

République Tunisienne
Ministère de l'Enseignement Supérieur et de la Recherche
Scientifique et de la Technologie de l'Information et de la
Communication.

Ecole Supérieure des Sciences Appliquées et de la
Technologie Privée de Gabès



Projet de fin d'études pour l'obtention du diplôme de licence

Parcours : Systèmes d'information d'entreprises
(BIS).

Titre de projet :

Conception et Réalisation d'une
plateforme E-learning

Réalise par :

Ahmed Guiza - Hachem
moussa

Lieu de stage :

Société laghazala du désert
formations et services

Encadrée par :

Mme. Nadia Ghaibi

DÉDICACE

REMERCIEMENTS

الخلاصة

RÉSUMÉ

ABSTRACT

TABLE DES MATIERES

| | |
|---|----|
| Introduction gÉnÉrale..... | 7 |
| chapitre I : Étude préalable | 8 |
| 1. Introduction | 8 |
| 2. PrÉsEntation de l'organisme d'accueil..... | 8 |
| 3. Mission du stage | 9 |
| 4. Analyse d'existant | 9 |
| 4.1. Étude de l'existant..... | 9 |
| 4.2. PrÉsEntation..... | 9 |
| 4.3. Critique de l'existant..... | 9 |
| 5. Solution proposÉE | 10 |
| 6. MÉthodologie de développement..... | 10 |
| 6.1. Introduction..... | 10 |
| 6.2. Méthode agile | 10 |
| 6.3. MÉthode adoptée | 11 |
| 7. Langage de modÉlisation UML..... | 14 |
| 8. Conclusion..... | 15 |
| chapitre 2 : Analyse et conception gÉnÉrale | 16 |
| 1. Introduction | 16 |
| 2. Spécification des besoins..... | 16 |
| 2.1. Identifications des acteurs..... | 16 |
| 2.2. Besoins fonctionnels | 17 |
| 2.3. Besoins non fonctionnels..... | 18 |
| 3. Pilotage du projet avec Scrum | 18 |
| 3.1. Les rôles Scrum | 18 |
| 3.2. Planification du Backlog..... | 20 |
| 4. Analyse de besoins | 22 |
| 4.1. Diagramme de cas d'utilisation globale..... | 22 |
| 4.2. Architecture logique..... | 23 |
| 4.3. Diagramme de classes entités | 24 |
| 5. Environnement de développement | 26 |
| 6. Conclusion..... | 27 |

| | |
|--|----|
| chapitre 3 : Conception détaillée et réalisation des sprints | 28 |
| 1. Introduction | 28 |
| 2. Sprint 0 Préparation | 28 |
| 2.3. Backlog du sprint | 28 |

TABLE DES FIGURES

| | |
|---|----|
| Figure 1 Logo de la société d'accueil..... | 9 |
| Figure 2 Cycle de vie Scrum | 12 |
| Figure 3 Unified Modeling Language | 14 |
| Figure 4 Equipe scrum | 19 |
| Figure 5 Diagramme de Gantt de notre projet | 20 |
| Figure 6 Diagramme de cas d'utilisation général | 22 |
| Figure 7 Architecture MERN..... | 23 |
| Figure 8 Modèle-Vue-Contrôleur REST API | 24 |
| Figure 9 Diagramme de classe de notre projet | 25 |
| Figure 10 Logo de notre plateforme | 29 |
| Figure 11 Maquette Figma..... | 30 |

INTRODUCTION GÉNÉRALE

Dans un contexte marqué par la digitalisation accélérée des secteurs clés de la société, l'éducation connaît une transformation profonde, portée par l'émergence des technologies de l'information et de la communication (TIC), cette transformation s'est accélérée sous l'effet de crises mondiales, comme la pandémie de COVID-19, révélant autant les potentialités du numérique que les lacunes des systèmes éducatifs traditionnels. Les plateformes d'apprentissage en ligne, ou *E-learning*, s'imposent aujourd'hui comme une réponse aux défis de l'accessibilité, de la flexibilité et de la personnalisation de l'éducation.

Ce projet de fin d'études s'inscrit dans cette dynamique en proposant le développement d'une plateforme *E-learning* innovante, conçue pour répondre aux besoins diversifiés des apprenants.

L'objectif principal de ce travail est de créer un environnement d'apprentissage interactif, intuitif et adaptatif, capable de transcender les limites géographiques et temporelles des méthodes pédagogiques traditionnelles. La plateforme vise à intégrer des fonctionnalités avancées telles que des cours modulaires, des évaluations automatisées, un suivi personnalisé des progrès, et des outils de collaboration en temps réel. Elle s'adresse à un public large, incluant étudiants et professionnels en reconversion.

Le développement de cette solution repose sur une approche méthodologique rigoureuse, alliant conception centrée utilisateur (UX/UI) et adoption de technologies modernes (Front-end avec React.js, Back-end avec Node.js, base de données MongoDB). Une méthodologie agile, centrée sur des cycles itératifs de tests utilisateurs, garantit une adéquation permanente entre les fonctionnalités et les attentes du public cible.

CHAPITRE I : ÉTUDE PRÉALABLE

I. INTRODUCTION

Le présent chapitre a pour objectif de poser les bases du projet et de présenter les éléments fondamentaux qui guideront la suite de la réalisation.

Dans un premier temps, il introduit l'organisme d'accueil, en mettant en avant son domaine d'activité. Ensuite, la mission du stage. Pour bien cerner le contexte, une analyse de l'existant est menée. Cette étude se décompose en une présentation et une critique de l'existant, permettant d'orienter la conception de la solution proposée.

Afin de structurer le développement de cette nouvelle solution, une méthodologie de développement est choisie. Après une brève introduction aux méthodes classiques et aux contraintes qui y sont liées, le chapitre se concentre sur les méthodes agiles et plus précisément sur **Scrum**.

Enfin, le chapitre se conclut par la présentation du langage de modélisation utilisé pour formaliser les différentes composantes du système à développer.

2. PRÉSENTATION DE L'ORGANISME D'ACCUEIL

Société Laghazala du désert formations et services est une société SARL créée en 2021 et à comme activité principale une école de formation professionnelle et une deuxième activité services informatiques basé à Kébili et à Gabes.



Figure 1 Logo de la société d'accueil

3. MISSION DU STAGE

Concevoir et développer une solution logicielle (Web) destinée à la gestion complète des apprenants dans un centre de formation.

4. ANALYSE D'EXISTANT

Avant de commencer un projet informatique, il est crucial d'analyser ce qui existe déjà. Cela implique de comprendre les fonctionnalités actuelles pour déterminer les lacunes et orienter le choix de la meilleure solution pour l'application à développer.

4.1. ÉTUDE DE L'EXISTANT

4.2. PRÉSENTATION

4.3. CRITIQUE DE L'EXISTANT

5. SOLUTION PROPOSÉE

6. MÉTHODOLOGIE DE DEVELOPPEMENT

6.1. INTRODUCTION

Selon des estimations, plus que 80% des projets qui utilisent des méthodologies classiques qui connaissent des retards et des dépassements budgétaires parce que ces méthodologies visent à prédire la façon dont les procédures devraient se passer selon un planning préétabli.

Notre projet suit le principe de développement itératif basé sur l'écoute du client, c'est pour cette raison que notre choix s'est orienté vers les méthodes agiles de développement et de gestion de projet.

6.2. METHODE AGILE

Les méthodes de développement dites « méthodes agiles » (en anglais Agile Modeling, noté AG) visent à réduire le cycle de vie du logiciel et par conséquent accélérer sa réalisation en développant une version minimale, puis en intégrant les fonctionnalités par un processus itératif basé sur une écoute client et des tests tout au long du cycle de développement.

L'origine des méthodes agiles est liée à l'instabilité de l'environnement technologique et au fait que le client est souvent dans l'incapacité de définir ses besoins de manière exhaustive dès le début du projet. Le terme « agile »

fait ainsi référence à la capacité d'adaptation aux changements de contexte et aux modifications de spécifications intervenant pendant le processus de développement.

Grâce aux méthodes agiles, le client est pilote à part entière de son projet et obtient très vite une première mise en production de son logiciel. Ainsi, il est possible d'associer les utilisateurs dès le début du projet.

6.3. MÉTHODE ADOPTÉE

Dans cette partie, nous présentons notre méthode de développement adoptée. Notre choix s'est porté sur la méthode **Scrum** car elle répond aux caractéristiques des méthodes agiles définies dans la section précédente.

Nous avons choisi la méthode Scrum pour les raisons suivantes :

- Elle est utilisée pour développer et tester les courtes itérations,
- Elle permet de produire la plus grande valeur métier dans la durée la plus courte,
- Elle permet l'augmentation de la productivité,
- Elle permet d'adapter le logiciel crée suivant l'évolution du projet

6.3.1. METHODE SCRUM

La méthode Scrum est un cadre de travail agile utilisé pour gérer des projets complexes, surtout dans le domaine du développement logiciel. Son objectif est de livrer un produit de qualité en respectant les besoins changeants des clients. Scrum fonctionne par itérations courtes appelées **sprints**.

Les principaux avantages de Scrum sont :

- Sa flexibilité (adaptation aux changements),

- Sa transparence (progrès visibles via un tableau de tâches ou **Scrum Board**),
- Son accent sur l'amélioration continue.

En divisant le projet en petites étapes, Scrum réduit les risques et garantit une valeur ajoutée régulière. Cette méthode est donc idéale pour les projets où les exigences évoluent fréquemment.

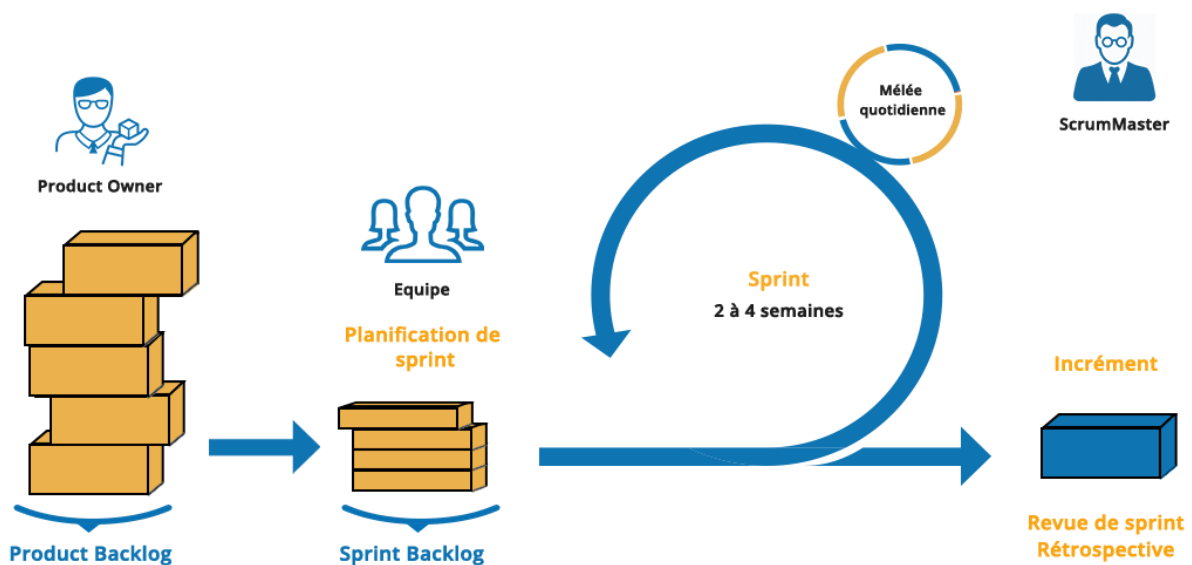


Figure 2 Cycle de vie Scrum

6.3.1.1. ÉQUIPE SCRUM

L'équipe Scrum se compose de trois rôles clés :

- **Product Owner**, qui représente le client du projet et priorise les fonctionnalités.
- **Scrum Master**, qui facilite l'application de la méthode et résout les obstacles et assure également l'organisation des réunions.
- **Équipe de développement**, qui réalise les tâches techniques.

6.3.1.2. BACKLOG DU PRODUIT

Chaque sprint commence par une planification où l'équipe définit les tâches à réaliser, sélectionnées dans une liste priorisée (le Product Backlog).

Les éléments du Backlog produit, appelé aussi des histoires utilisateurs, sont formulés en une ou deux phrases décrivant de manière claire et précise la fonctionnalité désirée par le client.

Généralement, écrit sous la forme suivante : En tant que X, je souhaite Y, afin de Z.

Le Backlog du produit comprend les champs suivants :

- ID : c'est un nombre unique et auto-incrémenté pour chaque histoire.
- Titre : c'est le résumé du user story,
- User Story : c'est une phrase décrivant la fonctionnalité désirée.
- Estimation : c'est l'effort nécessaire à la réalisation d'une histoire utilisateur. Il est estimé par l'équipe et il est fixe dans le temps : un point correspond à quatre heures de charge de travail pour une personne.

6.3.1.3. SPRINT

Le Sprint est une période qui varie entre 2 et 4 semaines au maximum. Au bout de cette période, l'équipe délivre un incrément du produit potentiellement livrable. Une fois la durée est fixée, elle reste constante pendant toute la durée du développement.

6.3.1.4. SCRUM MEETING

Le Scrum Meeting n'est pas une réunion pendant laquelle nous cherchons à résoudre les problèmes, mais uniquement à les identifier et les exprimer.

Le Scrum Master a pour rôle d'apporter des solutions ou de déléguer à un autre membre de l'équipe la résolution des problèmes soulevés durant le Scrum Meeting.

6.3.1.5. REVUE DE SPRINT

À la fin du sprint, tout le monde se réunit pour effectuer la revue du sprint. L'objectif de la revue du sprint est de valider le logiciel qui a été produit pendant le sprint. L'équipe commence par énoncer les items du Backlog de produit qu'elle a réalisés

7. LANGAGE DE MODÉLISATION UML

UML (Unified Modeling Language) est un langage graphique standardisé utilisé pour concevoir, visualiser et documenter des systèmes logiciels. Il propose des diagrammes normalisés (comme les diagrammes de classes, de cas d'utilisation, de séquence ou d'activité) pour représenter la structure, les interactions et les comportements d'un système.

UML facilite la communication entre les acteurs d'un projet et clarifie les exigences techniques dès les phases d'analyse et de conception.



Figure 3 Unified Modeling Language

8. CONCLUSION

Ce chapitre a permis de structurer les fondations du projet en clarifiant son contexte, ses objectifs et ses méthodes. Après avoir présenté l'organisme d'accueil et défini la mission du stage, une analyse critique de l'existant a identifié les lacunes des systèmes actuels, justifiant ainsi la nécessité d'une nouvelle solution.

L'adoption d'une méthodologie agile, spécifiquement Scrum, a été détaillée pour répondre aux défis des projets dynamiques. Cette approche garantit la flexibilité, transparence et livraison progressive de valeur.

Enfin, le choix du langage UML a été justifié pour modéliser visuellement les exigences et les architectures techniques, assurant une communication claire entre les parties prenantes.

En combinant rigueur analytique, agilité méthodologique et outils de modélisation standardisés, ce chapitre établit un cadre solide pour la phase de conception et de développement, qui sera approfondie dans les chapitres suivants.

CHAPITRE 2 : ANALYSE ET CONCEPTION GÉNÉRALE

I. INTRODUCTION

La planification joue un rôle essentiel dans la réalisation d'un projet et sert de fondement à chaque application. Ce chapitre se concentre sur la planification du backlog produit, qui vise à clarifier les besoins à satisfaire de manière précise.

2. SPECIFICATION DES BESOINS

2.1. IDENTIFICATIONS DES ACTEURS

Un acteur est une entité externe au système. Il représente une personne ou un autre système informatique qui attend un ou plusieurs services ouverts par une interface d'accès. Il interagit avec le système par envoi ou réception des messages. Par ailleurs, notre application va intervenir les différents acteurs suivants :

- **Internaute**
 - Il peut inscrire et consulter les formations disponibles.
- **Administrateur**
 - Il possède le droit d'accéder de gérer supprimer des formations.
 - Valider les demandes d'inscription en tant que formateur.
 - Il peut valider les utilisateurs.
- **Formateur**
 - Il prend en charge de gérer les modules, ses vidéos, ses tests.

- **Apprenant**

- Il peut inscrire à une formation consulter le cours et obtient un certificat.

2.2. BESOINS FONCTIONNELS

L'analyse fonctionnelle est une approche qui permet d'identifier et de définir les fonctions qu'un produit doit offrir afin de répondre aux besoins des utilisateurs. Elle vise à prendre en compte les attentes spécifiques de chaque utilisateur vis-à-vis du système conçu.

Ainsi, nous présentons l'ensemble des besoins fonctionnels à implémenter dans notre projet, en les répartissant selon les différents modules de notre application.

Cette application propose de nombreuses fonctionnalités :

- **Gestion des utilisateurs**

- Un internaute peut s'inscrire et consulter les formations disponibles.
 - Un administrateur valide les inscriptions des utilisateurs.
 - Un utilisateur validé peut s'inscrire à une formation.
 - L'inscription à une formation doit être validée par l'administrateur.

- **Gestion des formations**

- L'administrateur peut ajouter une formation.
 - Un formateur peut ajouter du contenu à une formation.
 - Une formation est composée de plusieurs modules.
 - Chaque module contient des vidéos et un test final.
 - Un apprenant reçoit un résultat après avoir terminé un test.

- **Système de messagerie**

- Un apprenant peut envoyer des messages au formateur.

- Un formateur peut répondre aux messages des apprenants.

2.3. BESOINS NON FONCTIONNELS

- **Sécurité et authentification**

- Utilisation de JWT pour l'authentification des utilisateurs.
- Sécurisation des routes selon les rôles (admin, formateur, apprenant).

- **Performance et scalabilité**

- Optimisation des requêtes pour de meilleures performances.
- Mise en cache des données fréquemment consultées.

- **Expérience utilisateur (UX/UI)**

- Interface claire et intuitive pour la navigation.
- Design responsive pour s'adapter aux différents écrans.

- **Disponibilité et fiabilité**

- Gestion des erreurs et affichage de messages clairs en cas de problème.
- Sauvegarde et récupération des données en cas de panne.

- **Extensibilité**

- Possibilité d'ajouter de nouveaux types de contenus (PDF, quiz).
- Système flexible permettant d'ajouter des fonctionnalités futures sans refonte complète.

3. PILOTAGE DU PROJET AVEC SCRUM

3.1. LES ROLES SCRUM

Pour mener à bien un projet en adoptant la méthodologie SCRUM, il est essentiel d'identifier en premier lieu l'équipe impliquée. La composition de notre équipe est illustrée dans la figure ci-dessous.

- ✓ Product owner (PO) : Mr. Nouri Khaireddine, représente le client du projet.
- ✓ Scrum Master (SM) : Mme. Ghaibi Nadia, notre encadrante.
- ✓ Equipe de développement :
 - Guiza Ahmed
 - Moussa Hachem

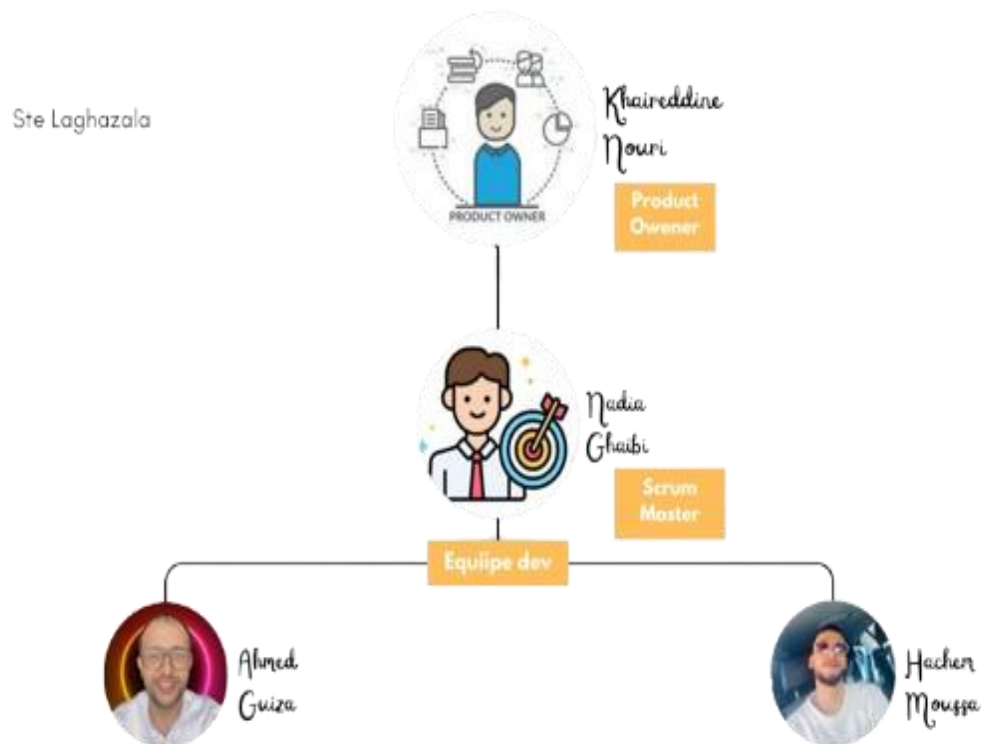


Figure 4 Equipe scrum

3.2. PLANIFICATION DU BACKLOG

La planification du backlog est une étape essentielle de la méthodologie Scrum. Cette section décrit en détail le processus de planification, incluant la création du diagramme de Gantt ainsi que l'organisation des sprints.

3.2.1. DIAGRAMME DE GANTT

Le diagramme de Gantt fournit une vision chronologique du projet, mettant en avant ses différentes phases, les jalons clés et les dépendances entre les tâches. Il constitue un outil visuel de référence pour la répartition du travail et l'optimisation des ressources tout au long du projet.

La figure ci-dessous présente le diagramme de Gantt de notre projet, offrant une représentation claire du calendrier prévu afin de faciliter la gestion du temps et des ressources.



Figure 5 Diagramme de Gantt de notre projet

3.2.2. PLANIFICATION DES SPRINTS

❖ Sprint 0

- Apprentissage des technologies (JavaScript, React, Jira)
- Préparation de l'environnement de développement.
- Conception du logo.
- Préparation des maquettes figma.

❖ Sprint 1

Sprint 1 1 mars – 29 mars (6 tickets)

Authentification, Vérification & Gestion de Base des Cours & gestion de base des inscriptions

| SCRUM | Description | Type | Statut | Assigné |
|----------|--|-----------------|---------|---------|
| SCRUM-6 | As user je peux inscrire avec email et mot de passe et coordonnées | AUTHENTIC... | À FAIRE | |
| SCRUM-7 | As user vérifie de passe et coordonnées | AUTHENTIC... | À FAIRE | |
| SCRUM-8 | As admin vérifier des nouveaux utilisateurs | AUTHENTIC... | À FAIRE | |
| SCRUM-9 | As user je veux connecter | AUTHENTIC... | À FAIRE | |
| SCRUM-10 | As an admin, ajouter la formation. | GESTION DES ... | À FAIRE | |
| SCRUM-11 | As user je veux consulter la liste des formations disponibles | GESTION DES ... | À FAIRE | |

❖ Sprint 2

Sprint 2 31 mars – 28 avr. (5 tickets)

| SCRUM | Description | Type | Statut | Assigné |
|----------|--|-----------------|---------|---------|
| SCRUM-12 | As a Verified_user, je veux demander l'inscription à une fo... | GESTION DES ... | À FAIRE | |
| SCRUM-13 | As an adm l'inscription à une formation des d'inscription | GESTION DES ... | À FAIRE | |
| SCRUM-35 | As instructor je peux ajouter les modules et son contenu à une ... | GESTION DES ... | À FAIRE | |
| SCRUM-43 | As instructor je veux ajouter des tests ou Quiz à un module. | GESTION DES ... | À FAIRE | |
| SCRUM-36 | As user_with_inscriptions je peux visualiser les vidéos et passer l... | GESTION DES ... | À FAIRE | |

❖ Sprint 3

Sprint 3 29 avr. – 20 mai (4 tickets)

| SCRUM | Description | Type | Statut | Assigné |
|----------|---|-----------------|---------|---------|
| SCRUM-37 | As user_with_inscriptions je vois ma progression | GESTION DES ... | À FAIRE | |
| SCRUM-39 | As system je m'assure que l'apprenant a visionné au moins 90% d'u... | GESTION DES ... | À FAIRE | |
| SCRUM-38 | As user_with_inscription je ne peux accéder au module suivant ... | GESTION DES ... | À FAIRE | |
| SCRUM-40 | As user_with_inscriptions je reçois un certificat téléchargeable l... | GESTION DES ... | À FAIRE | |

4. ANALYSE DE BESOINS

4.1. DIAGRAMME DE CAS D'UTILISATION GLOBALE

Nous donnons dans la figure ci-dessous une vue globale concernant le comportement fonctionnel du système. Ce diagramme permet de représenter les interactions entre les acteurs et les cas d'utilisation du système



Figure 6 Diagramme de cas d'utilisation général

4.2 ARCHITECTURE LOGIQUE

L'architecture MERN suit généralement une approche RESTful API, où le backend (Node.js et Express.js) expose des endpoints permettant au frontend (React) d'interagir avec la base de données MongoDB.

Elle peut également être structurée selon le modèle MVC (Model-View-Controller), en séparant la logique métier (Modèle), la gestion des requêtes (Contrôleur) et l'affichage (Vue).

MongoDB sert de base de données NoSQL pour stocker les informations. Express.js gère les requêtes et les réponses côté serveur. Node.js assure l'exécution du code backend en environnement JavaScript. Enfin, React prend en charge l'interface.

Cette organisation facilite la maintenabilité, la scalabilité et la communication entre le frontend et le backend.

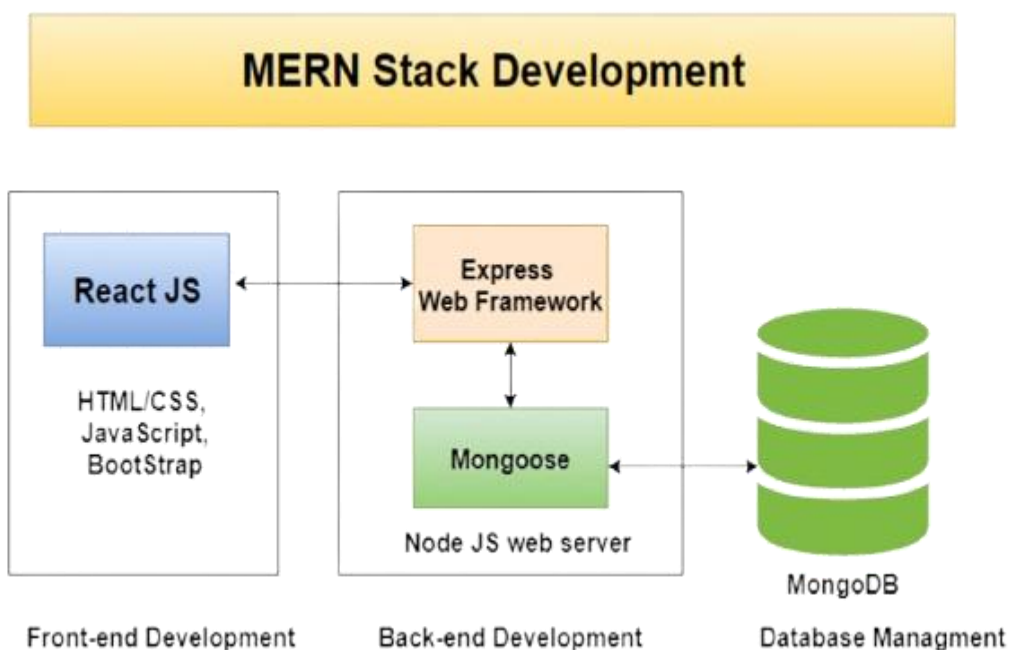


Figure 7 Architecture MERN

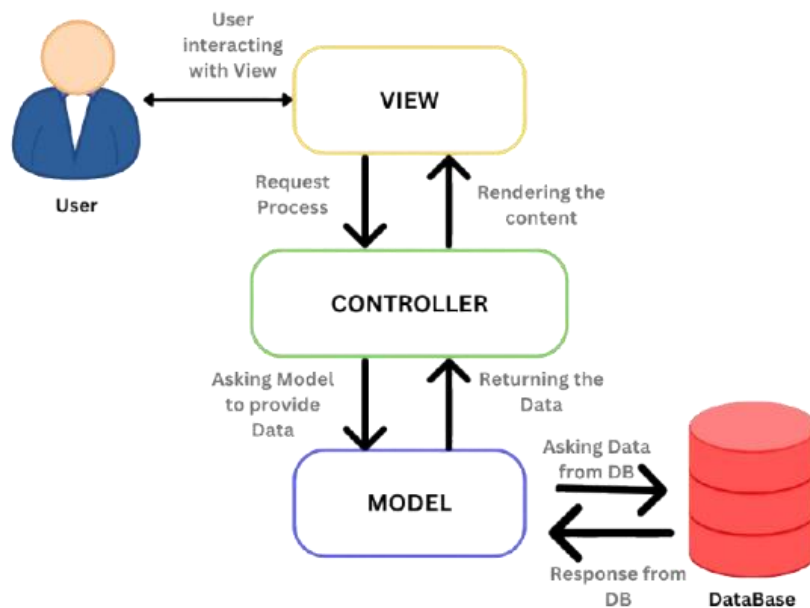


Figure 8 Modèle-Vue-Contrôleur REST API

4.3. DIAGRAMME DE CLASSES ENTITES

Passons maintenant à présenter le diagramme de classes entités de notre application, qui offre un aperçu détaillé de la structure de nos données et des interactions entre les entités

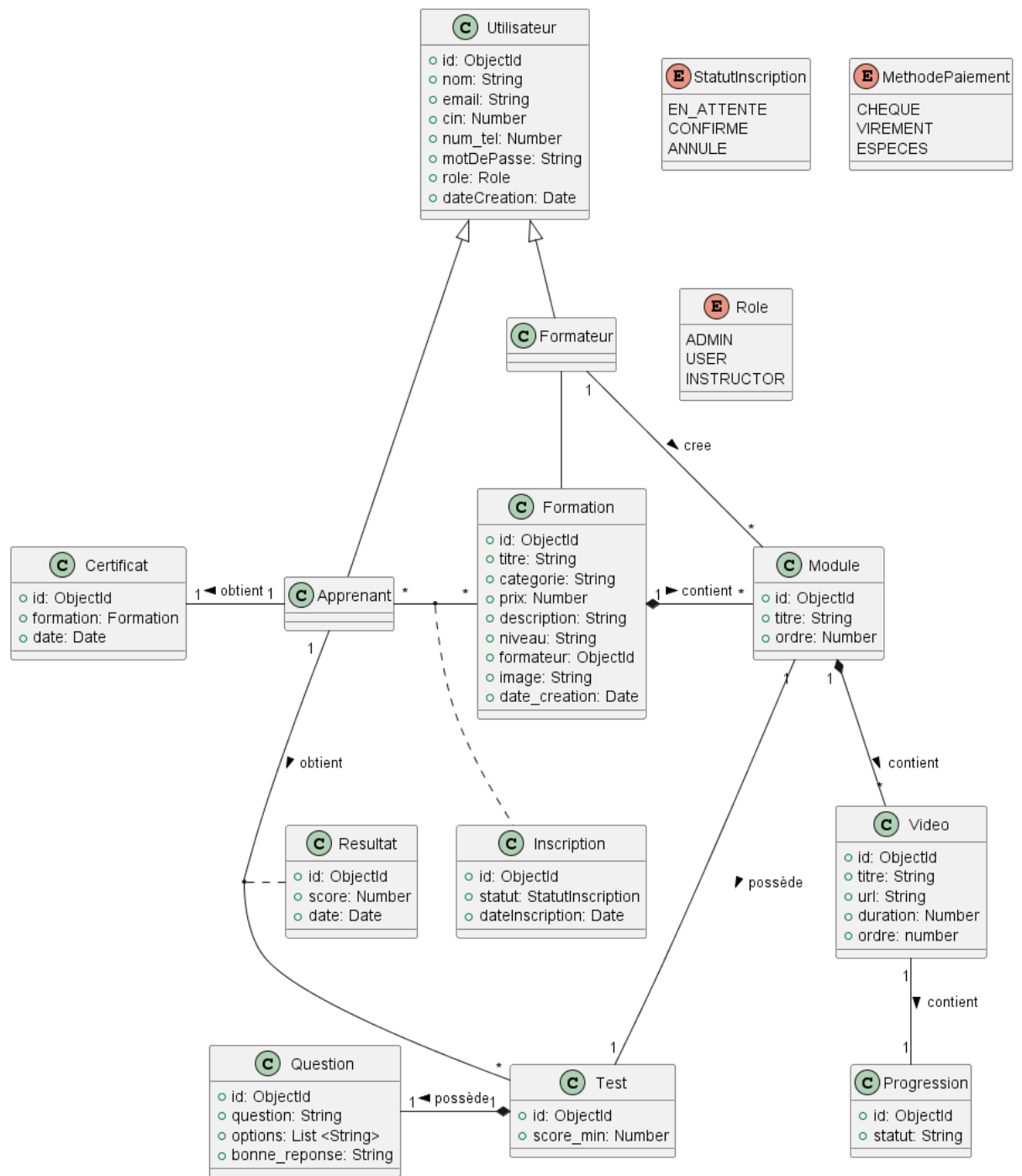


Figure 9 Diagramme de classe de notre projet

5. ENVIRONNEMENT DE DEVELOPPEMENT

Avant de commencer l'implémentation de notre projet, nous avons configuré notre environnement de développement en réunissant l'ensemble des outils logiciels, langages et frameworks nécessaires à sa réalisation.



Éditeur de code polyvalent de Microsoft, avec débogage intégré, extensions personnalisables et prise en charge de multiples langages.



Outil de test et documentation d'API, permettant d'envoyer des requêtes HTTP, automatiser des tests et simuler des endpoints.



Outil de design collaboratif pour créer des interfaces utilisateur, des prototypes et des systèmes de design. Accessible via navigateur et en temps réel.



Système de contrôle de version décentralisé pour suivre les modifications de code. Permet le travail en parallèle via des branches et fusion des changements



Plateforme d'hébergement de projets Git avec gestion de collaboration (issues, pull requests). Utilisé pour l'open source et le CI/CD.



Environnement d'exécution JavaScript côté serveur basé sur le moteur V8. Permet de construire des applications réseau scalables et asynchrones.



Framework web minimaliste pour Node.js, simplifiant la création d'APIs REST et d'applications avec routage et middleware.



Outil pour générer des diagrammes UML (classes, séquences) à partir de texte. Intégrable dans des docs, wikis ou IDE pour la modélisation visuelle.



Langage de script côté client et serveur (via Node.js) pour créer des sites interactifs. Essentiel pour le développement web moderne avec HTML/CSS.



Logiciel libre et gratuit de retouche photo et de création graphique, offrant des outils avancés (calques, filtres). Alternative puissante à Photoshop pour l'édition d'images et la création artistique.

6. CONCLUSION

CHAPITRE 3 : CONCEPTION DETAILLÉE ET REALISATION DES SPRINTS

I. INTRODUCTION

Dans ce chapitre nous présentons l'articulation entre la phase de conception technique et la mise en œuvre agile via des sprints. Ce chapitre détaille les itérations de développement, les décisions clés et les livrables intermédiaires validant les fonctionnalités prioritaires du projet.

2. SPRINT 0 PREPARATION

L'objectif de ce sprint est de de préparer l'environnement de travail et d'étudier la solution ainsi que les technologies à utiliser.

2.3. BACKLOG DU SPRINT

Objectif : Acquérir les compétences nécessaires et configurer l'environnement de travail pour démarrer le développement.

- ❖ Apprentissage des technologies (JavaScript, React, Jira) :
 - **JavaScript :**
 - Maîtrise des concepts de base (ES6+, manipulation du DOM, Promesses, async/await).
 - Compréhension des frameworks frontend (React) et des outils de gestion de packages (npm).
 - **React :**

- Apprentissage des composants fonctionnels, hooks (useState, useEffect), routage (React Router).
- Création d'interfaces dynamiques avec JSX et intégration d'API.
- Apprendre à utiliser redux pour la gestion de l'état.
- **Jira :**
 - Prise en main de la plateforme pour la gestion agile (création de tâches, suivi des sprints, tableaux Kanban).
 - Rédaction des user stories et estimation des points d'effort.
- ❖ Préparation de l'environnement de développement.
 - **Outils installés :**
 - VS Code : Configuration des extensions (ESLint, Prettier, React Snippets).
 - Node.js : Installation de la LTS (v18.x) et vérification avec node -v et npm -v.
 - Git : Initialisation du dépôt local et liaison avec GitHub.
 - Postman : Création d'un workspace pour tester les API.
 - MongoDB : Configuration d'un cluster Atlas et connexion à l'application via Mongoose.
 - **Structure du projet :**
 - Initialisation du projet React.
 - Configuration du backend Node.js (dossiers server, models, routes).
- ❖ Conception du logo.
 - Gimp : Conception du logo.



Figure 10 Logo de notre plateforme

❖ Préparation des maquettes figma.

- Figma : Conception des maquettes d'interface.

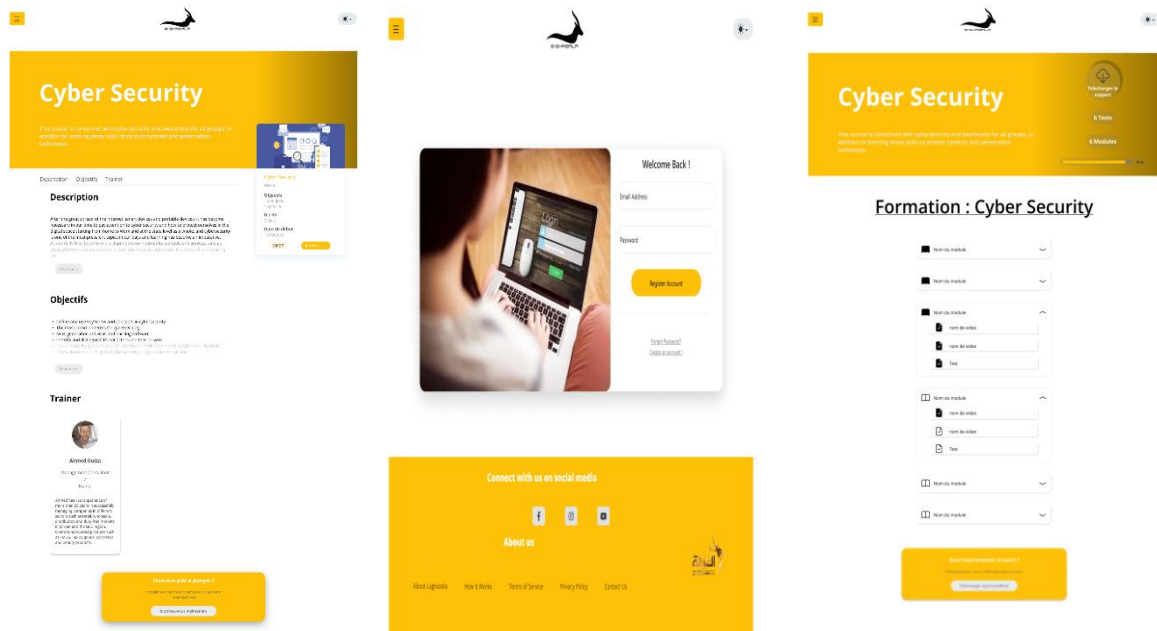


Figure 1 | Maquette Figma