République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université M'hamed Bougara - Boumerdès



Faculté des Sciences
Département d'Informatique

Domaine : Mathématiques Informatique
Filière : Informatique
Spécialité : Systèmes informatiques

Mémoire de fin d'études en vue de l'obtention du
Diplôme de Licence Académique

Thème

Développement d'une plateforme web pour la gestion centralisée des inscriptions universitaires.

Présenté par :
BAAZIZ DJAZIA
REZZOUK ABD EL MALEK
MEFTI SARAH
BAKHOUCHE SALEM

Soutenu le 26/6/4 Devant le jury composé de :
M.Rezki Mohamed Amine : Examinateur
M.YASSINE BAROUDI : Encadreur

Remerciements

Avant tout, nous remercions **Allah** de nous avoir accordé la santé,
la volonté et le courage nécessaires pour mener à bien ce travail.

Nous tenons à exprimer notre profonde gratitude à notre encadreur
M. Yassine Baroudi, pour son soutien continu, ses précieux conseils
et sa disponibilité tout au long de la réalisation de ce projet.
Son accompagnement bienveillant et son expertise nous ont été
d'une aide indispensable.

Nous remercions également les membres du jury d'avoir accepté
d'évaluer ce travail et de lui accorder de leur temps précieux.

Nos sincères remerciements vont aussi à l'ensemble des enseignants
du **Département d'Informatique** de l'Université M'hamed Bougara
de Boumerdès, qui ont contribué à notre formation tout au long
de notre cursus universitaire.

Enfin, nous dédions ce travail à nos **familles**,
qui ont été notre soutien le plus précieux.
Leur amour, leur patience et leurs encouragements constants
nous ont donné la force d'aller jusqu'au bout.
Qu'ils trouvent ici l'expression de notre reconnaissance
et de notre affection les plus sincères.

Liste des figures

Figure 1 : Diagramme de cas d'utilisation (Étudiant) 9
Figure 2 : Diagramme de cas d'utilisation (Département) 10
Figure 3 : Diagramme de cas d'utilisation (Administrateur) 10
Figure 4: Diagramme de classe 11
Figure 5 : Diagramme de séquence connexion étudiant 12
Figure 6: Diagramme de séquence inscription étudiant 12
Figure 7: Diagramme de séquence connexion département 13
Figure 8: Diagramme de séquence inscription département 13
Figure 9: Diagramme de séquence connexion admin 14
Figure 10: Diagramme de séquence inscription admin 14
Figure 11: Diagramme de séquence mot de passe oublié 15
Figure 12: Diagramme de séquence logique de garde de route 15
Figure 13: Diagramme de séquence formulaire d'inscription étudiant 16
Figure 14: Diagramme de séquence demande de bourse 16
Figure 15: Diagramme de séquence demande d'hébergement 17
Figure 16: Diagramme de séquence demande de transport 17
Figure 17: Diagramme de séquence examen des dossiers 18
Figure 18: Diagramme de séquence supervision admin et gestion des comptes . 18
Figure 19: Diagramme de séquence page d'accueil et role modal 19
Figure 20: Diagramme de séquence cycle de vie AuthContext 19
Figure 21: Diagramme de séquence Rehydration de session au rechargement ... 20
Figure 22: Logo VS Code 22
Figure 23: Logo CSS 22
Figure 24: Logo React.js 23
Figure 25: Logo Axios 23
Figure 26: Logo npm 23
Figure 27: Logo PHP 24
Figure 28: Logo Laravel 24
Figure 29: Logo SQLite 24
Figure 30: Logo GitHub 25
Figure 31: Image de fond 25
Figure 32: Page d'accueil de la plateforme Qaboul.dz 28
Figure 33: Interface du formulaire d'inscription universitaire 29
Figure 34: Interface Housing & Transport pour étudiant 30
Figure 35: Page Mon Profil de l'étudiant 30
Figure 36: Interface de gestion des comptes utilisateurs 31
Figure 37: Interface des paramètres système (Administrateur) 32
Figure 38: Interface de gestion des invitations 33
Figure 39: Interface de gestion des dossiers par département 34

Table des matières

Liste des figures 3
Liste des tableaux 5
Table des matières 4
Introduction Générale 8
Problématique 8

Chapitre 1 : MODÉLISATION ET CONCEPTION
1.1 Introduction 8
1.2 Contexte du projet 8
1.2.1 Introduction 8
1.2.2 Problématique 8
1.2.3 Identification des acteurs 9
1.2.4 Diagrammes de cas d'utilisation 9
1.2.5 Diagramme de classe 12
1.2.6 Diagramme de séquence 12
1.3 Conclusion 21

Chapitre 2 : Les outils utilisés
1 Introduction 23
2 Environnement de développement 23
2.1 Visual Studio Code (VS Code) 23
3 Technologies Front-End 23
3.1 CSS (Cascading Style Sheets) 23
3.2 React.js 24
3.3 Axios 24
3.4 npm (Node Package Manager) 24
4 Technologies Back-End 25
4.1 PHP 25
4.2 Laravel 25
5 Base de données 25
5.1 SQLite 25
6 Gestion des versions 26
6.1 GitHub 26
7 Ressources graphiques 26
7.1 Image de fond (Google Images) 26
8 Tableau récapitulatif des outils utilisés 27
Conclusion 27

Chapitre 3 : Réalisation
9 Introduction 29
10 Interface Publique 29
10.1 Page d'accueil (Landing Page) 29
11 Interface Étudiant 30
11.1 Formulaire d'Inscription (Enrollment) 30
11.2 Page Housing & Transport 30
11.3 Page Mon Profil (My Profile) 31

12 Interface Administrateur 32
12.1 Liste des Comptes Utilisateurs (User Accounts) 32
12.2 Page Paramètres Système (Settings) 32
12.3 Gestion des Invitations (Invite Management) 33
13 Interface Département 34
13.1 Gestion des Dossiers par Département 34
14 Conclusion 35

Conclusion Générale
Références 36

Introduction générale :

La gestion des inscriptions à l'université est une étape clé pour les étudiants. Mais les méthodes classiques ont des inconvénients, comme les longues files d'attente, les déplacements entre les services administratifs et une perte de temps importante. De plus, ces processus peuvent comporter des erreurs qui impactent la fiabilité des informations.

Notre projet est de créer une plateforme web centralisée pour gérer les inscriptions universitaires. Cela permettra de rendre le processus d'inscription plus simple et plus rapide, tout en réduisant les erreurs.

Cette plateforme donnera aux départements et à l'administration des outils pour gérer et suivre les inscriptions en temps réel. Cela assurera une gestion plus fluide et plus transparente.

Notre but est de faciliter la vie des étudiants et des équipes administratives. Nous voulons offrir une solution efficace pour la gestion des inscriptions universitaires.

Problématique :

L'inscription à l'université est un processus compliqué pour tout le monde. Les étudiants et l'administration doivent gérer beaucoup de documents et de dossiers en raison du grand nombre d'étudiants.

Certains étudiants doivent voyager très loin de leur région pour s'inscrire et attendre longtemps avant que leur dossier soit traité. Cela prend beaucoup de temps et il y a un risque que les documents soient perdus.

Face à ces contraintes, nous proposons la conception et le développement d'une plateforme web visant à faciliter le processus d'inscription universitaire, tant pour les étudiants que pour l'administration.

Chapitre 1:

Modélisation et conception

1.1 Introduction

Dans ce chapitre, nous présentons la modélisation et la conception de notre système de gestion des inscriptions universitaires. Cette étape est essentielle, car elle permet de définir les besoins du système et d'organiser ses différentes fonctionnalités avant la phase de réalisation.

Nous introduisons ensuite la modélisation UML à travers les diagrammes de cas d'utilisation, les diagrammes de séquence ainsi que le diagramme de classes afin de formaliser les fonctionnalités du système et représenter son architecture générale. Cette modélisation permet d'avoir une vision claire du fonctionnement de l'application avant sa phase de développement.

1.2 Contexte de projet

1.2.1 Identification des acteurs

Dans notre système, plusieurs acteurs interviennent dans le processus d'inscription universitaire :

- Étudiant : Il crée un compte, remplit les formulaires d'inscription (d'inscription, de demande de bourse, de demande hébergement/transport)...., dépose les documents et suit l'état de son dossier.

- Responsable de département : Il consulte les dossiers des étudiants, vérifie les documents, valide ou rejette les demandes.

- Administrateur : Il gère les utilisateurs, les départements, les filières et supervise l'ensemble du système.

1.2.2 Diagramme de cas d'utilisation (Use Case):

Le diagramme de cas d'utilisation représente les interactions entre les différents acteurs et le système.

Il montre les principales fonctionnalités offertes, telles que la création de compte, le dépôt de dossier, la validation par les départements et la gestion administrative.

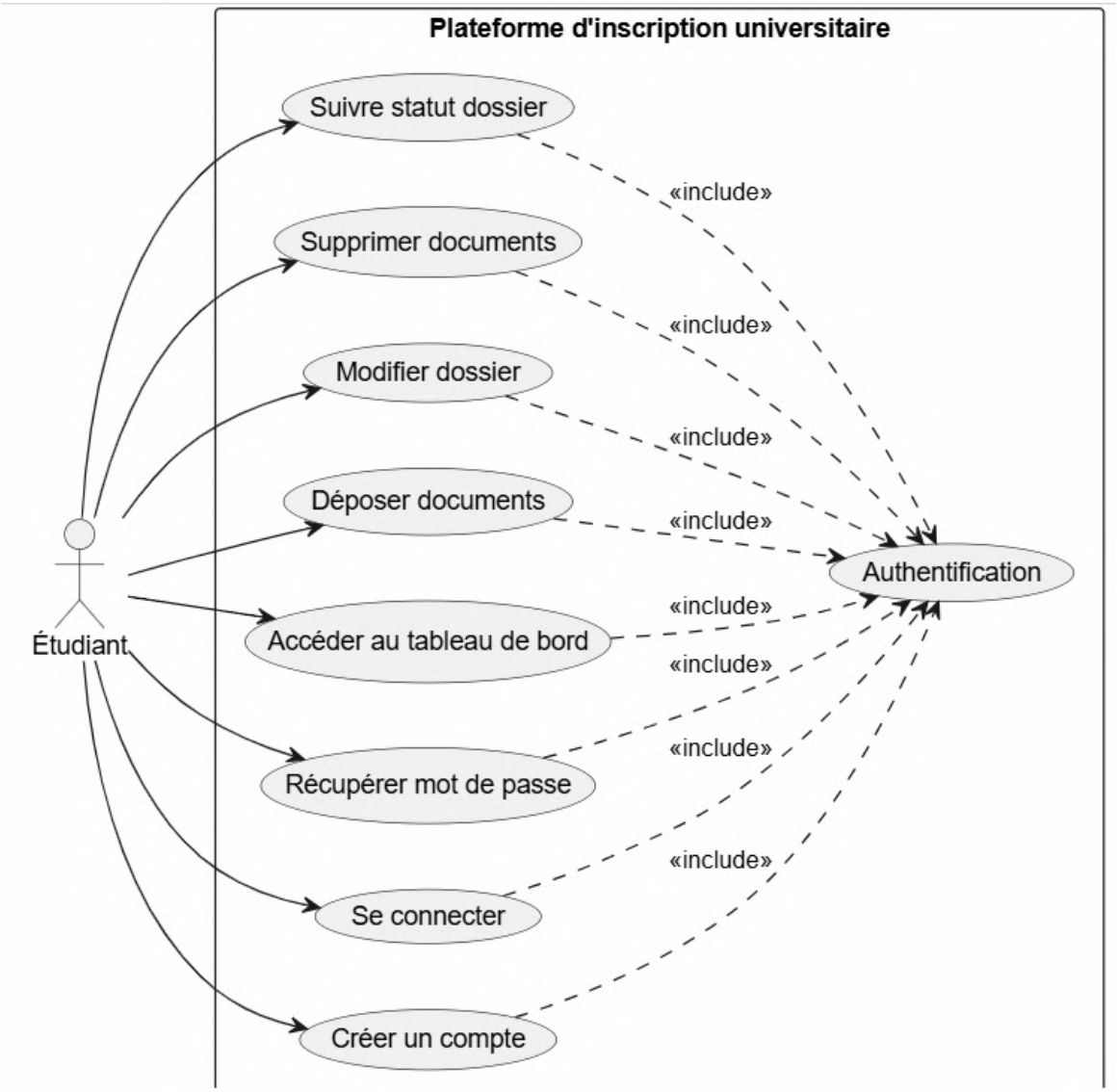


Figure 1: Diagramme de cas d'utilisation (Étudiant)

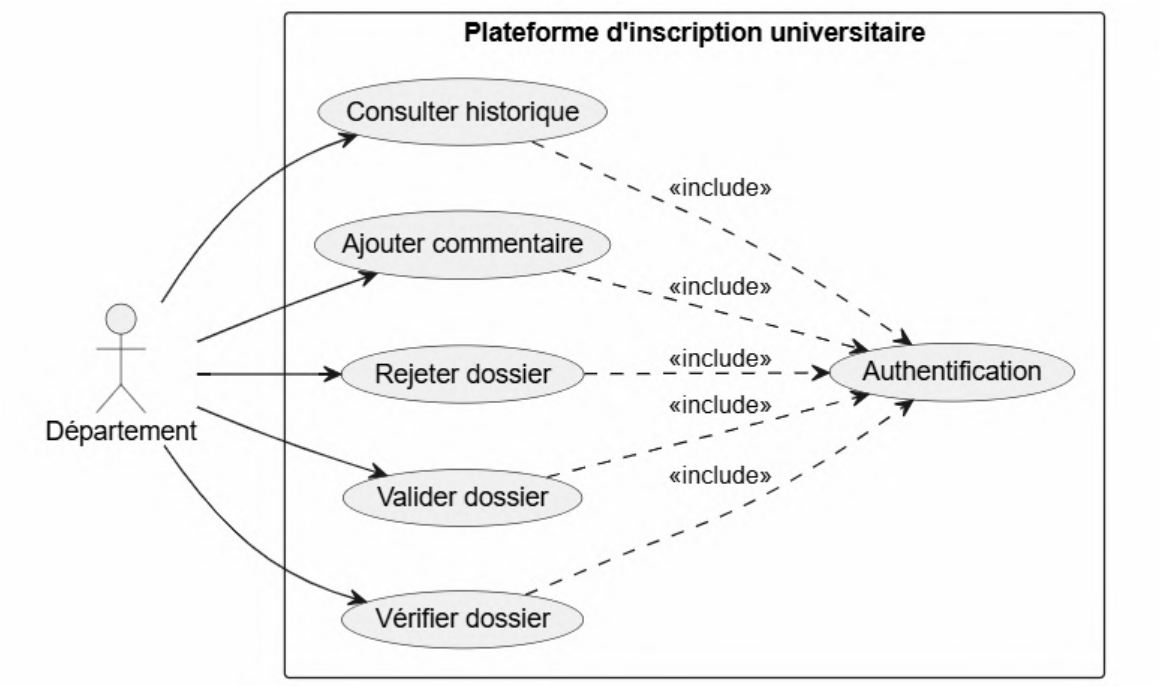


Figure 2: Diagramme de cas d'utilisation (Département)

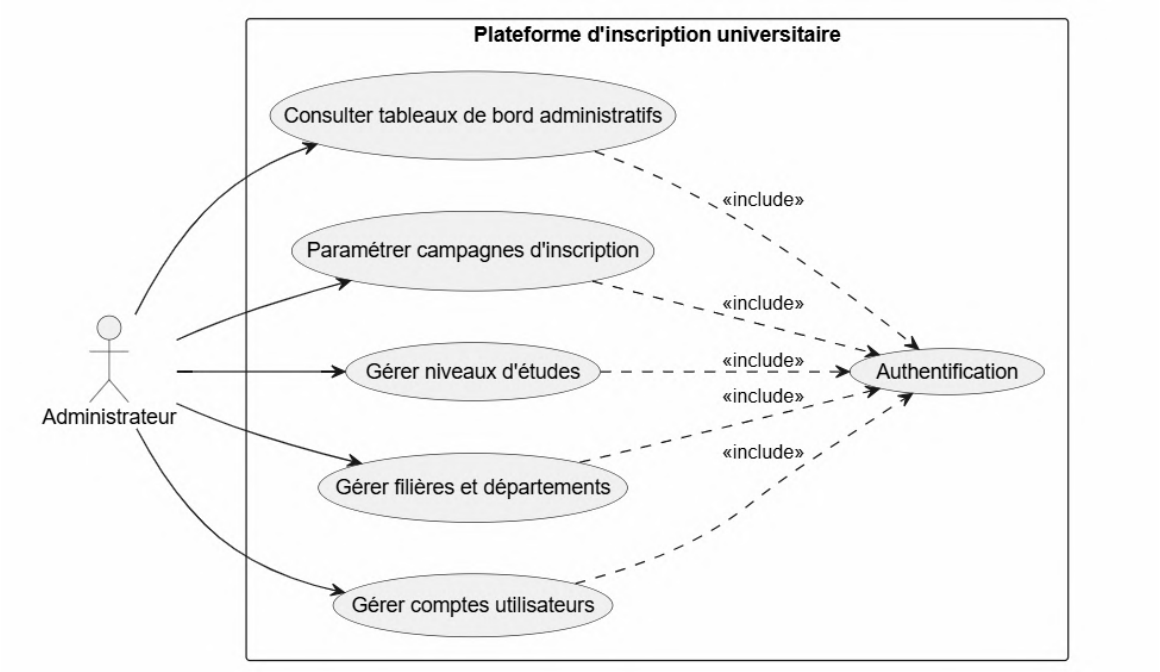


Figure 3: Diagramme de cas d'utilisation (Administrateur):

1.2.3 Diagramme de classe :

Le diagramme de classes représente la structure statique du système. Il permet d'identifier les différentes classes de l'application, leurs attributs, leurs méthodes ainsi que les relations existantes entre elles.

Il offre une vue globale de l'architecture du système et facilite sa conception et son développement.

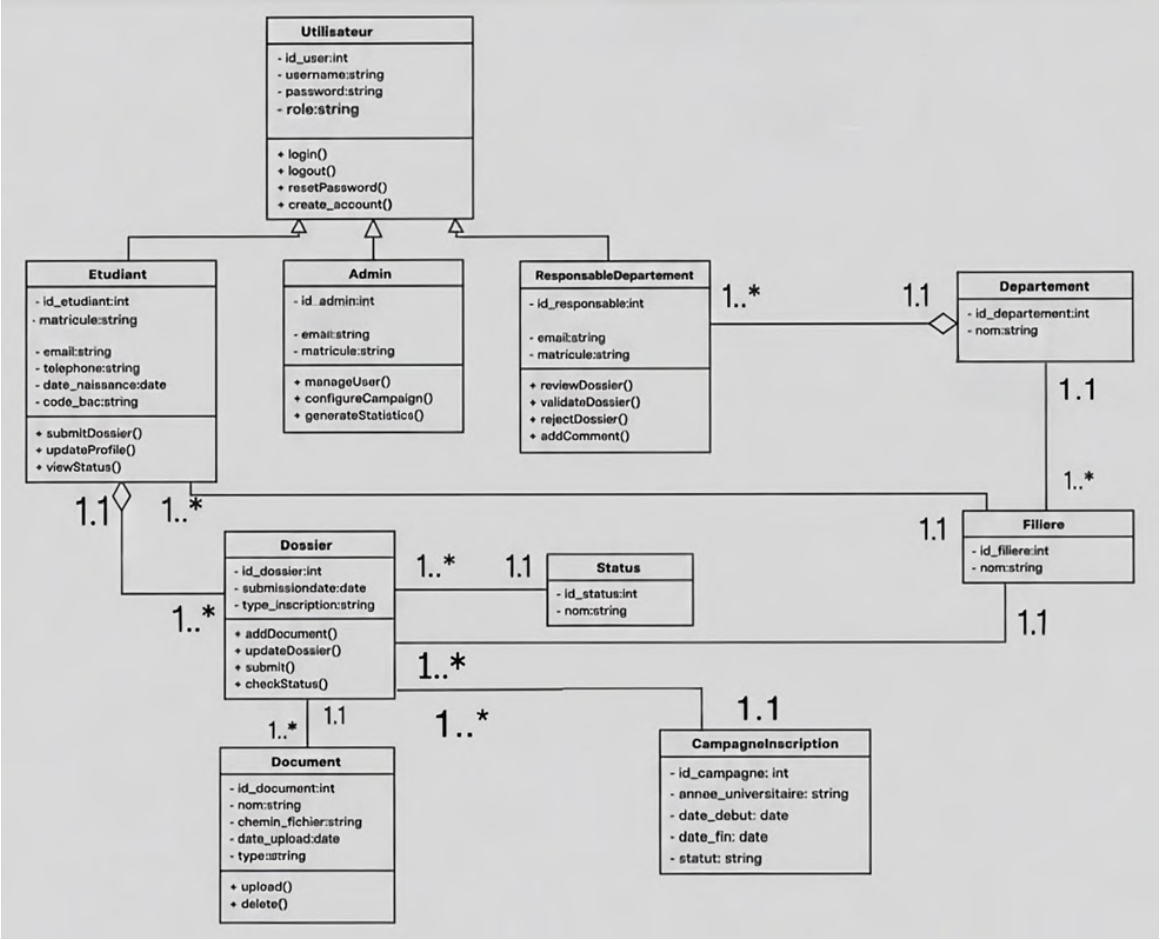


Figure 4: Diagramme de classe

1.2.4 Diagramme de séquence :

Le diagramme de séquence représente les interactions entre les différents acteurs et le système selon un ordre chronologique.

Il permet de décrire l'enchaînement des messages échangés ainsi que le déroulement des différentes opérations réalisées dans l'application.

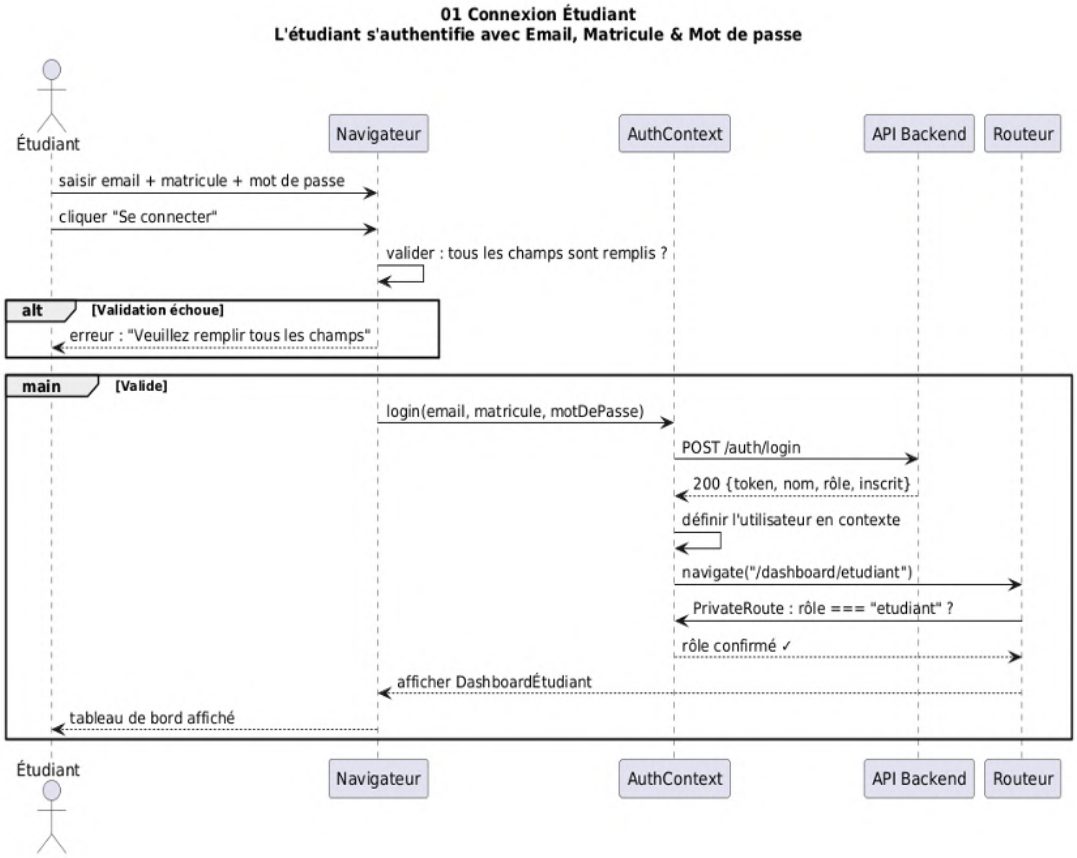


Figure 5: Diagramme de séquence connexion étudiant

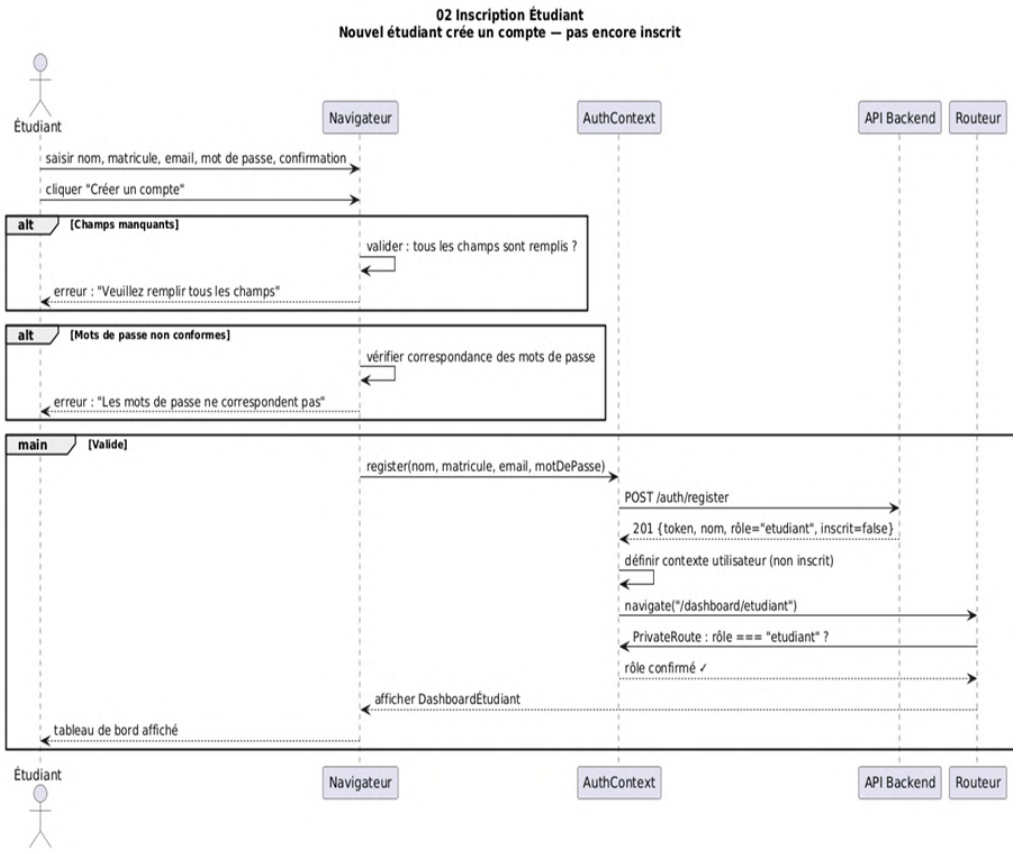


Figure 6: Diagramme de séquence inscription étudiant

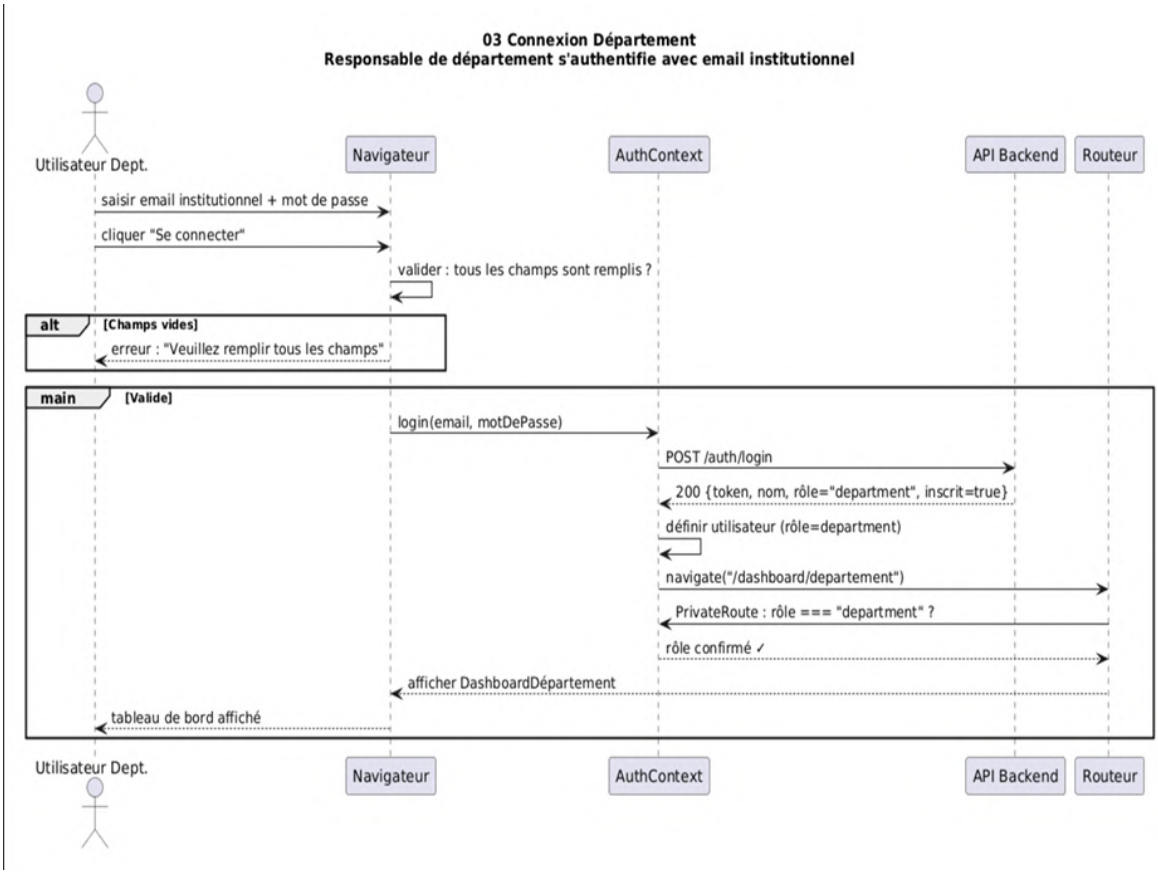


Figure 7: Diagramme de séquence connexion département

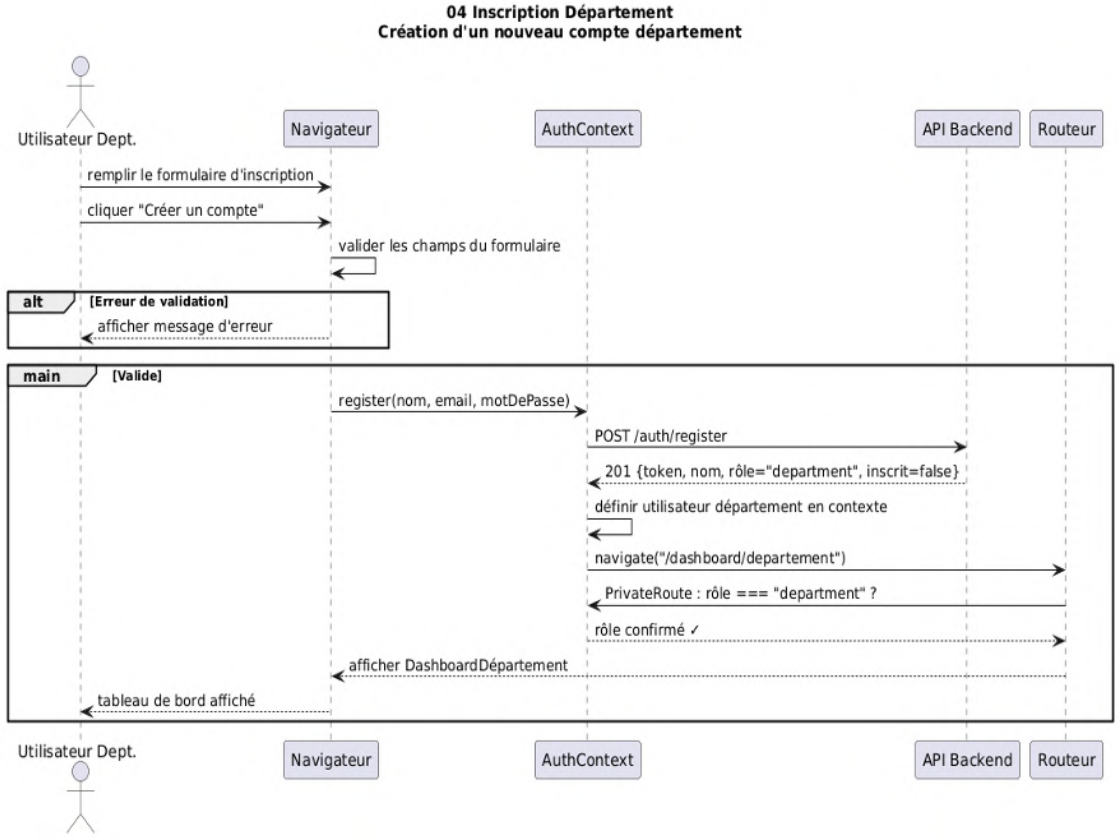


Figure 8: Diagramme de séquence inscription département

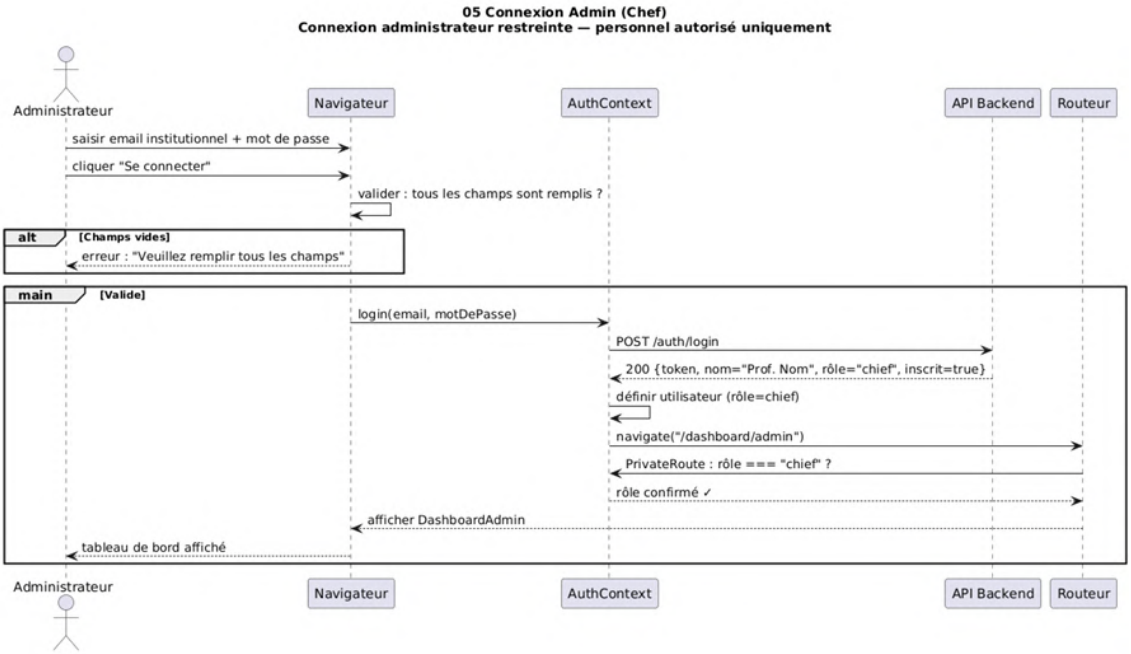


Figure 9: Diagramme de séquence connexion admin

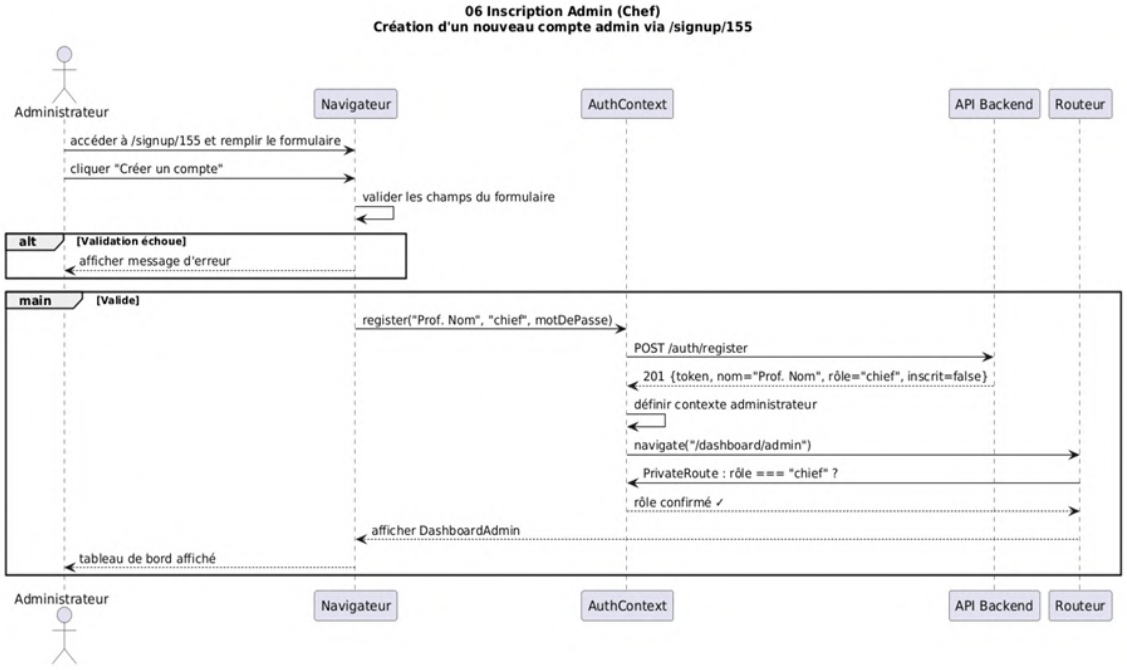


Figure 10: Diagramme de séquence inscription admin

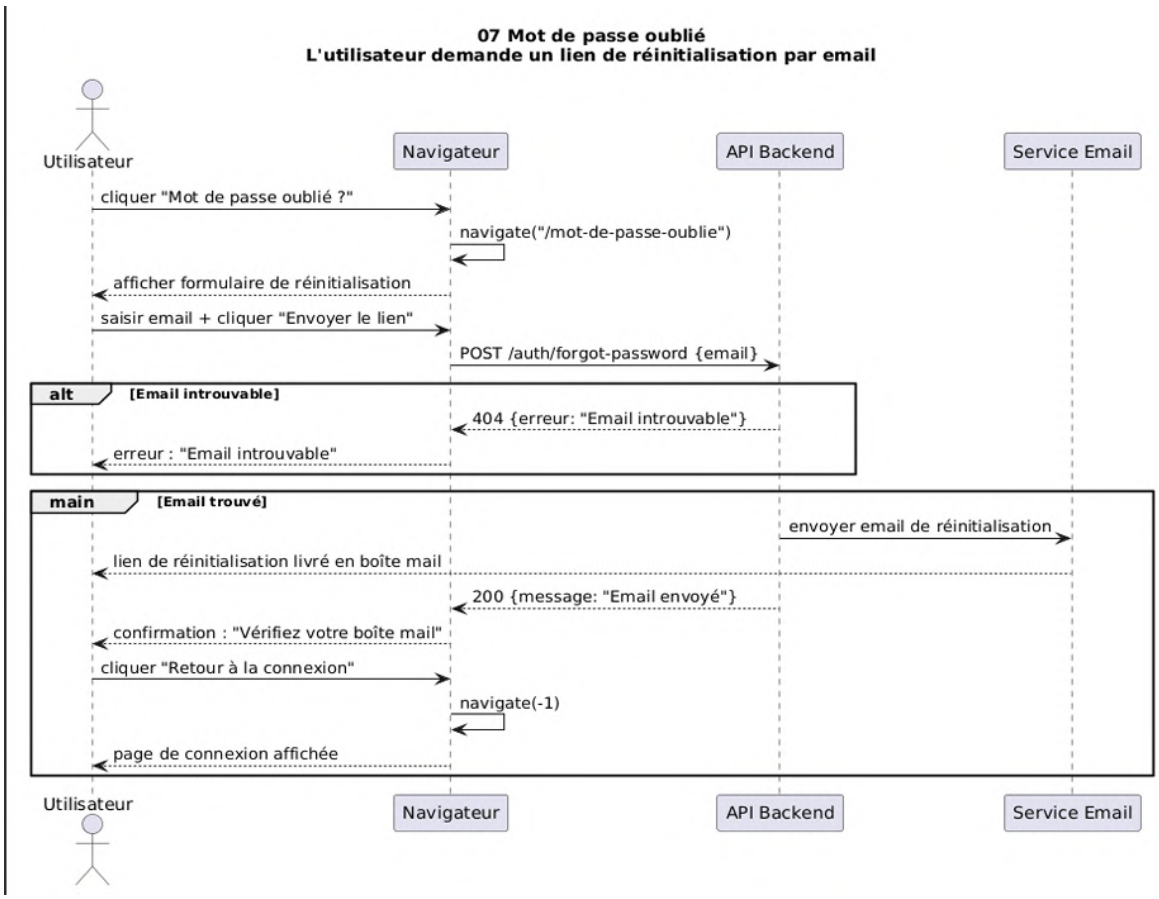


Figure 11: Diagramme de séquence mot de passe oublié

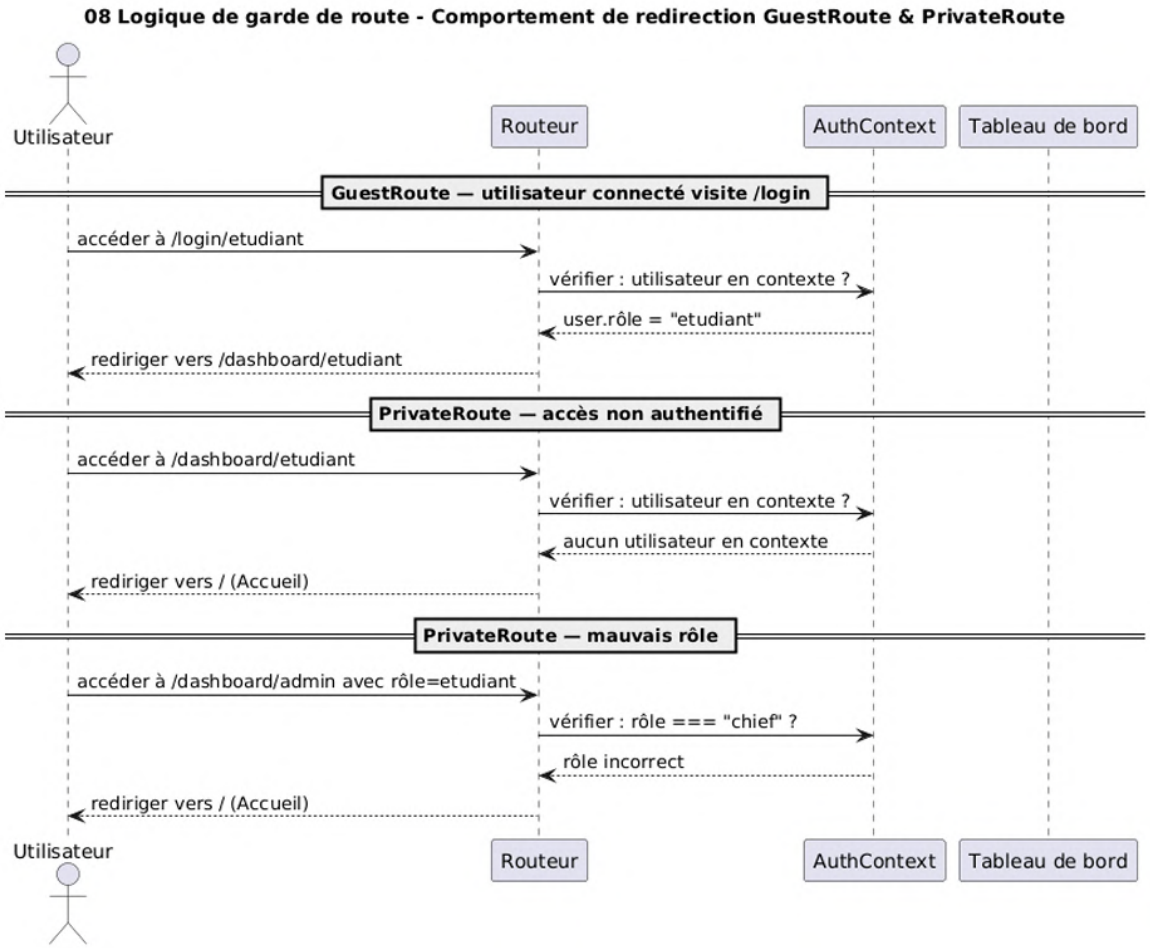


Figure 12: Diagramme de séquence logique de garde de route

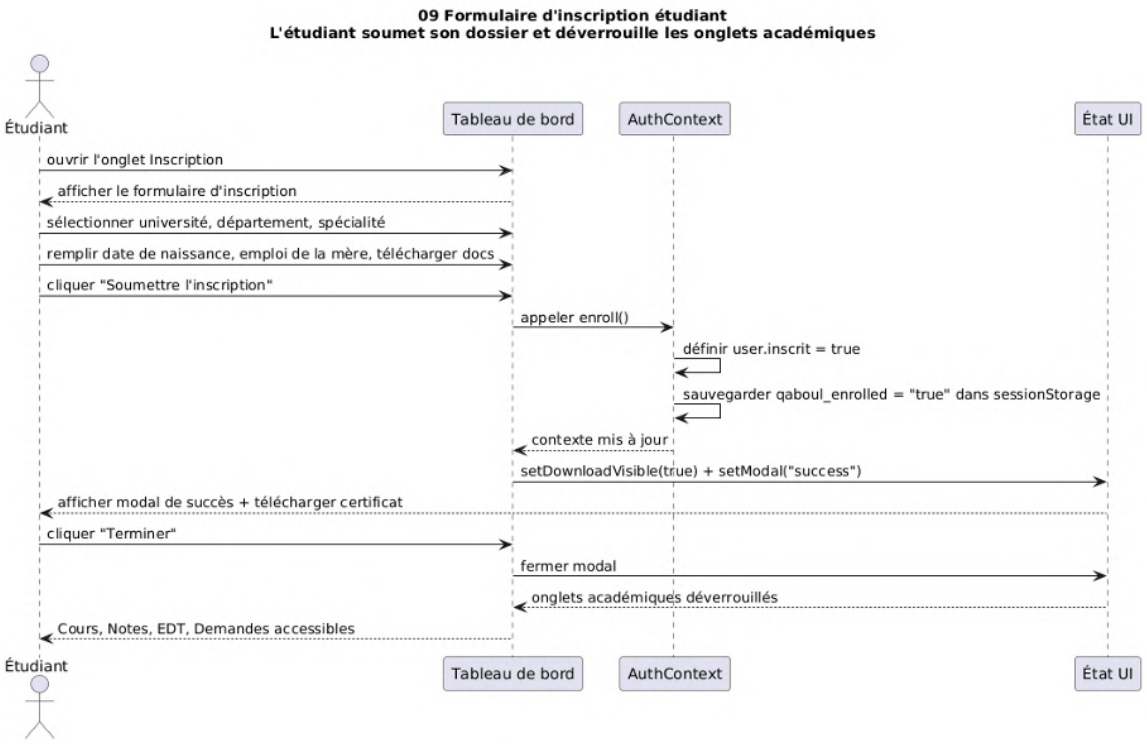


Figure 13: Diagramme de séquence formulaire d'inscription étudiant

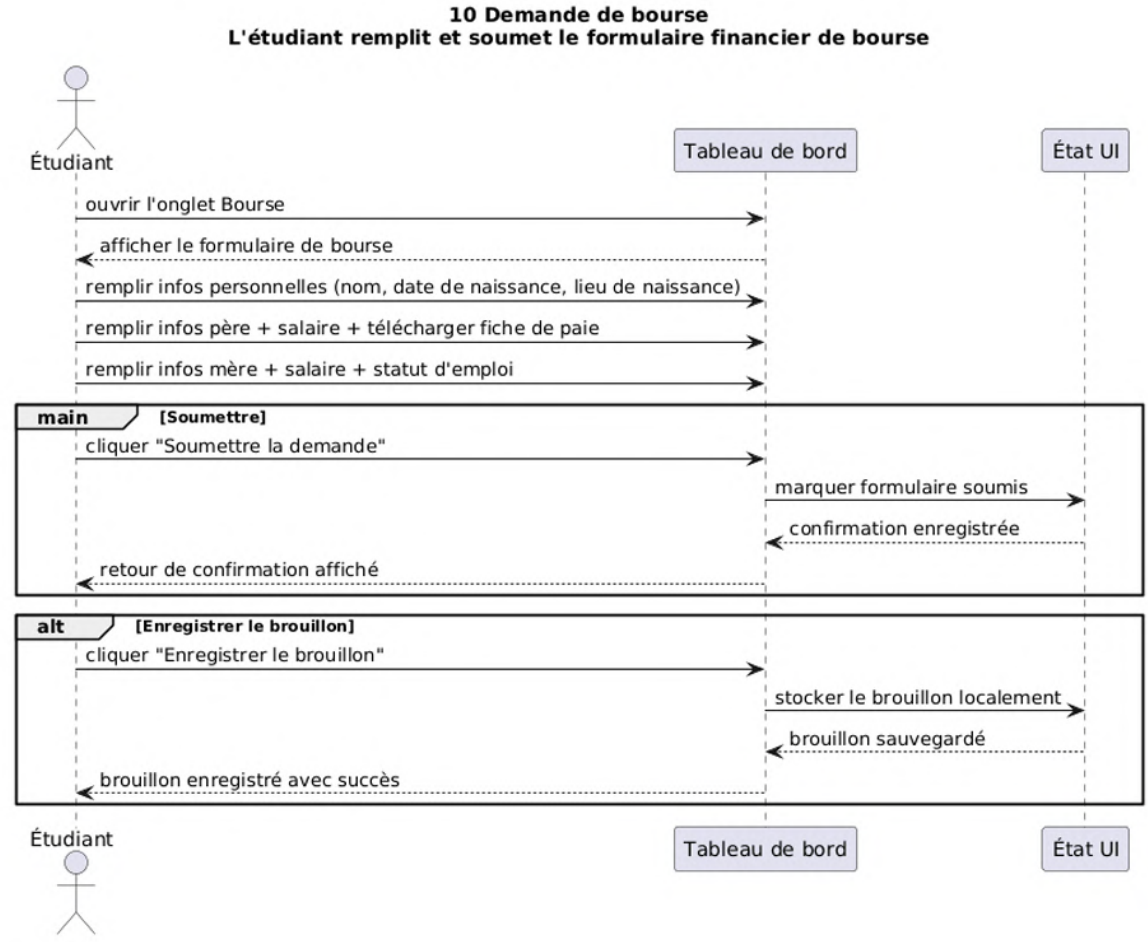


Figure 14: Diagramme de séquence demande de bourse

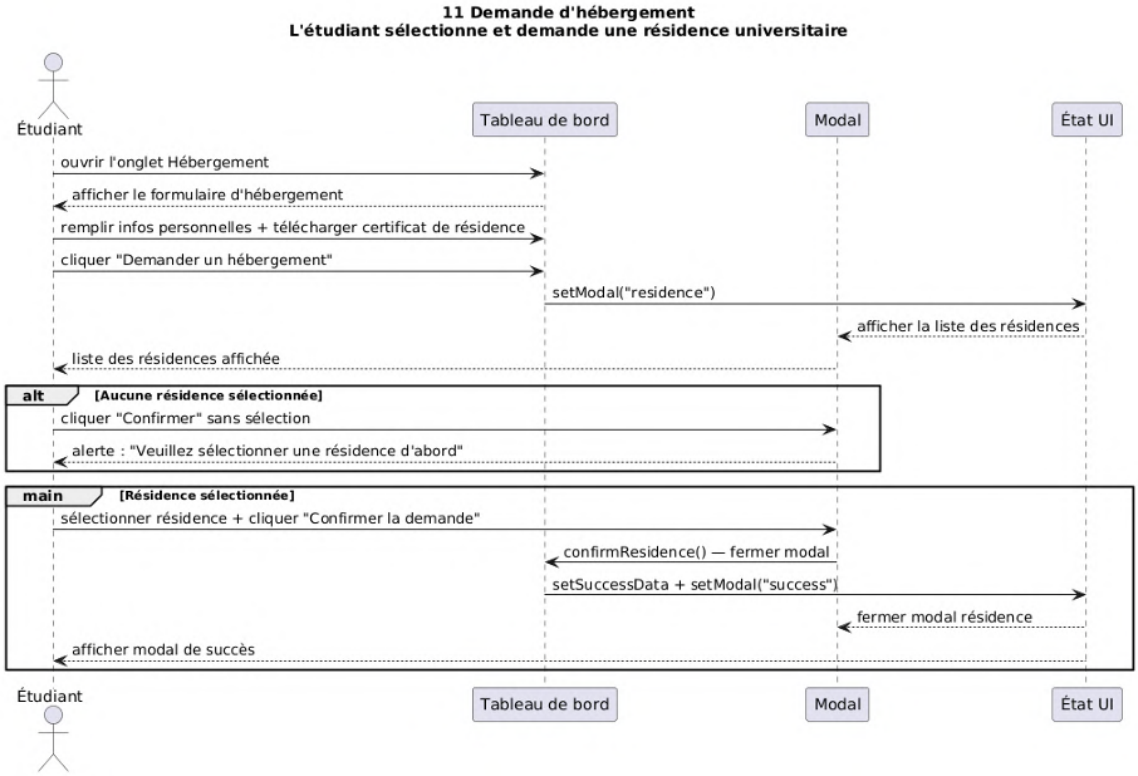


Figure 15: Diagramme de séquence demande d'hébergement

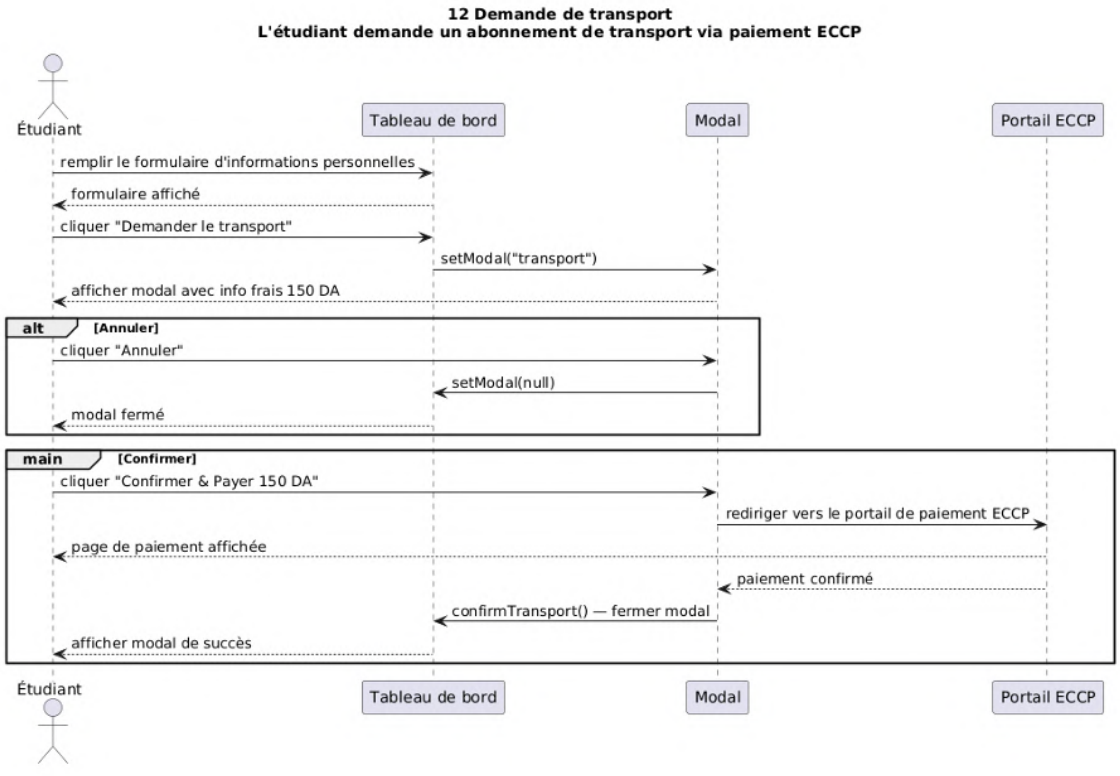


Figure 16: Diagramme de séquence demande de transport

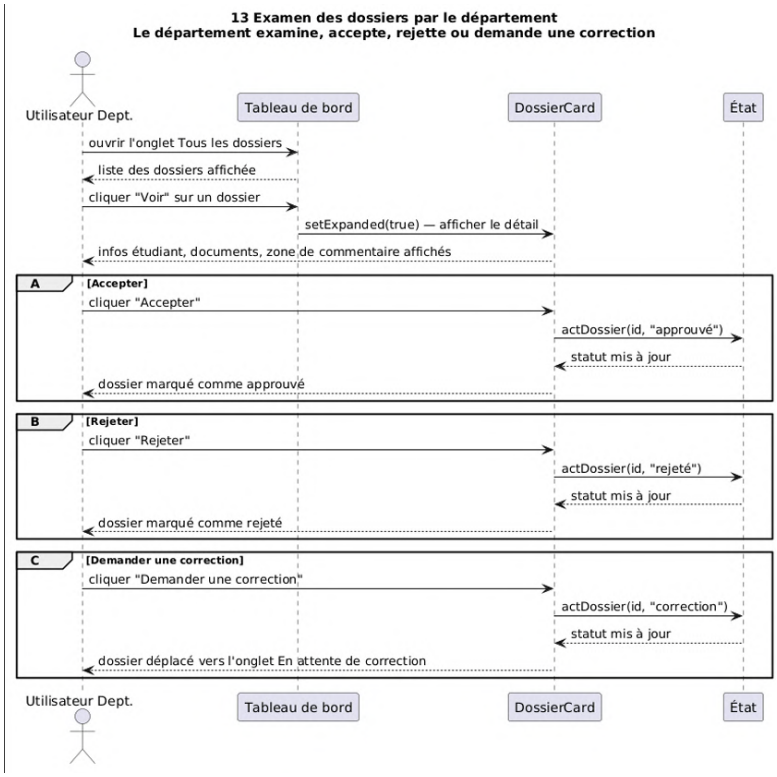


Figure 17: Diagramme de séquence examen des dossiers par le département

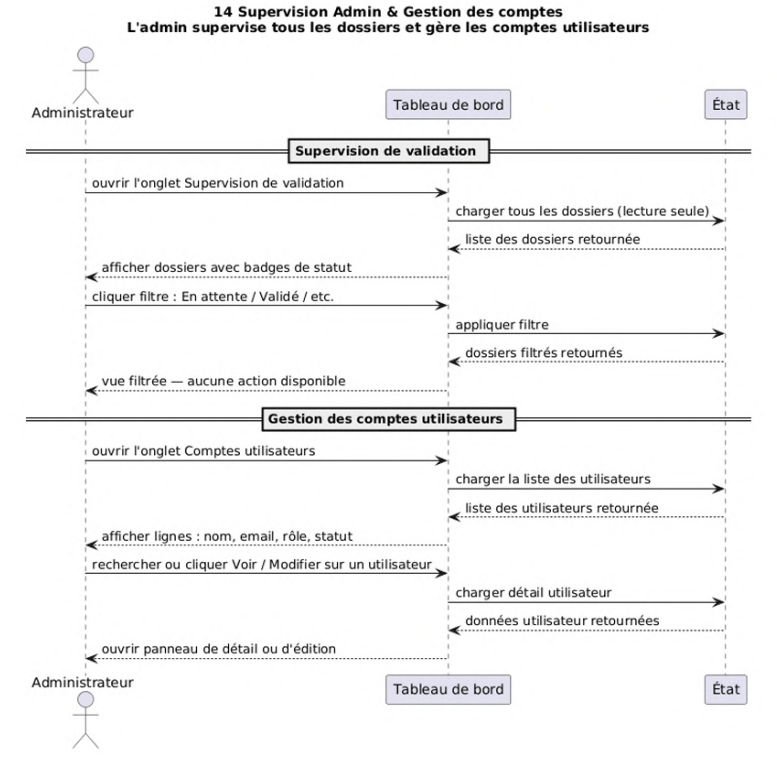


Figure 18: Diagramme de séquence supervision admin et gestion des comptes

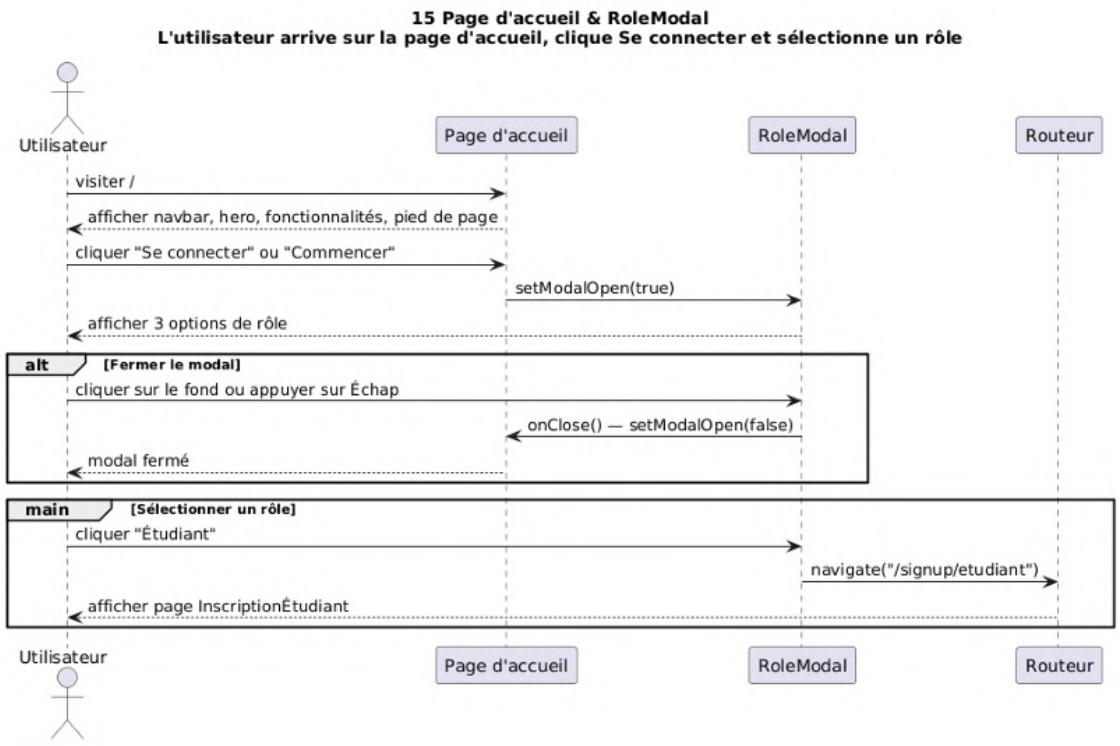


Figure 19: Diagramme de séquence page d'accueil et role modal

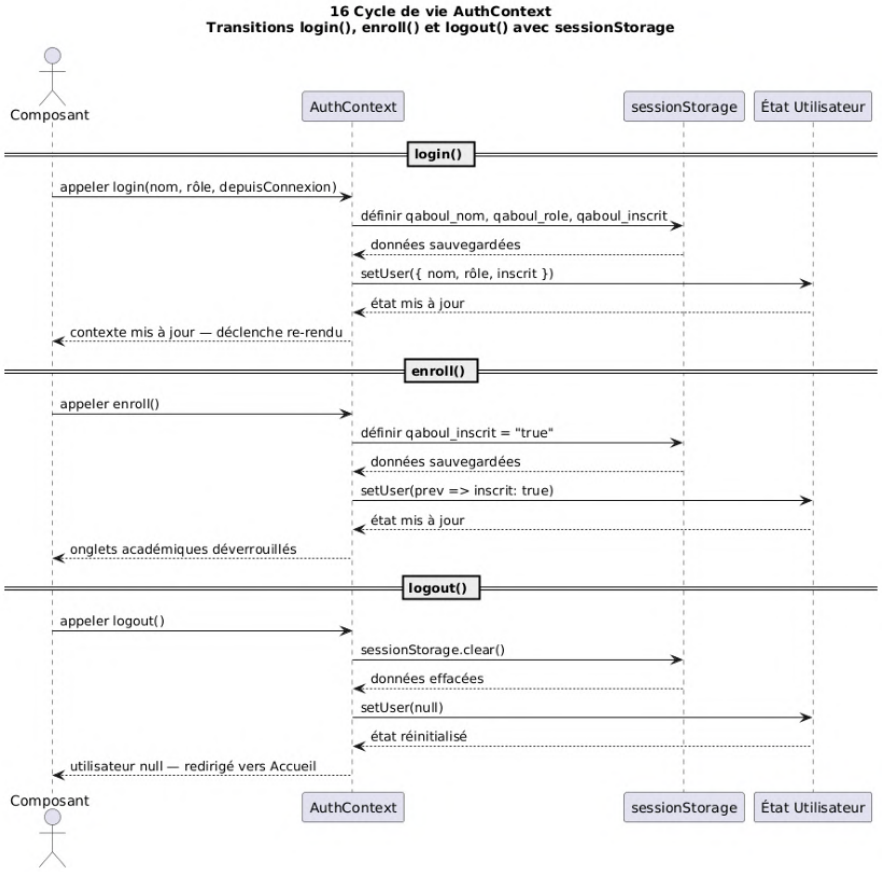


Figure 20: Diagramme de séquence cycle de vie AuthContext

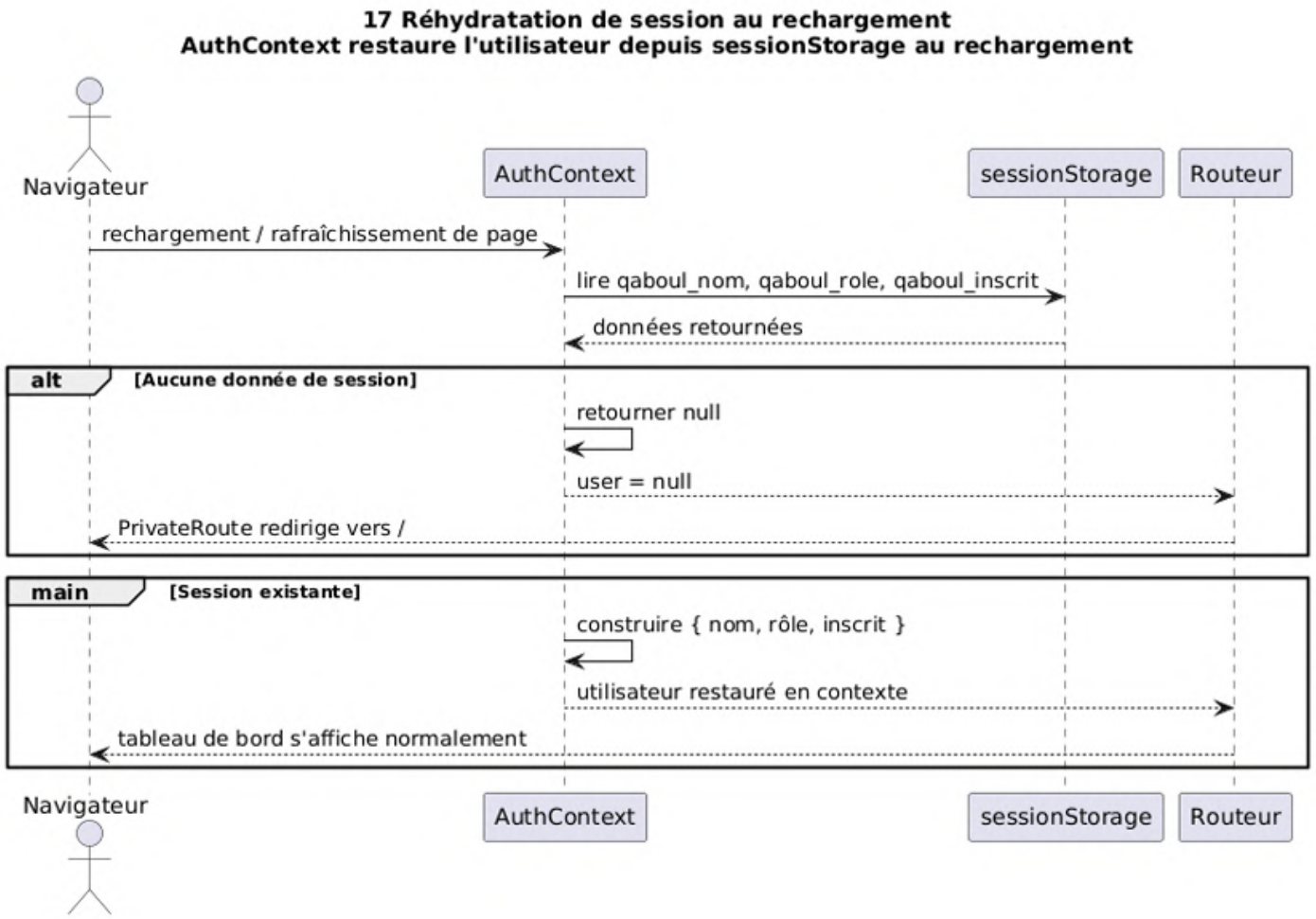


Figure 21: Diagramme de séquence Rehydration de session au rechargement

Conclusion

Ce chapitre a présenté la modélisation et la conception de la plateforme **Qaboul.dz**. À travers l'identification des acteurs et l'élaboration des diagrammes UML (cas d'utilisation, classes et séquences), nous avons pu définir clairement les besoins du système et organiser ses différentes fonctionnalités. Cette phase de conception constitue une base solide et essentielle pour la réalisation du projet, en garantissant une architecture cohérente et bien structurée. Dans le chapitre suivant, nous présenterons les outils et technologies utilisés pour le développement de la plateforme.

Chapitre 2:

Les outils utilisés

1 Introduction

Ce chapitre présente l'ensemble des outils et technologies utilisés pour le développement de la plateforme web **Qaboul.dz**. Ces outils couvrent plusieurs aspects du développement : l'environnement de travail, le développement *Front-end*, le *Back-end*, la base de données, la gestion des versions ainsi que les ressources graphiques. Chaque technologie a été choisie en fonction de sa pertinence, de sa popularité et de son adéquation avec les besoins du projet.

2 Environnement de développement

2.1 Visual Studio Code (VS Code)

Visual Studio Code est l'environnement de développement intégré (*IDE*) utilisé pour la totalité du développement du projet. Développé par Microsoft, il est reconnu pour sa légèreté, ses nombreuses extensions ainsi que son support étendu de plusieurs langages de programmation[1].

Définition : Visual Studio Code (*VS Code*) est un environnement de développement intégré léger et extensible, développé par Microsoft. Il offre plusieurs fonctionnalités telles que la coloration syntaxique, la complétion intelligente du code (*IntelliSense*), le débogage intégré, la gestion de Git ainsi qu'un large ensemble d'extensions. Il est disponible sur différents systèmes d'exploitation tels que Windows, macOS et Linux[1].



Figure 22: Logo VS Code

3 Technologies Front-End

3.1 CSS (Cascading Style Sheets)

Le CSS (*Cascading Style Sheets*) est le langage de mise en forme utilisé pour styliser l'interface visuelle de **Qaboul.dz**. Il a permis de définir les couleurs, les typographies, les mises en page ainsi que les animations de l'application[2].

Définition : Le CSS (*Cascading Style Sheets*) est un langage de feuilles de style utilisé pour décrire la présentation d'un document HTML. Il permet de contrôler l'apparence visuelle des éléments web tels que les couleurs, les polices, les espacements, les mises en page adaptatives (*responsive*) ainsi que les animations[2].



Figure 23: Logo CSS

3.2 React.js

React.js est le framework JavaScript utilisé pour le développement de l'interface utilisateur de **Qaboul.dz**. Il permet de construire des interfaces dynamiques, réactives et organisées en composants réutilisables[3].

Définition : React.js est une bibliothèque JavaScript *open-source* développée par Meta (*Facebook*), utilisée pour la création d'interfaces utilisateur. Elle repose sur une architecture basée sur les composants et utilise un *Virtual DOM* afin d'optimiser les performances. Elle est largement utilisée dans le développement des applications web de type *SPA* (*Single Page Application*)[3].

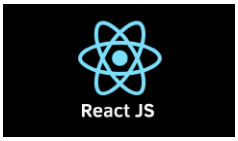


Figure 24: Logo React.js

3.3 Axios

Axios est la bibliothèque JavaScript utilisée pour effectuer les appels API entre le *Front-end React* et le *Back-end Laravel*. Elle facilite l'envoi des requêtes HTTP (*GET, POST, PUT, DELETE*) ainsi que la gestion asynchrone des réponses[4].

Définition : Axios est une bibliothèque JavaScript *open-source* basée sur les promesses (*Promises*), permettant d'effectuer des requêtes HTTP depuis le navigateur ou *Node.js*. Elle fournit une syntaxe simple pour consommer des API REST et offre plusieurs fonctionnalités telles que la gestion des en-têtes, des intercepteurs et des erreurs[4].



Figure 25: Logo Axios

3.4 npm (Node Package Manager)

npm est le gestionnaire de paquets utilisé pour installer et gérer les dépendances JavaScript du projet *Front-end*, notamment React, Axios ainsi que d'autres bibliothèques tierces[5].

Définition : npm (*Node Package Manager*) est le gestionnaire de paquets officiel de l'environnement *Node.js*. Il permet d'installer, de mettre à jour et de gérer les bibliothèques ainsi que les dépendances d'un projet JavaScript à travers un fichier de configuration nommé *package.json*[5].

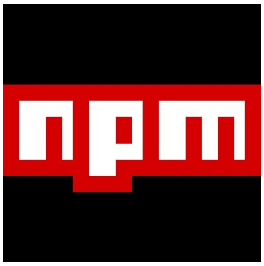


Figure 26: Logo npm

4 Technologies Back-End

4.1 PHP

PHP est le langage de programmation côté serveur utilisé comme base du *Back-end* de **Qaboul.dz**. Il assure le traitement des requêtes HTTP, la logique métier ainsi que la communication avec la base de données[6].



Figure 27: Logo PHP

Définition : PHP (*Hypertext Preprocessor*) est un langage de script *open-source* exécuté côté serveur et destiné au développement web. Il permet de générer du contenu dynamique, de gérer les sessions utilisateur, d'interagir avec les bases de données ainsi que de traiter les formulaires[6].

4.2 Laravel

Laravel est le framework PHP utilisé pour structurer et développer le *Back-end* de la plateforme. Il fournit une architecture MVC robuste, un système de routage, un ORM puissant (*Eloquent*) ainsi que des outils d'authentification intégrés[7].



Figure 28: Logo Laravel

Définition : Laravel est un framework PHP *open-source* reposant sur le modèle de conception MVC (*Modèle-Vue-Contrôleur*). Il propose plusieurs fonctionnalités avancées telles que le routage RESTful, l'ORM *Eloquent*, les migrations de base de données, les *middlewares*, l'authentification ainsi que la gestion des API, facilitant ainsi le développement d'applications web robustes et maintenables[7].

5 Base de données

5.1 SQLite

SQLite est le système de gestion de base de données relationnelle (SGBDR) utilisé pour le stockage persistant des données de la plateforme : comptes utilisateurs, dossiers d'inscription, campagnes, départements et filières[8].



Figure 29: Logo SQLite

Définition : SQLite est un système de gestion de base de données relationnelle *open-source* intégré directement dans l'application, sans nécessiter de serveur distinct. Il stocke l'intégralité de la base de données dans un seul fichier, et est reconnu pour sa légèreté, sa simplicité de configuration et sa compatibilité avec les frameworks web modernes comme Laravel[8].

6 Gestion des versions

6.1 GitHub

GitHub est la plateforme de gestion de versions utilisée tout au long du développement du projet. Elle a permis à l'équipe de collaborer efficacement, de suivre l'historique des modifications et de gérer les différentes versions du code source[9].

Définition : GitHub est une plateforme web de gestion de versions et de collaboration basée sur Git. Elle permet aux développeurs de stocker, versionner et partager leur code source, de gérer les branches de développement, les *pull requests* et les *issues*, facilitant ainsi le travail en équipe sur des projets logiciels[9].



Figure 30: Logo GitHub

7 Ressources graphiques

7.1 Image de fond (Google Images)

L'image de fond utilisée dans l'interface de la plateforme Qaboul.dz a été obtenue via Google Images. Il s'agit d'une photographie illustrant un campus universitaire, utilisée à des fins purement esthétiques pour renforcer l'identité visuelle de la plateforme[10].

Définition : Google Images est un moteur de recherche d'images développé par Google, permettant de trouver des images sur le web à partir de mots-clés. Dans le cadre de ce projet, il a été utilisé pour rechercher une image de fond représentant un environnement universitaire, intégrée dans l'interface principale de la plateforme[10].



Figure 31: Image de fond

8 Tableau récapitulatif des outils utilisés

Outil / Technologie	Catégorie	Rôle dans le projet
Visual Studio Code	IDE	Rédaction du code source (*front-end* & *back-end*)
CSS3	Style / Mise en forme	Mise en forme et design de l'interface utilisateur
React.js	UI Framework	Interface utilisateur dynamique (SPA)
Axios	API Calls	Communication HTTP entre React et Laravel
npm	Package Manager	Gestion des dépendances JavaScript
PHP 8	Langage Back-End	Logique serveur et traitement des requêtes
Laravel	PHP Framework	API REST, authentification, ORM, gestion des rôles
SQLite	Base de données	Stockage persistant des données de la plateforme
GitHub	Version Control	Gestion des versions et collaboration en équipe
Google Images	Ressource graphique	Image de fond de l'interface principale

Table 1: Tableau récapitulatif des outils utilisés

Conclusion

Ce chapitre a présenté l'ensemble des outils et technologies mis en œuvre pour le développement de la plateforme Qaboul.dz. L'utilisation de *React.js* pour le *front-end*, d'*Axios* pour la communication API, de *Laravel* pour le *back-end* et de *SQLite* pour la base de données constitue une *stack* moderne et cohérente, largement adoptée dans le développement web professionnel. La gestion du code source via *GitHub* a facilité la collaboration au sein de l'équipe. Dans le chapitre suivant, nous présenterons la réalisation concrète de la plateforme à travers ses différentes interfaces et fonctionnalités.

Chapitre 3:

Réalisation

9 Introduction

Ce chapitre présente la réalisation concrète de la plateforme *Qaboul.dz* à travers ses différentes interfaces. Les captures d'écran ci-dessous illustrent les fonctionnalités principales de la plateforme, couvrant l'interface publique d'accueil, les interfaces destinées aux étudiants ainsi que le panneau d'administration.

10 Interface Publique

10.1 Page d'accueil (Landing Page)

La page d'accueil de *Qaboul.dz* constitue le point d'entrée principal de la plateforme. Elle présente la plateforme comme un système national d'inscription universitaire en Algérie, avec un slogan clair : *"University Admissions, Simplified"*.

Elle met en avant les chiffres clés suivants :

- 48 universités partenaires,
- 120 000 étudiants par an,
- une progression 3 fois plus rapide.

Un bouton *"Get Started"* est proposé afin d'encourager les nouveaux utilisateurs à s'inscrire gratuitement.

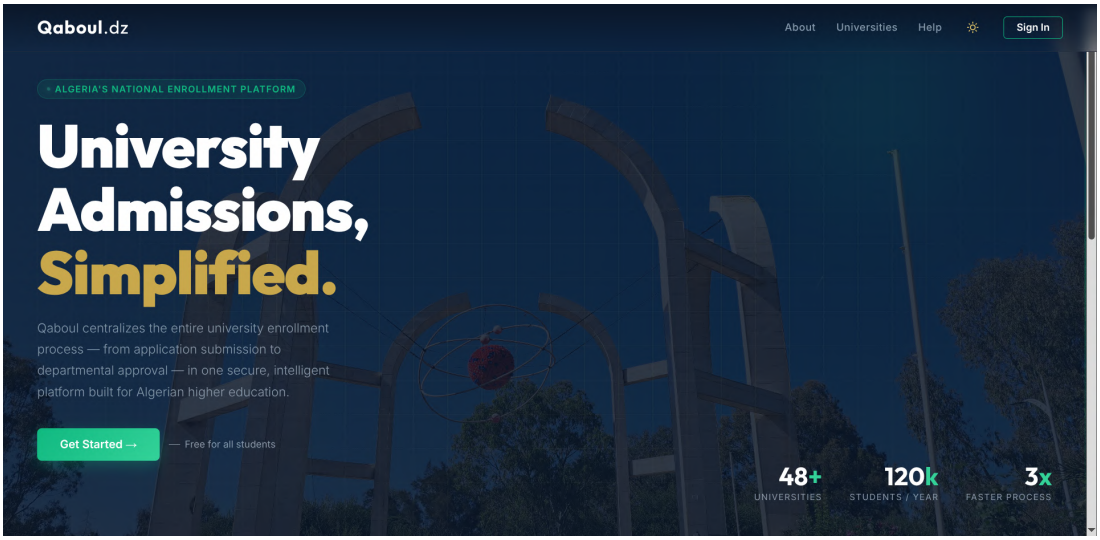


Figure 32: Page d'accueil de la plateforme Qaboul.dz

11 Interface Étudiant

11.1 Formulaire d'Inscription (Enrollment)

Le formulaire d'inscription permet à l'étudiant de soumettre son dossier universitaire. Il doit sélectionner son université (*UMBB — M'hamed Bougara University of Boumerdès*), son département, sa spécialité ainsi que son niveau d'étude. L'étudiant renseigne également son score du baccalauréat (noté sur 20) et sa date de naissance.

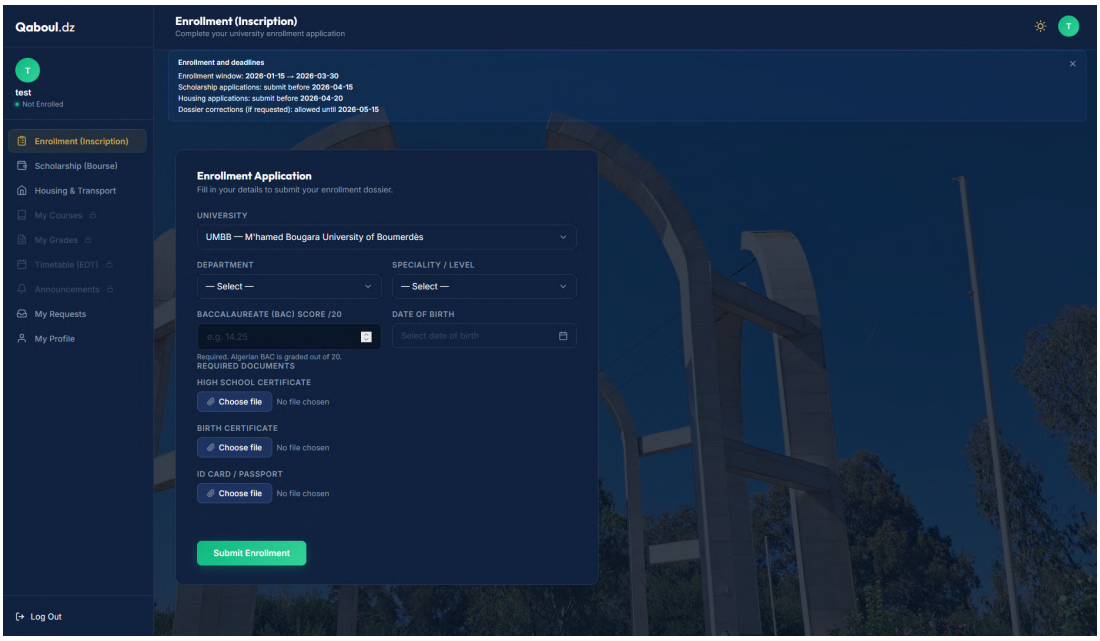


Figure 33: Interface du formulaire d'inscription universitaire

11.2 Page Housing & Transport

Cette interface permet à l'étudiant de soumettre une demande de logement universitaire ou de transport. Le formulaire recueille les informations personnelles de l'étudiant (nom, prénom, lieu de naissance, date de naissance, adresse actuelle) ainsi qu'un certificat de résidence en format PDF ou JPG.

L'étudiant peut choisir entre deux types de demandes :

- *Request Housing* (demande de logement),
- *Request Transport* (demande de transport).

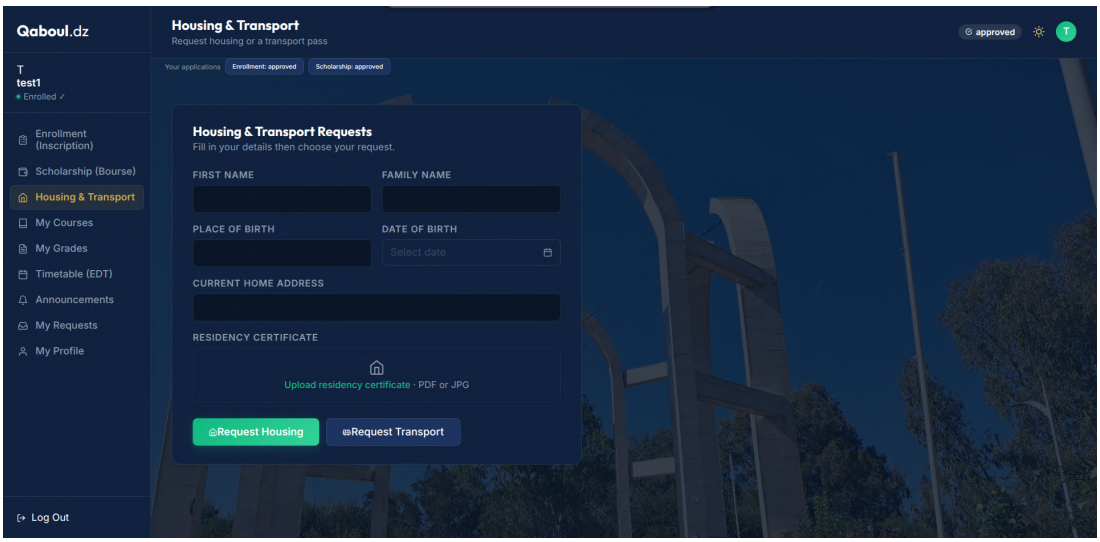


Figure 34: Interface Housing & Transport pour étudiant

11.3 Page Mon Profil (My Profile)

La page de profil permet à l'étudiant de consulter et de modifier ses informations personnelles : nom complet, adresse e-mail, mot de passe, matricule et score du baccalauréat.

Le statut d'inscription est affiché (*Enrolled*), indiquant que le dossier de l'étudiant a été validé avec succès. Des boutons *Edit* et *Change* permettent de mettre à jour les informations.

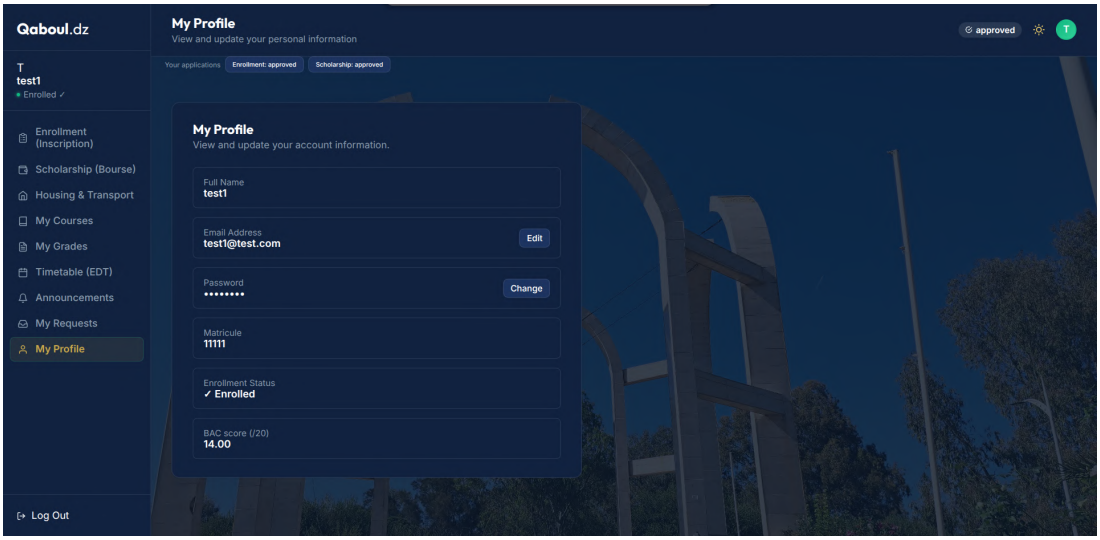


Figure 35: Page Mon Profil de l'étudiant

12 Interface Administrateur

12.1 Liste des Comptes Utilisateurs (User Accounts)

La page *User Accounts* permet à l'administrateur de consulter et gérer l'ensemble des comptes utilisateurs enregistrés sur la plateforme.

Cette interface affiche une liste détaillée des utilisateurs contenant plusieurs informations telles que le nom de l'utilisateur, l'adresse électronique, le rôle attribué (*Administrator, Department ou Student*) ainsi que le statut du compte (*Active*).

Une barre de recherche est également disponible afin de faciliter la recherche et le filtrage des comptes selon différents critères tels que le nom, l'adresse e-mail ou le rôle associé.

L'administrateur dispose aussi d'un bouton *Edit* permettant la modification des informations relatives à chaque compte utilisateur lorsque cela est nécessaire.

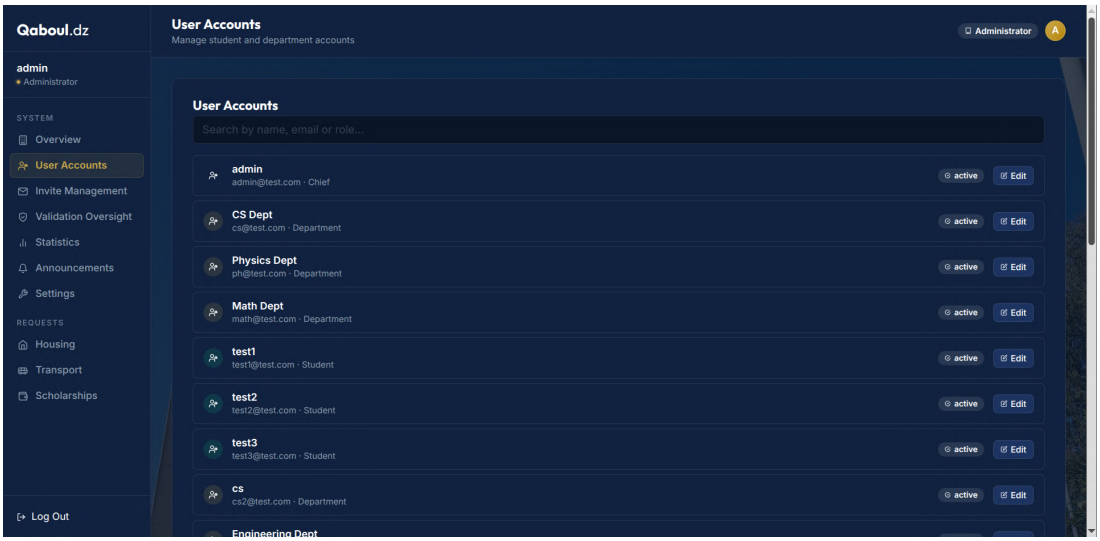


Figure 36: Interface de gestion des comptes utilisateurs

12.2 Page Paramètres Système (Settings)

L'interface d'administration offre un panneau de configuration complet du système d'inscription. L'administrateur peut définir le nom du système (*Qaboul Enrollment System*), les dates de début et de fin de la période d'inscription, la date limite de soumission des dossiers, la date d'approbation ainsi que la date limite de recours ou correction.

Le nombre maximum de documents par dossier est également paramétrable.

Deux boutons d'action sont disponibles :

- *Save Settings* : permet d'enregistrer les modifications,

- *Reset to Default* : rétablit les valeurs par défaut.

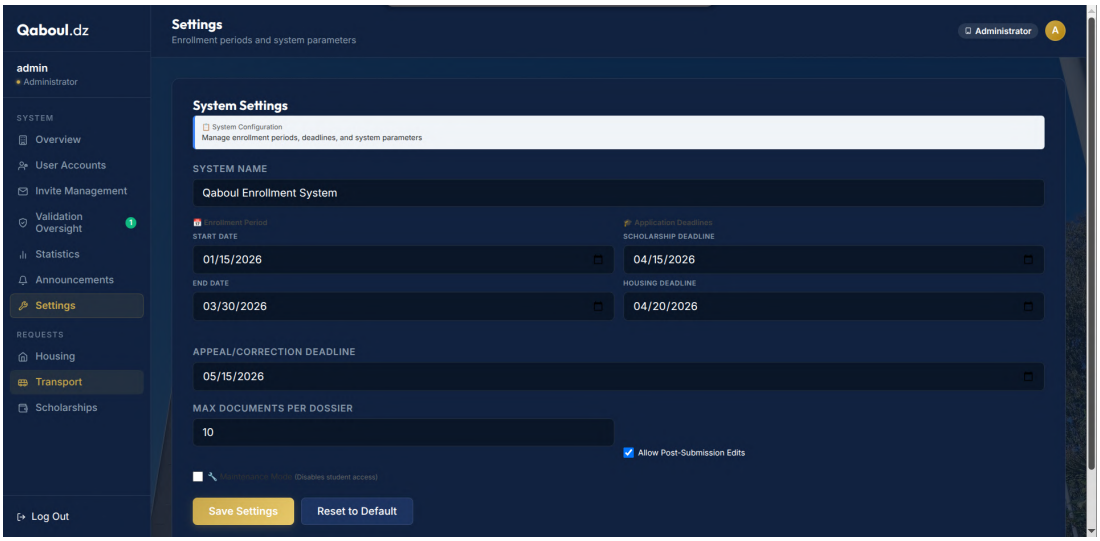


Figure 37: Interface des paramètres système (Administrateur)

12.3 Gestion des Invitations (Invite Management)

L'interface *Invite Management* permet à l'administrateur de générer des liens d'invitation sécurisés destinés à la création de nouveaux comptes de type département ou administrateur.

Afin d'assurer la sécurité du système, les liens générés sont à usage unique et possèdent une durée de validité limitée à 24 heures. Cette approche permet de réduire les risques liés aux accès non autorisés et de mieux contrôler la création des comptes utilisateurs.

L'administrateur doit sélectionner le type du compte à créer, soit *Department Account* ou *Admin Account*, puis cliquer sur le bouton *Generate Invite Link* afin de générer le lien d'invitation qui sera transmis au futur utilisateur.

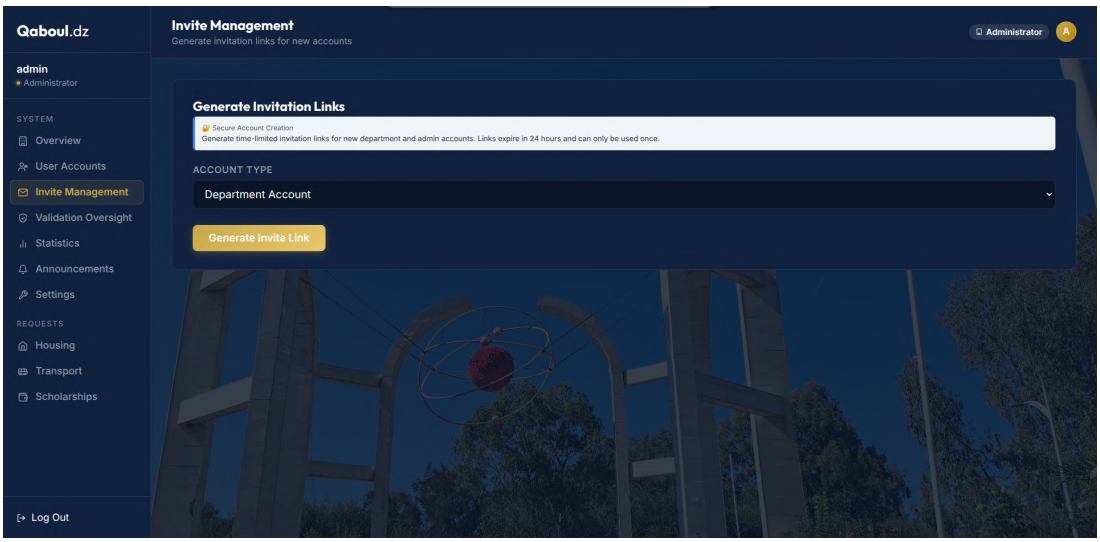


Figure 38: Interface de gestion des invitations

13 Interface Département

13.1 Gestion des Dossiers par Département

L'interface département permet au responsable du département (par exemple *CS Dept*) de consulter et traiter les dossiers d'inscription soumis par les étudiants. Un tableau de bord est mis à disposition afin d'afficher en temps réel différentes statistiques telles que le nombre total de dossiers, les dossiers en attente (*Pending*), les dossiers validés ainsi que ceux rejetés.

Chaque dossier contient plusieurs informations importantes, notamment le nom de l'étudiant, son niveau d'étude (*Licence L1, L2 ou L3*), son score au baccalauréat ainsi que la date de soumission du dossier.

Le responsable du département dispose également de plusieurs actions pour le traitement des dossiers. Il peut accepter un dossier (*Accept*), le rejeter (*Reject*) ou demander une correction (*Correction*) lorsque des modifications sont nécessaires. Un champ de commentaire optionnel est également disponible afin d'ajouter des remarques ou des observations complémentaires.

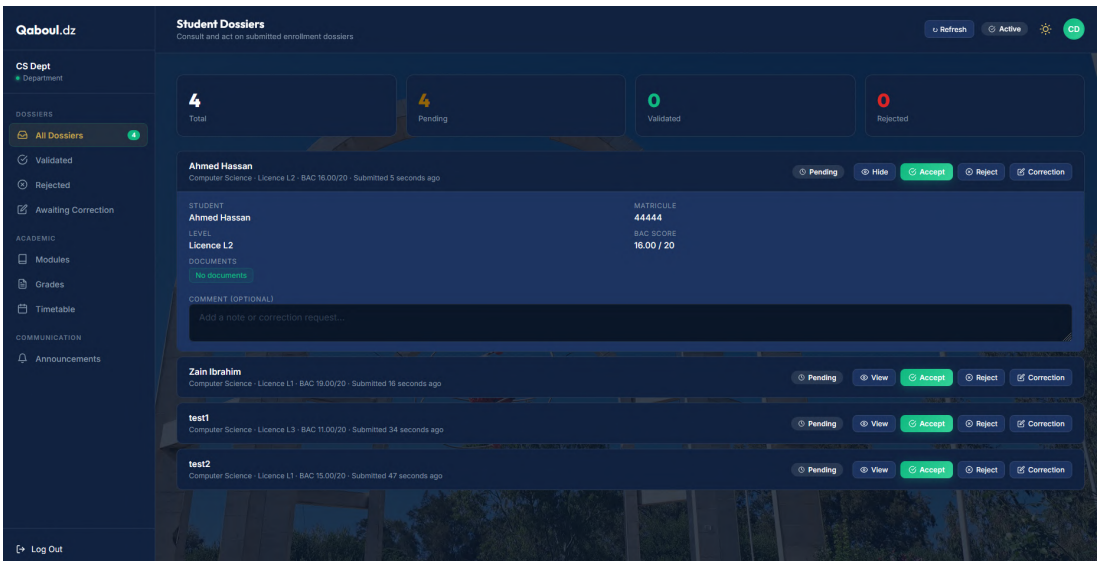


Figure 39: Interface de gestion des dossiers par département

14 Conclusion

Ce chapitre a présenté les principales interfaces réalisées dans le cadre du développement de la plateforme *Qaboul.dz*.

L'interface publique assure une présentation claire de la plateforme aux nouveaux utilisateurs. L'espace étudiant offre des fonctionnalités complètes de gestion du dossier d'inscription, de demandes de services (logement, transport), de consultation du profil et de l'emploi du temps.

Enfin, le panneau d'administration permet un contrôle total sur les paramètres du système d'inscription.

L'ensemble de ces interfaces témoigne d'une conception centrée sur l'utilisateur, alliant ergonomie et fonctionnalité.

Conclusion Générale

Ce projet nous a permis de concevoir et de développer **Qaboul.dz**, une plateforme web centralisée pour la gestion des inscriptions universitaires en Algérie. En s'appuyant sur des technologies modernes telles que *React.js*, *Laravel* et *SQLite*, nous avons pu proposer une solution numérique adaptée aux besoins des étudiants, des départements et des administrateurs.

Ce travail nous a permis d'acquérir des compétences solides en développement web *full-stack* et en modélisation UML, tout en répondant à une problématique concrète liée à la gestion administrative universitaire. Des améliorations futures, telles que l'intégration de notifications en temps réel ou le déploiement à l'échelle nationale, pourraient enrichir davantage cette plateforme.

References

[1] Microsoft. *Visual Studio Code – Documentation officielle*. Disponible sur : `https://code.visualstudio.com/docs`

[2] Mozilla Developer Network (MDN). *CSS : Cascading Style Sheets*. Disponible sur : `https://developer.mozilla.org/fr/docs/Web/CSS`

[3] Meta Open Source. *React – A JavaScript library for building user interfaces*. Disponible sur : `https://react.dev/`

[4] Axios. *Axios – Promise based HTTP client for the browser and node.js*. Disponible sur : `https://axios-http.com/docs/intro`

[5] npm. *npm – Node Package Manager*. Disponible sur : `https://www.npmjs.com/`

[6] The PHP Group. *PHP : Hypertext Preprocessor – Documentation officielle*. Disponible sur : `https://www.php.net/docs.php`

[7] Laravel LLC. *Laravel – The PHP Framework for Web Artisans*. Disponible sur : `https://laravel.com/docs`

[8] SQLite Consortium. *SQLite – Documentation officielle*. Disponible sur : `https://www.sqlite.org/docs.html`

[9] GitHub Inc. *GitHub – Documentation officielle*. Disponible sur : `https://docs.github.com/`

[10] Image de fond du campus universitaire. Disponible sur : `https://fs-umbb-beryl.vercel.app/campus3.jpg`