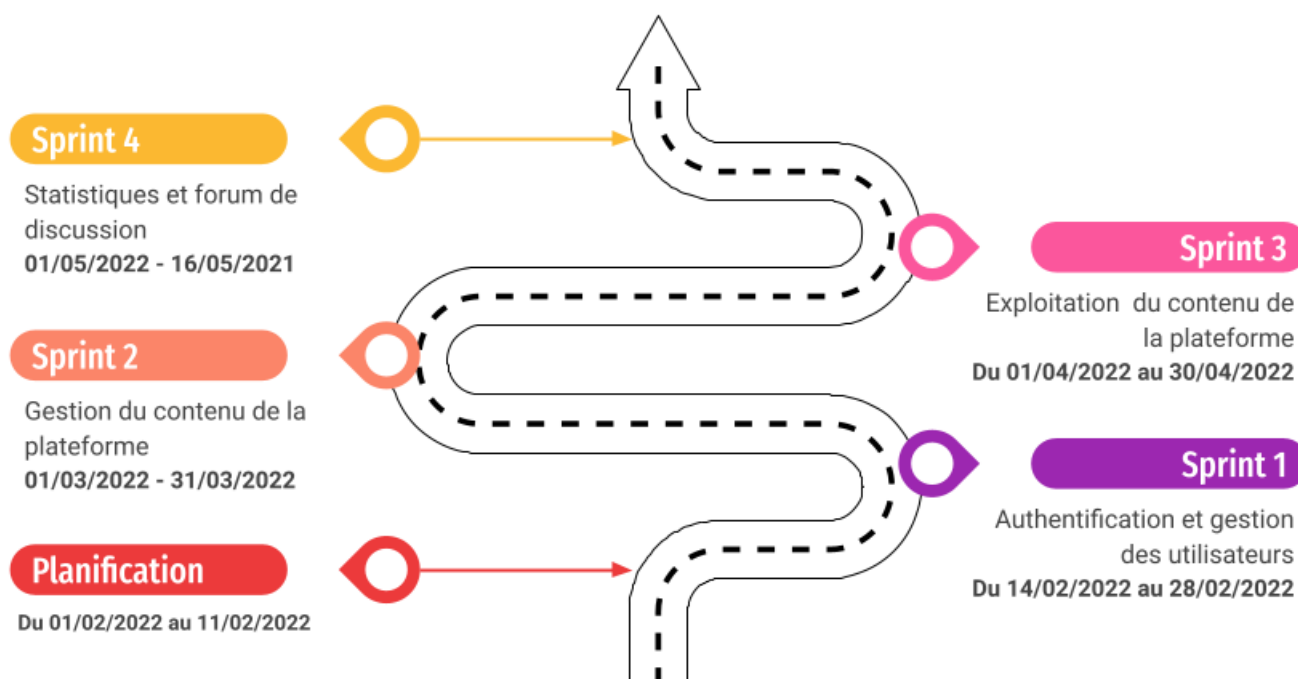
Nous présentons dans la figure suivante les besoins fonctionnels de chaque sprint.



**FIGURE 2.7 :** Besoins fonctionnels de chaque sprint

### 4.5 Planning de réalisation du projet

Afin de bien organiser le travail, nous avons réparti les tâches à faire par le nombre de jours nécessaires comme le montre la figure ci-dessous :



**FIGURE 2.8 :** Planning de réalisation du projet

### Conclusion

Dans ce chapitre, nous avons identifié les besoins fonctionnels et non fonctionnels et nous avons présenté l'application de la méthodologie scrum. Cette phase est très importante vu qu'elle offre une vision plus claire sur le système. Dans les chapitres qui suivent, nous élaborons une analyse détaillée de chaque sprint.

