



**REPUBLIQUE TUNISIENNE  
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR  
ET DE LA RECHERCHE SCIENTIFIQUE  
UNIVERSITE TUNIS EL MANAR**

**FACULTE DES SCIENCES DE TUNIS  
DEPARTEMENT DES SCIENCES DE L'INFORMATIQUE**

**RAPPORT DE  
PROJET FÉDÉRÉ LCS2**

**TITRE  
campus connect**

**Réalisé par :**

Hizaoui Hamza  
Bahri Ilef  
Boughalmi Amen Allah  
Bouabdallah Ameny

**Encadré par : Faouzi MOUSSA**

**Organisme d'accueil: FST – DSI**

## Fiche de synthèse

Dans le cadre de la gestion informatisée des stages et de l'archivage des rapports, nous vous demandons de renseigner les items suivants :

- Formation : LCS2
- Année Universitaire : 2023-2024
- Session (Principale / Juillet / Septembre) : Principale
- Auteur(s) :
  - Nom & Prénom : Hizaoui Hamza
  - Nom & Prénom : Bahri Ilef
  - Nom & Prénom : Boughalmi Amen Allah
  - Nom & Prénom : Bouabdallah Ameny
- Email et téléphone du Responsable du groupe : [ameny.bouabdallah@etudiant-fst.utm.tn](mailto:ameny.bouabdallah@etudiant-fst.utm.tn)  
22015816
- Titre du rapport : campus connect
- Titre traduit en français (pour les titres en langue étrangère) : campus connect
- Organisme d'accueil : FST – DSI
- Pays d'accueil : Tunisie
- Responsable de stage (nom et prénom) : MOUSSA Faouzi
- Email du Responsable de stage : [lcs.pf.2024@gmail.com](mailto:lcs.pf.2024@gmail.com)
- Mots-clés caractérisant votre rapport (4 à 5 mots maximum) : université éducation digitalisation

## **Charte de non plagiat**

## **Protection de la propriété intellectuelle**

Tout travail universitaire doit être réalisé dans le respect intégral de la propriété intellectuelle d'autrui. Pour tout travail personnel, ou collectif, pour lequel le candidat est autorisé à utiliser des documents (textes, images, musiques, films etc.), celui-ci devra très précisément signaler le crédit (référence complète du texte cité, de l'image ou de la bande-son utilisés, sources internet incluses) à la fois dans le corps du texte et dans la bibliographie. Il est précisé que l'UCO dispose d'un logiciel anti-plagiat, aussi est-il demandé à tout étudiant de remettre à ses enseignants un double de ses travaux lourds sur support informatique.

Cf. « Prévention des fraudes à l'attention des étudiants »

Je soussigné(e), Bouabdallah Ameny étudiant(e) en GLSI 2 m'engage à respecter cette charte.

Fait à Tunis le 10/03/2024

Signature :

## **Synthèse du rapport en français**

Le projet "Compus Connect" vise à moderniser et numériser les processus au sein de l'université, offrant une plateforme intégrée pour les étudiants, les enseignants et l'administration. Cette initiative s'articule autour de trois espaces distincts : celui des étudiants, des enseignants et des administrateurs.

L'espace étudiant constitue le point central pour les étudiants, offrant des fonctionnalités telles que l'inscription en ligne, l'accès aux documents académiques (cours, documents universitaires), la consultation du menu du restaurant universitaire et l'exploration des horaires de transport. Cette interface vise à simplifier et rendre plus accessible l'ensemble de l'expérience étudiante.

Pour les enseignants, l'espace dédié offre la possibilité de consulter les horaires de transport, le menu du restaurant universitaire et de déposer des cours en ligne. Il vise à faciliter la gestion des activités académiques, tout en favorisant une communication transparente et efficace avec les étudiants.

L'espace administrateur constitue le cœur de la gestion globale du système. L'administrateur peut consulter un tableau de bord des utilisateurs, superviser et ajuster les horaires de transport, ainsi que gérer le menu du restaurant universitaire. Cette fonctionnalité centralisée offre un contrôle complet sur les aspects logistiques et organisationnels de l'université.

En résumé, "Compus Connect" ambitionne de créer une expérience universitaire plus fluide et connectée, où les processus traditionnels sont transformés par la numérisation. La plateforme vise à améliorer l'efficacité administrative, à renforcer l'interaction entre les différentes parties prenantes et à offrir aux étudiants un accès facile aux ressources essentielles. Ce rapport détaille les fonctionnalités spécifiques de chaque espace, mettant en lumière l'impact positif attendu sur l'ensemble de la communauté universitaire.

# Table des matières

<b>Introduction Générale</b>	<b>12</b>
<b>Chapitre 1: État de l'Art et Fondements Métiers</b>	<b>13</b>
1- Contexte Métier du Projet	13
● Présentation du domaine d'application du projet	13
● Analyse des besoins et des problématiques spécifiques du domaine	13
2- Problématique Métier	14
● Identification des principaux défis ou des lacunes à résoudre dans le domaine	14
● Définition claire des objectifs visés par le projet	15
3- Proposition de Valeur	15
● Mise en évidence de la valeur ajoutée du projet pour les parties prenantes	15
Étudiants :	15
Enseignants :	16
Administrateurs :	16
● Description des avantages et des bénéfices attendus	16
4- Clients Cibles et Marché	17
● Analyse du Public Cible :	17
● Analyse de Marché :	17
5- Opportunités et menaces	18
● Identification des Opportunités :	18
● Identification des Menaces Potentielles :	18
6- Backlog Métier	19
User Stories pour l'Authentification :	19
User Stories pour l'Espace Étudiant :	19
User Stories pour l'Espace Enseignant :	19
User Stories pour l'Espace Administrateur :	19
<b>Chapitre 2 : Fondements Technologiques et Cadre Conceptuel du Projet</b>	<b>20</b>
1- Revue de la Littérature	20
2- Contexte Informatique	20
● Introduction aux concepts informatiques fondamentaux liés au projet	20
Architecture Orientée Services (SOA) :	20
Base de Données NoSQL (MongoDB) :	21
JavaScript pour le Développement Front-End et Back-End :	21
API RESTful :	21
● Revue des technologies et des langages de programmation pertinents	21
Node.js pour le Back-End :	21
React pour le Front-End :	21
MongoDB comme Système de Gestion de Base de Données :	21

Figma pour le Maquettage :	22
Gestion de Version avec Git :	22
<b>3- État de l'Art en Architecture Logicielle</b>	<b>22</b>
● Présentation des architectures logicielles couramment utilisées et analyse des avantages et des inconvénients de chaque architecture	22
1. Modèle-Vue-Contrôleur (MVC) :	22
2. Architecture Orientée Services (SOA) :	23
3. Architecture Microservices :	23
● Analyse des Avantages et Inconvénients :	24
<b>4- Outils et Méthodologies</b>	<b>24</b>
● Présentation des frameworks utilisés dans des projets similaires	24
-Express.js (Node.js Framework) :	24
-React.js (Front-End Library) :	24
-Mongoose (MongoDB Framework) :	24
● Exploration des design patterns pertinents pour le développement logiciel	25
-Pattern Observer :	25
-Pattern Singleton :	25
● Analyse des Environnements de Développement Intégrés (EDI) fréquemment employés	25
Visual Studio Code :	25
● Examen des langages de programmation adaptés au contexte du projet	25
JavaScript (Node.js et React) :	25
● Méthodologie Agile et Gestion du Backlog :	25
Scrum :	25
Backlog :	25
<b>5- Modèle d'Architecture</b>	<b>26</b>
Présentation (UI) :	26
Logique Métier (Business Logic) :	26
Accès aux Données (Data Access) :	26
Infrastructure :	26
<b>6- Cadre Conceptuel</b>	<b>26</b>
<b>Chapitre 3 : Conception UML de la Solution Informatique</b>	<b>28</b>
1- Introduction à la Conception	28
2- Analyse des Besoins	28
● Les besoins du projet	28
● Priorisation des fonctionnalités	29
3- Architecture Globale du Système	29
● Proposition d'une architecture générale pour la solution	29
● Diagrammes de composants	30
● Diagrammes de déploiement	31
● Diagrammes de packages	31

● Description des composants principaux et de leurs interactions	32
<b>4- Modélisation des Données</b>	<b>33</b>
● Conception du schéma de la base de données	33
● Diagrammes de classes	34
<b>5- Interfaces Utilisateur</b>	<b>35</b>
● Charte graphique et maquettes	35
Charte graphique	35
Maquettage:	36
Inscription	36
Connexion	38
réinitialiser votre mot de passe	38
Profil	40
consulter horaire transport	40
consulter muni resto	41
consulter documents	41
dépot fichier utiles/ cours	42
approuver cours	43
dépot horaire transport	44
dépot menu resto	44
gestion utilisateur	45
● Ergonomie et expérience utilisateur	46
Couleurs :	46
Police de Texte :	47
<b>6- Algorithmes et Méthodes</b>	<b>47</b>
● Algorithmes clés	47
Algorithme de Recherche :	47
Algorithme de Tri:	48
Algorithme d'Authentification :	48
● Diagrammes d'activités	50
authentification	50
Horaire transport	51
menu resto	52
Gestion annonce	53
Gestion cours	54
consultation cours	55
Consulter annonces	56
Approuver cours	57
● Explication des méthodes et techniques particulières	57
<b>7- Sécurité et Confidentialité</b>	<b>58</b>
● Stratégies de sécurité mises en place	58

● Les autorisations et les rôles	58
Étudiant :	58
Enseignant :	59
Administrateur :	59
● Mesures de protection des données sensibles	59
<b>8- Scénarios de Cas d'Utilisation</b>	<b>60</b>
● Diagrammes de cas d'utilisation inscription	60
● Diagrammes de cas d'utilisation connexion	61
● Diagrammes de cas d'utilisation dépôt cours	62
● Diagrammes de cas d'utilisation dépôt annonces	63
● Diagrammes de cas d'utilisation dépôt pv	64
● Diagrammes de cas d'utilisation dépôt documents	65
● Diagrammes de cas d'utilisation gestion menu resto	66
● Diagrammes de cas d'utilisation télécharger document	67
● Diagrammes de cas d'utilisation telecharger cours	68
● Diagrammes de cas d'utilisation approuver cours	69
● Diagrammes de cas d'utilisation consulter menu resto	70
● Diagrammes de cas d'utilisation gestion transport	71
● Diagrammes de cas d'utilisation consulter transport	72
● Diagrammes de cas d'utilisation consulter annonces	73
● Diagrammes de cas d'utilisation consulter pv	74
<b>9- Diagrammes de Séquence</b>	<b>75</b>
● Illustration des flux de contrôle entre les différents composants	75
Étudiant :	76
Enseignant :	76
Administrateur :	76
<b>10- Plan de Développement</b>	<b>77</b>
● Division du projet en modules ou itérations de développement	77
● Diagrammes de classes et structure du code	78
● Calendrier de mise en œuvre de la conception	78
<b>11- Conclusion</b>	<b>79</b>
<b>Chapitre 4 : Réalisation et validation</b>	<b>81</b>
<b>1- Introduction à la Réalisation</b>	<b>81</b>
● Présentation des objectifs de la phase de réalisation	81
● Enjeux liés à la mise en œuvre du projet	81
<b>2- Environnement de Développement</b>	<b>82</b>
● Description des outils, plateformes, et langages de programmation utilisés	82
Langages de Programmation :	82
JavaScript :	82
JSX CSS :	82

Frameworks et Bibliothèques :	82
React.js :	82
Node.js :	82
Express.js :	82
bcrypt :	82
MongoDB :	82
Postman:	83
Outils de Développement :	83
Visual Studio Code :	83
Git :	83
GitHub :	83
● Justification des choix technologiques	83
Évolutivité et Flexibilité :	83
Large Adoption et Support Communautaire :	83
Performance et Réactivité :	84
Sécurité :	84
Adaptabilité aux Besoins Spécifiques :	84
3- Développement des Modules	84
● Les modules identifiés dans la conception	84
Module d'Authentification :	84
Module de Gestion des Utilisateurs :	85
Module de Consultation des Cours et des Documents :	85
Module de Gestion des Cours :	85
Module de Gestion des Annonces :	85
Module de Gestion du Restaurant Universitaire :	85
Module de Gestion des Transports Universitaires :	85
● Utilisation de UML pour guider l'implémentation	85
4- Tests et Débogage	86
● Stratégies de tests appliquées aux différents modules	86
● Processus de débogage pour résoudre les erreurs identifiées	86
5- Intégration des Composants	86
● Mise en œuvre de l'architecture globale du système	86
● Coordination des différents modules pour assurer leur interopérabilité	86
6- Validation et Vérification	87
● Évaluation de la solution par rapport aux besoins initiaux	87
● Vérification de la conformité aux spécifications définies dans la conception	87
7- Optimisation des Performances	87
● Identification des points d'optimisation éventuels	87
● Actions entreprises pour améliorer les performances du système	88
8- Documentation Technique	88

● Rédaction d'une documentation détaillée pour faciliter la maintenance	88
Architecture du Système :	88
Fonctionnalités et Interfaces :	88
● Utilisation de UML pour compléter la documentation technique	89
<b>9- Formation et Transfert de Compétences</b>	<b>89</b>
● Formation des utilisateurs finaux et de l'équipe de maintenance	89
● Transfert des compétences nécessaires à la gestion du projet	89
<b>10- Conclusion du Chapitre</b>	<b>90</b>
● Bilan de la réalisation et de la validation	90
● Préparation à la phase de déploiement et d'exploitation du projet	90
<b>Conclusion Générale et Perspectives</b>	<b>91</b>
<b>Résumé</b>	<b>91</b>
<b>Abstract</b>	<b>92</b>
<b>ملخص</b>	<b>93</b>

## **Introduction Générale**

---

*L'évolution rapide de la technologie a profondément transformé le paysage de l'éducation, stimulant la demande croissante de solutions innovantes pour améliorer les processus universitaires. Dans ce contexte, le projet "Compus Connect" émerge comme une réponse stratégique à la nécessité de digitaliser et d'optimiser les opérations au sein d'une université. Cette initiative vise à créer une plateforme intégrée, offrant aux étudiants, enseignants et administrateurs un accès centralisé à des services essentiels tels que la gestion des cours, les informations universitaires, les horaires de transport et les menus du restaurant.*

*Cette démarche s'inscrit dans une perspective plus large de transformation numérique de l'éducation, cherchant à améliorer l'expérience universitaire, à renforcer la communication, et à simplifier la gestion logistique au quotidien. À travers ce rapport, nous explorerons en détail le contexte métier du projet, les défis spécifiques au domaine universitaire, la proposition de valeur de "Compus Connect", ainsi que les éléments fondamentaux du cadre conceptuel et des choix technologiques. En mettant l'accent sur la méthodologie Agile et la gestion du backlog, nous dévoilerons comment ces aspects contribuent à la réalisation réussie du projet.*

*Le projet "Compus Connect" représente une étape significative vers la création d'un environnement universitaire connecté, efficient et convivial. En examinant les éléments clés de son développement, ce rapport vise à fournir une vision globale du projet, soulignant son importance dans l'ère numérique actuelle et les avantages qu'il apportera à la communauté universitaire concernée.*

# Chapitre 1: État de l'Art et Fondements Métiers

## **1- Contexte Métier du Projet**

### **• Présentation du domaine d'application du projet**

Le projet "Compus Connect" émerge au sein du contexte universitaire, un environnement complexe et dynamique où la gestion efficace des informations et des processus est cruciale. Les universités sont des institutions caractérisées par la diversité des acteurs impliqués, comprenant des étudiants, des enseignants et du personnel administratif. Cette diversité crée un besoin constant de communication, de partage d'informations et de gestion des tâches quotidiennes.

Le domaine d'application s'étend à la fois sur la sphère académique et logistique, intégrant des aspects tels que la diffusion de cours, la consultation de documents universitaires, la gestion des emplois du temps, le suivi des services de restauration et la coordination des transports. En réunissant ces éléments au sein d'une plateforme numérique centralisée, " Compus Connect " cherche à simplifier les interactions et à offrir une expérience intégrée à tous les acteurs impliqués dans la vie universitaire.

### **• Analyse des besoins et des problématiques spécifiques du domaine**

-Communication et Partage d'Informations : Les universités sont souvent confrontées à des défis liés à la communication et à la diffusion rapide des informations. Les besoins spécifiques incluent la nécessité de partager des cours, des documents universitaires et des annonces importantes de manière efficace.

-Gestion Logistique : Les emplois du temps des étudiants et des enseignants, ainsi que la disponibilité des services de restauration et des transports, constituent des aspects logistiques essentiels à coordonner au sein d'une université. La centralisation de ces informations facilitera la planification et l'organisation au quotidien.

-Expérience Étudiante Améliorée : Les étudiants recherchent une expérience universitaire plus connectée et pratique. L'accès facile aux cours, la consultation des menus du restaurant universitaire et la connaissance des horaires de transport contribuent à une vie universitaire plus efficace et agréable.

-Gestion Administrative : Les administrateurs ont besoin d'un outil qui leur permette de gérer efficacement les utilisateurs, tout en assurant la coordination des services logistiques tels que les transports et les repas. Une vue globale de ces éléments est cruciale pour une administration proactive.

## 2- Problématique Métier

- **Identification des principaux défis ou des lacunes à résoudre dans le domaine**

-Fragmentation des Informations : Actuellement, les informations essentielles, telles que les cours, les documents universitaires, les horaires de transport et les menus du restaurant, sont dispersées dans divers formats et plateformes. Cela crée une fragmentation de l'information, rendant parfois difficile l'accès rapide et efficace à ces ressources cruciales.

-Communication inefficace : Les canaux de communication traditionnels peuvent être inefficaces pour diffuser rapidement des informations importantes à l'ensemble de la communauté universitaire. Les retards dans la transmission d'annonces cruciales peuvent entraîner des désagréments et une perte de temps pour les étudiants, les enseignants et le personnel administratif.

-Gestion Logistique Complexifiée : La coordination des emplois du temps, des services de restauration et des transports nécessite une attention constante. L'absence d'une plateforme centralisée pour gérer ces aspects logistiques peut entraîner des conflits d'horaires, des retards et une planification incohérente.

- **Définition claire des objectifs visés par le projet**

-Centralisation de l'Information : L'objectif principal est de centraliser toutes les informations pertinentes au sein de la plateforme "Compus Connect". Cela permettra aux utilisateurs d'accéder facilement aux cours, aux documents universitaires, aux horaires de transport et aux menus du restaurant à partir d'une seule interface.

-Amélioration de la Communication : Le projet vise à améliorer la communication en fournissant un moyen rapide et efficace de diffuser des annonces importantes à l'ensemble de la communauté universitaire. Une communication plus proactive contribuera à réduire les retards et à accroître l'efficacité globale.

-Optimisation de la Gestion Logistique : En centralisant les informations sur les emplois du temps, les menus du restaurant et les horaires de transport, le projet vise à optimiser la gestion logistique. Cela facilitera la planification quotidienne et réduira les conflits potentiels.

-Amélioration de l'Expérience Utilisateur : En offrant aux étudiants, aux enseignants et aux administrateurs un accès facile et rapide aux informations pertinentes, le projet vise à améliorer l'expérience utilisateur globale au sein de l'université. Une plateforme conviviale contribuera à une vie universitaire plus agréable et connectée.

### **3- Proposition de Valeur**

- **Mise en évidence de la valeur ajoutée du projet pour les parties prenantes**

#### **Étudiants :**

-Accès Centralisé : Les étudiants bénéficieront d'un accès centralisé à leurs cours, documents universitaires, horaires de transport et menus du restaurant, simplifiant ainsi leur vie quotidienne sur le campus.

-Communication Rapide : Recevoir rapidement les annonces et informations importantes contribuera à améliorer leur expérience universitaire, tout en favorisant une communication transparente.

#### **Enseignants :**

-Gestion Simplifiée : Les enseignants pourront déposer leurs cours de manière efficace, tout en accédant facilement aux horaires de transport et aux menus du restaurant. Cela leur permettra de consacrer davantage de temps à l'enseignement.

-Coordination Facilitée : Une vue centralisée des emplois du temps et des services logistiques simplifiera la coordination des activités académiques et logistiques.

#### **Administrateurs :**

-Gestion Intégrée : Les administrateurs auront un tableau récapitulatif des utilisateurs, facilitant ainsi la gestion administrative. La centralisation des informations sur les transports et les repas permettra une gestion plus efficace.

-Prise de Décision Informée : Une vue d'ensemble des activités au sein de l'université permettra aux administrateurs de prendre des décisions informées pour améliorer les services et l'efficacité globale.

- Description des avantages et des bénéfices attendus**

**Gain de Temps :** Les utilisateurs auront accès à toutes les informations nécessaires depuis une seule plateforme, éliminant ainsi le besoin de consulter plusieurs sources.

La gestion automatisée des tâches administratives réduira la charge de travail, permettant aux parties prenantes de se concentrer sur des activités plus importantes.

**Amélioration de l'efficacité :** La centralisation des informations et la communication rapide amélioreront l'efficacité des processus universitaires, réduisant les retards et les erreurs.

**La gestion logistique optimisée contribuera à une planification plus cohérente et à une utilisation plus efficace des ressources.**

**Expérience Utilisateur Améliorée :** Les étudiants bénéficieront d'une expérience plus fluide et connectée, renforçant ainsi leur engagement et leur satisfaction. Les enseignants apprécieront la simplicité dans la gestion de leurs cours et la coordination des activités académiques.

**Prise de décision Stratégique :** Les administrateurs auront accès à des données pertinentes, facilitant ainsi la prise de décision stratégique pour l'amélioration continue des services universitaires.

## **4- Clients Cibles et Marché**

- Analyse du Public Cible :**

"Compus Connect" cible les étudiants, enseignants et administrateurs universitaires, établissant une plateforme BtoC pour répondre aux besoins spécifiques de la communauté universitaire. Les étudiants bénéficieront d'une gestion centralisée de leurs activités académiques, les enseignants auront des outils simplifiés pour la coordination des cours, et les administrateurs disposent d'une vue d'ensemble pour optimiser les services universitaires.

- Analyse de Marché :**

L'application répond à une demande croissante de solutions numériques dans la gestion universitaire. Dans un contexte où la numérisation de l'éducation gagne en importance, "Compus Connect" se positionne comme une réponse pertinente en offrant une plateforme centralisée. Elle s'aligne sur la nécessité croissante d'optimiser l'efficacité opérationnelle, la communication et l'expérience étudiante, s'inscrivant ainsi dans une tendance de marché en constante évolution.

## 5- Opportunités et menaces

- **Identification des Opportunités :**

-Demande Croissante de Solutions Numériques :La transition vers des solutions numériques dans le secteur de l'éducation offre une opportunité pour "Compus Connect" d'adresser la demande croissante de gestion intégrée au sein des universités.

-Besoin d'Expérience Utilisateur Améliorée :L'accent accru sur l'expérience utilisateur dans l'éducation crée une opportunité pour le projet d'offrir une plateforme conviviale, améliorant ainsi la satisfaction des étudiants, des enseignants et des administrateurs.

-Intégration de Nouvelles Technologies :L'opportunité d'intégrer des technologies émergentes telles que l'intelligence artificielle, la réalité virtuelle ou l'analyse de données pour améliorer encore les fonctionnalités et les performances de la plateforme.

- **Identification des Menaces Potentielles :**

-Résistance au Changement :La résistance au changement au sein des institutions éducatives peut constituer une menace, car certains utilisateurs peuvent être réticents à adopter de nouvelles plateformes numériques.

-Sécurité des Données :La question de la sécurité des données est cruciale, et une violation de la confidentialité pourrait nuire à la réputation de la plateforme. Des mesures de sécurité robustes doivent être mises en place.

-Évolution des Normes et Régulations :Les changements fréquents dans les normes et régulations du secteur éducatif peuvent poser des défis en termes d'adaptation de la plateforme pour rester conforme.

## **6- Backlog Métier**

### **User Stories pour l'Authentification :**

- En tant qu'utilisateur, je veux pouvoir créer un compte
- En tant qu'utilisateur, je peut se connecter à mon compte
- En tant qu'utilisateur, je veux avoir la possibilité de réinitialiser mon mot de passe
- En tant qu'administrateur, je souhaite avoir la capacité de gérer les comptes d'utilisateurs

### **User Stories pour l'Espace Étudiant :**

- En tant qu'étudiant, je veux pouvoir consulter mes cours
- En tant qu'étudiant, je veux pouvoir consulter mon profil
- En tant qu'étudiant, je souhaite consulter les documents de l'université, tels que les emplois du temps ,le pv ou les annonces
- En tant qu'étudiant, je souhaite consulter les horaires de transport
- En tant qu'étudiant, je veux avoir un accès rapide au menu du restaurant universitaire avec des informations sur les plats et les horaires

### **User Stories pour l'Espace Enseignant :**

- En tant qu'enseignant, je veux pouvoir consulter mon profil
- En tant qu'enseignant, je veux pouvoir déposer mes cours sur la plateforme de manière intuitive et efficace
- En tant qu'enseignant, je souhaite consulter les horaires de transport
- En tant qu'enseignant, je veux accéder rapidement au menu du restaurant universitaire

### **User Stories pour l'Espace Administrateur :**

- En tant qu'administrateur, je veux pouvoir consulter mon profil
- En tant qu'administrateur, je veux avoir une vue d'ensemble du tableau de bord des utilisateurs
- En tant qu'administrateur, je souhaite avoir la capacité de gérer les horaires de transport, y compris les mises à jour et les changements éventuels
- En tant qu'administrateur, je veux être en mesure de mettre à jour le menu du restaurant universitaire

# Chapitre 2 : Fondements Technologiques et Cadre Conceptuel du Projet

---

## 1- Revue de la Littérature

La revue de la littérature a révélé un paysage diversifié de solutions numériques dans le domaine de la gestion universitaire. Plusieurs travaux de recherche ont mis en évidence l'importance croissante d'intégrer des technologies numériques pour améliorer l'efficacité et l'expérience utilisateur au sein des institutions éducatives. Les principales avancées comprennent la centralisation des informations, la gestion automatisée des tâches administratives et l'accent sur une communication transparente.

Les méthodologies étudiées ont souligné l'importance de la collaboration entre les parties prenantes lors du développement de solutions de gestion universitaire. L'utilisation de prototypes interactifs et de retours continus a été identifiée comme un moyen efficace d'adapter les solutions aux besoins spécifiques de la communauté universitaire. Les résultats de ces recherches ont souvent montré une amélioration significative de l'efficacité opérationnelle et de la satisfaction des utilisateurs grâce à l'adoption de plateformes numériques intégrées.

## 2- Contexte Informatique

### • Introduction aux concepts informatiques fondamentaux liés au projet

Le projet "Compus Connect" repose sur une architecture logicielle moderne, favorisant l'intégration fluide des différentes fonctionnalités pour répondre aux besoins spécifiques de la gestion universitaire. Les concepts informatiques fondamentaux qui sous-tendent ce projet incluent :

#### **Architecture Orientée Services (SOA) :**

La mise en œuvre d'une architecture orientée services permet une modularité et une extensibilité optimales, en facilitant l'intégration de nouvelles fonctionnalités et la maintenance du système.

### **Base de Données NoSQL (MongoDB) :**

MongoDB, une base de données NoSQL, a été sélectionnée pour sa flexibilité dans le stockage de données non structurées, adaptée aux besoins variés du projet, tels que la gestion des cours, des emplois du temps et des menus du restaurant.

### **JavaScript pour le Développement Front-End et Back-End :**

L'utilisation de JavaScript en tant que langage de programmation à la fois pour le développement front-end (React) et back-end (Node.js) offre une cohérence dans le code, favorisant la collaboration entre les équipes et la maintenance efficace.

### **API RESTful :**

L'adoption d'une architecture d'API RESTful facilite la communication entre les différentes parties du système, permettant une intégration fluide des services et une communication efficace entre le front-end et le back-end.

## **• Revue des technologies et des langages de programmation pertinents**

### **Node.js pour le Back-End :**

Node.js offre une exécution côté serveur rapide et évolutive, idéale pour gérer les opérations liées à la base de données, à la gestion des utilisateurs et à la logique métier.

### **React pour le Front-End :**

React, une bibliothèque JavaScript, est utilisé pour le développement du front-end en raison de sa modularité, de sa performance élevée et de sa facilité d'intégration.

### **MongoDB comme Système de Gestion de Base de Données :**

MongoDB, en tant que base de données NoSQL, offre une flexibilité de schéma qui s'adapte bien à la diversité des données à gérer, tout en assurant une évolutivité horizontale.

### **Figma pour le Maquettage :**

Figma est choisi comme outil de maquettage en raison de sa facilité d'utilisation collaborative, permettant aux concepteurs et aux développeurs de travailler en tandem pour concevoir et itérer sur l'interface utilisateur.

### **Gestion de Version avec Git :**

Git est utilisé pour la gestion de version, facilitant le suivi des modifications apportées au code source, la collaboration entre les membres de l'équipe et le déploiement continu.

## **3- État de l'Art en Architecture Logicielle**

- **Présentation des architectures logicielles couramment utilisées et analyse des avantages et des inconvénients de chaque architecture**

### **1. Modèle-Vue-Contrôleur (MVC) :**

L'architecture MVC est un modèle classique qui divise une application en trois composants principaux : le Modèle (pour la gestion des données), la Vue (pour l'interface utilisateur) et le Contrôleur (pour la logique métier).

-Avantages :

Séparation des responsabilités, facilitant la maintenance et l'évolutivité.

Réutilisation du code grâce à la modularité.

Possibilité de travailler sur chaque composant indépendamment.

-Inconvénients :

Peut devenir complexe pour des applications très grandes.

La communication entre les composants peut être complexe.

## **2. Architecture Orientée Services (SOA) :**

SOA est une approche où les services sont offerts par des composants indépendants et interopérables. Ces services sont accessibles via des interfaces standardisées, souvent des API, favorisant la réutilisation et l'intégration.

Avantages :

Flexibilité et réutilisabilité des services.

Interopérabilité entre systèmes hétérogènes.

Évolutivité grâce à la modularité des services.

Inconvénients :

La complexité de la mise en œuvre peut augmenter avec le nombre de services.

La gestion de la cohérence des données peut être un défi.

## **3. Architecture Microservices :**

Les microservices sont une évolution de l'approche SOA, où l'application est construite comme une collection de services autonomes, chacun déployable indépendamment.

Avantages :

Évolutivité et déploiement indépendant des services.

Facilité de maintenance et de mise à l'échelle horizontale.

Technologie adaptée aux équipes de développement indépendantes.

Inconvénients :

La complexité de la gestion des communications entre microservices.

Besoin d'une infrastructure robuste pour la gestion des services.

- **Analyse des Avantages et Inconvénients :**

-MVC : Convient bien aux applications de taille moyenne où la simplicité et la séparation des préoccupations sont essentielles. Cependant, peut devenir complexe pour des applications très étendues.

-SOA : Idéal pour les environnements où l'interopérabilité et la réutilisabilité des services sont cruciales. La complexité de la mise en œuvre peut être un défi, mais les avantages peuvent surpasser ces inconvénients dans des contextes spécifiques.

-Microservices : Offre une grande flexibilité et évolutivité, idéal pour des systèmes distribués complexes. Cependant, la gestion des communications entre microservices peut introduire une complexité supplémentaire.

Pour "Compus Connect", une approche SOA ou Microservices pourrait être bénéfique, en fonction des besoins spécifiques et de la complexité prévue de la plateforme. La prise de décision doit être basée sur la scalabilité, la maintenance, et l'interopérabilité souhaitées pour le contexte universitaire.

## 4- Outils et Méthodologies

- **Présentation des frameworks utilisés dans des projets similaires**

- Express.js (Node.js Framework)** :

Utilisé pour le développement côté serveur avec Node.js, il offre une structure légère et rapide pour construire des applications web robustes.

- React.js (Front-End Library)** :

Un framework JavaScript bien établi pour la construction d'interfaces utilisateur interactives et réactives.

- Mongoose (MongoDB Framework)** :

Facilite l'interaction avec la base de données MongoDB en fournissant des fonctionnalités de modélisation et de validation des données.

- **Exploration des design patterns pertinents pour le développement logiciel**

**-Pattern Observer :**

Pour la gestion des événements et la communication entre les composants de l'application, tels que la mise à jour des données en temps réel.

**-Pattern Singleton :**

Peut être utilisé pour garantir une seule instance de certaines classes critiques, comme la gestion de la base de données.

- **Analyse des Environnements de Développement Intégrés (EDI) fréquemment employés**

**Visual Studio Code :**

Un EDI léger, polyvalent et largement adopté, offrant une prise en charge complète de Node.js, React, et des extensions pour MongoDB.

- **Examen des langages de programmation adaptés au contexte du projet**

**JavaScript (Node.js et React) :**

Langage principal pour le développement back-end (Node.js) et front-end (React).

- **Méthodologie Agile et Gestion du Backlog :**

**Scrum :**

Méthodologie Agile bien adaptée aux projets de développement logiciel, offrant une approche itérative et incrémentale.

**Backlog :**

Intégré comme un outil essentiel de gestion des tâches et des fonctionnalités à développer. Le backlog est constamment ajusté pour refléter les priorités changeantes et les retours des utilisateurs.

## 5- Modèle d'Architecture

### **Présentation (UI) :**

La couche d'interface utilisateur de "Compus Connect" est la première interaction entre les utilisateurs et la plateforme. Elle gère l'affichage des cours, annonces, horaires de transport, et menus du restaurant. La couche de présentation assure une expérience utilisateur fluide, permettant l'inscription, la connexion, et la navigation intuitive à travers les fonctionnalités offertes.

### **Logique Métier (Business Logic) :**

Au cœur de l'architecture, la couche de logique métier définit les règles et processus fondamentaux de "Compus Connect". Elle traite les données entrantes de la couche de présentation, appliquant la logique pour la gestion des cours, annonces, horaires de transport, et menus du restaurant. Cette couche garantit la cohérence des opérations et facilite la prise de décision automatisée.

### **Accès aux Données (Data Access) :**

Responsable de la gestion efficace des informations essentielles, la couche d'accès aux données interagit avec la base de données MongoDB. Elle récupère et met à jour les données nécessaires à la logique métier, assurant la manipulation sécurisée des informations et la cohérence des données à travers la plateforme.

### **Infrastructure :**

La couche d'infrastructure constitue la base technique de "Compus Connect". Gérant la sécurité des connexions réseau, l'accès aux autorisations, et d'autres éléments techniques, cette couche offre la stabilité et les performances nécessaires au bon fonctionnement de l'ensemble du système.

## 6- Cadre Conceptuel

Le cadre conceptuel du projet "Compus Connect" repose sur des principes théoriques et pratiques du domaine informatique, guidant le développement vers des objectifs spécifiques. L'architecture logicielle choisie, SOA , détermine la

structure globale du projet, offrant flexibilité ou scalabilité selon les besoins. Les frameworks, tels que Node.js, React.js et Mongoose, assurent une base technique cohérente et performante. L'application de design patterns renforce la maintenabilité du code.

Les environnements de développement intégrés (EDI), comme Visual Studio Code , fournissent des outils optimisés pour le développement, le débogage et la collaboration. Le langage JavaScript unifie le développement back-end, front-end et l'interaction avec la base de données. La méthodologie Agile (Scrum ) et la gestion du backlog facilitent une approche itérative, permettant une adaptation continue aux changements et aux retours des utilisateurs.

Ce cadre conceptuel crée une base robuste, alignant les concepts du domaine informatique avec les objectifs spécifiques du projet, assurant cohérence, réactivité et efficacité tout au long du cycle de développement de "Compus Connect".

# Chapitre 3 : Conception UML de la Solution Informatique

## **1- Introduction à la Conception**

La phase de conception dans le développement de logiciels joue un rôle essentiel dans la transformation des exigences fonctionnelles et non fonctionnelles en solutions concrètes. Ce chapitre vise à présenter une vision d'ensemble de la conception de la plateforme "Compus Connect". Nous explorerons les objectifs de cette phase, qui consistent à définir une architecture solide, à spécifier les composants du système, et à élaborer des modèles UML détaillés pour guider le développement. En comprenant les principes et les méthodes de conception, nous sommes mieux équipés pour créer une solution logicielle robuste, évolutive et conforme aux besoins de nos utilisateurs.

## **2- Analyse des Besoins**

### **• Les besoins du projet**

La synthèse des besoins du projet "Compus Connect" révèle une palette diversifiée de fonctionnalités essentielles pour répondre aux attentes des utilisateurs au sein de l'environnement universitaire. Ces besoins, issus des discussions antérieures, englobent la nécessité de fournir aux étudiants un accès transparent aux cours, aux annonces universitaires et aux horaires de transport, ainsi que la facilitation des tâches administratives pour les enseignants et les administrateurs. Il est également crucial de garantir une authentification sécurisée pour tous les utilisateurs afin de protéger les données sensibles et de maintenir l'intégrité du système. Cette synthèse fournit un aperçu clair des fonctionnalités requises pour répondre efficacement aux besoins de la communauté universitaire et établir une plateforme numérique fonctionnelle et conviviale.

- **Priorisation des fonctionnalités**

La priorisation des fonctionnalités est une étape cruciale dans le processus de développement de la plateforme "Compus Connect". En se basant sur l'analyse des besoins, les fonctionnalités ont été classées en fonction de leur importance. Les fonctionnalités considérées comme essentielles pour le lancement initial de la plateforme, telles que **l'inscription des utilisateurs, la consultation des cours et la gestion des annonces universitaires**, sont placées en haut de la liste des priorités. Ensuite, des fonctionnalités telles que **la consultation des horaires de transport et des menus du restaurant** sont considérées comme des priorités moyennes, tandis que des fonctionnalités complémentaires telles que **la gestion des utilisateurs et l'authentification** sécurisée sont placées en bas de la liste des priorités, mais demeurent essentielles pour assurer la fonctionnalité globale et la sécurité de la plateforme. Cette approche de priorisation garantit que les ressources et les efforts de développement sont alloués de manière efficace, en se concentrant d'abord sur les fonctionnalités fondamentales pour offrir une expérience utilisateur robuste et satisfaisante.

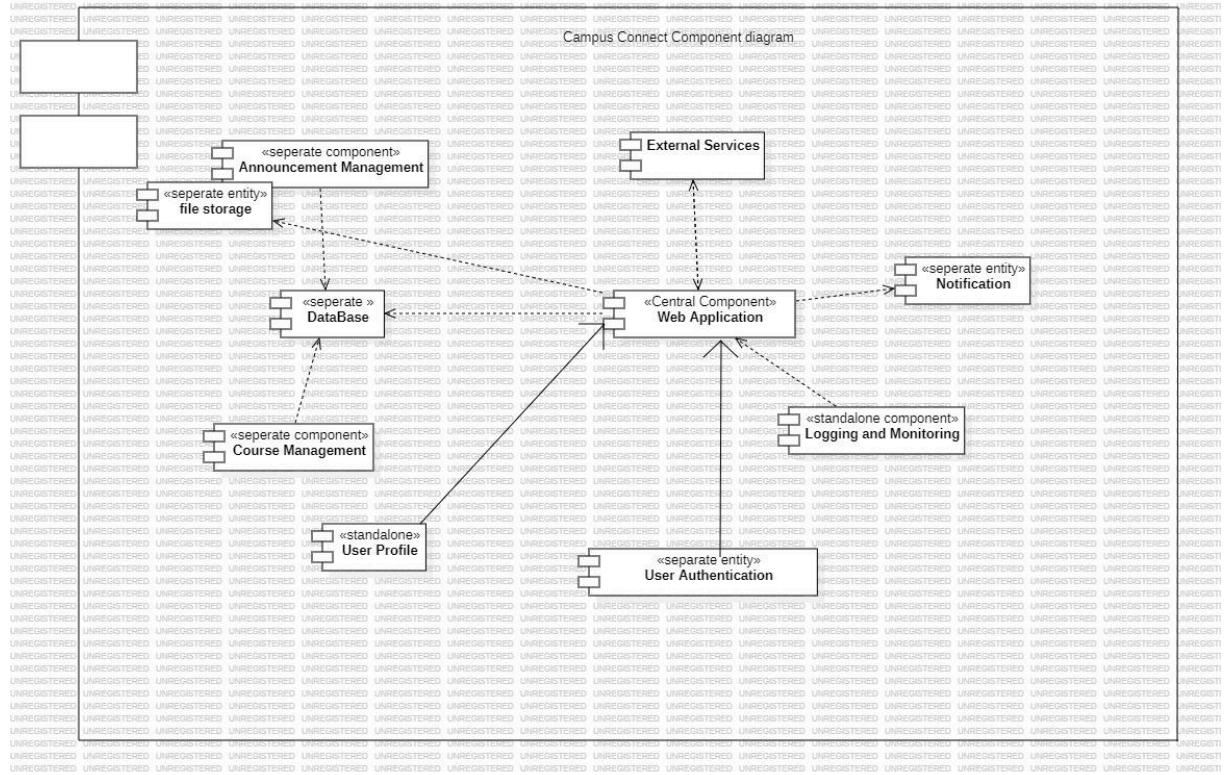
### **3- Architecture Globale du Système**

- **Proposition d'une architecture générale pour la solution**

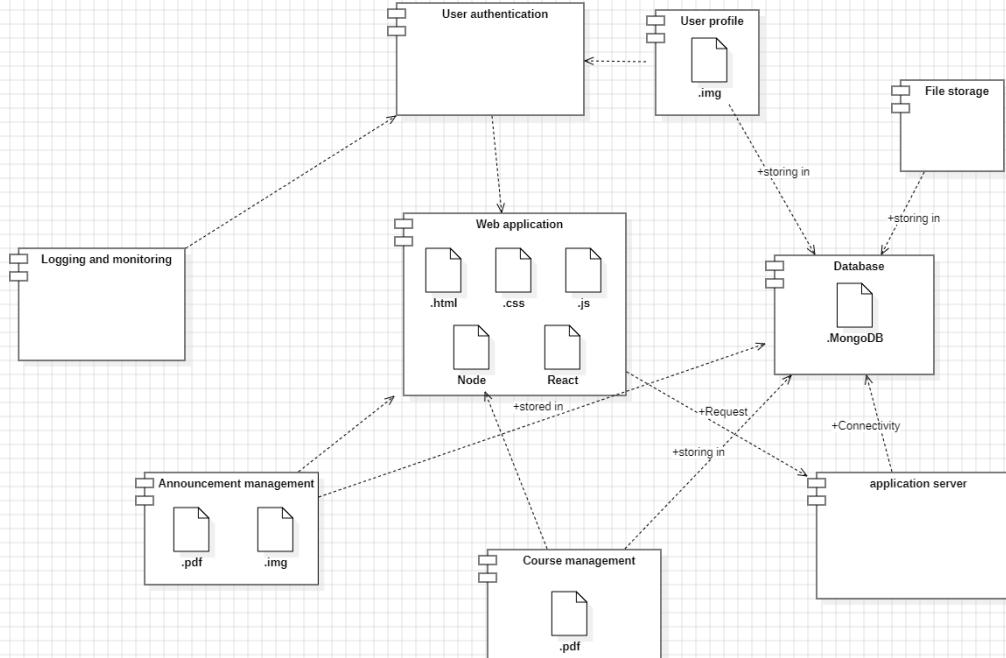
L'architecture proposée pour la plateforme "Compus Connect" est basée sur un modèle client-serveur, offrant ainsi une structure robuste et évolutive pour répondre aux besoins de la communauté universitaire. Le système sera divisé en plusieurs couches, chacune jouant un rôle spécifique dans le fonctionnement global de la plateforme. Du côté du serveur, nous envisageons d'utiliser une architecture à microservices, permettant une évolutivité horizontale et une flexibilité dans le développement et le déploiement des fonctionnalités. Cette approche nous permettra de gérer efficacement les différentes fonctionnalités de la plateforme tout en garantissant des performances optimales et une haute disponibilité. Pour le client, nous opterons pour une application web responsive, offrant une expérience

utilisateur fluide et cohérente sur une variété de dispositifs, y compris les ordinateurs de bureau, les tablettes et les smartphones.

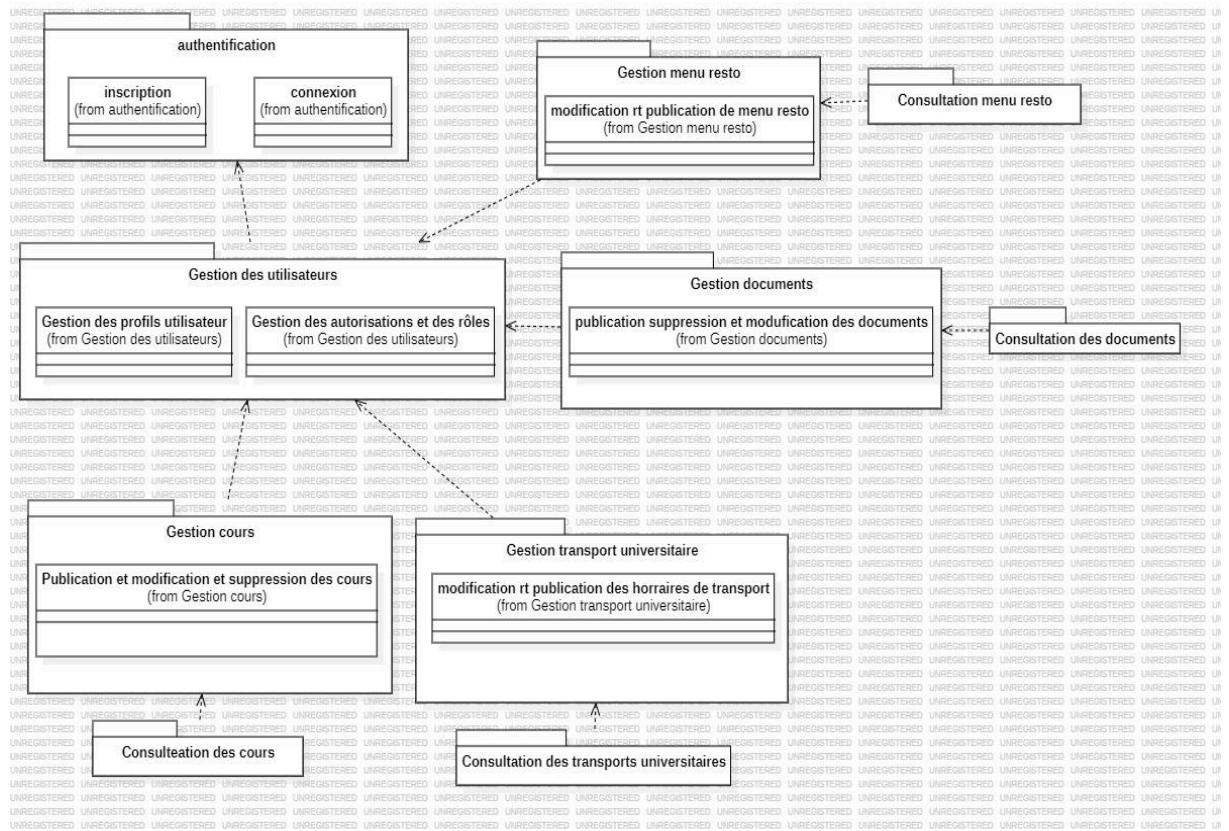
- Diagrammes de composants



## ● Diagrammes de déploiement



## ● Diagrammes de packages

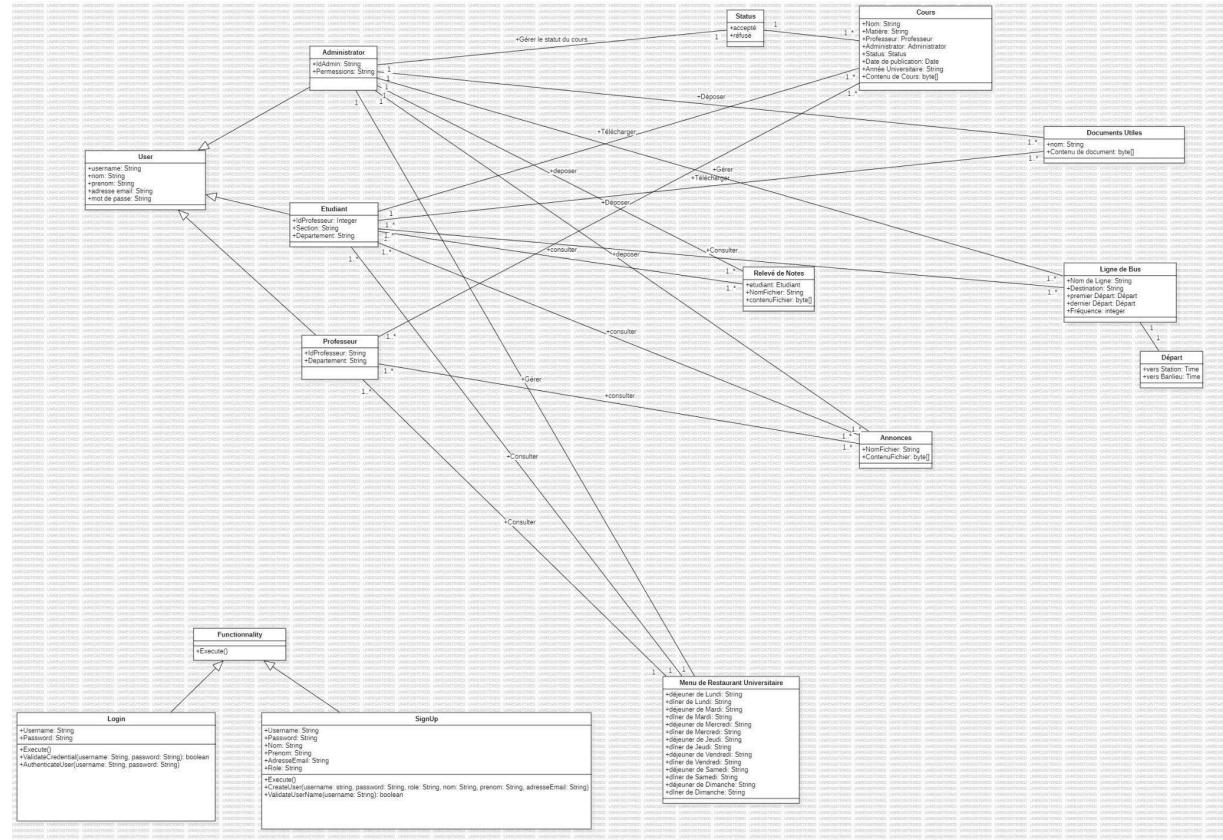


- **Description des composants principaux et de leurs interactions**

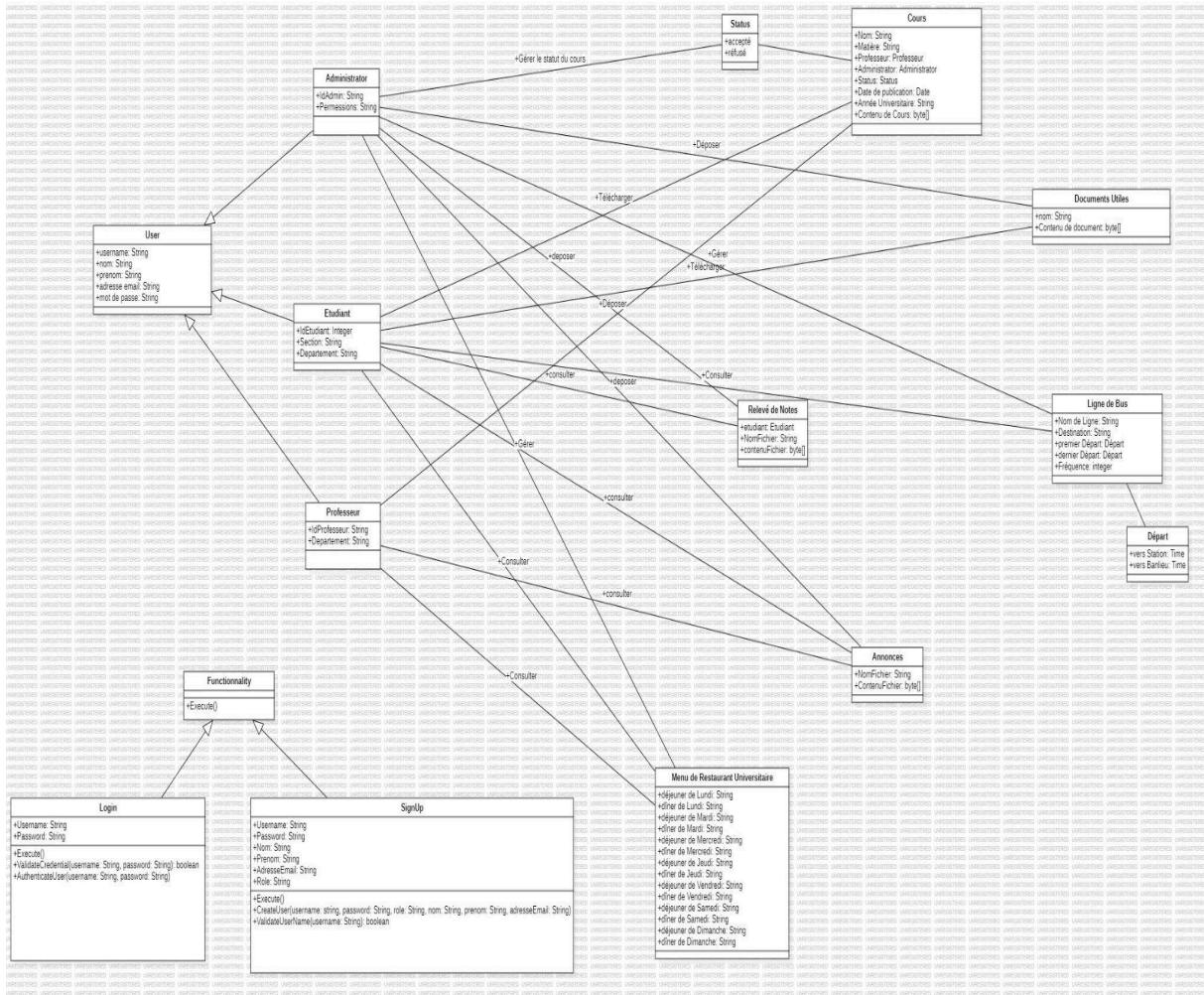
Les principaux composants de l'architecture incluent le front-end client, le back-end serveur, la base de données, et les microservices. Le front-end client sera développé en utilisant React.js pour une expérience utilisateur interactive et réactive. Il interagira avec le back-end serveur via des requêtes HTTP, permettant la récupération et la manipulation des données nécessaires à partir de la base de données. Le back-end serveur sera développé en utilisant Node.js avec Express, offrant une plateforme robuste pour la gestion des requêtes entrantes et la logique métier. Les microservices seront développés de manière à fournir des fonctionnalités spécifiques telles que la gestion des utilisateurs, la consultation des cours, les annonces universitaires, les horaires de transport, et les menus du restaurant. Ils communiqueront entre eux via des API REST, permettant une interaction fluide et une évolutivité indépendante. La base de données, basée sur MongoDB, sera utilisée pour stocker les données nécessaires au fonctionnement de la plateforme, assurant ainsi la persistance et la cohérence des données à travers le système. Ces composants interagiront de manière synergique pour fournir une plateforme complète et fonctionnelle répondant aux besoins de la communauté universitaire.

## 4- Modélisation des Données

### • Conception du schéma de la base de données



## ● Diagrammes de classes



## 5- Interfaces Utilisateur

- Charte graphique et maquettes

### Charte graphique

Logo :



Icône :



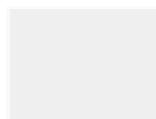
Couleurs :



#2d60b1



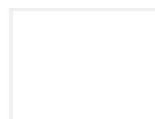
#ffda6f



#efefef



#000000



#ffffff



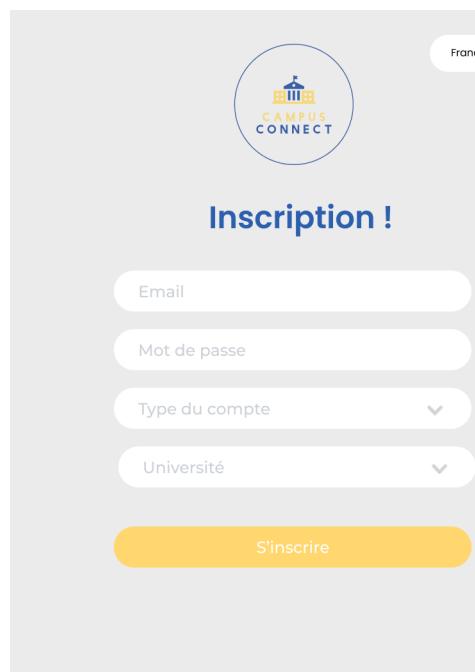
#CDD2D8

Police:

Poppins : Normal / Moyen / Semi gras / Gras

## Maquettage:

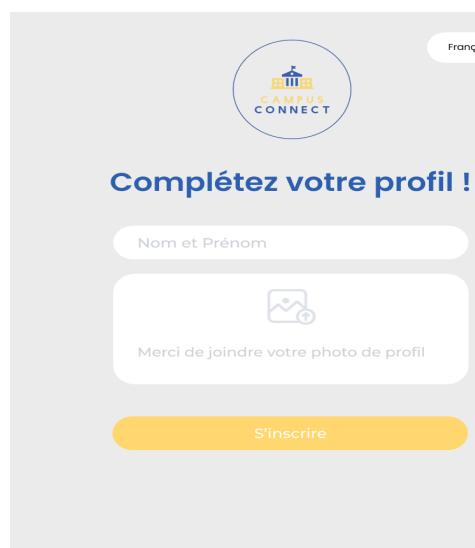
### *Inscription*



The registration form features a logo for "CAMPUS CONNECT" at the top left. A language selection dropdown shows "Français". Below the logo is the heading "Inscription !". The form includes fields for "Email", "Mot de passe", "Type du compte" (with a dropdown arrow), and "Université" (with a dropdown arrow). A large yellow "S'inscrire" button is centered at the bottom.



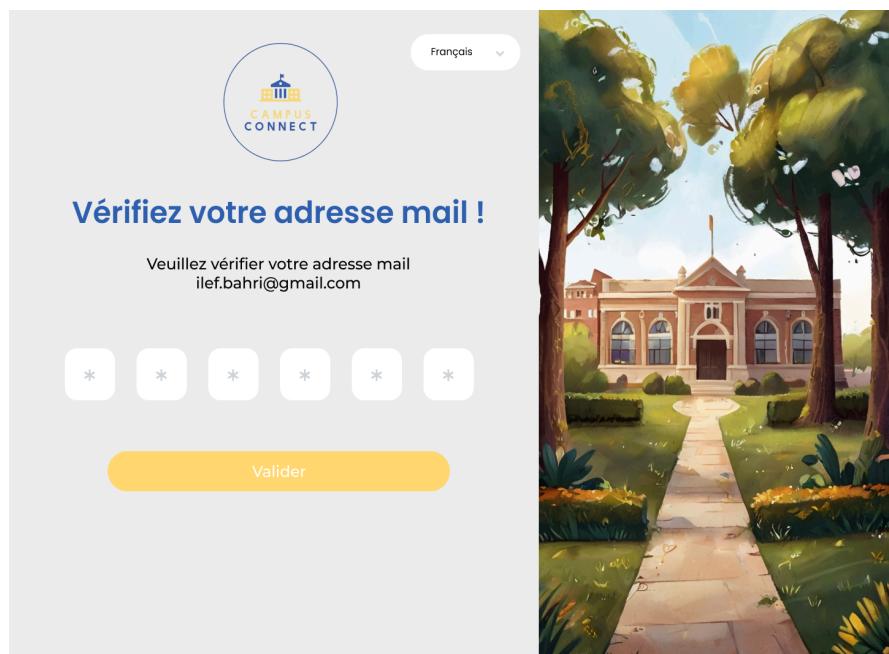
A vibrant, painterly illustration of a large, red-brick university building with multiple arched windows and a prominent central entrance. The building is set against a backdrop of lush green trees and a clear blue sky. In the foreground, a paved walkway leads towards the building through a garden with various plants and flowers.



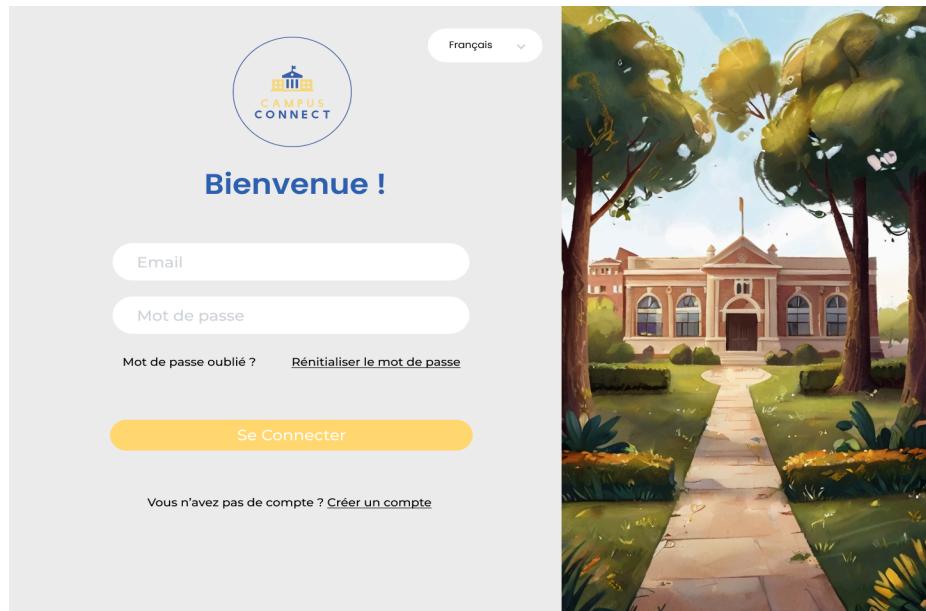
The profile completion form follows a similar layout to the registration page. It includes the "CAMPUS CONNECT" logo, a "Français" language option, and the heading "Complétez votre profil !". Fields for "Nom et Prénom" and a "Photo de profil" (with a placeholder icon) are present. A note below the photo field says "Merci de joindre votre photo de profil". A yellow "S'inscrire" button is at the bottom.



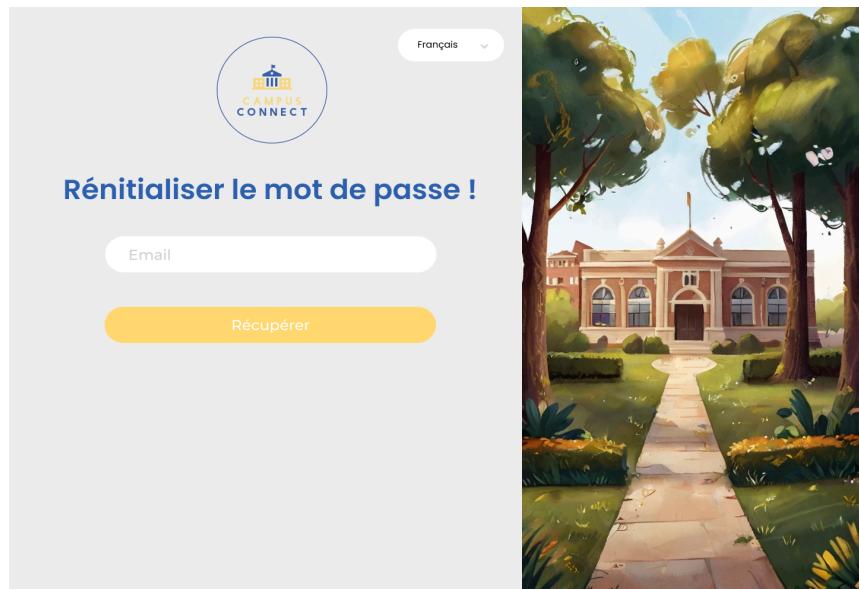
The same vibrant campus building illustration as the registration page, showing the red-brick building, surrounding trees, and a paved walkway leading towards it.

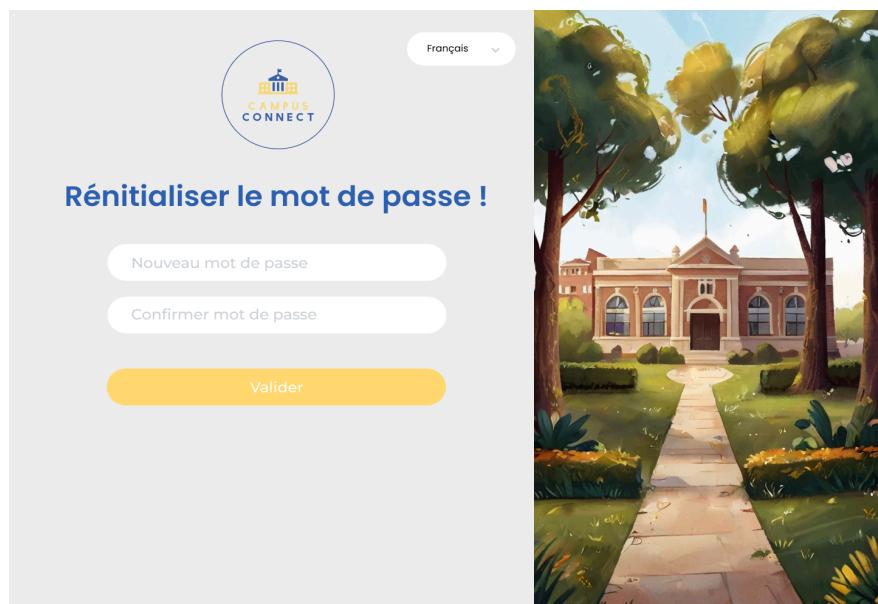
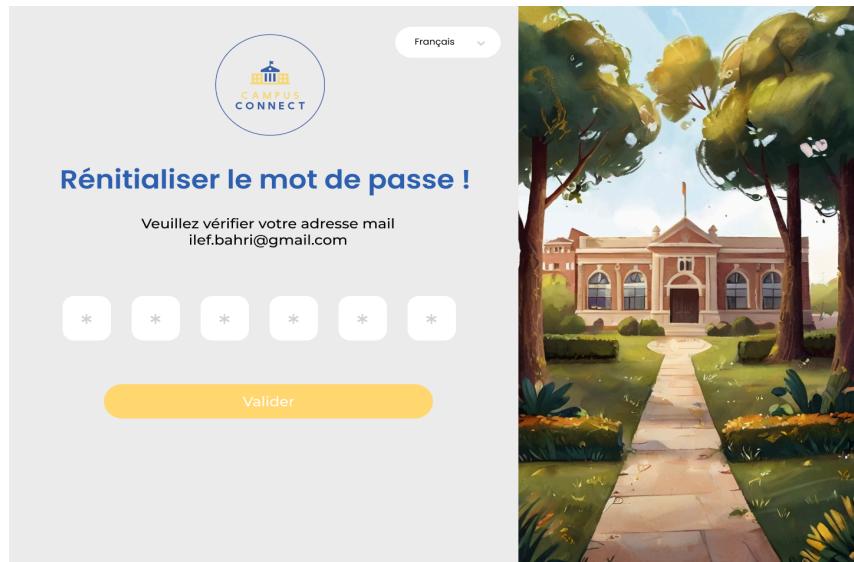


## Connexion



*réinitialiser votre mot de passe*





## Profil

The screenshot shows the 'Profil' (Profile) section of the Campus Connect application. At the top, there is a circular profile picture placeholder with the text 'Ajouter une photo'. Below it is a larger placeholder with a placeholder image of a man's face. A yellow button labeled 'Modifier photo de profil' (Edit profile photo) is located below the large placeholder. To the right of the placeholders is a form for updating personal information:

Nom et prénom	Sami Samsoun Nouveau nom et prénom : <input type="text" value="Flen Foulene"/>
Adresse mail	<u>user@gmail.com</u> Nouveau mail : <input type="text" value="user@gmail.com"/>
Mot de passe	***** Nouveau Mot de passe : <input type="text"/> Confirmer Mot de passe : <input type="text"/>

At the bottom right of the form area are two buttons: 'Annuler' (Cancel) and 'Enregistrer' (Save).

consulter horaire transport

The screenshot shows the 'Horaires transports' (Transport Timetables) section of the Campus Connect application. On the left is a vertical sidebar with icons for 'Profil', 'Documents', 'Menu', and 'Transport'. The main area has a title 'Horaires transports' and a table titled 'Horaires transports' with the following columns:

Ligne	Destination	Premier départ		Dernier départ		Fréquence
		Vers station	Vers banlieue	Vers station	Vers banlieue	

*consulter muni resto*

The screenshot shows a weekly meal menu grid titled "Menu restaurant de la semaine". The grid has columns for "Repas" (Lundi, Mardi, Mercredi, Jeudi, Vendredi, Samedi, Dimanche) and rows for "Déjeuner" and "Diner". Each cell contains a small placeholder icon.

Repas	Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi	Dimanche
Déjeuner	[Placeholder]						
Diner	[Placeholder]						

*consulter documents*

The screenshot shows a list of documents titled "Liste de documents". The table includes columns for "Nom de document", "Date de création", "Date de dépôt", "Déposé par", and "Gérer fichier". A search bar and a total document count of 24 are also visible.

	Nom de document	Date de création	Date de dépôt	Déposé par	Gérer fichier
<input checked="" type="checkbox"/>	Algebre 2 chap 3	12/06/2014	24/05/2023	A.Boudriga	<span>trash</span> <span>checkmark</span> <span>X</span>
<input checked="" type="checkbox"/>	Java- Exception	12/06/2014	01/02/2024	C.Abdelbasset	<span>trash</span> <span>checkmark</span> <span>X</span>
<input checked="" type="checkbox"/>	UI-UX Figma	12/06/2014	18/06/2022	C.Chaouachi	<span>trash</span> <span>checkmark</span> <span>X</span>
<input checked="" type="checkbox"/>	Droit	12/06/2014	10/10/2024	M.Riahi	<span>trash</span> <span>checkmark</span> <span>X</span>
<input checked="" type="checkbox"/>	Administration BD	12/06/2014	18/09/2022	M.Kammoun	<span>trash</span> <span>checkmark</span> <span>X</span>
<input checked="" type="checkbox"/>	Compilation 2	12/06/2014	09/07/2023	W.Krichene	<span>trash</span> <span>checkmark</span> <span>X</span>
<input checked="" type="checkbox"/>	Flex- Linux	12/06/2014	17/08/2022	F.Faleh	<span>trash</span> <span>checkmark</span> <span>X</span>

Liste de documents

Java-Exceptions  
Fichier Pdf

Details

Matière: Programmation Java  
Professeur: C.Abdelbaset  
Date de création: 12/06/2014  
Date de dépôt: 01/02/2024  
Année Universitaire: 2023/2024

Boudriga, bdelbaset, Chaouachi, M.Riahi, Hammouni, Krichene

*dépot fichier utiles/ cours*

Dépot Fichiers Utiles

Nom du document

Type de document

Date de dépôt

Merci de joindre votre photo de profil

Confirmer

### *approuver cours*

The screenshot shows a user interface for managing documents. On the left is a sidebar with icons for Utilisateurs, Documents, Menu, Transport, and Dépot Fichiers. The main area is titled 'Liste de documents' and displays a grid of documents with columns for name, date, and author. A modal dialog box is overlaid on the grid, containing a large green 'APPROVED' stamp and the question 'Etes vous sûre d'accepter ce fichier?'. It has two buttons: a red 'X' and a green 'Oui, Accepter'.

This screenshot shows the same 'Liste de documents' interface. A modal dialog box is overlaid on the grid, featuring an illustration of a hand holding a document with a red 'REFUSED' stamp. It asks 'Etes vous sûre de refuser ce Ficher?' and includes a red 'X' button and a green 'Oui, Réfuser' button.

### *dépot horaire transport*

Campus Connect

## Horaires transports

Ligne	Destination	Premier départ		Dernier départ		Fréquence
		Vers station	Vers banlieue	Vers station	Vers banlieue	
<input type="text"/>						
<input type="text"/>						
<input type="text"/>						
<input type="text"/>						
<input type="text"/>						
<input type="text"/>						
<input type="text"/>						

+

x

Valider

dépot menu resto



**PARIS**  
CITY CONNECT

[Français](#) ▾

## Menu restaurant de la semaine

---

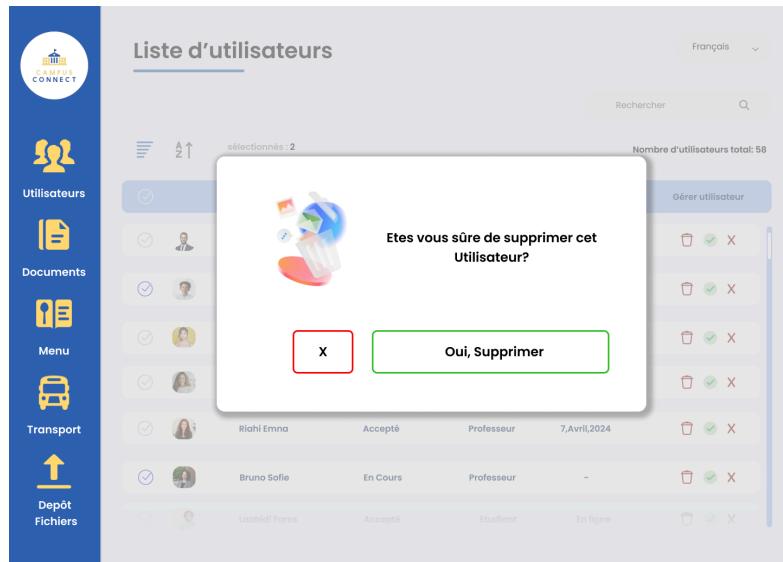
Utilisateurs
Documents
Menu
Transport

Repas	Lundi	Mardi	Mercredi	Jeudi	Vendredi	Samedi	Dimanche
Déjeuner							
Diner							

Valider

## gestion utilisateur

Nom & Prénom	Statut de compte	Type d'utilisateur	Dernière visite	Gérer utilisateur
John John	En Cours	Professeur	-	<span>trash</span> <span>checkmark</span> <span>X</span>
Barlet David	En cours	Etudiant	-	<span>trash</span> <span>checkmark</span> <span>X</span>
Miller Kate	Accepté	Etudiant	Mars,20,2024	<span>trash</span> <span>checkmark</span> <span>X</span>
Jones Samantha	Accepté	Etudiant	En ligne	<span>trash</span> <span>checkmark</span> <span>X</span>
Riahi Emna	Accepté	Professeur	7,Avril,2024	<span>trash</span> <span>checkmark</span> <span>X</span>
Bruno Sofie	En Cours	Professeur	-	<span>trash</span> <span>checkmark</span> <span>X</span>
Laobidi Fares	Accepté	Etudiant	En ligne	<span>trash</span> <span>checkmark</span> <span>X</span>



## ● Ergonomie et expérience utilisateur

Les choix d'ergonomie et d'expérience utilisateur dans la conception d'une plateforme comme "Compus Connect" jouent un rôle crucial dans la création d'une interface conviviale et intuitive pour les utilisateurs. Voici une brève explication des choix effectués, en tenant compte des couleurs et de la police de texte sélectionnées

### Couleurs :

Les choix de couleurs et de police de texte pour "Compus Connect" sont les suivants :

**Bleu (#2d60b1)** : Associé à la confiance et à la fiabilité, il est utilisé pour les éléments interactifs importants, tels que les boutons d'action ou les liens.

**Jaune (#ffda6f)** : Sa vivacité attire l'attention sur les informations cruciales ou les notifications, évoquant la positivité et l'énergie.

**Gris Clair (#efefef)** : Utilisé pour les arrière-plans ou les zones de contenu non interactives, il crée un contraste agréable tout en conservant une esthétique moderne.

**Noir (#000000)** : Sa lisibilité élevée le rend idéal pour le texte principal, offrant un fort contraste sur des arrière-plans colorés.

**Blanc (#ffffff)** : Il assure une visibilité maximale du contenu sur des surfaces sombres, grâce à son contraste élevé.

#### **Police de Texte :**

Poppins : La police de caractères Poppins a été choisie pour son style moderne et élégant, ainsi que pour sa lisibilité à diverses tailles. En utilisant différentes variations de poids (normal, moyen, semi-gras, gras), il est possible de créer une hiérarchie visuelle claire dans le texte, en mettant en évidence les éléments importants et en facilitant la lecture pour les utilisateurs.

:

## **6- Algorithmes et Méthodes**

### **• Algorithmes clés**

#### **Algorithm de Recherche :**

Un algorithme de recherche efficace pour trouver des informations spécifiques dans une base de données est l'algorithme de recherche linéaire. Voici un exemple d'algorithme de recherche linéaire en pseudo-code :

**Fonction rechercheLineaire(liste, elementRecherche)**

**Pour chaque élément dans la liste**

**Si l'élément courant est égal à l'élément recherché**

**Retourner l'indice de l'élément courant**

**Fin Pour**

**Retourner -1 (si l'élément n'est pas trouvé)**

**Fin Fonction**

### **Algorithme de Tri:**

Un algorithme de tri couramment utilisé et efficace est le tri rapide (quicksort).

Voici un exemple d'algorithme de tri rapide en pseudo-code :

Fonction triRapide(liste)

    Si la liste est vide ou ne contient qu'un élément

        Retourner la liste

    Sinon

        Sélectionner un élément pivot de la liste

        Partitionner la liste en deux sous-listes :

- Une sous-liste contenant les éléments plus petits que le pivot
- Une sous-liste contenant les éléments plus grands que le pivot

        Trier récursivement les deux sous-listes

        Concaténer les sous-listes triées et le pivot

        Retourner la liste triée

    Fin Si

Fin Fonction

### **Algorithme d'Authentification :**

Un algorithme couramment utilisé pour l'authentification des utilisateurs est le hachage sécurisé des mots de passe.

#### **-Lors de l'inscription d'un nouvel utilisateur :**

Le mot de passe fourni par l'utilisateur est haché à l'aide de la fonction de hachage bcrypt.

Le hachage résultant est stocké dans la base de données, au lieu du mot de passe en texte clair.

```

Fonction    inscription_utilisateur(nom_utilisateur,
mot_de_passe):
    sel = bcrypt.gensalt() # Générer un sel aléatoire
    mot_de_passe_hache=
        bcrypt.hashpw(mot_de_passe.encode('utf-8'), sel)
    # Hacher le mot de passe avec le sel
    Enregistrer          (nom_utilisateur,
mot_de_passe_hache) dans la base de données
    Retourner Succès

```

**-Lors de l'authentification de l'utilisateur :**

Le mot de passe fourni par l'utilisateur lors de la connexion est haché à l'aide de la même fonction de hachage bcrypt.

Le hachage résultant est comparé au hachage stocké dans la base de données.

Si les hachages correspondent, l'utilisateur est authentifié avec succès.

```

Fonctionauthentification_utilisateur(nom_utilisateur,
mot_de_passe):
    mot_de_passe_hache_stored = Récupérer
    mot_de_passe_hache pour nom_utilisateur dans la
    base de données
    Si mot_de_passe_hache_stored est vide alors
        Retourner Échec # L'utilisateur n'existe pas
    Fin Si

```

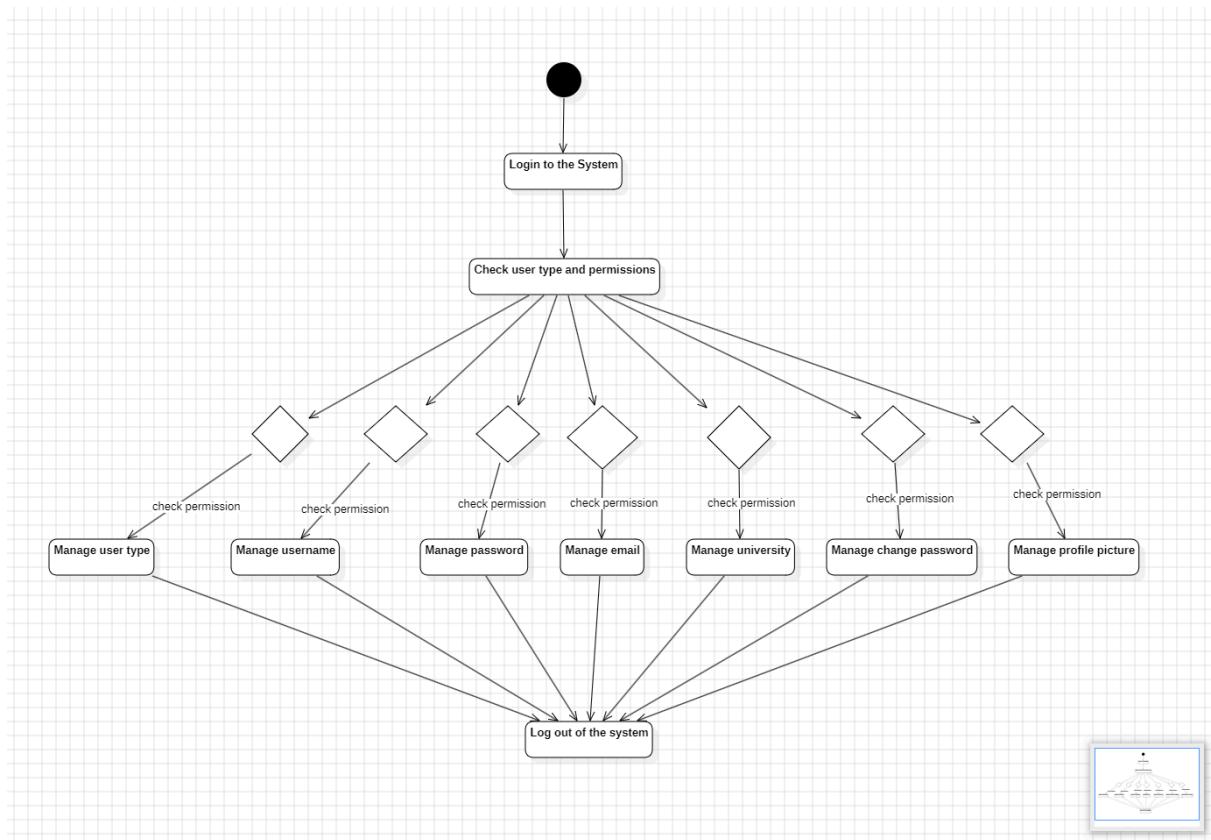
```

Si bcrypt.checkpw(mot_de_passe.encode('utf-8'),
mot_de_passe_hache_stored) alors
    Retourner Succès # Le mot de passe correspond au
    hachage
Sinon
    Retourner Échec # Le mot de passe ne correspond
    pas au hachage

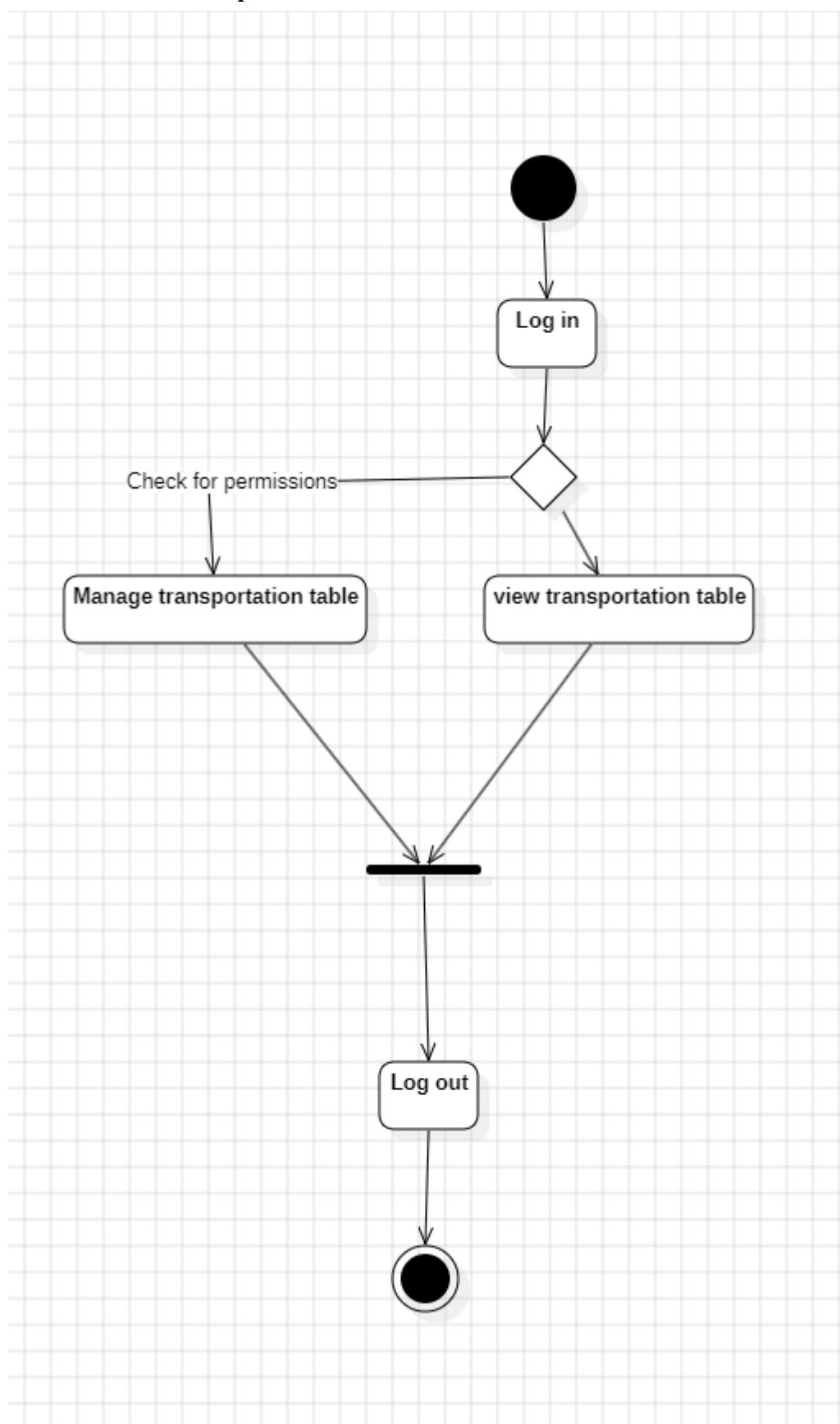
```

- **Diagrammes d'activités**

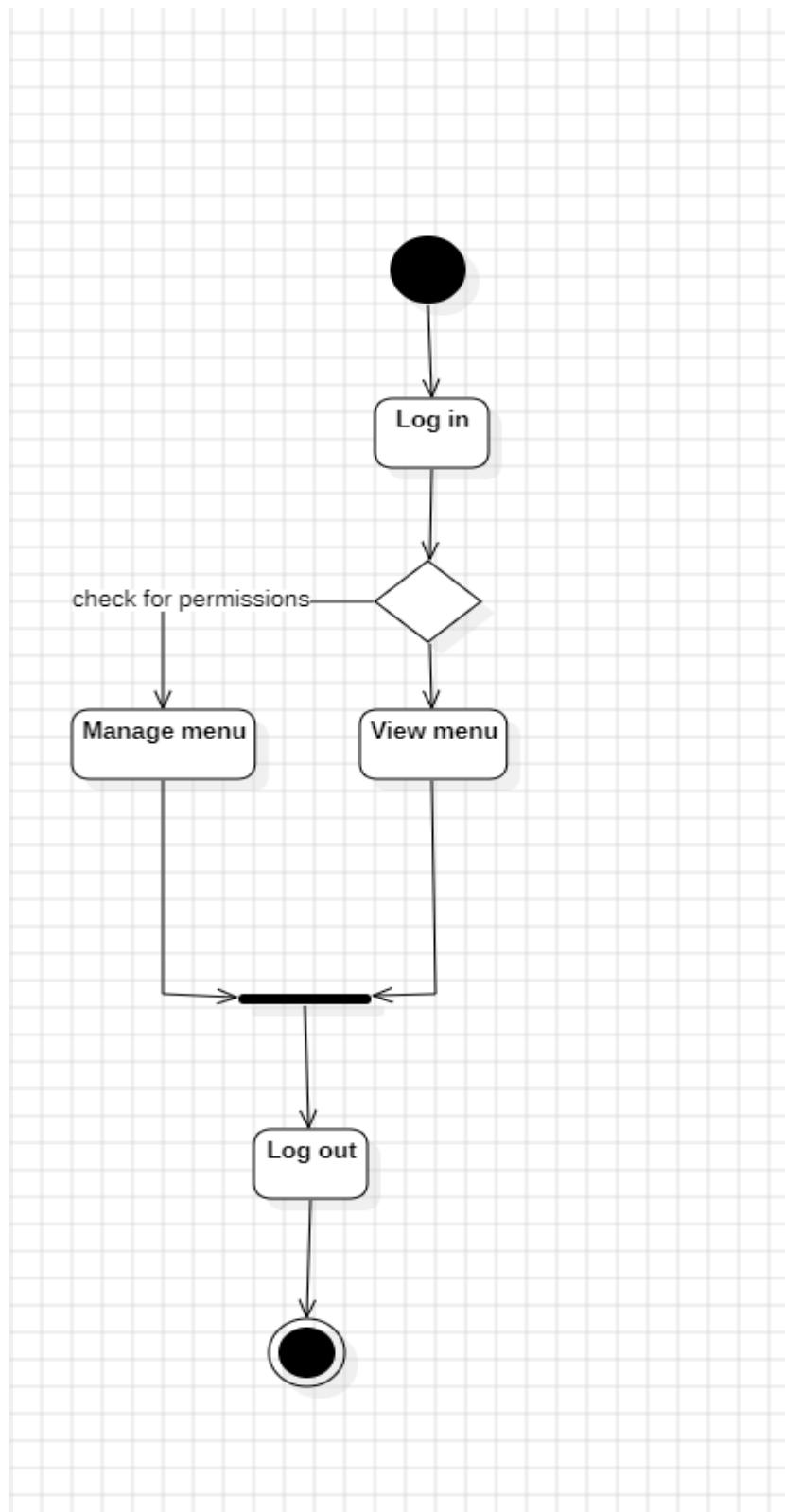
### **authentification**



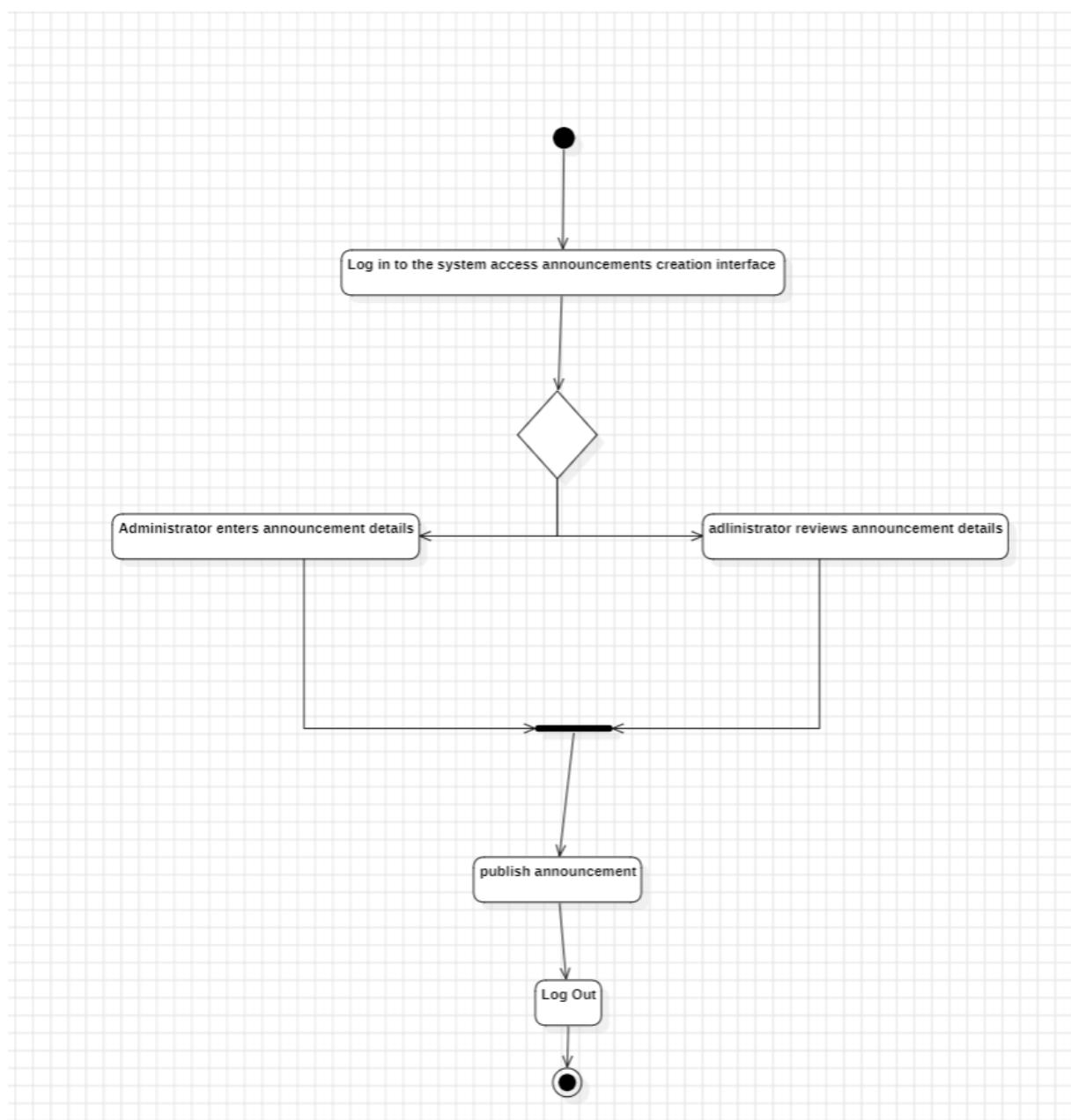
## Horaire transport



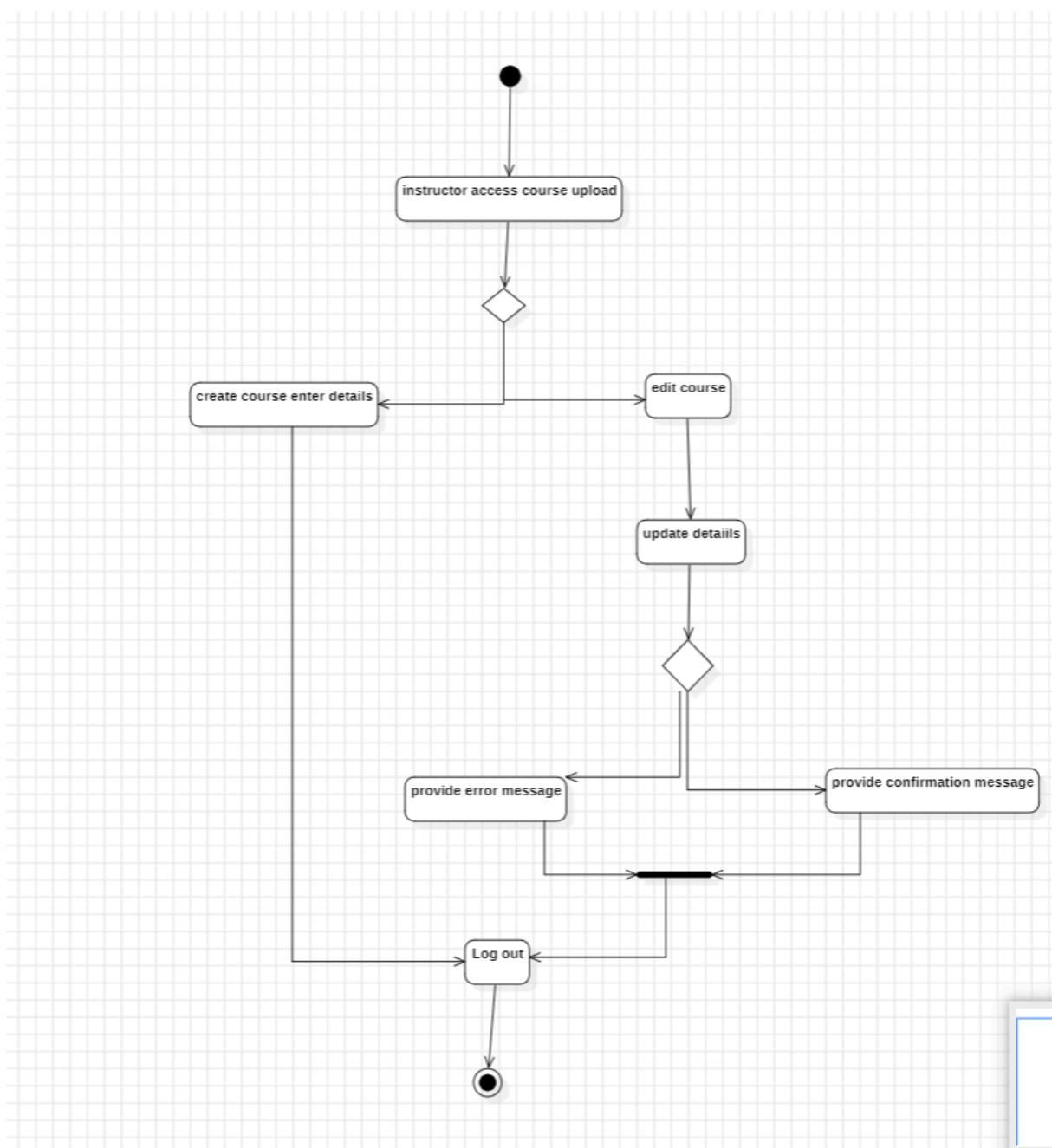
### menu resto



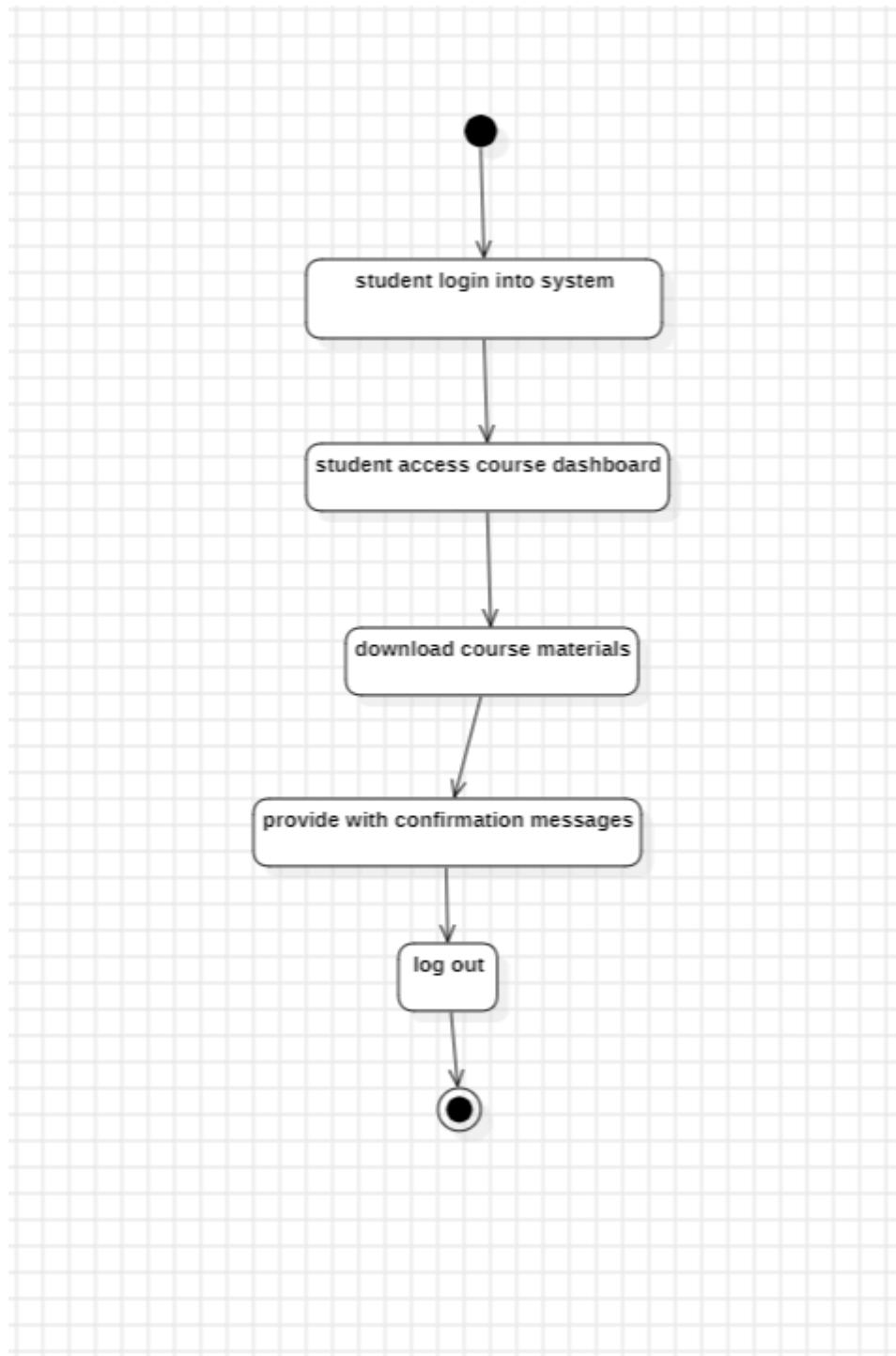
## Gestion annonce



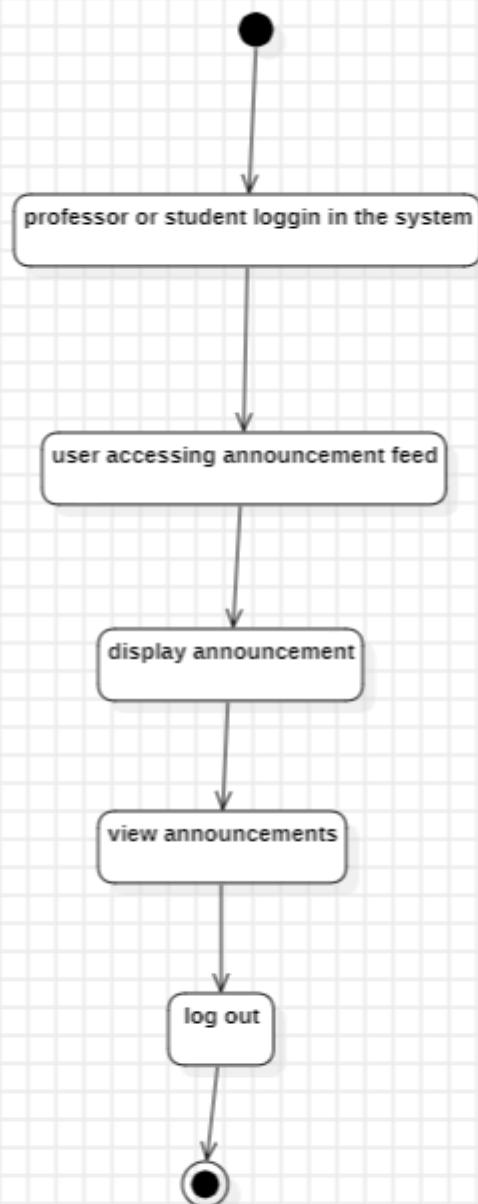
## Gestion cours



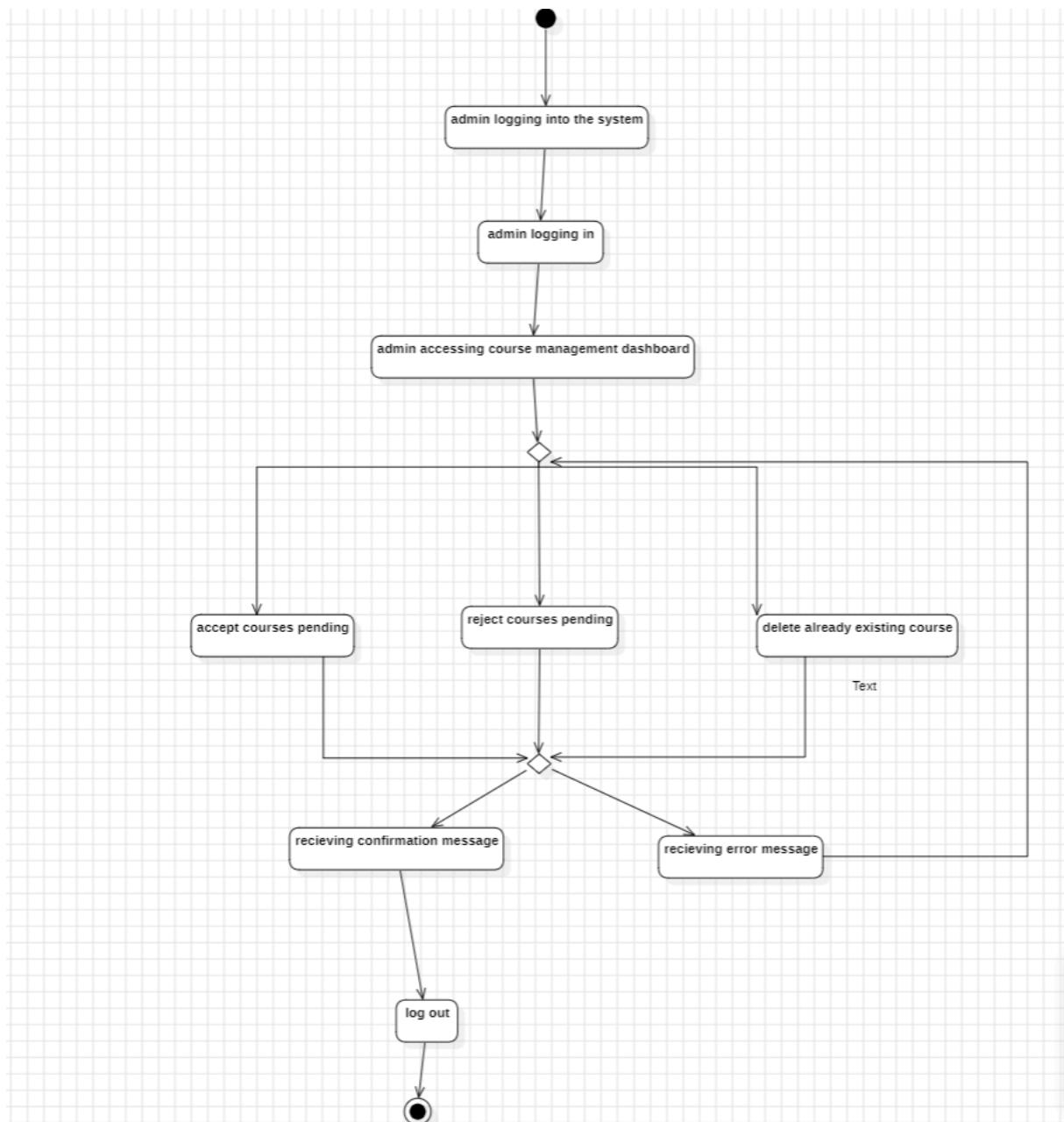
## **consultation cours**



## Consulter annonces



## Approuver cours



- **Explication des méthodes et techniques particulières**

En ce qui concerne les méthodes et techniques particulières, des pratiques de développement telles que le développement piloté par les tests (TDD), la programmation orientée objet (POO) et la conception modulaire peuvent être utilisées pour assurer un développement efficace et évolutif de la solution. L'utilisation de frameworks et de bibliothèques populaires tels que React.js pour le front-end et Node.js pour le back-end peut également contribuer à accélérer le

processus de développement et à garantir une maintenance facile de la solution. En outre, l'application de principes de conception UX/UI permettra de créer une interface utilisateur conviviale et intuitive, améliorant ainsi l'expérience globale des utilisateurs de "Compus Connect".

## 7- Sécurité et Confidentialité

- **Stratégies de sécurité mises en place**

Pour garantir la sécurité et la confidentialité des données sur la plateforme "Compus Connect", plusieurs stratégies ont été mises en place. Tout d'abord, l'authentification des utilisateurs est effectuée de manière sécurisée à l'aide de l'algorithme de hachage bcrypt, garantissant que les mots de passe sont stockés de manière cryptée dans la base de données. De plus, toutes les communications entre le client et le serveur sont sécurisées à l'aide du protocole HTTPS, assurant le chiffrement des données transitant sur le réseau. Des techniques de protection contre les attaques courantes telles que les attaques par injection SQL et les attaques par force brute sont également mises en œuvre pour renforcer la sécurité du système.

- **Les autorisations et les rôles**

Une gestion efficace des autorisations et des rôles est essentielle pour contrôler l'accès aux différentes fonctionnalités de la plateforme. Pour cela, un système de contrôle d'accès basé sur les rôles (RBAC) est mis en place. Les utilisateurs sont attribués à des rôles spécifiques, tels que étudiant, enseignant ou administrateur, chacun ayant des autorisations définies en fonction de ses responsabilités.

**Étudiant :**

- Consulter les cours disponibles.
- Consulter les différents documents universitaires, emploi du temps, pv, annonces....
- Consulter l'horaire des transports universitaires.
- Consulter le menu resto
- Modifier ses propres informations personnelles.

**Enseignant :**

- Publier de nouveaux cours et documents pédagogiques.
- Modifier et mettre à jour le contenu des cours existants.
- Consulter l'horaire des transports universitaires.
- Consulter le menu resto
- Modifier ses propres informations personnelles.

**Administrateur :**

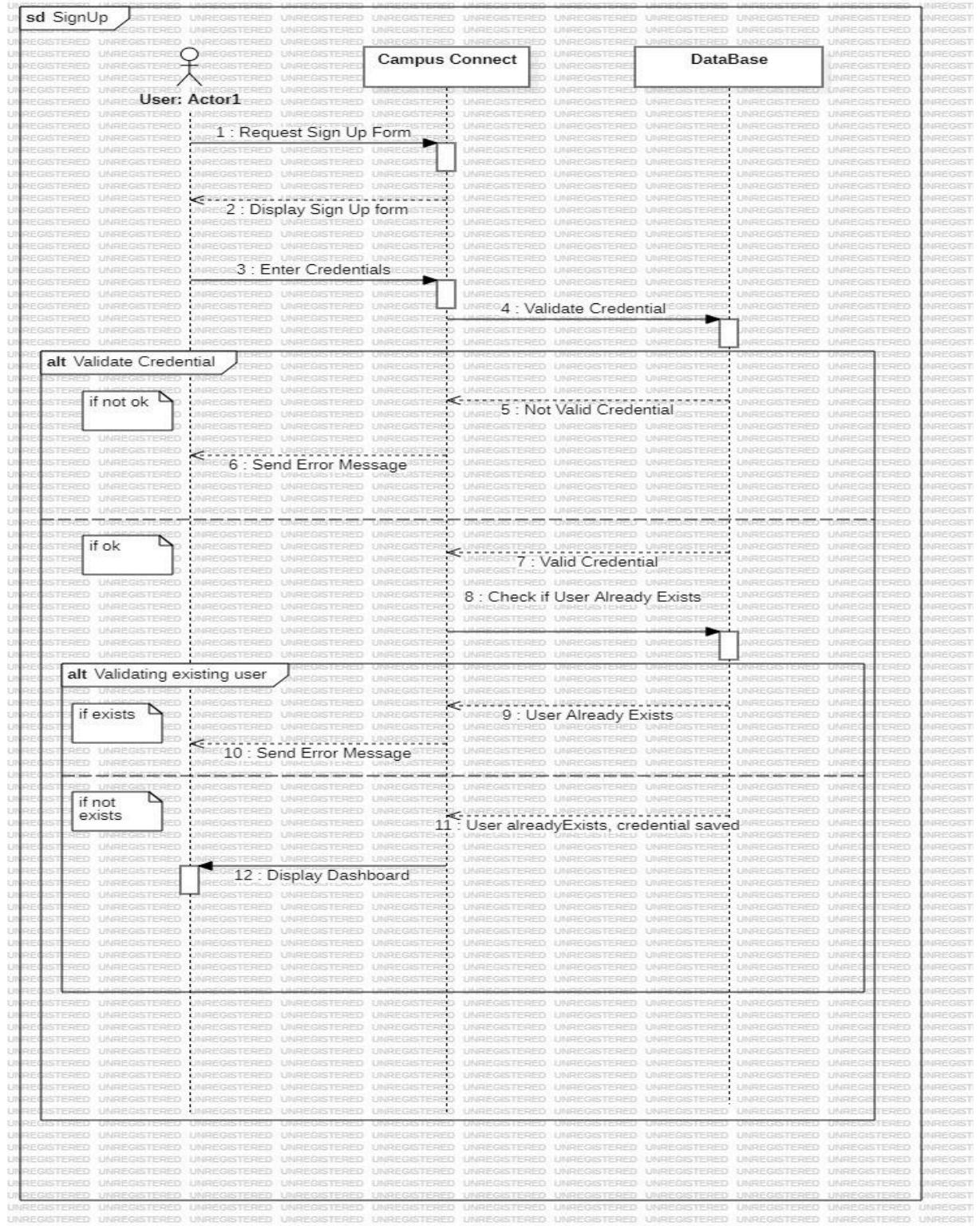
- Gérer les comptes utilisateur et les autorisations.
- Gérer les horaires des transports universitaires et le menu resto

**● Mesures de protection des données sensibles**

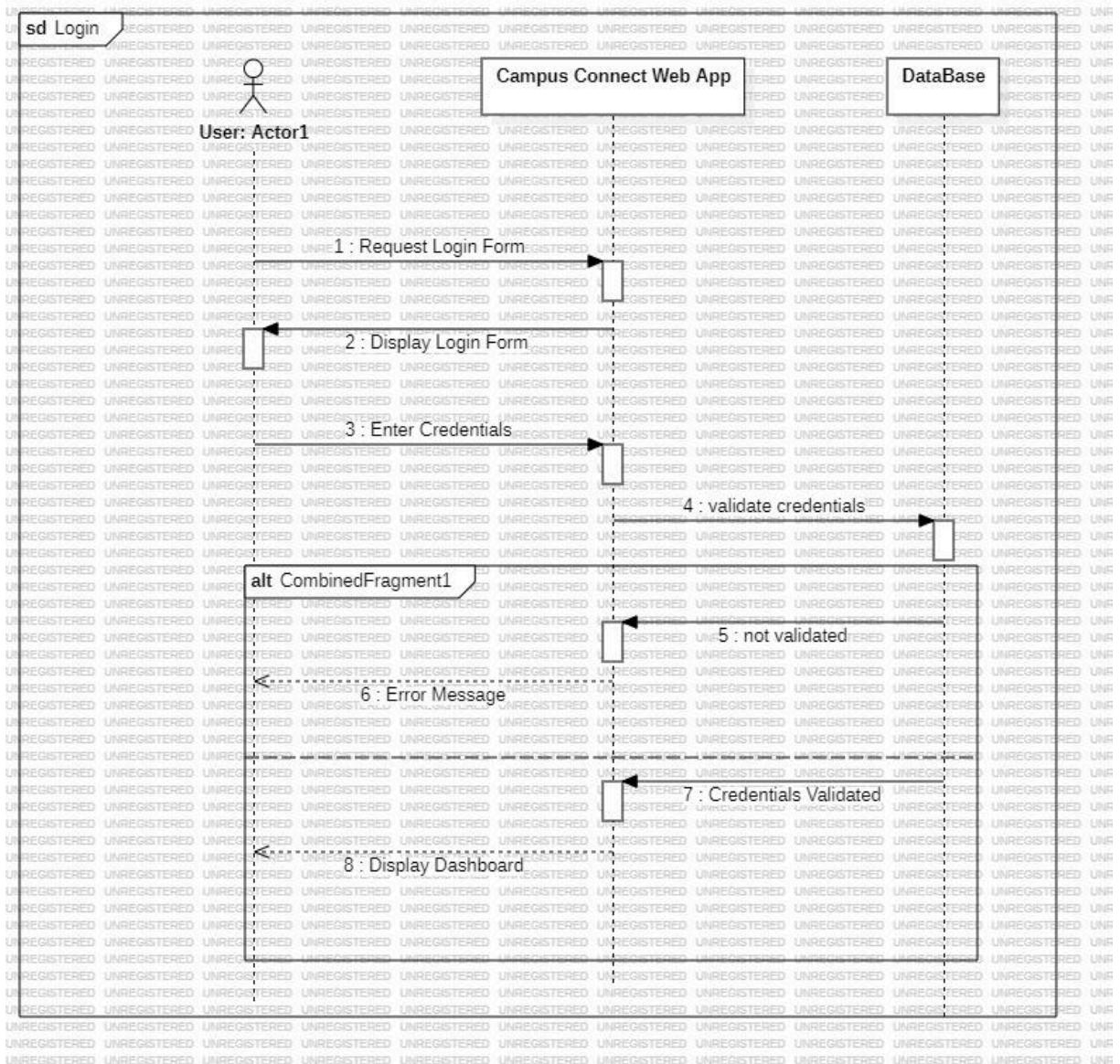
Pour assurer la protection des données sensibles, telles que les informations personnelles des utilisateurs et les données académiques, plusieurs mesures sont mises en place. Tout d'abord, l'accès aux données sensibles est strictement contrôlé en limitant l'accès uniquement aux utilisateurs autorisés. De plus, des techniques de cryptage sont utilisées pour protéger les données sensibles lorsqu'elles sont stockées dans la base de données, garantissant ainsi qu'elles ne peuvent être accédées que par des utilisateurs autorisés. Des politiques de sécurité strictes sont également mises en place pour régir le traitement et le partage des données sensibles, garantissant ainsi leur intégrité et leur confidentialité à tout moment.

## 8- Scénarios de Cas d'Utilisation

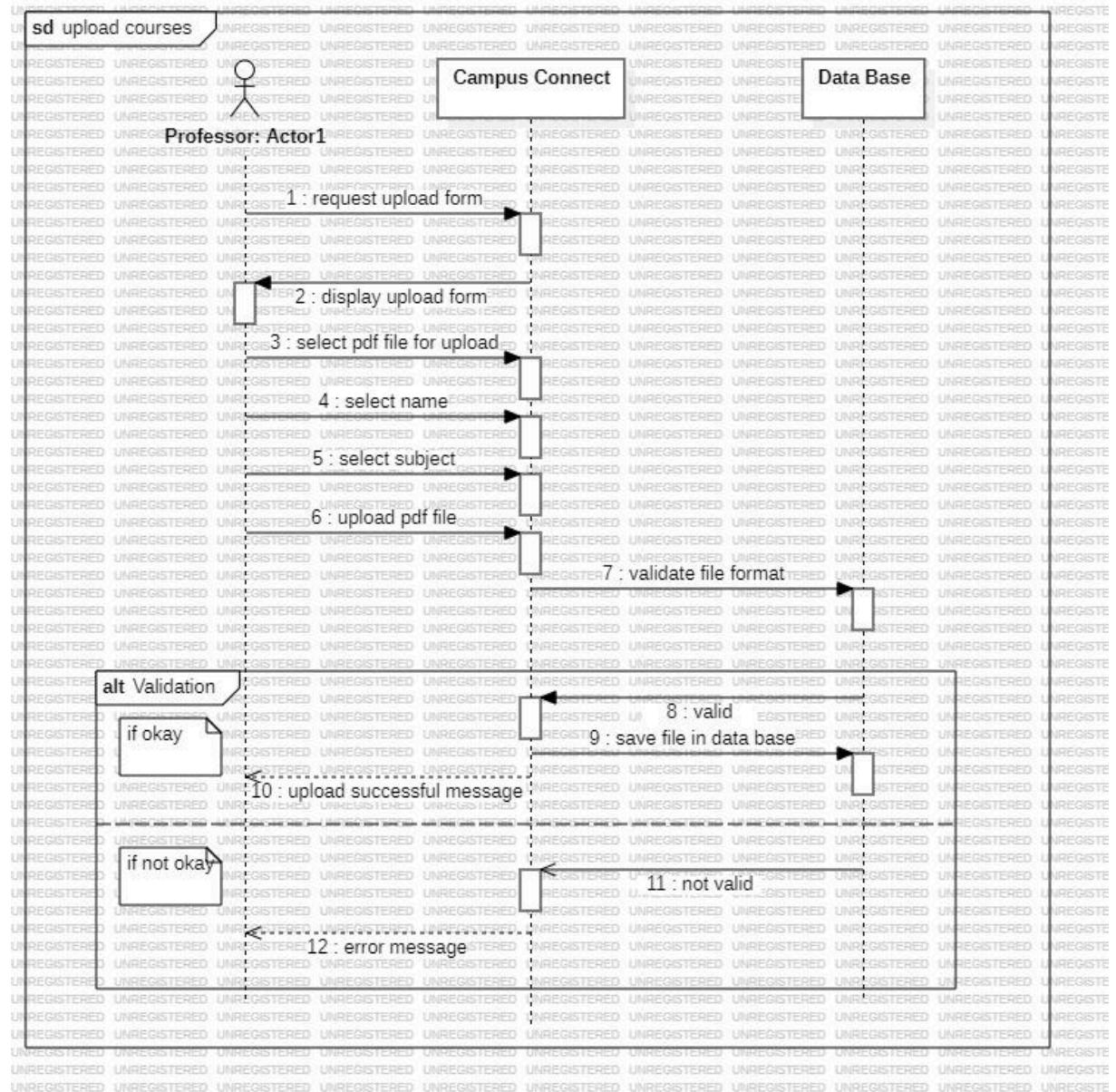
- Diagrammes de cas d'utilisation inscription



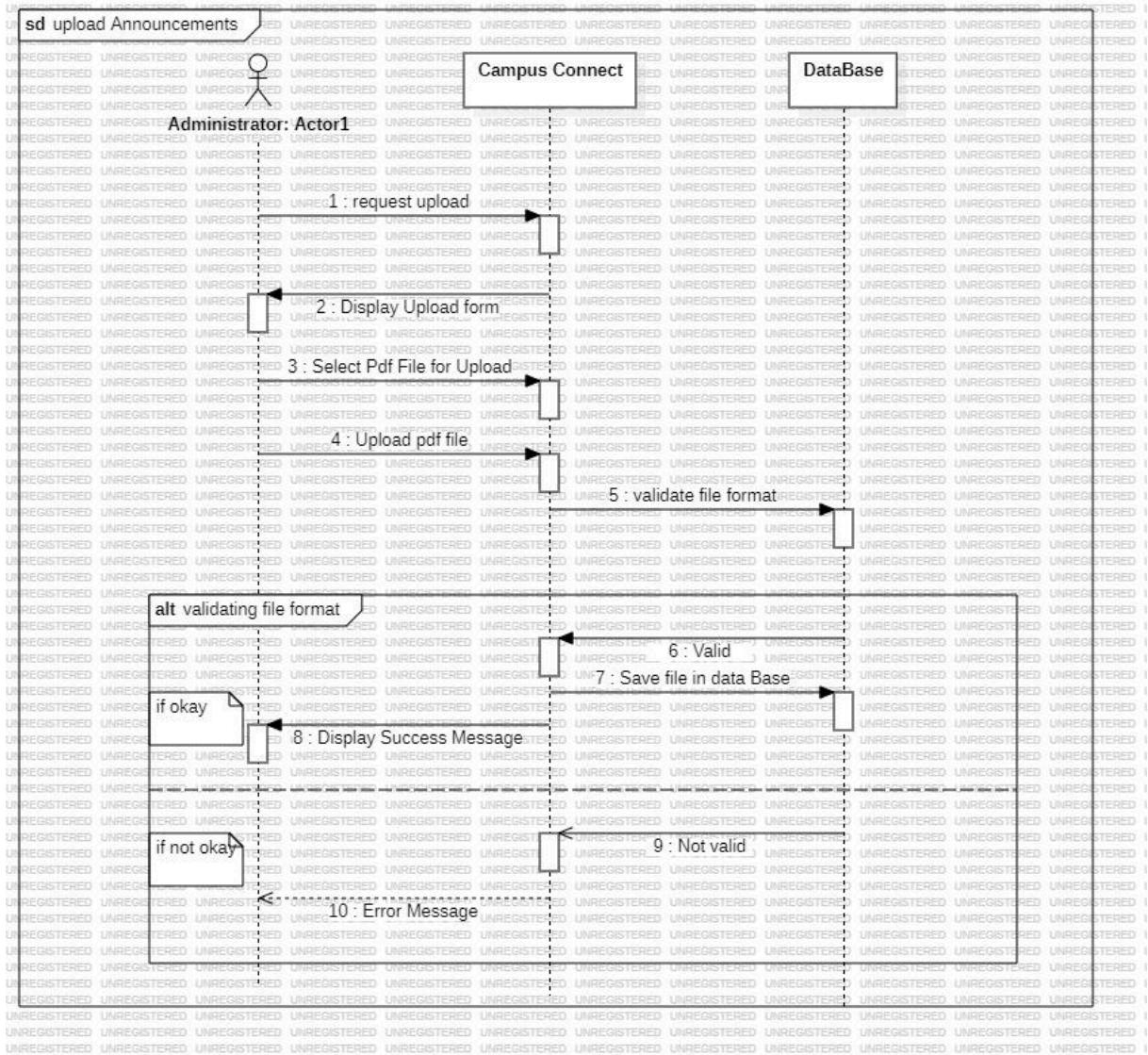
- Diagrammes de cas d'utilisation connexion



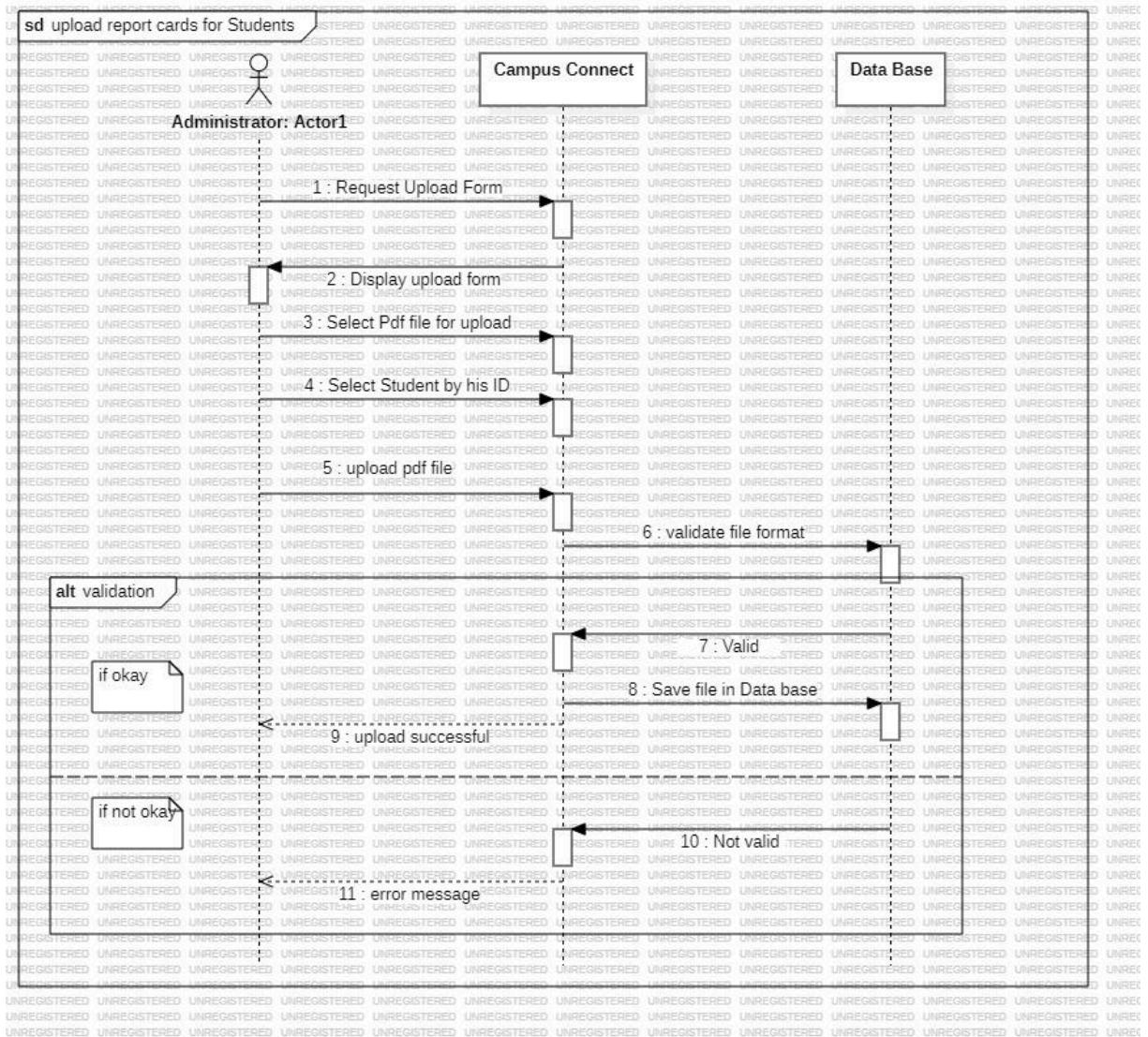
## ● Diagrammes de cas d'utilisation dépôt cours



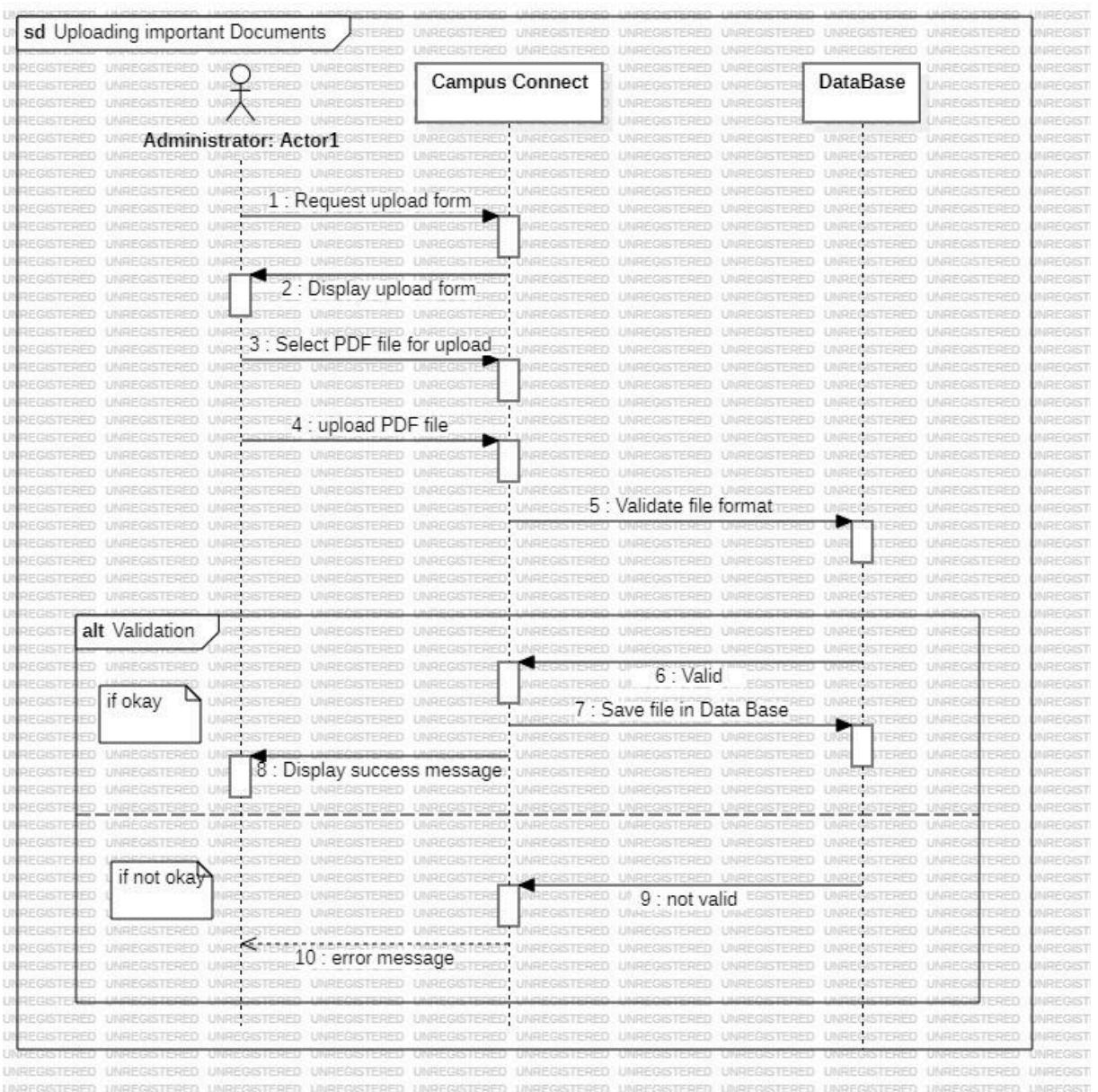
## ● Diagrammes de cas d'utilisation dépôt annonces



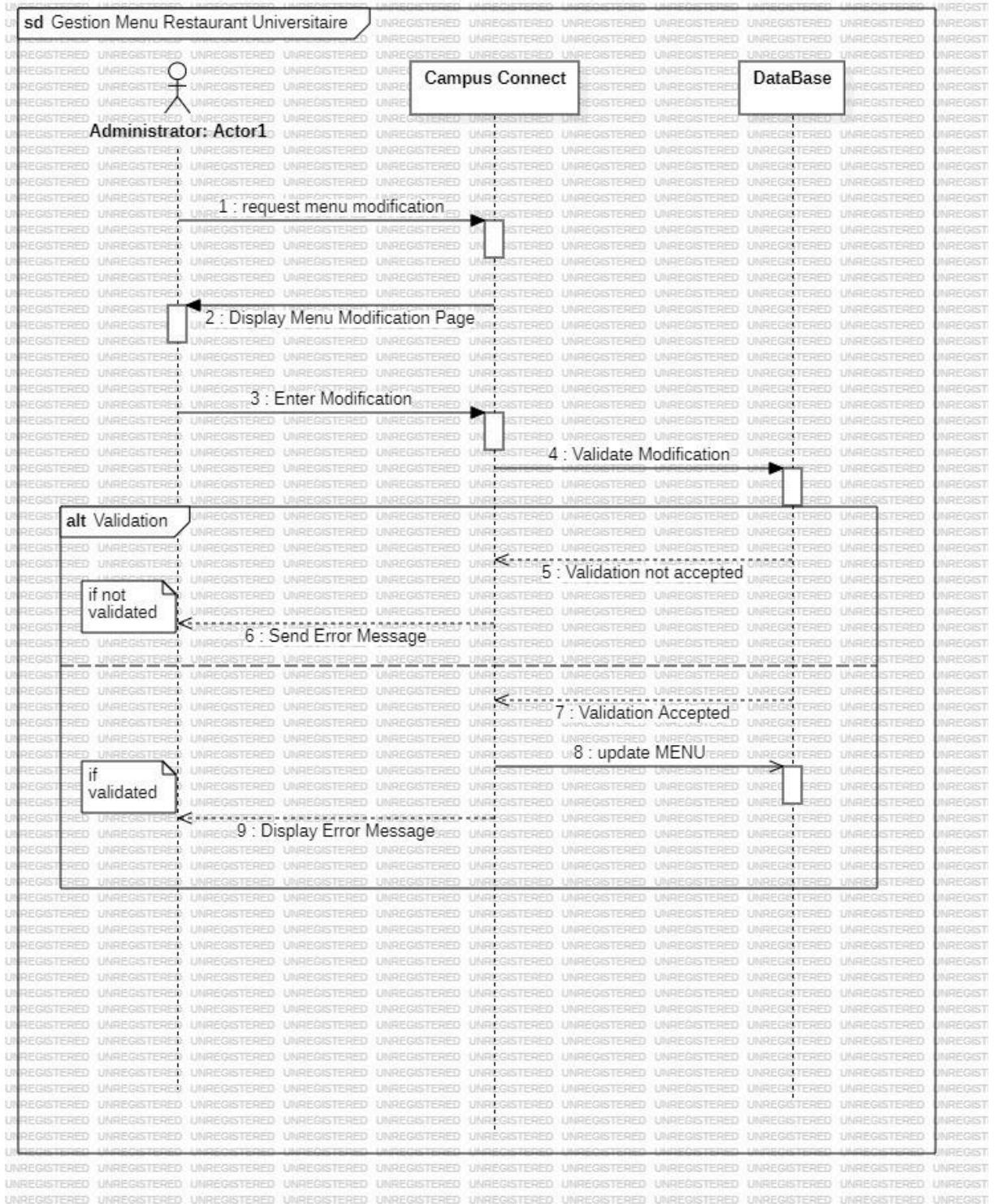
## • Diagrammes de cas d'utilisation dépôt pv



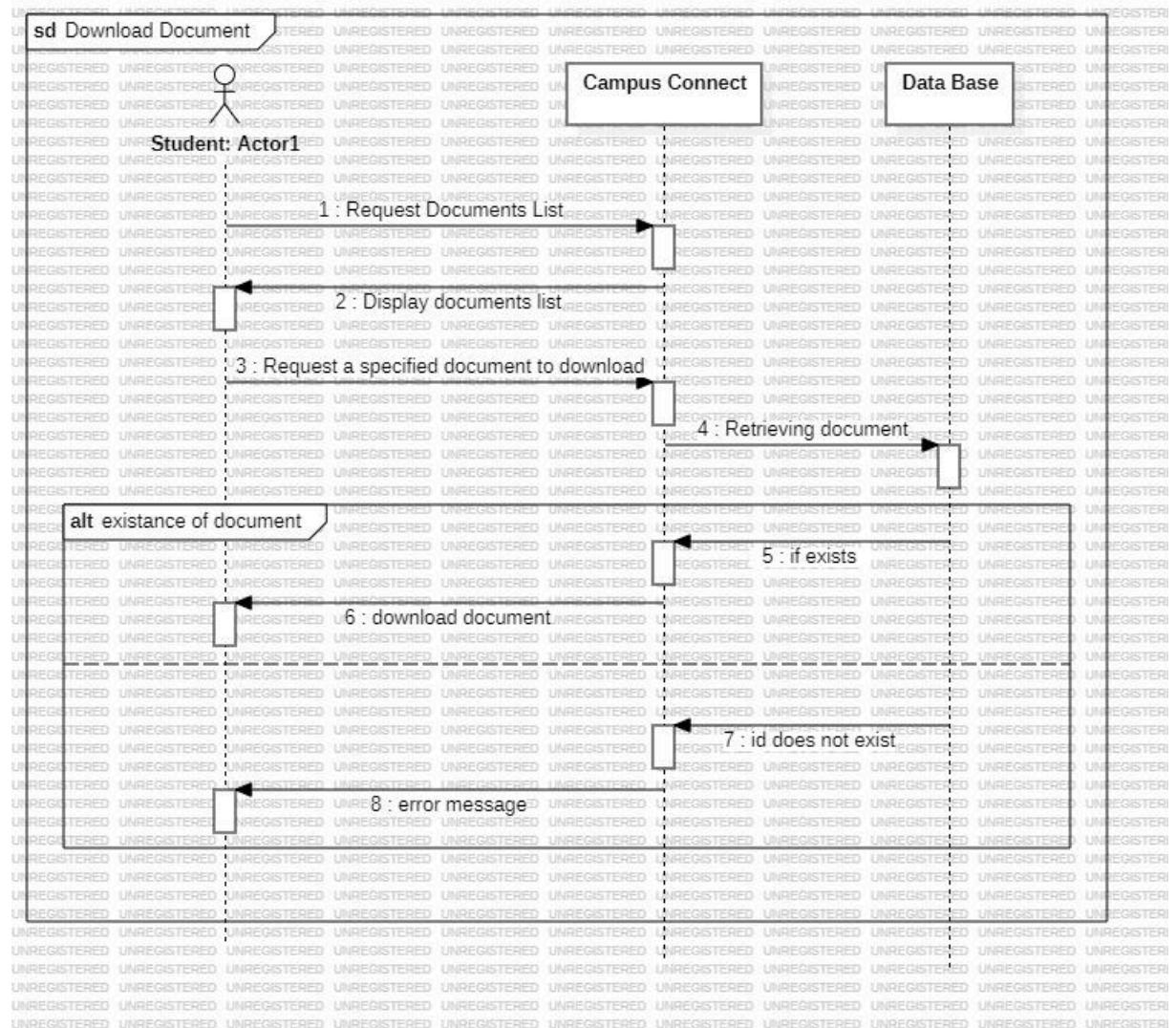
## ● Diagrammes de cas d'utilisation dépôt documents



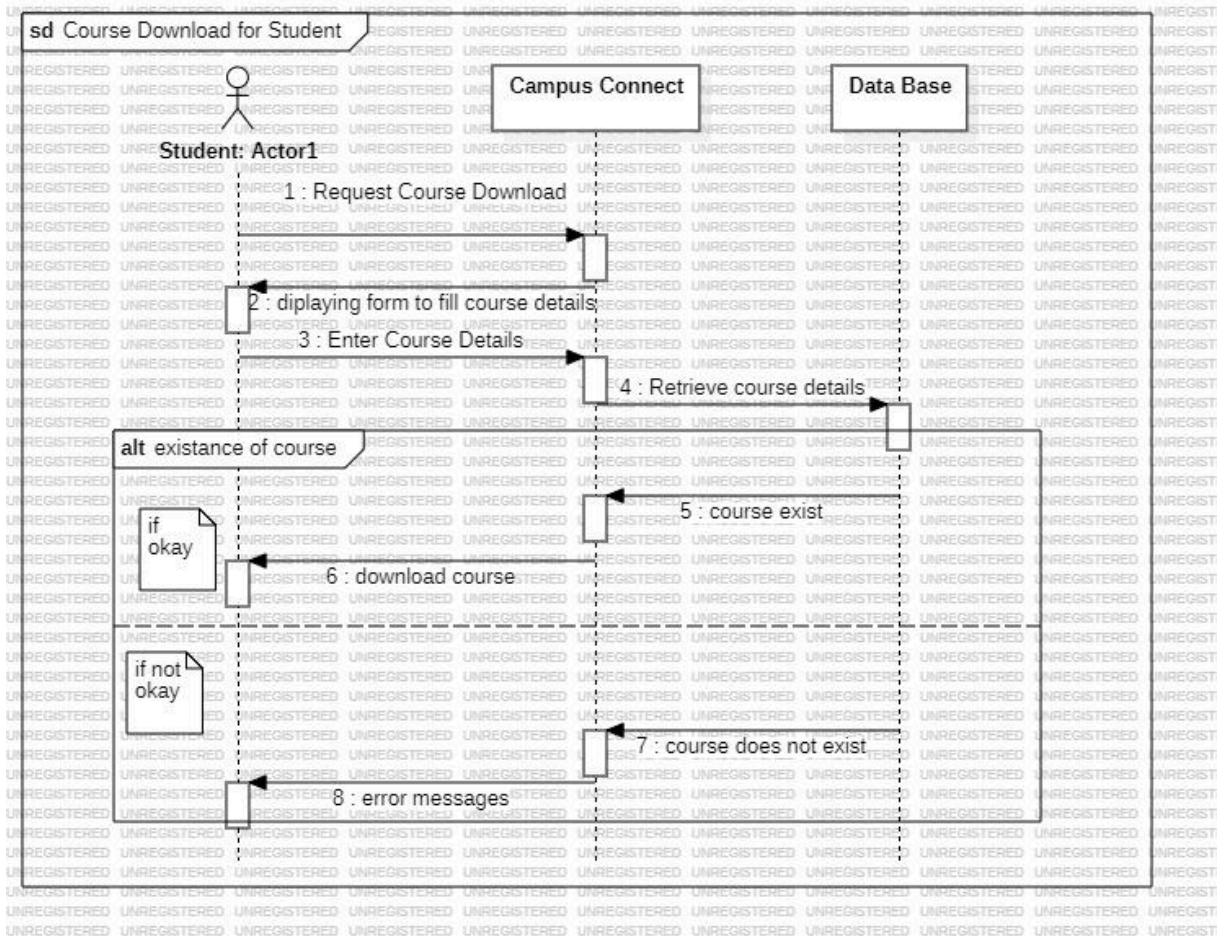
## • Diagrammes de cas d'utilisation gestion menu resto



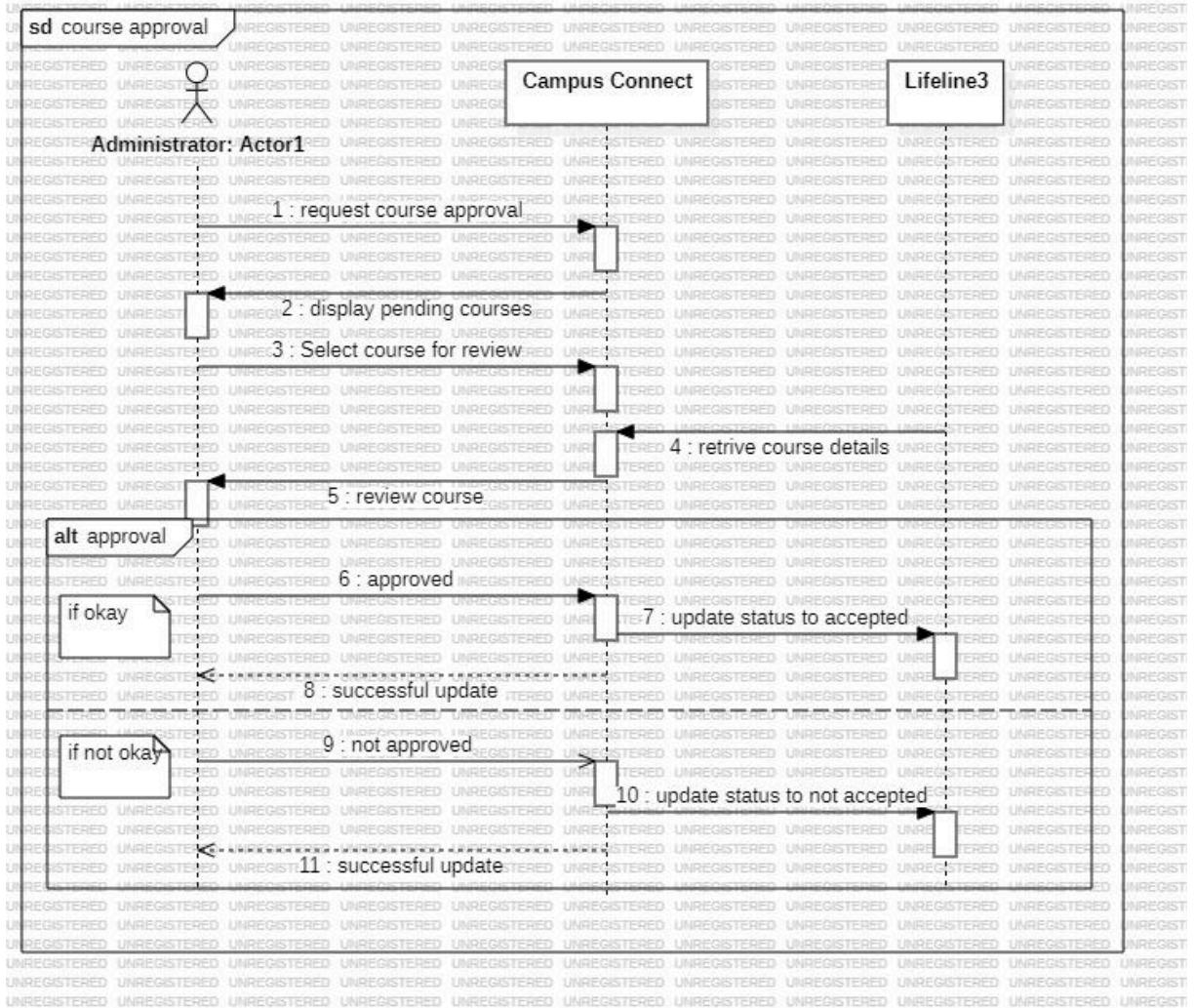
- Diagrammes de cas d'utilisation télécharger document



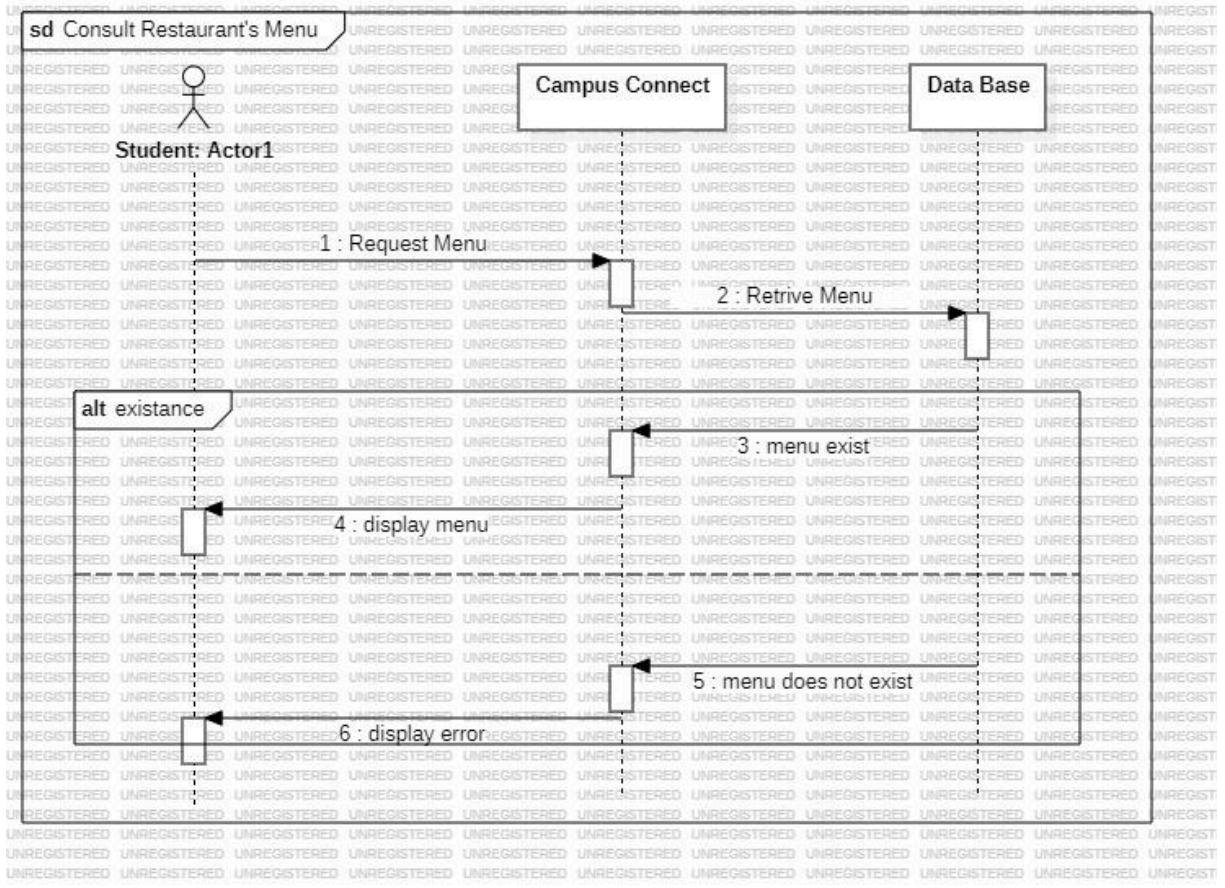
- Diagrammes de cas d'utilisation telecharger cours



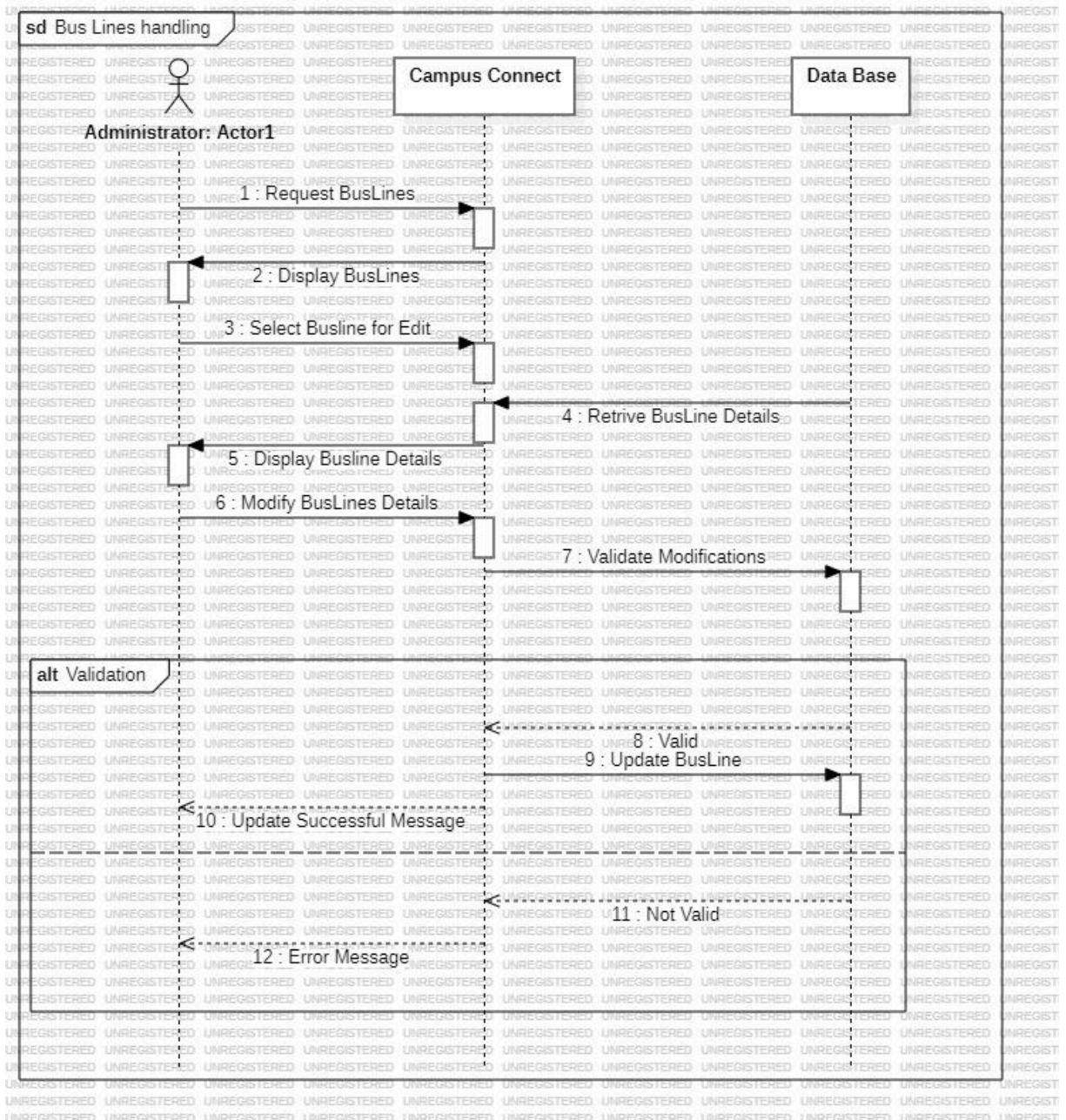
- Diagrammes de cas d'utilisation approuver cours



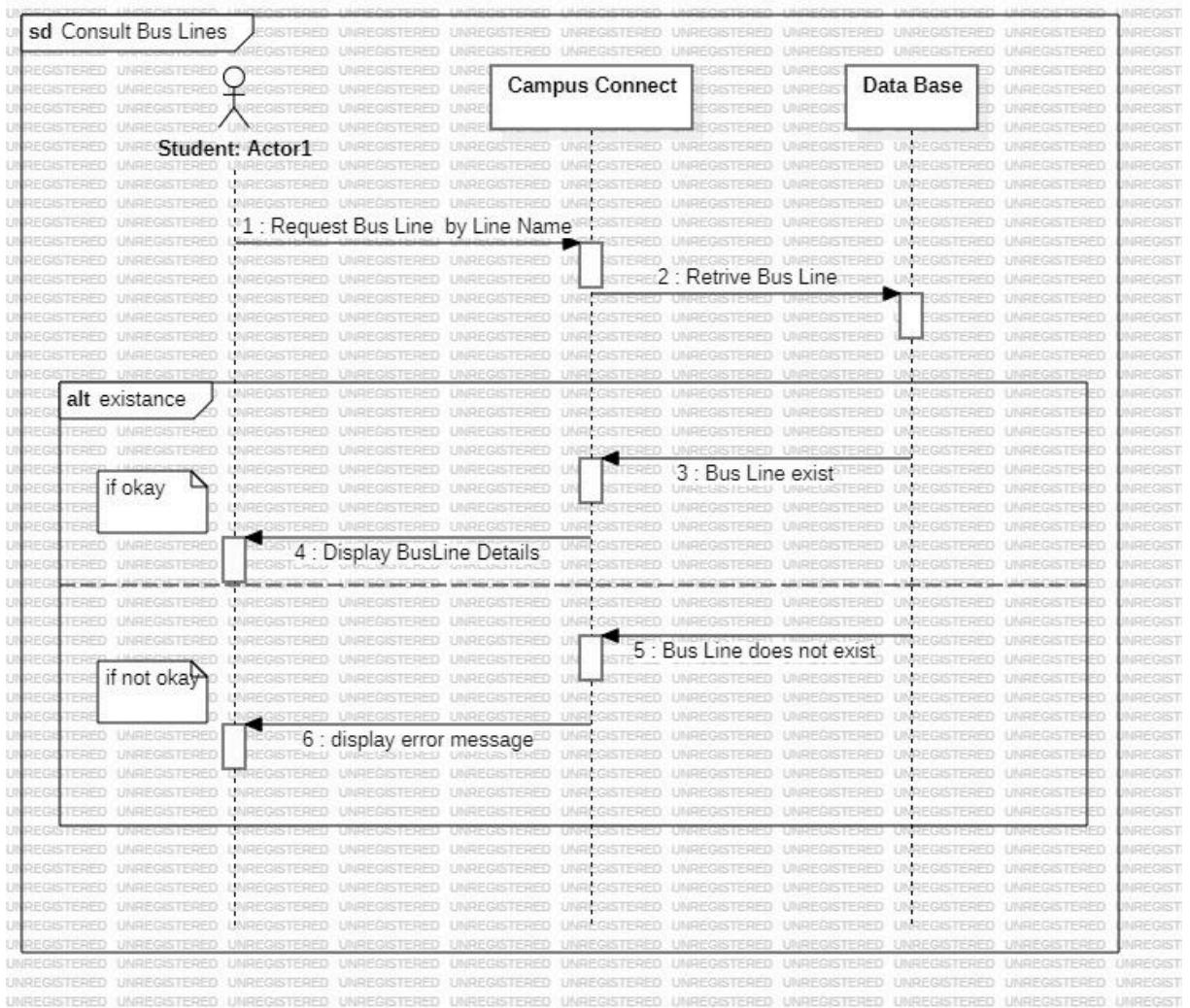
## ● Diagrammes de cas d'utilisation consulter menu resto



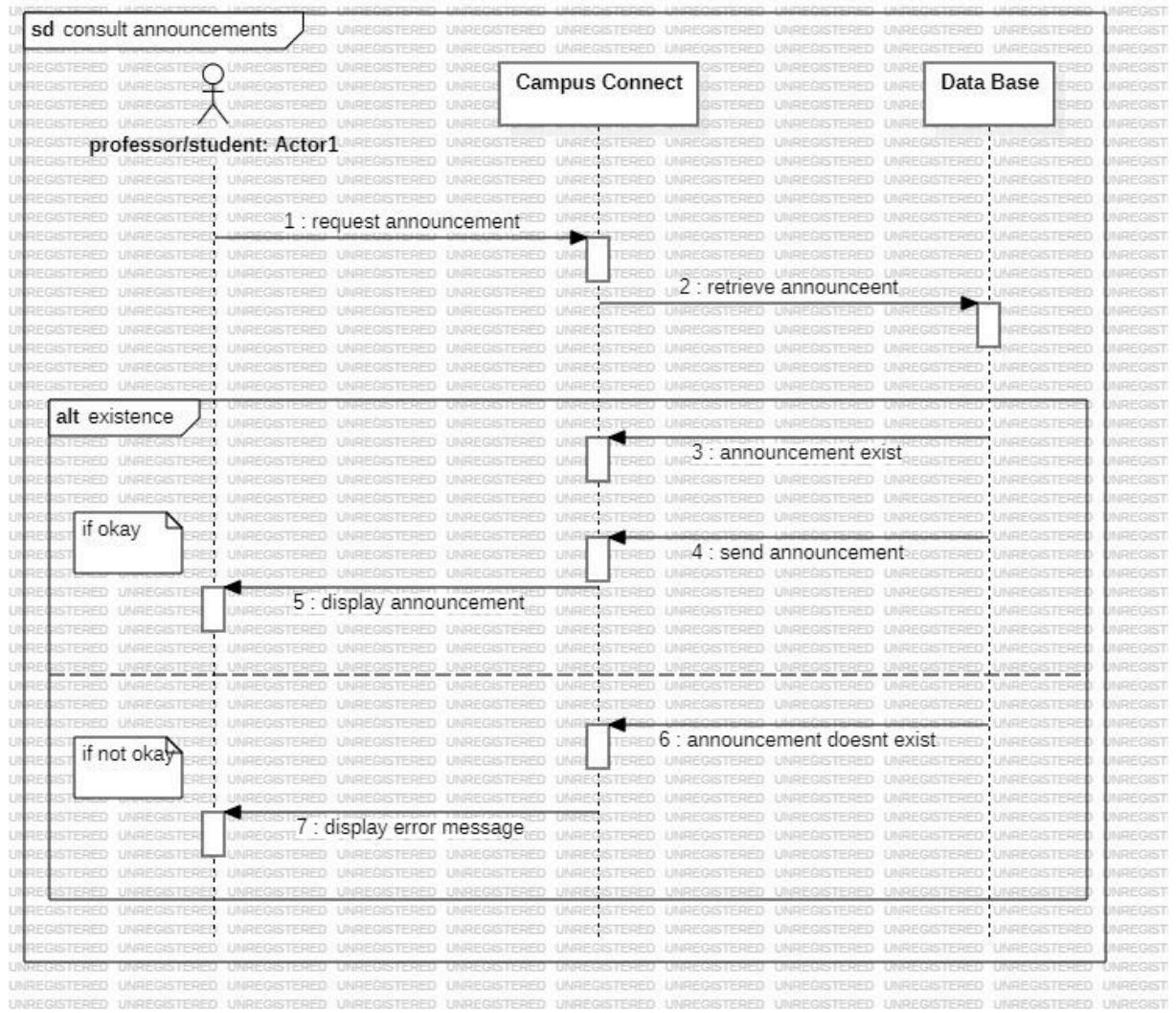
## • Diagrammes de cas d'utilisation gestion transport



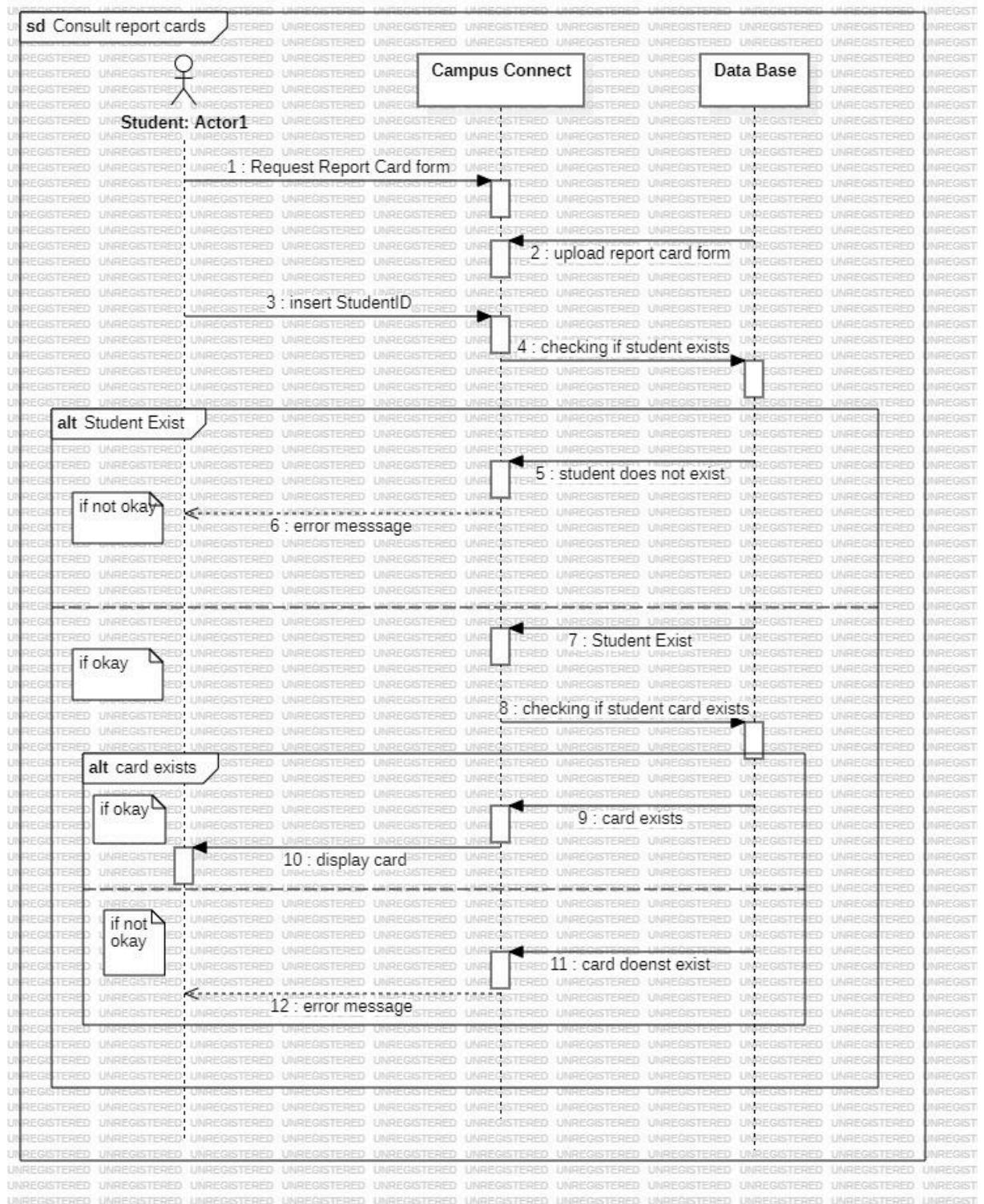
- Diagrammes de cas d'utilisation consulter transport



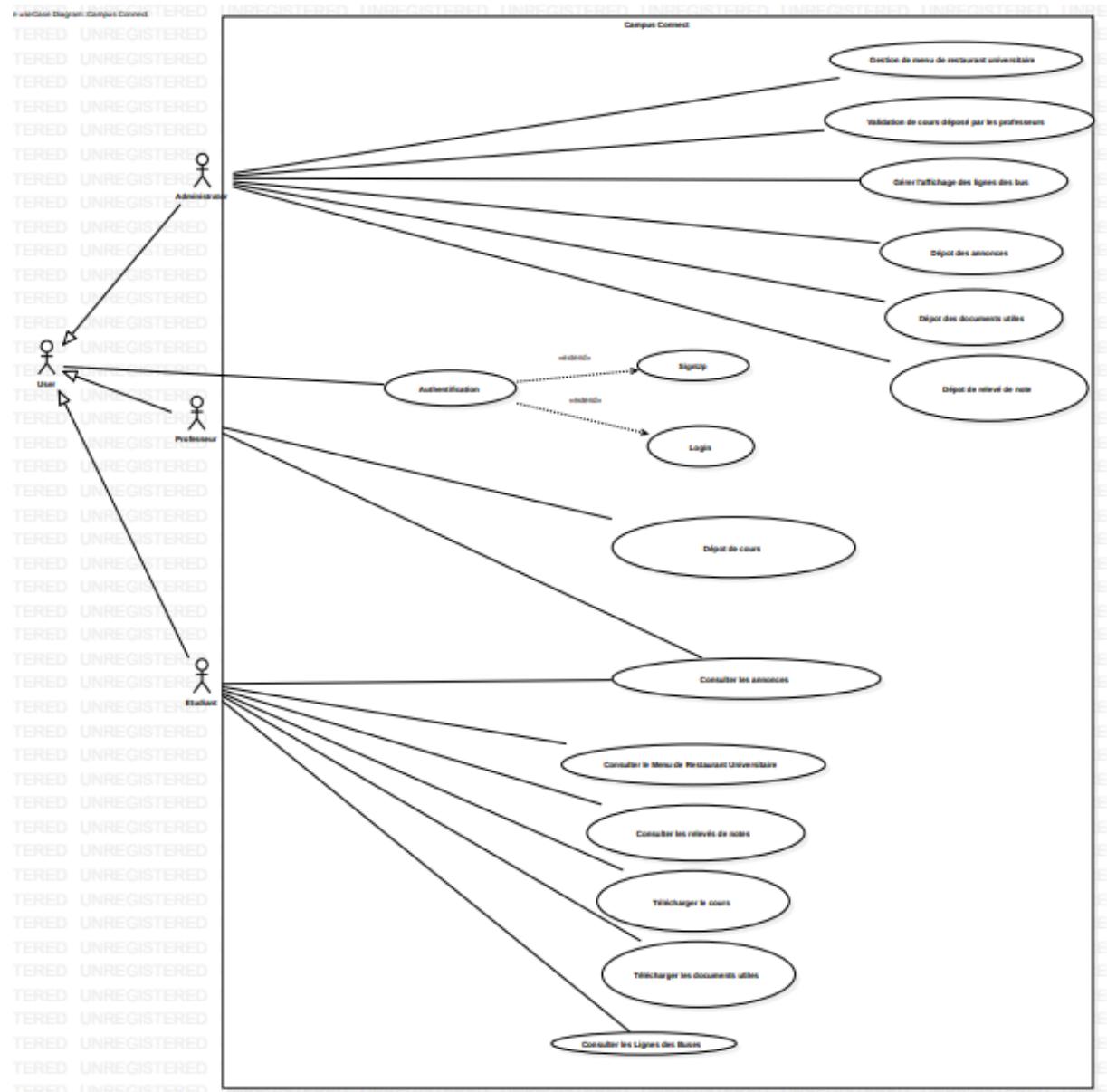
## ● Diagrammes de cas d'utilisation consulter annonces



## ● Diagrammes de cas d'utilisation consulter pv



## 9- Diagrammes de Séquence



- **Illustration des flux de contrôle entre les différents composants**

Dans notre projet "Compus Connect", chaque acteur - étudiant, enseignant et administrateur - a des fonctions et des permissions spécifiques qui déterminent leurs interactions avec la plateforme. Voici un aperçu des fonctionnalités pour chaque acteur :

**Étudiant :**

- Authentification : L'étudiant peut s'authentifier en utilisant ses identifiants.
- Consultation des cours : Accès à la liste des cours disponibles.
- Consultation des documents : Accès aux documents académiques pertinents.
- Menu du restaurant : Consultation du menu du restaurant universitaire.
- Horaires de transport : Accès aux horaires de transport.
- Téléchargement de documents : Possibilité de télécharger des documents liés aux cours.

**Enseignant :**

- Authentification : L'enseignant peut s'authentifier en utilisant ses identifiants.
- Gestion des cours : Publication, modification et suppression de cours.
- Annonces : Publication d'annonces importantes pour les étudiants.
- Menu du restaurant : Consultation du menu du restaurant universitaire.
- Horaires de transport : Accès aux horaires de transport.

**Administrateur :**

- Authentification : L'admin peut s'authentifier en utilisant ses identifiants.
- Gestion des documents : Publication, modification et suppression de documents partagés.
- Annonces : Publication d'annonces institutionnelles.
- Approuver les cours : Validation des cours proposés par les enseignants.
- Gestion des utilisateurs : Création, modification et suppression de comptes utilisateur.
- Menu du restaurant : Gestion du menu du restaurant universitaire.
- Horaires de transport : Gestion des horaires de transport.

Ces rôles et fonctions spécifiques garantissent que chaque acteur accède uniquement aux fonctionnalités qui lui sont appropriées, assurant ainsi une expérience utilisateur sécurisée et optimale sur la plateforme "Compus Connect".

## **10- Plan de Développement**

### **• Division du projet en modules ou itérations de développement**

Dans le plan de développement de notre projet "Compus Connect", nous allons diviser le projet en modules ou itérations de développement pour faciliter sa gestion et son avancement progressif. Chaque module représentera une partie spécifique de la plateforme et sera développé de manière itérative pour assurer un processus de développement efficace et agile.

**Module d'Authentification :** Développement du système d'authentification permettant à tous les utilisateurs de se connecter à la plateforme.

**Module de Gestion des Utilisateurs :** Implémentation des fonctionnalités de gestion des utilisateurs telles que la création, la modification et la suppression de comptes.

**Module de Consultation des Cours et des Documents :** Développement de fonctionnalités permettant aux étudiants de consulter les cours disponibles ainsi que les documents associés.

**Module de Gestion des Cours :** Mise en place des fonctionnalités permettant aux enseignants de publier, de modifier et de supprimer des cours.

**Module de Gestion des Annonces :** Développement des fonctionnalités d'annonces pour permettre aux enseignants et aux administrateurs de publier des informations importantes.

**Module de Gestion du Restaurant Universitaire :** Implémentation des fonctionnalités permettant de consulter le menu du restaurant universitaire et de gérer son contenu.

**Module de Gestion des Transports Universitaires :** Développement des fonctionnalités permettant de consulter les horaires des transports universitaires et de les gérer.

**Module d'Administration :** Mise en place des fonctionnalités d'administration pour permettre aux administrateurs de gérer les documents, les annonces, les cours, les utilisateurs, le menu du restaurant et les horaires de transport.

- **Diagrammes de classes et structure du code**

Nous utiliserons également des diagrammes de classes pour représenter la structure du code de manière visuelle et intuitive. Ces diagrammes permettront de définir les différentes classes et relations entre elles, ainsi que les attributs et les méthodes associés à chaque classe. Cela offrira une vue d'ensemble claire de l'architecture du code et facilitera la collaboration entre les développeurs impliqués dans le projet.

- **Calendrier de mise en œuvre de la conception**

Voici une proposition de calendrier de mise en œuvre pour la conception du projet "Compus Connect" en fonction des modules précédemment identifiés :

**Module d'Authentification :**

Durée estimée : 5 jours

Semaine de début : 7/3/2024

Semaine de fin : 12/3/2024

**Module de Gestion des Utilisateurs :**

Durée estimée : 5 jours

Semaine de début : 13/3/24

Semaine de fin : 18/3/2024

**Module de Consultation des Cours et des Documents :**

Durée estimée : 3 jours

Semaine de début : 18/3/2024

Semaine de fin : 21/3/2024

**Module de Gestion des Cours :**

Durée estimée : 5 jours

Semaine de début : 22/3/2024

Semaine de fin : 27/3/2024

**Module de Gestion des Annonces :**

Durée estimée : 3 jours

Semaine de début : 28/3/2024

Semaine de fin : 31/3/2024

### **Module de Gestion du Restaurant Universitaire /Transports Universitaires:**

Durée estimée : 5 jours

Semaine de début : 1/4/2024

Semaine de fin : 6/4/2024

### **Module d'Administration :**

Durée estimée : 6 jours

Semaine de début : 7/4/2024

Semaine de fin : 13/4/2024

## **11-Conclusion**

Dans ce chapitre, nous avons abordé la conception UML de la solution informatique pour notre projet "Compus Connect". Nous avons commencé par une introduction à la conception, soulignant l'importance de cette phase dans la transformation des exigences en solutions concrètes. Ensuite, nous avons analysé les besoins du projet, en mettant en lumière les fonctionnalités essentielles pour répondre aux attentes des utilisateurs universitaires.

Nous avons ensuite priorisé ces fonctionnalités pour guider le développement, en nous concentrant sur les aspects cruciaux tels que l'authentification, la consultation des cours, et la gestion des annonces. Après cela, nous avons proposé une architecture globale pour la solution, basée sur un modèle client-serveur avec une approche à microservices, afin d'assurer une évolutivité et une flexibilité optimales. Ensuite, nous avons discuté de la modélisation des données, en concevant le schéma de la base de données et en utilisant des diagrammes de classes pour représenter la structure du code. Nous avons également exploré les interfaces utilisateur, en décrivant la charte graphique et les maquettes pour assurer une expérience utilisateur agréable.

Dans la section sur les algorithmes et les méthodes, nous avons proposé des algorithmes clés tels que l'algorithme de recherche linéaire, le tri rapide, et l'algorithme de hachage bcrypt pour l'authentification.

En ce qui concerne la sécurité et la confidentialité, nous avons mis en place des stratégies telles que l'authentification sécurisée, le chiffrement des données et le contrôle d'accès basé sur les rôles pour garantir la protection des informations sensibles.

Enfin, nous avons défini les scénarios de cas d'utilisation et illustré les flux de contrôle entre les différents composants de la plateforme "Compus Connect". Ces composants ont été divisés en modules ou itérations de développement, avec un calendrier de mise en œuvre proposé pour guider le processus de développement.

Dans la prochaine phase de réalisation et de validation, nous allons mettre en œuvre ces concepts et travailler sur la construction effective de la plateforme "Compus Connect". Nous veillerons à ce que chaque module soit développé de manière itérative et conforme aux spécifications établies lors de la phase de conception. En parallèle, nous procéderons à des tests rigoureux pour valider le bon fonctionnement de chaque fonctionnalité et assurer la qualité globale de la solution.

# Chapitre 4 : Réalisation et validation

## **1- Introduction à la Réalisation**

- Présentation des objectifs de la phase de réalisation**

L'objectif principal de la phase de réalisation est de concrétiser la vision définie lors de la phase de conception en développant un système informatique opérationnel et conforme aux spécifications. Cette phase vise à :

- Traduire les concepts et les modèles de conception en code informatique.
- Développer et intégrer les différents modules de la plateforme "Compus Connect" de manière cohérente.
- Assurer la qualité, la performance et la sécurité du système tout au long du processus de développement.
- Collaborer étroitement avec les parties prenantes pour s'assurer que le produit final répond à leurs attentes et exigences.

- Enjeux liés à la mise en œuvre du projet**

La réalisation du projet "Compus Connect" est confrontée à plusieurs défis et enjeux :

- Respect des délais :** Il est essentiel de respecter les échéances fixées et de gérer efficacement les ressources disponibles pour garantir la livraison du projet dans les délais impartis.
- Qualité du code et maintenabilité :** Le code développé doit être robuste, évolutif et facile à maintenir afin de garantir la durabilité et la pérennité de la plateforme.
- Gestion des risques :** Identifier, évaluer et atténuer les risques potentiels tout au long du processus de réalisation est essentiel pour éviter les retards et les problèmes imprévus.
- Collaboration et communication :** Assurer une communication efficace et une collaboration étroite entre les membres de l'équipe de développement, les parties prenantes est essentiel pour garantir le succès du projet.

## **2- Environnement de Développement**

- **Description des outils, plateformes, et langages de programmation utilisés**

### **Langages de Programmation :**

#### **JavaScript :**

Utilisé pour le développement tant du côté client que serveur grâce à son large écosystème de frameworks et de bibliothèques, notamment React.js pour le front-end et Node.js pour le back-end.

#### **JSX CSS :**

Utilisés pour la création de l'interface utilisateur et la mise en forme des éléments graphiques.

### **Frameworks et Bibliothèques :**

#### **React.js :**

Framework JavaScript moderne utilisé pour la création d'interfaces utilisateur interactives et réactives.

#### **Node.js :**

Plateforme de développement JavaScript côté serveur permettant d'exécuter du code JavaScript sur le serveur.

#### **Express.js :**

Framework minimaliste pour la création d'applications web et d'APIs robustes avec Node.js.

#### **bcrypt :**

Bibliothèque de hachage de mot de passe sécurisée utilisée pour l'authentification des utilisateurs.

#### **MongoDB :**

Base de données NoSQL utilisée pour stocker les données de l'application.

**Postman:**

Une plateforme de collaboration pour le développement d'API, pour tester et documenter nos API. Son intégration a facilité les tests automatisés et le débogage, renforçant ainsi la qualité de notre code.

**Outils de Développement :****Visual Studio Code :**

Éditeur de code léger et puissant, offrant une multitude de fonctionnalités pour le développement web.

**Git :**

Système de contrôle de version distribué utilisé pour la gestion du code source et la collaboration entre les développeurs.

**GitHub :**

Plateforme de développement collaboratif basée sur Git, utilisée pour héberger les dépôts de code, gérer les projets et faciliter la collaboration.

- **Justification des choix technologiques**

**Évolutivité et Flexibilité :**

Les langages et frameworks sélectionnés offrent une grande flexibilité et permettent une évolutivité aisée de la plateforme, ce qui est essentiel pour répondre aux besoins changeants des utilisateurs et de l'environnement universitaire.

**Large Adoption et Support Communautaire :**

JavaScript, React.js et Node.js sont largement adoptés par la communauté de développement, ce qui garantit un support étendu, une documentation abondante et un écosystème de modules complémentaires robuste.

#### **Performance et Réactivité :**

React.js offre des performances élevées et une expérience utilisateur réactive grâce à sa gestion efficace du DOM virtuel, ce qui est essentiel pour fournir une interface utilisateur fluide et intuitive.

#### **Sécurité :**

L'utilisation de bibliothèques comme bcrypt pour le hachage sécurisé des mots de passe contribue à renforcer la sécurité de la plateforme en protégeant les données sensibles des utilisateurs.

#### **Adaptabilité aux Besoins Spécifiques :**

Les technologies choisies permettent une personnalisation et une adaptation précises aux besoins spécifiques de la plateforme "Compus Connect", offrant ainsi une solution sur mesure répondant aux exigences fonctionnelles et non fonctionnelles du projet.

En combinant ces critères, nous avons élaboré un environnement de développement robuste et performant, capable de soutenir efficacement la réalisation de la plateforme "Compus Connect" dans son ensemble.

### **3- Développement des Modules**

#### **• Les modules identifiés dans la conception**

Chaque module identifié dans la phase de conception a été développé en suivant une approche itérative et modulaire. Des équipes dédiées ont travaillé sur chaque module, en se concentrant sur la mise en œuvre des fonctionnalités spécifiques assignées à chacun. Les détails de la création des modules comprenaient la définition des interfaces, l'implémentation des fonctionnalités, ainsi que la gestion des dépendances entre les modules pour assurer une intégration harmonieuse.

#### **Module d'Authentification :**

Ce module a été conçu pour gérer les processus d'inscription et de connexion des utilisateurs, en assurant la sécurité et la confidentialité des informations d'identification.

**Module de Gestion des Utilisateurs :**

Ce module a permis la création, la modification et la suppression des comptes utilisateur, ainsi que la gestion des autorisations et des rôles.

**Module de Consultation des Cours et des Documents :**

Ce module a fourni aux utilisateurs la possibilité de consulter les cours disponibles et les documents associés, en garantissant un accès rapide et facile à ces ressources académiques.

**Module de Gestion des Cours :**

Ce module a facilité la publication, la modification et la suppression des cours par les enseignants, assurant ainsi la mise à jour régulière du contenu pédagogique.

**Module de Gestion des Annonces :**

Ce module a permis la publication d'annonces importantes pour les utilisateurs, offrant un moyen efficace de communiquer des informations institutionnelles.

**Module de Gestion du Restaurant Universitaire :**

Ce module a permis la consultation du menu du restaurant universitaire et la gestion de son contenu, offrant aux utilisateurs une visibilité sur les repas disponibles.

**Module de Gestion des Transports Universitaires :**

Ce module a fourni aux utilisateurs les horaires des transports universitaires et les informations pertinentes sur les trajets, facilitant ainsi les déplacements sur le campus.

- **Utilisation de UML pour guider l'implémentation**

UML a été utilisé comme un outil précieux pour guider l'implémentation des modules. Les diagrammes de classes, de séquence et d'activités ont fourni une représentation visuelle claire de la structure, du comportement et des interactions des composants du système. Cela a permis aux développeurs de mieux comprendre les spécifications fonctionnelles et de concevoir des solutions techniques efficaces et cohérentes.

## 4- Tests et Débogage

- **Stratégies de tests appliquées aux différents modules**

Chaque module a été soumis à des tests unitaires et d'intégration pour vérifier son bon fonctionnement individuel ainsi que son interaction avec les autres modules. Des tests fonctionnels ont également été effectués pour s'assurer que les fonctionnalités répondent aux exigences spécifiées dans la phase de conception. En outre, des tests de performance ont été réalisés pour évaluer les performances du système dans des conditions réelles d'utilisation.

- **Processus de débogage pour résoudre les erreurs identifiées**

Lors de la phase de débogage, les erreurs identifiées ont été systématiquement enregistrées, analysées et résolues. Les développeurs ont utilisé des outils de suivi des bogues pour gérer efficacement les rapports d'erreurs et suivre l'état de résolution. Des techniques de débogage telles que l'inspection du code, l'utilisation de journaux de débogage et la reproduction des scénarios d'erreur ont été employées pour identifier et corriger les problèmes de manière efficace et efficiente. Une fois les erreurs corrigées, les modules ont été soumis à des tests supplémentaires pour garantir que les corrections étaient satisfaisantes et n'avaient pas introduit de nouveaux problèmes.

## 5- Intégration des Composants

- **Mise en œuvre de l'architecture globale du système**

La mise en œuvre de l'architecture globale du système a été réalisée en suivant les principes définis dans la phase de conception. Les composants front-end, back-end et la base de données ont été déployés selon les spécifications techniques, garantissant ainsi une infrastructure robuste et évolutive pour la plateforme. Les microservices ont été configurés et intégrés pour assurer une interaction fluide et efficace entre les différents modules, favorisant ainsi la scalabilité et la flexibilité du système.

- **Coordination des différents modules pour assurer leur interopérabilité**

La coordination des différents modules a été réalisée en établissant des interfaces claires et bien définies entre les composants. Les équipes de développement ont travaillé en étroite collaboration pour s'assurer que les interfaces étaient respectées

et que les échanges de données se faisaient de manière harmonieuse. Des tests d'intégration ont été effectués pour valider l'interopérabilité des modules, en vérifiant notamment que les communications entre les composants étaient correctement établies et que les données étaient transmises de manière fiable et cohérente. En cas de problèmes d'interopérabilité, des ajustements ont été apportés aux interfaces ou aux fonctionnalités des modules pour résoudre les conflits et assurer un fonctionnement harmonieux du système dans son ensemble.

## 6- Validation et Vérification

- **Évaluation de la solution par rapport aux besoins initiaux**

La validation de la solution a été réalisée en évaluant sa capacité à répondre aux besoins initiaux définis dans la phase d'analyse des besoins. Les fonctionnalités de la plateforme ont été testées en situation réelle pour vérifier si elles étaient conformes aux attentes des utilisateurs finaux. Cette évaluation a permis de valider la capacité de la solution à répondre efficacement aux attentes des utilisateurs et à fournir une expérience utilisateur optimale.

- **Vérification de la conformité aux spécifications définies dans la conception**

La vérification de la conformité aux spécifications définies dans la phase de conception a été réalisée en comparant la solution implémentée aux modèles UML et aux spécifications techniques établis. Des tests de conformité ont été effectués pour vérifier si les fonctionnalités développées étaient en accord avec les diagrammes de classes, les diagrammes de séquence et les diagrammes de cas d'utilisation définis dans la conception. Les résultats de ces tests ont permis de s'assurer que la solution était conforme aux spécifications établies et qu'elle respectait les normes et les standards de qualité définis pour le projet.

## 7- Optimisation des Performances

- **Identification des points d'optimisation éventuels**

L'identification des points d'optimisation potentiels a été réalisée en effectuant des analyses approfondies des performances du système. Cela a impliqué la collecte de données sur les temps de réponse, les temps de chargement des pages, l'utilisation des ressources système, et d'autres métriques pertinentes. Les zones présentant des

goulots d'étranglement ou des performances insatisfaisantes ont été identifiées pour cibler les efforts d'optimisation.

- **Actions entreprises pour améliorer les performances du système**

Pour améliorer les performances du système, plusieurs actions ont été entreprises, notamment l'optimisation du code, la réduction des requêtes réseau, la mise en cache des données fréquemment utilisées, et l'amélioration de l'architecture logicielle. Des techniques telles que la compression des ressources statiques, le chargement asynchrone des données, et l'utilisation de technologies de mise en cache ont été mises en œuvre pour réduire les temps de chargement et améliorer la réactivité du système. De plus, des tests de charge ont été effectués pour évaluer les performances du système dans des conditions de charge élevée et identifier les éventuels points de défaillance. Ces actions ont permis d'optimiser les performances du système et d'offrir une expérience utilisateur fluide et réactive.

## 8- Documentation Technique

- **Rédaction d'une documentation détaillée pour faciliter la maintenance**

### **Architecture du Système :**

Le système "Compus Connect" est basé sur une architecture client-serveur, avec une application web côté client et un ensemble de microservices côté serveur. Du côté client, l'application web offre une interface utilisateur conviviale permettant aux utilisateurs d'accéder aux fonctionnalités de la plateforme via leur navigateur web. Côté serveur, les microservices interagissent pour fournir les différentes fonctionnalités du système, telles que l'authentification, la gestion des utilisateurs, la consultation des cours, la publication d'annonces, la gestion du menu du restaurant universitaire, et la gestion des horaires de transport. Cette architecture modulaire permet une évolutivité facile et une maintenance efficace du système.

### **Fonctionnalités et Interfaces :**

Chaque module offre un ensemble spécifique de fonctionnalités et d'interfaces pour interagir avec le système. Ces fonctionnalités sont conçues pour répondre aux besoins variés des utilisateurs de la plateforme "Compus Connect" et sont implémentées de manière à garantir une expérience utilisateur optimale.

- **Utilisation de UML pour compléter la documentation technique**

Des modèles UML ont été utilisés (chapitre 3) pour compléter la documentation textuelle, offrant des représentations visuelles des différentes parties du système. Les diagrammes de classes, de séquence, de composants, et de déploiement ont été créés pour illustrer la structure, le comportement et l'architecture du système de manière claire et compréhensible.

## 9- Formation et Transfert de Compétences

- **Formation des utilisateurs finaux et de l'équipe de maintenance**

La formation des utilisateurs finaux et de l'équipe de maintenance revêt une importance cruciale pour garantir une adoption fluide et efficace du système "Compus Connect" ainsi que pour assurer sa maintenance continue. Les sessions de formation pour les utilisateurs finaux mettront l'accent sur la familiarisation avec l'interface et les fonctionnalités clés, tandis que la formation de l'équipe de maintenance leur fournira une compréhension approfondie de l'architecture, des composants et des outils de dépannage, afin de garantir un fonctionnement optimal du système et une réponse rapide aux incidents.

- **Transfert des compétences nécessaires à la gestion du projet**

Le transfert des compétences est un élément clé pour assurer une transition harmonieuse vers la gestion autonome du projet "Compus Connect". Pour faciliter ce processus, des sessions de transfert de compétences seront organisées pour l'équipe de gestion du projet, couvrant des aspects tels que la planification et la coordination des tâches, la communication avec les parties prenantes, la gestion des ressources et des délais, ainsi que l'évaluation des risques et des opportunités. Ces sessions permettront à l'équipe de gestion du projet de prendre en charge efficacement toutes les activités liées au projet, en garantissant une exécution réussie et une livraison dans les délais des différentes phases du projet. De plus, des documents de référence et des guides pratiques seront fournis pour aider l'équipe de gestion du projet dans ses responsabilités quotidiennes et assurer la continuité des opérations tout au long du cycle de vie du projet.

## **10- Conclusion du Chapitre**

- Bilan de la réalisation et de la validation**

La réalisation et la validation du projet "Compus Connect" ont représenté une étape cruciale dans son développement. Grâce à un travail acharné et à une collaboration étroite, nous avons réussi à concrétiser la vision initiale en une solution logicielle fonctionnelle et répondant aux besoins de la communauté universitaire. Les tests rigoureux, les processus de débogage et les phases d'intégration ont permis de garantir la qualité et la fiabilité de la solution, validant ainsi son adéquation avec les spécifications définies lors de la conception.

- Préparation à la phase de déploiement et d'exploitation du projet**

À présent, alors que nous clôturons la phase de réalisation et de validation, notre attention se tourne vers le déploiement et l'exploitation du projet "Compus Connect". Nous nous préparons à mettre en place les infrastructures nécessaires, à former les utilisateurs finaux et à assurer une transition en douceur vers l'utilisation opérationnelle de la solution. Des stratégies de déploiement bien planifiées et une gestion efficace des ressources seront essentielles pour garantir le succès de cette phase. Nous sommes confiants que notre solution apportera une valeur ajoutée significative à la communauté universitaire et nous sommes impatients de voir son impact positif une fois déployée.

## Conclusion Générale et Perspectives

La conclusion de ce rapport marque la fin d'un parcours intense et stimulant dans le développement du projet "Compus Connect". Au cours de ce projet, plusieurs défis ont été identifiés et relevés, notamment la nécessité de répondre à une variété de besoins au sein de la communauté universitaire, tout en garantissant la sécurité et la convivialité de la plateforme.

Grâce à une analyse approfondie des besoins, une conception UML rigoureuse et une mise en œuvre méthodique, nous avons pu élaborer une solution logicielle robuste et fonctionnelle. L'utilisation de méthodes agiles et de bonnes pratiques de développement a permis de gérer efficacement le processus de développement et d'assurer une progression régulière du projet.

Sur le plan technique, nous avons acquis une expérience précieuse dans l'utilisation d'outils et de technologies modernes telles que React.js, Node.js, MongoDB et Postman. La collaboration au sein de l'équipe et la communication transparente ont été des éléments clés de notre réussite, nous permettant de surmonter les obstacles et d'atteindre nos objectifs.

Sur le plan métier, nous avons approfondi notre compréhension des besoins des utilisateurs et des processus liés à la gestion d'une plateforme numérique dans un environnement universitaire. Nous avons également développé nos compétences en matière de documentation technique, de tests et de débogage, ce qui nous sera certainement bénéfique dans nos futures entreprises.

En termes de perspectives, le projet "Compus Connect" offre un potentiel d'extension et d'amélioration continu. Des fonctionnalités supplémentaires telles que la mise en place de notifications push, l'intégration de modules d'apprentissage en ligne, ou l'optimisation de la gestion des transports universitaires pourraient être envisagées pour enrichir davantage l'expérience des utilisateurs.

En conclusion, le projet "Compus Connect" représente une étape importante dans notre parcours académique et professionnel. Il nous a permis de mettre en pratique nos connaissances théoriques, d'acquérir de nouvelles compétences et de collaborer efficacement en équipe. Nous sommes fiers du résultat obtenu et confiants dans le potentiel de notre solution pour répondre aux besoins de la communauté universitaire.

## Résumé

---

Le rapport présente le développement et la mise en œuvre du système "Compus Connect", une plateforme destinée à faciliter la communication et la gestion des activités au sein du campus universitaire.

Le contexte du projet, les objectifs et les enjeux sont exposés, mettant en lumière la nécessité d'une solution centralisée pour répondre aux besoins divers des étudiants, du corps professoral et du personnel administratif.

**Conception du Système :** Une analyse détaillée des besoins est effectuée, conduisant à l'identification des fonctionnalités essentielles et à la conception d'une architecture modulaire. Des diagrammes UML sont utilisés pour visualiser la structure et le comportement du système.

Le développement itératif des modules est décrit, mettant l'accent sur la collaboration entre les équipes et l'intégration des fonctionnalités spécifiques à chaque module. Les tests, la débogage et l'intégration des composants sont également abordés.

Des stratégies d'optimisation sont explorées pour améliorer l'efficacité et la réactivité du système, avec une attention particulière portée aux points critiques.

Une documentation technique pour faciliter la maintenance est soulignée, avec l'utilisation de diagrammes UML pour compléter les explications textuelles.

Des plans de formation pour les utilisateurs finaux et l'équipe de maintenance sont présentés, soulignant l'importance de la connaissance approfondie du système pour assurer son utilisation efficace et sa gestion continue.

Un bilan de la réalisation du projet est dressé, mettant en évidence les défis relevés, les solutions apportées et les acquis techniques et métier. Des perspectives d'avenir sont également envisagées pour l'évolution du système.

**Mots-clés :** université, éducation, digitalisation

## Abstract

---

The report presents the development and implementation of the “Compus Connect” system, a platform intended to facilitate communication and management of activities within the university campus.

The project context, objectives and issues are outlined, highlighting the need for a centralized solution to meet the diverse needs of students, faculty and administrative staff.

**System Design:** A detailed needs analysis is carried out, leading to the identification of essential functionalities and the design of a modular architecture. UML diagrams are used to visualize the structure and behavior of the system.

Iterative development of the modules is described, emphasizing collaboration between teams and the integration of functionality specific to each module. Testing, debugging, and component integration are also covered.

Optimization strategies are explored to improve system efficiency and responsiveness, with particular attention paid to critical points.

Technical documentation to facilitate maintenance is emphasized, with the use of UML diagrams to complement textual explanations.

Training plans for end users and the maintenance team are presented, emphasizing the importance of in-depth knowledge of the system to ensure its effective use and continued management.

An assessment of the completion of the project is drawn up, highlighting the challenges faced, the solutions provided and the technical and professional achievements. Future perspectives are also considered for the evolution of the system.

**Keywords :** university, education, digitalization

## ملخص

---

هو عبارة عن منصة تهدف إلى تسهيل الاتصال وإدارة ،"Compus Connect" يعرض التقرير تطوير وتنفيذ نظام الأنشطة داخل الحرم الجامعي.

تم توضيح سياق المشروع وأهدافه وقضايا، مع تسليط الضوء على الحاجة إلى حل مركزي لتلبية الاحتياجات المتنوعة للطلاب وأعضاء هيئة التدريس والموظفين الإداريين.

تصميم النظام: يتم إجراء تحليل تفصيلي للاحتجاجات، مما يؤدي إلى تحديد الوظائف الأساسية وتصميم بنية معيارية. لتصور بنية النظام وسلوكه UML تُستخدم مخططات

تم وصف التطوير التكراري للوحدات، مع التركيز على التعاون بين الفرق وتكامل الوظائف الخاصة بكل وحدة. يتم أيضًا تناول الاختبار وتصحيح الأخطاء وتكامل المكونات.

يتم استكشاف استراتيجيات التحسين لتحسين كفاءة النظام واستجابته، مع إلقاء اهتمام خاص لل نقاط الحرجة لاستكمال التوضيحات النصية UML يتم التركيز على الوثائق الفنية لتسهيل الصيانة، مع استخدام مخططات يتم عرض خطط التدريب للمستخدمين النهائين وفريق الصيانة، مع التركيز على أهمية المعرفة المتعمقة بالنظام لضمان استخدامه الفعال وإدارته المستمرة.

يتم إعداد تقييم لإنجاز المشروع، مع تسليط الضوء على التحديات التي واجهها والحلول المقدمة والإنجازات الفنية والمهنية. كما يتم النظر في وجهات النظر المستقبلية لتطور النظام

**الكلمات المفاتيح :** الجامعة، التعليم، الرقمنة