

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي



République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid -Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'étude Pour l'Obtention du diplôme de Master en
Informatique sous l'arrêté ministériel N°1275

Option : Génie Logiciel (GL)

THEME :

Application de Gestion d'Hébergement et de Bourses pour
la Direction des Œuvres Universitaires

Réalisé par :

- **Mr AZZOUZ Abdelkader**

Soutenu le 24 Juin 2025 devant la commission composée de :

Présidents :	Mr BENAMMAR Abderahim
Encadreur :	Mr MERZOUG Mohammed
Examinatrice :	Mme Yamouni

Année universitaire : 2024 / 2025

Dédicace

Je dédie ce travail :

Je voudrais dédier mon travail à toutes les personnes qui m'ont aidé durant mon cursus
Universitaire.

Je voudrais principalement remercier mes chers parents, mes chers frères et aussi ma chère femme, ma grand-père ainsi que toute ma famille sans qui je n'aurais jamais pu faire ce travail.

À mes amis et collègues, avec qui j'ai partagé des moments de travail, d'entraide et de réussite.

Remerciements

Je voudrais remercier dieu "الله عز وجل" pour toute l'énergie qu'il m'a donné durant ces années.

Je tiens à remercier sincèrement tous ceux qui ont contribué de près ou de loin à la réalisation de ce projet.

Mes remerciements s'adressent particulièrement à mon encadrant pour ses conseils, son accompagnement et sa disponibilité tout au long de ce travail.

Je remercie également mes enseignants, mes camarades et ma famille pour leur soutien moral et leur encouragement constant.

Merci à tous !

Abdelkader AZZOUZ

Liste des figures

Figure 1 : La position géographique de la direction des œuvres universitaires Mansourah Tlemcen	5
Figure 2 : Organigramme de la direction des œuvres universitaires	8
Figure 3 : Diagramme de classes du système de gestion des demandes d’hébergement et de bourse	21
Figure 4 : Diagramme de séquence – Authentification de l’étudiant.....	23
Figure 5 : Diagramme de séquence – Authentification de l’administrateur.....	24
Figure 6 : Diagramme de séquence – Soumission d’une demande par un étudiant.....	25
Figure 7 :Diagramme de séquence – Traitement, filtrage et analyse des demandes par l’administrateur	29
Figure 8 : Diagramme de séquence – Interaction entre l’étudiant et l’administrateur	33
Figure 9 : Diagramme des cas d’utilisation illustrant les interactions entre	35
Figure 10 : Interface d'authentification Admin	42
Figure 11 : Interface Statistique et widget	42
Figure 12 : Interface Ajouter et importation des étudiants.....	43
Figure 13 : Interface Filtrage des étudiants	43
Figure 14 : Interface liste des étudiants faire les Demandes	44
Figure 15 : Interface Création Compte Etudiants.....	45
Figure 16 : Interface d'authentification Etudiants	45
Figure 17 : Interface du menu du tableau de bord étudiant.....	46
Figure 18 : Interface Statuts et Notifications Etudiants	47
Figure 19 : Interface du formulaire automatique pour la demande étudiante.	47
Figure 20 : Interface des statuts pour la demande étudiante	48

Liste des tableaux

Tableau1 Modalités de Bourse Universitaire pour les étudiants nationaux 11

Tableau 2 Analyse comparative de l'application réalisée et des applications similaires 16

Table des matières

I. Table des matières

Introduction Générale	1
Chapitre 1 : Organisation des Œuvres Universitaire et Définition des Services	4
I. Présentation de l'organisme d'accueil	5
A. Position géographique :	5
II. Présentation de la direction des œuvres universitaires :	5
1. Historique de la Direction (DOUTM)	5
2. Fonctions et objectifs de la (DOUTM)	6
3. Organigramme de la direction des œuvres universitaires :	7
B. service de l'hébergement :	8
1. Les tâches du service de l'hébergement	8
2. Description des tâches	9
C. service de la bourse :	10
1. Critère pour bénéficier de bourse universitaire :	10
2. Condition que l'étudiant :	11
D. Comparaison avec des systèmes similaires	14
1. Introduction des systèmes similaires	14
2. Le système CROUS – France :	14
3. Comparaison avec application proposée	16
4. le système souple et adapté aux universités	17
Chapitre 2 : Modalisation et Conception de L'Application :	18
E. Analyse des besoins fonctionnels techniques	19
1. Besoins fonctionnels	19
F. Modélisation UML	21
1. Diagramme de classes :	21
2. Diagramme séquence login étudiants	23

3. Diagramme séquence login administrateur	24
4. Diagramme de séquence pour la demande des étudiants	25
5. Diagramme de séquence de l'administrateur (gestion, Statistique, filtrage des demandes, etc.).....	29
6. Diagramme de séquence – Interaction étudiant/administrateur dans la gestion des demandes.....	33
1. Introduction	38
2. Langage utilisé :	38
2.1. Langue de l'interface utilisateur (UI).....	38
2.2. Langages de programmation et technologies utilisées	38
2.2.1. Langage côté serveur (back-end).....	38
2.2.2. Langage côté client (front-end).....	39
3. Langages de documentation et modélisation	41
3.1.Définition :	41
3.2. Langage utilisé par PlantUML :	41
3.3. Types de diagrammes UML supportés par PlantUML :	41
G. Interfaces principales	42
1. Cote Administrateur :	42
3.4. Tableaux de Bord Administrateur :.....	42
2. Importation des fichiers (listes des étudiants) :	43
3. Filtrage des étudiants par Cycle :	43
4. Liste des demandes pour traiter :	44
5. Coté Utilisateur (étudiants) :.....	45
6. Coté Utilisateur (login étudiants) :	45
7. Tableaux de Bord Utilisateur (étudiants) :	46
8. Interface Statuts et Notification d'étudiants :	47
9. Formulaire on ligne des demandes :	47
10. Statuts d'hébergements d'un étudiant pour faire la demande :	48

Conclusion et perspectives	49
A. Conclusion générale	50
B. Perspectives d'amélioration	51
C. Bilan personnel.....	51

Introduction Générale

Dans le cadre de la politique sociale visant à soutenir les étudiants tout au long de leur vie universitaire, les œuvres universitaires jouent un rôle impératif en offrant des services tels que l'hébergement, la restauration, les bourses et le transport. Ces services jouent un rôle central dans la fourniture d'égalité des chances, l'évitement des abandons scolaires et l'amélioration des conditions de vie des étudiants. Pourtant, la gestion semi-informatisée ou manuelle de ces services entraîne des retards réguliers, des erreurs, un manque de transparence et une charge administrative. Comme solution à ces problèmes, l'utilisation d'une application web centralisée et automatique est inévitable afin d'optimiser le traitement des demandes, de faciliter le processus de prise de décision pour l'administration, et de permettre aux étudiants de suivre leur dossier de manière claire et efficace. C'est avec cette vision en tête que notre projet intitulé "Gestion de l'hébergement et des bourses des travaux universitaires" est né. Ce projet vise à concevoir et développer une plateforme numérique qui sera bénéfique pour les étudiants afin de soumettre leurs candidatures en ligne, de suivre l'avancement de leurs dossiers, et pour l'administration afin de traiter les demandes en temps opportun, de gérer les affectations de chambres et d'allouer les bourses financières selon des critères prédéfinis. Ce système est construit sur une architecture web moderne qui utilise des technologies telles que Laravel, PHP, MySQL, Bootstrap et Livewire. Il dispose de deux interfaces différentes : l'une pour les administrateurs et l'autre pour les étudiants, chacune conçue pour répondre à leurs besoins spécifiques. Cet article dévoile donc les différentes étapes de conception, de développement et de mise en œuvre de ce système, tout en élaborant sur les décisions technologiques ainsi que sur les solutions proposées aux problèmes de gestion existants des travaux universitaires.

❖ Contexte

Les Œuvres Universitaires jouent un rôle pivot dans la vie étudiante, particulièrement pour les étudiants venant de régions éloignées ou de familles à faibles ressources. L'accueil des demandes d'hébergement et de bourses fait partie intégrante des activités de la Direction des Œuvres Universitaires (DOU), qui a la charge d'assurer un environnement de vie convenable et une assistance financière aux étudiants. Ce rôle est essentiel pour favoriser l'accès à l'éducation supérieure, en permettant aux étudiants de se concentrer sur leurs études sans être préoccupés par leur logement ou leur situation financière.

Traditionnellement, la gestion des demandes d'hébergement et de bourses repose sur des méthodes manuelles et des processus papier, ce qui engendre souvent des inefficacités. De nombreuses universités et institutions font face à des défis croissants en termes de volumes de demandes à traiter et de la gestion des données des étudiants. Ce problème est accentué par l'absence de système centralisé et automatisé, qui complique la prise de décisions et augmente les risques d'erreurs humaines.

❖ **Problématique**

Malgré l'importance des services d'hébergement et de bourse dans le soutien aux étudiants, leur gestion reste lourde et inefficace dans de nombreuses universités. L'utilisation de procédures manuelles, le manque de centralisation des informations et l'absence de suivi numérique entraînent des retards dans le traitement des dossiers, des erreurs d'attribution et un manque de transparence. Par conséquent, une question fondamentale se pose : Comment concevoir et réaliser un système informatisé capable de gérer efficacement les demandes de logement et de bourse des étudiants, avec un suivi facile, un traitement sécurisé et un accès pratique pour les différents acteurs impliqués ?

❖ **La Solution Proposée**

Le projet a pour objectif de développer une application web centralisée de gestion des demandes d'hébergement et de bourses des étudiants destinée à la Direction des Œuvres Universitaires (DOU). Cette plateforme est conçue pour alléger les démarches administratives, pour diminuer les erreurs et pour augmenter la transparence du processus.

➤ **Portail étudiant :**

- Soumission en ligne des demandes d'hébergement et de bourses
- Suivi en temps réel des statuts (en cours, accepté, refusée)
- Notifications automatiques.

➤ **Portail Administrateur :**

- Tableau de bord de la gestion centralisée des demandes
- Attribution manuelle ou automatique des chambres et bourses conformément à des critères fixés.
- Notifications pour les nouvelles demandes ou en attente

➤ **Sécurité & Centralisation :**

- Base de données sécurisée pour toute donnée sensible
- Authentification et contrôle des droits d'accès

➤ Statistique & Analyse :

Statistiques créées sur les demandes et les attributions. Aide à la prise de décision grâce aux indicateurs disponibles.

Avantages de la Solution

Effet administratif : automatisation des processus et simplification des tâches manuelles.

- Transparence et accessibilité pour les étudiants.
- Réduction des erreurs et des litiges.
- Meilleure répartition des ressources (places en résidence, bourses).

Cette solution de pointe répond aux besoins de modernisation croissants des services universitaires en se concentrant sur l'efficacité, la transparence, et une expérience utilisateur optimale pour les étudiants aussi bien que les administrateurs.

Ce mémoire est structuré autour de trois chapitres principaux, précédés d'une introduction générale, pour présenter la conception et la réalisation d'une application de gestion des demandes d'hébergement et de bourses dans le cadre des Œuvres Universitaires.

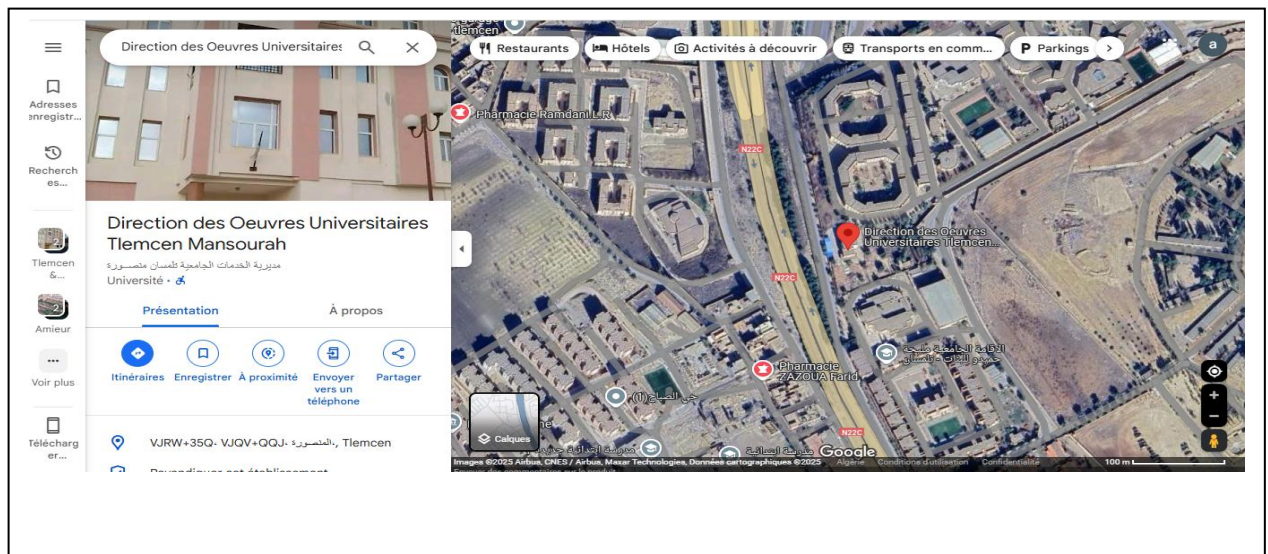
Chapitre 1 : Organisation des Œuvres Universitaire et Définition des Services

I. Présentation de l'organisme d'accueil

A. Position géographique :

Mitoyennes du université AbouBakrBelkaidTlemcen [1] , les neufs résidences des œuvres universitaireMansourah Tlemcen sont situées à **Adresse : Cité universitaire Malika Kaïd - Mansourah - Tlemcen**

Tél : +213 43 38 15 19



❖ Coordonnées GPS et Google Map **34.89032067666966, -1.3545124760124607**

Figure 1 : La position géographique de la direction des œuvres universitaires Mansourah Tlemcen

II. Présentation de la direction des œuvres universitaires :

1. Historique de la Direction (DOUTM) [2]

Aperçu sur l'histoire des œuvres universitaires Tlemcen. Le COUS (Centre des Œuvres Universitaires et Scolaires), fut créé en 1975, et fournissait ses prestations dans les quartiers universitaires à partir de l'hôtel ALBERT, et l'hôtel Agadir jusqu'au 31/12/1986, où ils furent transférés dans les résidences universitaires, telles que: Bekhti AbdEl Madjid, et 19 mai 1956. A compter du 01/01/1978 la formule COUS a été renommée en COSU (Centre des Œuvres Scolaires et Universitaires) où les mêmes fonctions ont été persistées, telles que : l'hébergement, les bourses, le transport universitaire , la restauration, ainsi que des subventions aux activités culturelles , sportives et scientifiques. par la suite le COSU Tlemcen a été élargi par d'autre structures d'hébergements. COSU est resté actif jusqu'au 30/06/1996.

Au début de cette date les centres des œuvres universitaires ont été placés directement sous la tutelle du ministère de l'enseignement supérieur et de la recherche scientifique. Puis sous la tutelle de l'office nationale des Œuvres universitaires. Suite à cette décision les résidences universitaires sont devenues autonomes sur le plan de gestion financière avec la création d'une coordination de wilaya entre les résidences universitaires, de telle sorte que le coordinateur doit exercer ses fonctions en coordination avec le représentant régional. Cette structure est restée jusqu'au 30/06/2005 où une restructuration en conformité avec la création de la Direction des œuvres universitaires (D.O.U) a été décrétée. En particulier dans la wilaya de Tlemcen.

2. Fonctions et objectifs de la (DOUTM)

La direction des œuvres universitaires TLEMCEN MANSOURAH fait bon usage des structures d'hébergement et de restauration, à l'instar des autres directions des œuvres universitaires de la wilaya de TLEMCEN, œuvre de manière à offrir à la communauté étudiante résidente les meilleures conditions de vie pour leur épanouissement ; Des conditions inspirées des meilleures universités du monde imprégnée de l'esprit des valeurs du savoir, à la quête de la connaissance, qui visent de manière permanente à concrétiser la bonne connaissance scientifique et ancrer la citoyenneté chez les étudiants(es) résidents(es) dans le but d'investir dans la ressource humaine à même de contribuer au développement socio-économique du pays. Dans le même sillage, la direction des œuvres universitaires de Tlemcen Mansourah tente toujours d'être à l'avant garde pour exécuter les orientations et les instructions de la direction générale de l'Office des œuvres universitaires dans la perspective de réaliser les objectifs de la communauté étudiante qui consistent à mettre l'étudiant résident au centre de l'opération d'apprentissage pour pouvoir à la fin de son parcours universitaire être un bon citoyen au service de son pays et ouvert sur le monde, étant en mesure de résoudre ses propres problèmes. Des objectifs que se sont fixés les cadres de la direction des œuvres universitaires TLEMCEN MANSOURAH ainsi que tous les employés et travailleurs dépendants de la direction ainsi que tous les partenaires sociaux tels que les syndicats des étudiants. En multipliant tout les efforts pour réaliser l'objectif assigné consistant à offrir aux étudiants un cadre d'épanouissement. La direction des œuvres universitaires Tlemcen Mansourah a pour missions la prise en charge totale en matière d'application de la politique nationale des universitaires, le contrôle et la coordination entre les résidences universitaires.

Comme elle est chargée de veiller à l'amélioration des conditions de vie de l'étudiant à l'intérieur de la résidence universitaire ainsi que l'élaboration du plan de transport des résidences universitaires et le suivi de son exécution. Parmi ses autres prérogatives, on relèvera la promotion des activités scientifiques, culturelles, sportives et loisirs dans un cadre d'hygiène et de sécurité.

3. Organigramme de la direction des œuvres universitaires :

La direction des œuvres universitaires Tlemcen Mansourah (DOUTM) est un établissement public à caractère administratif placé sous la tutelle directe de l'office national des œuvres universitaires et le ministère de l'enseignement supérieur et de la recherche scientifique,

elle est composée de sept (08) résidences universitaires:

- Résidence Universitaire Maliha Hamidou
- Résidence Universitaire Malika Kaïd
- Résidence Univeristaire Mansourah 3
- Résidence Universitaire Mansourah 4
- Résidence Universitaire Mansourah 5
- Résidence Universitaire Mansourah 6
- Résidence Universitaire Mansourah 7
- Résidence Universitaire 900 lits
- Résidence Universitaire TidjaniHadam

Ses rôles principaux: La direction des œuvres universitaires Tlemcen Mansourah (DOUTM) applique la politique nationale dans le domaine des œuvres universitaires et assure l'amélioration continue des conditions de vie matérielle et morale de l'étudiant(e) inscrit dans des établissements d'enseignement supérieur national. Assurer le suivi, la coordination et le contrôle des résidences universitaires. Fournir et améliorer les conditions d'hébergement et de restauration des étudiants(es). Contribuer à la préparation et l'exécution du plan de transport universitaire. Suivre, contrôle et gestion des bourses universitaires. Développement des activités culturelles, sportives et scientifique. Assurer la protection de santé des étudiants(es) par le biais des Unités Médicales Préventives (U.M.P). La réception des étudiants(es) étrangers inscrits dans des établissements de l'enseignement supérieur national conformément aux règles édictées par la loi. En conformité avec la réglementation applicable administrative

au niveau de l'office national des œuvres universitaires et la répartition des tâches des services concernés sur le terrain, la direction des œuvres universitaires Tlemcen Mansourah parraine le travail avec soin et intérêts.

Voici l'organigramme des différents services de cette direction, qu'il sera présenté comme suit

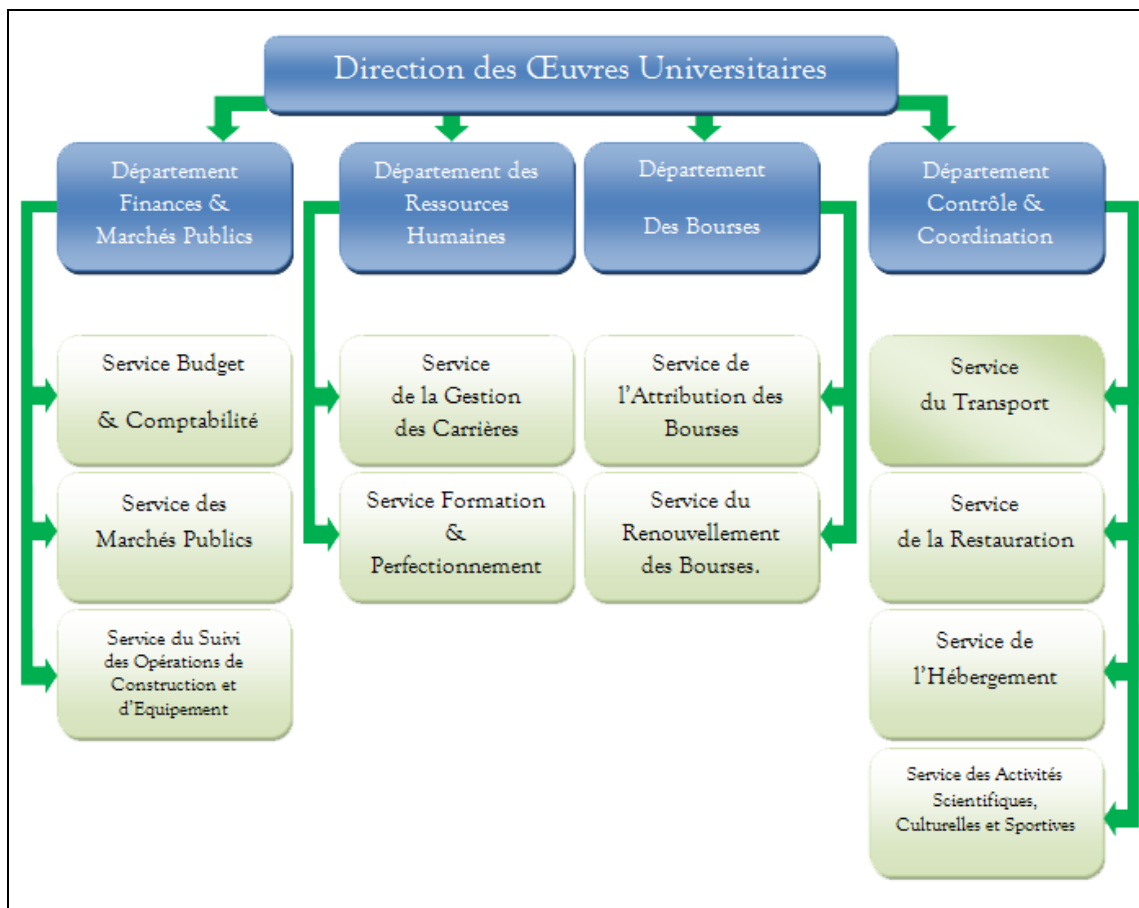


Figure 2 : Organigramme de la direction des œuvres universitaires

B. service de l'hébergement [3]:

L'hébergement est l'un des principaux objectifs de la direction des œuvres universitaires Tlemcen Mansourah (DOUTM), avec un total de sept (09) résidences universitaires dispersées autour de l'université de TLEMCEN, qui accueilleraient des étudiants(es) de différents coins du territoire national, y compris des étudiants(es) étrangers (ères).

1. Les tâches du service de l'hébergement

Les tâches inhérentes au service de l'hébergement se récapitulent comme suit :

- Effectuer les inscriptions des nouveaux bacheliers reçus et la réinscription des anciens étudiants se fera au niveau de chaque résidence.
- Etablissement des listes globales des étudiants et leurs répartitions par résidence, et par bloc.

- Etablissement des statistiques sur l'état de l'hébergement des résidences (nombre de places libres, les abandons, ... etc.).
- Préparation des procès verbaux globaux.- Contrôle des dossiers.- ... etc.

2. Description des tâches

1. Inscription

Chaque nouveau bachelier qui se présente pour l'inscription au sein du service de l'hébergement au niveau de la D.O.U, doit être muni d'un dossier complet et une décision d'attribution de chambre lui sera remise. L'étudiant se présente à nouveau avec cette décision, auprès du service de l'hébergement afin de régler les frais du loyer et du transport. En revanche, il a une affectation de chambre, une prise en charge de couchage et une carte de résidence lui est délivrée en même temps.

2. Réadmission

Elle concerne les anciens résidents, l'étudiant doit exprimer son vœux de renouveler sa chambre en fin d'année pour l'année d'après, le renouvellement de chambre se compose de trois phases :

- Première phase : en fin d'année, l'étudiant doit fournir une demande de réadmission, visée par le médecin de la résidence, une photo récente et une enveloppe timbrée et libellée à l'adresse du résident.
- Deuxième phase : l'étudiant doit rendre son couchage, la clé de sa chambre, ainsi la carte de résidence.
- Troisième phase : à l'entrée universitaire, l'étudiant fournit une copie du certificat de scolarité de l'année en cours et paie les frais du loyer et transport. Par contre, il aura une prise en charge de couchage et une nouvelle carte de résidence lui sera remise.

3. Transfert

Concernant les transferts d'une résidence à une autre, le résident doit fournir une demande de transfert suivie de l'attestation d'inscription où la carte de résidence de l'année en cours au sein de sa résidence, le transfert ne s'effectuera que s'il y a un accord entre la résidence d'origine et la résidence d'accueil en fonction du nombre de places disponibles.

4. Abandon

Lorsqu'un étudiant ne se présente pas à la réadmission ou lorsqu'il retire son dossier d'inscription où il est déclaré défaillant par le conseil qui se déroule au niveau de la D.O.U ou

de la résidence. A ce moment la, l'étudiant est déclaré abandon, par la suite s'il désire se reloger, il doit fournir une justification qui lui permet une réadmission au sein de la résidence sinon l'étudiant sera en état exclu.

C.service de la bourse [4]:

1. Critère pour bénéficier de bourse universitaire :

Afin de bénéficier de la bourse universitaire, postulez à travers la Platform de **Progress** progrès de pour les nouveaux étudiants en parallèle avec le processus d'orientation et de confirmation à partir du 04 Septembre 2021.

« <https://progres.mesrs.dz/webonou> »

Le dossier sera enseigné au niveau des comités concernés selon des critères spécifiques.

L'étudiant nécessite doit renouveler son dossier au début de chaque année Universitaire.

Le dépôt des dossiers de Bourse sera à partir toujours la fin septembre à chaque Année.

❖ Note :

Pour bénéficier de la bourse universitaire, il est obligatoire de passer par la Platform de **Progress**.

Sinon, l'étudiant perd ses droits de bourse universitaire de l'année universitaire current.

Chaque étudiant bénéficie d'une bourse Universitaires, Estimée à **4 050.00 DA** ou **3 600.00 DA** ou **2 700.00 DA** Pour les études du 1^{er} Cycle pour les élèves et **5 850.00 DA** ou **7 200.00 DA** Pour les études du 2^{ème} Cycle Selon le Revenu Annuel Total des Parents et selon l'année scolaire de l'étudiant en proportion de la deuxième Cycle, les étudiants payés tous les trois mois (**Trimestriel**).

Cycle		Montan de Bourse Trimestriel	Tableau Conditions salaire des parents
1 ^{er} Cycle		4 050,00 DA	- Le revenu des parents est inférieur ou égal à 04 fois le salaire minimum national garanti SNMG.
		3 600,00 DA	- Le revenu des parents est inférieur ou égal à 06 fois le salaire minimum national garanti SNMG.
		2 700,00 DA	- Le revenu des parents est inférieur ou égal à 07 fois le salaire minimum national garanti SNMG.
2 ^{eme} Cycle	Master 01	5 850,00 DA	- Le revenu des parents ne devrait pas dépasser 08 fois le salaire minimum national garanti.
	Mastero2	7 200,00 DA	- Le revenu des parents ne devrait pas dépasser 08 fois le salaire minimum national garanti.

Tableau1Modalités de Bourse Universitaire pour les étudiants nationaux

2. Condition que l'étudiant :

- Soit inscrit au niveau d'un établissement de l'enseignement supérieur.
- N'ayant pas un retard de scolarité supérieur ou égal à 04 semestres.
- Le Revenu des parents ne doit pas dépasser 08 fois le SNMG.
- Il n'a pas de congé académique ou la somme de sa note dans l'année en cours n'est pas égale à zéro.
- Un Extrait de rôle des parents apuré (net de tous dettes).

Dossier de bourse Cycle 01 Licence à fournir :

Dossier de dépôt licence :

- Copie Certificat original d'inscription de l'année universitaire courante.
- Copie Relevé de notes du Bac.

- Chèque CCP (barré) de l'étudiant.
- Relevé des émoluments annuels des parents.
- Un Extrait de rôle des parents apuré pour les parents commerçants et artisans.
- Déclaration d'inscription de bourse Imprimé de site de Progress.

Note a propos le Relevé des émoluments des parents :

➤ **Les parents salariés :**

- Relevé du salaire annuel.
- Extrait de rôles appuré.

➤ **Les Retraités :**

- Relevé de la pension.
- Extrait de rôles appuré.

➤ **Les commerçants et artisans :**

Bulletin des revenus des commerçants **C20**, délivrés par les services des impôts.

Extrait de rôles dépourvu de dettes.

➤ **Les non-salariés et chômeurs :**

- Attestation de non-activité délivrée par l'APC.
- Extrait de rôles appuré.

➤ **Les personnes décédées ou divorcées :**

- Extrait des actes de décès (pour les parents décédés).
- Certificat de divorce (parents divorcés).

Observation :

La réponse aux demandes de Bourse s'effectuera en fonction des critères d'accès à ce service, le tirage de la décision sera ce site web à fin du mois septembre pour les étudiants inscrit aux universités de Tlemcen.

Dossier de Renouveau licence :

- Chèque CCP (barré) de l'étudiant.
- Copie du certificat d'inscription de l'année universitaire courant.
- Copie de l'attestation du succès.

Dossier de bourse Cycle 02 Master fournir :

Dossier de dépôt Master :

- Chèque CCP (barré) de l'étudiant.
- Copie Certificat original d'inscription de l'année universitaire courante.
- Copie Relevé de notes du Bac.
- Copie de Diplôme de Licence.
- Relevé des émoluments annuels des parents.
- Un Extrait de rôle des parents apuré pour les parents commerçants et artisans.
- Attestation de non-activité délivrée par l'APC de l'étudiant.

Note a propos le Relevé des émoluments des parents :

Les parents salariés :

- Relevé du salaire annuel.
- Extrait de rôles apuré.

Les Retraités :

- Relevé de la pension.
- Extrait de rôles apuré.

Les commerçants et artisans :

- Bulletin des revenus des commerçants C20, délivrés par les services des impôts.
- Extrait de rôles dépourvu de dettes.

Les non-salariés et chômeurs :

- Attestation de non-activité délivrée par l'APC.
- Extrait de rôles apuré.

Les personnes décédées ou divorcées :

- Extrait des actes de décès (pour les parents décédés).
- Certificat de divorce (parents divorcés).

Dossier de Renouveau Master :

- Chèque CCP (barré) de l'étudiant.
- Copie du certificat d'inscription de l'année universitaire courant.
- Copie de l'attestation du succès.
- Attestation de non-activité délivrée par l'APC de l'étudiant.

D. Comparaison avec des systèmes similaires

Voici un exemple de **texte rédigé complet** contenant :

- Une **introduction du système PROGRES (Algérie)**
- Une **introduction du système CROUS (France)**
- Une **comparaison détaillée** avec votre propre application de gestion d'hébergement et de bourses.

1. Introduction des systèmes similaires

A. Le système PROGRES – Algérie [5] :

Le système PROGRES (Plateforme de Gestion des Étudiants) est une plateforme numérique nationale développée par le Ministère de l'Enseignement Supérieur et de la Recherche Scientifique (MESRS) en Algérie. Elle a pour objectif de centraliser et de moderniser la gestion administrative des étudiants dans les établissements universitaires algériens.

Cette plateforme regroupe plusieurs fonctionnalités essentielles telles que :

- L'inscription en ligne aux établissements,
- La demande de bourse,
- La demande d'hébergement,
- Le retrait de documents (certificats de scolarité, relevés de notes),
- Et le suivi académique.

PROGRES est accessible à tous les étudiants via un portail sécurisé, mais son interface et ses processus sont parfois jugés complexes ou peu intuitifs par les utilisateurs. De plus, bien qu'elle soit nationale, la plateforme offre peu de marge d'adaptation locale pour les Directions des Œuvres Universitaires (DOU), ce qui limite sa flexibilité dans le traitement spécifique des demandes.

2. Le système CROUS – France [6]:

Le **Centre Régional des Œuvres Universitaires et Scolaires (CROUS)** est un organisme public français chargé d'améliorer les conditions de vie des étudiants. Il offre, via le portail national **messervices.etudiant.gouv.fr**, un accès numérique aux services de demande de bourse sur critères sociaux, de logement universitaire, de restauration, d'activités culturelles, et de soutien social.

Le CROUS propose une gestion intégrée et automatisée des demandes. Le système est relié à d'autres bases de données publiques (comme les revenus fiscaux) pour simplifier les

démarches des étudiants. Son interface est moderne, accessible, et conforme aux normes de sécurité européennes (RGPD).

Ce système favorise l'autonomie de l'étudiant dans ses démarches, tout en assurant une bonne traçabilité et un traitement équitable grâce à des critères clairs et automatisés. Cependant, il peut parfois manquer de souplesse pour les cas particuliers, notamment ceux ne répondant pas exactement aux critères standards.

3. Comparaison avec application proposée

Mon projet d'**Application de Gestion d'Hébergement et de Bourses pour la Direction des Œuvres Universitaires (DOU)** vise à combler certaines lacunes observées dans les systèmes existants. Il s'inscrit dans une logique de **digitalisation ciblée et personnalisée**, adaptée au contexte spécifique de chaque DOU.

Critères	PROGRES (Algérie)	CROUS (France)	application DOU
Objectif principal	Centraliser la gestion des étudiants	Faciliter l'accès aux aides pour étudiants	Gérer localement les bourses et hébergement
Niveau d'échelle	National	National	National ou local
Flexibilité	Faible personnalisation locale	Critères stricts centralisés	Adaptable aux besoins locaux
Interface	Complexe, peu intuitive	Moderne et ergonomique	Simplifiée et adaptée à l'utilisateur
Accès	Étudiant via identifiant universitaire	Étudiant via identifiant national	Étudiant via numéro bac ou matricule
Services inclus	Bourse, hébergement, inscription, certificats	Bourse, hébergement, restauration	Bourse, hébergement, suivi des demandes
Notifications	Faibles, souvent par email	Notifications intégrées	Notifications en temps réel
Technologies utilisées	Java EE, Oracle	PHP, PostgreSQL, Web services	Laravel (PHP), MySQL, Livewire, Bootstrap
Sécurité / Confidentialité	Conforme aux normes du MESRS	Conforme RGPD	Conforme RGPD et login étudiant

Tableau 1 Analyse comparative de l'application réalisée et des applications similaires

4. le système souple et adapté aux universités

La solution représente une **approche plus ciblée, légère et agile**, particulièrement adaptée à un environnement régional. Contrairement aux systèmes centralisés comme PROGRES ou CROUS, elle permet une **gestion autonome et personnalisée** des demandes selon les critères et contraintes spécifiques de chaque établissement universitaire. Cela permet de répondre efficacement aux attentes des étudiants tout en facilitant le travail administratif des gestionnaires locaux.

Chapitre2 :Modalisation et Conception de L'Application

E. Analyse des besoins fonctionnels techniques

1. Besoins fonctionnels

L'application doit permettre la gestion des services d'hébergement et de bourse pour les étudiants universitaires. Elle est destinée à trois types d'utilisateurs : **les étudiants**, et l'équipe administrative (chef de service et les agents de chaque service soit hébergement et bourses).

1. Authentification & Gestion des comptes

- Authentification pour les étudiants à l'aide de :
 - ✓ Année du bac + Matricule bac
- Avant de permettre la création d'un compte étudiant, le système doit **vérifier** si la combinaison **Année du bac + Matricule bac** existe déjà dans la base de données. Cela sert à **confirmer l'identité de l'étudiant** avant qu'il puisse créer un compte.
- Authentification pour les administrateurs et agents :
 - ✓ Email + mot de passe sécurisé
- Gestion interface login **chaque étudiant et chaque administrateur ait sa propre interface de connexion (login)**

2. Gestion des demandes d'hébergement

1. Remplissage en ligne du formulaire de demande d'hébergement automatique. (Le système récupère automatiquement ses **informations depuis la base de données** (table étudiants))
2. Suivi du **statut de la demande** :
 - En attente, Acceptée, Refusée
 - Affectation à une résidence, chambre (type de lits)

3. Gestion des demandes de bourse

- Soumission de la demande de bourse
 - Consultation de l'état de la demande
1. Affectation des hébergements
 - Interface pour les agents afin d'assigner :
 - Résidence (R1, R2, ...)
 - Chambre (type lits (1 lit, 2 lits, 3 lits))
 - Affectation automatique ou manuelle en fonction des disponibilités

5. Notifications

- Système de notifications pour :
 - Informer les étudiants des mises à jour sur leurs demandes
 - Notifications internes pour les agents (nouvelles demandes)

6. Tableau de bord (Dashboard)

- Étudiant : suivi de ses demandes, notifications, statut
- Administrateur / agent : vue d'ensemble des demandes, statistiques, filtrage

7. Gestion des résidences et chambres

- Ajout, modification, suppression de :
 - Résidences
 - Chambres (2 lits, 3 lits, chambres spéciales)

8. Export / Import et Statistiques

- Importation et Exportation des étudiants Dans la base de données en PDF/Excel
- Statistiques sur :
 - Nombre de demandes
 - Les demandes traitées (acceptées et refusées)
 - Filtrage par statuts et par Adresse.

F. Modélisation UML

1. Diagramme de classes :

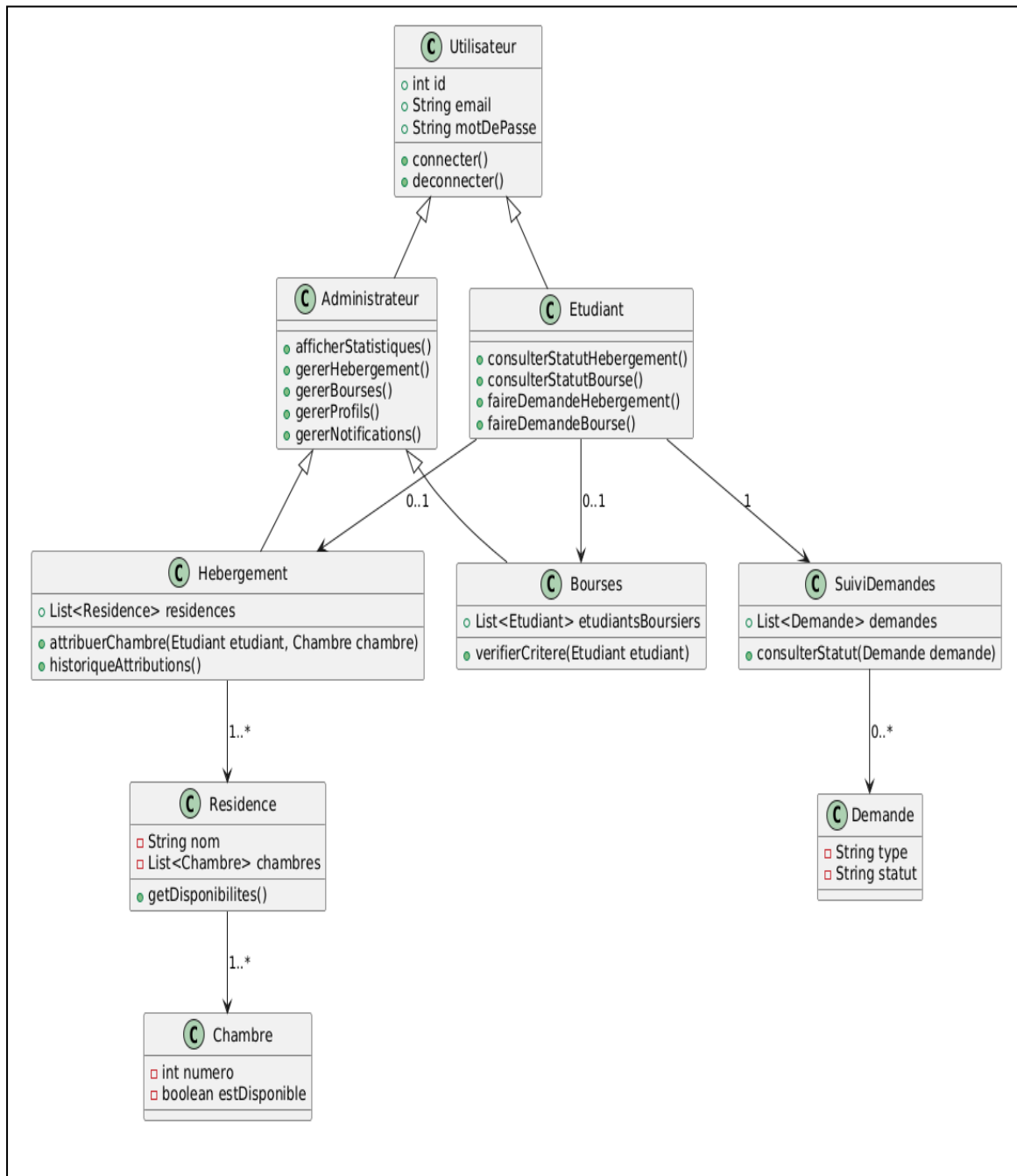


Figure 3 : Diagramme de classes du système de gestion des demandes d'hébergement et de bourse

Explication du diagramme de Classe :

- **Etudiant** : Représente un étudiant avec des informations telles que son ID, son nom, son prénom, sa date de naissance, son année de bac, son matricule et son email.

L'étudiant peut soumettre des demandes d'hébergement et de bourse, et consulter le statut de ses demandes.

- **Hebergement** : Représente une demande d'hébergement. Elle contient des informations comme l'ID de la demande, la date de demande, le statut (acceptée, en attente, refusée), et un indicateur `is_seen` pour savoir si la demande a été vue par un administrateur.
- **Bourse** : Représente une demande de bourse, avec des informations similaires à celles de la demande d'hébergement (montant, statut, `is_seen`).
- **Administrateur** : Représente un administrateur, qui a la capacité de valider les demandes de bourses et d'hébergement, ainsi que d'affecter des chambres aux étudiants.
- **Residence** : Représente une résidence universitaire. Elle contient des informations comme le nom et le type de la résidence (par exemple, "résidence pour filles", "résidence pour garçons").
- **Chambre** : Représente une chambre dans une résidence, avec des informations comme le numéro de la chambre, sa capacité et son type.
- **Notification** : Représente une notification envoyée à l'étudiant pour l'informer de l'état de sa demande. Chaque notification est liée à une demande spécifique.

Relations :

- **L'étudiant** a le droit de déposer une demande pour chaque demande de résidence et de bourse.
- Un **administrateur** peut gérer plusieurs demandes d'hébergement et de bourse, et envoyer des notifications.
- Une **demande d'hébergement** peut être associée à une **chambre** et à une **résidence**.
- Une **demande de bourse** peut envoyer une **notification** à l'étudiant.

2. Diagramme séquence login étudiants

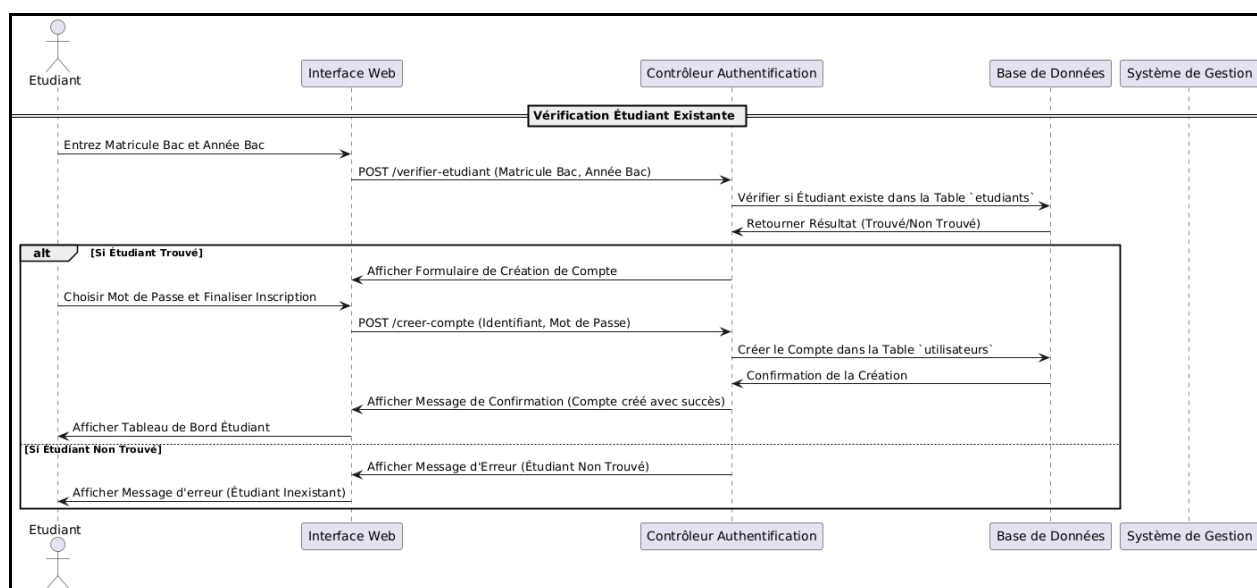


Figure 4 : Diagramme de séquence – Authentification de l'étudiant

Diagramme de séquence ajusté :

1. Vérification de l'étudiant:

- L'**étudiant** entre son **matricule_bac** et son **annee_bac** dans l'interface web.
- L'**interface web** envoie ces informations au **Contrôleur d'authentification** via une requête `POST /verifier-etudiant`.
- Le **Contrôleur d'authentification** vérifie la présence de l'étudiant dans la **base de données** en recherchant les valeurs dans la table `etudiants`.
- Si l'étudiant est trouvé (l'entrée existe), il est autorisé à créer un compte avec un mot de passe de son choix. Le système crée alors un compte utilisateur dans la table `utilisateurs` et affiche une confirmation.
- Si l'étudiant n'est pas trouvé, un message d'erreur est affiché, indiquant que l'étudiant ne peut pas procéder à la création de compte.

Scénarios :

- **Scénario 1** : L'étudiant existe dans la table étudiante et peut créer un compte.
- **Scénario 2** : L'étudiant n'existe pas dans la table étudiants et ne peut pas créer de compte, recevant un message d'erreur.

3. Diagramme séquence login administrateur

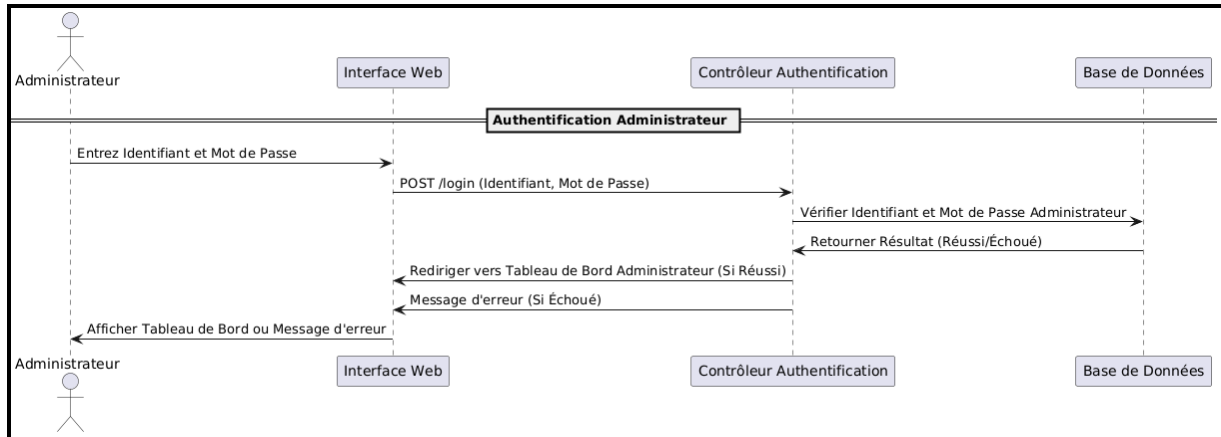


Figure 5 : Diagramme de séquence – Authentification de l'administrateur

Explication du diagramme de séquence :

1. Étapes d'authentification :

- **L'administrateur** entre ses informations d'identification (identifiant et mot de passe) dans l'interface web.
- L'**interface web** envoie ces informations au **Contrôleur d'authentification** via une requête `POST /login`.
- Le contrôleur d'authentification vérifie ces informations dans la **base de données**.
- Si les informations sont correctes (l'authentification réussit), le contrôleur redirige l'administrateur vers son tableau de bord.
- Si les informations sont incorrectes, un message d'erreur est renvoyé et affiché à l'administrateur.

4. Diagramme de séquence pour la demande des étudiants

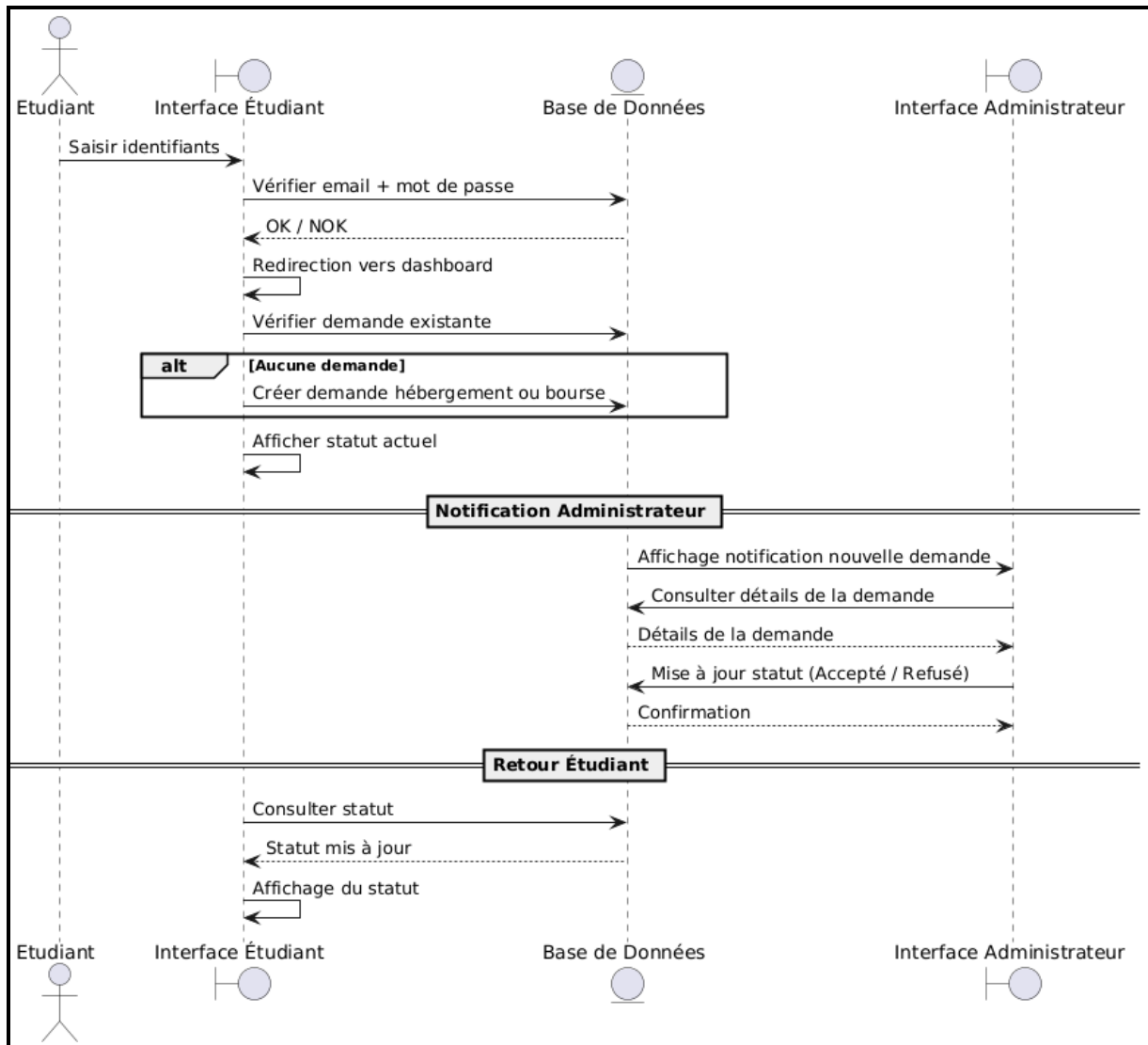


Figure 6 : Diagramme de séquence – Soumission d’une demande par un étudiant

Explication du Diagramme de Séquence Simplifié

Ce diagramme de séquence représente l'interaction entre les différents acteurs du système, notamment l'étudiant, l'interface étudiant, l'interface administrateur, et la base de données. Voici une explication détaillée de chaque étape de ce diagramme :

1. Authentification de l'étudiant

- **Acteur** : Étudiant
- **Interface Étudiant** : L'étudiant saisit ses informations de connexion (identifiants comme l'email et le mot de passe).
- **Étapes** :
 1. **Étudiant** → **Interface Étudiant** : L'étudiant saisit ses identifiants (email et mot de passe).
 2. **Interface Étudiant** → **Base de Données** : L'interface envoie les informations d'identification à la base de données pour vérification.
 3. **Base de Données** → **Interface Étudiant** : La base de données vérifie les identifiants. Si l'authentification est réussie, elle renvoie un **OK**, sinon un **NOK** (dans le cas où les identifiants sont incorrects).
 4. Si l'authentification réussit (OK), l'interface redirige l'étudiant vers le tableau de bord de l'étudiant.

2. Vérification de la demande

- **Acteur** : Étudiant
- **Interface Étudiant** : Après l'authentification réussie, l'étudiant se voit présenter un tableau de bord. Le système vérifie s'il a déjà une demande en cours (pour l'hébergement ou la bourse).
- **Étapes** :
 1. **Interface Étudiant** → **Base de Données** : L'interface demande à la base de données s'il existe déjà une demande d'hébergement ou de bourse pour cet étudiant.
 2. Si **aucune demande** n'est trouvée, l'interface crée une nouvelle demande dans la base de données (envoi de l'information à la base de données pour la création).
 3. **Base de Données** → **Interface Étudiant** : La base de données confirme la création de la demande.
 4. **Interface Étudiant** : Affiche l'état de la demande à l'étudiant (statut de la demande, par exemple "En attente", "Acceptée", ou "Refusée").

3. Notification Administrateur

- **Acteur** : Administrateur
- **Interface Administrateur** : Lorsque l'étudiant soumet une demande, un administrateur doit être notifié et traiter cette demande.
- **Étapes** :
 1. **Base de Données** → **Interface Administrateur** : Dès qu'une nouvelle demande est enregistrée, la base de données envoie une notification à l'interface administrateur.
 2. **Interface Administrateur** → **Base de Données** : L'administrateur clique sur la notification et consulte les détails de la demande dans la base de données.
 3. **Base de Données** → **Interface Administrateur** : La base de données renvoie les détails de la demande à l'interface administrateur.
 4. **Interface Administrateur** → **Base de Données** : L'administrateur met à jour le statut de la demande (par exemple, "Acceptée" ou "Refusée").
 5. **Base de Données** → **Interface Administrateur** : La base de données confirme que le statut a bien été mis à jour.

4. Retour de l'étudiant

- **Acteur** : Étudiant
- **Interface Étudiant** : Une fois que l'administrateur a mis à jour le statut de la demande, l'étudiant peut consulter ce statut.
- **Étapes** :
 1. **Étudiant** → **Interface Étudiant** : L'étudiant actualise son tableau de bord pour vérifier l'état de sa demande.
 2. **Interface Étudiant** → **Base de Données** : L'interface demande à la base de données le statut actuel de la demande de l'étudiant.
 3. **Base de Données** → **Interface Étudiant** : La base de données renvoie le statut mis à jour (par exemple, "Acceptée", "Refusée").
 4. **Interface Étudiant** : Le statut de la demande est affiché à l'étudiant.

Points clés :

- **Acteurs principaux** : L'étudiant et l'administrateur, avec l'interface de chaque côté (Interface Étudiant et Interface Administrateur).

- **Base de données** : Elle joue un rôle central dans la gestion des demandes, la vérification des identifiants et la mise à jour des statuts.
- **Séquences et Interactions** :
 - L'étudiant commence par s'authentifier.
 - Ensuite, il peut créer une demande (hébergement ou bourse).
 - L'administrateur traite la demande et met à jour le statut.
 - Enfin, l'étudiant peut voir l'évolution du statut de sa demande.

Ce diagramme de séquence simplifié décrit les interactions et le processus global du système de gestion d'hébergement et de bourses. Il permet d'illustrer clairement la chaîne d'actions et de responsabilités entre les utilisateurs (étudiants et administrateurs) et la base de données, tout en mettant en évidence le rôle de chaque acteur et la façon dont l'information est traitée au sein du système.

5. Diagramme de séquence de l'administrateur (gestion, Statistique, filtrage des demandes, etc.)

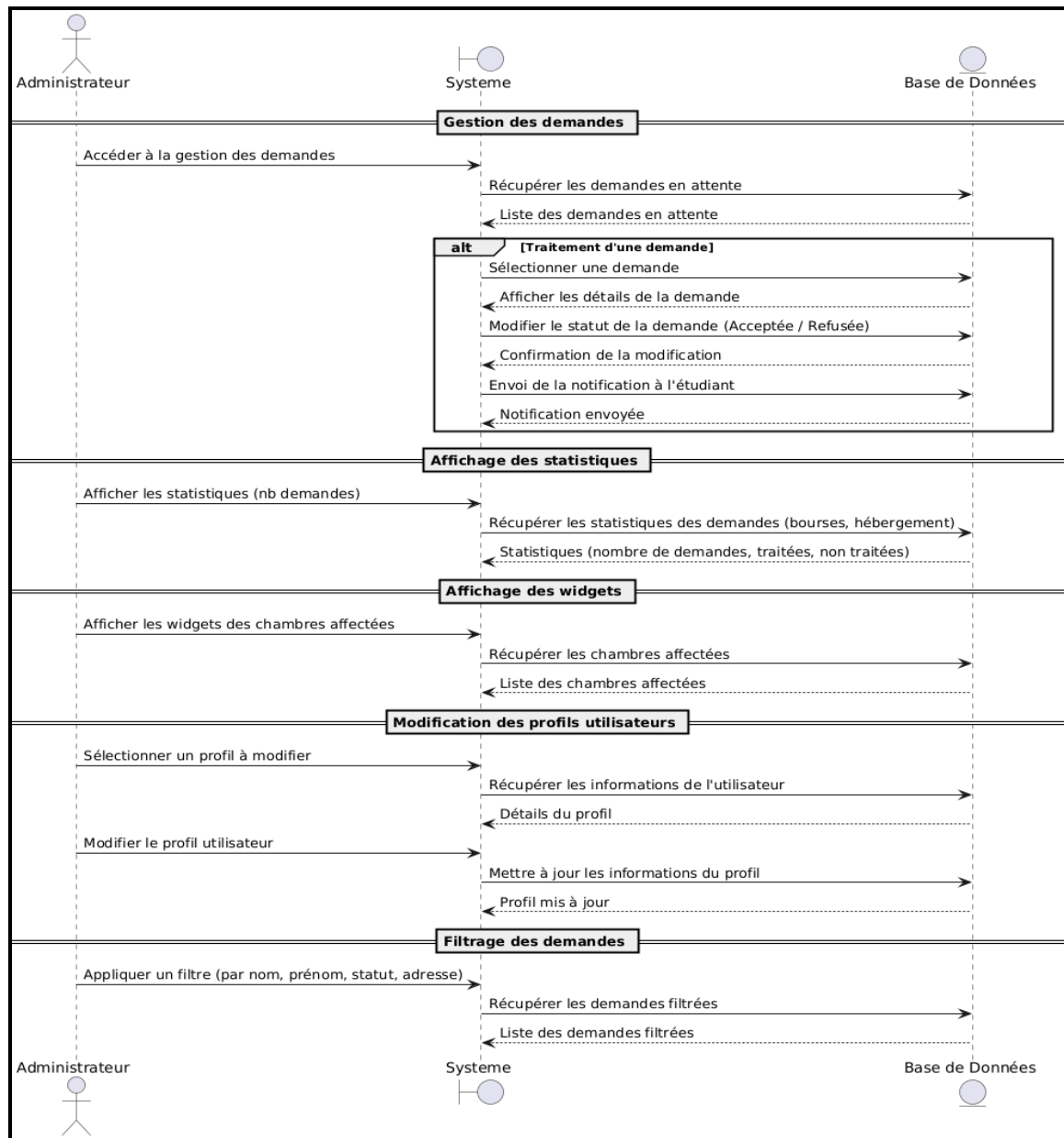


Figure 7 : Diagramme de séquence – Traitement, filtrage et analyse des demandes par l'administrateur

Le diagramme de séquence UML que vous avez partagé montre l'interaction entre l'administrateur, l'interface du système (UIAdmin), et la base de données (DB) pour plusieurs actions dans un système de gestion des demandes. Voici l'explication détaillée des différentes sections de ce diagramme.

1. Gestion des demandes

Cette section décrit le flux permettant à l'administrateur de gérer les demandes (hébergement ou bourses) soumises par les étudiants.

- **Administrateur (Admin)** : L'administrateur commence par accéder à l'interface de gestion des demandes via l'interface `UIAdmin`.
- **Système (UIAdmin)** : L'interface du système demande à la base de données la liste des demandes en attente. Ces demandes sont celles qui n'ont pas encore été traitées.

Base de Données (DB) : La base de données renvoie la liste des demandes en attente à l'interface du système.

Traitement d'une demande :

- **Sélection de la demande** : L'administrateur sélectionne une demande spécifique à traiter.
- **Détails de la demande** : Le système récupère les détails de la demande de la base de données et les affiche à l'administrateur.
- **Modification du statut** : L'administrateur modifie le statut de la demande (par exemple, "Acceptée" ou "Refusée").
- **Confirmation dans la base de données** : La base de données confirme que le statut de la demande a bien été mis à jour.
- **Envoi de notification à l'étudiant** : Après la modification du statut, une notification est envoyée à l'étudiant pour l'informer de l'état de sa demande.
- **Confirmation de la notification** : La base de données confirme l'envoi de la notification.

2. Affichage des statistiques

Cette section permet à l'administrateur de consulter des statistiques sur les demandes.

- **Administrateur (Admin)** : L'administrateur demande à afficher des statistiques, telles que le nombre total de demandes reçues, ainsi que celles qui ont été traitées et celles en attente.
- **Système (UIAdmin)** : L'interface admin interroge la base de données pour récupérer ces statistiques.
- **Base de Données (DB)** : La base de données renvoie les statistiques à l'interface administrateur, qui les affiche à l'administrateur.

3. Affichage des widgets

Cette section affiche des widgets qui permettent à l'administrateur de visualiser des informations supplémentaires comme les chambres affectées.

- **Administrateur (Admin)** : L'administrateur demande à voir les widgets concernant les chambres affectées.
- **Système (UIAdmin)** : L'interface admin demande à la base de données de récupérer les chambres affectées.
- **Base de Données (DB)** : La base de données renvoie la liste des chambres affectées, qui est ensuite affichée sous forme de widgets.

4. Modification des profils utilisateurs

Cette section permet à l'administrateur de modifier le profil d'un utilisateur, qu'il soit étudiant ou autre.

- **Administrateur (Admin)** : L'administrateur sélectionne un profil utilisateur qu'il souhaite modifier.
- **Système (UIAdmin)** : L'interface admin demande à la base de données les informations relatives au profil de l'utilisateur sélectionné.
- **Base de Données (DB)** : La base de données renvoie les détails du profil à l'interface administrateur.
- **Modification du profil** : L'administrateur modifie les informations du profil utilisateur dans l'interface admin.
- **Mise à jour dans la base de données** : L'interface admin envoie les informations modifiées à la base de données pour mettre à jour le profil.
- **Confirmation de la mise à jour** : La base de données confirme que le profil a bien été mis à jour.

5. Filtrage des demandes

L'administrateur peut filtrer les demandes selon différents critères (nom, prénom, statut, adresse).

- **Administrateur (Admin)** : L'administrateur applique un filtre sur les demandes pour les rechercher en fonction de critères spécifiques (nom, prénom, statut de la demande, adresse, etc.).
- **Système (UIAdmin)** : L'interface admin demande à la base de données de récupérer les demandes filtrées en fonction des critères.
- **Base de Données (DB)** : La base de données renvoie la liste des demandes filtrées à l'interface administrateur.

Ce diagramme montre un système de gestion complet dans lequel l'administrateur peut :

1. Accéder aux demandes en attente et les traiter (modifier leur statut et envoyer des notifications à l'étudiant),
2. Afficher des statistiques relatives aux demandes,
3. Visualiser des widgets comme les chambres affectées,
4. Modifier les profils des utilisateurs (étudiants ou autres administrateurs),
5. Appliquer des filtres pour rechercher des demandes spécifiques.

Chaque action effectuée dans l'interface admin interagit avec la base de données pour récupérer ou mettre à jour des informations essentielles au bon fonctionnement du système.

6. Diagramme de séquence – Interaction étudiant/administrateur dans la gestion des demandes.

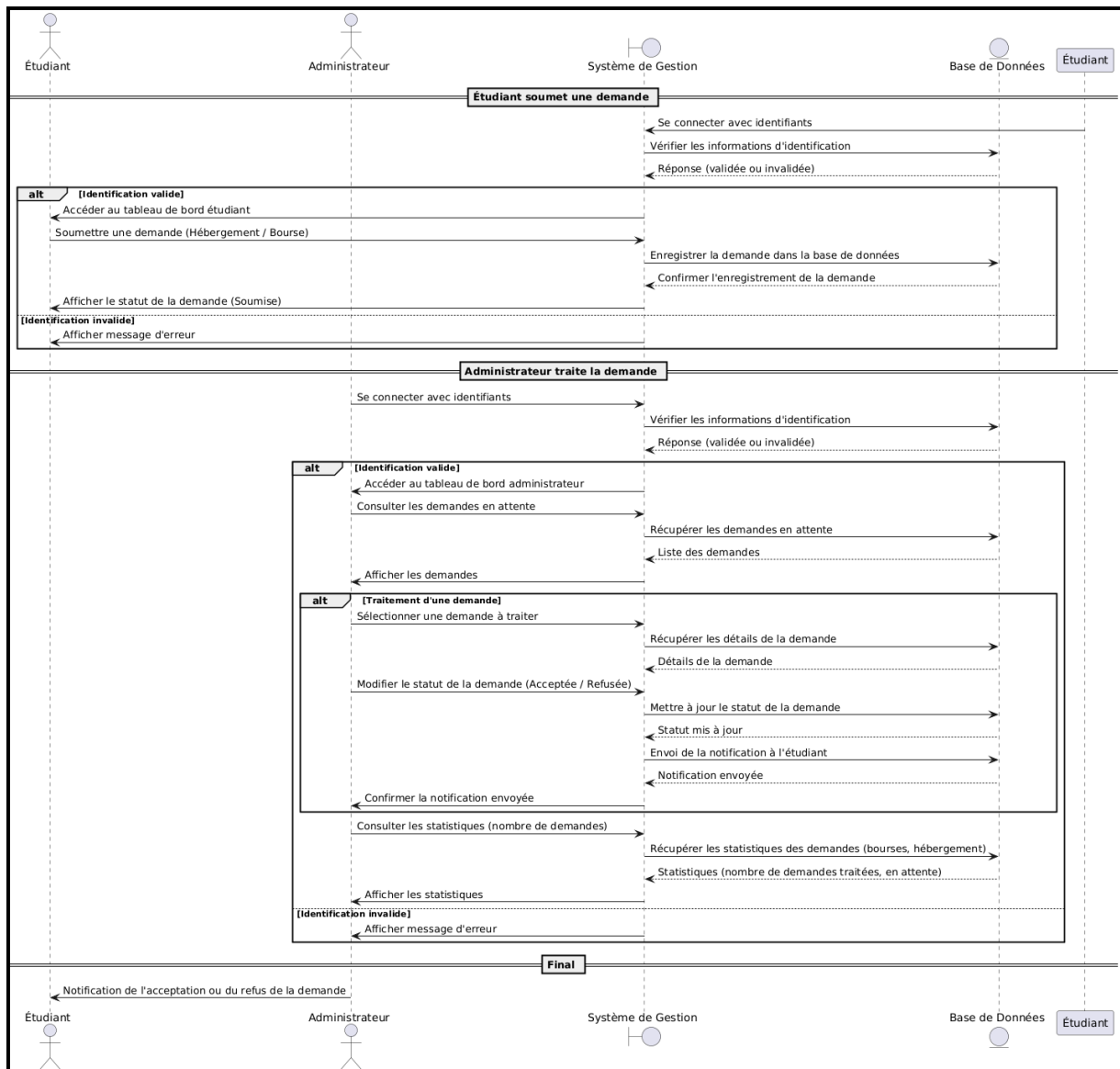


Figure 8 : Diagramme de séquence – Interaction entre l'étudiant et l'administrateur

Explication du diagramme :

1. Étudiant soumet une demande :

- L'étudiant se connecte en entrant ses identifiants.
- Le système vérifie ces identifiants dans la base de données. Si l'identification est valide, l'étudiant accède à son tableau de bord.
- Ensuite, l'étudiant soumet une demande d'hébergement ou de bourse.
- Le système enregistre la demande dans la base de données et affiche un message de confirmation de soumission.

- Si l'identification échoue, un message d'erreur est affiché à l'étudiant.
- 2. Administrateur traite la demande :**
- L'administrateur se connecte au système.
 - Après validation, l'administrateur consulte les demandes en attente dans son tableau de bord.
 - Le système récupère les demandes en attente depuis la base de données et les affiche à l'administrateur.
 - L'administrateur sélectionne une demande pour la traiter (acceptation ou refus).
 - Le système met à jour le statut de la demande dans la base de données et envoie une notification à l'étudiant concerné.
 - L'administrateur peut aussi consulter les statistiques des demandes traitées et en attente dans le système.
- 3. Finalisation :**
- Une fois la demande traitée par l'administrateur, une notification est envoyée à l'étudiant pour l'informer du résultat (acceptation ou refus).

Ce diagramme illustre le flux des opérations de soumission, de traitement et de gestion des demandes de bourses et d'hébergement, impliquant à la fois l'étudiant (qui soumet la demande) et l'administrateur (qui traite et gère ces demandes). Le système centralisé de gestion interagit avec la base de données pour stocker et récupérer les informations nécessaires tout au long du processus.

Diagramme de cas d'utilisation- Interaction entre l'étudiant, Administrateur et le système.

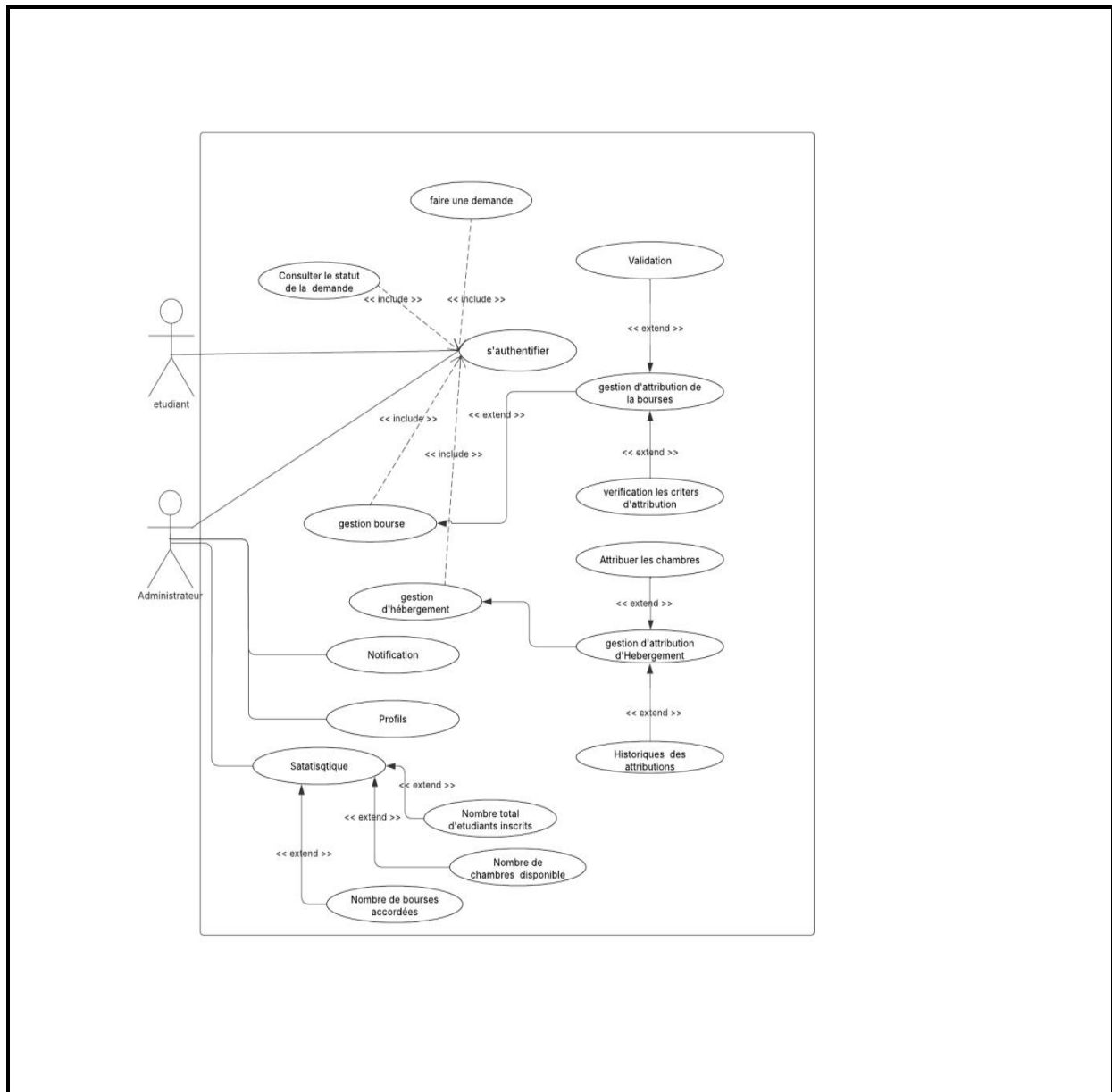


Figure 9 : Diagramme des cas d'utilisation illustrant les interactions entre

Explication du diagramme de cas d'utilisation

Acteurs :

1. **Étudiant** : L'utilisateur qui soumet une demande d'hébergement ou de bourse et consulte le statut de sa demande.
2. **Administrateur** : L'utilisateur qui gère les demandes des étudiants, affiche les statistiques, modifie les profils des utilisateurs, et gère les chambres affectées.

Cas d'utilisation pour l'étudiant :

1. **Se connecter** : L'étudiant se connecte au système en utilisant ses identifiants (matricule_bac et année_bac).
2. **Soumettre une demande** : L'étudiant soumet une demande d'hébergement ou de bourse.
3. **Consulter le statut de la demande** : L'étudiant peut consulter l'état actuel de sa demande (en attente, acceptée ou refusée).
4. **Recevoir une notification** : L'étudiant reçoit une notification après que sa demande a été traitée par l'administrateur.

A. Cas d'utilisation pour l'administrateur :

1. **Gérer les demandes (Valider/refuser)** : L'administrateur traite les demandes des étudiants en validant ou en rejetant celles-ci.
2. **Afficher les statistiques** : L'administrateur consulte les statistiques liées aux demandes d'hébergement et de bourses (nombre de demandes, nombre acceptées, refusées, etc.).
3. **Modifier les profils utilisateurs** : L'administrateur peut modifier les informations des étudiants ou d'autres utilisateurs dans le système.
4. **Filtrer les demandes** : L'administrateur peut appliquer des filtres (par nom, prénom, statut, adresse) pour trier les demandes d'hébergement et de bourses.
5. **Afficher les chambres affectées** : L'administrateur peut consulter les chambres affectées aux étudiants dans les résidences universitaires.

Ce diagramme présente l'interaction des **acteurs** (Étudiant et Administrateur) avec le système. L'étudiant soumet ses demandes et suit leur statut, tandis que l'administrateur gère les demandes, consulte des statistiques, et modifie les informations des utilisateurs. Les cas d'utilisation reflètent les principales fonctionnalités du système.

Chapitre 3 : Implémentation et Réalisation de L'Application

1. Introduction

Après les études théoriques et préliminaires menées précédemment nous allons entamer le volet pratique de notre projet. Nous définirons les étapes de réalisation, et la mise en œuvre de notre système **de Gestion d'Hébergement et de Bourses pour la Direction des Œuvres Universitaires**. Nous commencerons par une brève présentation des outils de développement utilisés. Et nous terminerons par l'implémentation de notre système.

2. Langage utilisé :



est un framework PHP [7]. créé par Taylor Otwell en 2011, qui permet de simplifier le développement d'applications web tout en gardant le code bien organisé. Depuis sa création, Laravel est devenu l'un des frameworks PHP les plus populaires et les plus utilisés, avec une communauté de développeurs en constante croissance.

Laravel est basé sur le modèle MVC (Modèle-Vue-Contrôleur), qui permet de séparer le code en trois couches distinctes pour une meilleure organisation et une maintenance plus facile. Le modèle s'occupe de la logique de l'application et de l'interaction avec la base de données, la vue est responsable de l'affichage de l'interface utilisateur, et le contrôleur agit comme une passerelle entre le modèle et la vue, en traitant les requêtes entrantes et en envoyant les données à la vue.

Laravel fournit également de nombreux composants prêts à l'emploi, tels que l'authentification, la validation, la gestion des fichiers, la mise en cache, la gestion des sessions, la gestion des tâches planifiées et bien d'autres.

2.1. Langue de l'interface utilisateur (UI)

- **Français** : C'est la langue principale utilisée dans l'interface.
- Tous les menus, boutons, libellés, messages, statuts, notifications et pages sont en **français**, pour être compréhensibles par les étudiants et les administrateurs dans le contexte universitaire algérien.

2.2. Langages de programmation et technologies utilisées

L'application utilise plusieurs langages, selon les couches (front-end, back-end, base de données, etc.) :

2.2.1. Langage côté serveur (back-end)

- **PHP (Laravel)** :
Le cœur de l'application est développé en **Laravel**, un framework PHP moderne.

- Laravel permet une architecture MVC (Modèle-Vue-Contrôleur).
- Il gère les routes, les contrôleurs, les modèles, les migrations, l'authentification, les notifications, etc.
- Exemple : DemandeController, EtuController, routes/web.php.

2.2.2. Langage côté client (front-end)

- **HTML5 / CSS3 :**

Pour la structure et le style des pages web.

- **Bootstrap :**

Utilisé pour le design responsive (mobile/tablette/PC), avec des composants UI modernes.

- **JavaScript :**

Pour les interactions dynamiques côté client (modals, filtres, notifications en temps réel...).

- **Livewire (Laravel) :**

Un outil qui combine le PHP et JavaScript sans écrire du JavaScript manuellement, pour des composants dynamiques (formulaires interactifs, filtres...).



- **PHP** est un langage de script qui s'exécute côté serveur [8], le code PHP étant inclus dans une page HTML classique. Il peut donc être comparé à d'autres langages de script qui fonctionnent sur le même principe : ASP (Active Server Pages), JSP (Java Server Pages) ou PL/SQL Server Pages (PSP).

À la différence d'un langage comme le JavaScript, où le code est exécuté côté client (dans le navigateur), le code PHP est exécuté côté serveur. Le résultat de cette exécution est intégré dans la page HTML qui est envoyée au navigateur. Ce dernier n'a aucune connaissance de l'existence du traitement qui s'est déroulé sur le serveur.

Cette technique permet de réaliser des pages web dynamiques dont le contenu peut être complètement ou partiellement généré au moment de l'appel de la page, grâce à des informations récupérées dans un formulaire ou extraites d'une base de données.



dans le contexte de **PHP** est un **système de gestion de base de données relationnelle (SGBDR) [9]** utilisé pour stocker, gérer et interroger les données de manière

structurée. PHP, en tant que langage de programmation côté serveur, interagit avec MySQL pour effectuer des opérations telles que l'enregistrement, la lecture, la mise à jour et la suppression de données dans une base de données.

MySQL est un système de base de données relationnelle que PHP utilise pour stocker et manipuler les données dynamiquement dans une application web.

PHP utilise MySQL :

Il existe deux grandes façons de connecter PHP à MySQL :

1. Avec l'extension MySQLi (MySQL Improved)

```
$conn = new mysqli("localhost", "root", "motdepasse", "nom_de_base");  
if ($conn->connect_error) {  
die("Connexion échouée : " . $conn->connect_error);  
}
```

echo "Connexion réussie !";

2. Avec PDO (PHP Data Objects) :

```
try {  
    $conn = new PDO("mysql:host=localhost;dbname=nom_de_base", "root",  
    "motdepasse");  
  
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);  
  
echo "Connexion réussie !";  
  
} catch (PDOException $e) {  
  
echo "Erreur : " . $e->getMessage();  
  
}
```

d'utilisation typique (avec MySQLi) :

```
$sql = "SELECT * FROM etudiants";  
  
$result = $conn->query($sql);  
  
while($row = $result->fetch_assoc()) {  
  
echo "Nom: " . $row["nom"] . " - Prénom: " . $row["prenom"] . "<br>";  
  
}
```

MySQL avec PHP ?

- Gratuit et open-source
- Facile à utiliser avec PHP
- Bien supporté dans les hébergements web
- Permet la construction d'applications web dynamiques (comme la gestion d'hébergement et de bourses)

3. Langages de documentation et modélisation



En génie logiciel et développement d'applications, **PlantUML [10]** est un langage de **documentation textuelle** utilisé pour **modéliser visuellement** les systèmes. Il permet de générer automatiquement des **diagrammes UML** à partir d'un texte simple et lisible.

3.1.Définition :

PlantUML est un **langage de script de modélisation UML** qui permet de créer des diagrammes comme les cas d'utilisation, les séquences, les classes, les activités, etc., à partir de **fichiers texte**.

3.2. Langage utilisé par PlantUML :

PlantUML a son **propre langage déclaratif**, inspiré de la syntaxe naturelle, souvent appelé **PlantUML script**. Ce n'est **pas un langage de programmation**, mais un **langage de description**.

3.3. Types de diagrammes UML supportés par PlantUML :

- ❖ 1. Diagramme de cas d'utilisation (@startuml ... usecase)
- ❖ 2. Diagramme de classe (class, interface, extends, implements)
- ❖ 3. Diagramme de séquence (actor, participant, ->)
- ❖ 4. Diagramme d'activité (start, if, else, endif)
- ❖ 5. Diagramme d'états ([*], state, -->)
- ❖ 6. Diagramme de composants (component, interface)
- ❖ 7. Diagramme de déploiement
- ❖ 8. Diagrammes ER (Entité-Relation)

G. Interfaces principales

1. Cote Administrateur :

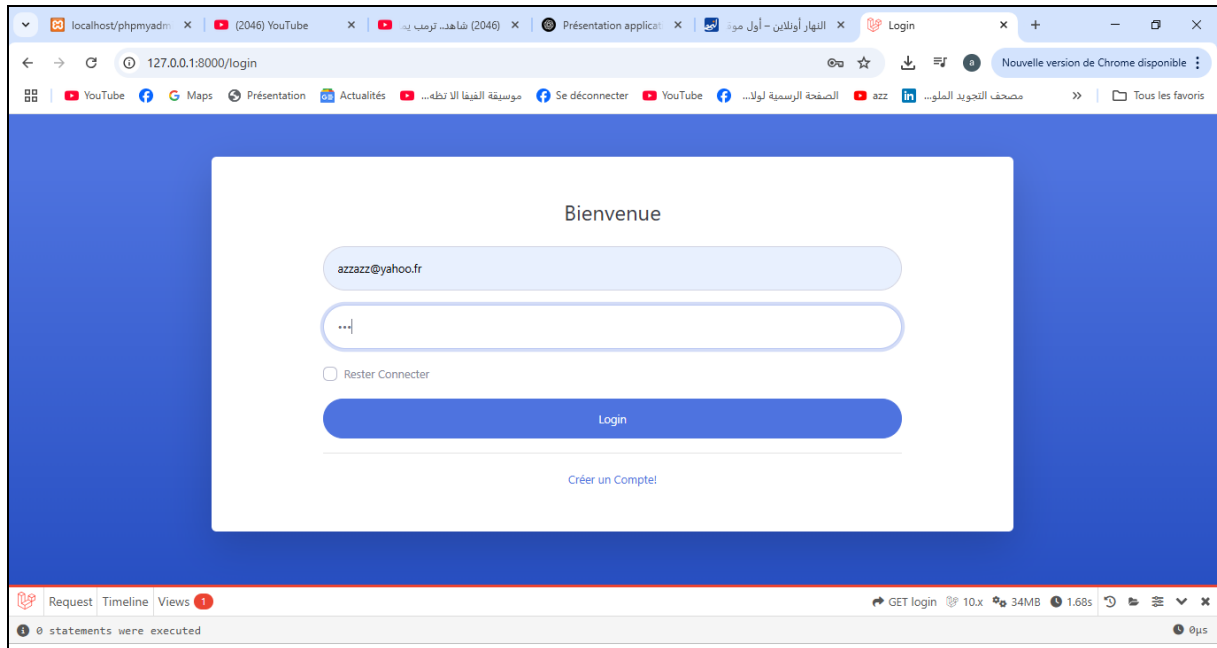


Figure 10 : Interface d'authentification Admin

3.4. Tableaux de Bord Administrateur :

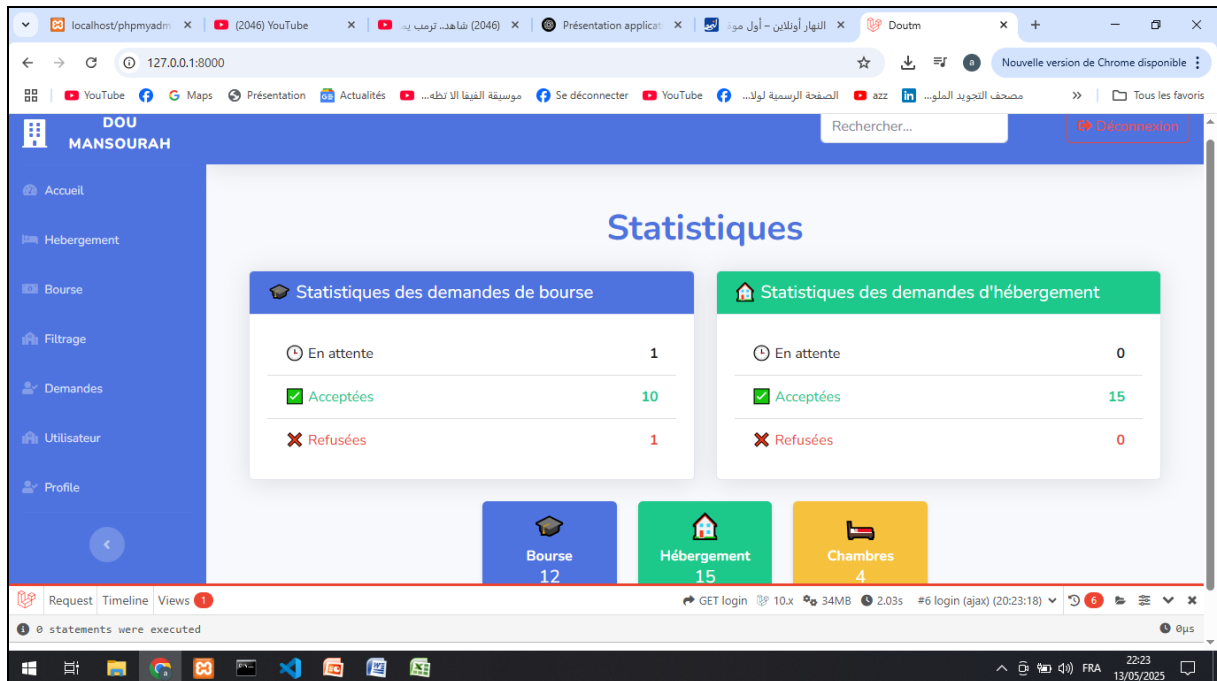


Figure 11 : Interface Statistique et widget

2. Importation des fichiers (listes des étudiants) :

The screenshot shows the 'Liste des étudiants' (Students List) interface. On the left is a sidebar with navigation links: Accueil, Hébergement, Bourse, Filtrage, Demandes, Utilisateur, and Profile. The main content area has a title 'Liste des étudiants' and a button 'Ajouter un étudiant'. Below this is a section for file operations with a dropdown 'Choisir un fichier Excel' (currently showing 'Aucun fichier choisi') and buttons 'Importer' and 'Exporter'. A table of students is displayed below:

#	Matricule	Année	NomA	PrénomA	NomF	PrénomF	Filière	Cycle	Action
1	37074510	2023	shaoy	romysaaa	CHAQUI	ROMAISSA	CHIMIE	L	Détails Modifier Supprimer
2	37015791	2019	blaaydy	hraaa	BELAIDI	HIRAA	CHIMIE ANALYTIQUE	M	Détails Modifier Supprimer

At the bottom, there is a status bar showing '0 statements were executed' and a log of GET login requests.

Figure 12 : Interface Ajouter et importation des étudiants

3. Filtrage des étudiants par Cycle :

The screenshot shows the 'DOU MANSOURAH' interface. The sidebar is the same as in Figure 12. The main content area has a search bar and a 'Cycle Type' dropdown menu set to 'All'. Below this is a table of students:

NOM	PRÉNOM	DATE NAISSANCE	CYCLE	FILIERE	NIVEAU	CRÉER LE	Action
ZERROUKI	FATIMA ZOHRA	2002-07-13	Master	chimie	M1	2025-03-19 12:54:44	Delete
TOUATI	WAFAE	2001-09-08	Master	chimie	M1	2025-03-19 12:54:45	Delete
ROSTANE	CHAIMAA	2002-01-09	Master	chimie	M1	2025-03-19 12:54:45	Delete

The status bar at the bottom shows '32 statements were executed' and a log of GET structure requests.

Figure 13 : Interface Filtrage des étudiants

4. Liste des demandes pour traiter :

Demandes de Bourse

ID	Matricule Bac	Année Bac	Nom	Prénom	Date de Naissance	Adresse	Motivation	Status	Étudiant	Actions
43	37074510	2023	CHAOUI	ROMAÏSSA	2004-09-10 00:00:00	Quartier Boujlida Tlemcen	123456	Changer le statut Acce	N/A	
44	37027237	2020	KEDJAR	IMANE	2001-05-15 00:00:00	شارع عزوز محمد رقم 10 تيزومة	1234	Changer le statut	N/A	

Demandes d'Hébergement

ID	Matricule BAC	Année BAC	Nom	Prénom	Date de Naissance	Adresse	Status	Étudiant	Actions
59	37018738	2018	BELLIFA	CHAÏMA	1999-11-14 00:00:00	شارع العربي التبيسي رقم 33	Changer le statut Acce	N/A	
60	37018738	2018	BELLIFA	CHAÏMA	1999-11-14 00:00:00	شارع العربي التبيسي رقم 33	Changer le statut Acce	N/A	
61	37018738	2018	BELLIFA	CHAÏMA	1999-11-14 00:00:00	شارع العربي التبيسي رقم 33	Changer le statut Acce	N/A	
62	37018738	2018	BELLIFA	CHAÏMA	1999-11-14 00:00:00	شارع العربي التبيسي رقم 33	Changer le statut	N/A	

Figure 14 : Interface liste des étudiants faire les Demandes

5. Coté Utilisateur (étudiants) :

The screenshot shows a web browser window with the URL `127.0.0.1:8000/etudiant/register`. The page has a blue background and a white central form titled "Créer un compte étudiant". The form contains five input fields: "Matricule BAC", "Année BAC", "Email", "Mot de passe", and "Confirmer le mot de passe". Below these fields is a blue button labeled "Créer un compte". The browser's developer tools are open at the bottom, showing the "Request" tab with a GET request to `etudiant/register`.

Figure 15 : Interface Création Compte Etudiants

6. Coté Utilisateur (login étudiants) :

The screenshot shows a web browser window with the URL `127.0.0.1:8000/etudiant/login`. The page has a blue background and a white central form titled "Connexion Étudiant". The form contains two input fields: "Email" and "Mot de passe". Below these fields is a blue button labeled "Se connecter". At the bottom of the form, there is a link that says "Créer un compte étudiant". The browser's developer tools are open at the bottom, showing the "Request" tab with a GET request to `etudiant/login`.

Figure 16 : Interface d'authentification Etudiants

7. Tableaux de Bord Utilisateur (étudiants) :

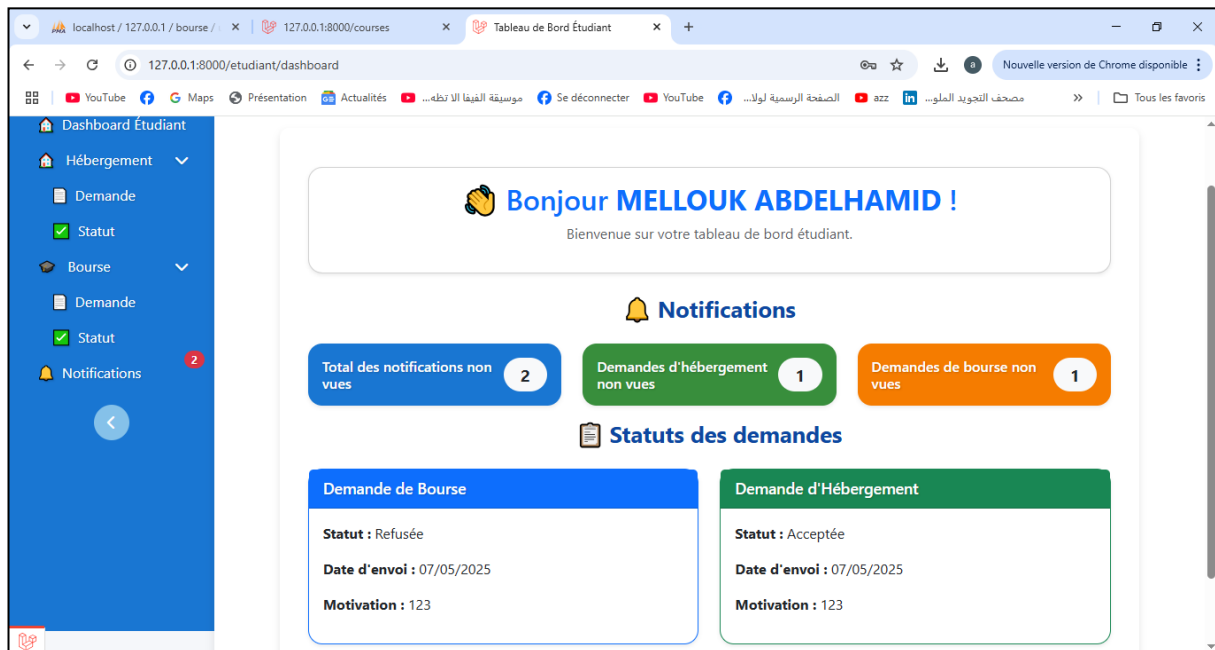
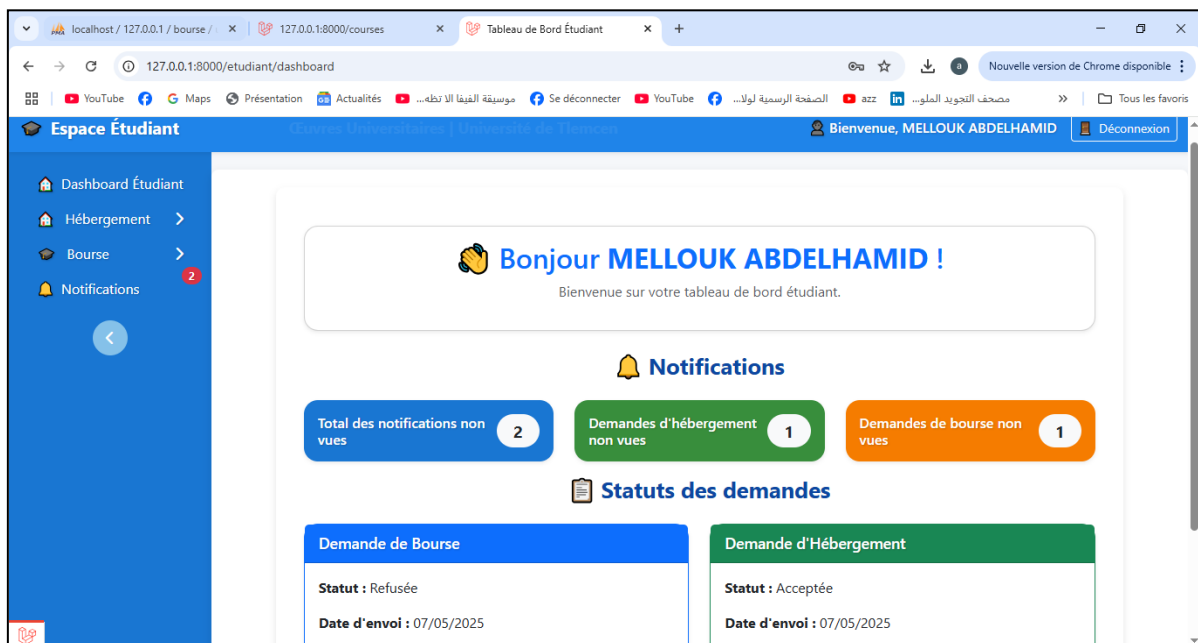


Figure 17 : Interface du menu du tableau de bord étudiant



8. Interface Statuts et Notification d'étudiants :

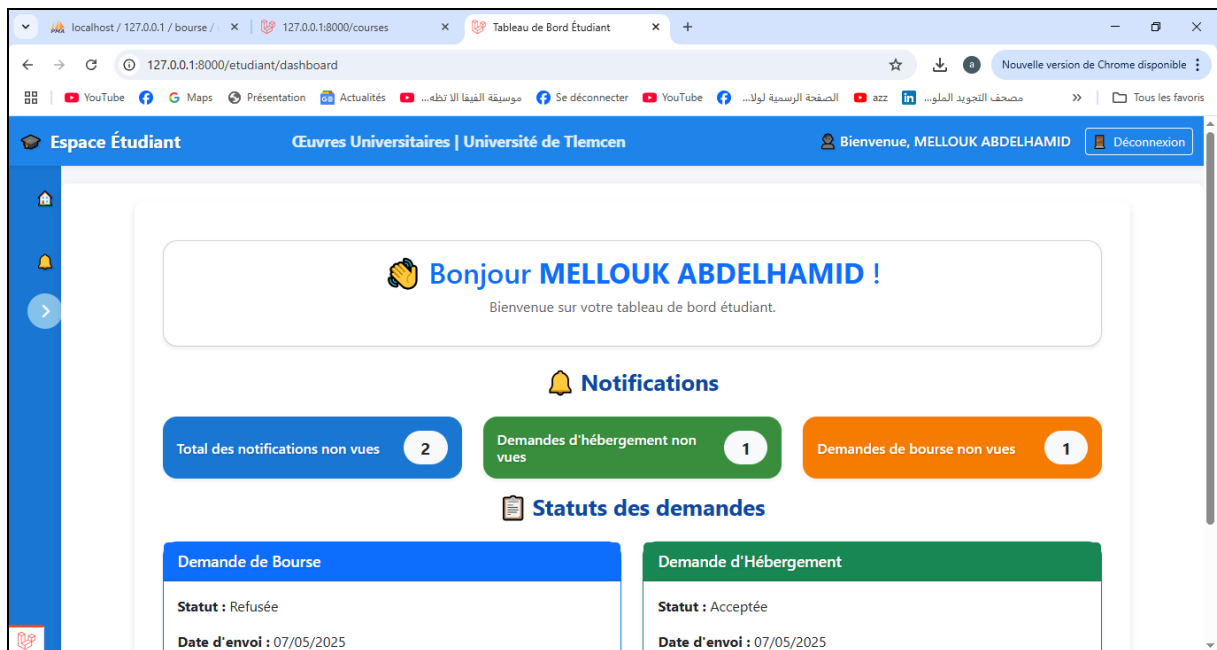


Figure 18 : Interface Statuts et Notifications Etudiants

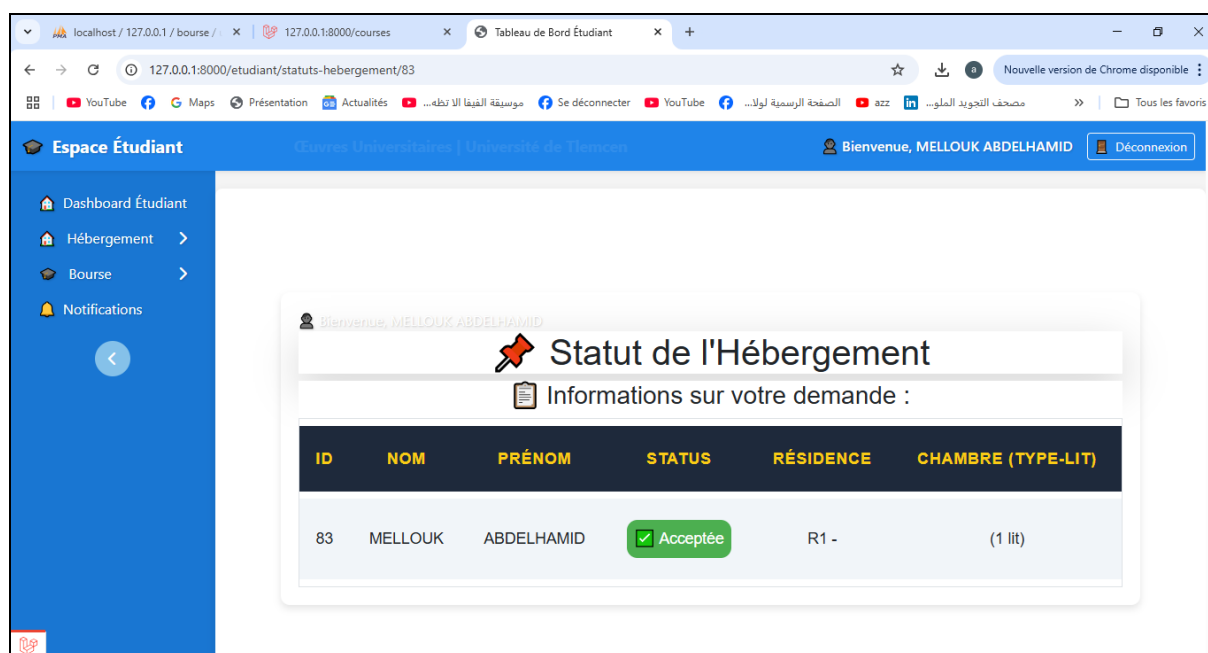
9. Formulaire on ligne des demandes :

The screenshot shows a web browser window with the URL `127.0.0.1:8000/demande-hebergement`. The page is titled 'Demande hebergement'. The form contains the following fields:

- Matricule Bac: 37017002
- Année Bac: 2020
- Nom: MELLOUK
- Prénom: ABDELHAMID
- Date de Naissance: 2002-11-01
- Adresse: حي سيدي بوركاب

Figure 19 : Interface du formulaire automatique pour la demande étudiante.

10. Statuts d'hébergements d'un étudiant pour faire la demande :



The screenshot displays a web application interface for a student's accommodation status. The browser address bar shows the URL `127.0.0.1:8000/etudiant/statuts-hebergement/83`. The page header includes the 'Espace Étudiant' logo, the text 'Cours Universitaires | Université de Tlemcen', and a welcome message 'Bienvenue, MELLOUK ABDELHAMID' with a 'Déconnexion' button.

The main content area features a sidebar on the left with links to 'Dashboard Étudiant', 'Hébergement', 'Bourse', and 'Notifications'. The central panel is titled 'Statut de l'Hébergement' and contains a section 'Informations sur votre demande :'. Below this is a table with the following data:

ID	NOM	PRÉNOM	STATUS	RÉSIDENCE	CHAMBRE (TYPE-LIT)
83	MELLOUK	ABDELHAMID	✓ Acceptée	R1 -	(1 lit)

Figure 20 : Interface des statuts pour la demande étudiante

Conclusion et perspectives

A. Conclusion générale

Le développement de cette application de gestion de l'hébergement et de la bourse des œuvres universitaires s'inscrit dans une démarche de modernisation et de digitalisation des services sociaux destinés aux étudiants. Il répond à un besoin réel d'optimisation des processus, d'amélioration de la communication et de centralisation des informations.

Ce projet nous a permis d'acquérir des compétences techniques solides dans les domaines de l'analyse, de la conception, du développement web, et de la gestion de base de données. Il a également renforcé notre capacité à répondre à un cahier des charges, à collaborer efficacement et à fournir une solution opérationnelle à un problème concret.

Grâce à cette plateforme, les étudiants bénéficient désormais d'un accès rapide et simplifié à leurs services universitaires, tandis que l'administration dispose d'un outil performant pour la gestion quotidienne de ses activités.

Enfin, ce projet offre de nombreuses perspectives d'évolution, telles que l'intégration d'un module de paiement en ligne, la génération de certificats de bourse, ou encore le développement d'une application mobile pour une accessibilité optimale.

La gestion centralisée des demandes d'hébergement et de bourses représente un enjeu stratégique pour les œuvres universitaires. Le système que nous avons conçu et implémenté permet :

- Aux **étudiants** d'effectuer leurs demandes en ligne, de consulter le **statut de leurs dossiers**, et de recevoir des **notifications en temps réel**.
- Aux **administrateurs** de gérer efficacement les dossiers, de filtrer les demandes, d'**affecter des chambres** et de prendre des décisions grâce à des interfaces conviviales et adaptées.

L'utilisation du framework **Laravel** a permis de structurer l'application selon une architecture robuste, modulaire et sécurisée, tout en assurant une expérience utilisateur fluide aussi bien du côté étudiant que du côté administratif.

Difficultés rencontrées

Au cours de la réalisation du projet, nous avons rencontré certaines difficultés, notamment :

- La **mise en place d'une authentification personnalisée** pour les étudiants, qui nécessitait de sortir du schéma standard de Laravel.
- La gestion dynamique des affectations de chambres avec **modal Bootstrap** et **AJAX**.
- La séparation claire des rôles et des droits d'accès entre les utilisateurs du système.

Ces défis ont été surmontés grâce à une bonne documentation, des tests rigoureux et une organisation méthodique du code.

B. Perspectives d'amélioration

Plusieurs pistes d'évolution peuvent être envisagées pour enrichir et renforcer le système :

- **Ajout de rôles et permissions avancés** (chef de service, agents, etc.)
- **Développement d'une application mobile** pour étudiants
- **Intégration de statistiques interactives** et tableaux de bord dynamiques (ex : avec Chart.js)
- **Export PDF/Excel des demandes**
- **Ajout d'un calendrier d'attribution et de relogement**
- **Notifications par e-mail ou SMS** pour les changements de statut
- **Gestion de la disponibilité en temps réel des chambres et historique des affectations.**

C. Bilan personnel

Ce projet m'a permis de renforcer mes compétences en :

- **Développement web avec Laravel**
- **Modélisation de base de données relationnelle**
- **Conception d'interfaces utilisateurs centrées sur l'expérience**
- **Travail en mode projet** et gestion des livrables techniques

Il a aussi été l'occasion de travailler sur un **besoin concret et utile** au sein d'un environnement universitaire, ce qui renforce la pertinence de l'application dans un contexte réel.

Références et bibliographie

[1] : Google Maps. (2025). *Cité universitaire Malika Kaïd – Mansourah, Tlemcen*. Consulté le 2 mai 2025, à partir de <https://www.google.com/maps>

[2] :DOUTM. (2025). *Historique des œuvres universitaires à Tlemcen*. Consulté le 3 juin 2025, à partir de <http://www.doutlemcen.dz>

[3] : DOUTM. (2025). *Service de l'hébergement : modalités d'inscription, réadmission, transfert et abandon*. Consulté le 1er juin 2025, à partir de <http://www.doutlemcen.dz/services/hebergement>

[4] :Ministère de l'Enseignement Supérieur et de la Recherche Scientifique (MESRS). (2021). *Plateforme PROGRES – Bourse universitaire : critères, montants et procédures*. Office National des Œuvres Universitaires. Consulté le 5 avril 2025, à partir de <https://progres.mesrs.dz/webonou>

[5] :Direction des Œuvres Universitaires – Constantine Khroub. (2021). Bourses universitaires – critères d'éligibilité, montants et modalités. Consulté le 20 juin 2025, sur https://www.dou-constantine-khroub.com/Bourse_univ.php

[6] :Centre National des Œuvres Universitaires et Scolaires (CNOUS). (2024). *Portail national des services étudiants - CROUS*. messervices.etudiant.gouv.fr. Consulté le 3 juin 2025, à partir de <https://www.messervices.etudiant.gouv.fr>

[7] :Laravel. (2025). *Documentation officielle du Framework Laravel*. Consulté sur : <https://laravel.com/docs>

[8] :PHP Group. (2025). *Documentation officielle de PHP*. Disponible sur : <https://www.php.net/manual/fr/>

[9] :PHP Group. (2025). *MySQLi et PDO – Documentation PHP*. Disponible sur :

- <https://www.php.net/manual/fr/book.mysql.php>
- <https://www.php.net/manual/fr/book.pdo.php>

[10] Arlow, J., & Neustadt, I. (2005). *UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design*. Addison-Wesley.

PlantUML. *PlantUML Language Reference Guide*. Documentation officielle. Disponible sur : <https://plantuml.com/fr/guide> (consulté en mai 2025).

Résumé

Ce projet consiste à développer une application web pour la gestion des demandes d'hébergement et de bourses au sein de la Direction des Œuvres Universitaires. L'application offre deux interfaces : une pour les étudiants (soumission et suivi des demandes) et une pour les administrateurs (gestion, filtrage et affectation). Elle est réalisée avec Laravel, MySQL, et utilise Bootstrap/Livewire pour une interface moderne et réactive.

Mots-clés : Laravel, MySQL, Hébergement, Bourse, Étudiant, Œuvres universitaires, Application web, Digitalisation.

Abstract

This project aims to develop a web application for managing student housing and scholarship requests within the Directorate of University Services. The application offers two interfaces: one for students (submission and tracking of requests), and another for administrators (management, filtering, and allocation). It is built using Laravel, MySQL, and uses Bootstrap/Livewire for a modern and responsive interface.

Keywords: Laravel, MySQL, Housing, Scholarship, Student, University services, Web application, Digitalization.

الملخص:

يهدف هذا المشروع إلى تطوير تطبيق ويب لإدارة طلبات الإيواء والمنح الجامعية على مستوى مديرية الخدمات الجامعية. يوفر التطبيق واجهتين: واحدة للطلبة (لإرسال ومتابعة الطلبات) وأخرى للإدارة (للاشراف، التصفية، والتوزيع). تم تطوير النظام باستخدام Laravel وMySQL، ويعتمد على Bootstrap/Livewire لتوفير واجهة عصرية وتفاعلية.

الكلمات المفتاحية: لارفيل، ماي إس كيو إل، الإيواء، المنحة، الطالب، الخدمات الجامعية، تطبيق ويب، الرقمنة.