# REALISATION DU PROJET

* 1. Introduction

La phase de réalisation est la partie essentielle durant le développement de notre projet informatique. De ce fait, nous allons présenter les outils et les langages de programmation utilisés pour le développement de ce projet. Ainsi qu’une présentation de la plateforme via ces interfaces graphiques.

* 1. Langages et framework de développement
     1. Les langages de programmations utilisées
        1. Langage de balise HyperText Markup Language : HTML

L’HTML signifie « HyperText Markup Language » qu’on peut traduire par « Langage de balises pour l’hypertexte ». Il est utilisé afin de créer et de représenter le contenu d’une page web et sa structure. D’autres technologies sont utilisées avec HTML pour décrire la présentation d’une page et/ou ses fonctionnalités interactives. [3.01]

L’ « hypertexte » désigne les liens qui relient les pages web entre elles, que ce soit au sein d’un même site web ou entre différents sites web. Les liens sont un aspect fondamental du web. Ce eux qui forment cette « toile », mot traduit par web en anglais. En téléchargeant du contenu sur l’Internet et en les reliant à des pages créées par d’autres personnes. [3.01]

Le HTML a été inventer par Tim Berners-Lee en 1990 alors qu’il travaillait au CERN de Genève. Son objectif principal était de relise des pages entre elles via des liens hypertexte et de rendre les documents lisibles sur tous les ordinateurs indépendamment de leur système d’exploitation. Depuis le HTML a évolué à travers plusieurs versions : HTML en 1991, HTML+, HTML 2, HTML3.2, HTML 4.01 en 1999 et XHTMLen 2000.

En 2004, le Web Hypertexte Application Technologie working Group (WebWG) a été créé pour relancer le développement du HTML, conduisant à la spécification de HTML 5 en 2007. HTML 5 a été officialiser en aout 2012 et continue de s’améliorer. Cette version met l’accent sur la compatibilité avec les contenus existants et l’adaptation aux besoins des utilisateurs grâce à une approche pragmatique des évolutions technologiques [3.02]

* + - 1. Cascading Style Sheet : CSS

Le Cascading Style Sheets ou le CSS, est un langage de feuille de style utilise pour décrire la présentation d’un document écrit en HTML ou XML. Il permet de séparer le contenu HTML de la présentation, en contrôlant l’apparence des éléments, la mise en page, les polices, les couleurs, etc., sur différents supports comme les écrans, le papier ou la voix

CSS est l’un des langages principaux du Web ouvert et a été standardisé par le World Wide Web Consortium (W3C). Auparavant, le développement des différentes parties de la spécification CSS était réaliser de façon synchrone, permettant d’avoir une version pour l’ensemble de la recommandation. Nous avons donc entendu parler de CSS1, CSS2.1, voire CSS3. Toutefois, il n’y aura pas de version CSS4 ou d’autres version globale numérotée de CSS. [3.03]

* + - 1. Langage de programmation : JavaScript

Le JavaScript souvent abréger en « JS » est un langage de script léger, orienté objet, principalement connu comme le langage de script des pages web. Mais il est aussi utilisé dans de nombreux environnements extérieurs aux navigateurs web tels que Node.JS, Apache CouchDB, voire Adobe Acrobat. Le code JavaScript est interprété ou compilé à la volée. C’est un langage à objets utilisant le concept de prototype, disposant d’un typage faible et dynamique qui permet de programmer suivant plusieurs paradigmes de programmation : fonctionnelle, impérative et orienté objet. [3.04]

Le JavaScript est un langage de programmation informatique open source créée en 1995 par Brendan Eich, un programmeur travaillant pour Netscape Communication Corporation. Crées-en seulement 10 jours, le JavaScript est appelé à l’origine « Mocha », mais a ensuite été renommé « LiveScripté » avant de se fixer finalement sur JavaScript, dans une démarche stratégique visant à surfer sur la popularité de Java à l’époque. [3.05]

Avec HTML et CSS, le JavaScript représente en fait l’un des trois principaux langages de programmation utilisé dans le développement web. [3.05]

1. JavaScript moderne à partir de 2015

Le JavaScript continue d’évoluer grâce à des mise à jour régulières de la spécifications ECAMScript. Les fonctionnalités introduites dans les version suivantes (ES2016, ES2017, etc…) comprennent async/await, de nouvelles méthodes et une syntaxe améliorée. Ces dernières années, avec le développement de cadres et de bibliothèques puissants pour les développements frontaux et dorsaux, l’écosystème JavaScript s’est considérablement développé. [3.05]

1. L’utilisation de JavaScript de nos jours

De nos jours, JavaScript est un langage de programmation d’une impressionnante polyvalence qui peut être déployé dans divers aspects du développement web. Sa flexibilité et son large éventail de cadres le rendent idéal pour les types de développement web suivants [3.05] :

* Développement de l’interface utilisateur (UI) : dans le cadre du développement web coté client, JavaScript est essentiel pour créer des interfaces utilisateur dynamiques et interactives. Il est utilisé pour manipuler le Document Object Model ou DOM et répondre aux actions de l’utilisateur, afin d’offrir une expérience transparente et attrayante.
* Développement web coté serveur : avec l’introductions de Node JS, JavaScript peut être utiliser cote serveur, ce qui permet aux développeurs de créer des applications serveur évolutives et performantes. Cela permet d’utiliser un langage unifié pour le développement coté client et cote serveur.
* Développement complet : JavaScript est un composant clé des piles MEAN qui signifie MongoDB, Express.JS, Angular.JS, Node.JS et MERN, MongoDB, Express.JS, React.JS, Node.JS, qui sont des choix populaires pour le développement à pile complète. Ces piles permettent aux développeurs d’utiliser JavaScript dans l’ensemble de l’application, de la base de données au frontend.
* Développement d’application mobile : dans des cadres comme React Native et NativeScript, JavaScript est utiliser pour créer des applications mobiles multiplateformes. Cela permet aux développeurs d’écrire du code une seule fois et de le déployer sur les plateformes IOS et Android.
* Applications à page unique (SPA) : JavaScript est un langage de base pour la construction de SPA, ou une seule page HTML est dynamiquement mise à jour lorsque l’utilisateur interagit avec l’application. Des framework comme Angular.JS, React.JS et Vue.JS simplifient le développement des SPA en fournissant des architectures structurées.
* API Web et intégration : couramment utilisé pour interagir avec des API externes, JavaScript permet l’intégration de services et de données tiers dans les applications web. Cela est essentiel pour créer des applications riches en fonctionnalité et axées sur les données.
* Développement de jeux sur le web : pour créer des jeux sur le web, JavaScript peut être déployé en combinaison avec HTML 5 et CSS3. Des bibliothèques comme Phaser et Three.js facilitent la création de jeux interactifs et visuellement attrayantes directement dans le navigateur.
  + 1. Les frameworks utilisées
       1. Framework Vue JS

Le framework Vue.JS est un framework évolutif pour construire des interfaces utilisateur. A la différence des autre framework monolithiques, Vue a été conçu et pense pour pouvoir être adopté de manière incrémentale. Le cœur de la bibliothèque se concentre uniquement sur la partie vue, et il est vraiment simple de l’intégrer avec d’autres bibliothèque ou projets existants. D’un autre côté, Vue est tout à fait capable de faire tourner des applications web mono-pages quand il est couplé avec des outils modernes et des bibliothèques complémentaires. [3.06]

1. Historique

Vue JS a été créé à l’origine par une seule personne, Evann You, ancien ingénieur de Google ayant notamment travaillé avec Angular.JS puis sur le framework Meteor. Après avoir expérimenté une mécanique de réactivité qu’il jugeait plus intéressante que celle d’Angular, il décide de publier ses premiers résultats en juillet 2013. Cinq ans plus tard, Vue.JS est le 3ème projet sur GitHub en nombre de stars et Evan enchaine les conférences à travers le monde. [3.07]

1. Equipe de développement

Vue JS dispose aujourd’hui d’une équipe internationale d’une trentaine de personnes, constituée de contributeurs bénévoles qui se sont formé avec les années. La décentralisation complétée de l’équipe est à la fois une contrainte et une force qui lui a permis de diffuser le framework beaucoup plus rapidement à plusieurs endroits à la fois. [3.07].

1. Comparaison avec les autres frameworks

Tâchons de comparer Vue avec les deux autres framework JS les plus populaires, React et Angular. [3.07].

Premièrement, les points communs entre ces 3 frameworks :

* Très populaire, utilisés par de grosse entreprise
* Mature, stables, support à long terme financé
* Codebase orienté composants
* Paradigme principalement déclaratif et non impératif
* Adapté aux stacks modernes avec ES6+/ TypeScript
* Large écosystème de composants et d’outillage

Deuxièmement, le positionnement : React, Vue et Angular se positionnent dans des catégories différentes et cela s’observe dès les premières lignes d’introduction. React se définit comme une bibliothèque indépendante du reste de la stack technique, Angular se définit comme le framework unique répondant à tous les besoins, desktop comme mobile. Tandis que Vue se positionne entre deux comme un framework progressif, polyvalent et que l’on peut adopter par étapes. Un juste milieu entre une bibliothèque et un framework tout équipé.

Enfin une comparaison de langage, style et stack technique

Comparaison entre divers framework

|  |  |  |  |
| --- | --- | --- | --- |
|  | React | Vue.js | Angular |
| Langage | JSX, (TypeScript) | TypeScript, JSX | TypeScript |
| Gestion d’état | State centraliser non mutable (flux) | State centraliser en option (Vuex), sinon interne et mutable | State interne et mutable, pas de state centraliser officiel |
| Gestion de la réactivité | Manuelle (setState + VDOM diffing) | Automatique (observers/Proxies) | Automatique (Zones/dirty-checking) |
| Stack | Ne se suffit pas à lui-même, à intégrer dans une stack web avec d’autres outils en externe. Ecosystème riche | Quelques outils officiels maintenus par la team Vue, d’autres à chercher en externe. Promeut des solutions tierces si elles sont appropriées | Framework full stack et auto suffisant, avec options de sécurité incluses nativement. Complet mais plus fermé. |

State : objet de données représentant l’état de l’application ;

Interne : propre à chaque composant

* + - 1. Express.JS

Express JS, parfois aussi appelé « Express », est un framework backend Node.JS minimaliste, rapide et de type Sinatra qui offre des fonctionnalités et des outils robustes pour développer des applications backend évolutives. Il vous offre le système de routage et des fonctionnalités simplifiées pour étendre le framework en développant des composants et des parties plus puissants en fonctions des cas d’utilisations de notre application. [3.08]

C’est de ce fait le framework standard pour le développement de serveur en Node.JS. L’auteur original, TJ Holowaychuck, le décrit comme étant un serveur inspiré de Sinatra dans le sens qu’il est relativement minimaliste tout en permettant d’étendre ses fonctionnalités via des plugins.

Express fournit un ensemble d’outils pour les applications web, les requêtes et les réponses http, le routage et les intergiciels permettant de créer et de déployer des applications à grande échelle, prêtes pour l’entreprise. Il fournit également un outil d’interface de ligne de commande appelé Node Package Manager ou NPM, où l’on peut s’approvisionner en paquets développés. [3.08]

Notamment utilisé dans un large éventail de choses dans l’écosystème JavaScript/Node.JS, nous pouvons également développer des applications, des points de terminaison API, des systèmes de routage et des frameworks avec lui.

Voici quelques-uns des types d’applications créer avec Express JS :

* Applications à page unique
* Outils de collaboration en temps réel
* Applications de streaming
* Application Fintech
  + - 1. Socket.IO

Le Socket.IO est une bibliothèque JavaScript open source qui fournit une solution pour la communication en temps réel entre un client et un serveur dans les applications web modernes. Créée par Guillermo Rauch en 2010, elle est développée en continu par une communauté de contributeurs.

Couramment utilisée dans les applications web qui nécessitent une communication en temps réel, telles que les applications de messagerie instantanée, les jeux en ligne et les tableaux de bord en temps réel. Socket.IO fonctionne avec des technologies web telles que Node.JS et les WebSockets pour offrir une communication bidirectionnelle efficace entre un client et un serveur.

Contrairement au WebSockets, Socket.IO n’est pas une spécification officielle, elle a donc sa propre API. Toutefois, elle prend en charge les WebSockets et d’autres protocoles de communication en temps réel tels que Server-Sent Event ou SSE et Long Polling, selon la disponibilité du navigateur et du serveur. [3.09]

Le Socket.IO offre plusieurs avantages par rapport au WebSockets :

* Socket.IO diffuse un message à tous les clients connectés.
* Les proxys et les répartiteurs de charge rendent la mise en œuvre et l’échelle des WebSockets difficiles. Socket.IO prend en charge ces technologies en standard.
* Comme mentionné précédemment, Socket.IO peut passer à d’autre technologie que WebSockets lorsque le client ne le prend pas en charge
* Lors des interruptions de connexion, WebSockets ne se reconnectera pas automatiquement tandis que Socket.IO le fait grâce à un mécanisme de ping/pong, qui vérifie périodiquement l’état de la connexion.
  + - 1. Tailwind CSS

Tailwind CSS est un framework utility-First CSS avec des classes prédéfinis que nous pouvons utiliser pour construire et concevoir des pages web directement dans votre balisage. Il nous permet d’écrire du CSS dans notre HTML sous la forme de classes prédéfinies. [3.10]

Créer en 2017 par Adam Wathan, un homme passionné par le développement web et est à l’origine de nombreux projet tels que Laravel Valet, Jigsaw et bien sûr Tailwind CSS. C’est en travaillant sur un projet nommé KiteTail que lui vient l’idée de créer le framework Tailwind CSS. Avec ce framework, il est possible de créer un design d’interface au sein même du fichier HTML. Cette façon de programmer n’interfère pas avec les pratiques recommandées par le W3C comme celle de séparer le HTML des feuilles de styles CSS. Pour utiliser Tailwind, il faut prendre les classes CSS prédéfinies par le framework en les appelant dans un fichier HTML comme ceci : [3.11]

* 1. Logiciels de développement
     1. Editeur de code Visual Studio Code

Visual Studio Code est un éditeur de code source léger, mais puissant, qui s’exécute sur notre bureau et est disponible pour Windows, MacOs et Linux. Il est livré avec la prise en charge intégrée pour JavaScript, TypeScript et Node.JS et dispose d’un riche écosystème d’extensions pour d’autres langages et environnements d’exécution tels que C++, C#, Java, Python, PHP, Go, .Net [3.12].

Il offre de nombreuses fonctionnalités à l’utilisateur comme la coloration syntaxique, l’auto-complétions, la mise en évidence des erreurs, la navigation de code, le débogage, la gestion de versions, l’intégration avec Git. Grâce à l’aide d’une grande variété d’extensions développées par la communauté, l’éditeur permet aux développeurs de personnaliser ce dernier selon leurs besoins.

* + 1. Visual Paradigm

Visual Paradigm est un outil complet qui intègre de nombreuses fonctionnalités et modules essentiels pour la modélisation et le développement de logiciels. Voici quelques-unes des principales caractéristiques [3.13] :

* Modélisation UML : Visual paradigm prend en charge tous les diagrammes UML standard, y compris les diagrammes de cas d’utilisation, de classes, de séquence, d’activité et d’état.
* Modélisation de processus métier : Grace à la notation BPMN, nous pouvons modéliser, documenter et optimiser des processus métier
* Conception de base de données : Créer des modèles de donnée conceptuels logiques et physiques à l’aide de diagrammes entité-relation ou ERD.
* Architecture d’entreprise : utiliser ArchiMate pour modéliser et documenter l’architecture d’entreprise, y compris les aspects métier, applicatifs et technologiques
* Gestion des exigences : capturer, organiser et gérer les exigences de nos projets de manière centralisée.
* Génération de code : générer automatiquement du code source à partir de nos modèles UML, économisant ainsi du temps et des efforts.
* Développement agile : panifier et suivit de nos sprints, créez des backlog de produits et de sprints, et gérer les taches de manière efficace.

Visual paradigm se distingue par sa capacité à s’intégrer à d’autres outils et plateformes populaires, tels que Git, Jira, Confluence et Microsoft Office. Cette interopérabilité facilite la collaboration au sein des équipes et améliore l’efficacité globale du processus de développement [3.13].

* + 1. Plateforme GitHub

GitHubest un plateforme web et un service de cloud qui aide les développeurs à stocker et à gérer leur code, ainsi qu’à suivre et contrôler les modifications qui lui sont apportée. Créer en 2008 par Tom Preston-Werner, Chris Wanstrath et PJ Hyett. Preston-Werner a expliqué que l’idée de créer GitHub était venue de l’utilisation de Git. A l’époque, il n’y avait pas de plateforme en ligne permettant de stocker et de partager du code avec d’autre développeurs. [3.15]

GitHub se base sur deux principes : [3.14]

* Contrôle de version : le contrôle de version aide les développeurs à suivre et à gérer les modifications apportées au code d’un projet logiciel. Au fur et à mesure qu’un projet prend de l’ampleur, le contrôle de version devient essentiel.
* Git : c’est un système de contrôle de version open-source spécifique crée par Linus Torvalds en 2005. Concrètement, git est un système de contrôle de version distribué, ce qui signifie que l’ensemble de la base du code et de l’historique est disponible sur l’ordinateur de chaque développeur. Ce qui permet des branchements et une fusion facile.
  + 1. Plateforme Node.JS

Node JS est une plateforme de développement JavaScript. Ce n’est pas un serveur, ce n’est pas un framework, c’est juste le langage JavaScript avec des bibliothèques permettant de réaliser des actions comme écrire sur la sortie standard, ouvrir/fermer des connections réseau ou encore créer un fichier. [3.16]

On peut souvent le confondre avec un serveur car c’est son origine : Node.JS a été créé par Ryan Dahl dans le but de pouvoir créer des applications temps réel ou le serveur est capable de pousser de l’information au client. C’est dans ce but qu’il utilise la bibliothèque libuv pour réaliser son modèle d’entrer sortie non bloquante. [3.16]

Node.JS présente de nombreux intérêts :

* Logiciel libre sous la licence MIT
* Performance du moteur V8
* Modèle non bloquant
* Communauté très active
* Système de paquet intégrer : NPM

Grâce à une vaste liste de fonctionnalités, Node.JS a connu une croissance rapide au cours de ces dernière année : [3.17]

* Facile : Easy-Node.JS est assez facile à prendre en main. C’est un choix incontournable pour les débutants en développement web.
* Evolutif : il offre une grande évolutivité aux applications. Comme Node.JS est un single-thread, il est capable de gérer un grand nombre de connexions simultanées avec un débit élevé.
* Vitesse : l’exécution non bloquante des threads rend Node.JS encore plus rapide et plus efficace
* Paquets : un vaste ensemble de paquets Node.JS open source est disponible et peut simplifier notre travail
* Backend solide : Node.JS est écrit en C et C++, ce qui le rend rapide et ajoute des fonctionnalités comme le support réseau.
* Multi-plateforme : la prise en charge multi-plateforme nous permet de créer des sites web SaaS, des applications de bureau et même des applications mobiles, le tout en utilisant Node.JS
* Maintenable : Node.JS est un choix facile pour les développeurs, car le frontend et le backend peuvent être gères avec JavaScript comme un seul langage.
  + 1. Logiciel X(cross) Apache MariaDB Perl PHP : Xampp

Le Logiciel Xampp est un logiciel de développement web permettant de mettre en place un serveur web local, un serveur FTP et un serveur de messagerie électronique qui fonctionne sous tous types de systèmes d’exploitation comme Windows, Linux et Mac. Il contient du MySQL que nous allons utiliser comme base de données pour notre plateforme.

* 1. Base de données : MySQL

Pour bien réaliser notre plateforme, nous avons besoin d’une base de données que nous utiliserons en tandem avec les autre frameworks et logiciels précédemment mentionnés. Cette base nous servira aussi à stocker les informations qui seront communiquer aux utilisateurs de notre plateforme. La base de données que nous utiliserons sera le MySQL.

MySQL est un système de gestion de base de données relationnelles SQL ou Structured Query Language open source développé et supporté par Oracle. [3.18]

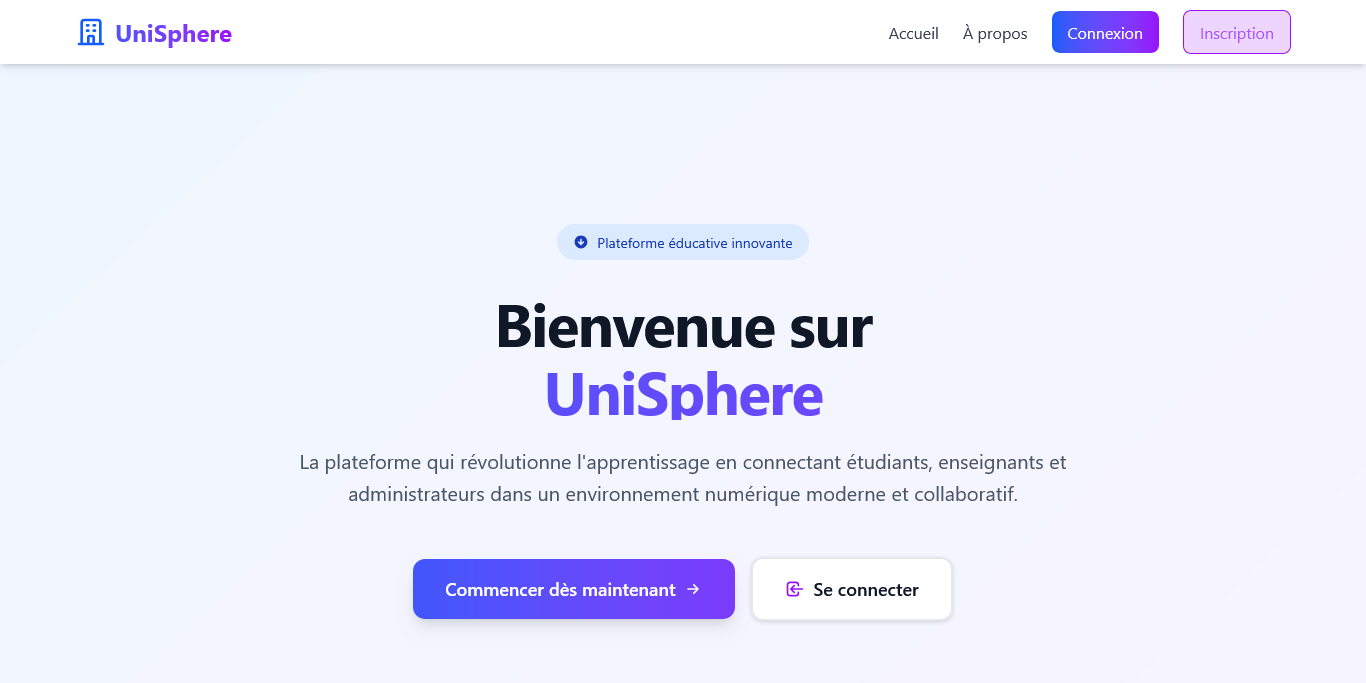
MySQL est un SGBDR open source qui utilise SQL pour créer et gérer des bases de données. En tant que base de données relationnelle, MySQL stocke les données dans des tables de lignes et de colonnes organisées en schémas. Un schéma définit l’organisation et le stockage des données et décrit la relation entre les différentes tables. Avec ce format, les développeurs peuvent facilement stocker, extraire et analyser de nombreux types de données, y compris du texte simple, des chiffres, des dates, des heures et plus récemment, des vecteurs et JSON [3.19]

Nous avons choisi d’utiliser MySQL pour diverses raisons :

* La rapidité : le serveur MySQL est très rapide
* La facilité d’utilisation : MySQL est beaucoup plus simple à utiliser que la plupart des serveurs de bases de données commerciaux ;
* API diverse : on peut effectuer diverse opération sur une base MySQL en utilisant des interfaces écrites en C, Perl, C++, Java, Python, PHP ;
* Les connexion et sécurité : MySQL dispose d’un système de sécurité permettant de gérer les personnes et les machines pouvant accéder aux différentes bases.
  1. Interfaces graphiques

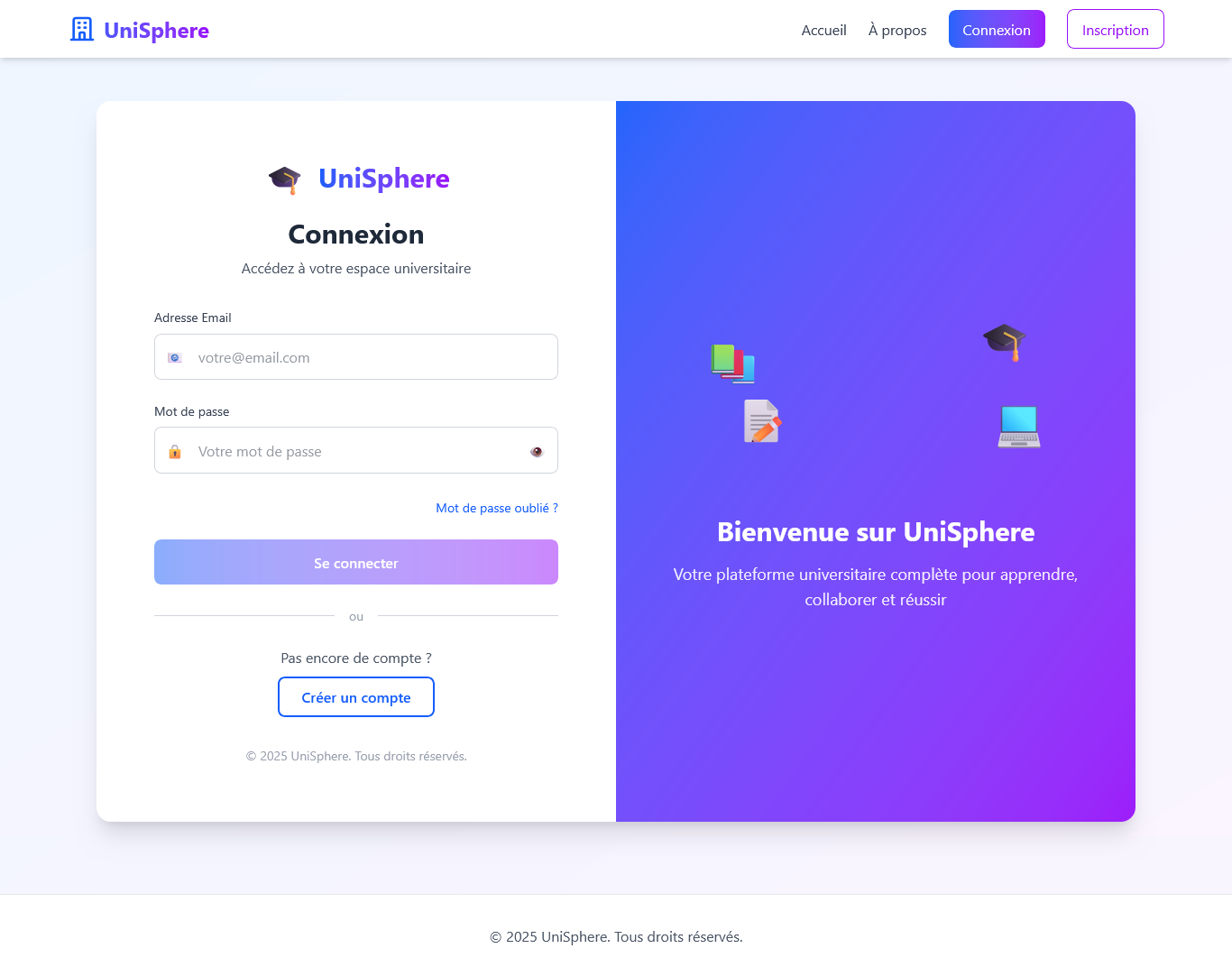
Dans cette partie du chapitre, nous allons présenter quelques illustrations de notre plateforme :

* + 1. Page d’accueil d’UniSphere



Page d’accueil de l’UniSphere

* + 1. Pages d’authentification



Page d’authentification

* + 1. Tableau de bord étudiant

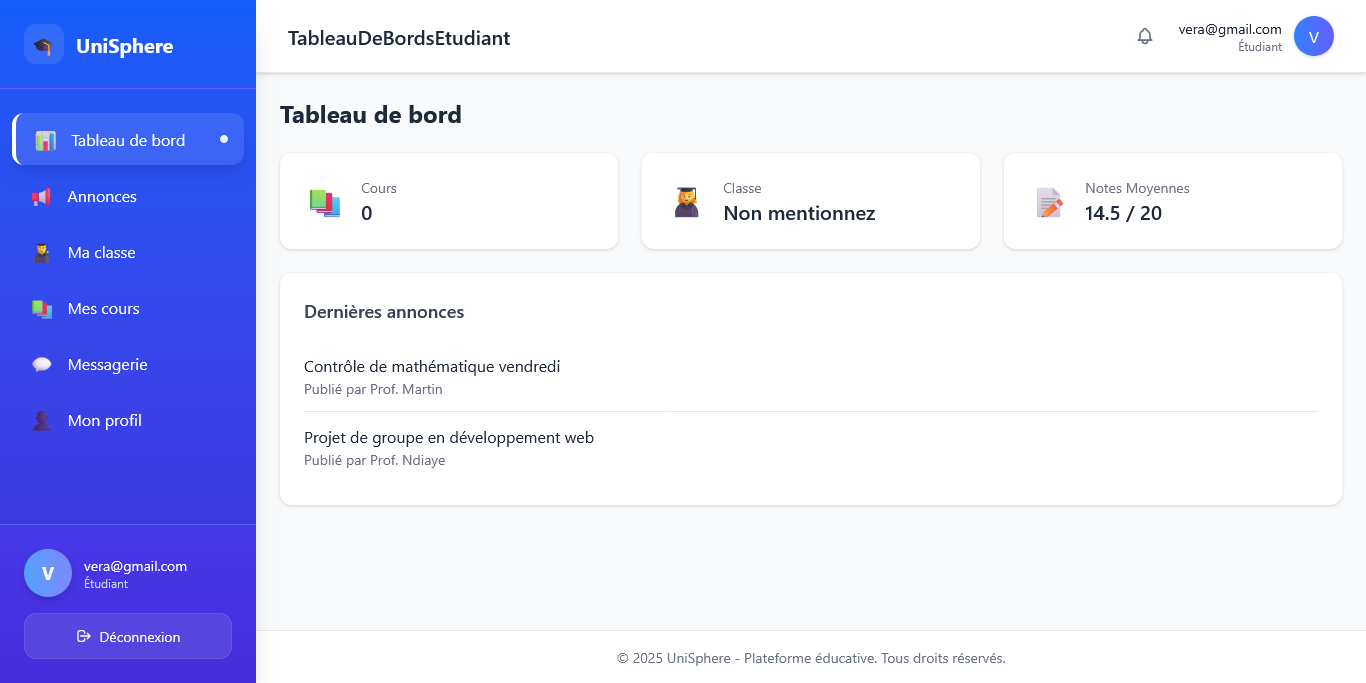
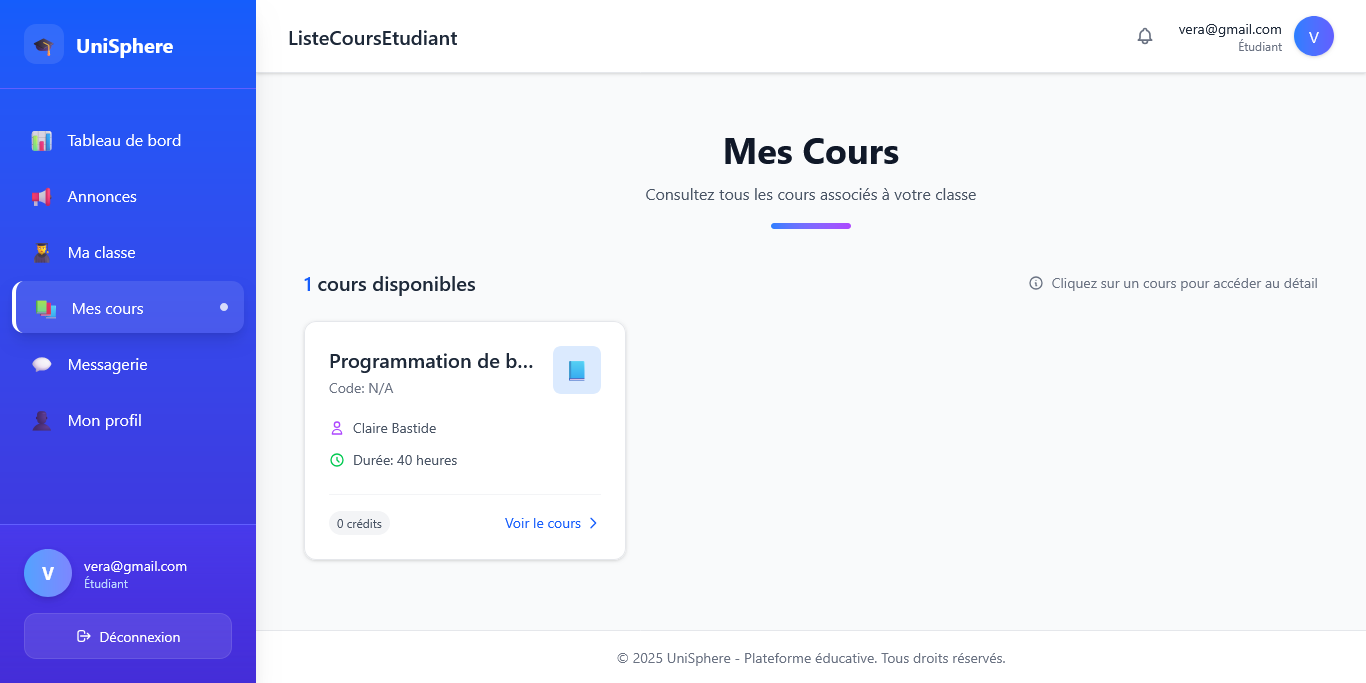


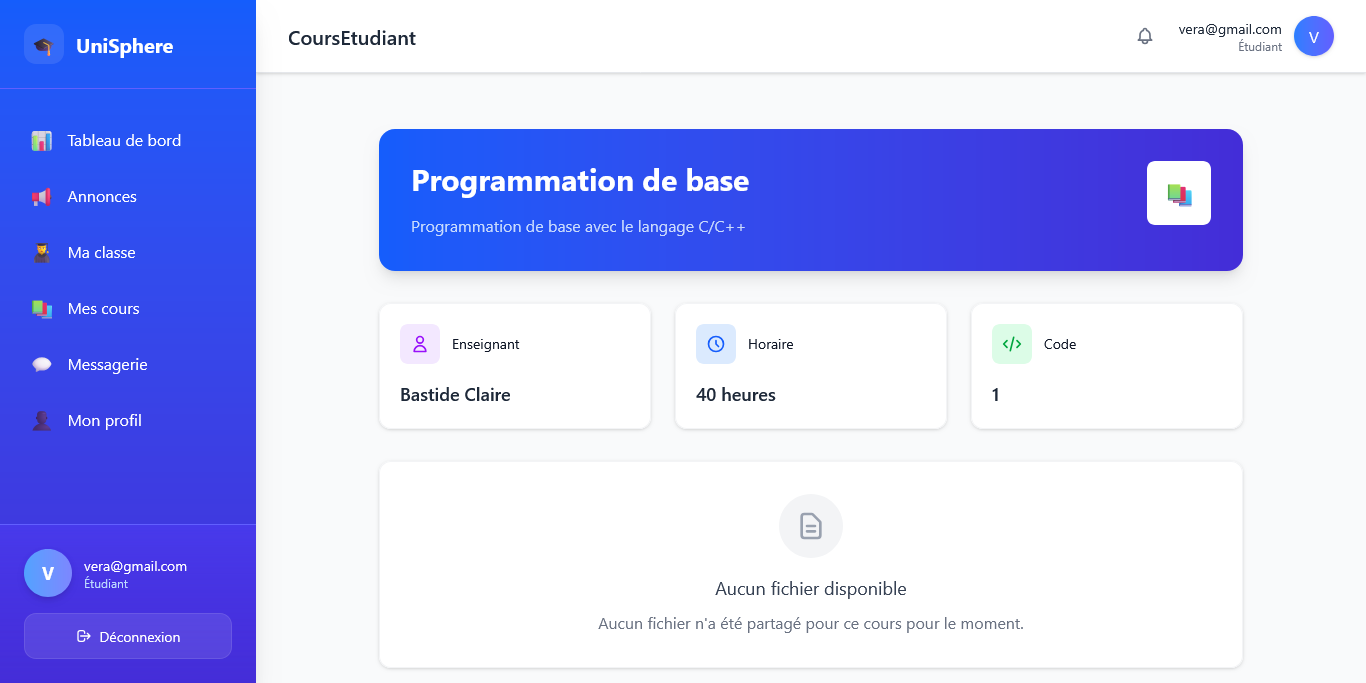
Tableau de bord : Etudiant

* + 1. Liste des cours de l’étudiant



Liste des cours : Etudiant

* + 1. Affichage d’un cours



Affichage d’un cours : Etudiant

* + 1. Tableau de bord enseignants

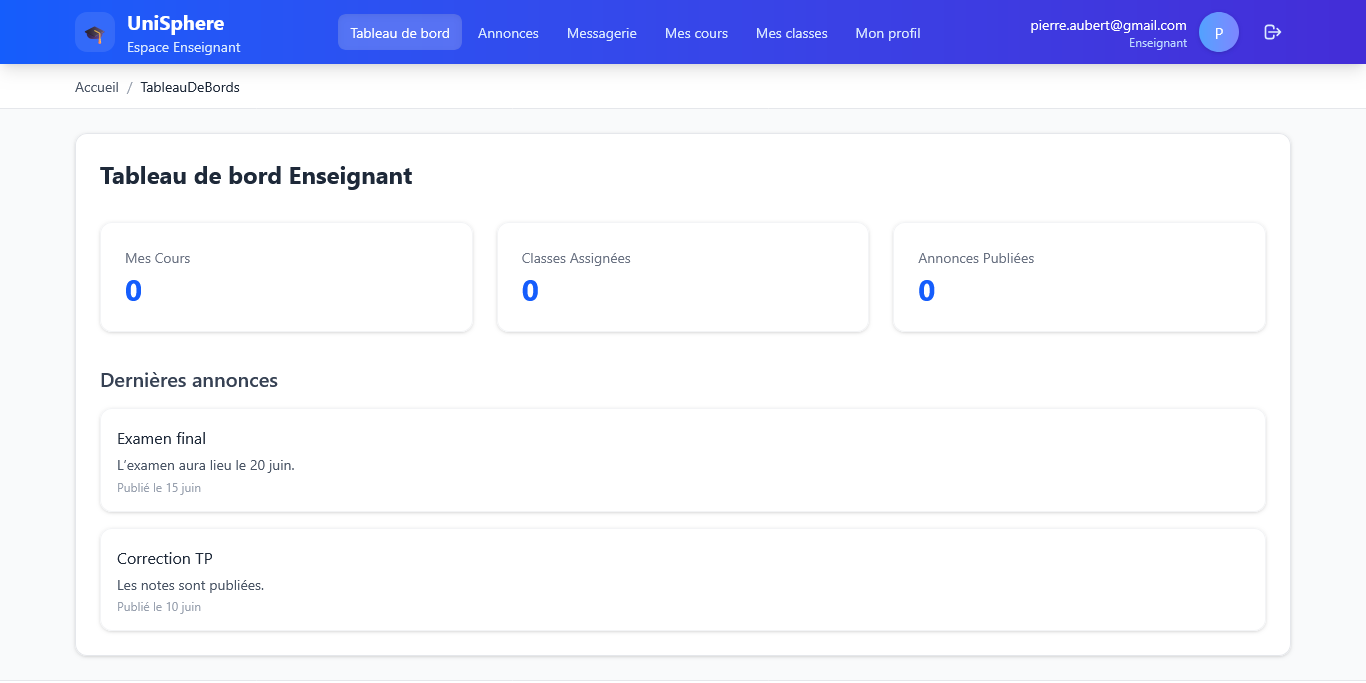
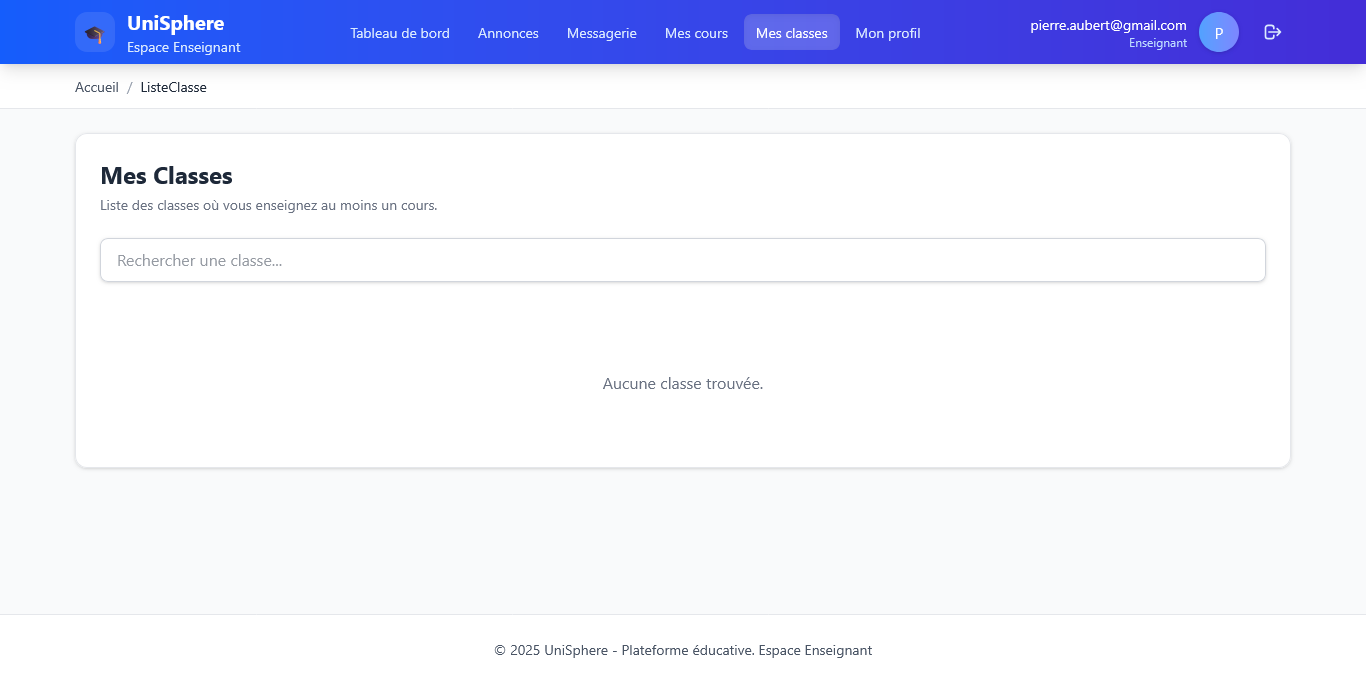


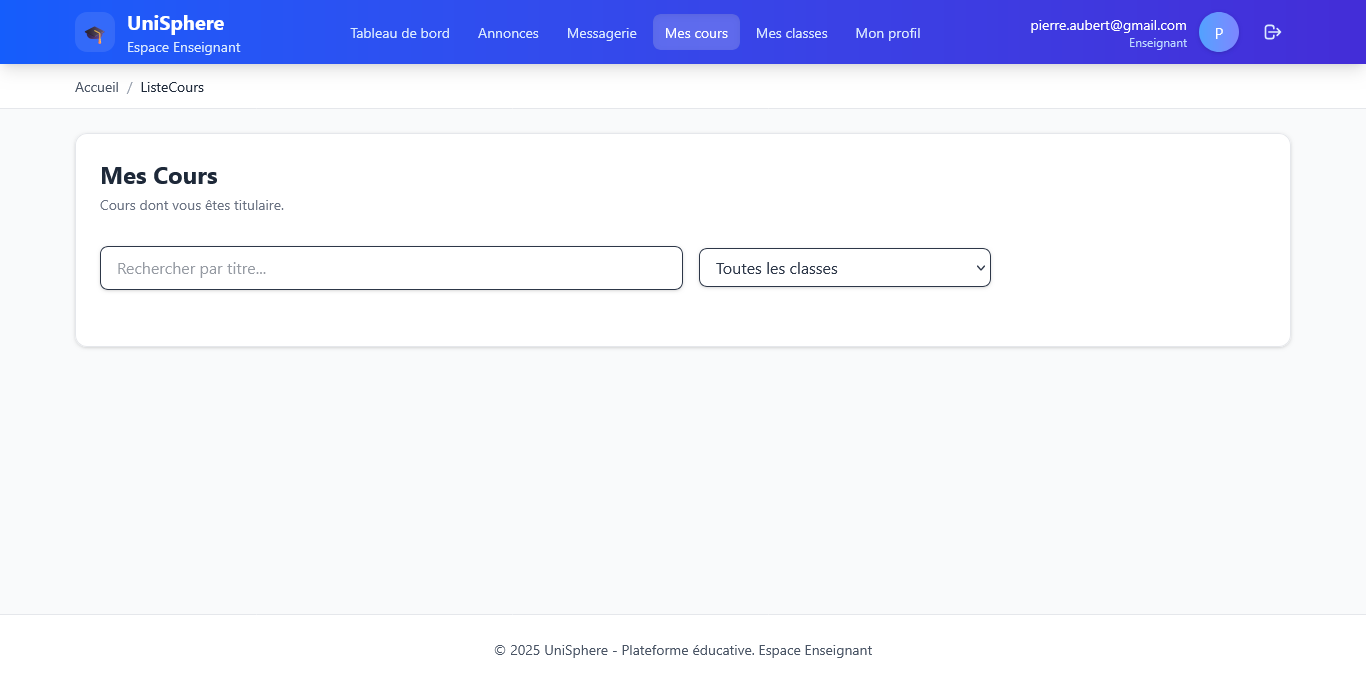
Tableau de bord : Enseignant

* + 1. Liste des classes de l’enseignant



Liste des classes : Enseignant

* + 1. Liste des cours de l’enseignant



Liste des cours : Enseignant

* + 1. Tableau de bord de l’administrateur

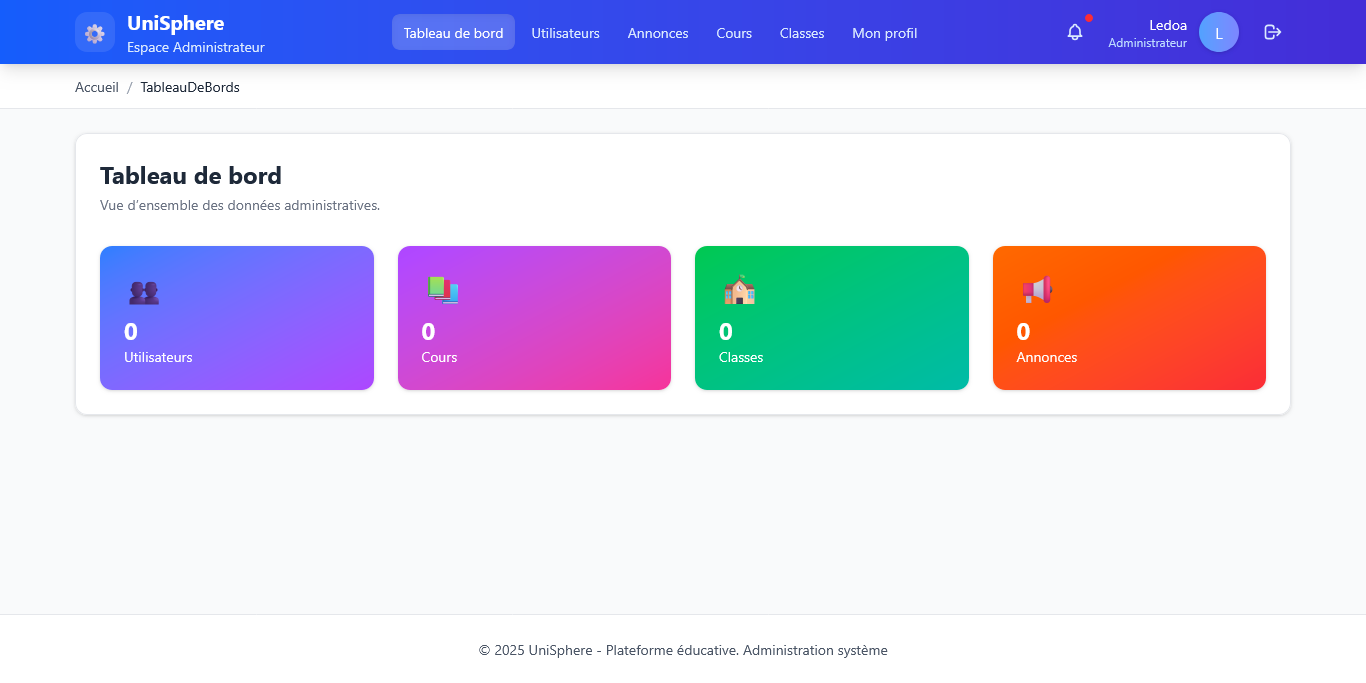
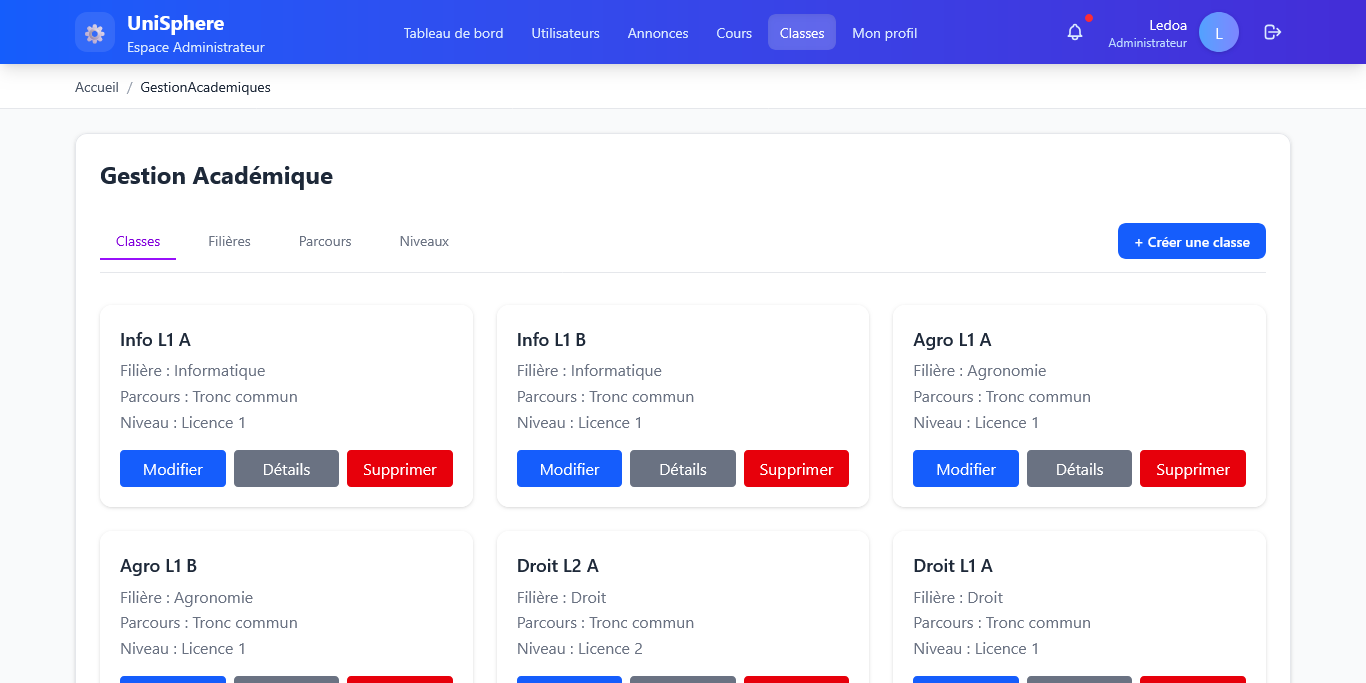


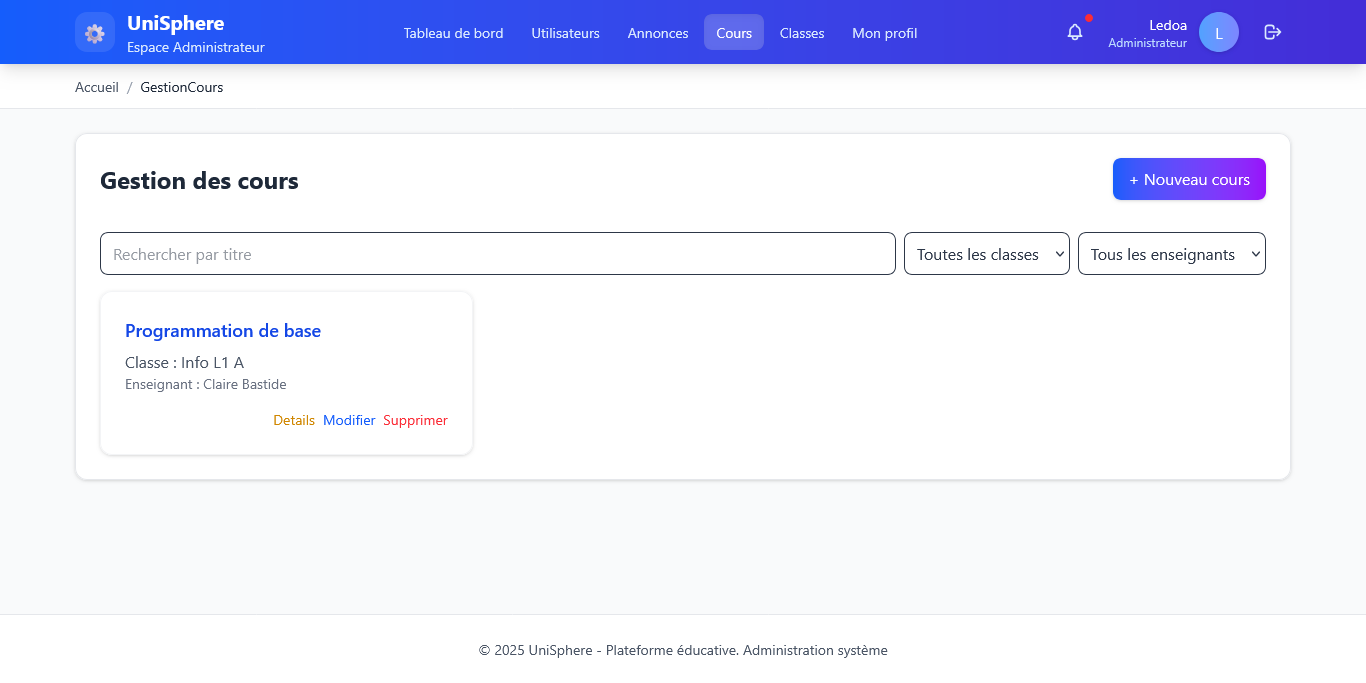
Tableau de bord : Administrateur

* + 1. Gestionnaire des classes



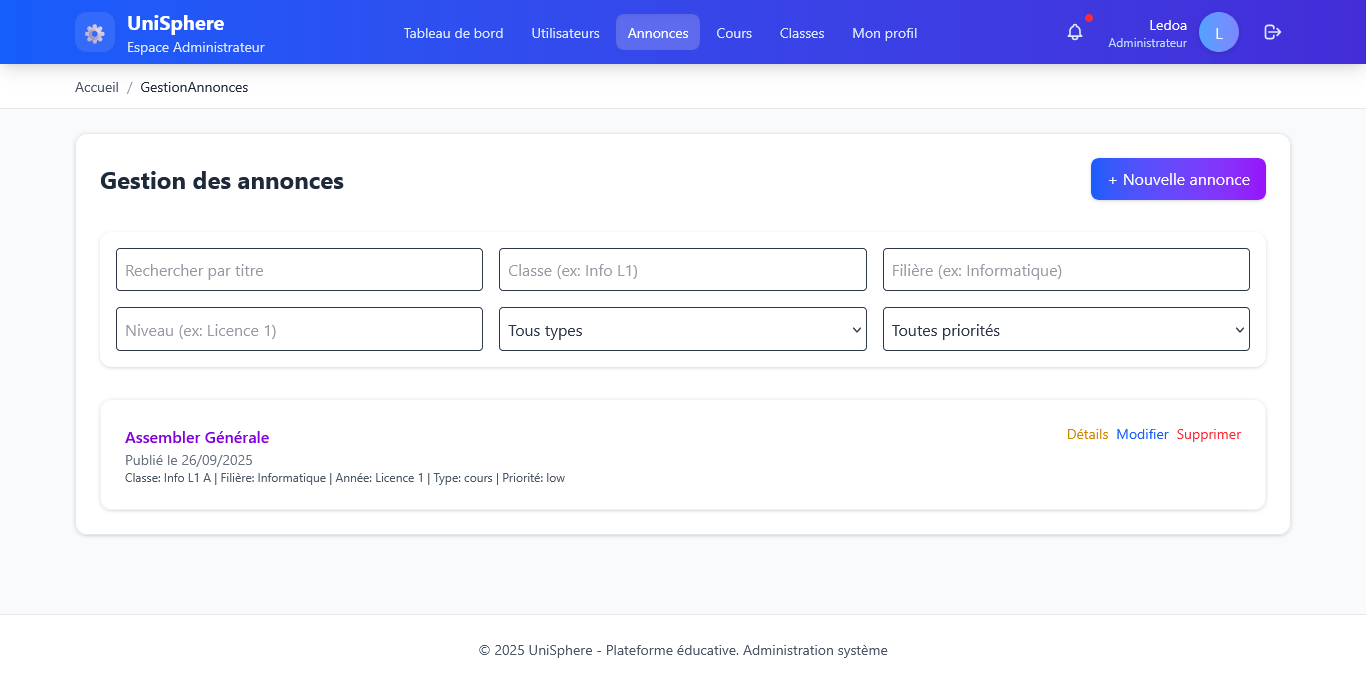
Gestionnaire des classes

* + 1. Gestionnaire des cours



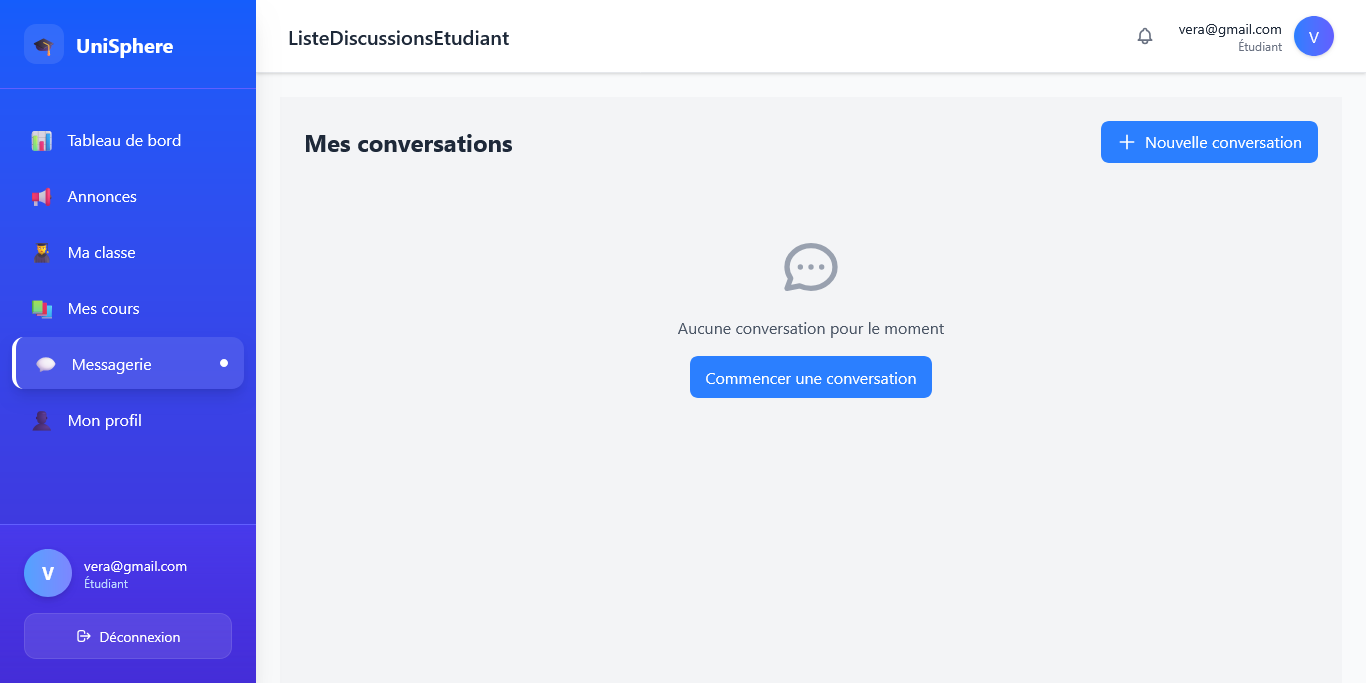
Gestionnaire des cours

* + 1. Gestionnaire des annonces



Gestionnaire des annonces

* + 1. Messagerie



Messagerie

* 1. Conclusion

Pour conclure, nous avons pu expliquer les langages de programmation, outils et logiciels utiliser lors du développement de notre plateforme. Mais aussi présenter les résultats par les illustrations des interfaces de notre plateforme UniSphere. Ainsi dans le prochain chapitre nous allons effectuer à l’évaluation de notre projet et ces perspectives d’avenir.

# EVALUATION ET PERSPECTIVE D’AMELIORATION DU PROJET

* 1. Introduction

La phase d'évaluation constitue une étape cruciale dans le cycle de développement de tout projet informatique [4.01]. Ce chapitre a pour objectif d'évaluer de manière exhaustive la plateforme UniSphere sous plusieurs angles : fonctionnel, technique, ergonomique et pédagogique. Nous allons voir dans ce chapitre les analyses détaillées des résultats obtenus, des limites constatés lors de la réalisation, ainsi que des possibles amélioration future du projet.

* 1. Méthodologie d’évaluation

Afin de mener une évaluation complète et objective, nous avons adopté une approche mixte, combinant plusieurs méthodes de collecte de données quantitatives et qualitatives [4.02]. Cette approche nous permet de trianguler les résultats et d'obtenir une vision complète des forces et faiblesses de la plateforme.

* + 1. Cadre méthodologie générale

Notre méthodologie s'appuie sur le cadre d'évaluation des technologies éducatives proposé par l’UNESCO [4.03], qui recommande une approche systémique intégrant les dimensions techniques, pédagogiques et organisationnelles. Nous avons adapté ce cadre aux spécificités de notre plateforme en y intégrant des éléments propres aux systèmes de gestion de l'apprentissage modernes.

* + 1. Protocol d’évaluation détaillé

Méthodologie d’évaluation de la plateforme

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Type d’évaluation | Méthode utilisée | Participants | Durée | Objectifs spécifiques | Indicateur mesures |
| Test fonctionnels | Scénarios de test manuel [4.04] | Développeur 1 personne | 3 semaines | Vérifier le bon fonctionnement des fonctionnalités | Taux de réussite des test, nombre de bugs critiques |
| Tests de performance | Outils Artillery.io, JMeter [4.05] | Environnement de test dédié | 1 semaine | Mesurer les temps de réponse et la capacité de charge | Temps de réponse, débit, utilisation ressources |
| Test de sécurité | Analyse manuelle outils de scan [4.06] | Equipe sécurité  2 personnes | 1 semaine | Identifier les vulnérabilité potentielles | Nombre de vulnérabilités critique/moyennes/faibles |
| Test utilisateurs | Questionnaires, observations directes, entretiens [4.07] | 15 utilisateurs  5 par profil | 2 semaines | Evaluer l’expérience utilisateur et l’utilité perçue | Score SUS, taux de satisfaction feedback qualitatifs |
| Comparaison avec les plateforme existants | Analyse comparative structurée | Benchmark avec Moodle et Google Classroom | 1 semaine | Positionner UniSphere dans écosystème | Scores comparatifs sur divers critères |

* + 1. Profile des utilisateurs pour les tests

La sélection des utilisateurs tests a été effectuée selon une méthode d'échantillonnage raisonné visant à représenter la diversité des profils d'utilisateurs cibles de la plateforme

Caractéristiques détaillées des utilisateurs participants à l’évaluation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Profil | Nombre | Niveau technique | Expérience avec les LMS | Fréquence d’utilisation prévue |
| Etudiants | 5 | Intermédiaire | Aucune | Quotidienne à hebdomadaire |
| Enseignants | 5 | Débutant à intermédiaire | Aucune | Quotidienne à hebdomadaire |
| Administrateurs | 5 | Avancé | Expérimenté | Quotidienne |

* + 1. Instrument de collecte de données

Nous avons développé et adapté plusieurs instruments de collecte de données spécifiquement pour cette évaluation

* Grille d’évaluation technique : basée sur les norme ISO 25010 [4.08]
* Questionnaire de satisfaction utilisateur : adapté du System Usability Scale (SUS) [4.02]
* Guide d’entretien semi-directif : pour les retours qualitatifs
* Scenario de test utilisateurs : couvrant les principales fonctionnalités
* Checklist de sécurité : basée sur l’OWASP ASVS [4.06]
  1. Résultats de l’évaluation fonctionnelle

L’évaluation fonctionnelle visait à vérifier que toutes les fonctionnalités planifiées avaient été correctement implémentées et qu’elles répondaient aux besoins exprimés

* + 1. Etat d’avancement des fonctionnalités

Etat détaillé de la réalisation des fonctionnalité principales

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fonctionnalité | Etat | Taux de réalisation | Complexité | Notes d’implémentation |
| Authentification | Implémentée | 90 % | Faible | JWT avec expiration configurable |
| Gestion des utilisateurs | Implémentée | 90 % | Moyenne | Interface admin complète, gestion des rôles |
| Messagerie instantanée | Implémentée | 75 % | Elevée | Socket.IO avec transfert de fichiers |
| Gestion des annonces | Implémentée | 90 % | Moyenne | Ciblage multi-niveaux, pièces jointes |
| Gestion des cours | Implémentée | 90 % | Moyenne | Uploadé multiples, validation formats |
| Système de note | Implémentée | 85 % | Moyenne | Calcul automatique, historisation |
| Interface responsive | Implémentée | 75 % | Elevée | Compatible mobile, tablette, desktop |

* + 1. Analyse des écarts fonctionnels

Malgré un taux de réalisation global élevée, il y a quelque écart par rapport aux spécification initiales qui ont été identifier :

* Fonctionnalité de gestion des emplois du temps : reportée à une version ultérieure pour des raisons de complexité technique
* Export avancé des données : limitation aux format basique csv, PDF

Analyse des écarts fonctionnels et leur impact

|  |  |  |  |
| --- | --- | --- | --- |
| Fonctionnalité manquante | Impact utilisateur | Solution alternative | Planning de correction |
| Gestion des emplois du temps | Elevée | - | - |
| Export donnée avancé | Moyen | Exports basique disponibles | Dans 3 mois |

* 1. Résultats de l’évaluation technique

L’évaluation technique s’est concentrée sur les aspects performance, sécurité, maintenabilité et évolutivité de la plateforme.

* + 1. Test de performance

Les tests de performance ont été conduits dans un environnement reflétant les conditions réelles d'utilisation avec une charge graduellement croissante

Résultats détaillés des tests de performance

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Métrique | Valeur | Objectif | Seuil d’acceptation | Statut | Observation |
| Temps de réponse moyen | 187ms | < 500ms | < 1000ms | Atteint | Excellente réactivité |
| Utilisateurs simultanés | 512 | 300 | 200 | Dépassé | Bonne scalabilité |
| Taux d’erreurs | 0.18% | < 1% | <5% | Atteint | Très faible taux d'erreur |
| Temps de chargement de page | 1.8 s | < 3s | < 5s | Atteint | Optimisation correcte |
| Disponibilité | 99.82% | > 99.5% | > 99% | Atteint | Haute disponibilité |
| Utilisation CPU | 68% pic | < 80% | <90% | Atteint | Marge de manœuvre |

* + 1. Evaluation de sécurité

L’évaluation de sécurité a suivi la méthodologie OWASP [4.06] et incluait des tests manuels

Résultats de l’audit de sécurité

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Catégorie de risque | Nombre de vulnérabilité | Niveau de risque | Statut | Mesure correctives |
| Critique | 0 | Aucun | Conforme |  |
| Élevée | 2 | Faible | A modifier | Validation renforcée |
| Moyen | 5 | Acceptable | Conforme | Surveillance continue |
| Faible | 12 | Négligeable | Accepté | Aucune actions |

Les principaux points forts identifiés en matière de sécurité :

* Implémentation robuste de l'authentification JWT
* Hachage des mots de passe avec bcrypt
* Protection contre les injections SQL
* Validation stricte des types de fichiers uploadés
  + 1. Qualité du code et maintenabilité

L'analyse statique du code a été réalisée à l'aide d'outils spécialisés pour évaluer la maintenabilité à long terme

Métrique de qualité du code

|  |  |  |  |
| --- | --- | --- | --- |
| Métrique | Valeur | Seuil acceptable | Interprétation |
| Complexité cyclomatique moyenne | 8.2 | <15 | Bonne maintenabilité |
| Taux de duplication du code | 3.1% | <5% | Très faible duplication |
| Couverture des tests | 78% | >70% | Couverture satisfaisante |
| Dettes techniques | 12 jours | <30 jours | Niveau acceptable |
| Violations des règles de codage | 45 | >100 | Qualité de code bonne |

* 1. Evaluation de l’expérience utilisateur

L’évaluation de l'expérience utilisateur a constitué un volet essentiel de notre démarche, visant à mesurer l'utilisabilité, l'utilité perçue et la satisfaction globale.

* + 1. Résultats quantitatifs

Résultats détaillés du questionnaire de satisfaction sur une échelle de 1 à 5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Critère d’évaluation | Étudiants (moyenne) | Enseignants (moyenne) | Administrateurs (moyenne) | Moyenne générale | Écart-type |
| Facilité d’utilisation | 4.2 | 3.8 | 4.5 | 4.2 | 0.35 |
| Performance perçue | 4.0 | 4.2 | 4.3 | 4.2 | 0.15 |
| Utilité perçue | 4.5 | 4.3 | 4.7 | 4.5 | 0.20 |
| Design de l'interface | 4.1 | 3.9 | 4.2 | 4.1 | 0.15 |
| Fonctionnalités communication | 4.7 | 4.5 | 4.3 | 4.5 | 0.20 |
| Gestion des documents | 4.3 | 4.4 | 4.6 | 4.4 | 0.15 |
| Navigation générale | 4.0 | 3.7 | 4.4 | 4.0 | 0.35 |

Le score SUS (System Usability Scale) moyen calculé est de 82.5, ce qui place UniSphere dans la catégorie "Excellent" selon l'échelle de notation SUS [4.02]

* + 1. Analyse qualitative des retours utilisateurs

Les entretiens et observations ont permis de recueillir des retours riches et détaillés sur l'expérience utilisateur

Points forts fréquemment mentionnées :

* "La centralisation de toutes les ressources pédagogiques au même endroit est un gain de temps considérable" (Enseignant, 15 ans d'expérience)
* "L'interface moderne et épurée facilite la prise en main, même pour les moins technophiles" (Étudiant, première année)
* "La messagerie intégrée permet des échanges rapides sans avoir à basculer entre différentes applications" (Administrateur)

Points d’amélioration identifier :

* "Sur mobile, certains boutons sont trop petits et difficiles à cliquer précisément" (Retour fréquent)
* "L'absence de notifications push hors ligne limite l'immédiateté des communications" (Besoin exprimé)
* "La courbe d'apprentissage pour les fonctionnalités avancées pourrait être plus progressive" (Enseignant)
  + 1. Analyse des tâches utilisateurs

L'observation des utilisateurs lors de la réalisation de tâches spécifiques a permis d'identifier les points de friction dans l'interface

Temps de réalisation des tâches principales (en secondes)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tâche | Expert | Novice | Écart | Difficulté perçue (1-5) |
| Se connecter | 8.2 | 12.5 | 4.3 | 1.2 |
| Consulter un cours | 5.3 | 15.8 | 10.5 | 1.8 |
| Envoyer un message | 12.1 | 28.4 | 16.3 | 2.4 |
| Uploader un fichier | 15.7 | 42.3 | 26.6 | 3.1 |
| Créer une annonce | 25.4 | 68.9 | 43.5 | 3.8 |

* 1. Comparaison avec les plateformes existantes

Une analyse comparative approfondie a été menée pour positionner UniSphere par rapport aux solutions dominantes du marché.

* + 1. Méthodologie de comparaison

La comparaison s'est appuyée sur une grille d'évaluation comprenant 20 critères répartis en 5 catégories principales, chaque critère étant noté sur 5 points.

Grille d’évaluation comparative détaillée

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Catégorie | Critère | Pondération | UniSphere | Moodle | Google Classroom |
| Fonctionnalités | Gestion des cours | 5% | 4.5 | 5 | 4 |
| Fonctionnalités | Communication | 5% | 5 | 3 | 3.5 |
| Fonctionnalités | Évaluation | 5% | 4 | 5 | 4.5 |
| Technique | Performance | 5% | 4.5 | 3.5 | 4.5 |
| Technique | Sécurité | 5% | 4 | 4.5 | 4.5 |
| Utilisabilité | Interface | 5% | 4.5 | 3 | 4.5 |
| Utilisabilité | Accessibilité | 5% | 4 | 3.5 | 4 |
| Économique | Coût total | 5% | 5 | 4 | 3.5 |
| Économique | Maintenance | 5% | 4 | 3 | 5 |
| Stratégique | Flexibilité | 5% | 5 | 4.5 | 3 |

**Score total sur 100** | **100%** | **86.5** | **78.0** | **81.5**

* + 1. Analyse des avantages compétitifs

Points forts d'UniSphere :

* Communication intégrée : solution tout-en-un contrairement aux alternatives
* Flexibilité et personnalisation : architecture modulaire permettant des adaptations poussées
* Coût total de possession : solution open source sans frais de licence

Points faibles relatifs :

* Écosystème de plugins : moins étendu que Moodle
* Intégration écosystémique : moins poussée que Google Classroom
* Maturité : solution nouvelle face à des acteurs établis
  1. Analyse des limites et contraintes

Une analyse approfondie des limitations identifiées durant les tests permet d'anticiper les challenges pour le déploiement à grande échelle

* + 1. Limitations techniques

Analyse détaillée des limitations techniques

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Limitation | Impact | Gravité | Cause racine | Solution temporaire | Solution long terme |
| Dépendance Internet | Accès impossible hors ligne | Élevée | Architecture web classique | Aucune | Application mobile offline (V2) |
| Compatibilité IE | Non supporté | Moyenne | Technologies modernes | Message d'orientation | Pas de support planifié |
| Stockage fichiers | Limite espace disque | Moyenne | Architecture monolithique | Politique de rétention | Stockage cloud scalable |
| Performance mobile | Interface moins fluide | Faible | Optimisations insuffisantes |  | Optimisations CSS avancées |

* + 1. Limitations fonctionnelles

L'analyse des limitations fonctionnelles révèle des écarts entre les attentes utilisateurs et les fonctionnalités actuelles

Principales limitations identifiées :

* Absence de calendrier intégré nécessitant l'utilisation d'outils externes
* Pas de système de visioconférence natif
* Gestion des versions dedocuments limitée
* Reporting et analytics basiques
* Personnalisation de l'interface utilisateur restreinte
  + 1. Challenges organisationnels

Le déploiement d'UniSphere soulève également des défis organisationnels qui nécessitent une attention particulière.

Analyse des challenges organisationnels

|  |  |  |  |
| --- | --- | --- | --- |
| Challenge | Impact potentiel | Stratégie d'atténuation | Responsable |
| Résistance au changement | Adoption lente | Programme de formation et accompagnement | Direction pédagogique |
| Besoin de formation | Sous-utilisation des fonctionnalités | Guides utilisateurs, tutoriels vidéo | Équipe support |
| Charge administrative | Surcharge équipe IT | Automatisation des processus | DSI |
| Maintenance continue | Dégradation performance | Plan de maintenance préventive | Equipe technique |

* 1. Plan d’amélioration et feuille de route

Sur la base des résultats de l'évaluation, un plan d'amélioration structuré a été élaboré pour guider les évolutions futures de la plateforme.

* + 1. Court terme : 0-6 mois

Plan d’amélioration à court terme

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Amélioration | Priorité | Complexité | Impact | Ressources nécessaires | Échéance |
| Notifications push [4.19] | Élevée | Moyenne | Élevé | 1 développeur, 3 semaines | 3 mois |
| Optimisation mobile | Élevée | Faible | Moyen | 1 développeur, 2 semaines | 2 mois |
| Mode sombre | Moyenne | Faible | Faible | 1 développeur, 1 semaines | 4 mois |
| Recherche avancée | Moyenne | Moyenne | Moyen | 1 développeur, 4 semaines | 6 mois |

* + 1. Moyen terme :6-18 mois

Les améliorations à moyen terme visent à combler les lacunes fonctionnelles identifiées et à enrichir l'écosystème de la plateforme.

Améliorations planifiées :

* Application mobile native (React Native)
* Module de visioconférence intégré (WebRTC)
* Système de quiz interactifs avancé
* Tableau de bord Analytics pour enseignants
* Intégration calendrier avec synchronisation
  + 1. Long terme : +18 mois

La vision long terme s'oriente vers des fonctionnalités innovantes positionnant UniSphere comme une plateforme d'apprentissage nouvelle génération.

Axes stratégiques :

* Intelligence artificielle pour la recommandation personnalisée [4.09]
* Adaptive Learning avec parcours individualisés
* Blockchain pour la certification des compétences
  1. Impact et retombée potentielles

L'analyse d'impact permet de quantifier les bénéfices attendus du déploiement d'UniSphere dans un établissement d'enseignement supérieur.

* + 1. Impact pédagogique

Pour les étudiants :

* Accès unifié à l'ensemble des ressources pédagogiques [4.03]
* Autonomie accrue dans le suivi de leur progression académique
* Interaction facilitée avec les enseignants et pairs
* Réduction de la fracture numérique grâce à l'accessibilité multiplateforme

Pour les enseignants :

* Gain de temps significatif dans la gestion des cours [4.10]
* Outils d'évaluation plus efficaces et diversifiés
* Meilleur suivi de l'engagement et de la progression des étudiants
* Flexibilité accrue dans la création et distribution des contenus
  + 1. Impact organisationnel

Pour l'administration :

* Rationalisation des coûts (réduction des licences logicielles)
* Uniformisation des processus académiques
* Meilleure traçabilité des activités pédagogiques
* Indépendance stratégique vis-à-vis des éditeurs propriétaires

Analyse des coûts bénéfices simplifiée

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Poste | Coût/année (solution actuelle) | Coût/année (UniSphere) | Économie | Commentaires |
| Licences logicielles | 75 000 000 AR (25 AR/an) | 0 AR | 75 000 000 AR | Élimination des coûts de licence Moodle/Blackboard |
| Maintenance infrastructure | 40 000 000 AR (13.3 AR/an) | 25 000 000 AR (8.3 AR /an) | 15 000 000 AR | Réduction des besoins de maintenance |
| Formation utilisateurs | 60 000 000 AR (20 AR /an | 40 000 000 AR (13.3 AR /an | 20 000 000 AR | Internalisation partielle du support |
| Support technique | 25 000 000 AR (ponctuel) | 35 000 000 AR (investissement initial) | -10 000 000 AR | Investissement formation accru initial |
| Total | 200 000 000 AR | 100 000 000 AR | 100 000 000 AR | Économie de 50% |

* + 1. Impact sur l’écosystème éducatif

Au-delà de l'établissement, UniSphere contribue à :

* Démocratiser l'accès aux outils numériques éducatifs avancés
* Favoriser l'innovation pédagogique par des fonctionnalités adaptées
* Créer un écosystème open source autour des technologies éducatives
* Réduire la dépendance aux solutions propriétaires internationales
  1. Perspectives de valorisation et déploiement

Les résultats positifs de l'évaluation ouvrent la voie à plusieurs scénarios de valorisation et de déploiement à plus large échelle

* + 1. Modèles de déploiement envisageables

Analyse des modèles de déploiement possibles

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Modèle** | **Avantages** | **Inconvénients** | **Public cible** | **Potentiel économique** |
| **Open source communautaire** | Adoption large, contributions externes | Financement incertain | Universités, développeurs | Faible (prestige) |
| **SaaS (Abonnement)** | Récurrent revenue, scalabilité | Concurrence forte | Établissements moyens | Élevé |
| **Modèle hybride** | Best of both worlds | Complexité de gestion | Marché segmenté | Moyen à élevé |
| **Partenariats institutionnels** | Stabilité, impact | Dépendance aux partenaires | Réseaux universitaires | Variable |

* + 1. Stratégie de contribution communautaire

Pour favoriser l'adoption et l'amélioration continue, une stratégie de contribution communautaire structurée est proposée :

* Documentation complète traduite en plusieurs langues
* API ouverte pour les développeurs tiers
* Programme de contributions avec reconnaissance des contributeurs
* Forums et communauté d'entraide et d'échange
  + 1. Perspectives de recherche et développement

La plateforme UniSphere ouvre également des perspectives pour la recherche dans plusieurs domaines :

* Analyse des données d'apprentissage (Learning Analytics)
* Personnalisation des parcours pédagogiques
* Évaluation de l'impact des technologies sur l'apprentissage
* Interopérabilité entre systèmes éducatifs
  1. Conclusion

En conclusion, l'évaluation exhaustive menée dans ce chapitre démontre qu'UniSphere atteint ses objectifs principaux avec un niveau de qualité satisfaisant. Dans ce chapitre nous avons pu démontrer que la plateforme répond aux besoins identifiés initialement et présente même des avantages compétitifs significatifs par rapport aux solutions existantes.