



Département Génie Informatique

Ref: PFA2-2024-05

Rapport de Projet de Fin d'Année

de

Deuxième année en Génie Informatique

Présenté et soutenu publiquement le 08/05/2024

Par
ABIDELLAOUI Mariem
CHOUCHANE Chifa
MOUSSA Salsabil
KHADHRAOUI Feryel

Application de révision intelligente

Composition du jury

Madame

RIAHI Meriem

Présidente

Madame

BEN YOUNES Ahlem

Encadrante

Année universitaire : 2023-2024

Remerciements

Nous profitons par le biais de ce rapport, pour exprimer nos vifs remerciements à toute personne qui a contribué de près ou de loin à la réalisation de cet humble travail.

*Nous commençons par remercier infiniment Mme **Ahlem Ben Younes**, notre encadrante à L'École Nationale Supérieure d'Ingénieur de Tunis pour sa rigueur scientifique, ses directives son sens d'écoute et d'échange qui nous a aider à éclairer les ambiguïtés rencontrées durant la réalisation du projet. Nous exprimons nos sincères remerciements pour sa gentillesse, son suivi et ses orientations.*

Nos remerciements les plus distingués sont aussi adressés aux membres de jury qui ont eu l'honneur d'avoir accepté d'évaluer ce travail tout en espérant qu'ils trouvent dans ce rapport la qualité de clarté qu'ils attendent.

Dédicace

Nous dédions ce projet à nos chères familles, en particulier à nos parents, qui ont toujours fait des sacrifices pour notre éducation et notre bien-être. Leur soutien constant et leur patience ont été une source immense de motivation pour nous. Nous tenons également à remercier nos frères, nos sœurs et nos meilleurs amis pour l'amour qu'ils nous apportent chaque jour et pour leur extrême générosité. Et enfin, nous voulons exprimer notre gratitude à nos collègues et professeurs.

Table de matière

Introduction générale	1
Chapitre 1 Cadre générale.....	3
Introduction	4
1. Contexte du projet.....	4
2. Étude et critique de l'existant	4
2.1 Solutions existantes.....	4
2.1.1 Khan Academy	4
2.1.2 Coursera.....	5
2.2 Tableau comparatif des solutions existantes	6
3. Solutions proposées	8
4. Méthodologie et formalisme adoptés.....	8
4.1 Cycle en cascade avec des prototypes.....	8
4.2 Langage de modélisation.....	9
5. Conclusion.....	10
Chapitre2 Analyse et spécification des besoins	11
Introduction	12
1. Spécification des besoins	12
1.1 Identification des besoins fonctionnels	12
1.2 Identification des besoins non fonctionnels	13
2. Analyse.....	13
2.1 Identification des acteurs.....	13
2.2 Diagramme des cas d'utilisation global	14
2.3 Raffinement des cas d'utilisation.....	15
2.3.1 Raffinement du cas d'utilisation « Consulter la liste des cours »	15
2.3.2 Raffinement de cas d'utilisation « Consulter les publications ».....	17
Conclusion	18
Chapitre 3 Conception	19
Introduction	20
1. Modèle architectural	20
2. Description de la base de données MongoDB.....	21
3. Diagrammes de séquence	22
3.1 Diagramme de séquence du cas d'utilisation « Gérer groupe messages ».....	22
3.2 Diagramme de séquence du cas d'utilisation « Passer en mode révision »	24
4. Conclusion.....	25

Chapitre 4 Réalisation et mise en œuvre	26
Introduction	27
1. Environnement de travail	27
1.1 Environnement matériel.....	27
1.2 Environnement logiciel.....	28
2. Architecture utilisée	30
3. Interfaces réalisées.....	32
3.1 Interface Signup.....	32
3.2 Interface login.....	33
3.3 Espace étudiant	33
3.3.1 La page d'accueil	33
3.3.2 Gestion d'une publication.....	34
3.3.3 Poster des commentaires	34
3.3.4 Page des matières	35
3.3.5 Liste des cours d'une matière	35
3.3.6 Ajouter un cours aux favoris.....	36
3.3.7 Télécharger un cours	37
3.3.8 Affichage d'un cours.....	37
3.3.9 Communication avec le Chatbot.....	37
3.3.10 Groupe messagerie étudiant.....	38
3.3.11 Mode révision	39
3.4 Espace professeur.....	40
3.4.1 Ajouter une matière	40
3.4.2 Ajouter un cours	41
3.4.3 Liste des matières	41
3.5 Espace admin	42
3.5.2 Interface de gestion des professeurs actuels dans la plateforme	43
4. Conclusion.....	43
Chapitre 5 Réalisation du module Chatbot et Chronomètre Intelligent	44
Introduction	45
1. Chatbot Intelligent	45
1.1 fonctionnalités de chatbot.....	45
1.2 Conception du Chatbot.....	46
1.3 Intégration dans la Plateforme de Révision.....	47
2. Chronomètre intelligent	48

2.1 Fonctionnalités du chronomètre	48
2.2 Conception du chronomètre	49
3.3 Intégration du modèle YOLOv8 dans l'application	50
Conclusion	51
Conclusion générale	52

Table de figures

FIGURE 1. 1 : INTERFACES DE L'APPLICATION "KHAN ACADEMY"	5
FIGURE 1. 2 : INTERFACES DE L'APPLICATION "COURSERA"	6
FIGURE 1. 3 : CYCLE DE VIE EN CASCADE	9
FIGURE 2. 1 : DIAGRAMME DES CAS D'UTILISATION GLOBAL	14
FIGURE 2. 2 : LE RAFFINEMENT DU CAS D'UTILISATION « CONSULTER LA LISTE DES COURS »	15
FIGURE 2. 3 : LE RAFFINEMENT DU CAS D'UTILISATION « CONSULTER LES PUBLICATIONS »	17
FIGURE 3. 1 : LES DIFFERENTS COMPOSANTS DU MODELE MVC.....	21
FIGURE 3. 2 : LE DIAGRAMME DE SEQUENCE DU CAS D'UTILISATION « GERER GROUPE MESSAGES ».....	23
FIGURE 3. 3 : LE DIAGRAMME DE SEQUENCE DU CAS D'UTILISATION « PASSER EN MODE REVISION ».....	24
FIGURE 4. 1 ARCHITECTURE DE L'APPLICATION.....	31
FIGURE 4. 2 INTERFACE SIGNUP.....	32
FIGURE 4. 3 INTERFACE LOGIN	33
FIGURE 4. 4 INTERFACE PAGE D'ACCUEIL	33
FIGURE 4. 5 INTERFACE GESTION PUBLICATION	34
FIGURE 4. 6 INTERFACE AJOUT DES COMMENTAIRES	34
FIGURE 4. 7 INTERFACE DES MATIERES	35
FIGURE 4. 8 LISTE DES COURS D'UNE MATIERE	36
FIGURE 4. 9 AJOUTER AUX FAVORIS	36
FIGURE 4. 10 TELECHARGER UN COURS	37
FIGURE 4. 11 INTERFACE D'AFFICHAGE D'UN COURS.....	37
FIGURE 4. 12 INTERFACE DE CHATBOOT INTELLIGENT	38

FIGURE 4. 13 INTERFACE DE LA MESSAGERIE DES ETUDIANTS.....	38
FIGURE 4. 14 INTERFACES D'AJOUT ET MIS A JOUR DU GROUPE	39
FIGURE 4. 15 MODE REVISION ACTIVE	39
FIGURE 4. 16 INITIALISATION DU CHRONOMETRE.....	40
FIGURE 4. 17 INTERFACE D'AJOUT DE MATIERE.....	41
FIGURE 4. 18 INTERFACE D'AJOUT DU COURS	41
FIGURE 4. 19 LISTE DES MATIERES	42
FIGURE 4. 20 INTERFACE D'APPROBATION DES PROFESSEURS.....	42
FIGURE 4. 21 INTERFACE DE GESTION DE PROFESSEURS.....	43
FIGURE 5. 1 CONCEPTION DE CHATBOT	47
FIGURE 5. 2 INTEGRATION DE L'API	48
FIGURE 5. 3 CONCEPTION DU CHRONOMETRE	49
FIGURE 5. 4 INTEGRATION DU MODELE YOLOV8	51

Table des tableaux

TABLEAU 1 1 : TABLEAU COMPARATIF DES SOLUTIONS EXISTANTES	7
TABLEAU 2 .1 : DESCRIPTION TEXTUELLE DU CAS D'UTILISATION « CONSULTER LA LISTE DES COURS ».....	15
TABLEAU 2. 2 : DESCRIPTION TEXTUELLE DU CAS D'UTILISATION « CONSULTER LES PUBLICATIONS »	17
TABLEAU 3. 1 : DESCRIPTION DE LA BASE DES DONNEES.....	21
TABLEAU 4. 1: ENVIRONNEMENT MATERIEL	28
TABLEAU 4. 2 : ENVIRONNEMENT LOGICIEL.....	28

Introduction générale

Dans un monde où l'accès à l'information est devenu plus facile que jamais, les méthodes traditionnelles d'apprentissage et de révision évoluent pour répondre aux besoins d'une génération numérique. Dans ce contexte, les plateformes éducatives en ligne gagnent en popularité en offrant des outils et des fonctionnalités innovantes pour soutenir l'apprentissage des étudiants.

Dans ce cadre que nous proposons une application de révision intelligente, une plateforme conçue pour faciliter l'apprentissage collaboratif et individuel, ainsi que la communication au sein des communautés étudiantes spécialisées.

Cette application offrira un espace partagé où les étudiants d'une même spécialité pourront se connecter, partager des documents, poser des questions et interagir avec des publications pertinentes. Elle facilitera également la communication à travers des groupes de messagerie, favorisant ainsi la collaboration et le partage des connaissances. De plus, l'application intégrera un espace dédié aux cours dispensés par les professeurs, permettant aux étudiants d'interagir avec le contenu et de bénéficier d'un soutien supplémentaire grâce à un chatbot intelligent.

En outre, notre application comprendra des fonctionnalités innovantes telles qu'un espace de révision intelligent qui adaptera le contenu en fonction des besoins individuels des utilisateurs, ainsi qu'un chronomètre qui arrêtera le temps lorsque l'étudiant quittera son bureau, favorisant ainsi une gestion efficace du temps d'étude. Enfin, elle offrira des espaces d'administration pour gérer les utilisateurs de l'application et pour les professeurs afin de gérer leurs cours.

L'intérêt majeur de notre application réside dans sa capacité à répondre aux besoins divers et évolutifs des étudiants en offrant un espace centralisé et interactif où ils peuvent accéder à des ressources pédagogiques, collaborer avec leurs pairs et bénéficier d'un soutien personnalisé. Ce projet est détaillé dans ce présent rapport organisé en quatre chapitres :

-Le premier chapitre abordera 1 "Introduction générale" du projet, décrivant le "Cadre général

du projet", tout en offrant une analyse des solutions existantes et en présentant la solution envisagée.

-Le deuxième chapitre, intitulé "Spécification des besoins et Analyse", se focalisera sur l'identification des besoins fonctionnels et non fonctionnels de l'application, ainsi que sur l'analyse des différents cas d'utilisation.

-Le troisième chapitre, "Conception", sera dédié à la présentation des modèles conceptuels, contribuant à la préparation du projet en vue de son implémentation.

-Le quatrième chapitre, "Réalisation et mise en œuvre", exposera l'environnement et les outils de développement utilisés, tout en mettant en lumière les interfaces les plus pertinentes.

-Le dernière chapitre, " Réalisation du module Chatbot et Chronomètre Intelligent ", se penchera sur les fonctionnalités et l'intégration du module Chatbot et du Chronomètre Intelligent dans l'application.

Finalement, ce rapport est clôturé par une "Conclusion générale " pour récapituler le travail réalisé et donner quelques perspectives.

Chapitre 1

Cadre générale

Introduction

Au commencement de ce chapitre, nous poserons les bases du projet en général. Par la suite, nous allons nous pencher précisément sur l'analyse et l'évaluation de la conjoncture présente. En ultime partie, nous délivrerons une présentation globale de la solution préconisée et du processus de mise en place envisagé.

1. Contexte du projet

Avec l'avènement de la transformation numérique dans divers secteurs, y compris l'éducation, de nouvelles opportunités émergent pour améliorer les méthodes d'apprentissage et de révision des étudiants.

Dans ce contexte, notre projet vise à répondre à ces besoins en créant une plateforme de révision interactive et intelligente destinée aux étudiants. Cette plateforme offre une approche innovante pour aider les étudiants à mieux comprendre et à retenir le contenu des cours, en utilisant des fonctionnalités telles que l'interaction avec un chatbot intelligent et la révision collaborative.

L'objectif principal est de fournir aux étudiants un outil interactif et collaboratif qui facilite leur préparation aux examens et qui encourage leur engagement actif dans le processus d'apprentissage. En offrant des fonctionnalités avancées telles que la gestion du temps avec un chronomètre, la détection d'absence et des suggestions personnalisées, cette plateforme cherche à optimiser l'environnement de révision pour maximiser l'efficacité de l'apprentissage.

En résumé, le projet se situe à l'intersection de l'éducation et de la technologie, visant à exploiter les avantages de la transformation numérique pour améliorer les méthodes de révision des étudiants et à favoriser leur succès académique.

2. Étude et critique de l'existant

Dans la portion suivante, nous présenterons et nous critiquerons les solutions existantes avant de proposer notre propre résolution.

2.1 Solutions existantes

2.1.1 Khan Academy

Khan Academy est une plateforme éducative en ligne qui propose une grande variété de ressources d'apprentissage gratuites. Elle propose des leçons vidéo, des exercices interactifs et

des matériaux d'apprentissage dans des sujets allant des mathématiques et des sciences à l'histoire et à l'art. La plateforme utilise une approche d'apprentissage personnalisé, permettant aux étudiants d'apprendre à leur propre rythme et de recevoir des commentaires instantanés sur leur progression.

Une interface de cette application est présentée par la figure 1.1.

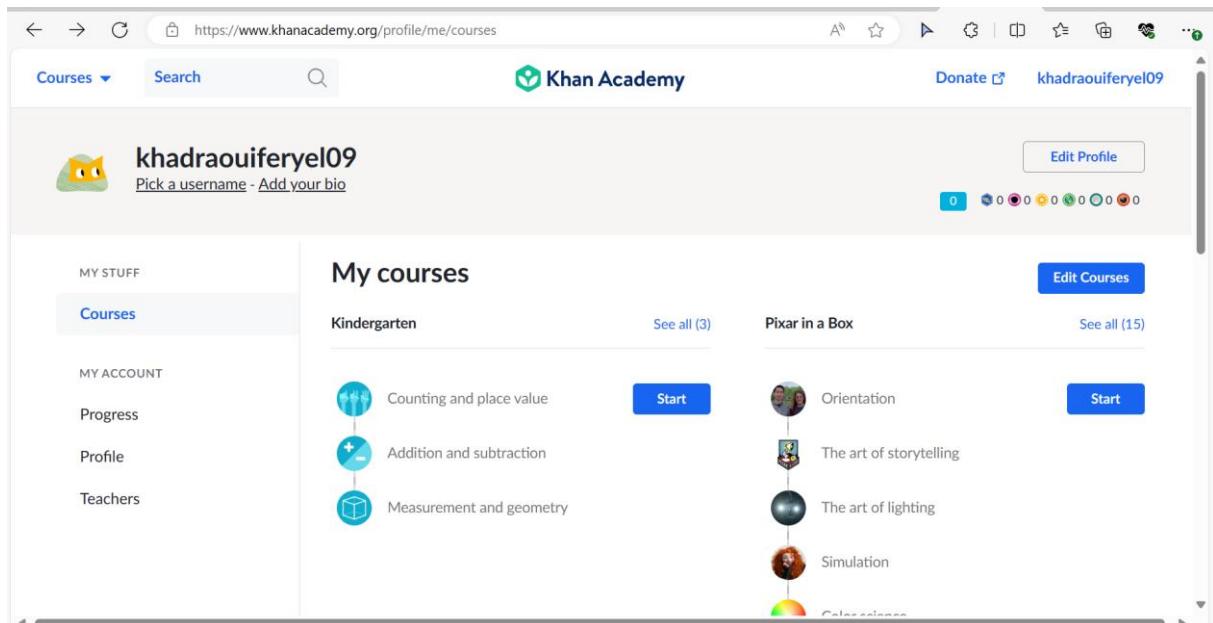


Figure 1. 1 : Interfaces de l'application "Khan Academy"[1]

2.1.2 Coursera

Coursera est une plateforme en ligne qui propose des cours universitaires et des formations professionnelles dispensés par des institutions renommées du monde entier. Les cours sur Coursera sont généralement dispensés sous forme de vidéos, de lectures, d'exercices pratiques et de devoirs. Les étudiants peuvent suivre ces cours à leur propre rythme, avec la possibilité d'obtenir une certification officielle à la fin du programme moyennant des frais.

Une interface de cette application est présentée par la figure 1.2.

The screenshot shows the Coursera Plus application interface. At the top, there is a header with the Coursera logo, a 'PLUS' badge, and navigation links for 'Plus de 7 000 cours, Projets Guidés, Spécialisations et Certificats Professionnels' and 'Commencer un essai gratuit de 7 jours'. Below the header, a message says 'Afficher 4 résultats sur 3841 pour « Computer Science »'. Three course cards are displayed:

- Python for Data Science, AI & Development** by IBM. It has a rating of 4.6 (35369) and 810 000 students. It is categorized as Beginner.
- Learn to Program: The Fundamentals** by University of Toronto. It has a rating of 4.7 (6371) and 450 000 students. It is categorized as Beginner.
- Google Cybersecurity** by Google. It has a rating of 4.8 (27150) and 690 000 students. It is a Professional Certificate.

Figure 1. 2 : Interfaces de l'application "Coursera"[2]

2.2 Tableau comparatif des solutions existantes

Après avoir effectué une étude approfondie des solutions existantes, nous avons identifié plusieurs fonctionnalités intéressantes pour notre application, on les présente dans le tableau 1.1. Ces fonctionnalités comprennent :

- **Offre des cours gratuits** : La plateforme propose une sélection de cours et de ressources pédagogiques accessibles gratuitement, sans aucun coût pour les utilisateurs.
- **Dispense des cours universitaires** : La plateforme offre des cours dispensés par des universités ou des professeurs universitaires, couvrant une variété de sujets académiques.
- **Plateforme adaptée à l'apprentissage autonome** : La plateforme est conçue pour permettre aux apprenants d'étudier de manière autonome.
- **Éducation collaborative en groupes entre les étudiants** : La plateforme favorise l'apprentissage collaboratif en fournissant aux étudiants des outils de communication et de partage d'informations. Cela encourage l'engagement et l'interaction entre les apprenants.

- **Partage de cours ou de problèmes entre les étudiants** : Les étudiants peuvent partager des cours, des ressources pédagogiques ou des problèmes avec d'autres étudiants sur la plateforme, favorisant ainsi la collaboration, l'entraide et le partage des connaissances.
- **Mode de révision intelligent** (assure que l'étudiant étudie avec un modèle de détection) : La plateforme intègre des algorithmes de détection afin de garantir la présence de l'étudiant pendant ses sessions de révision.
- **Communication avec un chatbot** : La plateforme intègre un chatbot qui permet aux utilisateurs de poser des questions, d'obtenir des informations supplémentaires et d'interagir avec la plateforme de manière conversationnelle.

Aucune entrée de table d'illustration n'a été trouvée.

Tableau 1 1 : Tableau comparatif des solutions existantes

Critères de Comparaison	Khan Academy	Coursera
Offre des cours gratuits.	Oui	Non
Dispense des cours universitaires.	Non	Oui
Plateforme adaptée à l'apprentissage autonome.	Oui	Oui
Éducation collaborative en groupes entre les étudiants	Non	Oui
Partage de cours ou de problèmes entre les étudiants	Non	Oui
Mode de révision intelligent (assure que l'étudiant étudie avec un modèle de détection)	Non	Non
Communication avec un chatbot	Non	Non

3. Solutions proposées

Nous envisageons le développement d'une plateforme de révision interactive et intelligente pour les étudiants, intégrant la plupart des fonctionnalités présentes dans les plateformes existantes ainsi que module intelligent. Notre proposition inclut les éléments suivants : L'accès aux cours sera facilité pour les étudiants, qui pourront aisément sélectionner et consulter les cours déposés par les enseignants grâce à une interface conviviale et intuitive. Pour favoriser la révision collaborative, notre plateforme permettra la formation de groupes d'étude où les membres pourront échanger des messages et partager des ressources pertinentes pour la révision. Elle offrira également la possibilité de partager des documents et des supports de cours entre les étudiants. Ces ressources pourront être consultées par les autres membres de la plateforme, qui auront également la possibilité de laisser des commentaires sur ces publications. Notre plateforme comprendra un mode révision avancé avec des fonctionnalités sophistiquées telles qu'un chronomètre intégré pour la gestion du temps, la détection d'absence pour encourager une présence régulière lors des sessions de révision, ainsi que des suggestions personnalisées pour optimiser l'environnement de révision en fonction des besoins individuels de chaque étudiant. Un chatbot intelligent sera intégré à notre plateforme pour répondre aux questions des étudiants concernant le contenu des cours. En implémentant ces fonctionnalités, notre solution vise à créer un environnement d'apprentissage interactif et efficace, favorisant la préparation aux examens et encourageant l'engagement actif des étudiants dans leur processus d'apprentissage.

4. Méthodologie et formalisme adoptés

Dans cette section, nous explorerons la méthodologie et le formalisme adoptés dans le cadre de notre projet.

4.1 Cycle en cascade avec des prototypes

Dans le cadre de notre projet, nous avons adopté le modèle en cascade, également connu sous le nom de cycle en cascade, pour la gestion du développement logiciel. Ce modèle séquentiel organise le processus de développement en une série de phases linéaires, avec chaque phase dépendant des résultats de la phase précédente. Nous avons intégré des prototypages dans la phase d'analyse pour mieux comprendre les besoins et les exigences du projet, permettant ainsi une exploration plus approfondie des fonctionnalités et des solutions potentielles dès les

premières étapes du processus de développement. Cette approche nous a permis d'élaborer des spécifications détaillées et précises avant de passer à la phase de conception, ce qui a contribué à réduire les risques et les incertitudes tout au long du cycle de développement.

La figure 1.3 illustre le cycle de vie en cascade.

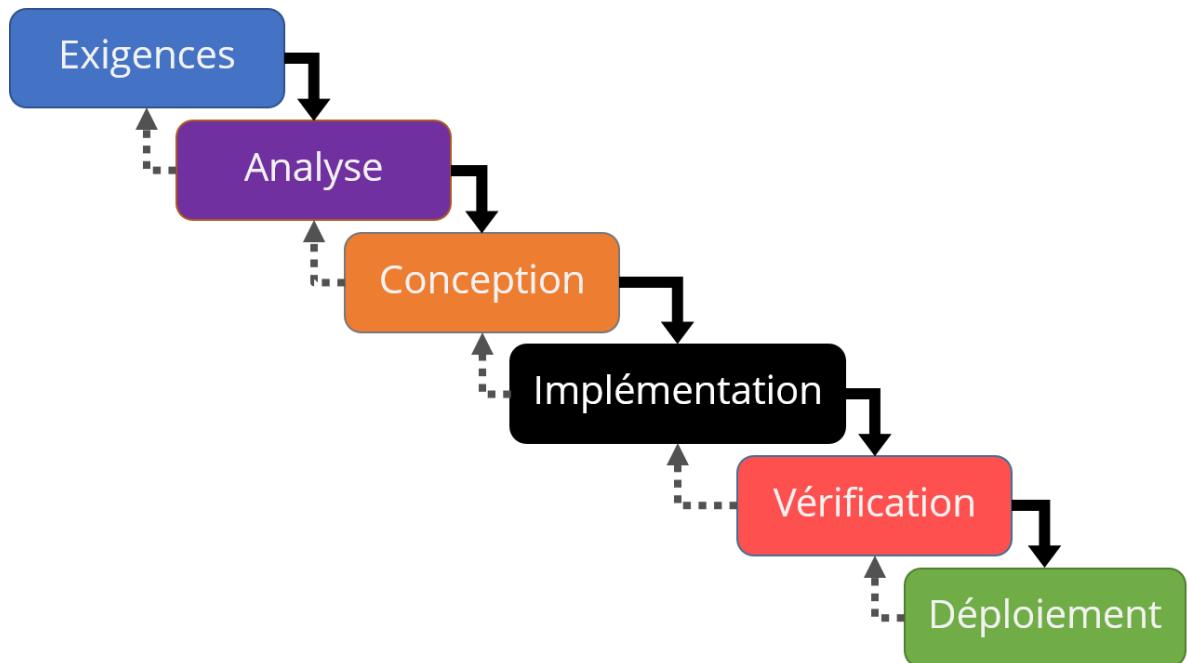


Figure 1. 3 : Cycle de vie en cascade

4.2 Langage de modélisation

Pour modéliser les fonctionnalités que doit offrir notre application, nous avons choisi le langage de modélisation UML puisqu'il s'agit d'un langage simple, formel et basé sur les notions de l'orienté objet.

Nous avons utilisé les diagrammes suivants :

- **Diagramme de cas d'utilisation** : Il montre les interactions entre les acteurs externes et un système logiciel, mettant en évidence les différents cas d'utilisation du système, qui sont des séquences d'actions réalisées par un acteur pour atteindre un objectif spécifique.
- **Diagramme de séquence** : Il illustre comment les objets interagissent dans une séquence temporelle, montrant comment ils communiquent les uns avec les autres dans une série de messages dans le cadre d'un scénario spécifique. Ce diagramme est utile pour comprendre le flux des opérations et des interactions entre les différents composants d'un système.

5. Conclusion

Dans ce chapitre, nous avons présenté le contexte général du projet ainsi que nous avons détaillé et critiqué les solutions existantes. Nous avons également détaillé la solution proposée et la méthodologie de développement. Dans le prochain chapitre, nous allons étudier l'analyse et spécification des besoins.

Chapitre2

Analyse et spécification des besoins

Introduction

Ce chapitre abordera les exigences fonctionnelles et non fonctionnelles de notre projet qui consiste à concevoir une plateforme de révision interactive et intelligente pour les étudiants. Dans laquelle on va repérer les principaux acteurs impliqués dans le système et présenter le diagramme de cas d'utilisation.

1. Spécification des besoins

La spécification des besoins est la première phase dans toute application à développer. Dans cette partie, nous allons identifier les besoins fonctionnels ainsi que non fonctionnels. On va détailler aussi les raffinements de cas d'utilisation.

1.1 Identification des besoins fonctionnels

- **Sélection des cours**
 - Les étudiants doivent pouvoir sélectionner les cours qu'ils suivent parmi une liste fournie par les professeurs.
 - La liste des cours doit être filtrée par niveau et domaine d'études.
- **Gestion des publications**
 - Les étudiants peuvent poster des publications et aussi des commentaires.
- **Effectuer des cours**
 - Les étudiants et les enseignants peuvent ajouter, modifier ou effacer des cours
- **Interaction avec le chatbot intelligent**
 - Le chatbot intelligent doit pouvoir répondre aux questions des étudiants sur le contenu des cours.
 - Les questions peuvent être formulées en langage naturel et le chatbot doit fournir des réponses précises et pertinentes.
 - Le chatbot doit s'adapter au niveau de connaissance de l'étudiant et à sa progression dans le cours.
- **Révision collaborative**
 - Les étudiants doivent pouvoir créer des groupes de révision et inviter d'autres étudiants à les rejoindre.
 - Les membres d'un groupe de révision doivent pouvoir échanger des messages et partager des ressources.

- La plateforme doit offrir des outils de collaboration tels que le partage de documents et la visioconférence.
- **Mode révision**
 - Le mode révision doit offrir des fonctionnalités avancées pour optimiser l'environnement de révision.
 - Ces fonctionnalités incluent la gestion du temps avec chronomètre, la détection d'absence et des suggestions personnalisées.

1.2 Identification des besoins non fonctionnels

Ce sont les exigences qui ne concernent pas spécifiquement le comportement du système mais qui assurent sa bonne qualité. Les principaux besoins non fonctionnels de la plateforme sont :

- **Maintenabilité** : Le code de la plateforme doit être bien écrit et facile à comprendre afin d'assurer une maintenance facile et rapide.
- **Usabilité** : L'application doit être facile à utiliser, intuitive et conviviale.
- **Sécurité** : Le système assure une sécurité renforcée, préservant la confidentialité des données. Chaque utilisateur dispose d'un identifiant unique et d'un mot de passe crypté. De plus, l'utilisation de jetons d'authentification limite l'exposition des informations sensibles lors des échanges client-serveur.
- **Ergonomie** : L'application doit garantir une interface utilisateur intuitive et efficace, optimisant ainsi l'expérience de l'utilisateur.

2. Analyse

L'analyse permet d'identifier les divers acteurs qui peuvent interagir avec l'application. Elle offre une représentation fonctionnelle des divers participants et des fonctionnalités du système à travers un diagramme de cas d'utilisation.

2.1 Identification des acteurs

Cette application possède trois acteurs principaux :

- **Administrateur** : C'est la personne responsable de l'administration de l'application, il gère l'acceptation et refus des professeurs
- **Professeur** : Une personne qui accède à l'application avec admission du administrateur.

- **Etudiant** : Une personne qui possède un compte et accède à la plateforme.

2.2 Diagramme des cas d'utilisation global

Le diagramme de cas d'utilisation offre une perspective du système du point de vue de l'utilisateur, définissant ainsi les exigences qu'un utilisateur attend du système. Il englobe tous les scénarios d'utilisation liés directement ou indirectement aux acteurs.

La figure 2.1 illustre tous les cas d'utilisation fondamentaux pour obtenir une vue d'ensemble du fonctionnement de l'application.

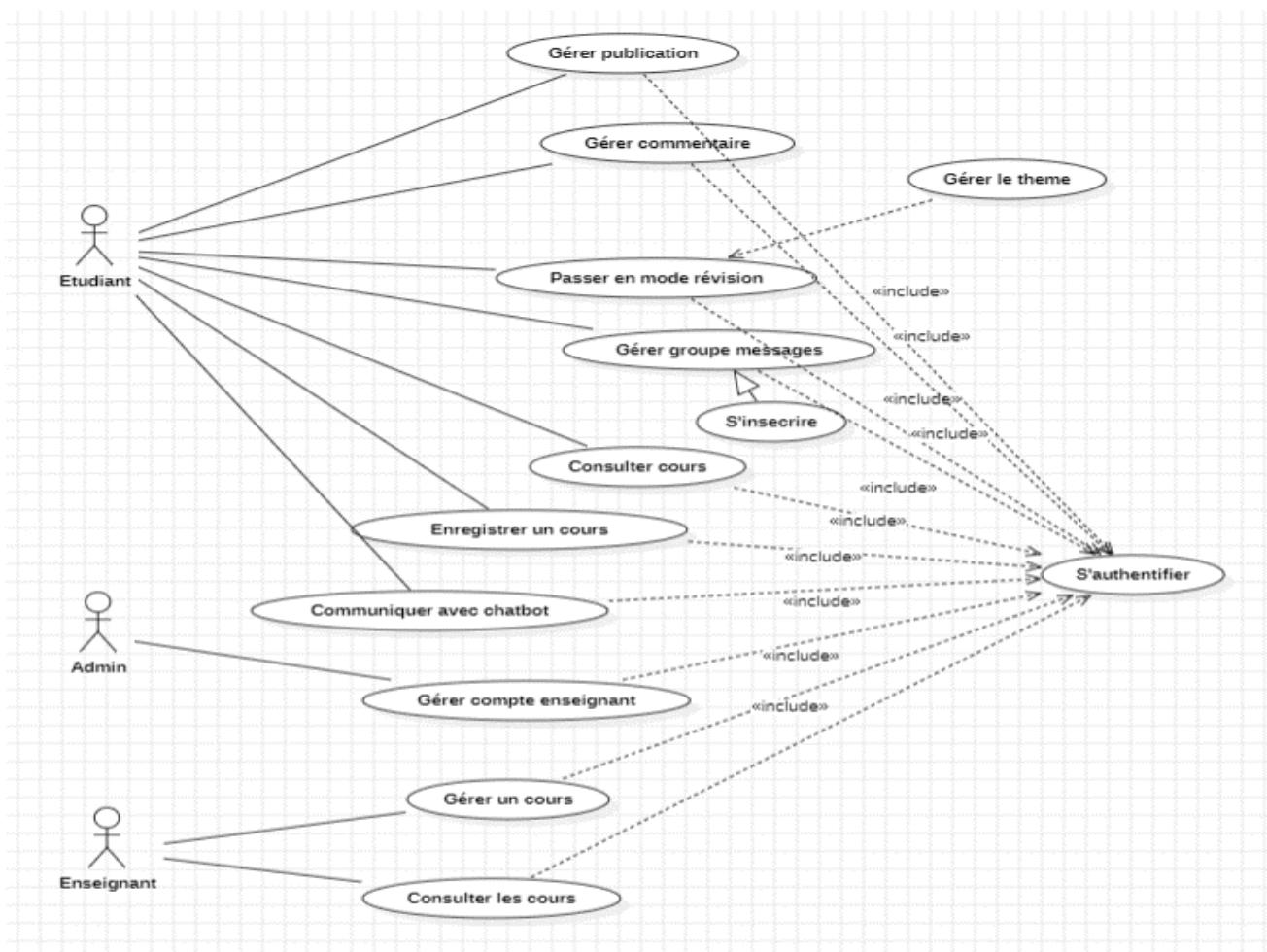


Figure 2. 1 : Diagramme des cas d'utilisation global

2.3 Raffinement des cas d'utilisation

Dans cette section, nous présentons une série de cas d'utilisation pour une analyse approfondie. Cela comprend des diagrammes de cas d'utilisation détaillés et des descriptions textuelles pour chaque cas.

2.3.1 Raffinement du cas d'utilisation « Consulter la liste des cours »

- Diagramme de cas d'utilisation :

La figure 2.2 illustre le raffinement du cas d'utilisation "Consulter la liste des cours ". L'étudiant a la possibilité de partager un cours ou un document déjà publié par le professeur avec ces collègues dans les groupes de messages, de télécharger un cours, de consulter un cours pour le lire dans ce stage il peut aussi interagir avec un chatbot pour mieux comprendre le cours, ainsi que d'ajouter un cours à sa liste de favoris.

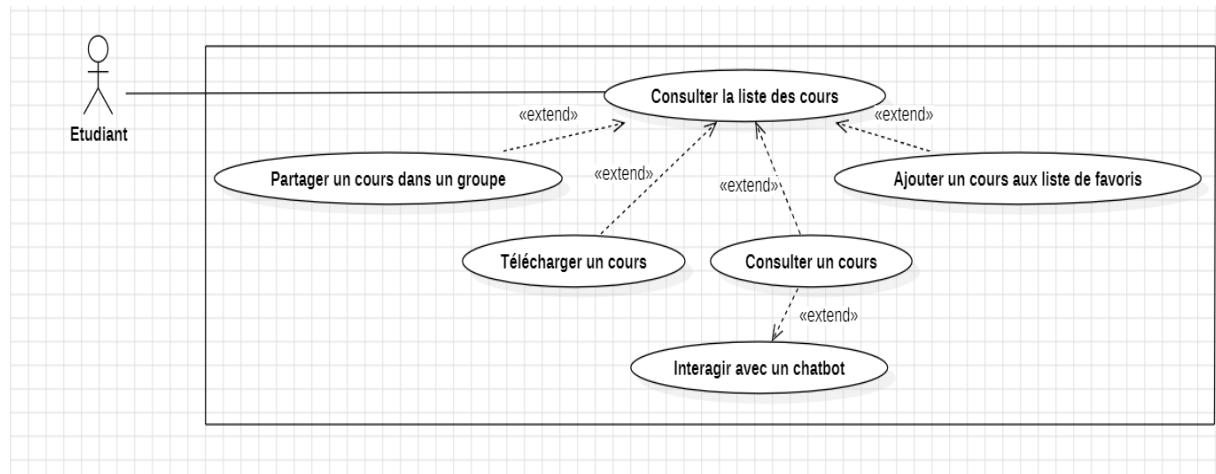


Figure 2. 2 : Le raffinement du cas d'utilisation « Consulter la liste des cours »

- Description textuelle :

Le tableau 2.1 présente la description textuelle du cas d'utilisation « Consulter la liste des cours ».

Tableau 2 .1 : Description textuelle du cas d'utilisation « Consulter la liste des cours »

Titre :	Consulter la liste des cours
Acteur :	Étudiant

Résumé :	C'est le cas d'utilisation qui permet à l'étudiant de consulter la liste des cours ajoutés par le professeur
Préconditions :	Authentification
Scénario nominal :	<p>1- L'étudiant demande de consulter la liste des cours.</p> <p>2- Le système affiche la liste des cours déjà ajoutés par les professeurs.</p> <p>3- L'étudiant sélectionne un cours pour le lire.</p> <p>4- Le système affiche le cours et le chatbot</p> <p>5- L'étudiant interagit avec le chatbot</p> <p>6- Le système affiche la réponse du chatbot</p>
Scénario alternatif :	<p>1- À l'étape 3 de scénario nominal l'étudiant peut sélectionner un cours pour le partager avec ses collègues</p> <ul style="list-style-type: none"> - Le système affiche la liste des groupes où l'étudiant est déjà inscrit - L'étudiant choisit un groupe pour partager le cours avec. <p>2- À l'étape 3 de scénario nominal l'étudiant peut sélectionner un cours pour le télécharger</p> <p>3- À l'étape 3 de scénario nominal l'étudiant peut sélectionner un cours pour l'ajouter à sa liste de favoris.</p>

2.3.2 Raffinement de cas d'utilisation « Consulter les publications »

- **Diagramme de cas d'utilisation :**

La figure 2.3 illustre le raffinement du cas d'utilisation "Consulter les publications". L'étudiant a la possibilité de créer une publication pour partager des documents ou poser des questions, de consulter la liste des commentaires d'une publication dans ce stage l'étudiant peut ajouter un commentaire, modifier ou supprimer son commentaire, d'aimer une publication, ainsi que de partager une publication dans un groupe avec ses collègues.

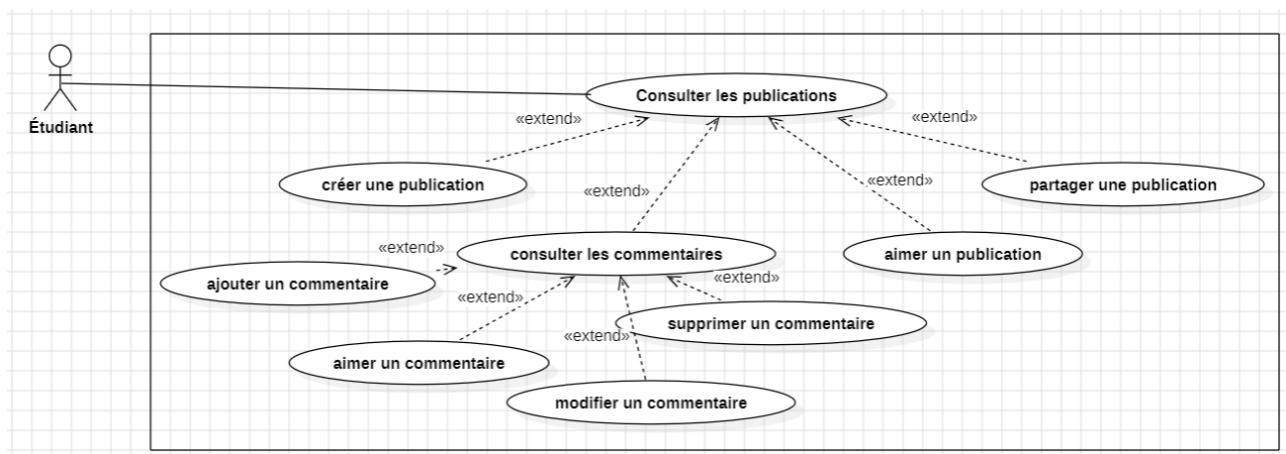


Figure 2. 3 : Le raffinement du cas d'utilisation « Consulter les publications »

- **Description textuelle :**

Le tableau 2.1 présente la description textuelle du cas d'utilisation « Consulter les publications ».

Tableau 2. 2 : Description textuelle du cas d'utilisation « Consulter les publications »

Titre :	Consulter les publications
Acteur :	Étudiant
Résumé :	C'est le cas d'utilisation qui permet à l'étudiant de consulter les publications
Préconditions :	Authentification
Scénario nominal :	1- L'étudiant demande de consulter les publications. 2- Le système affiche les publications déjà publiée par les étudiants.

	<p>3- L'étudiant demande de créer une publication.</p> <p>4- Le système affiche l'espace pour le saisir d'un texte et pour ajouter des documents.</p> <p>5- L'étudiant écrire le contenu de sa publication, ajouter des documents et demander de la publiée.</p> <p>6-le système publié la publication.</p>
Scénario alternatif :	<p>1-A l'étape 3 de scénario nominal l'étudiant peut demander de consulter la liste des commentaires d'une publication.</p> <ul style="list-style-type: none"> - Le système affiche la liste des commentaires - L'étudiant peut créer un nouveau commentaire, aimer un commentaire, modifier ou supprimer son commentaire <p>2- A l'étape 3 de scénario nominal l'étudiant peut sélectionne une publication pour la partager dans un groupe de message</p> <p>3- A l'étape 3 de scénario nominal l'étudiant peut sélectionne une publication pour l'ajouter à sa liste de favoris.</p> <p>4- A l'étape 3 de scénario nominal l'étudiant peut aimer une publication</p>

Conclusion

Ce chapitre a été dédié à la spécification et l'analyse des besoins du projet. Cette analyse nous a permis de mieux comprendre la problématique afin de fixer les différentes fonctionnalités à développer.

Chapitre 3

Conception

Introduction

Dans ce chapitre, nous explorons en profondeur la phase de conception de notre projet. Nous débutons par présenter le modèle architectural, offrant ainsi un aperçu de la structure fondamentale du système. Ensuite, nous nous plongeons dans les spécificités en détaillant les composants clés du système et leurs interactions. Cette analyse est complétée par des diagrammes de séquences, offrant une vision chronologique des échanges entre les composants du système. Nous mettons en relief l'importance cruciale de la phase de conception dans le processus de développement logiciel, soulignant son rôle dans l'établissement des bases nécessaires à une implémentation réussie.

1. Modèle architectural

Dans le cadre du développement de mon application, on a adopté le motif d'architecture MVC (Modèle-Vue-Contrôleur) pour garantir une structure logicielle claire et modulaire. Le modèle MVC (Modèle-Vue-Contrôleur) est un motif d'architecture logicielle largement utilisé dans le développement d'applications web et logicielles. Il divise une application en trois composants principaux, chacun ayant des responsabilités distinctes :

Modèle (Model) :

Le modèle représente les données et la logique métier de l'application. Il encapsule l'état de l'application et définit les opérations qui peuvent être effectuées sur ces données. Dans notre application Spring Boot, le modèle est souvent représenté par des classes d'entité JPA.

Vue (View) :

La vue est responsable de l'interface utilisateur de l'application. Elle affiche les données provenant du modèle et permet à l'utilisateur d'interagir avec l'application. Dans notre application web, les composants d'Angular jouent souvent le rôle de vues. Il est important de souligner que la vue est passive et ne doit pas contenir de logique métier. Elle se contente d'afficher les données fournies par le contrôleur ou le modèle.

Contrôleur (Controller) :

Le contrôleur agit comme un intermédiaire entre la vue et le modèle. Il reçoit les entrées de l'utilisateur via la vue, interagit avec le modèle pour effectuer les opérations nécessaires et met à jour la vue en conséquence. Dans notre application Spring Boot, les contrôleurs sont des

classes annotées avec `@RestController` qui traitent les requêtes HTTP et retournent les réponses appropriées.

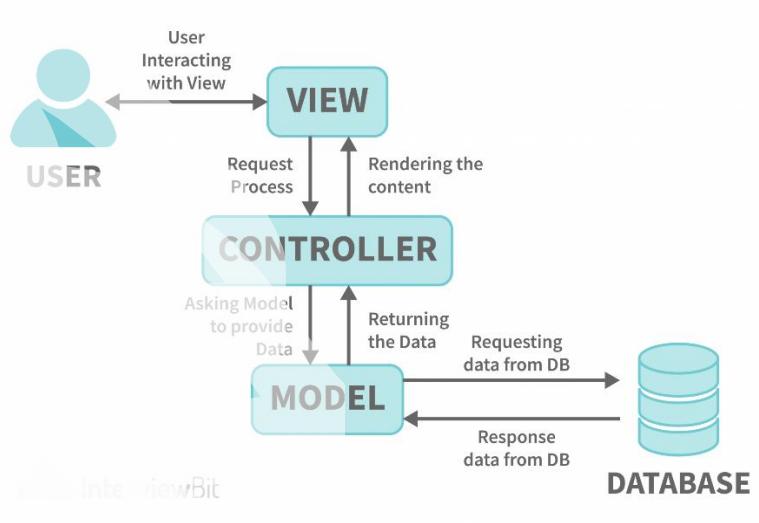


Figure 3. 1 : Les différents composants du modèle MVC[3]

2. Description de la base de données MongoDB

Ce document présente une description détaillée de la structure de la base de données MongoDB pour notre plateforme, mettant en évidence les entités clés et leurs relations.

Tableau 3. 1 : Description de la base des données

Nom	Structure du document	Description
User	{id:String, firstName: String, lastName: String, email:String, phone:String, birthday:Date, speciality:String, password:String, role:String, favorites>List<String> }	Contient des informations sur les utilisateurs de la plateforme.
Post	{id:String, description: String, userId: String, commentList: List<Comment>, documentUrl:String }	Contient des informations sur les publications créées par les utilisateurs.

Comment	{id:String, text: String, authorId: String}	Contient des informations sur les commentaires ajoutés aux publications.
Groupe	{id:String, name:String, pic:String, lastMessage:String }	Contient des informations sur les groupes créés par les utilisateurs
Message	{id:String, content:String, sender:String, groupId:String, timestamp:String, img:String }	Contient des informations sur les messages échangés au sein des groupes
Course	{id:String, title:String, description:String, subject:String, cours>List<String>, td>List<String>, tp>List<String>}	Contient des informations sur les cours proposés
Subject	{id:String, title:String, description:String), professor:String, image:String, speciality:String }	Contient des informations sur les matières enseignées

3. Diagrammes de séquence

Les diagrammes de séquence sont des outils de modélisation utilisés pour représenter les interactions entre les différents objets d'un système, ainsi que les messages qu'ils s'échangent pour accomplir leur tâche.

3.1 Diagramme de séquence du cas d'utilisation « Gérer groupe messages »

La Figure 3-2 présente en détail un diagramme de séquence du cas d'utilisation " Gérer groupe messages ".

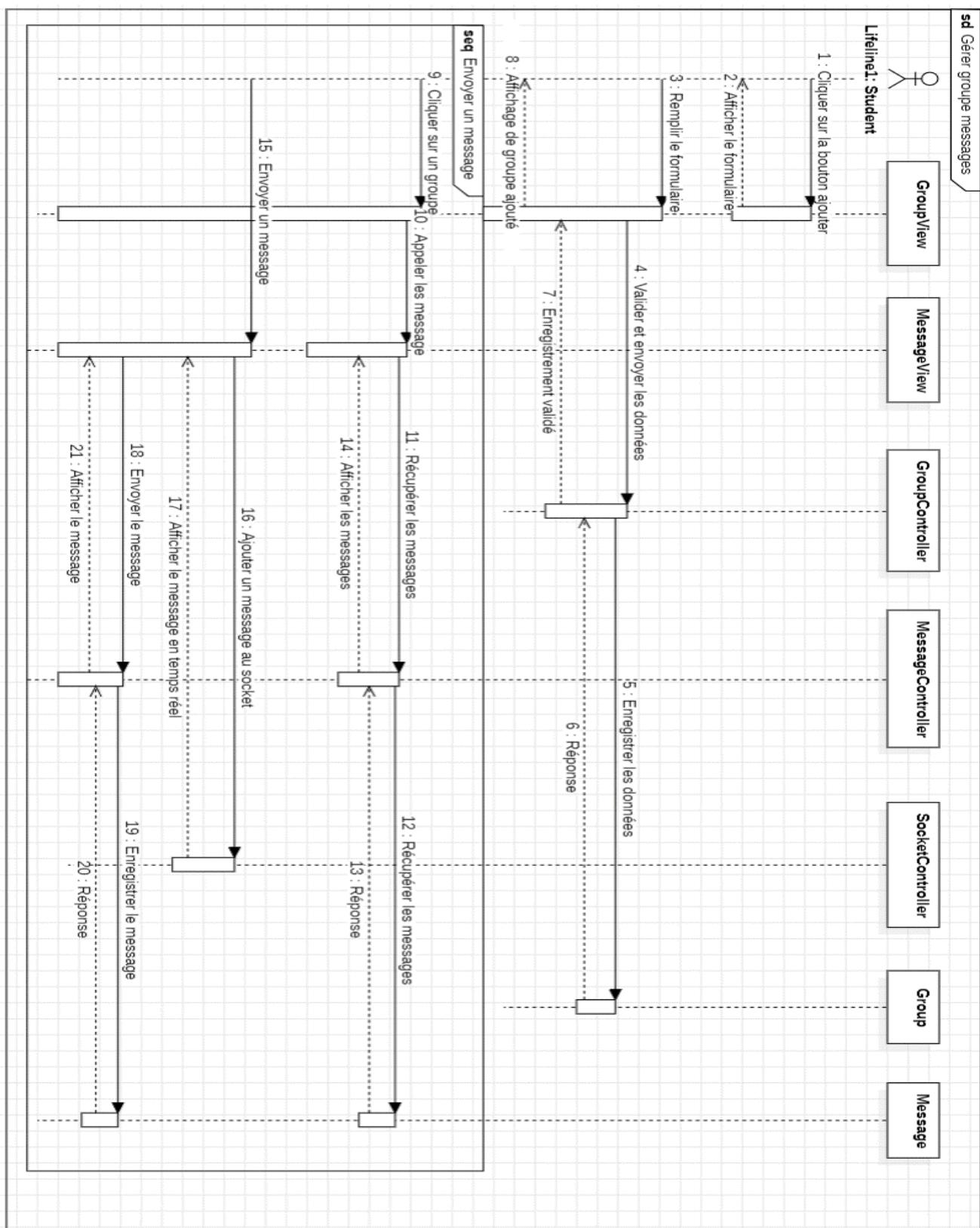


Figure 3. 2 : Le diagramme de séquence du cas d'utilisation « Gérer groupe messages »

3.2 Diagramme de séquence du cas d'utilisation « Passer en mode révision »

La Figure 3.3 présente en détail un diagramme de séquence du cas d'utilisation " Passer en mode révision ".

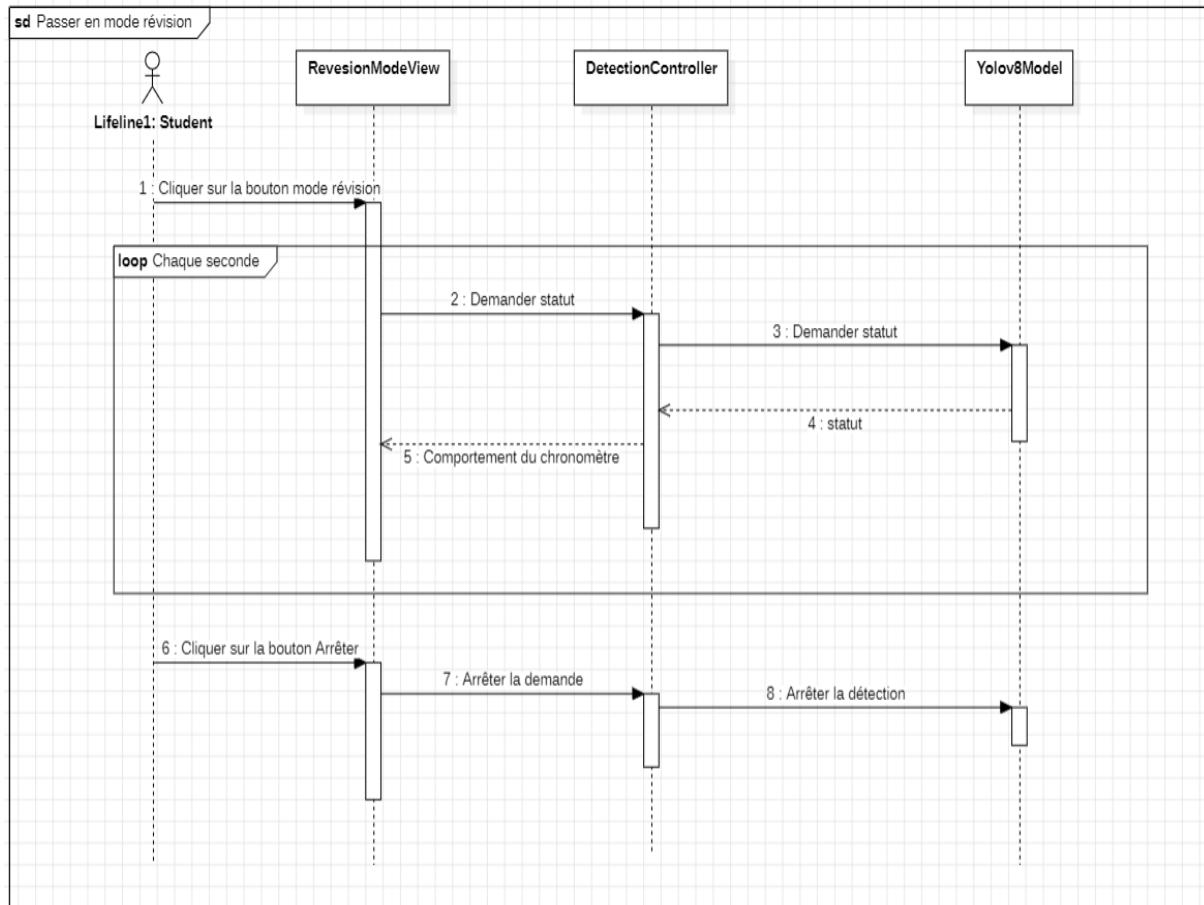


Figure 3. 3 : Le diagramme de séquence du cas d'utilisation « Passer en mode révision »

4. Conclusion

Dans cette section, nous avons approfondi l'étude conceptuelle du projet, une étape essentielle pour le bon fonctionnement de l'application. Dans le prochain chapitre, nous aborderons la mise en œuvre du projet.

Chapitre 4

Réalisation et mise en œuvre

Introduction

Dans ce chapitre nous allons présenter les différents outils utilisés pour réaliser notre plateforme, ainsi qu'un aperçu sur le travail réalisé à travers la description de quelques interfaces de l'application.

Chemin de navigation

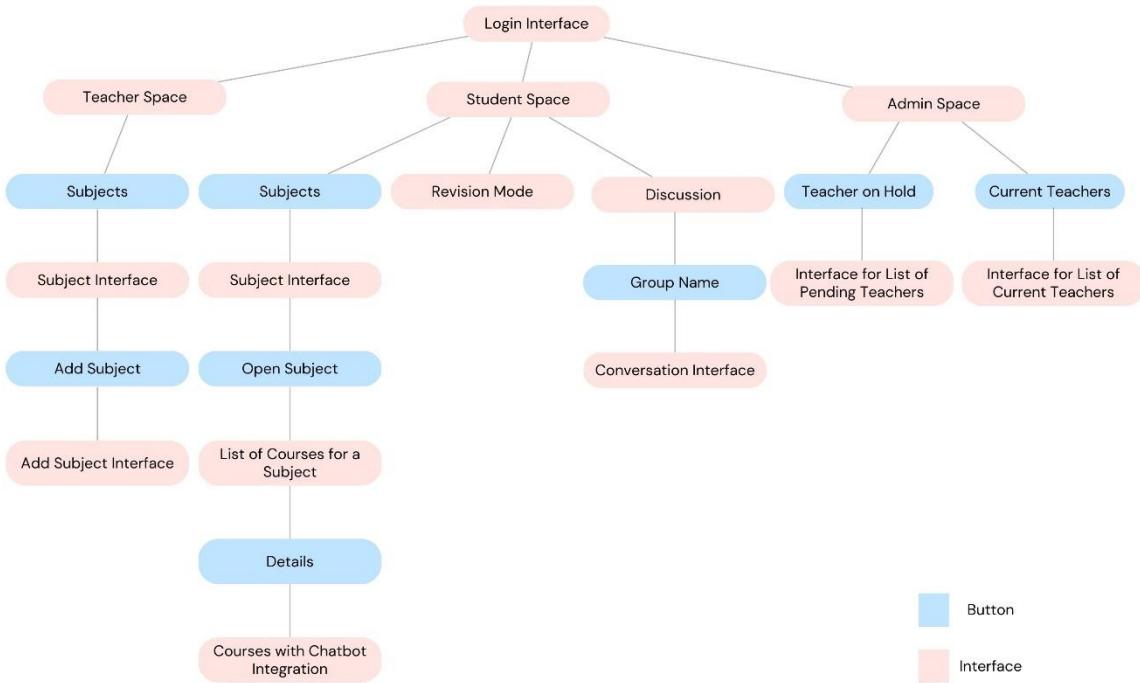


Figure 4. 1 Chemin de navigation

1. Environnement de travail

Dans cette partie, nous allons décrire l'environnement de travail matériel ainsi que logiciel utilisé.

1.1 Environnement matériel

Ce travail a été réalisé en utilisant des ordinateurs dont les caractéristiques sont décrites dans le tableau suivant.

Tableau 4. 1: Environnement matériel

Caractéristique	PC1	PC2	PC3	PC4
Processeur	Intel CORE i5	Intel CORE i5	Intel CORE i5	Intel CORE i5
RAM	8GO	16GO	16GO	8GO
Système d'exploitation	Windows 10 (64Bits)	Windows 11 (64Bits)	Windows 10 (64Bits)	Windows 10 (64Bits)

1.2 Environnement logiciel

Tout au long du projet nous avons utilisé un ensemble d'outils logiciels qui sont les suivants :

Tableau 4. 2 : Environnement logiciel

Logiciel/technologie	Description
 starUML	StarUML est un outil de modélisation UML puissant et convivial utilisé pour concevoir des diagrammes de classes, des diagrammes de séquence et d'autres modèles pour la conception logicielle. [4]
 mongoDB	MongoDB est un système de gestion de bases de données NoSQL orientée documents écrit en C++. Il permet de manipuler des objets structurés au format BSON sans schéma prédéterminé. Les données prennent la forme de documents enregistrés eux-mêmes dans des collections. Contrairement aux bases de données relationnelles, les champs d'un enregistrement sont libres et peuvent être différents d'un enregistrement à un autre au sein d'une même collection. [5]
 Visual Studio Code	Microsoft Visual Studio Code est un éditeur de code léger mais puissant qui fonctionne sur toutes les plateformes. Il prend en charge la plupart des langages connus comme JavaScript, TypeScript ou encore C++, Python, etc. Il dispose des fonctionnalités essentielles pour l'écriture du code comme

	indentation automatique, coloration syntaxique, auto complétion, code refactoring. etc. [6]
	Angular5 est une plateforme de développement qui permet de créer des applications web dynamiques en offrant une architecture basée sur des composants tout en bénéficiant des nouvelles fonctionnalités de typeScript.. [7]
	Un framework qui facilite le développement des applications fondées sur Spring en offrant des outils permettant d'obtenir une application packagée en jar , totalement autonome. Il a pour but de faciliter la configuration d'un projet Spring et de réduire le temps alloué au démarrage d'un projet. Spring est un projet OpenSource pour l'environnement Java, développé par Rod Jonhson et son utilité première était de remplacer les EJBs. Son infrastructure est similaire à un serveur d'application JavaEE mais l'avantage de Spring est que les classes n'ont pas besoin d'implémenter une interface pour être prises en charge. C'est en ce sens que Spring est qualifié de conteneur léger.[8]
 Python	Python est largement reconnu comme l'un des langages de programmation les plus polyvalents et conviviaux. Cependant, sa popularité ne se limite pas à sa flexibilité dans le développement de diverses applications. Il a également émergé comme un outil de choix dans le domaine en pleine expansion de l'intelligence artificielle (IA). [9]
 Postman	Postman est une plateforme de développement d'API qui permet aux développeurs de tester, de déboguer et de documenter les API de manière efficace, offrant des fonctionnalités telles que la création de requêtes HTTP, la gestion des environnements et la génération de rapports. [10]
 Figma	StarUML est un outil de modélisation UML puissant et convivial utilisé pour concevoir des diagrammes de classes, des diagrammes de séquence et d'autres modèles pour la conception logicielle. [11]

	Flask est un micro framework open-source de développement web en Python. Il est classé comme microframework car il est très léger. [12]
	Git est un système de contrôle de version distribué gratuit et open source conçu pour gérer tout, des petits aux très grands projets, avec rapidité et efficacité. [13]
	GitHub est une plateforme de développement collaboratif de logiciels qui permet aux équipes de travailler ensemble sur des projets, de gérer les versions du code source et de suivre les modifications apportées au code. [14]

2. Architecture utilisée

Dans notre application Nous avons utilisé une architecture à quatre serveurs, chaque serveur remplit un rôle spécifique dans l'ensemble de l'application.

Serveur Angular (Client Léger) :

Le serveur Angular agit comme le client léger dans votre architecture. Il fournit l'interface utilisateur de notre plate-forme, qui est rendue dans le navigateur web des utilisateurs. Il communique avec le serveur d'application (Spring Boot) pour récupérer les données nécessaires à l'affichage et envoyer les requêtes de l'utilisateur.

Serveur Spring Boot (Serveur d'Application) :

Le serveur Spring Boot est le cœur de notre application. Il contient la logique métier, traite les requêtes des clients et génère les réponses appropriées. Il interagit avec la base de données MongoDB pour stocker et récupérer les données nécessaires à l'application. Il

communique également avec le serveur Python pour des fonctionnalités spécifiques, comme le chatbot intelligent et la détection de présence.

Serveur MongoDB (Tiers de Données) :

Le serveur MongoDB est responsable du stockage des données de notre application. Il est interrogé par le serveur d'application (Spring Boot) via des requêtes HTTP pour récupérer les données nécessaires et mettre à jour les données en fonction des opérations effectuées par l'application.

Serveur Python (Fonctionnalités Spécifiques) :

Le serveur Python héberge des fonctionnalités spécifiques telles que le chatbot intelligent et la détection de présence. Il communique avec le serveur d'application (SpringBoot) pour échanger des données via des requêtes HTTP. Il fournit des services spécialisés qui complètent les fonctionnalités d'intelligence de notre application.

La Figure 4.2 présente L'architecture logiciel de notre application.

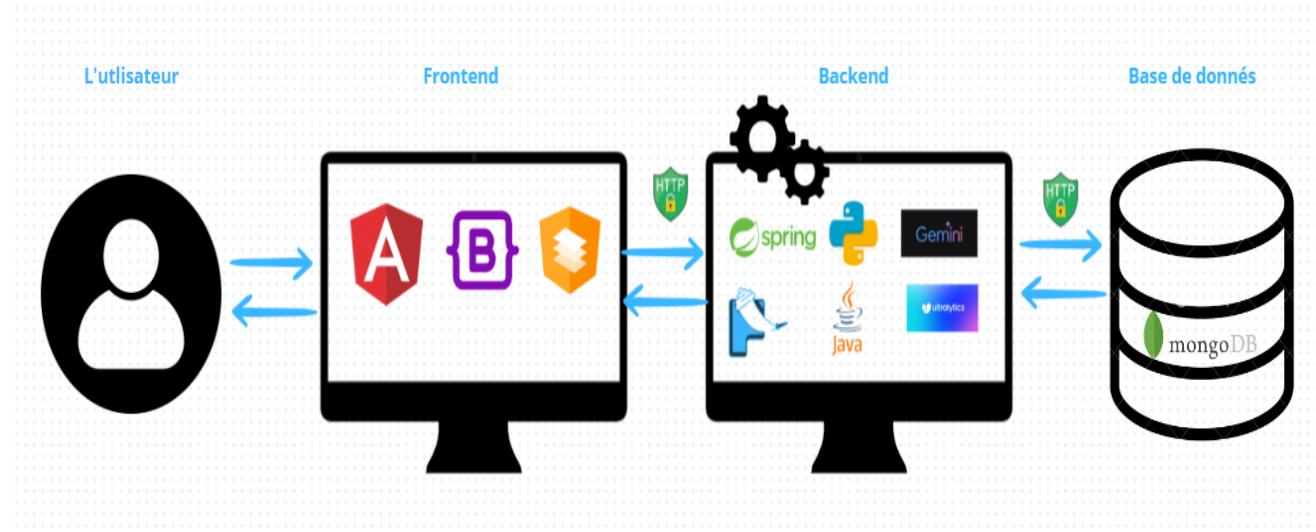


Figure 4. 2 Architecture de l'application

3. Interfaces réalisées

Nous allons présenter dans cette partie quelques interfaces de notre l'application "Study-App" ainsi les scénarios d'exécution pour montrer son fonctionnement.

3.1 Interface Signup

La figure 4.3 illustre l'interface permettant aux enseignants et aux étudiants de notre plateforme de s'inscrire et d'enregistrer leurs données dans la base de données.

The screenshot shows the 'Sign Up' interface. At the top, a blue header bar displays the text 'Sign Up'. Below the header, there are nine input fields arranged vertically. Each field has a label in bold black text followed by a white input box. The labels are: 'First Name', 'Last Name', 'Email', 'Password', 'Confirm Password', 'Phone Number', 'Birthday' (with a date input box showing 'jj / mm /aaaa' and a calendar icon), 'Role' (with a dropdown menu), and 'Speciality' (with a white input box). Below these fields is a section labeled 'Add Picture' containing a file input button ('Choisir un fichier') and a message ('Aucun fichier choisi'). At the bottom of the form is a large blue 'Sign Up' button. Below the button, a small note says 'Already have an account? [Log In](#)'.

Figure 4. 3 Interface Signup

3.2 Interface login

La figure 4.4 illustre l'interface à partir de laquelle tous les utilisateurs se permettent de s'authentifier à l'application.

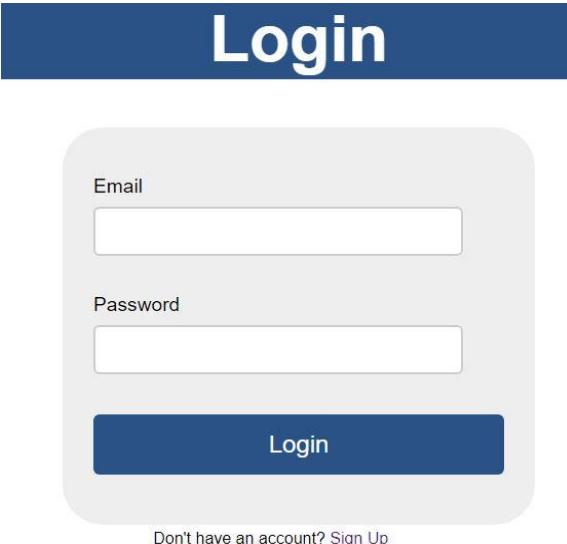


Figure 4. 4 Interface Login

3.3 Espace étudiant

3.3.1 La page d'accueil

L'interface « Page d'accueil » illustrée par la figure 4.5 offre à l'étudiant la possibilité de découvrir les publications récentes partagées par d'autres étudiants.

Figure 4. 5 Interface page d'accueil

3.3.2 Gestion d'une publication

Dans l'interface « Gestion publication » illustrée par la figure 4.6, l'étudiant a la possibilité d'ajouter une nouvelle publication, de la modifier ou de la supprimer.

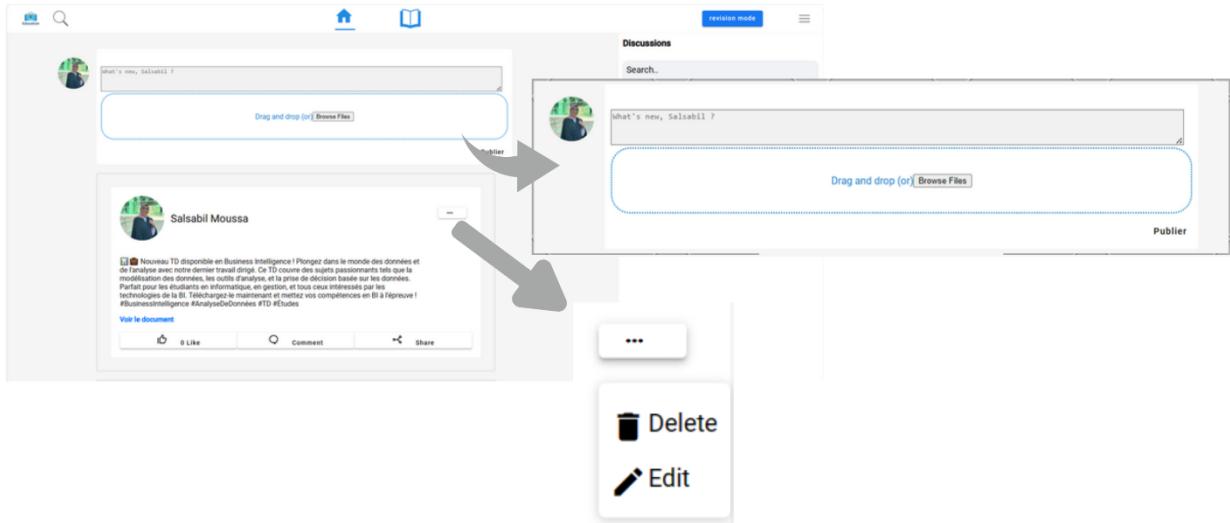


Figure 4. 6 Interface gestion publication

3.3.3 Poster des commentaires

Les étudiants peuvent également laisser des commentaires sur les publications, comme illustré dans la figure 4.7. Cela permet un échange interactif et une discussion autour des différents contenus partagés

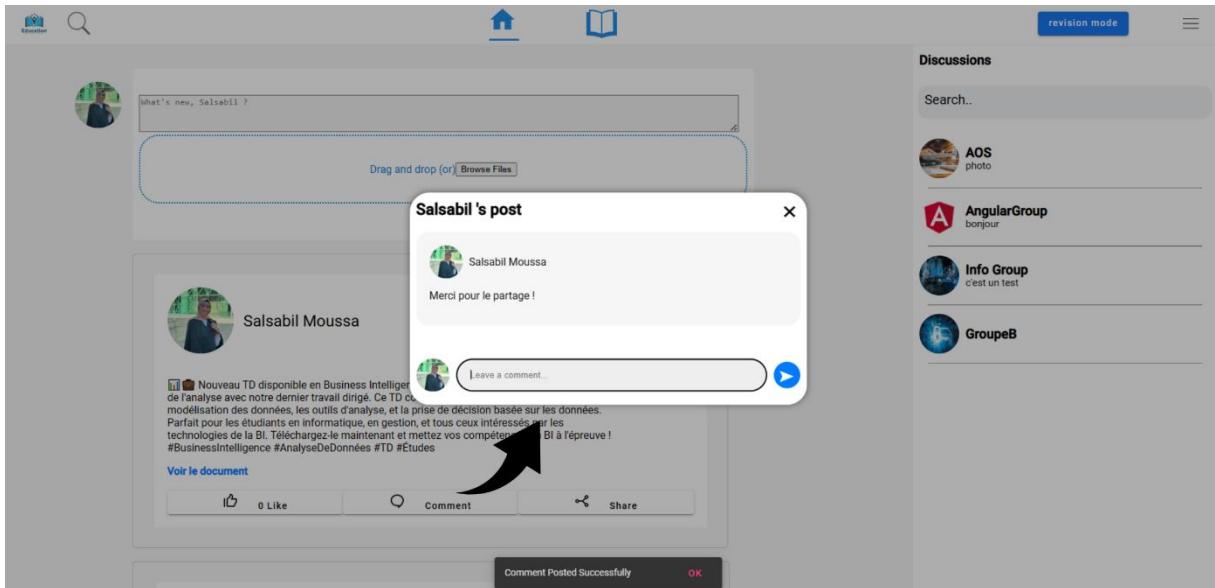


Figure 4. 7 Interface ajout des commentaires

3.3.4 Page des matières

Depuis l'interface des matières de notre StudyApp les étudiants peuvent explorer les matières spécifiques à leur spécialité, avec des noms de matières clairement indiqués accompagnés de descriptions informatives. Pour faciliter la recherche, une fonction de recherche permet aux étudiants de trouver rapidement des matières en saisissant des mots-clés tels que le nom de la matière ou celui du professeur. Une fois une matière sélectionnée, les étudiants peuvent accéder à une liste complète des cours disponibles dans cette matière.

La figure 4.8 illustre l'interface des matières.

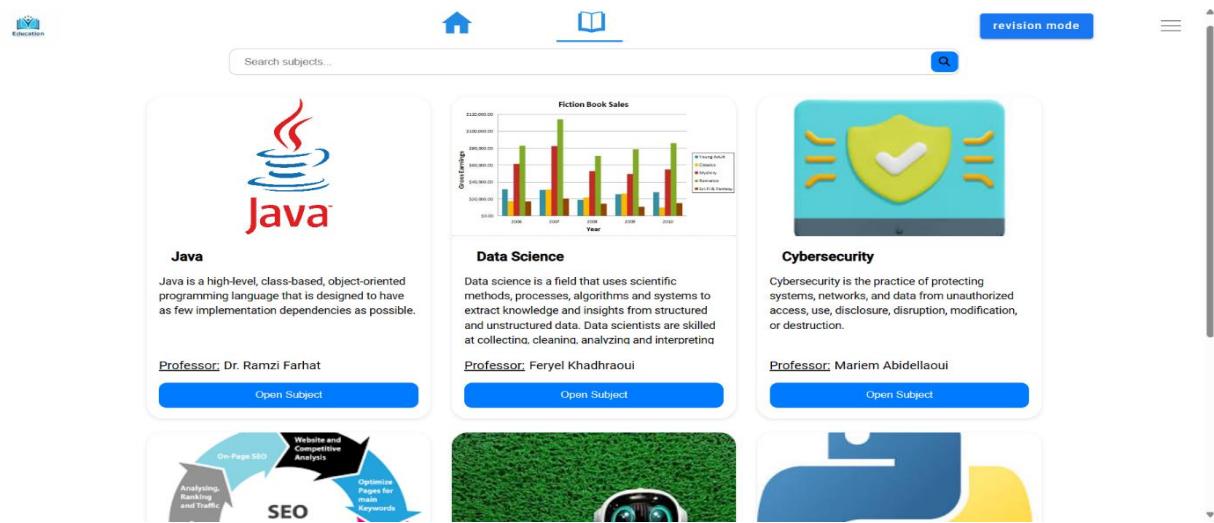


Figure 4. 8 Interface des matières

3.3.5 Liste des cours d'une matière

Après avoir sélectionné une matière, l'étudiant accède à la liste des cours associés à cette matière. Chaque cours est répertorié avec son nom et sa description. De plus, les leçons, travaux dirigés (TD) et travaux pratiques (TP) sont présentés dans une liste organisée. À partir de cette interface, l'étudiant peut ouvrir un document pour le lire, le télécharger ou l'ajouter à sa liste de favoris pour un accès ultérieur simplifié.

La figure 4.9 illustre l'interface des cours d'une matière.

The screenshot shows a digital learning environment. At the top, there's a navigation bar with icons for home, search, and revision mode. Below the navigation, a main content area displays a course titled "Chapter One: Introduction to Java Programming". This course page includes a brief introduction and a "LESSONS" section. The "LESSONS" section lists three items: "CHAPITRE 3 LES FONCTIONS EN C.pdf", "TD", and "TP". Each lesson item has a small icon, a download button, and a favorite button. To the right of the main content, there's a sidebar with a list titled "Object-Oriented Programming in Java".

Figure 4. 9 Liste des cours d'une matière

3.3.6 Ajouter un cours aux favoris

La figure 4.10 décrit le processus d'enregistrement d'un cours dans les favoris. Lorsque l'étudiant choisit d'ajouter un document à ses favoris à partir de l'interface des cours, une alerte de succès s'affiche pour confirmer que l'action a été effectuée avec succès. Cette alerte offre une confirmation visuelle immédiate à l'étudiant, lui indiquant que le document a été ajouté à sa liste de favoris avec succès.

This screenshot illustrates the process of adding a course to favorites. It shows two states of a course page for "Chapter One: Introduction to Java Programming". In the first state, a hand cursor is hovering over the favorite icon for the "CHAPITRE 3 LES FONCTIONS EN C.pdf" lesson. In the second state, a green success message "Added to favorites successfully" is displayed above the same lesson, indicating the action was successful. The rest of the interface remains consistent between the two states.

Figure 4. 10 Ajouter aux favoris

3.3.7 Télécharger un cours

L'action de téléchargement permet à l'étudiant de sauvegarder le contenu d'un document localement sur sa machine. En cliquant sur le bouton de téléchargement à partir de l'interface des cours, l'étudiant peut récupérer le contenu du cours et le stocker dans sa machine. La figure 4.11 illustre l'action de téléchargement du cours.

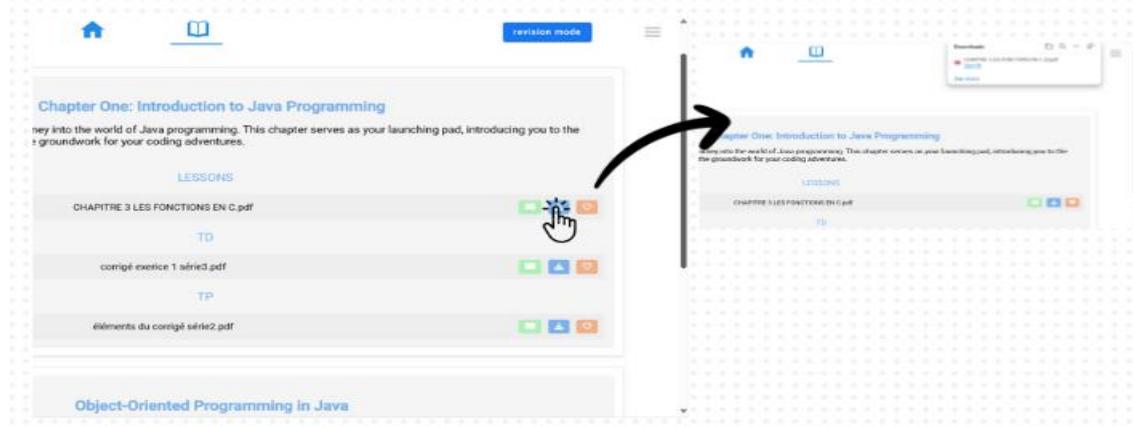


Figure 4. 11 Télécharger un cours

3.3.8 Affichage d'un cours

Lorsque l'étudiant clique sur le bouton de lire, le cours s'ouvre sous forme PDF, comme illustré dans la figure 4. 12. Ce format permet à l'étudiant de consulter le contenu du cours de manière claire et organisée.

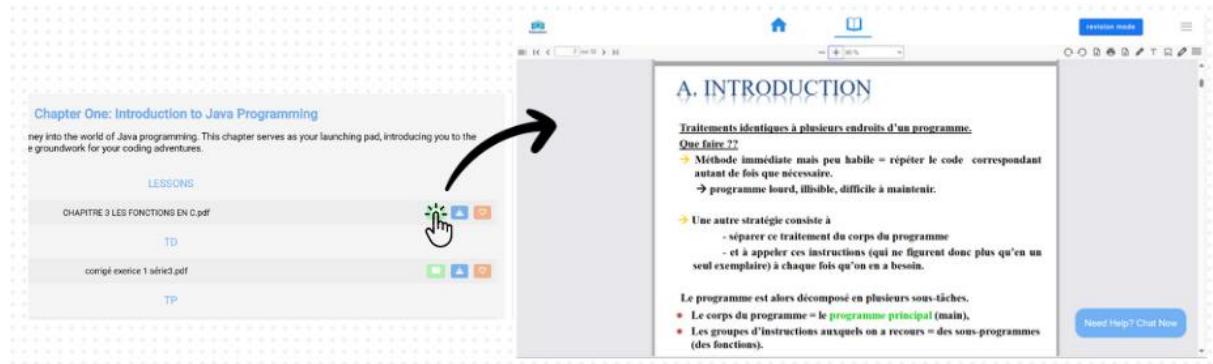


Figure 4. 12 Interface d'affichage d'un cours

3.3.9 Communication avec le Chatbot

En cliquant sur le bouton "Need Help ? Chat now", ils ouvrent une interface de chatbot intelligent où ils peuvent poser des questions sur n'importe quel aspect du cours. Cela leur offre

une solution rapide et personnalisée pour surmonter leurs difficultés d'apprentissage sans avoir à attendre ou à chercher des réponses ailleurs. Comme illustré par figure 4.13.

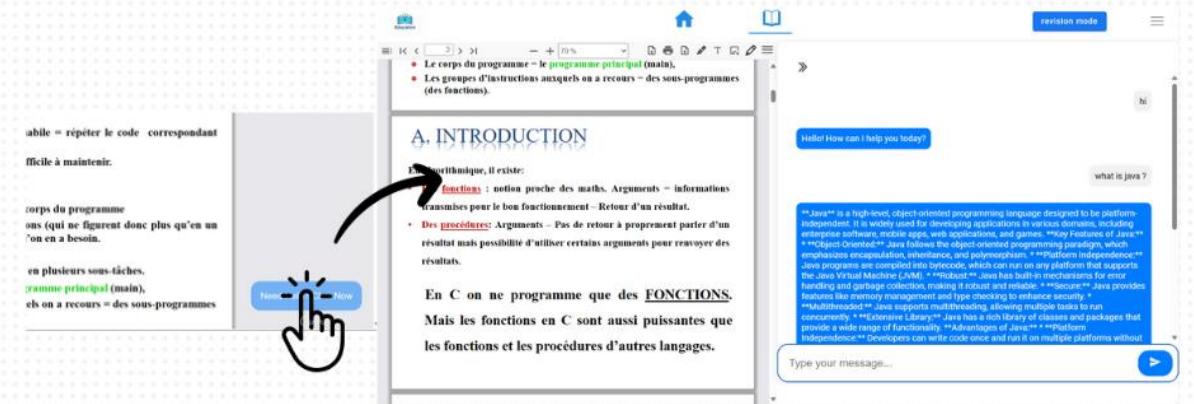


Figure 4. 13 Interface de chatboot intelligent

3.3.10 Groupe messagerie étudiant

La figure 4.14 présente l'interface de messageries aux groupes des étudiants où peuvent discuter ensemble et partager des fichiers.

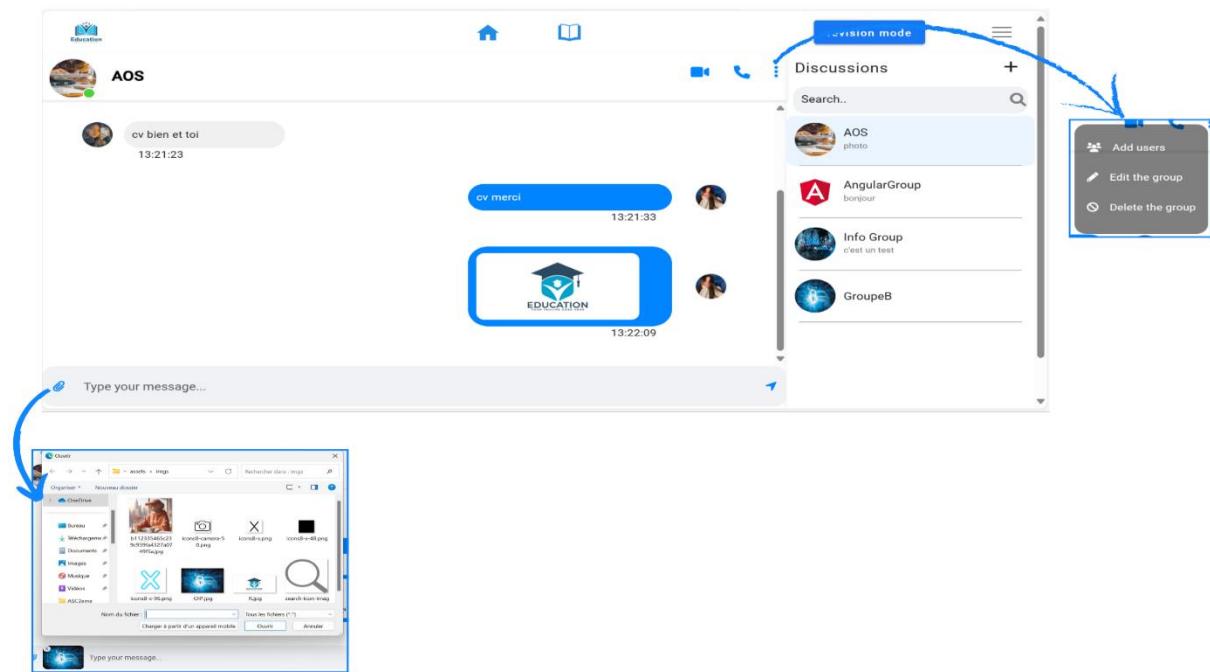


Figure 4. 14 Interface de la messagerie des étudiants

La figure 4.15 illustre les interfaces d'ajout et mis à jour d'un groupe.

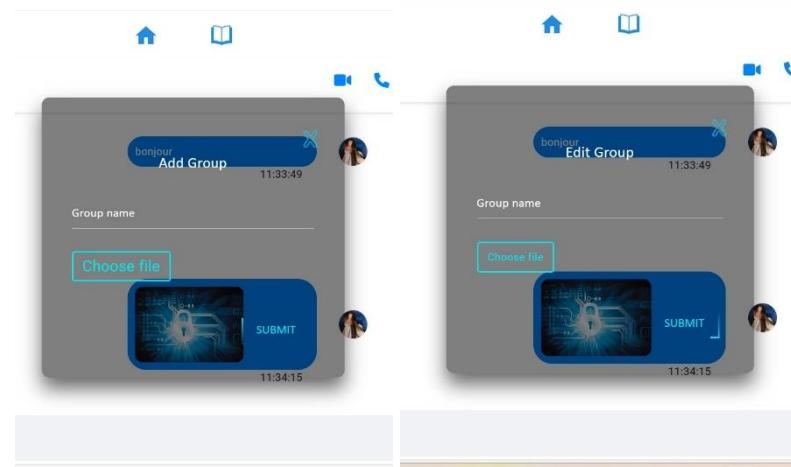


Figure 4. 15 Interfaces d'ajout et mis à jour du groupe

3.3.11 Mode révision

Lorsqu'un étudiant active le mode révision, il est dirigé vers l'interface « Mode révision activé », illustrée par la figure 4.16.

Dans cette interface, le chronomètre démarre automatiquement avec une session de 25 minutes de travail. L'étudiant a la possibilité de personnaliser le thème du chronomètre et de jouer de la musique pour accompagner sa session de révision. De plus, il peut quitter le mode de révision à tout moment en cliquant sur le bouton "exit revision mode".

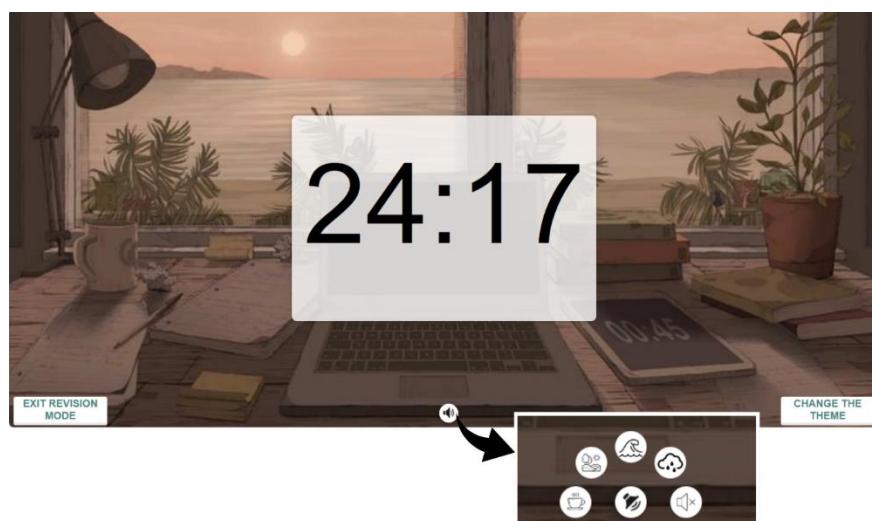


Figure 4. 16 Mode révision activé

Les deux interfaces illustrées par la figure 4.17 montrent le processus d'initialisation du chronomètre pour une pause, accompagné d'une notification pour indiquer à l'étudiant qu'il est temps de faire une pause ou de reprendre le travail.

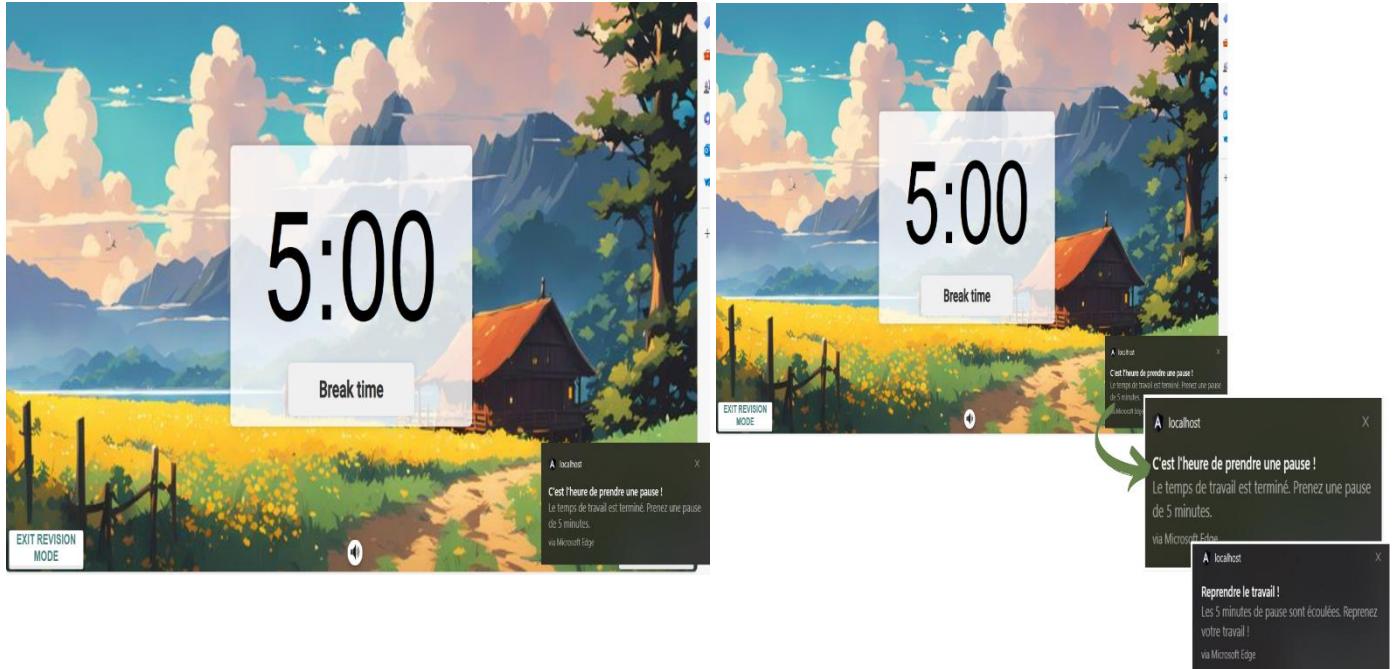


Figure 4. 17 Initialisation du chronomètre

3.4 Espace professeur

3.4.1 Ajouter une matière

L'interface d'ajout d'une nouvelle matière offre aux professeurs la possibilité de saisir le nom de la matière, une description et télécharger une photo représentative. Une fois soumis, les détails de la matière sont enregistrés dans la base de données. La figure 14.18 illustre cette fonctionnalité.

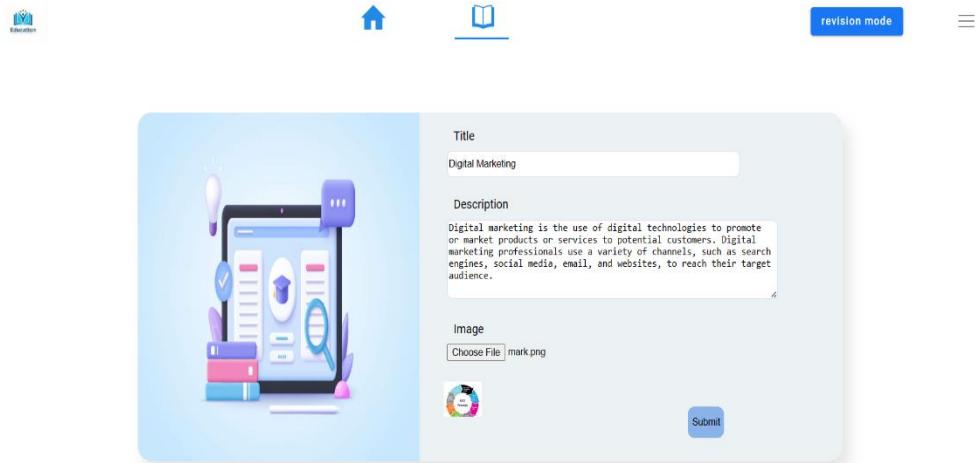


Figure 4.18 Interface d'ajout de matière

3.4.2 Ajouter un cours

La figure 4.19 présente l'interface d'ajout d'un nouveau cours, elle permet aux professeurs de saisir le titre du cours, une description détaillée, ainsi que d'ajouter des documents pour les leçons, les travaux dirigés (TD) et les travaux pratiques (TP). Une fois ces détails saisis, le professeur peut soumettre les informations, qui seront ensuite enregistrées dans la base de données.

Figure 4. 19 Interface d'ajout du cours

3.4.3 Liste des matières

L'interface "Liste des matières" illustrée par la figure 4.20 offre aux professeurs une vue pratique et organisée de leurs matières existantes, tout en leur permettant d'en ajouter de nouvelles. Chaque professeur peut facilement gérer ses propres matières grâce à des options claires telles que la modification ou la suppression. De plus, cette interface permet un accès rapide aux listes de cours associées à chaque matière, offrant ainsi aux professeurs la

The screenshot shows a user interface for managing subjects. At the top, there are navigation icons for Home, Bookmarks, and Revision mode, along with a search bar labeled "Search subjects..." and a magnifying glass icon. Below the search bar is a green button labeled "+ Add New Subject". The main content area displays two subjects in cards:

- Digital Marketing**: A circular icon featuring a globe and the words "SEO Process". A brief description states: "Digital marketing is the use of digital technologies to promote or market products or services to potential customers. Digital marketing professionals use a variety of channels, such as search engines, social media, email, and websites, to reach their target audience." To the right are three small icons: a blue square with a checkmark, a green square with a video camera, and a red square with a trash can.
- Artificial intelligence (AI)**: An icon of a white robot head. A brief description states: "Artificial intelligence (AI) is a broad and rapidly evolving field that focuses on creating machines capable of simulating human intelligence." To the right are three small icons: a blue square with a checkmark, a green square with a video camera, and a red square with a trash can.

Figure 4. 20 Liste des matières

3.5 Espace admin

3.5.1 Interface d'approbation des professeurs

Lorsqu'un professeur s'inscrit sur la plateforme, ses coordonnées seront automatiquement transmises à l'administrateur du système. L'administrateur aura alors la possibilité d'examiner les informations fournies par le professeur et de décider s'il approuve ou non son inscription. La figure 4.21 montre le processus d'approbation par l'administrateur ce qui assure le contrôle de qualité et une sécurité renforcée de la plateforme.

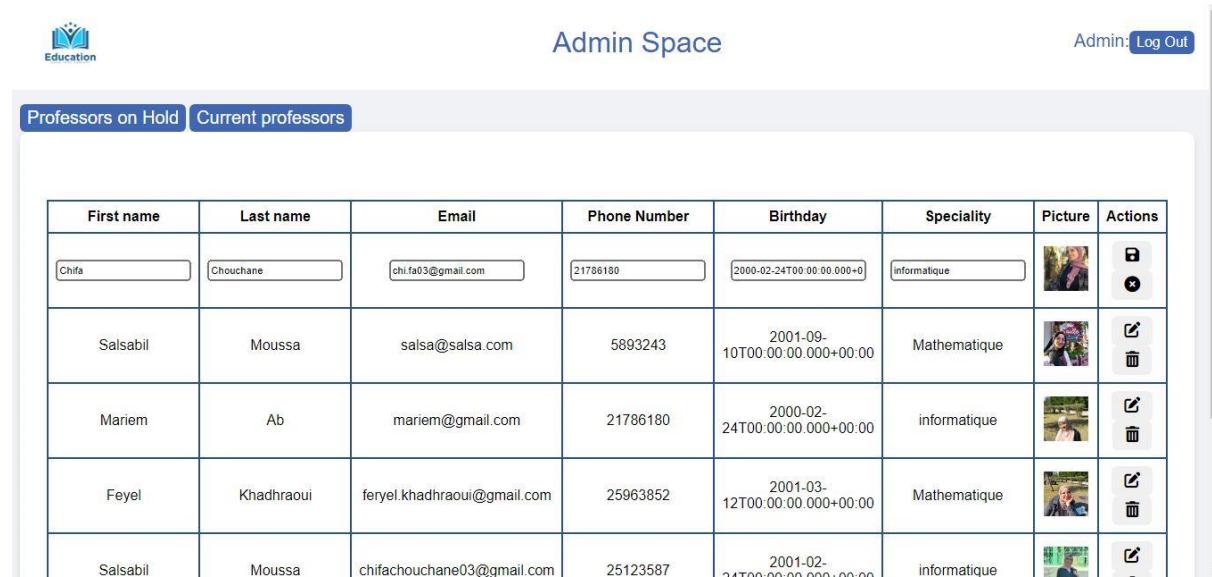
The screenshot shows the "Admin Space" interface. At the top, there are tabs for "Professors on Hold" and "Current professors", with "Professors on Hold" being active. In the top right corner, it says "Admin: Log Out". The main content area is a table titled "Admin Space" listing three professors:

First name	Last name	Email	Phone Number	Birthday	Speciality	Picture	Actions
Mariem	Abidellaoui	mariem.abidellaoui@gmail.com	50214368	2001-05-05T00:00:00.000+00:00	informatique		<input checked="" type="checkbox"/>
Chifa	Chouchane	chifachouchane03@gmail.com	52148367	2000-02-20T00:00:00.000+00:00	Mathematique		<input checked="" type="checkbox"/>
Feryel	Khadhraoui	fery.khadhraoui@gmail.com	50236485	2001-05-12T00:00:00.000+00:00	Mathematique		<input checked="" type="checkbox"/>

Figure 4. 21 Interface d'approbation des professeurs

3.5.2 Interface de gestion des professeurs actuels dans la plateforme

La figure 4.22 montre la liste des professeurs actuels offrant une vue organisée et conviviale des enseignants en poste, facilitant ainsi la gestion de leur profil. Ce qui permet à l'admin de gérer les informations des enseignants grâce à des options claires telles que la modification ou la suppression de leurs profils.



The screenshot shows a web-based administration interface titled "Admin Space". At the top right, there is a "Log Out" button. Below the title, there are two tabs: "Professors on Hold" and "Current professors", with "Current professors" being the active tab. The main area displays a table with the following data:

First name	Last name	Email	Phone Number	Birthday	Speciality	Picture	Actions
Chifa	Chouchane	chi фа03@gmail.com	21786180	2000-02-24T00:00:00.000+00:00	informatique		
Salsabil	Moussa	salsa@salsa.com	5893243	2001-09-10T00:00:00.000+00:00	Mathematique		
Mariem	Ab	mariem@gmail.com	21786180	2000-02-24T00:00:00.000+00:00	informatique		
Feyel	Khadhraoui	feyel.khadhraoui@gmail.com	25963852	2001-03-12T00:00:00.000+00:00	Mathematique		
Salsabil	Moussa	chifachouchane03@gmail.com	25123587	2001-02-24T00:00:00.000+00:00	informatique		

Figure 4. 22 Interface de gestion de professeurs

4. Conclusion

En résumé, ce chapitre a présenté les outils, l'environnement de travail et les interfaces clés de notre plateforme "Study-App". Grâce à une architecture robuste et des fonctionnalités innovantes, notre application offre une expérience d'apprentissage en ligne intuitive et enrichissante pour les étudiants, les enseignants et les administrateurs.

Chapitre 5 :

Réalisation du
module
Chatbot et
Chronomètre
Intelligent

Introduction

L'intégration de l'intelligence artificielle (IA) dans notre plateforme de révision marque une étape importante dans notre engagement à fournir des solutions éducatives innovantes. En introduisant deux composants clés : un chatbot intelligent et un chronomètre intelligent basé sur un modèle de détection d'objets. Nous visons à améliorer l'expérience d'apprentissage des étudiants de manière significative. Dans ce chapitre, nous examinerons de près chaque composant en explorant leurs fonctionnalités, leur intégration dans notre plateforme de révision, et leur impact sur l'expérience d'apprentissage des étudiants.

1. Chatbot Intelligent

Le chatbot intégré dans notre plateforme de révision est un outil interactif conçu pour aider les étudiants dans leur processus d'apprentissage. Il utilise l'API Gemini de Google, un modèle de langage large (large language model LLM), pour répondre aux questions des étudiants sur le contenu des cours. Grâce à ses fonctionnalités intelligentes, il fournit des explications claires et des réponses pertinentes pour soutenir les étudiants dans leur compréhension des cours.

1.1 fonctionnalités de chatbot

Nous avons intégré à notre chatbot un modèle de langage large (LLM). Ce modèle, un type d'intelligence artificielle, est entraîné sur de vastes ensembles de données textuelles afin de comprendre et de générer du langage naturel. Les LLM sont capables d'exécuter diverses tâches, telles que la génération de texte, la traduction automatique et la réponse à des questions. Grâce à l'exposition à de grandes quantités de données, ces modèles acquièrent une compréhension approfondie du langage humain.

En utilisant ce LLM, notre chatbot offre plusieurs fonctionnalités pour soutenir les étudiants dans leur apprentissage. Il répond aux questions des étudiants de manière instantanée et précises. De plus, le chatbot fournit des explications claires et pertinentes sur les sujets abordés dans les cours, simplifiant ainsi les concepts complexes et aidant les étudiants à mieux comprendre le contenu étudié. Il offre également une guidance et une assistance personnalisées en recommandant des ressources supplémentaires, des exercices pratiques ou des exemples

adaptés aux besoins spécifiques de chaque étudiant. L'interaction avec le chatbot est intuitive et conviviale, permettant aux étudiants de poser des questions sous forme de texte naturel et de recevoir des réponses de manière fluide et efficace. Enfin, le chatbot est accessible à tout moment pour répondre aux besoins des étudiants et améliorer leur expérience d'apprentissage.

1.2 Conception du Chatbot

Pour concevoir le chatbot, nous avons suivi un processus simple et efficace. Lorsque l'étudiant pose une question dans l'interface utilisateur frontend d'Angular, celle-ci envoie une requête HTTP à notre application backend Flask. Cette requête contient la question de l'étudiant sous forme de données JSON. Ensuite, notre application Flask reçoit cette requête et extrait la question de l'étudiant. En utilisant l'API Gemini, notre application génère une réponse appropriée à la question. Enfin, la réponse est renvoyée au frontend Angular via une réponse HTTP, où elle est affichée à l'étudiant. Ce processus assure une interaction fluide et rapide entre l'étudiant et le chatbot, améliorant ainsi son expérience d'apprentissage.

La figure 1.3 illustre la conception du chatbot.

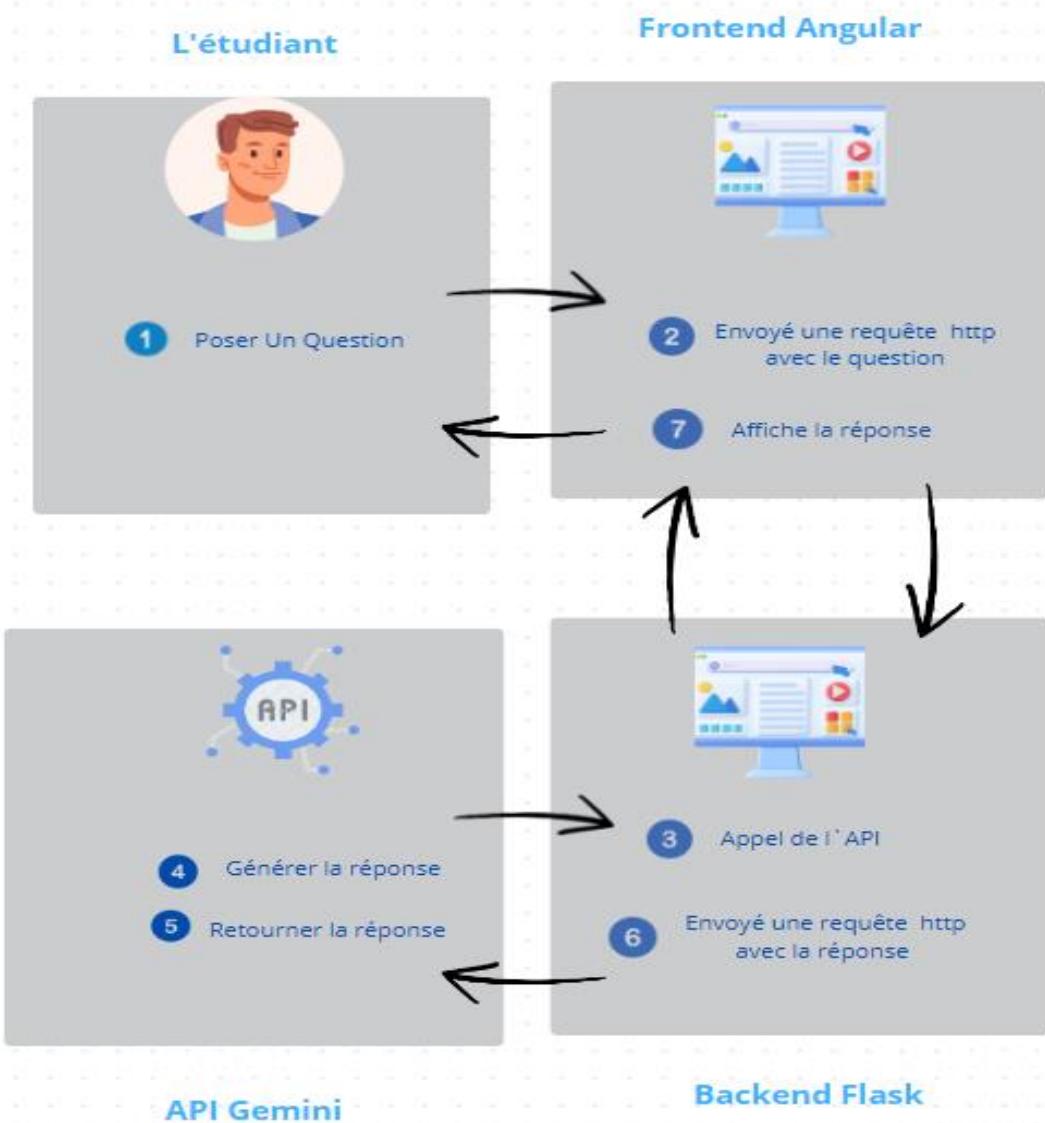


Figure 5. 1 Conception du chatbot

1.3 Intégration dans la Plateforme de Révision

Pour intégrer l'API Gemini, nous avons développé une application Flask, en suivant quelques étapes simples. Tout d'abord, nous avons installé la bibliothèque `google.generativeai` qui nous permet d'interagir avec l'API Gemini. Ensuite, nous avons configuré notre application Flask pour utiliser cette bibliothèque en fournissant notre clé API Gemini que nous avons obtenue depuis la plateforme Gemini. Cela garantit que notre application peut communiquer avec l'API Gemini et générer du contenu de manière appropriée.

La figure 1.3 illustre l'intégration de l'API dans L'application Flask.

```

1 Début
2 Importer le module Flask depuis flask
3 Importer le module GenerativeAI de google.generativeai
4
5 Configurer le générateur AI avec la clé API fournie
6
7 Définir une application Flask
8
9 Fonction generate_content():
10 Extraire le contenu JSON de la requête POST
11 Initialiser un modèle de génération AI avec le nom 'gemini-pro'
12 Générer du contenu en utilisant le modèle |
13 Retourner le contenu généré sous forme de réponse JSON
14 Fin

```

Figure 5. 2 Intégration de l'API

2. Chronomètre intelligent

Le chronomètre intelligent, qui utilise le modèle de détection d'objet YOLOv8, est intégré à notre plateforme de révision. Cet outil interactif est spécialement conçu pour accompagner les étudiants dans leur apprentissage, en s'inspirant de la méthode pomodoro. Il permet aux étudiants de gérer efficacement leur temps d'étude et leurs pauses.

2.1 Fonctionnalités du chronomètre

Nous avons intégré dans notre chronomètre le modèle YOLOv8, un algorithme de détection d'objets puissant développé par Ultralytics, reconnu pour sa rapidité et sa précision. Grâce à ce modèle, notre chronomètre est capable de détecter la présence de l'étudiant chaque seconde pendant son fonctionnement. Lorsqu'il détecte la présence, le chronomètre continue de compter le temps. En revanche, en cas d'absence, il se met automatiquement en pause pour éviter de gaspiller des minutes inutiles. Ainsi, les étudiants n'ont pas besoin de mettre le chronomètre en pause manuellement, ce qui simplifie leur expérience.

De plus, notre chronomètre utilise la méthode Pomodoro. Lorsqu'un étudiant active le mode révision, le chronomètre démarre avec une session de 25 minutes de travail, suivie de 5 minutes de pause, une notification invite alors l'étudiant à prendre une pause. Ce cycle se répète tant que

le mode est activé, alternant entre sessions de travail et pauses. Cette approche vise à maintenir la concentration et la productivité tout en évitant la fatigue.

2.2 Conception du chronomètre

Le processus débute avec l'activation du mode révision dans l'interface Angular. À chaque seconde, Angular envoie une requête au backend Spring Boot pour signaler l'activation du mode révision. Spring Boot, à son tour, envoie une requête à l'application Flask. Il capture une image de la vidéo en cours, la transmet au modèle YOLO pour la détection d'objets, puis envoie la réponse à Spring Boot. Si une personne est détectée, Spring Boot renvoie cette information à Angular, qui maintient le fonctionnement du chronomètre. Dans le cas contraire, Angular met en pause le chronomètre.

La figure 5. 3 illustre la conception du chronomètre.

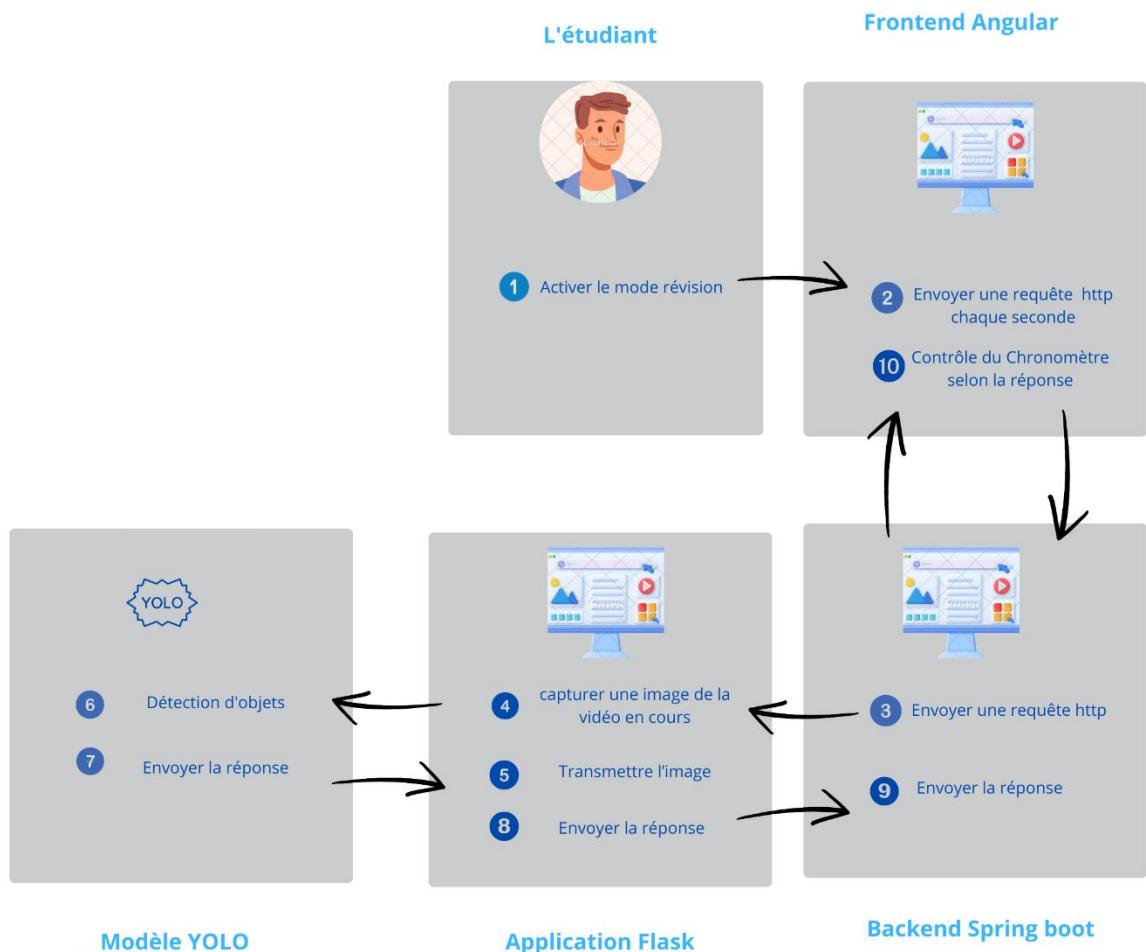


Figure 5. 4 Conception du chronomètre

3.3 Intégration du modèle YOLOv8 dans l'application

Pour intégrer le modèle YOLOv8, nous avons développé une application Flask, en suivant quelques étapes simples. Tout d'abord, nous avons importé les bibliothèques nécessaires, notamment Flask pour la création de l'API Web, OpenCV pour la manipulation d'images, et YOLO depuis la bibliothèque Ultralytics pour la détection d'objets.

Ensuite, nous avons chargé le modèle YOLOv8 pré-entraîné à partir d'un fichier. Ce modèle est capable de détecter divers objets dans les images.

Nous avons défini les noms de classe pour les objets que le modèle peut détecter, ce qui nous permettra d'interpréter les résultats de la détection.

Ensuite, nous avons créé une route API /détection pour effectuer la détection d'objets en temps réel. À chaque appel de cette route, notre application ouvre une capture vidéo depuis la webcam, lit une image à partir de cette capture, puis passe cette image au modèle YOLOv8 pour la détection d'objets.

Après avoir obtenu les résultats de détection, nous vérifions si une personne est détectée dans l'image en analysant les classes des boîtes englobantes détectées. Si une personne est détectée, nous retournons "oui", sinon nous retournons "non".

La figure 5. 5 illustre l'intégration du modèle YOLOv8 dans L'application Flask.

Début

Importer les bibliothèques Flask, OpenCV et YOLO
Initialiser une application Flask

Charger le modèle YOLO pré-entraîné depuis un fichier

Définir les noms de classe pour les objets détectables

Définir une route API '/detection' pour la détection d'objets:

- Ouvrir une capture vidéo depuis la webcam
- Lire une image à partir de la capture vidéo
- Utiliser le modèle YOLO pour détecter les objets dans l'image
- Vérifier si la personne est détectée dans l'image
- Libérer la capture vidéo après la détection

Si la personne est détectée:

- Retourner "oui"

Sinon:

- Retourner "non"

Fin

Figure 5. 6 Intégration du modèle YOLOv8

Conclusion

Dans ce chapitre, nous avons exploré l'intégration de l'intelligence artificielle dans notre plateforme de révision, en mettant en évidence leurs fonctionnalités.

Conclusion générale

Les révolutions technologiques dans le domaine de l'informatique ont rendu possible le développement de nouveaux services offrant une valeur ajoutée considérable par rapport à certaines méthodes traditionnelles, notamment en matière d'apprentissage et de révision. C'est dans ce contexte que s'inscrit notre projet de fin d'année, qui vise à développer et à mettre en place une plateforme de révision intelligente répondant aux besoins évolutifs des étudiants.

Pour ce faire, nous avons entamé par une étude générale du projet afin de décrire et de comprendre les principaux concepts sur lesquels repose notre solution. Ensuite, nous sommes passés à la phase d'analyse et de spécification des besoins, mettant l'accent sur les exigences auxquelles notre système devrait répondre. En se basant sur ces besoins, nous avons procédé à la conception, puis à une étude technique détaillée, présentant l'environnement de travail et les technologies utilisées, pour enfin réaliser notre application.

Ce travail nous a été très instructif en termes de connaissances acquises. Il nous a offert l'opportunité de renforcer nos compétences dans des technologies telles que Angular, Spring Boot, et surtout, d'explorer de nouveaux outils dans le domaine de l'intelligence artificielle, tels que l'intégration des API de Gemini (chatbot intelligent) et de Yolov8 pour la détection de présence (chronomètre intelligent). De plus, il nous a permis de travailler en équipe, ce qui a joué un rôle essentiel dans l'amélioration de nos compétences relationnelles.

En outre, l'application présentée dans le rapport constitue seulement une première version. Notre solution présente plusieurs points d'amélioration et de finalisation qui seront traités prochainement. Parmi ces améliorations, nous envisageons la création d'une interface "study with me", permettant aux utilisateurs de se connecter en ligne et d'étudier ensemble, même à distance, dans un environnement collaboratif. De plus, l'intégration d'une fonctionnalité de vidéoconférence pourrait enrichir davantage cette expérience d'apprentissage. En ce qui concerne la partie de la révision, il serait bénéfique d'ajouter une fonctionnalité permettant aux étudiants de consulter l'historique de leurs activités tout au long de leur période de révision, en leur fournissant des statistiques sur leur productivité.

REFERENCES

- [1]: <https://fr.khanacademy.org/> (consulté le 05/04/2024)
- [2]: <https://www.coursera.org/> (consulté le 05/04/2024)
- [3]:<https://fr.khanacademy.org/>(consulté le 05/04/2024)
- [4]<https://www.lucidchart.com/pages/fr/langage-uml> (consulté le 15/04/2024)
- [5]<https://blog.erlem.fr/programmation/36-nosql/154-nosql-mongodb> (Consulté le 1 Mai 2024)
- [6] <https://www.sodifrance.fr/blog/visual-studio-code-quesaco/> (consulté le 15/04/2024)
- [7] <https://www.supinfo.com/articles/single/4968-angular2> (consulté le 15/04/2024)
- [8][https://www.supinfo.com/articles/single/362-springframework.](https://www.supinfo.com/articles/single/362-springframework) (Consulté le 15/04/2024)
- [9] <https://www.pythoniaformation.com/blog/articles-sur-python/culture-data/python-ia> (consulté le 15/04/2024)
- [10] <https://www.postman.com/>, (consulté le 15/04/2024)
- [11] <https://www.figma.com/>, (consulté le 15/04/2024)
- [12]<https://www.jedha.co/formation-python/framework-python-flask#:~:text=Flask%20sous%20Python%20est%20consid%C3%A9r%C3%A9,%20bo%C3%A9te%20%C3%A0%20outils%20Werkzeug.> (Consulté 15/05/2024)
- [13] <https://git-scm.com/> (consulté le 15/04/2024)
- [14] <https://github.com/> (consulté le 29/04/2024)

Résumé

Ce rapport décrit notre projet de fin d'année en deuxième année à l'École Nationale Supérieure d'Ingénieurs de Tunis (ENSIT). Il consiste à développer une plateforme de révision intelligente qui facilitera la révision et les échanges entre les étudiants. Pour sa réalisation, nous avons utilisé les technologies suivantes : Framework Angular pour le frontend, Framework SpringBoot pour le backend, et pour la partie intelligence artificielle on a utilisé le Framework Flask pour intégrer les API de Gemini (chatbot intelligent) et Yolov8 pour la détection de présence (chronomètre intelligent).

Mots clés : plateforme de révision intelligente, chronomètre, détection de présence, Angular, SpringBoot, Flask, API Gemini, Yolov8.

Abstract

This report describes our end-of-year project in the second year at the National School of Engineers of Tunis (ENSIT). It involves developing an intelligent revision platform that will facilitate revision and communication among students. For its implementation, we used the following technologies : Angular framework for the frontend, Spring Boot framework for the backend, and for the artificial intelligence part, we integrated the APIs of Gemini (intelligent chatbot) and Yolov8 for presence detection (intelligent timer).

Keyword: Intelligent revision platform, progress tracking, timer, and presence detection, Angular, SpringBoot, Flask, Gemini API, Yolov8.

الملخص

يصف هذا التقرير مشروعنا للسنة الثانية في المدرسة الوطنية العليا للمهندسين بتونس (ENSIT). يتمثل هذا المشروع في تطوير منصة للمراجعة الذكية التي تسهل عملية المراجعة والتبادل بين الطلاب. لتحقيق ذلك، استخدمنا التقنيات التالية: إطار عمل Angular للواجهة الأمامية، إطار عمل SpringBoot للواجهة الخلفية، وللجزء الذكاء الاصطناعي، استخدمنا إطار Flask لدمج واجهات برمجة التطبيقات API Gemini (الدردشة الذكية) و Yolov8 لاكتشاف الحضور (المؤقت الذكي).

كلمات مفتاح: منصة مراجعة ذكية، جلسات مراجعة شخصية، موارد تعليمية، تتبع التقدم، مؤقت، نظام كشف الحضور، Angular، Yolov8، Gemini API، Flask، SpringBoot