



# Ministère de l'Enseignement Supérieur de la Recherche et de l'Innovation



UFR Sciences et Technologies

Département Informatique

## Mémoire de fin d'études

Pour l'obtention du diplôme de master

Mention : Informatique | Spécialité : Génie logiciel

### Sujet

**Système d'information pour la gestion des mémoires de  
master 2 et des alumnis : Cas du département d'Informatique  
de l'UASZ**

Présenté par : M. Papa Samba NDIAYE

Le 18/02/2021

Sous la direction de : M. Louis DIEME et Dr Serigne DIAGNE

Sous la supervision du Pr. Alassane DIEDHIOU

### Membres du jury :

|                    |                               |           |             |
|--------------------|-------------------------------|-----------|-------------|
| Alassane DIEDHIOU  | Professeur                    | Président | UASZ        |
| Louis DIEME        | Chef de département IT        | Encadreur | Gainde 2000 |
| Serigne DIAGNE     | Maître de Conférence          | Encadreur | UASZ        |
| Youssou DIENG      | Maître de Conférence          | Membre    | UASZ        |
| Papa Alioune CISSE | Maître de Conférence assimilé | Membre    | UASZ        |

## RESUME

Les étudiants en master 2 à l'Université Assane SECK de Ziguinchor (UASZ), notamment ceux du département d'informatique ont souvent beaucoup de mal pour soutenir leur mémoire dans les délais. En effet un mémoire de master 2 se fait en un semestre (6 mois). Cependant, les délais sont toujours dépassés et les durées des mémoires tournent en moyenne autour de 1 an et demi voir 2 ans. Ceci constitue un véritable problème tant du côté des étudiants qui perdent beaucoup de temps mais aussi de celui de l'administration et des enseignants qui voient le nombre d'étudiant en master 2 augmenter à chaque année avec l'accumulation de plusieurs promotions en mémoire.

Dans ce mémoire, nous proposons ainsi une application informatique dans le but de donner aux étudiants et enseignants une plateforme d'échange et de partage. Cette plateforme permettra aux enseignants (encadrants) d'être en contact avec les étudiants qu'ils encadrent pour suivre l'évolution de leurs travaux en temps réel et intervenir au besoin pour des corrections, remarques, suggestions, etc. Elle permet également de créer un réseau d'alumnis pour faciliter les échanges avec les anciens étudiants et l'obtention de stages et contrats pour les futurs diplômés mais aussi d'avoir une base de données de potentiels encadrants de mémoires.

**Mots clés :** Gestion, Mémoire, Master 2, Alumni, Django, Département informatique.

## ABSTRACT

Masters 2 students at Assane SECK University in Ziguinchor (UASZ), especially those in the computer science department, often find it very difficult to support their thesis on time. Indeed, a master 2 thesis is done in one semester (6 months). However, the deadlines are always exceeded and the durations of the memories turn on average around 1 year and a half see 2 years. This is a real problem not only for students who waste a lot of time, but also for the administration and teachers who see the number of Master 2 students increasing each year with the accumulation of several promotions in memory.

In this thesis, we thus propose a computer application in order to give students and teachers a platform for exchange and sharing. This platform will allow teachers (supervisors) to be in contact with the students they supervise to follow the progress of their work in real time and intervene as needed for corrections, remarks, suggestions, etc. It also makes it possible to create a network of alumni to facilitate exchanges with former students and to obtain internships and contracts for future graduates, but also to have a database of potential thesis supervisors.

**Keywords:** Management, Thesis, Master 2, Alumni, Django, IT department.

## DEDICACES

*Par la grâce du seigneur qui nous a donné la force de réaliser ce travail je dédie ce mémoire de fin d'étude à la mémoire de mon frère Sayssa NDIAYE qui représentait la grandeur de l'homme par excellence tant côté académique avec un master 2 en droit des affaires que coté familial, l'homme le plus généreux, aimable, disponible et franc que j'ai connu à ce jour. Mes premiers pas à l'université se sont faits à ses côtés. Il m'a guidé lors des démarches d'inscription et aujourd'hui je lui dédie entièrement ce mémoire et l'ensemble des réussites souhaitées qui découleront de ce travail.*

*Je ne t'aime ni avec mon cœur, ni avec mon esprit.*

*Le cœur peut s'arrêter, l'esprit peut oublier.*

*Je t'aime avec mon âme.*

*L'âme jamais ne s'arrête ni n'oublie*

*Jusqu'au jour où je te rejoindrai*

*J'espère que tu es bien installé pour regarder comment ta famille se bat pour réaliser tes rêves les uns après les autres.*

*Frères un jour liés pour toujours.*

## REMERCIEMENTS

*Je voudrais tout d'abord adresser toute ma gratitude aux directeurs de ce mémoire, M. Louis DIEME et le Docteur Serigne DIAGNE, pour leurs patiences, leurs disponibilités et surtout leurs judicieux conseils, qui ont contribué à alimenter ma réflexion.*

*Nous souhaitons adresser nos remerciements les plus sincères au corps professoral, pédagogique, administratif et les intervenants externes de l'Université Assane SECK de ZIGUINCHOR, pour la richesse et la qualité de leur enseignement et qui déploient de grands efforts pour assurer à leurs étudiants une formation actualisée spécialement le département informatique.*

*Nos remerciements vont aussi aux membres de jury en la personne de **Alassane DIEDHIU**, **Youssou Dieng** et **Papa Alioune CISSE** qui ont accepté d'évaluer ce modeste travail spécialement le président du jury **Alassane DIEDHIU** qui ménage à chaque fois son calendrier personnel et professionnel pour venir présider nos soutenances.*

*Un grand merci à ma mère et mon père, pour leur amour, leurs conseils ainsi que leur soutien inconditionnel, à la fois moral et économique, qui m'a permis de réaliser les études que je voulais et par conséquent ce mémoire.*

*Un grand merci à ma famille à savoir mes frères et sœurs ainsi que mes oncles et tantes pour leurs soutiens.*

*Je voudrais exprimer ma reconnaissance envers les amis et collègues qui m'ont apporté leur soutien moral et intellectuel tout au long de ma démarche.*

*Je remercie spécialement les anciens étudiants en la personne de Adama SEYE, Modou DIOP, Henry DIALLO, Anna BAKHOUM, Keba Mbaye DIEDHIU, Cherif Ahmed Tidiane AIDARA qui m'ont accompagné durant tout le travail en me prodiguant documents et conseils.*

*Un grand merci aux personnes qui ont pris le temps de relire et de corriger les imperfections de ce mémoire. Ainsi que toutes les autres personnes qui ont pris part à ce mémoire, d'une façon ou d'une autre, et l'ont rendu possible.*

# SOMMAIRE

|   |     |
|---|-----|
| RESUME.....   | I   |
| ABSTRACT .....  | II  |
| DEDICACES .....   | III |
| REMERCIEMENTS .....   | IV  |
| LISTE DES FIGURES .....   | IX  |
| LISTE DES TABLEAUX.....   | XI  |
| LISTE DES ABREVIATIONS .....  | XII |
| INTRODUCTION GENERALE.....  | 1   |
| CHAPITRE 1 : CONTEXTE JUSTIFICATIF DU SUJET .....   | 4   |
| 1.1. Présentation du département informatique de l'UASZ .....   | 4   |
| 1.2. Le déroulement d'un mémoire de master informatique à l'UASZ.....                                       | 5   |
| 1.2.1. Les acteurs .....  | 5   |
| 1.2.2. Le processus d'encadrement.....  | 6   |
| 1.2.3 Les problèmes liés à l'écosystème des mémoires de master 2 du département informatique de l'UASZ..... | 7   |
| 1.3 Problématique du sujet .....  | 8   |
| 1.3.1 Solution proposée .....   | 9   |
| 1.3.2 Objectifs spécifiques du sujet.....   | 10  |
| 1.4. Conclusion.....  | 10  |
| CHAPITRE 2 : PROCESSUS DE DEVELOPPEMENT DE L'APPLICATION.....   | 12  |
| 2.1. Les méthodes traditionnelles .....   | 12  |
| 2.1.1. Méthode traditionnelle – Cascade .....   | 12  |
| 2.1.2. Méthode traditionnelle - Cycle en V .....  | 14  |
| 2.2. Les méthodes agiles.....   | 15  |
| 2.2.1. Méthode agile – Scrum.....   | 15  |
| 2.2.2. Méthode agile – Kanban.....  | 16  |
| 2.2.3. Méthode agile - eXtreme Programming (XP) .....   | 17  |
| 2.3. Etude comparative entre les méthodes traditionnelles et agiles .....                                   | 19  |
| 2.4. Notre choix : SCRUMBAN.....  | 21  |
| 2.4.1. Description de la méthode SCRUMBAN.....  | 21  |
| 2.4.2. Justification de notre choix sur SCRUMBAN .....  | 23  |
| 2.5. Adaptation du Framework SCRUMBAN à notre contexte .....  | 24  |
| 2.5.1. L'organisation de l'équipe projet .....  | 24  |
| 2.5.2. Déroulement du projet sur SCRUMBAN.....  | 25  |
| 2.5.3. SCRUMBAN et l'outil Trello.....  | 25  |
| 2.6. Conclusion.....  | 26  |

|  |    |
|--|----|
| CHAPITRE 3 : SPECIFICATION ET ANALYSE DES BESOINS FONCTIONNELS.....  | 27 |
| 3.1. Présentation de UML.....  | 27 |
| 3.2. Spécification des besoins fonctionnels.....   | 27 |
| 3.2.1. Identification des acteurs.....   | 28 |
| 3.2.2. Identification des fonctionnalités du système.....  | 29 |
| 3.2.3. Diagrammes de cas d'utilisation.....  | 30 |
| 3.2.3.1. Diagramme de cas d'utilisation du responsable de master, du chef de département et des enseignants..... | 31 |
| 3.2.3.2. Diagramme de cas d'utilisation de l'étudiant et de l'Alumni.....  | 32 |
| 3.2.3.3. Diagramme de cas d'utilisation du recruteur.....  | 32 |
| 3.3. Analyse des besoins fonctionnels.....   | 33 |
| 3.3.1. Analyse de l'authentification.....  | 33 |
| a. Description du cas d'utilisation « s'authentifier ».....  | 34 |
| b. Diagramme d'activité du cas d'utilisation « s'authentifier ».....   | 34 |
| c. Diagramme de séquence du cas d'utilisation « s'authentifier ».....  | 35 |
| 3.3.2. Analyse de la proposition du sujet de mémoire.....  | 36 |
| a. Description du cas d'utilisation « proposer sujet de mémoire ».....   | 36 |
| b. Diagramme d'activité du cas d'utilisation « proposer sujet de mémoire ».....                                  | 37 |
| c. Diagramme de séquence du cas d'utilisation « proposer sujet de mémoire ».....                                 | 37 |
| 3.3.3. Analyse de l'ajout d'un document de mémoire.....  | 38 |
| a. Description du cas d'utilisation « ajouter un document de mémoire ».....                                      | 38 |
| b. Diagramme d'activité du cas d'utilisation « ajouter un document de mémoire ».....                             | 39 |
| c. Diagramme de séquence du cas d'utilisation « ajouter un document de mémoire ».....                            | 39 |
| 3.3.4. Analyse de l'encadrement d'un mémoire.....  | 40 |
| a. Description du cas d'utilisation « encadrer un mémoire ».....   | 40 |
| b. Diagramme d'activité du cas d'utilisation « encadrer un mémoire ».....  | 41 |
| 3.3.5. Analyse de la rubrique entreprise.....  | 42 |
| a. Description du cas d'utilisation « rubrique entreprise ».....   | 42 |
| b. Diagramme d'activité du cas d'utilisation « rubrique entreprise ».....  | 43 |
| c. Diagramme de séquence du cas d'utilisation « rubrique entreprise ».....                                       | 44 |
| 3.4. Besoins non fonctionnels.....   | 44 |
| 3.4.1. Les contraintes techniques.....   | 44 |
| 3.4.2. Les contraintes ergonomiques :.....   | 45 |
| 3.5. Conclusion.....   | 45 |
| CHAPITRE 4 : CONCEPTION DU SYSTEME.....  | 46 |
| 4.1. Conception générale du système.....   | 46 |
| 4.1.1. Architecture du système.....  | 46 |
| a. L'architecture MVC.....   | 46 |

|   |   |           |
|---|---|-----------|
| b.  | La spécificité de Django : le modèle MVT .....                | 47        |
| c.  | Différence entre les modèles de conception MVC et MVT : ..... | 48        |
| d.  | Architecture 3-tiers .....                                    | 49        |
| 4.1.2.  | Diagramme de composants .....                                 | 51        |
| 4.1.3.  | Diagramme de packages .....                                   | 52        |
| 4.1.4.  | Diagramme de déploiement .....                                | 54        |
| 4.1.5.  | Architecture réseau du système .....                          | 54        |
| 4.2.  | Conception détaillée du système .....                         | 56        |
| 4.2.1.  | Dictionnaire de données .....                                 | 56        |
| 4.2.2.  | Diagramme de classe du système .....                          | 56        |
| 4.3.  | Conclusion .....  | 58        |
| <b>CHAPITRE 5 : IMPLEMENTATION ET PRESENTATION DE L'APPLICATION .....</b> |   | <b>59</b> |
| 5.1.  | Implémentation de l'application .....                         | 59        |
| 5.1.1.  | Outils de conception et de développement du système .....     | 59        |
| a.  | Outils de conception : visual paradigm .....                  | 59        |
| b.  | Environnement de développement : Visual Studio Code .....     | 59        |
| c.  | Outil de versionning : git .....                              | 59        |
| 5.1.2.  | Serveurs et base de données .....                             | 60        |
| a.  | Serveur web : serveur local de Django .....                   | 60        |
| b.  | Base de données : SQLite et PostgreSQL .....                  | 60        |
| 5.1.3.  | Technologies utilisées pour la partie Front-End .....         | 61        |
| a.  | Html 5 .....  | 61        |
| b.  | CSS 3 .....   | 61        |
| c.  | Bootstrap (version 4) .....                                   | 61        |
| d.  | JQuery .....  | 62        |
| e.  | React js .....  | 62        |
| 5.1.4.  | Technologies utilisées pour la partie Back-End .....          | 62        |
| a.  | Langage de programmation : python (version 3.8.5) .....       | 62        |
| b.  | Framework Django (version 3.1.1) .....                        | 63        |
| c.  | Memcached .....   | 63        |
| d.  | RabbitMQ .....  | 63        |
| 5.1.5.  | L'arborescence de l'application .....                         | 64        |
| 5.2.  | Présentation de l'application .....                           | 69        |
| 5.3.  | Mise en production sur Heroku .....                           | 77        |
| 5.4.  | Conclusion .....  | 78        |
| <b>CHAPITRE 6 : SECURITE ET PERFORMANCE DU SYSTEME .....</b>              |   | <b>79</b> |
| 6.1.  | Audit de la sécurité du système .....                         | 79        |

|  |   |        |
|--|---|--------|
| 6.1.1.   | Sécurisation de la recherche d'information : Maltego..... | 79     |
| 6.1.2.   | Audit des failles web : Owasp Zap et Vega.....            | 80     |
| 6.2.   | Audit des performances du système : Lighthouse.....       | 83     |
| 6.3.   | Conclusion.....   | 83     |
| CONCLUSION GENERALE ET PERSPECTIVES.....   |   | 84     |
| BIBLIOGRAPHIE.....   |   | i      |
| WEBOGRAPHIE.....   |   | ii     |
| Annexe.....  |   | iv     |
| Annexe 1 : Formulaire d'enquête google Forms sur le mémoire de master 2 au département informatique de l'UASZ..... |   | iv     |
| Annexe 2 : Tableau Trello de l'organisation du projet sous SCRUMBAN.....   |   | ix     |
| Annexe 3 : Dictionnaire de données du système.....   |   | x      |
| Annexe 4 : Rapport de l'audit de la sécurité du système avec Owasp Zap.....  |   | xiii   |
| Annexe 5 : rapport de l'analyse de performance avec lighthouse.....  |   | xxii   |
| Annexe 6 : Documentation technique du système pour le déploiement local et sur le cloud (heroku).....              |   | xxviii |

## LISTE DES FIGURES

|  |    |
|--|----|
| Figure 1 : Schéma de la méthode en cascade .....   | 14 |
| Figure 2 : Schéma du cycle en V.....   | 14 |
| Figure 3 : Schéma de la méthodologie scrum .....   | 16 |
| Figure 4 : Schéma de la méthode kanban.....  | 17 |
| Figure 5 : Schéma de la méthode XP .....   | 18 |
| Figure 6 : pourcentage de réussite des projets en traditionnel contre agile.....                                   | 21 |
| Figure 7 : Schéma de la méthode SCRUMBAN.....  | 23 |
| Figure 8 : organisation d'un projet sur SCRUMBAN et Trello.....  | 26 |
| Figure 9 : Diagramme de cas d'utilisation du responsable de master, du chef de département et de l'enseignant..... | 32 |
| Figure 10 : Diagramme de cas d'utilisation de l'étudiant et de l'alumni.....                                       | 32 |
| Figure 11 : Diagramme de cas d'utilisation du recruteur.....   | 33 |
| Figure 12 : Diagramme d'activité du cas d'utilisation "s'authentifier" .....                                       | 35 |
| Figure 13 : Diagramme de séquence du cas d'utilisation "s'authentifier" .....                                      | 36 |
| Figure 14 : Diagramme d'activité du cas d'utilisation "proposer sujet de mémoire" .....                            | 37 |
| Figure 15 : Diagramme de séquence du cas d'utilisation "proposer sujet de mémoire" .....                           | 38 |
| Figure 16 : Diagramme d'activité du cas d'utilisation "ajouter un document de mémoire" .....                       | 39 |
| Figure 17 : Diagramme de séquence du cas d'utilisation "ajouter un document de mémoire" .....                      | 40 |
| Figure 18 : Diagramme d'activité du cas d'utilisation "encadrer un mémoire" .....                                  | 41 |
| Figure 19 : Diagramme de séquence du cas d'utilisation "encadrer un mémoire" .....                                 | 42 |
| Figure 20 : Diagramme d'activité du cas d'utilisation "rubrique entreprise" .....                                  | 43 |
| Figure 21 : Diagramme de séquence du cas d'utilisation "rubrique entreprise".....                                  | 44 |
| Figure 22 : Schéma de l'architecture MVC .....   | 47 |
| Figure 23 : Schéma de l'architecture MVT.....  | 48 |
| Figure 24 : Architecture 3-tiers .....   | 51 |
| Figure 25 : Architecture de l'application .....  | 51 |
| Figure 26 : diagramme de composant du système .....  | 52 |
| Figure 27 : diagramme de package du système.....   | 53 |
| Figure 28 : Diagramme de déploiement du système .....  | 54 |
| Figure 29 : Architecture réseau du système .....   | 55 |
| Figure 30 : Diagramme de classe du système .....   | 57 |
| Figure 31 : structuration du projet sous Django.....   | 64 |
| Figure 32 : application principale du projet .....   | 65 |

|   |    |
|---|----|
| Figure 33 : l'application account.....                                  | 66 |
| Figure 34 : fichier forms.py.....                                       | 67 |
| Figure 35 : fichier models.py .....                                     | 68 |
| Figure 36 : fichier admin.py .....                                      | 68 |
| Figure 37 : fichier views.py.....                                       | 69 |
| Figure 38 : vue globale de l'application.....                           | 70 |
| Figure 39.a : choix du profil.....                                      | 70 |
| Figure 39.b : formulaire d'inscription .....                            | 71 |
| Figure 40 : formulaire d'authentification du système .....              | 71 |
| Figure 41: page d'accueil du système pour le responsable de master..... | 72 |
| Figure 42 : page d'accueil sur terminal mobile.....                     | 72 |
| Figure 43 : actualisation du profil de l'utilisateur .....              | 73 |
| Figure 45 : affichage d'un enseignant .....                             | 74 |
| Figure 46 : ajout d'un document de mémoire .....                        | 75 |
| Figure 47 : liste des documents de mémoire .....                        | 76 |
| Figure 48 : ajout d'un sujet de mémoire .....                           | 76 |
| Figure 49 : liste des sujets de mémoire .....                           | 77 |
| Figure 50 : liste des tâches à faire .....                              | 77 |
| Figure 51 : timeline de l'encadrement.....                              | 77 |
| Figure 52 : audit système avec maltego .....                            | 80 |
| Figure 53 : audit sécurité web avec Zap.....                            | 81 |
| Figure 54 : Audit sécurité web avec Vega .....                          | 82 |
| Figure 55 : Résultat des tests de performance.....                      | 83 |

## LISTE DES TABLEAUX

|   |    |
|---|----|
| Tableau 1 : Comparaison entre les méthodes traditionnelles et les méthodes agiles ..... | 20 |
| Tableau 2 : Equipe du projet .....  | 24 |
| Tableau 3 : Identification des acteurs .....  | 28 |
| Tableau 4 : Identification des fonctionnalités du système .....                         | 29 |
| Tableau 5 : Description de cas d'utilisation « s'authentifier » .....                   | 34 |
| Tableau 6 : Description de cas d'utilisation « proposer sujet de mémoire » .....        | 36 |
| Tableau 7 : Description de cas d'utilisation « ajouter un document de mémoire » .....   | 38 |
| Tableau 8 : Description de cas d'utilisation « encadrer un mémoire » .....              | 40 |
| Tableau 9 : Description de cas d'utilisation « rubrique entreprise » .....              | 42 |
| Tableau 10 : Différence entre les modèles de conception MVC et MVT : .....              | 48 |

## LISTE DES ABREVIATIONS

|                |   |
|----------------|---|
| <b>A:</b>      | <b>Assistant</b>  |
| <b>API :</b>   | <b>Application Programming Interface</b>                            |
| <b>BSD4 :</b>  | <b>Berkeley Software Distribution 4</b>                             |
| <b>CIPSI :</b> | <b>Modélisation Informatique pour la Conception et l'Innovation</b> |
| <b>CSRF :</b>  | <b>Cross-Site Request Forgery</b>                                   |
| <b>CSS :</b>   | <b>Cascading Style Sheets</b>                                       |
| <b>DB :</b>    | <b>Database</b>   |
| <b>DEA :</b>   | <b>Diplôme d'Etude Approfondie</b>                                  |
| <b>DNS:</b>    | <b>Domain Name System</b>   |
| <b>DOM :</b>   | <b>Document Object Model</b>  |
| <b>HTML :</b>  | <b>Hypertext Markup Language</b>                                    |
| <b>HTTP:</b>   | <b>HyperText Transfer Protocol</b>                                  |
| <b>HTTPS:</b>  | <b>HyperText Transfer Protocol Secure</b>                           |
| <b>IaaS :</b>  | <b>Infrastructure-as-a-Service</b>                                  |
| <b>IDE:</b>    | <b>Integrated Development Environment</b>                           |
| <b>ITHE :</b>  | <b>Ingénierie Territoriale, Habitat et Environnement</b>            |
| <b>JS :</b>    | <b>JavaScript</b>   |
| <b>LAN :</b>   | <b>Local Area Network</b>   |
| <b>LI3 :</b>   | <b>Laboratoire d'Informatique et d'Ingénierie pour l'Innovation</b> |
| <b>LMD :</b>   | <b>Licence Master Doctorat</b>                                      |
| <b>M1:</b>     | <b>Master 1</b>   |
| <b>M2:</b>     | <b>Master 2</b>   |

|                 |   |
|-----------------|---|
| <b>MAC :</b>    | <b>Maître-assistant CAMES</b>   |
| <b>MCC :</b>    | <b>Maître de Conférences CAMES</b>                                    |
| <b>MICI :</b>   | <b>Modélisation Informatique pour la Conception et l'Innovation</b>   |
| <b>MVC :</b>    | <b>Modèle-Vue-Contrôleur</b>  |
| <b>.NET :</b>   | <b>dot net</b>  |
| <b>OS :</b>     | <b>Operating System</b>   |
| <b>OWASP :</b>  | <b>Open Web Application Security Project</b>                          |
| <b>PaaS :</b>   | <b>Platform-as-a-Service</b>  |
| <b>PATS :</b>   | <b>Personnel Administratif Technique et de Service</b>                |
| <b>PER :</b>    | <b>Personnel d'Enseignement et de Recherche</b>                       |
| <b>PTC :</b>    | <b>Professeur Titulaire CAMES</b>                                     |
| <b>SaaS :</b>   | <b>Software-as-a-Service</b>  |
| <b>SGBDRO :</b> | <b>Système de Gestion de Base de Données Relationnelles et Objets</b> |
| <b>SSL :</b>    | <b>Secure Socket Layer</b>  |
| <b>SQL :</b>    | <b>Structured Query Language</b>                                      |
| <b>TLS :</b>    | <b>Transport Layer Security</b>                                       |
| <b>UASZ :</b>   | <b>Université Assane SECK de Ziguinchor</b>                           |
| <b>UML:</b>     | <b>Unified Modeling Language</b>                                      |
| <b>W3C :</b>    | <b>World Wide Web Consortium</b>                                      |
| <b>XML :</b>    | <b>Extensible Markup Language</b>                                     |
| <b>XSS :</b>    | <b>Cross-Site Scripting</b>   |
| <b>ZAP :</b>    | <b>Zed Attack Proxy</b>   |

## INTRODUCTION GENERALE

L'enseignement postbac au Sénégal est organisé autour de deux composantes. D'une part, l'enseignement supérieur qui est placé sous la tutelle du ministère de l'Enseignement supérieur. Il offre des études de niveau Bac+3 à Bac+8, organisées selon le schéma LMD (Licence en 3 ans, Master en 5 ans et Doctorat en 8 ans). D'autre part, l'enseignement professionnel géré par le ministère de l'Emploi et de la formation professionnelle, qui propose des parcours de techniciens spécialisés sur 2 ou 3 ans après le baccalauréat.

Le Master 2 [1] ou M2 correspond à la 5e année des études supérieures post-bac. Depuis l'avènement du système LMD en 2012, il remplace les anciens DESS (pour le Master professionnel) et DEA (pour diplôme d'étude approfondie). A la fois un grade et diplôme, il permet de valider un niveau bac+5 avec 300 crédits (en comptant la licence) et d'obtenir un niveau de compétences reconnu par les entreprises. Le Master est un vrai tremplin vers la vie professionnelle !

L'objectif du Master 2 [1] est de mener l'étudiant soit vers l'emploi, en lui permettant de se spécialiser et de se professionnaliser ; soit vers l'enseignement et la recherche de haut niveau. En fonction de son projet, l'étudiant va donc devoir choisir entre un master recherche, qui laisse une large place à l'approfondissement théorique et l'initiation à la recherche ou bien un master professionnel. L'année comprend deux semestres dont le premier en présentiel à l'université et le second la rédaction d'un mémoire.

A l'UASZ la durée du master peut aller jusqu'à 4 ans (dépassant les 2 initiales) à cause de la dernière année de master et notamment le mémoire.

Qu'il soit porté sur la recherche ou le monde professionnel le mémoire de master 2 demande un écosystème stable permettant un encadrement, un suivi et une documentation à portée de main. L'encadrement consiste à travailler sur un sujet professionnel ou de recherche qui répond à un besoin bien défini par les PER, PATS ou un acteur externe. Ensuite le sujet est proposé à un étudiant ou choisi par ce dernier pour une durée de 6 mois qui représente le deuxième semestre du master 2.

A la suite de quoi le travail est lancé entre l'étudiant et son/ses encadreur(s). Le bon déroulement du travail est généralement entaché par un manque de suivi qui provient généralement des étudiants et des enseignants mais aussi par des perturbations comme les grèves.

A cela on peut y ajouter le manque de suivi des anciens étudiants qui permettrait de créer un véritable réseau professionnel pour permettre aux étudiants de décrocher plus facilement des stages de fin de cycle ou des emplois après obtention de leur diplôme.

Pour résoudre ce problème et créer tout un écosystème convivial pour les étudiants en Master 2 du département informatique de l'UASZ, on a jugé nécessaire de mettre en place une application sécurisée, fiable et ultra flexible qui va accompagner les étudiants dans leurs mémoires et leur insertion dans le monde professionnel.

Cette application sera articulée autour de trois problématiques qui sont :

- **L'aide au mémoire** : qui permettra
  - **Aux professeurs** de proposer des sujets de mémoire sur la plateforme et de suivre la progression de l'ensemble des étudiants qu'ils encadrent ;
  - **Aux étudiants** : de consulter les documents de mémoire des anciens étudiants et de choisir un sujet de mémoire en ligne.
- **Le suivi des alumnis** : qui permettra de garder une trace de tous les étudiants diplômés du master pour faciliter la création d'un réseau professionnel permettant aux étudiants de trouver plus facilement un stage ou un emploi en entreprise ;
- **L'authenticité des diplômes et projets auprès des entreprises** : cette rubrique permettra à une entreprise de pouvoir vérifier l'authenticité des diplômes d'un étudiant ainsi que de prouver son savoir-faire à travers les 2 grands projets que l'étudiant a réalisés à savoir le projet (ou stage de master 1) et le mémoire (ou stage de master 2).

Ainsi notre solution logicielle devra répondre à toutes les normes et bonnes pratiques utilisées dans le monde de la conception informatique à savoir la sécurité, la scalabilité, la facilité d'utilisation ainsi que l'accessibilité à internet et adaptable à toute forme de support et d'écran.

Pour l'analyse et la conception des fonctionnalités de notre système, nous avons utilisé le langage UML et pour le rendu et le traitement des informations nous avons porté notre choix sur des technologies comme python (Django), React.js, Bootstrap 4 et JQuery.

La suite du travail est organisée en 6 chapitres :

- Dans le **chapitre 1** nous présentons le *contexte justificatif du sujet*. Nous commençons par présenter le département informatique, puis nous détaillons le déroulement d'un mémoire de master 2 informatique, enfin nous donnons la problématique du sujet.

- Dans le **chapitre 2** nous parlons du *processus de développement de l'application*. Nous présentons les méthodes traditionnelles [8] et agiles [5] avec une comparaison des deux approches, ensuite nous parlons de notre choix entre ces deux approches et nous terminons par l'adaptation du choix fait au contexte de notre projet
- Dans le **chapitre 3** nous présentons *la spécification et analyse des besoins fonctionnels*. Nous présentons UML, ensuite nous spécifions les besoins fonctionnels du système, enfin nous parlons des besoins fonctionnels et non fonctionnels.
- Dans le **chapitre 4** nous évoquons la *conception du système*. Nous parlons de la conception générale ainsi que de la conception détaillée du système.
- Dans le **chapitre 5** il s'agit de *l'implémentation et la présentation de l'application*. On présente d'abord les outils et technologies utilisés pour l'implémentation du système, ensuite on présente le fonctionnement de l'application à travers ces interfaces.
- Dans le **chapitre 6** nous présentons *la sécurité et la performance du système*. Nous auditons d'abord la sécurité du système, ensuite nous auditons la performance du système.

Nous terminerons ce mémoire par une **conclusion** et une liste de suggestions en **perspectives**.

# CHAPITRE 1 : CONTEXTE JUSTIFICATIF DU SUJET

Ce chapitre permet de justifier le besoin réel de mettre en place un écosystème numérique pour la gestion et le suivi du mémoire de master. Pour mieux détailler la situation on a subdivisé le chapitre autour de trois points majeurs. Tout d'abord on va parler du département informatique de l'UASZ dans la première section (1.1), ensuite on va expliquer le déroulement d'un mémoire de master à l'UASZ dans la deuxième section (1.2), pour terminer ce chapitre par les problèmes liés au système de suivi des étudiants en master 2 dans la troisième section (1.3).

## 1.1. Présentation du département informatique de l'UASZ

La filière informatique est issue de la filière initiale Mathématiques Physique Chimie et Informatique (MPCI). Dès la licence 2, l'étudiant de MPCI a la possibilité de choisir entre Mathématiques-Informatique (MI) et Mathématiques - Physique (MP). Il faut attendre la licence 2 (semestre 4) pour pouvoir se spécialiser en informatique. Dès lors, la spécialisation complète en informatique ne commence qu'à partir de la licence 3.

Depuis quelques années, avec la Montée en Puissance de l'Université Assane Seck de Ziguinchor, le département d'informatique a mis en place un master : le Master Génie Logiciel. A partir de l'année universitaire 2019-2020, **une nouvelle spécialité** en master est ouverte à savoir le master Réseau et Système. De plus, pour éviter la spécialisation tardive des étudiants en informatique, **une nouvelle filière à partir de la licence 1** dénommée Licence en Ingénierie Informatique (L2I) est ouverte pour l'année universitaire 2020-2021.

Par ailleurs, le département propose aussi des formations professionnelles payantes qui sont :

- La Licence professionnelle créateur multimédia ;
- La Licence professionnelle en systèmes informatiques et logiciels.

Il existe également des formations certifiantes dispensées au sein de ses académies : **Cisco** et **Microsoft**.

Le département dispose aussi d'un laboratoire nommé le laboratoire d'informatique et d'ingénierie pour l'innovation (LI3) qui mène ces activités autour de 3 thématiques de recherche à savoir :

- MICI : Modélisation Informatique pour la Conception et l'Innovation.
- CIPSI : Conception et Ingénierie de Produits et Systèmes Innovants.
- ITHE : Ingénierie Territoriale, Habitat et Environnement.

En 2019 - 2020 le département informatique comptait :

- **18** enseignants chercheurs ;
- **20** étudiants en licence informatique ;
- **29** étudiants en master 1 génie logiciel ;
- **39** étudiants en master 2 génie logiciel.

## **1.2. Le déroulement d'un mémoire de master informatique à l'UASZ**

Le mémoire de master est une tâche assez complexe et demande un accompagnement rigoureux pour aider l'étudiant à mener à bien l'ensemble des tâches qui lui sont assignées.

Dans cette partie nous allons parler des acteurs qui interviennent dans le mémoire d'un étudiant, ensuite des différentes étapes qui composent la phase d'encadrement et l'après encadrement pour terminer par les problèmes liés à cet écosystème des mémoires de fin de cycles.

### **1.2.1. Les acteurs**

Un mémoire de master peut faire intervenir plusieurs types d'acteurs parmi lesquels on a :

#### **- Les enseignants**

Les enseignants sont des acteurs du système éducatif. A l'UASZ, on décompte près de 18 enseignants permanents dans le département informatique. Ils sont de différentes spécialités et sont chargés de dispenser des enseignements mais aussi d'assurer l'encadrement des étudiants pour la réalisation de mémoires et thèses. Dans le domaine universitaire, un enseignant a un grade. Les différents grades sont les suivants :

- Professeur Titulaire CAMES (PTC) ;
- Maître de Conférences CAMES (MCC) ;
- Maître-assistant CAMES (MAC) ;
- Assistant (A).

Les enseignants avec un grade de Maître-assistant CAMES (MAC) sont habilités à encadrer un mémoire de master 2

#### **- Les étudiants**

Les étudiants sont les acteurs centraux du système. Ainsi pour avoir droit à postuler pour un mémoire de master 2, il faut avoir validé le master 1 et être régulièrement inscrit en master 2. A la suite de quoi l'étudiant entamera la phase de recherche de sujet.

- **Le coordonnateur du master**

La coordination du master est un poste occupé par un enseignant pour un mandat de 3 ans. Le rôle du coordonnateur dans la gestion des mémoires est de s'assurer que tous les étudiants en semestre 4 de master ont un sujet de mémoire ou un stage en entreprise. Ensuite il est chargé de superviser l'état d'avancement global de l'ensemble des étudiants d'une promotion. Malheureusement le coordonnateur du master n'a pas tous les outils (surtout numérique) œuvrant à lui faciliter ses tâches surtout dans les prises de décision rapide et la supervision.

- **Les entreprises**

L'autre manière de valider un mémoire de master est le stage en entreprise qui est suivi d'un document scientifique (rapport de mémoire) attestant d'un savoir-faire et d'application de bonnes méthodes apprises tout au long du stage et du cursus de master.

## 1.2.2. Le processus d'encadrement

Le processus d'encadrement englobe plusieurs phases :

- **La recherche de sujet par les étudiants** : elle consiste à chercher un sujet de mémoire correspondant à l'étudiant. Ainsi des sujets de mémoire d'ordre professionnels et de recherche sont proposés aux étudiants. L'étape suivante consiste à choisir un sujet et se rapprocher de l'enseignant qui l'a proposé pour la validation et les détails du sujet ainsi que les conditions théoriques du travail. Il faut noter aussi qu'un sujet peut être proposé par l'étudiant lui-même, en ce moment il s'agira de se rapprocher d'un professeur au choix de l'étudiant pour discuter de l'encadrement. Le troisième cas de figure est l'expression d'un besoin articulé sur un sujet proposé par un membre de l'université pour régler un problème bien déterminé. En ce moment cette entité se muera en acteur et accompagnera l'étudiant et son encadreur dans le processus de développement du produit.
- **La recherche de stage** : c'est une option aussi viable que la première mais parfois plus laborieuse à décrocher du fait de la quasi absence d'une relation **UASZ-ENTREPRISE** à travers ses anciens étudiants qui faciliterait grandement la recherche de stage aux étudiants en fin de cycle. Pour ce faire l'étudiant est obligé de décrocher le stage de lui-même et gérer la lettre de convention qui lui permettra de se présenter à l'entreprise d'accueil comme étudiant de l'UASZ. Le dernier cheminement pour cette approche

consistera à trouver deux encadreurs : l'un dans l'entreprise d'accueil qui se chargera de le suivre et de lui fixer des tâches conformément au sujet du mémoire et l'autre encadreur sera un enseignant de l'UASZ qui sera la vitrine de l'université et se chargera d'accompagner l'étudiant dans le respect des chartes définies pour la réalisation d'un travail scientifique à la hauteur du diplôme de master 2.

- **L'encadrement** : c'est la partie la plus fastidieuse qui consiste à développer en un temps bien déterminer (théoriquement) le travail à faire. L'étudiant consultera les plus anciens pour apprendre les bonnes pratiques pour mener à bien un mémoire de master 2. Cette documentation se fait d'une façon laborieuse en étant obligé de contacter le propriétaire du document à consulter ou d'aller à la bibliothèque pour consulter le manuel. Ensuite vient le moment des rencontres avec l'encadreur pour faire avancer le travail et discuter des différentes phases du travail à faire. Malheureusement cette étape n'est pas toujours la meilleure car créant des déconvenues entre enseignant et étudiant entraînant le retard du mémoire. Dans cette logique aussi, l'enseignant n'a aucun outil pour horodater l'étudiant par rapport au travail à faire
- **L'après soutenance** : Après l'acquisition du master 2 l'UASZ a tendance à perdre la trace de ces anciens étudiants ne gardant plus un lien direct avec eux.

### 1.2.3 Les problèmes liés à l'écosystème des mémoires de master 2 du département informatique de l'UASZ

Pour donner suite aux remarques faites au-dessus sur l'écosystème qui entoure la gestion des mémoires, on peut en dégager une multitude de problèmes comme :

- **Absence d'archive numérique des mémoires** : Les étudiants en master n'ont aucune archive numérique des sujets de master traités par des anciens étudiants. Donc pour se documenter ils sont obligés de contacter le propriétaire d'un document ou d'aller voir l'archive physique à la bibliothèque.
- **Retard des sujets de mémoires** : Pour avoir un sujet de mémoire les étudiants sont pour la plus grande majorité obligés, d'attendre la fin des enseignements du semestre 3 de master et une période de recueil des sujets avant de pouvoir consulter les sujets proposés par les enseignants.
- **Manque de suivi rigoureux des mémoires** : A ce jour il n'existe aucun moyen pour un enseignant de suivre rigoureusement un étudiant qu'il encadre ou de savoir le nombre d'étudiant qu'il encadre et qu'il a encadré dans sa carrière et les sujets traités

si ce n'est à l'aide de logiciel tierce adapté pour l'occasion. Ce qui constitue un oubli de taille tant pour l'enseignant que pour les archives du département informatique.

- **Aucune trace des anciens étudiants :** Après leur obtention du master 2 et leur immersion en entreprise le département ne garde plus aucune trace de ces alumnis si ce n'est par contact isolé avec des tierces. Ce qui constitue un frein évident à la construction d'une relation durable avec les entreprises.
- **Aucune possibilité en main pour les étudiants de connaître les domaines et les travaux réalisés par les enseignants :** Les étudiants n'ont aucun aperçu des domaines de recherche de leur enseignant ainsi que de leurs travaux de recherche qui pourraient constituer un partenariat étudiant-enseignant pour un mémoire de recherche.

**Ainsi** avec le fort apport du numérique et l'augmentation de l'effectif du master (avec l'arrivée d'une seconde spécialité) il est impératif de résoudre les problèmes tournant autour des mémoires de master 2.

Dans le prochain point nous allons aborder la problématique du sujet ainsi que la solution que nous avons proposée au niveau des objectifs.

**(Voir en annexe 1 le formulaire d'enquête à l'encontre des étudiants sur le mémoire de master 2 au département informatique de l'UASZ)**

### 1.3 Problématique du sujet

Les problèmes tournant autour des mémoires de master ont toujours été entiers. Etudiants comme enseignant sont toujours confrontés à un dilemme au moment de l'encadrement que ça soit du côté des documents et autres. Le cœur du problème a été révélé par la covid 19 en mettant en suspend brutalement toute relation physique et en même temps beaucoup de mémoire avec, sans tenir compte d'un problème de suivi notoire qui s'est posé avec des enseignants et/ou étudiants perdus de vue.

Donc le problème repart de plus bel avec des étudiants qui ne respectent plus les délais ou qui rejettent la faute sur les enseignants et vice versa. On assiste ainsi à une accumulation de plusieurs promotions de master 2 qui n'ont pas soutenu leur mémoire (appart quelques-uns). En 2020 on a trois promotions en instance de soutenance et dont la soutenance se fait toujours attendre.

Du côté des enseignants, manque de rigueur ou lenteur dans le processus d'encadrement leurs sont souvent reprochés par les étudiants. Même si c'est le cas de certains parmi eux d'autres respectent le canevas de travail fixé avec l'étudiant mais là encore le travail est souvent

retardé car l'étudiant attend tout de l'enseignant or le mémoire est un travail individuel de l'étudiant.

Ainsi dans cette partie nous proposons une solution aux multitudes de problèmes qui constituent l'écosystème des mémoires de master 2 ainsi que l'ensemble des objectifs spécifiques du sujet de notre mémoire.

### 1.3.1 Solution proposée

Face aux problèmes et difficultés rencontrés par le département informatique de l'UASZ pour la gestion de l'écosystème des mémoires, nous avons proposé une solution informatique (application web utilisable sur toute forme de support) pour pallier à cette situation.

L'application permettra entre autres :

- **Un tableau de bord pour le responsable du master** : le responsable du master aura un tableau de bord détaillé avec plusieurs variables comme le nombre d'étudiant ayant un sujet, le nombre d'étudiants sans sujet, les étudiants ayant soutenu leur mémoire, le nombre de sujets proposés etc.
- **De disposer d'une archive numérique des anciens sujets de mémoire** : cette fonctionnalité permettra aux étudiants désireux de se documenter, de pouvoir accéder à un ensemble de sujet de mémoire traité par des anciens étudiants
- **De proposer un sujet de mémoire en tout temps** : Chaque enseignant, qu'il soit permanent ou non, pourra proposer des sujets sans limite, en tout temps et en toute période. Ce qui permettra aux étudiants d'avoir des sujets à temps.
- **De suivre des mémoires et connaissances et une planification claire avec les étudiants encadrés** : l'enseignant pourra horodater un calendrier de travail et suivre les différentes tâches du mémoire d'un étudiant. Il pourra aussi savoir le nombre d'étudiant avec lequel il travaille ainsi que l'avancement des travaux
- **De permettre aux étudiants de mieux connaître leurs enseignants** : chaque étudiant pourra accéder au profil d'un enseignant pour connaître son domaine de compétence et de recherche ainsi que ses travaux en cours.
- **De permettre aux enseignants de mieux connaître leurs étudiants** : Chaque enseignant pourra accéder au profil d'un étudiant pour connaître leurs domaines de spécialité qui peut même aider à proposer un sujet ciblé.
- **De garder une trace des anciens étudiants** : Le département informatique gardera une trace de ces anciens étudiants ainsi qu'un moyen de les contacter.

- **Du côté professionnel :** Chaque entreprise désireux de recruter un étudiant après le master pourra consulter son profil pour savoir dans quel domaine de compétences il s'est spécialisé, pour s'assurer de l'authenticité du diplôme obtenu et pour connaître les travaux qu'il a réalisés pour son mémoire de master.

### 1.3.2 Objectifs spécifiques du sujet

Les principaux objectifs de notre solution informatique pour la gestion de l'écosystème des mémoires du département informatique de l'UASZ sont les suivant :

- Doter le responsable du master d'un tableau de bord qui sera un outil incontournable d'aide à la décision ;
- Permettre aux étudiants d'accéder facilement aux mémoires soutenus par les anciens étudiants ;
- Faciliter la proposition de sujet de mémoire ;
- Permettre aux étudiants d'avoir des sujets à temps ;
- Permettre aux enseignants non permanents de proposer des sujets de mémoire plus facilement ;
- Permettre aux enseignants de suivre les étudiants qu'ils encadrent dans le temps ;
- Créer un canal de communication fiable entre étudiants et enseignants ;
- Permettre aux étudiants de connaître les spécialités des enseignants ;
- Donner l'occasion aux étudiants de connaître les travaux de recherche des enseignants ;
- Pouvoir connaître les domaines de prédilection des étudiants ;
- Garder une trace des alumnis ;
- Pouvoir décrocher des stages pour les étudiants à travers les alumnis ;
- Permettre aux entreprises de s'assurer de l'authenticité des diplômes d'un étudiant ;
- Permettre à l'étudiant de faire connaître ses compétences aux entreprises.

## 1.4. Conclusion

Ce chapitre nous a permis de présenter le département informatique de l'UASZ ainsi que d'expliquer le déroulement d'un mémoire de master. Nous avons terminé notre analyse par décliner la problématique du sujet. Pour réaliser l'ensemble des tâches susmentionnées nous avons eu besoin de travailler avec toutes les parties prenantes du projet. C'est ainsi que notre choix s'est porté sur la méthodologie SCRUMBAN qui est une démarche adaptative,

Système d'information pour la gestion des mémoires de master 2 et des alumnis : Cas du département d'Informatique de l'UASZ.

collaborative et inclusive. L'utilisation de la méthodologie SCRUMBAN est détaillée au chapitre suivant

# CHAPITRE 2 : PROCESSUS DE DEVELOPPEMENT DE L'APPLICATION

Dans la gestion de projet informatique, on distingue deux approches très importantes et différentes l'une de l'autre : **Les méthodes dites traditionnelles** et **Les méthodes dites agiles** [5]. Dans les méthodes dites traditionnelles [8], on se trouve dans un principe de projet non-itératif. Le client fournit un cahier des charges qui détaille le projet qu'il souhaite mettre en place. Après plusieurs échanges avec le client pour être certain que le besoin est clair, le prestataire va pouvoir, réaliser le projet et le livrer au client à la fin. Et si le besoin du client a changé pendant cette phase de réalisation du projet, c'est compliqué de faire machine arrière : le prestataire s'engage sur un périmètre défini (une liste de fonctionnalités bien détaillée), c'est donc plus difficile quand il s'agit d'enlever une brique complète ou bien de la refaire autrement.

Pour les méthodes dites agiles [5], on se trouve au contraire dans un principe de projet itératif, c'est-à-dire qu'on construit l'application au fur et à mesure et créer des livraisons intermédiaires. Pour ce genre de méthode, il faut évidemment que le client soit bien présent. En travaillant en parallèle avec le prestataire, il aura beaucoup plus de libertés s'il souhaite changer une fonctionnalité ou demander une évolution non pensée au début du projet. L'idée globale de l'agilité sur un projet informatique c'est de développer de manière itérative un produit, en créant des lots qui sont fonctionnels et qui ont un intérêt.

On parle dans la suite de ce chapitre de deux méthodes traditionnelles (Cascade, Cycle en V) dans la section 1 et de trois méthodes agiles [9] (Scrum, Kanban, eXtreme Programming) dans la section 2.

## 2.1. Les méthodes traditionnelles

### 2.1.1. Méthode traditionnelle – Cascade

C'est la méthode traditionnelle la plus linéaire possible. On retrouve plusieurs phases dans cette méthode, et chaque phase dépend de la phase précédente, et nécessite d'être claire et validée. En effet, un changement de conception en plein milieu du développement pourrait avoir un coût élevé et un impact assez fort.

On retrouve donc les phases suivantes :

**Exigences** : phase d'ateliers, réunions, analyse des besoins afin de présenter le projet ;

**Analyse** : phase de specifications, ou le prestataire redige des documents qui vont detaillier l'application, fonctionnellement et techniquement ;

**Conception** : definition et mise en place de l'architecture par le prestataire ;

**Mise en oeuvre** : implmentation de l'application (phase de developpement) ;

**Validation** : phase de test de l'application par le client avec eventuellement des correctifs a apporter (on appelle cela la phase de recette) ;

**Mise en service** : mise a disposition de l'outil au client, afin d'effectuer la mise en production pour les utilisateurs finaux ;

Dans cette methode, il n'y a donc pas vraiment de retour en arriere possible, car on valide une phase avant de passer a la phase suivante. L'inconvenient principal est que le produit est visible par le client que vers la fin du projet (on parle d'effet tunnel) et que le produit est tres rigide : si le client souhaite changer d'avis sur une fonctionnalite en pleine phase de developpement, c'est complique (principalement au niveau contractuel). L'avantage est que le projet beneficie d'une documentation claire et precise, et que l'on valide chaque etape avant de passer a la suivante (ca a un cote plus rassurant).

La methode en Cascade fonctionne bien pour des petits projets, ou le client est certain du produit final qu'il souhaite.

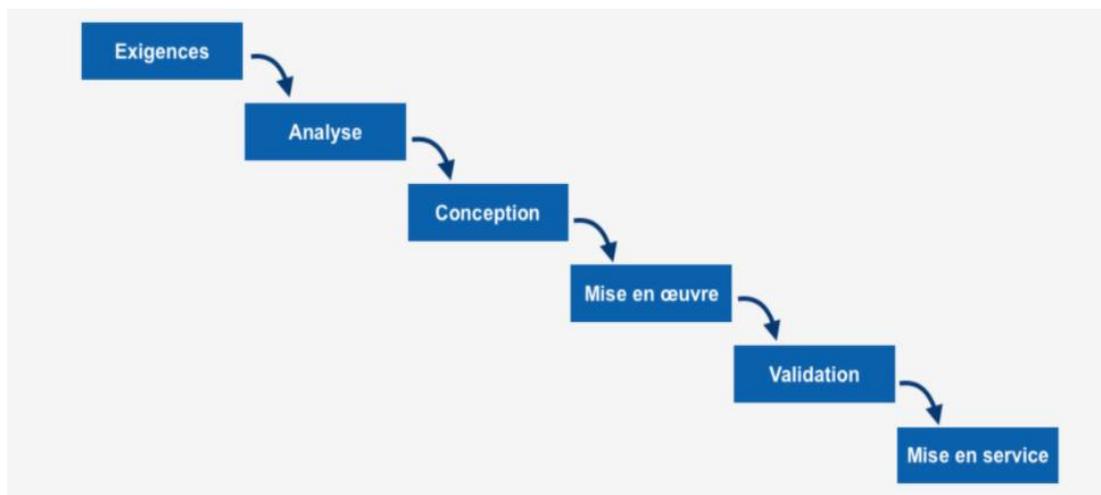


Figure 1 : Schéma de la méthode en cascade

### 2.1.2. Méthode traditionnelle - Cycle en V

Cette méthode est utilisée pour pallier aux problèmes du modèle en Cascade, car elle autorise des retours en arrière. On retrouve donc à peu près les différentes phases de la méthode en Cascade (Analyse, Spécifications, Conception, Implémentation, Tests), mais on peut autoriser des changements qui engendreront un nouveau passage par les phases précédentes.

Par exemple, si lors de la phase de recette le client pense à une nouvelle fonctionnalité, ou bien à l'évolution d'une fonctionnalité livrée, on va pouvoir faire un retour en arrière afin de spécifier ces changements, en adaptant si besoin l'architecture, en développant cette partie et en effectuant de nouveaux tests.

Avec cette méthode, on retrouve globalement les mêmes inconvénients qu'avec la méthode en cascade, c'est-à-dire cet effet tunnel, avec en plus un temps supplémentaire de rédaction de la documentation. A contrario, le projet sera clairement bien défini et il peut être intéressant d'avoir une bonne documentation sur le projet, surtout si on veut faire de la maintenance à long terme. Cette méthode fonctionnera donc bien pour les petits/moyens projets, avec un client qui a une vision assez claire de son projet.

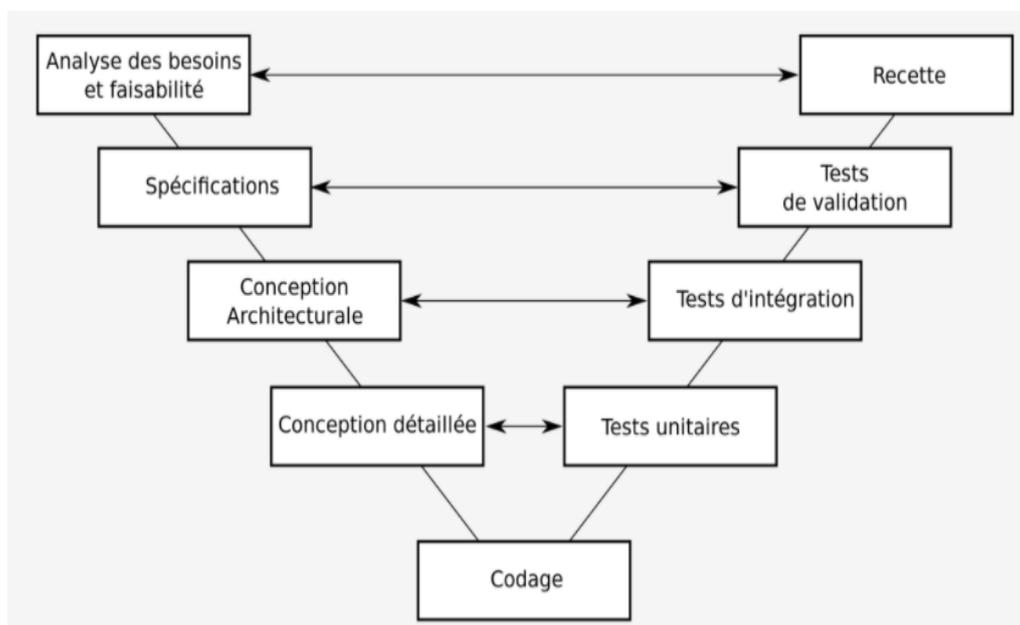


Figure 2 : Schéma du cycle en V

## **2.2. Les méthodes agiles**

### **2.2.1. Méthode agile – Scrum**

La méthode Scrum est l'une des méthodes les plus populaires dans le monde de l'agile.

Au début du cycle, on définit une liste de tâches au travers d'une réunion de planification de sprint. Ensuite, l'équipe développe les tâches sur le périmètre défini, pour une date précise de livraison. A la fin du sprint, les acteurs se réunissent pour faire une rétrospective, afin de discuter des difficultés rencontrées, ou de la charge des tâches dans le sprint, afin d'ajuster pour le cycle suivant.

L'avantage premier dans cette méthode, c'est que les livraisons se font de manière continue, sur de petites périodes. Si le client change d'avis, l'équipe en discute au prochain sprint et les tâches sont adaptées pour le nouveau besoin. L'autre avantage, c'est la communication entre tous les acteurs. Dans la méthode Scrum, l'équipe réalise ce qu'on appelle une mêlée quotidienne (ou Daily meeting en Anglais). Ce point quotidien qui doit durer 15 min en moyenne permet d'avoir une vision claire de l'état d'avancement du produit et des problèmes rencontrés.

Dans la méthode Scrum, si le client n'est pas présent, ça ne fonctionnera pas bien. Effectivement, définir des petits lots est un avantage, mais cela implique que le client doit être présent pour pouvoir définir les tâches à venir, les détailler, suivre le produit et voir s'il correspond toujours à son besoin. Un autre inconvénient, c'est qu'il y a généralement peu de documentations sur ce genre de projets.

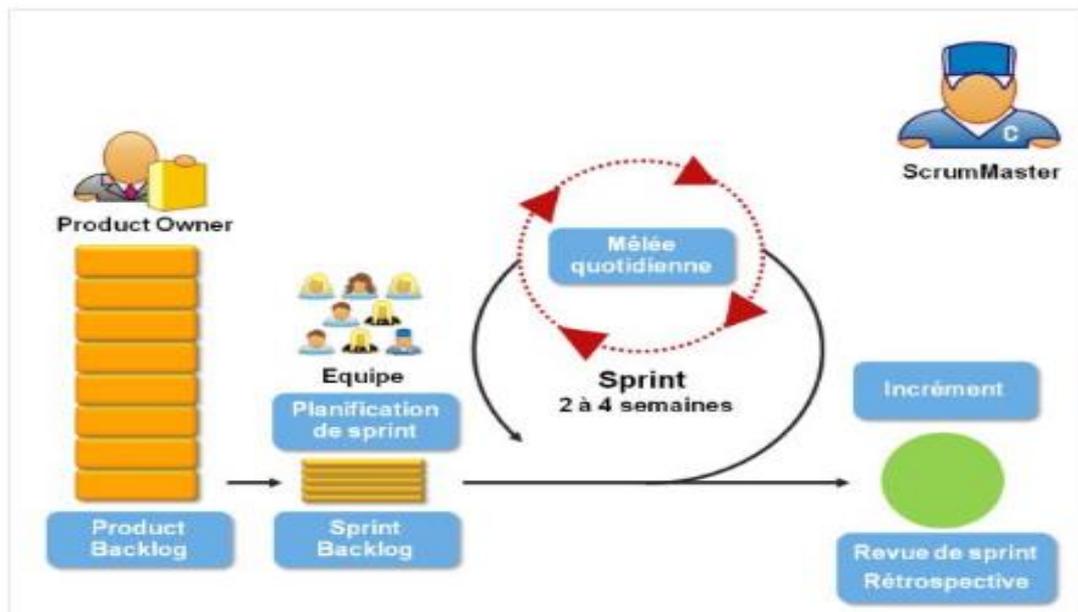


Figure 3 : Schéma de la méthodologie scrum

### 2.2.2. Méthode agile – Kanban

La méthode Kanban est une méthode souvent utilisée et assez réputée car elle permet d'avoir une vision claire de l'état d'avancement d'un projet.

Elle est très visuelle car l'objectif est de créer des colonnes pour y mettre des post-it, qui correspondent aux tâches d'un projet. On peut définir quelques colonnes de base afin de rester simple, comme on peut très bien définir beaucoup de colonne afin d'avoir beaucoup de statuts différents. L'objectif, c'est de faire avancer son post-it sur la colonne suivante, en évitant les retours en arrière. Chaque entreprise, chaque équipe projet va définir ses colonnes à sa manière, mais on retrouve souvent quelques colonnes similaires :

- **ToDo** : liste des tâches à faire ;
- **Doing** : liste des tâches en cours ;
- **Done** : liste des tâches terminées.

Après ça, on peut imaginer énormément de colonnes, qui correspondraient à d'autres statuts : à respecter, à tester, à livrer en recette, à livrer en production, en attente, à jeter (celle-là on va éviter...).

Kanban est une méthode qui est assez personnalisable selon le contexte. Elle peut être aussi très utile en complément d'une autre méthode. On pourrait très bien coupler la méthode Kanban

à la méthode Scrum. On remplira en début de sprint notre colonne **ToDo** avec les tâches à réaliser, en espérant voir toutes ces tâches dans la colonne **Done** à la fin du sprint.

L'avantage de cette méthode, c'est qu'elle est rapide à mettre en place, simple à adapter et elle est très visuelle pour avoir une vue globale du projet.

L'inconvénient est qu'il est difficile d'avoir une visibilité sur les dates de livrables, surtout si on utilise seulement cette méthode. L'autre inconvénient, c'est qu'on dispose de peu de documentations, et on a tout intérêt à avoir un client présent pour bien expliquer les tâches, car avoir quelques mots sur un post-it ne suffiront pas.

En tout cas, cette méthode est très bien pour être utilisée avec des outils comme Redmine, Jira, Mantis ou même Trello, où l'on définit (et bien détailler) des tâches avec un système de workflow et de statuts.



Figure 4 : Schéma de la méthode kanban

### 2.2.3. Méthode agile - eXtreme Programming (XP)

XP, ou eXtreme Programming n'est pas directement une méthode mais plutôt une liste de concepts pour rendre son projet agile [9]. Au total, on retrouve 13 concepts dans l'extreme programming, dont :

- **L'intégration continue** : permet de déployer et livrer automatiquement le projet (en utilisant Jenkins par exemple), dans le but d'effectuer des petites livraisons ;

- **Le travail en binôme (pair-programming)** : très utile pour la montée en compétence ou l'intégration d'une nouvelle personne dans une équipe ;
- **Le développement dirigé par les tests (Test Driven Development)** : pratique visant à développer les tests avant les features ;
- **La revue de code (code review)** : principe permettant de faire relire et valider son code par un autre développeur avant d'effectuer des changements.
- **Le refactoring** : bonne pratique permettant de repasser sur du code existant pour l'améliorer, le rendre plus lisible, en suivant des conventions de code sur le projet.

L'avantage des concepts de l'eXtreme Programming, c'est qu'ils permettent d'avoir vraiment un produit de qualité, et une bonne connaissance du projet par tous les acteurs. En revanche, l'inconvénient de ces concepts, c'est qu'ils ont un coût non négligeable. Par exemple, certains clients pourront très bien dire : pourquoi je paierai 2 développeurs sur une seule tâche, alors qu'ils pourraient faire 2 tâches en parallèle ?

Les concepts de l'eXtreme Programming ont l'avantage de pouvoir être mixés avec des méthodes traditionnelles ou d'autres méthodes agiles.

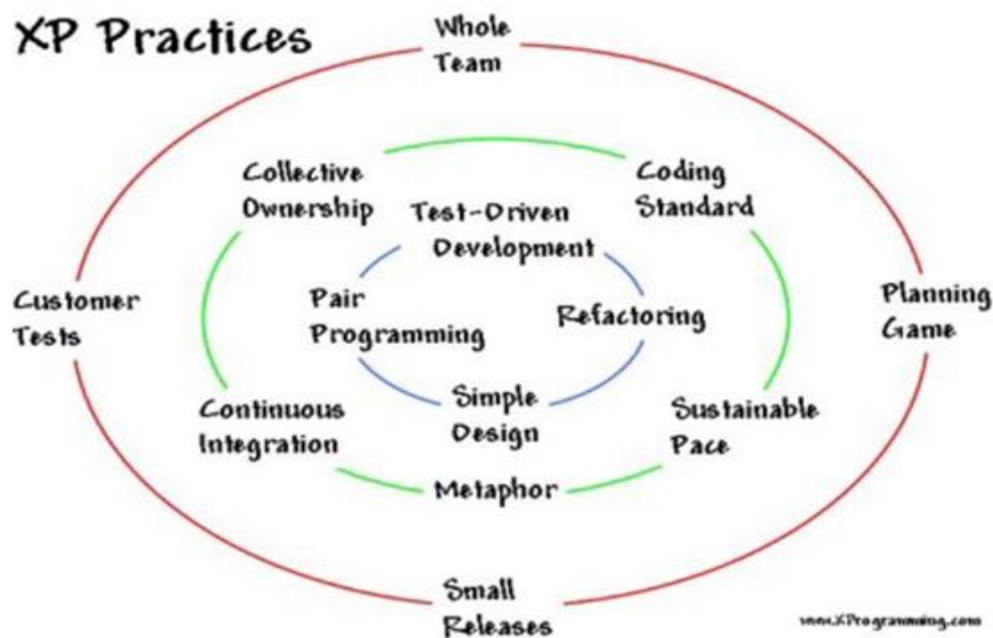


Figure 5 : Schéma de la méthode XP

## 2.3. Etude comparative entre les méthodes traditionnelles et agiles

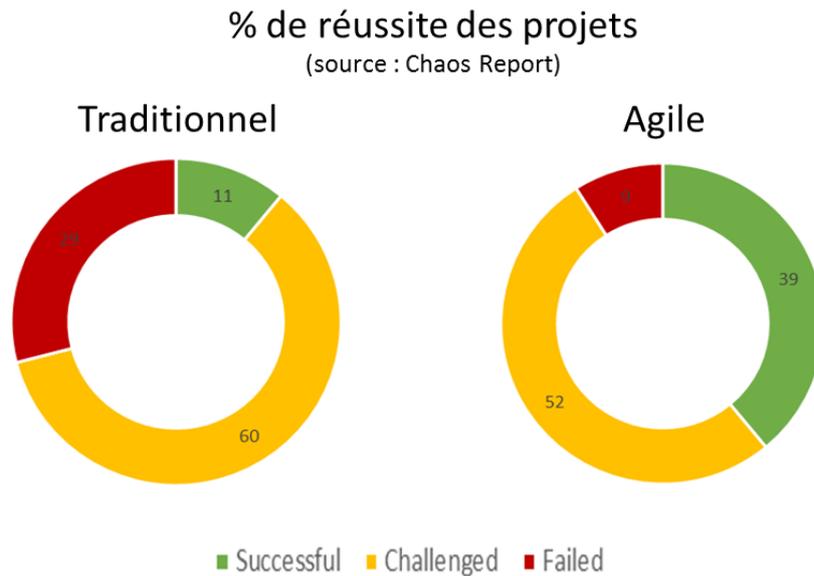
- Dans les méthodes traditionnelles, **le client n'est pas obligé d'être très présent**. On bénéficie d'une **documentation importante**, afin de bien cadrer et spécifier le projet. En revanche, on aura un **effet tunnel** pour le client, qui n'aura son produit qu'à la fin du cycle du projet.
- Dans les méthodes agiles, **le client doit être fortement présent**. On fait des **livraisons itératives**, sous forme de petits lots. **La communication est un point essentiel** dans ces méthodes, alors qu'en revanche on **négligera l'écriture d'une documentation claire et précise**.

Tableau 1 : Comparaison entre les méthodes traditionnelles et les méthodes agiles

|                                  | Approche traditionnelle  | Approche agile   |
|----------------------------------|--|--|
| Cycle de vie...                  | Phasage séquentiel, sans rétroaction, organisé autour de plusieurs phases clés : <ol style="list-style-type: none"> <li>1. Spécification des besoins ;</li> <li>2. Conception (design) ;</li> <li>3. Mise en œuvre ;</li> <li>4. Test ;</li> <li>5. Installation ;</li> <li>6. Maintenance.</li> </ol> | Itératif et incrémental : le projet va être organisé autour de « sprints » de 2 à 4 semaines, qui vont livrer une partie exploitable de la solution (ex : je livre la cuisine, et je peux tout de suite préparer des petits plats, même si je n'ai pas encore de salle à manger !).  |
| Contraint par...                 | Le périmètre et un niveau de qualité attendu.  | Les coûts et les délais... et un niveau de qualité correspondant aux besoins du client.  |
| Périmètre...                     | Stabilisé au début du projet.  | Susceptible d'évoluer au cours du projet.  |
| Equipe...                        | Une équipe avec des experts dédiés sur des lots déterminés, pilotés par un chef de projet. Et des équipiers potentiellement mobilisés sur d'autres projets en parallèle. Vie d'équipe rythmée par des réunions d'avancement toutes les n semaines.   | Une équipe dédiée au projet, au sein de laquelle la polyvalence est encouragée, afin de faire baisser le « bus factor ». Vie d'équipe rythmée par une réunion courte quotidienne (daily scrum ou daily stand-up) pour faire le point sur ce qui a été fait la veille, ce qui va être fait le jour même, les difficultés rencontrées. |
| Chef de projet...                | Oui, il y en a un !  | Pas de chef de projet, mais un Product Owner qui représente les intérêts du commanditaire et des utilisateurs, et un ScrumMaster qui est le garant de la bonne application de la méthode.  |
| Contrôle qualité...              | Réalisé à la fin du cycle de développement, une fois le produit terminé.   | Régulier, à chaque itération, sur le produit et sur le processus. Le client visualise et valide les premiers résultats très tôt, et reste associé au contrôle très fréquemment.  |
| Réaction au changement...        | Processus de gestion des modifications robuste mais lourd, autorisant finalement assez peu de changements par rapport à la cible initiale.   | Changement accepté favorablement (ex : je me rends compte que ma cuisine est finalement suffisamment grande pour que la famille puisse s'y retrouver, donc je n'ai plus besoin de salle à manger !).   |
| Suivi de l'avancement...         | Comparaison aux objectifs de contenu, qualité, coûts, délais, risques initiaux. Mesure et analyse des écarts.  | Mesuré au travers de la livraison d'incréments de la solution et du reste à faire.   |
| Le projet est un succès quand... | Les engagements initiaux sont respectés (ex : la maison est construite conformément au plan de l'architecte).  | Le client est satisfait par la solution qui lui est livrée (ex : une cuisine + une chambre + une salle de bain).   |

## L'impact sur la réussite des projets

Le **Standish Group** a publié en 2015 la nouvelle version du **Rapport CHAOS**, qui présente une photographie de l'état de l'art du développement logiciel. Le rapport, qui analyse 50 000 projets à travers le monde (de petites modifications à la mise en œuvre de refontes globales), intègre notamment une comparaison des performances des projets en fonction de l'approche utilisée :



*Figure 6 : pourcentage de réussite des projets en traditionnel contre agile*

## 2.4. Notre choix : SCRUMBAN

### 2.4.1. Description de la méthode SCRUMBAN

Son nom parle de lui-même, la méthode Scrumban fait partie des méthodes agiles [6] qui combine les avantages de la méthode Scrum et de la méthode Kanban. Son approche est cadrée comme l'est la méthode Scrum et permet de gérer des projets en flux continu grâce au Kanban.

Le Scrumban fonctionne sur une planification basée sur la demande client. Les demandes sont priorisées au fur et à mesure de l'avancement du projet. La méthode Scrumban fonctionne sous le principe d'itérations de développement (semblables à des sprints). Il n'y a pas de notion de sprint backlog. Le product owner transmet à l'équipe les tâches à développer dès que les développeurs ont terminé leurs tâches en cours. La priorisation des tâches se fait en temps réel afin de livrer aux utilisateurs le maximum de valeur et ce le plus tôt possible. La visualisation de l'avancement des tâches s'effectue dans le tableau Kanban. L'approche Scrumban conserve toutes les réunions de la méthode Scrum sans exception, ce qui permet d'avoir toujours accès à

un cadre d'organisation pour l'équipe. La méthode Scrumban fonctionne avec un travail en équipe qui s'appuie sur un tableau visuel (on utilise donc le tableau de visualisation Kanban. Dans notre cas on a utilisé TRELLO). C'est une méthode **itérative, incrémentale** et **adaptative**.

Tout comme la méthode Scrum, la méthode Scrumban fonctionne sous le principe d'itérations définies au démarrage du projet. La durée d'une itération, semblable à un sprint de développement, se mesure en semaines et ne peut excéder deux semaines. Les tâches à réaliser sont basées sur la demande. Elles sont intégrées dans la colonne du tableau « A faire » et sont également priorisées selon leur degré d'importance. Le tableau permet de visualiser l'avancement de l'équipe avec les colonnes « En cours », « En attente » et « Terminé » comme le propose la méthode Kanban. Les tâches sont déplacées de colonnes en colonnes en fonction de leur avancement.

A la différence de la méthode Scrum, l'équipe ne prévoit pas en avance l'ensemble des tâches à effectuer lors de l'itération. Par contre Les tâches seront priorisées au fur et à mesure de la capacité d'avancement de l'équipe. Il n'y a pas sprintbacklog ou de périmètre défini par le product owner comme le propose la méthode Scrum. Le travail à faire est proposé au fur et à mesure par le product owner et les développeurs développent au fil de l'eau en s'adaptant.

### ***Comment évaluer une équipe Scrumban (et Kanban) ?***

Pour évaluer les équipes Kanban et Scrumban il y a différents indicateurs à notre disposition :

- **Le lead time** : Indicateur de la fluidité de traitement des demandes par l'équipe. Le lead time représente le temps que met une demande depuis sa création jusqu'à sa finalisation (colonne « Terminé »).
- **Le cycle time** : Véritable révélateur de problèmes ou d'améliorations, le cycle time représente le nombre de jours où une demande reste dans la colonne « En cours de traitement » c'est-à-dire juste avant la colonne « Terminé ».
- **Le diagramme de flux cumulés** : Ce diagramme permet de voir le nombre de demandes qui sont en cours dans chaque colonne du tableau Kanban (de façon cumulée). Cet indicateur permet de mettre en évidence les goulots d'étranglements au sein de l'étape « En cours de traitement » par exemple, où trop de demandes arrivent comparé à la capacité de traitement des développeurs, ce qui se termine par des demandes jamais traitées.

### ***Les avantages de la méthode Scrumban.***

**Les avantages financiers :** Approprié aux projets où il est impossible d'estimer le travail à faire dans les semaines à venir.

### **Les avantages organisationnels :**

- Bénéficier des avantages des deux méthodes agiles [7] : Scrum & Kanban ;
- L'une des méthodes les plus agiles ;
- Plus grande flexibilité et adaptabilité que la méthode Scrum, il n'y a pas de sprint à respecter.

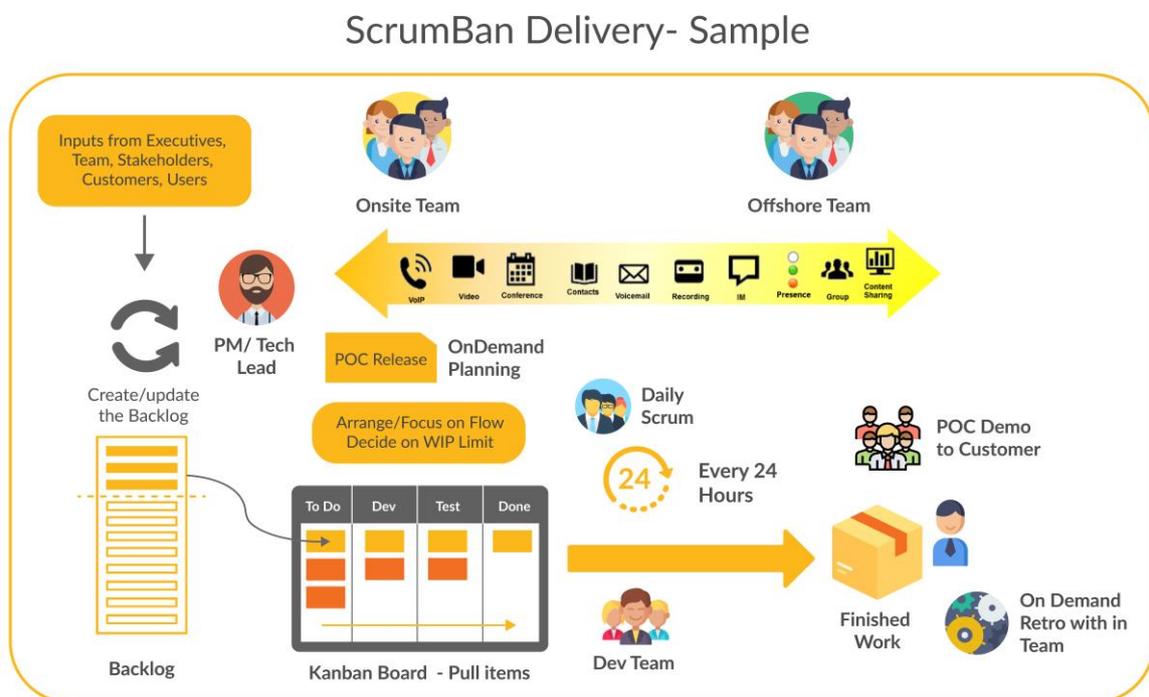


Figure 7 : Schéma de la méthode SCRUMBAN

## **2.4.2. Justification de notre choix sur SCRUMBAN**

Compte tenu des nombreux avantages des méthodes agiles par rapport aux méthodes traditionnelles, notre choix s'est naturellement porté sur les méthodes agiles du fait de sa flexibilité ainsi que de sa tolérance par rapport aux changements. Le grand atout qui nous a le plus poussé à opter pour cette approche est l'inclusion du client dans le processus de développement de son produit.

L'approche agile SCRUMBAN [6] a été choisi pour nous permettre d'apporter à SCRUM un coup de jeunesse et beaucoup plus de flexibilité qu'il ne l'était. En plus, travaillant avec une équipe offshore cette approche nous permettra d'avoir une meilleure représentation visuelle du travail à faire, en cours et terminé.

Jusque-là nous avons étudié les aspects théoriques de SCRUMBAN. Ainsi nous l'adapterons dans le contexte de notre projet pour le prochain point.

## 2.5. Adaptation du Framework SCRUMBAN à notre contexte

Dans cette partie, nous mettons en évidence la façon dont nous avons adapté la méthode Scrumban à notre projet, en commençant par la manière à laquelle l'équipe s'est organisée, ensuite décrire le déroulement du développement. Nous tenons à signaler que, dans ce projet, la méthode Scrumban n'a pas été utilisée dans son intégralité, mais nous avons seulement fait appel à ses pratiques.

### 2.5.1. L'organisation de l'équipe projet

Le tableau ci-dessous décrit l'organisation de l'équipe projet en utilisant le framework SCRUMBAN

Tableau 2 : Equipe du projet

| Personnes          | Rôles         | Fonctions  |
|--------------------|---------------|--|
| Dr. Serigne DIAGNE | Product Owner | Enseignant-chercheur au Département Informatique de l'UASZ       |
| Dr. Louis DIEME    | Scrum Master  | Enseignant chercheur, chef de département technique à Gainde2000 |
| Dr. Serigne DIAGNE |               | Enseignant-chercheur au Département Informatique de l'UASZ       |

|                      |                     |   |
|----------------------|---------------------|---|
| M. Papa Samba NDIAYE | Scrum Team (équipe) | Etudiant en Master 2 Génie Logiciel au Département d'Informatique de l'UASZ |
|----------------------|---------------------|---|

## 2.5.2. Déroutement du projet sur SCRUMBAN

Vu le contexte qui entourait l'exécution du projet (Scrum master et product owner à distance) nous avons profité des avantages qu'offraient la méthode Scrumban. A cause de la distance, nous avons fait d'internet notre principal allié. Nous avons eu assez peu de correspondance avec le product owner mais le travail a été plus acharné avec le scrum master. La réunion des mots scrum et kanban (Scrumban) n'a jamais eu autant de sens que dans notre contexte car on a exploité toutes les forces de cette méthode. Les rencontres (mêlées) se faisaient par internet plus précisément par Skype et le tableau kanban illustré dans notre cas par Trello permettait au Scrum master d'avoir une vue de premier ordre sur l'ensemble du travail à savoir ce qu'il y avait à faire, ce qui était fait et ce qui restait.

Ainsi nous avons des réunions dès que possible avec le Scrum master afin de lui présenter l'avancer des tâches ce qui ne l'interdisait à cause de la grande souplesse de Scrumban d'insérer une nouvelle tâche dans un processus en cours et d'en définir son niveau de criticité. La nouvelle tâche était immédiatement prise en charge sans contrarier le processus de développement. Ceci est possible grâce à l'approche Scrumban qui nous donne le loisir d'intégrer de nouvelles petites tâches à tout moment, d'en gérer la criticité et de l'exécuter sans pour autant bouleverser la méthode de développement en cours.

## 2.5.3. SCRUMBAN et l'outil Trello

Lorsqu'on met en place les méthodes agiles Scrum ou Kanban de gestion de projets, peu importe l'outil que nous utilisons. Ce qui compte, c'est la qualité de la collaboration en équipe et, surtout, la création de valeur. Cela dit, un bon outil aide les membres de l'équipe à visualiser facilement tous les détails du développement d'un produit, afin que tous puissent se concentrer sur la résolution de problèmes complexes à chaque sprint.

Notre choix s'est porté sur l'outil **Trello** [38] qui est un outil Kanban (SCRUMBAN aussi) idéal pour gérer les sprints car il permet aux équipes d'être constamment à jour. Et les fonctionnalités de Trello n'entravent pas la collaboration en équipe comme cela pourrait être le cas avec des outils plus complexes.

La figure 8 montre l'organisation d'un projet SCRUMBAN sur l'outil de gestion de projet Trello

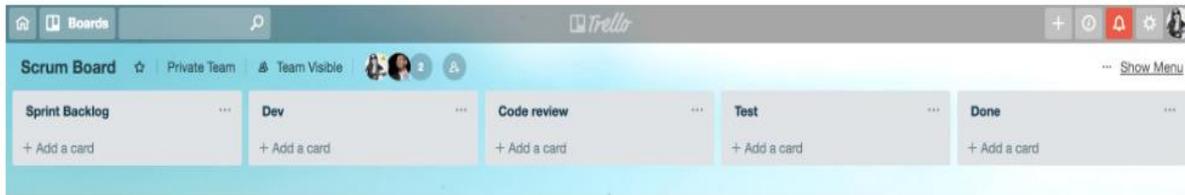


Figure 8 : organisation d'un projet sur SCRUMBAN et Trello

Les éléments clés d'un simple tableau Scrum pour les équipes de développement de logiciels sont les suivants :

- **Une liste de backlog pour le Sprint.** Celle-ci est pleine en début de sprint et vide à la fin du sprint.
- **Une ou plusieurs listes "Projets en cours".** Dans la figure 8, nous avons **Développement**, **Code review** (révision de code) et **Test**, mais on doit faire correspondre ces listes à des processus spécifiques. Au-delà de la liste "En cours", il est utile de diviser le processus de développement en plusieurs listes afin de mieux accompagner l'évolution des tâches du projet.
- **Une liste "Terminé (done)"** contenant tout le travail déjà accompli (un petit rappel de ce qui est accompli).

*Le tableau SCRUMBAN sur TRELLO de notre projet est disponible à l'annexe 2.*

## 2.6. Conclusion

Scrumban s'est avéré être un cadre très utile pour réaliser ensemble le travail de support et de développement. Scrumban a ainsi l'avantage de s'adapter dans des contextes de production où il est difficile de quantifier le travail à effectuer sur les semaines à venir. Il n'y a pas de sprint à respecter, la flexibilité et l'adaptabilité sont d'autant plus grandes dans ce contexte.

Le chapitre suivant est une illustration de l'utilisation du Framework Scrumban dans la spécification et l'analyse des besoins fonctionnels de notre système.

# CHAPITRE 3 : SPECIFICATION ET ANALYSE DES BESOINS FONCTIONNELS

Suite à la définition du processus de développement de notre application, nous entamons, dans ce chapitre, la spécification (identification des acteurs et fonctionnalités) et l'analyse (description des fonctionnalités) des besoins fonctionnels.

Nous y parlons d'abord de la modélisation objet en faisant appel à UML [36] dans la section 3.1 afin de traduire, en diagrammes fonctionnels, les spécifications des besoins fonctionnels avant d'aborder les spécifications des besoins dans la section 3.2 et l'analyse des besoins fonctionnels dans la section 3.3.

## 3.1. Présentation de UML

On a utilisé dans ce projet comme langage de modélisation UML (**Unified Modeling Language** ou « langage de modélisation unifié »). UML est un langage de modélisation graphique, apparu dans le monde du génie logiciel, dans le cadre de la « conception orientée objet ».

Il présente plusieurs avantages à citer :

- UML est basée sur les mécanismes d'abstraction, d'hierarchie et de décomposition.
- UML est un support de communication performant : il permet grâce à sa représentation graphique, d'exprimer visuellement une solution objet, de faciliter la comparaison et l'évolution de solution ;
- UML est un langage formel et normalisé, il permet d'accéder à des informations précises et de garantir la stabilité ;
- UML facilite la compréhension de représentations abstraites complexes ;
- UML est basé sur la mise en place de diagrammes représentant les vues statiques et dynamiques dans le développement d'une application informatique.

## 3.2. Spécification des besoins fonctionnels

Dans cette partie, il est nécessaire de délimiter le contexte du sujet en faisant une étude fonctionnelle du système. Il s'agit donc de montrer l'ensemble des acteurs agissant au sein de notre système, ensuite les fonctionnalités et terminer par les cas d'utilisation spécifiques.

### 3.2.1. Identification des acteurs

Un acteur est l'archétype de l'utilisateur (personne, processus externe, ...) qui interagit avec le système. Par défaut, c'est un acteur principal, c'est-à-dire qu'il agit directement sur le système et en attend des résultats ou biens, il peut être un acteur secondaire qui est souvent sollicité pour des informations supplémentaires.

L'ensemble des acteurs de notre système ainsi que leurs cas d'utilisation sont répertoriés dans le tableau suivant :

Tableau 3 : Identification des acteurs

| <u>Acteurs</u>                  | <u>Actions</u>  |
|---------------------------------|---|
| <b>Responsable de master</b>    | Il est l'administrateur du système et un professeur aussi. Il consulte le tableau de bord pour l'aide à la prise de décision mais aussi poste des sujets de mémoire, consulte le profil des étudiants et alumnis mais aussi assure le bon déroulement de l'encadrement des mémoires de master 2.<br>Il peut aussi au besoin supprimer un utilisateur. |
| <b>Chef de département</b>      |   |
| <b>Enseignant</b>               | Poste des sujets de mémoires dans la plateforme, encadre un étudiant avec un timeline complet de suivi jusqu'à la soutenance, consulte les profils des étudiants, des alumnis, met à jour son profil  |
| <b>Etudiant</b>                 | Consulte les documents de mémoire postés par les anciens étudiants, actualise son profil, choisit parmi les sujets de mémoire postés par les enseignants, consulte le profil d'un enseignant, consulte le profil d'un alumni.   |
| <b>Alumni (ancien étudiant)</b> | Poste son document de mémoire, ajoute des informations supplémentaires comme l'entreprise ou il travaille.  |
| <b>Recruteur</b>                | Consulte les profils des étudiants, les rapports de mémoire des étudiants ainsi que les procès-verbaux de soutenance.   |

Un acteur est toujours lié à une ou des action(s) du système. Ces actions sont souvent appelées des **fonctionnalités**. Nous allons, dans la section suivante, identifier l'ensemble des fonctionnalités de notre système.

### 3.2.2. Identification des fonctionnalités du système

Les fonctionnalités sont les actions ou services pouvant être rendus par le système en cas de sollicitation de l'acteur concerné. Ces services peuvent être externes (nécessitant un acteur) ou interne, c'est-à-dire non-visible, mais fonctionnant en interne ne nécessitant pas l'intervention d'un acteur. Ainsi nous avons pu analyser l'ensemble des fonctionnalités de notre système qui sont répertoriés dans le tableau suivant :

Tableau 4 : Identification des fonctionnalités du système

| <u>Fonctionnalités</u>  | <u>Acteurs</u>  |
|---|---|
| <b>1. S'inscrire</b>  | Tous les utilisateurs   |
| <b>2. S'authentifier</b>  | Tous les utilisateurs   |
| <b>3. Visualiser le tableau de bord</b>                                       | Responsable du master, chef de département  |
| <b>4. Ajouter un document de mémoire</b>                                      | Alumni  |
| <b>5. Gérer (supprimer ou modifier un document de mémoire)</b>                | Alumni  |
| <b>6. Télécharger un document de mémoire</b>                                  | Etudiant, alumni, responsable du master, chef de département, enseignant, recruteur |
| <b>7. Ajouter un sujet de mémoire</b>   | Enseignant  |
| <b>8. Gérer un sujet de mémoire (retirer ou modifier) un sujet de mémoire</b> | Enseignant  |
| <b>9. Choisir un étudiant pour traiter le sujet de mémoire</b>                | Enseignant  |
| <b>10. Encadrer un mémoire</b>  | Enseignant  |
| <b>11. Arrêter un encadrement</b>   | Enseignant  |
| <b>12. Consulter le profil d'un enseignant</b>                                | Enseignant, responsable du master, chef de département, étudiant, alumni            |
| <b>13. Mettre à jour le profil d'un enseignant</b>                            | Enseignant  |
| <b>14. Supprimer un enseignant</b>  | Responsable du master, chef du département  |
| <b>15. Consulter le profil d'un étudiant</b>                                  | Enseignant, responsable du master, chef de département, étudiant, alumni            |

|  |  |
|--|--|
| <b>16. Mettre à jour le profil d'un étudiant</b> | Etudiant   |
| <b>17. Supprimer un étudiant</b>                 | Responsable du master  |
| <b>18. Passer d'étudiant à alumni</b>            | Etudiant   |
| <b>19. Consulter le profil d'un alumni</b>       | Enseignant, responsable du master, chef de département, étudiant, alumni |
| <b>20. Mettre à jour le profil d'un alumni</b>   | Alumni   |
| <b>21. Supprimer un alumni</b>                   | Responsable du master, chef de département, Alumni                       |
| <b>22. Consulter pv soutenance mémoire</b>       | Enseignant, responsable du master, chef de département, Recruteur        |
| <b>23. Changer son mot de passe</b>              | L'utilisateur lui-même   |
| <b>24. Réinitialiser son mot de passe</b>        | L'utilisateur lui-même   |
| <b>25. Définir la langue du système</b>          | L'utilisateur lui-même   |
| <b>26. Consulter le calendrier</b>               | L'utilisateur lui-même   |

Les liaisons entre acteurs et fonctionnalités sont représentées, en UML, par des diagrammes de cas d'utilisation. Nous allons, dans la section qui suit, élaborer les diagrammes de cas d'utilisation de notre système.

### **3.2.3. Diagrammes de cas d'utilisation**

Les diagrammes de cas d'utilisation (DCU) sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet.

Un cas d'utilisation représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système. Il est une unité significative de travail. Dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs (actors), ils interagissent avec les cas d'utilisation (use cases). Dans notre cas nous avons défini trois diagrammes de cas d'utilisation répartis entre les différents acteurs du système.

- Le premier diagramme concerne le responsable de master, le chef de département et les enseignants ;
- Le deuxième diagramme concerne les étudiants et les alumnis ;
- Le troisième diagramme sera pour les recruteurs.

### 3.2.3.1. Diagramme de cas d'utilisation du responsable de master, du chef de département et des enseignants

Le diagramme de cas d'utilisation du responsable de master, du chef de département et des enseignants nous permet de rassembler trois acteurs par relation d'héritage. Le responsable du master étant l'administrateur du système, il reste néanmoins un enseignant et peut effectuer l'ensemble des tâches attribuable à un enseignant. Le chef du département reste aussi un enseignant et le chef du département informatique donc il jouit des mêmes privilèges que le responsable du master. Ces privilèges pourraient être accordés à un enseignant désigné comme adjoint du responsable de master en place.

Le diagramme de cas d'utilisation suivant illustre l'ensemble des acteurs cités ci-dessous et l'ensemble des fonctionnalités auxquelles ils ont accès

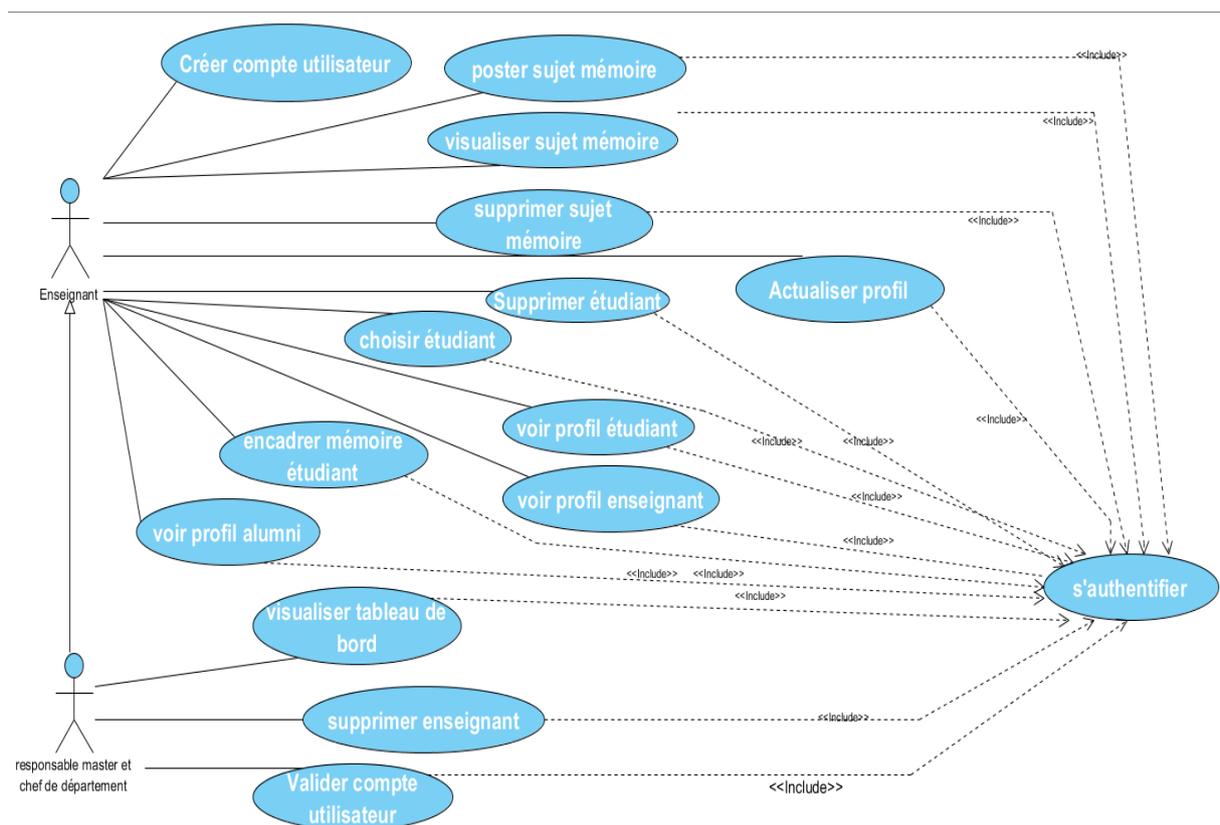


Figure 9 : Diagramme de cas d'utilisation du responsable de master, du chef de département et de l'enseignant

### 3.2.3.2. Diagramme de cas d'utilisation de l'étudiant et de l'Alumni

Le diagramme de cas d'utilisation de l'étudiant et de l'alumni nous permet de représenter l'ensemble des fonctionnalités dont à accès l'étudiant et son agrégat l'alumni.

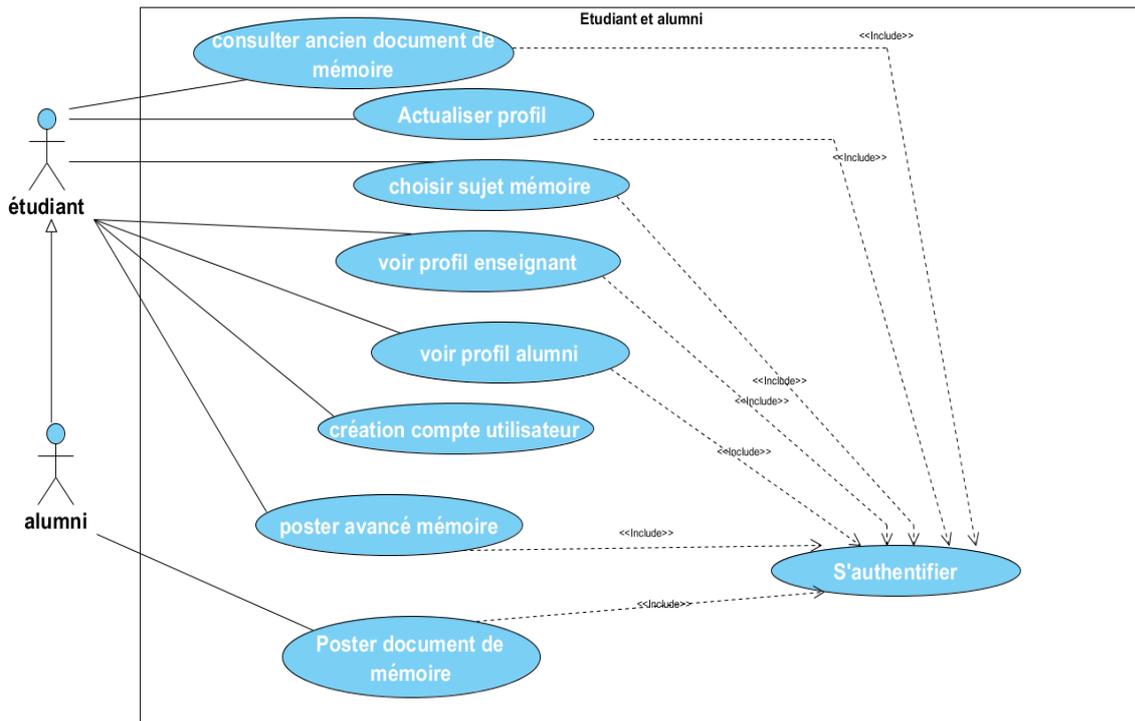


Figure 10 : Diagramme de cas d'utilisation de l'étudiant et de l'alumni

### 3.2.3.3. Diagramme de cas d'utilisation du recruteur

Le diagramme de cas d'utilisation du recruteur nous permet de spécifier l'ensemble des fonctionnalités auxquelles les recruteurs auront accès dans notre système.

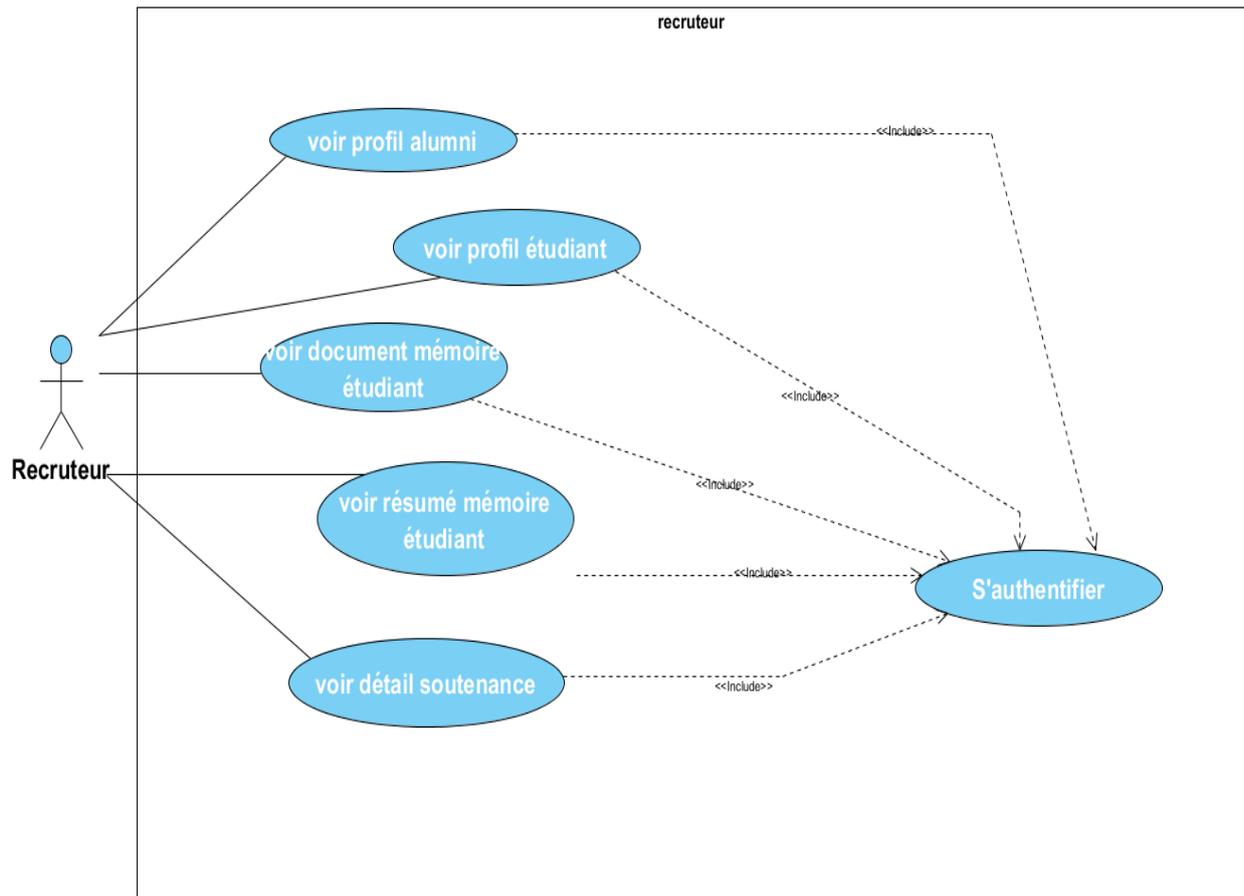


Figure 11 : Diagramme de cas d'utilisation du recruteur

Les besoins specifies ont besoins d'analyse plus complete dans des systemes assez complexes comme le notre. C'est ce qui fera l'objet de la prochaine section dans laquelle nous analysons les besoins fonctionnels de notre systeme.

### 3.3. Analyse des besoins fonctionnels

L'analyse des besoins fonctionnels est une demarche qui consiste a decrire les fonctionnalites du systeme. Il sera question dans cette partie d'analyser les besoins fonctionnels les plus complexes de notre systeme. Nous allons commencer par analyser l'authentification avant d'enchaîner par les besoins : Proposer un sujet de memoire, Ajouter un document de memoire, Encadrer un memoire et rubrique entreprise.

#### 3.3.1. Analyse de l'authentification

Dans cette section, nous decrivons d'abord le cas d'utilisation « s'authentifier », ensuite nous elaborons ses diagrammes d'activite et de sequence.

### a. Description du cas d'utilisation « s'authentifier »

Le tableau 5 ci-dessous permet de décrire le cas d'utilisation « s'authentifier ».

Tableau 5 : Description de cas d'utilisation « s'authentifier »

| Description du cas d'utilisation « s'authentifier » |  |
|---|--|
| <b>Titre</b>  | S'authentifier   |
| <b>Résumé</b>                                       | Permet de vérifier l'accès au système  |
| <b>Acteurs</b>                                      | Responsable du master, Chef de département, Enseignant, Etudiant, Alumni, Recruteur.   |
| <b>Pré condition</b>                                | Avoir un compte d'utilisateur  |
| <b>Scénario nominal</b>                             | <ul style="list-style-type: none"><li>• L'utilisateur saisit son identifiant et son mot de passe ;</li><li>• Le système vérifie les informations saisies ;</li><li>• Le système récupère le profil de l'utilisateur.</li></ul> |
| <b>Post condition</b>                               | Affichage de la page d'accueil correspondant au profil de l'utilisateur  |
| <b>Exception</b>                                    | <ol style="list-style-type: none"><li>1. Saisie de l'identifiant ou du mot de passe incorrect</li><li>2. Compte inexistant</li></ol>   |

### b. Diagramme d'activité du cas d'utilisation « s'authentifier »

Un diagramme d'activité permet de modéliser un processus interactif, global ou partiel pour un système donné (logiciel, système d'information). Une activité est l'exécution d'une partie du cas d'utilisation. Elle est représentée par un rectangle aux bords arrondis. La Figure 12 représente le diagramme d'activité du cas d'utilisation « s'authentifier »

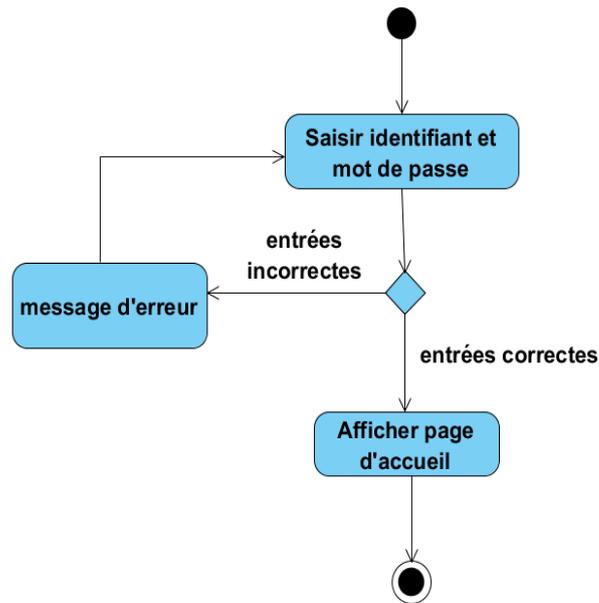


Figure 12 : Diagramme d'activite du cas d'utilisation "s'authentifier"

### c. Diagramme de sequence du cas d'utilisation « s'authentifier »

Un diagramme de sequence est un type de diagramme d'interaction, car il decrit comment et dans quel ordre plusieurs objets fonctionnent ensemble. Ces diagrammes sont utilises a la fois par les developpeurs logiciels et les managers d'entreprises pour analyser les besoins d'un nouveau systeme ou documenter un processus existant. Les diagrammes de sequence sont parfois appeles diagrammes d'evenements ou scenarios d'evenements.

La figure 13 represente le diagramme de sequence illustrant le scenario nominal de l'authentification.

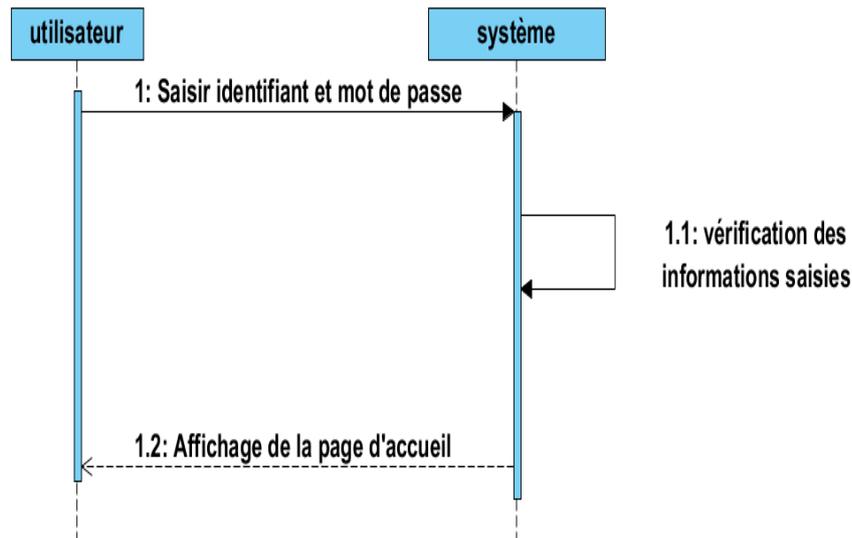


Figure 13 : Diagramme de séquence du cas d'utilisation "s'authentifier"

### 3.3.2. Analyse de la proposition du sujet de mémoire

#### a. Description du cas d'utilisation « proposer sujet de mémoire »

Tableau 6 : Description de cas d'utilisation « proposer sujet de mémoire »

| Description du cas d'utilisation « proposer sujet de mémoire » |   |
|--|---|
| <b>Titre</b>   | Proposer un sujet de mémoire  |
| <b>Résumé</b>  | Permet aux enseignants de proposer des sujets de mémoire sur la plateforme.   |
| <b>Acteurs</b>   | Responsable du master, Chef de département, Enseignant  |
| <b>Pré condition</b>   | Être un enseignant  |
| <b>Scénario nominal</b>  | <ul style="list-style-type: none"> <li>• L'enseignant renseigne le titre du sujet</li> <li>• L'enseignant renseigne la description (corps) du sujet</li> <li>• L'enseignant renseigne les critères d'éligibilité</li> <li>• L'enseignant valide</li> <li>• Le système crée le sujet de mémoire</li> </ul> |
| <b>Post condition</b>  | Le sujet est placé dans la page des sujets de mémoire   |
| <b>Exception</b>   | Erreur dans les données saisies   |

**b. Diagramme d'activité du cas d'utilisation « proposer sujet de mémoire »**

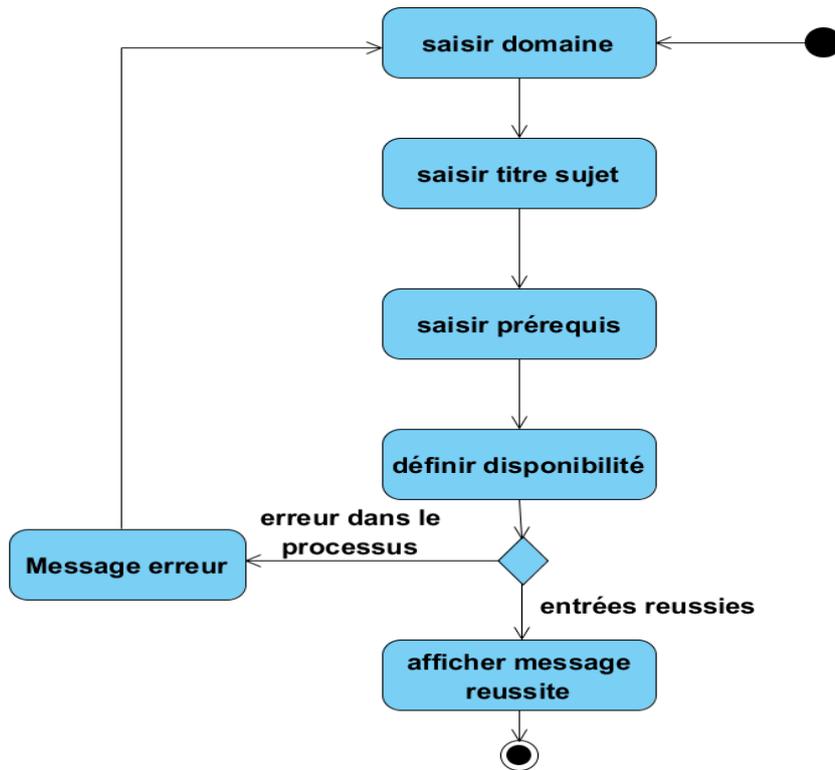


Figure 14 : Diagramme d'activité du cas d'utilisation "proposer sujet de mémoire"

**c. Diagramme de séquence du cas d'utilisation « proposer sujet de mémoire »**

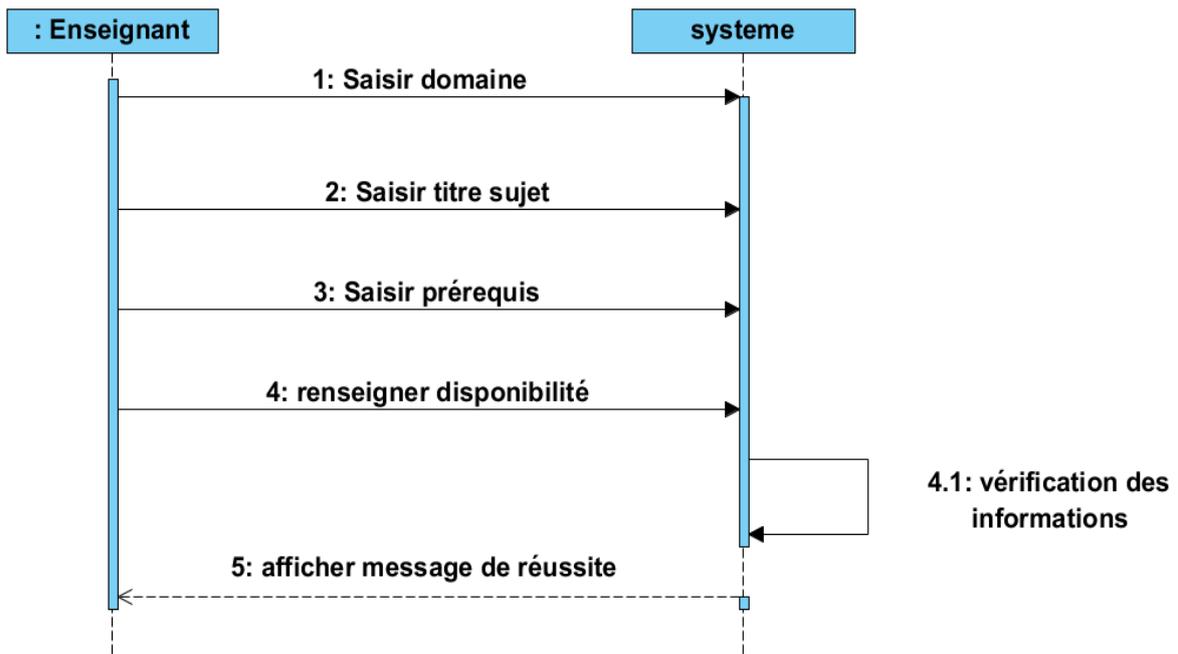


Figure 15 : Diagramme de séquence du cas d'utilisation "proposer sujet de mémoire"

### 3.3.3. Analyse de l'ajout d'un document de mémoire

#### a. Description du cas d'utilisation « ajouter un document de mémoire »

Tableau 7 : Description de cas d'utilisation « ajouter un document de mémoire »

| Description du cas d'utilisation « ajouter un document de mémoire » |   |
|---|---|
| <b>Titre</b>  | Ajouter un document de mémoire  |
| <b>Résumé</b>   | Permet d'ajouter le document de mémoire d'un étudiant ayant déjà soutenu son mémoire de master 2  |
| <b>Acteurs</b>  | Alumni,   |
| <b>Pré condition</b>  | Avoir soutenu son master 2  |
| <b>Scénario nominal</b>   | <ul style="list-style-type: none"> <li>• l'alumni renseigne le titre du mémoire</li> <li>• l'alumni ajoute la couverture du mémoire</li> <li>• l'alumni ajoute le document de mémoire</li> <li>• l'alumni valide</li> <li>• le système enregistre les informations et affiche un message de réussite</li> </ul> |
| <b>Post condition</b>   | Le document de mémoire est placé dans la page dédiée au mémoire des alumnis   |
| <b>Exception</b>  | Document trop lourd pour être téléversé<br>Erreur dans la saisie des données<br>Le format de l'image de couverture n'est pas adapté   |

**b. Diagramme d'activité du cas d'utilisation « ajouter un document de mémoire »**

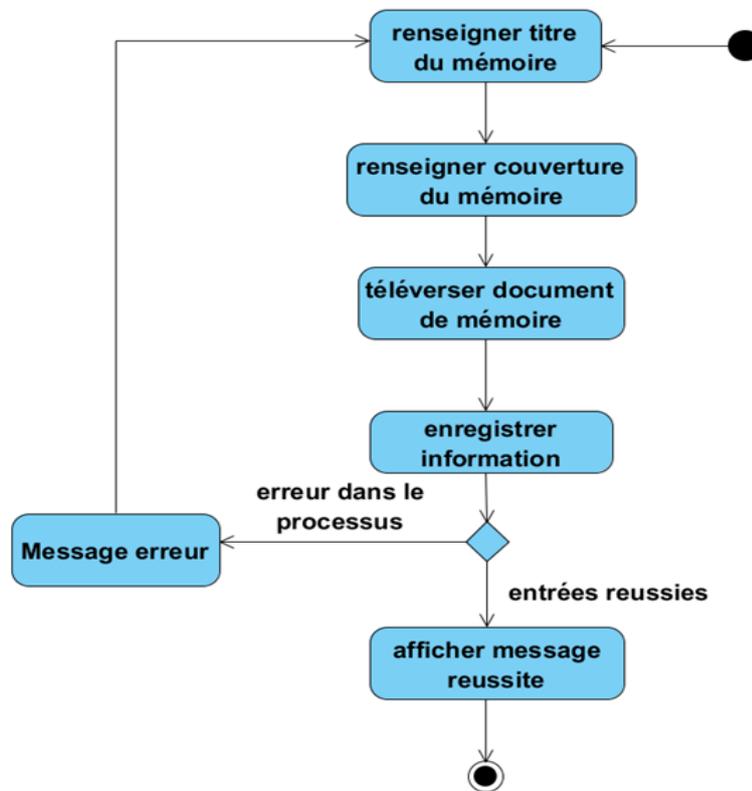


Figure 16 : Diagramme d'activité du cas d'utilisation "ajouter un document de mémoire"

**c. Diagramme de séquence du cas d'utilisation « ajouter un document de mémoire »**

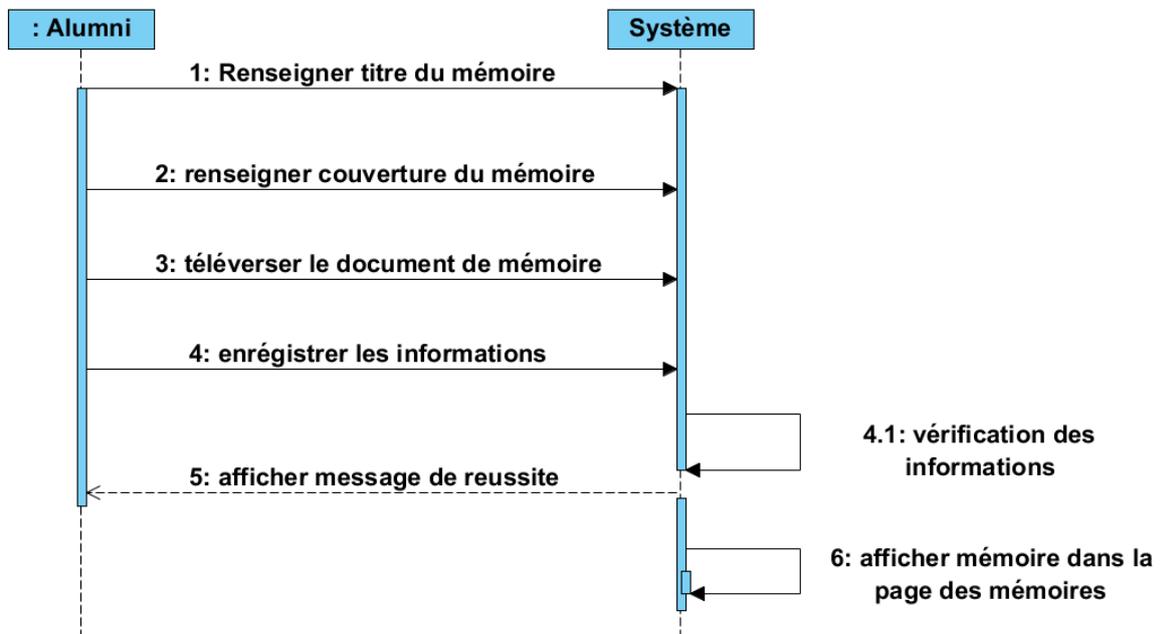


Figure 17 : Diagramme de séquence du cas d'utilisation "ajouter un document de mémoire"

### 3.3.4. Analyse de l'encadrement d'un mémoire

#### a. Description du cas d'utilisation « encadrer un mémoire »

Tableau 8 : Description de cas d'utilisation « encadrer un mémoire »

| Description du cas d'utilisation « encadrer un mémoire » |  |
|--|--|
| <b>Titre</b>   | Encadrer un mémoire  |
| <b>Résumé</b>  | Permet à un enseignant de suivre rigoureusement l'encadrement d'un étudiant  |
| <b>Acteurs</b>   | Responsable du master, Chef de département, Enseignant.  |
| <b>Pré condition</b>                                     | Être un enseignant   |
| <b>Scénario nominal</b>                                  | <ul style="list-style-type: none"> <li>• L'étudiant choisit le sujet de l'enseignant.</li> <li>• L'enseignant valide l'étudiant</li> <li>• L'enseignant définit la date de début de l'encadrement</li> <li>• L'étudiant poste son travail</li> <li>• L'enseignant corrige</li> <li>• Le système envoie des messages de rappel en cas de manque d'activité selon une période définie par l'encadreur</li> <li>• L'enseignant clôture l'encadrement à la fin.</li> </ul> |
| <b>Post condition</b>                                    | Affichage du message indiquant la fin de l'encadrement   |
| <b>Exception</b>   | L'enseignant refuse l'étudiant<br>L'étudiant ne respecte pas les échéances et l'enseignant met fin à l'encadrement   |

**b. Diagramme d'activité du cas d'utilisation « encadrer un mémoire »**

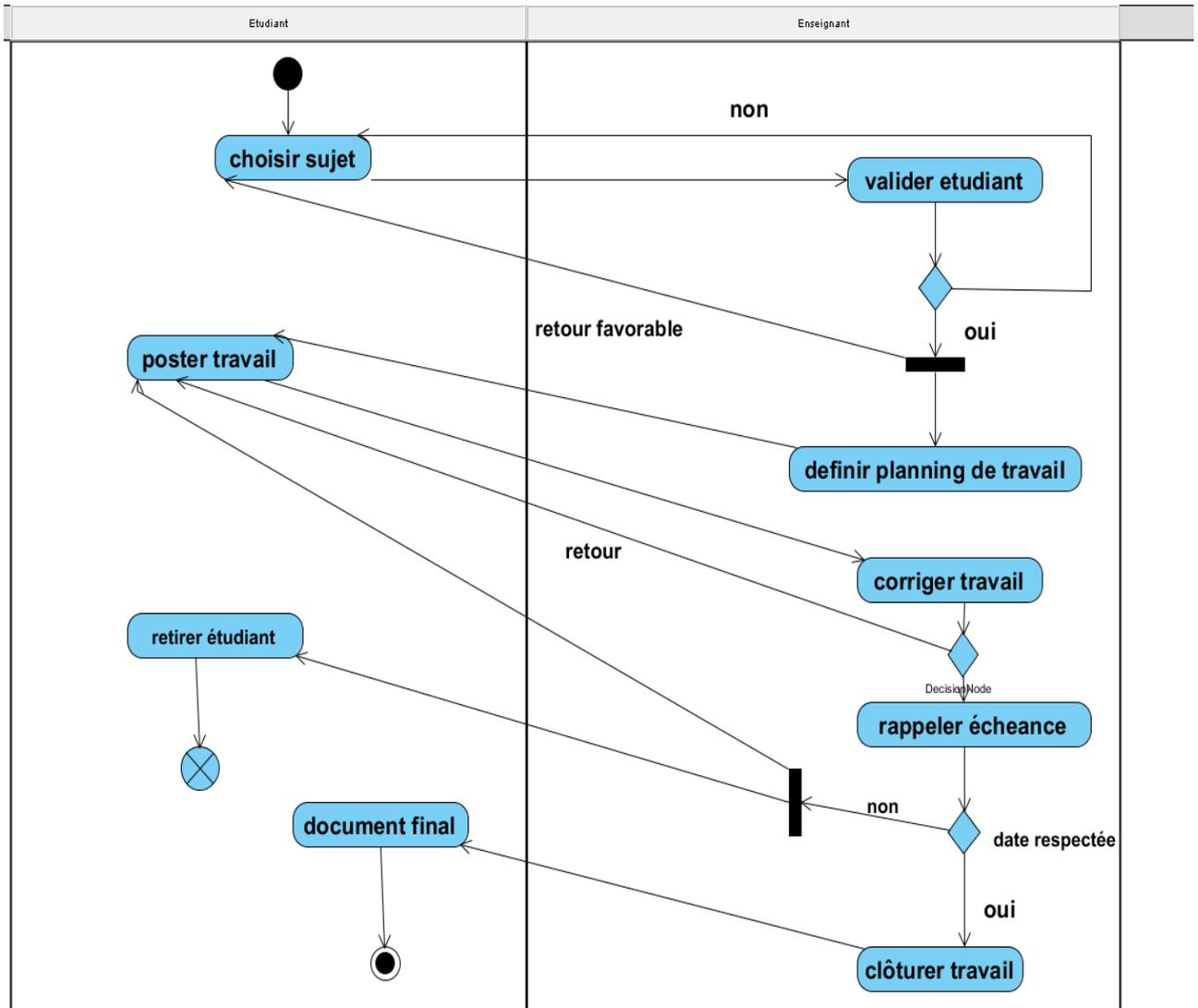


Figure 18 : Diagramme d'activité du cas d'utilisation "encadrer un mémoire"

### c. Diagramme de séquence du cas d'utilisation « encadrer un mémoire »

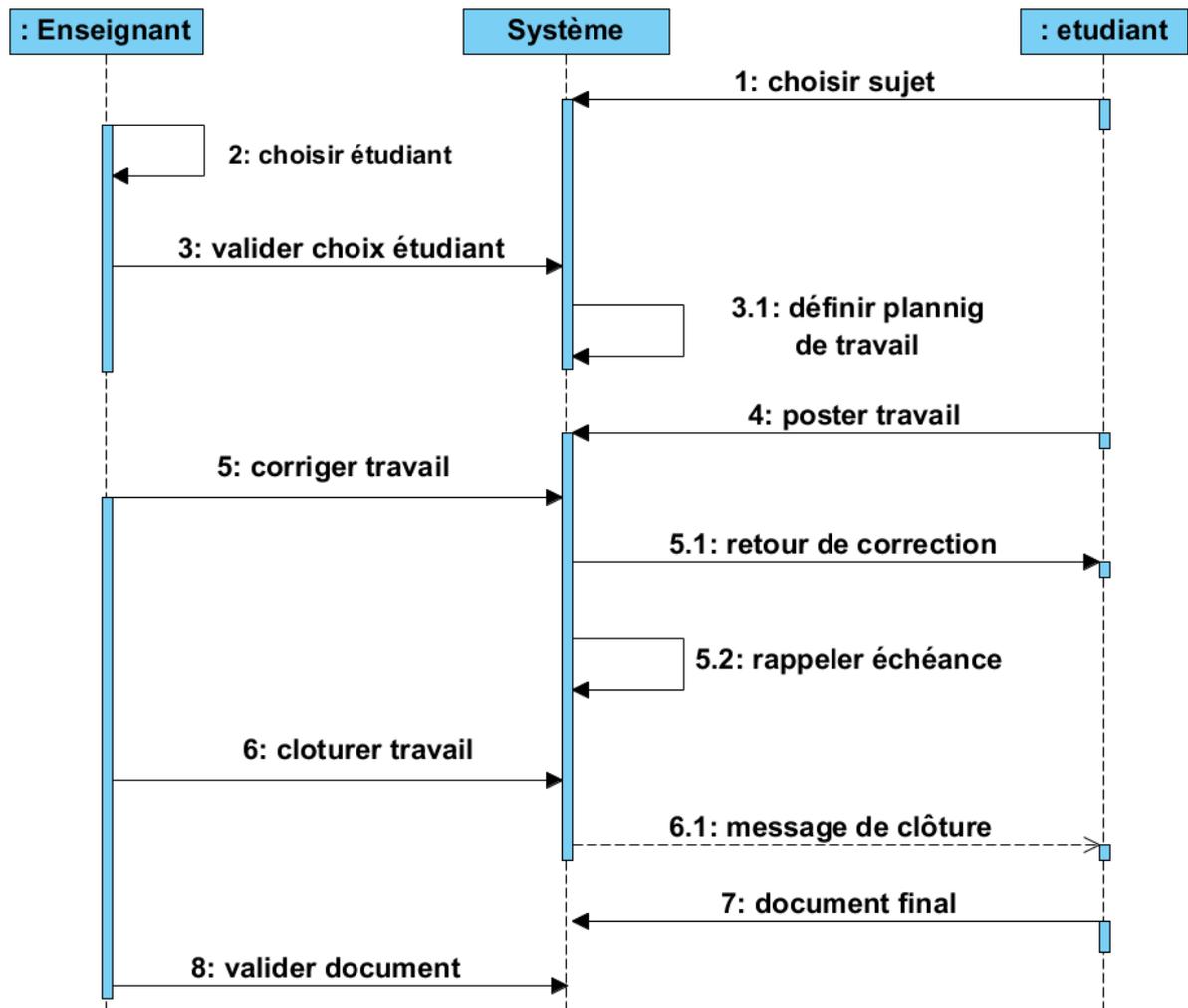


Figure 19 : Diagramme de séquence du cas d'utilisation "encadrer un mémoire"

## 3.3.5. Analyse de la rubrique entreprise

### a. Description du cas d'utilisation « rubrique entreprise »

Tableau 9 : Description de cas d'utilisation « rubrique entreprise »

| Description du cas d'utilisation « rubrique entreprise » |  |
|--|--|
| <b>Titre</b>   | Rubrique entreprise  |
| <b>Résumé</b>  | Permet au recruteur de pouvoir consulter le profil d'un étudiant |
| <b>Acteurs</b>   | Recruteur.   |
| <b>Pré condition</b>                                     | Être un recruteur  |

|                         |   |
|-------------------------|---|
| <b>Scénario nominal</b> | <ul style="list-style-type: none"> <li>Le recruteur demande à consulter le profil d'un étudiant</li> <li>Le système lui accorde la permission</li> <li>Le recruteur affiche le profil de l'étudiant</li> <li>Le recruteur peut lire le résumé du mémoire de l'étudiant</li> <li>Le recruteur peut consulter le document de mémoire de l'étudiant</li> </ul> |
| <b>Post condition</b>   | La page du profil de l'étudiant est affichée.   |
| <b>Exception</b>        | Le système refuse l'autorisation d'afficher le profil de l'étudiant   |

**b. Diagramme d'activité du cas d'utilisation « rubrique entreprise »**

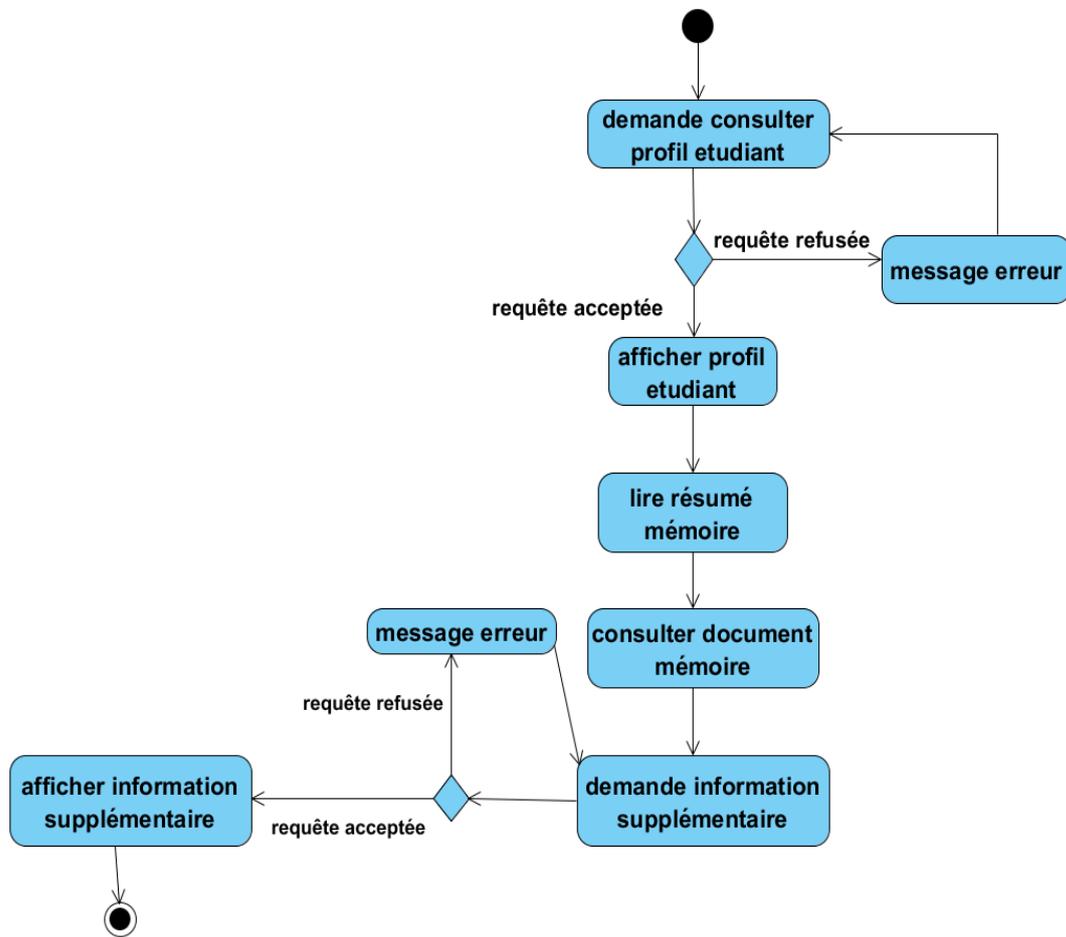


Figure 20 : Diagramme d'activité du cas d'utilisation "rubrique entreprise"

### c. Diagramme de séquence du cas d'utilisation « rubrique entreprise »

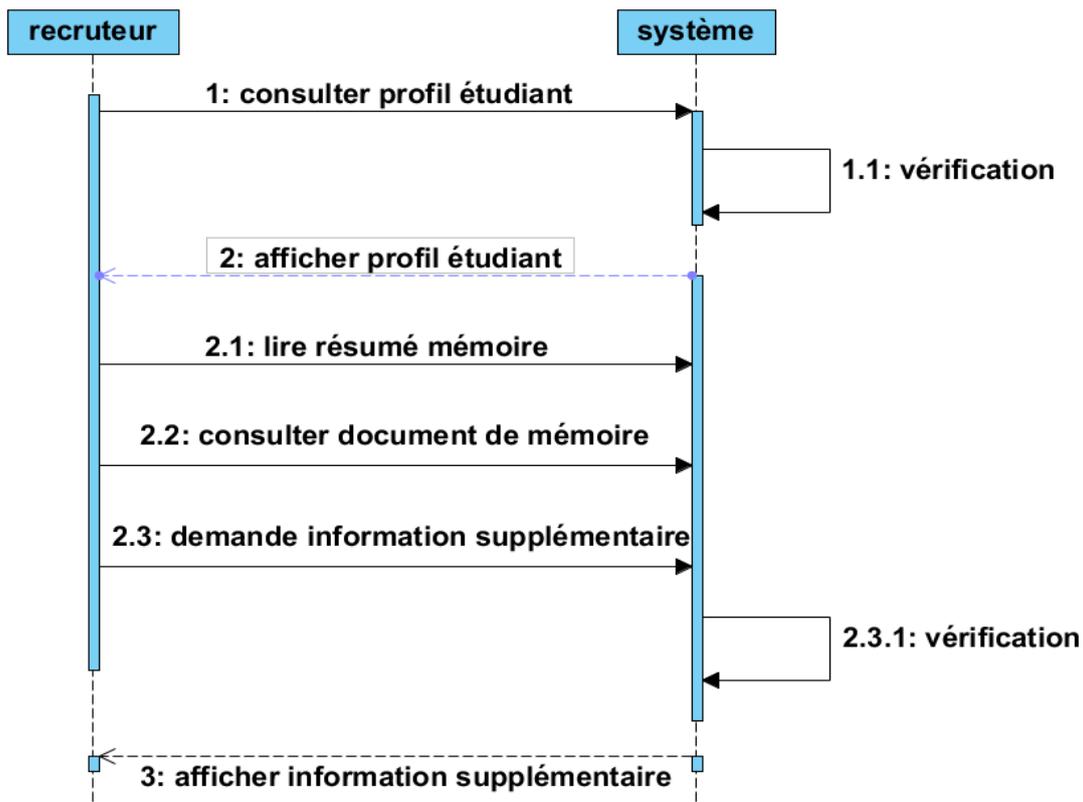


Figure 21 : Diagramme de séquence du cas d'utilisation "rubrique entreprise"

## 3.4. Besoins non fonctionnels

Un besoin non fonctionnel est une restriction ou une contrainte qui pèse sur un service du système, telles que les contraintes liées à l'environnement, à l'implémentation, les exigences en matière de performances, les dépendances du projet, de facilité de maintenance, d'extensibilité et de fiabilité.

Dans le cadre de ce travail, l'application devra répondre à ces besoins : contraintes techniques et contraintes ergonomiques.

### 3.4.1. Les contraintes techniques

- **La sécurité** : Tous les accès des utilisateurs (administrateur, enseignant ou étudiant) doivent être protégés par un login et un mot de passe pour parvenir à la sécurité de la plate-forme ;
- **Efficacité** : L'application doit être fonctionnelle indépendamment de toutes circonstances pouvant entourer l'utilisateur (une solution efficace) ;

- **Validité** : (correction, justesse, conformité), réaliser exactement les tâches définies dans la spécification ;
- **La performance** : Temps de réponse court c'est-à-dire quasi instantanément ;
- **La fiabilité** : Les données fournies par l'application doivent être fiables et la solution doit rendre des résultats corrects.

### 3.4.2. Les contraintes ergonomiques :

Pour notre application, nous devons obéir aux critères ergonomiques de Scapin et Bastien suivantes :

- ✓ **Guidage** : L'ensemble des moyens mis en œuvre pour conseiller, orienter, informer et conduire l'utilisateur lors de ses interactions avec l'ordinateur ;
- ✓ **Grouper/Distinction entre Items** : Groupement des différents éléments visuels de façon cohérente et ordonnée ;
- ✓ **Feedback immédiat** : Dans tous les cas, l'ordinateur doit répondre à l'utilisateur en fonction des actions et des requêtes de ce dernier ;
- ✓ **Brièveté** : Limiter le travail de lecture, d'entrée et les étapes par lesquelles doivent passer les utilisateurs ;
- ✓ **Adaptabilité** : Capacité à réagir selon le contexte et selon les besoins et les préférences des utilisateurs ;
- ✓ **Prise en compte de l'expérience de l'utilisateur** : Le système doit respecter le niveau d'expérience de l'utilisateur : il doit être simple, compréhensible et facile à utiliser ;
- ✓ **Correction des erreurs** : Mettre à la disposition des utilisateurs des moyens pour corriger leurs erreurs. Par exemple, fournir la possibilité de modifier les commandes lors de leur saisie ;
- ✓ **Comptabilité** : Il faut qu'il y ait accord entre les caractéristiques des utilisateurs et des tâches, d'une part, et l'organisation des sorties, des entrées et du dialogue d'une application donnée, d'autre part. Par exemple, les termes doivent être familiers aux utilisateurs, et relatifs à la tâche à réaliser.

## 3.5. Conclusion

Dans ce chapitre, nous avons procédé à la spécification et l'analyse des besoins fonctionnels à travers les cas d'utilisation de notre système. L'analyse de ces cas d'utilisation à travers les diagrammes de cas d'utilisation, de séquence et d'activité nous permet de mieux aborder la conception du système qui fera l'objet du chapitre suivant.

## CHAPITRE 4 : CONCEPTION DU SYSTEME

Suite à la spécification et l'analyse des besoins fonctionnels, nous entamons, dans ce chapitre, la conception du système.

Nous y parlons d'abord de la conception générale du système dans la section 4.1 ensuite nous aborderons la conception la conception détaillée dans la section 4.2.

### 4.1. Conception générale du système

Dans cette section, nous aborderons l'architecture ainsi que les diagrammes de composants, de packages et de déploiement du système.

#### 4.1.1. Architecture du système

##### a. L'architecture MVC

Lorsqu'on parle de Framework qui fournit une interface graphique à l'utilisateur (soit une page web, comme ici avec Django, soit l'interface d'une application graphique classique, comme celle de traitement de texte par exemple), nous parlons souvent de l'architecture MVC. Il s'agit d'un modèle distinguant plusieurs rôles précis d'une application, qui doivent être accomplis. Comme son nom l'indique, l'architecture (ou « patron ») Modèle-Vue-Contrôleur est composé de trois entités distinctes, chacune ayant son propre rôle à remplir.

Tout d'abord, le **modèle** représente une information enregistrée quelque part, le plus souvent dans une base de données. Il permet d'accéder à l'information, de la modifier, d'en ajouter une nouvelle, de vérifier que celle-ci correspond bien aux critères (on parle d'intégrité de l'information), de la mettre à jour, etc. Il s'agit d'une interface supplémentaire entre le code et la base de données, mais qui simplifie grandement les choses, comme nous le verrons par la suite.

Ensuite la **vue** qui est, comme son nom l'indique, la visualisation de l'information. C'est la seule chose que l'utilisateur peut voir. Non seulement elle sert à présenter une donnée, mais elle permet aussi de recueillir une éventuelle action de l'utilisateur (un clic sur un lien, ou la soumission d'un formulaire par exemple). Typiquement, un exemple de vue est une page web.

Finalement, le **contrôleur** prend en charge tous les événements de l'utilisateur (accès à une page, soumission d'un formulaire, etc.). Il se charge, en fonction de la requête de l'utilisateur, de récupérer les données voulues dans les modèles. Après un éventuel traitement sur des données, il transmet ces données à la vue, afin qu'elle s'occupe de les afficher. Lors de l'appel

d'une page, c'est le contrôleur qui est chargé en premier, afin de savoir ce qu'il est nécessaire d'afficher.

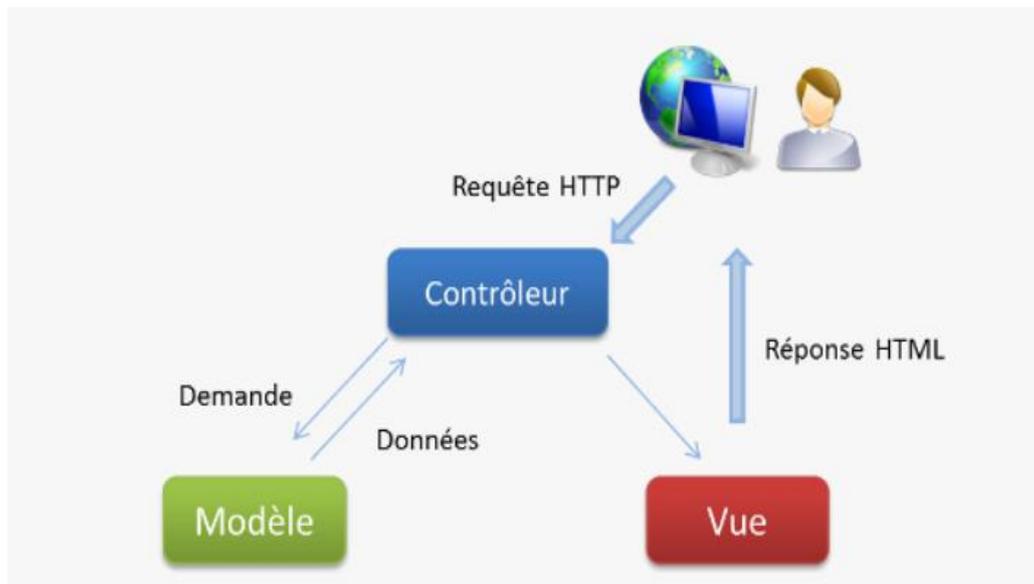


Figure 22 : Schéma de l'architecture MVC

### b. La spécificité de Django : le modèle MVT

L'architecture utilisée par Django diffère légèrement de l'architecture MVC classique. En effet, la « magie » de Django réside dans le fait qu'il gère lui-même la partie contrôleur (gestion des requêtes du client, des droits sur les actions...). Ainsi, nous parlons plutôt de Framework utilisant l'architecture **MVT : Modèle-Vue-Template**.

- Le **modèle** interagit avec une base de données via un ORM. Tous les modèles sont réunis dans un fichier python models.py.
- La **vue** reçoit une requête HTTP et renvoie une réponse HTTP convenable (par exemple si la requête est une interaction avec une base de données, la vue appelle un modèle pour récupérer les items demandés). Les vues se trouvent dans le fichier views.py
- Le **Template** est un fichier HTML récupéré par la vue et envoyé au visiteur avec les données des modèles.

La figure ci-dessous montre comment les différents composants de l'architecture MVT [22] de Django interagissent pour répondre à la requête d'un utilisateur. Ici le contrôleur ne correspond pas au contrôleur du MVC, mais à Django en lui-même qui gère en interne tout ce qui est liée au choix de la vue à laquelle envoyer la requête HTTP, ...

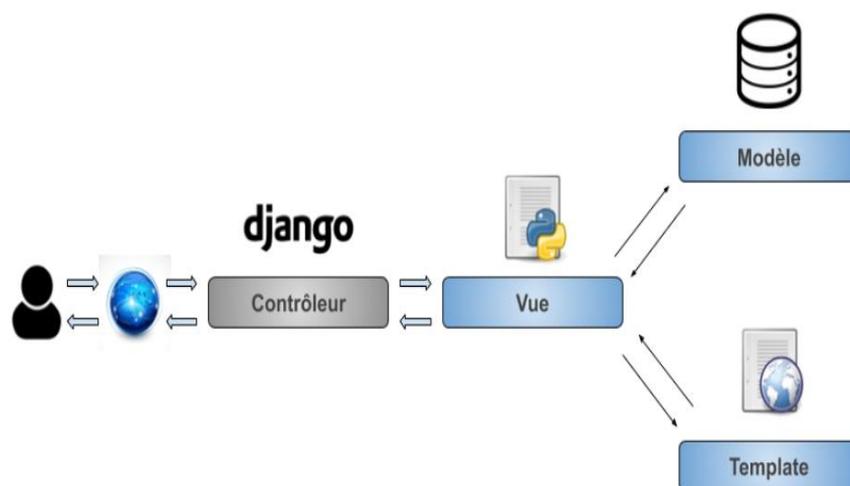


Figure 23 : Schéma de l'architecture MVT

### c. Différence entre les modèles de conception MVC et MVT :

Tableau 10 : Différence entre les modèles de conception MVC et MVT :

| S.NO. | CONTRÔLEUR DE VUE MODÈLE (MVC)   | MODÈLE DE VUE DE MODÈLE (MVT)   |
|-------|--|---|
| 1.    | MVC a un contrôleur qui pilote à la fois le modèle et la vue.                                  | MVT a des vues pour recevoir une requête HTTP et renvoyer une réponse HTTP. |
| 2.    | La vue indique comment les données utilisateur seront présentées.                              | Les modèles sont utilisés dans MVT à cette fin.                             |
| 3.    | Dans MVC, nous devons écrire tout le code spécifique au contrôle.                              | La partie contrôleur est gérée par le framework lui-même.                   |
| 4.    | Hautement couplé   | Couplage lâche  |
| 5.    | Les modifications sont difficiles  | Les modifications sont faciles  |
| 6.    | Convient pour le développement de grandes applications mais pas pour les petites applications. | Convient aux petites et grandes applications.                               |
| 7.    | Le flux est clairement défini donc facile à comprendre.  | Le flux est parfois plus difficile à comprendre que MVC.                    |
| 8.    | Cela n'implique pas de mappage d'URL.  | Le mappage de modèle d'URL a lieu.  |
| 9.    | Les exemples sont ASP.NET MVC, Spring MVC, etc.  | Django utilise le modèle MVT.   |

### En résumé

- Django respecte l'architecture MVT, directement inspirée du très populaire modèle MVC [20] ;

- Django gère de façon autonome la réception des requêtes et l'envoi des réponses au client (partie contrôleur) ;
- Un projet est divisé en plusieurs applications, ayant chacune un ensemble de vues, de modèles et de schémas d'URL ;
- Si elles sont bien conçues, ces applications sont réutilisables dans d'autres projets, puisque chaque application est indépendante.

#### **d. Architecture 3-tiers**

##### ➤ **Définition et concepts**

L'architecture trois tiers ou architecture à trois niveaux est l'application du modèle plus général qu'est le multi-tiers. L'architecture logique du système est divisée en trois niveaux ou couches :

- La **présentation** des données : correspondant à l'affichage, la restitution sur le poste de travail, le dialogue avec l'utilisateur ;
- Le **traitement** métier des données : correspondant à la mise en œuvre de l'ensemble des règles de gestion et de la logique applicative ;
- Et enfin l'**accès aux données** persistantes : correspondant aux données qui sont destinées à être conservées sur une durée ou de manière définitive (stockage sur un support physique).

Dans cette approche, les couches communiquent entre elles au travers d'un « modèle d'échange », et chacune d'entre elles propose un ensemble de services rendus. Les services d'une couche sont mis à disposition de la couche supérieure. On s'interdit par conséquent qu'une couche invoque les services d'une couche plus basse que la couche immédiatement inférieure ou plus haute que la couche immédiatement supérieure (chaque niveau ne communique qu'avec ses voisins immédiats).

Le rôle de chacune des couches et leur interface de communication étant bien définis, les fonctionnalités de chacune d'entre elles peuvent évoluer sans induire de changement dans les autres couches. Cependant, une nouvelle fonctionnalité de l'application peut avoir des répercussions dans plusieurs d'entre elles. Il est donc essentiel de définir un modèle d'échange assez souple, pour permettre une maintenance aisée de l'application. Ce modèle d'architecture 3-tiers a pour objectif de répondre aux préoccupations suivantes :

- Allègement du poste de travail client (notamment vis à vis des architectures classiques client-serveur de données - typiques des applications dans un contexte Postgre/Unix) ;

- Prise en compte de l'heterogeneite des plates-formes (serveurs, clients, langages, etc.) ;
- Introduction de clients dits « legers » (plus liee aux technologies Intranet/HTML qu'au 3-tiers proprement dit) ;
- Et enfin, meilleure repartition de la charge entre differents serveurs d'application.

➤ **Les trois (03) couches**

- **Couche Présentation (premier niveau) :** Elle correspond à la partie de l'application visible et interactive avec les utilisateurs. On parle alors d'Interaction Homme-Machine. En informatique, elle peut être réalisée par une application graphique ou textuelle. Elle peut aussi être représentée en HTML pour être exploitée par un navigateur Web. La couche présentation relaie les requêtes de l'utilisateur à destination de la couche métier, et en retour lui présente les informations renvoyées par les traitements de cette couche. Il s'agit donc ici d'un assemblage de services métiers et applicatifs offerts par la couche inférieure.
- **Couche Métier / Business (second niveau) :** Elle correspond à la partie fonctionnelle de l'application, celle qui implémente la « logique », et qui décrit les opérations que l'application va appliquer sur les données en fonction des requêtes des utilisateurs, effectuées au travers de la couche présentation. Les différentes règles de gestion et de contrôle du système sont mises en œuvre dans cette couche. La couche métier offre des services applicatifs et métiers à la couche présentation. Pour fournir ces services, elle s'appuie, le cas échéant, sur les données du système, accessibles au travers des services de la couche inférieure. En retour, elle renvoie à la couche présentation les résultats qu'elle a calculés.
- **Couche d'accès aux données (troisième niveau) :** Elle consiste en la partie gérant l'accès aux gisements de données du système. Ces données sont pérennes, car destinées à durer dans le temps, de manière plus ou moins longue ou définitive (stockées sur un support physique). Les données peuvent être stockées indifféremment dans de simples fichiers texte, ou eXtensible Markup Language (XML), ou encore dans une base de données. Quel que soit le support de

stockage choisi, l'accès aux données doit être le même. Cette abstraction améliore la maintenance du système.

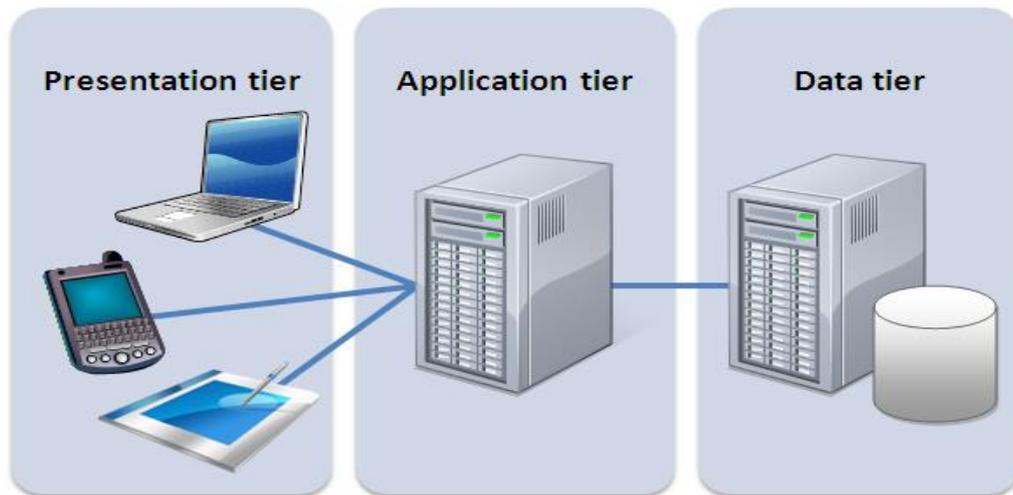


Figure 24 : Architecture 3-tiers

L'architecture de notre système est représentée par la figure ci-dessous :

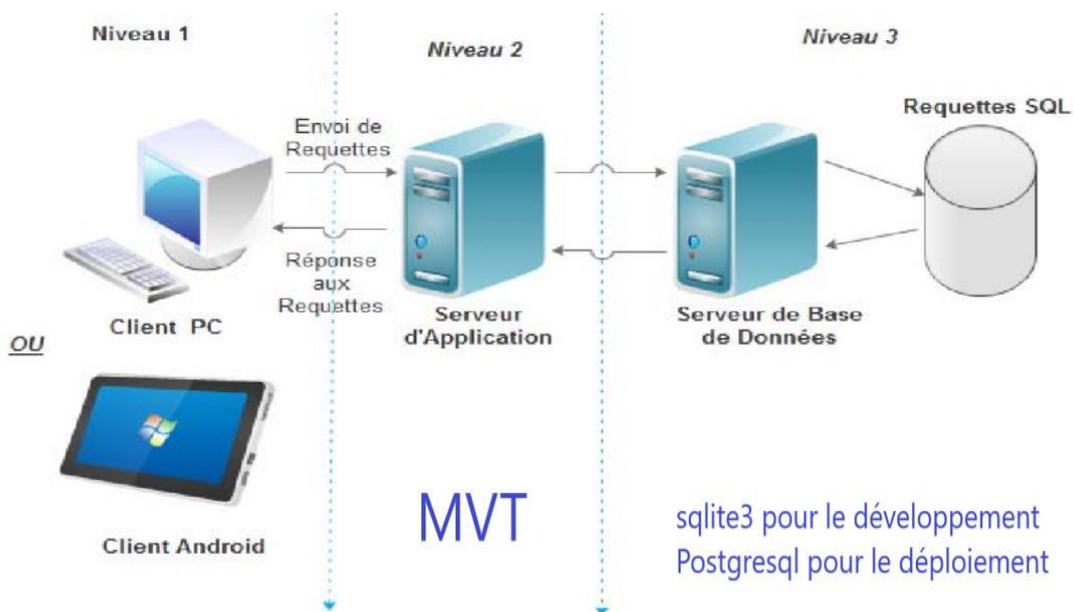


Figure 25 : Architecture de l'application

### 4.1.2. Diagramme de composants

Le diagramme de composants décrit l'organisation du système du point de vue des éléments logiciels comme les modules (paquetages, fichiers sources, bibliothèques, exécutables), des données (fichiers, bases de données) ou encore d'éléments de configuration (paramètres, scripts, fichiers de commandes). Ce diagramme permet de mettre en évidence les dépendances entre les

composants (qui utilise quoi). La figure 26 représente le diagramme de composants de notre système.

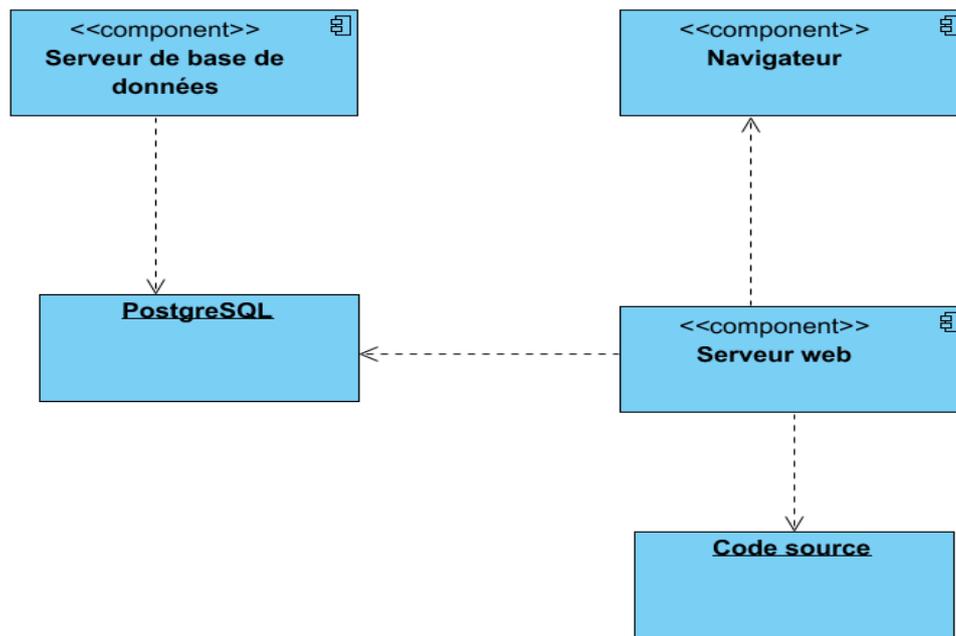


Figure 26 : diagramme de composant du système

### 4.1.3. Diagramme de packages

Les diagrammes de package [33] (ou diagramme de paquetages en français) sont des diagrammes structurels utilisés pour représenter l'organisation et la disposition de divers éléments modélisés sous forme de paquetages. Un paquetage est un regroupement d'éléments UML apparentés, tels que des diagrammes, des documents, des classes ou même d'autres paquetages. Tous les éléments du diagramme sont imbriqués dans des paquetages, qui sont eux-mêmes représentés sous forme de dossiers de fichiers et organisés de manière hiérarchique. Les diagrammes de paquetages sont le plus souvent utilisés pour donner un aperçu visuel de l'architecture en couches d'un classifieur UML, tel qu'un système logiciel. La Figure 27 représente le diagramme de packages [33] de notre système.

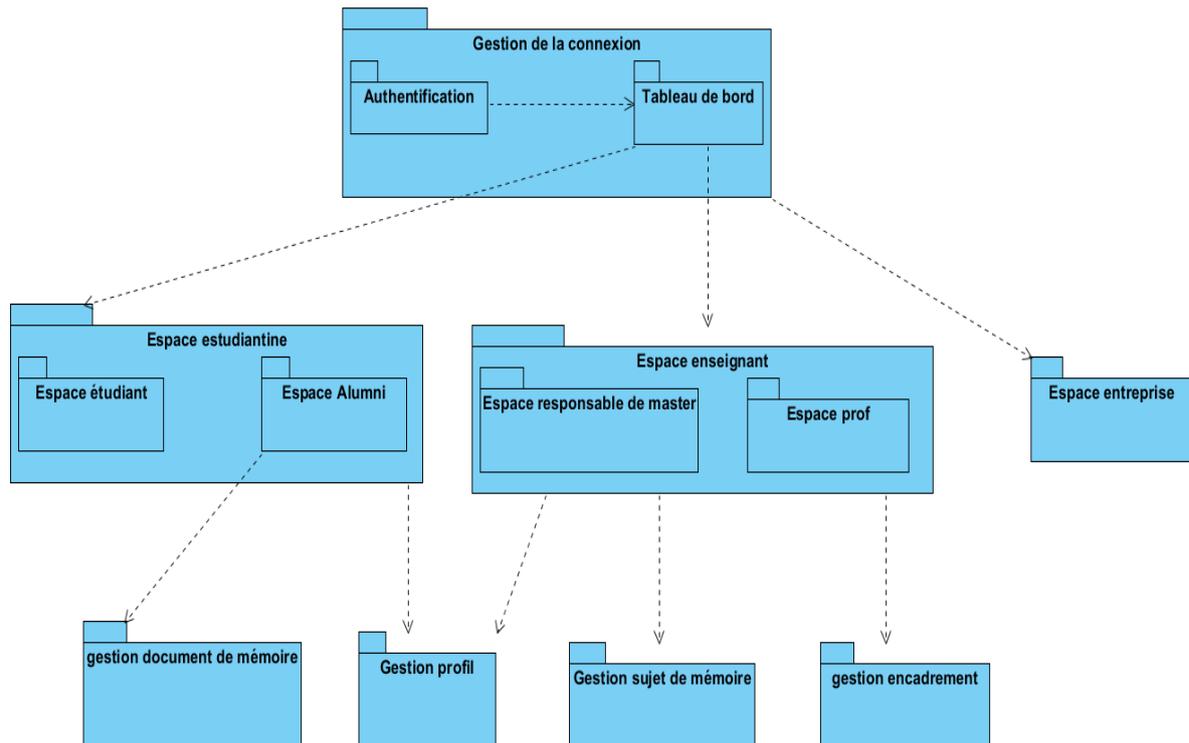


Figure 27 : diagramme de package du systeme

Notre diagramme de package est compose de 8 packages a savoir :

- **La gestion de la connexion** : ce package gere la connexion en tenant compte du profil de chaque utilisateur en le connectant dans sa session ;
- **L'espace etudiantine** : ce package permet de gerer l'ensemble des etudiants et des alumns ;
- **L'espace des enseignants** : ce package permet de gerer l'ensemble des enseignants ainsi que le responsable de master ;
- **L'espace entreprise** : Ce package gere l'interaction des entreprises avec le systeme ;
- **Gestion document de memoire** : ce package gere les documents de memoire des alumns ;
- **Gestion profil** : ce package gere la modification du profil pour l'ensemble des utilisateurs du systeme ;
- **Gestion sujet de memoire** : ce package gere l'ajout des sujets de memoire dans le systeme ;
- **Gestion encadrement** : ce package gere l'ensemble des taches ayant un rapport avec l'encadrement et le suivi d'un memoire de master 2.

#### 4.1.4. Diagramme de déploiement

En UML, un diagramme de déploiement est une vue statique qui sert à représenter l'utilisation de l'infrastructure physique par le système et la manière dont les composants du système sont répartis ainsi que les relations entre eux. Les éléments utilisés par un diagramme de déploiement sont principalement les nœuds, les composants, les associations et les artefacts [37]. Les caractéristiques des ressources matérielles physiques et des supports de communication peuvent être précisées par stéréotype. La Figure 28 illustre le diagramme de déploiement de notre système.

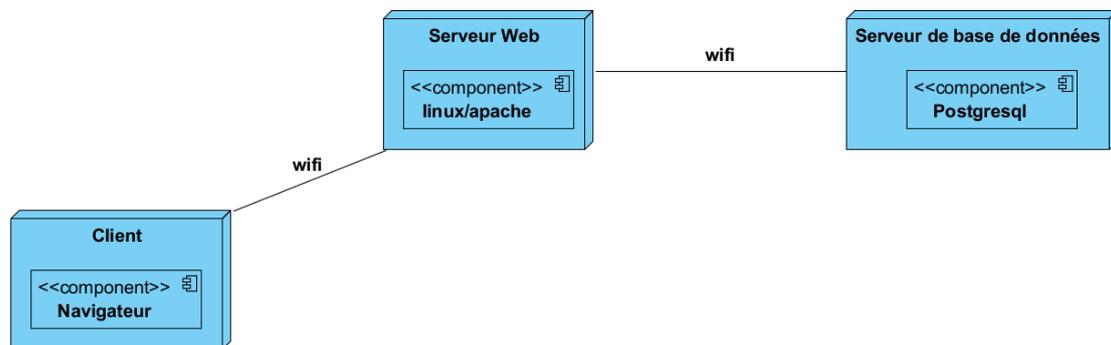


Figure 28 : Diagramme de déploiement du système

#### 4.1.5. Architecture réseau du système

L'architecture de réseau est l'organisation d'équipements de transmission, de logiciels, de protocoles de communication et d'infrastructure filaire ou radioélectrique permettant la transmission des données entre les différents composants.

##### Utilité des réseaux informatiques :

- Permettre le partage des ressources : numériques (fichiers) ou physiques (une imprimante)
- Accroître la résistance aux pannes
- Diminuer les coûts
- Accéder à des services à distance : bases de données, logiciels
- Communication : email...

Nous avons pu élaborer l'architecture réseau pour permettre à notre application d'être incorporée dans le réseau sans difficulté. La figure 29 illustre l'architecture du réseau.

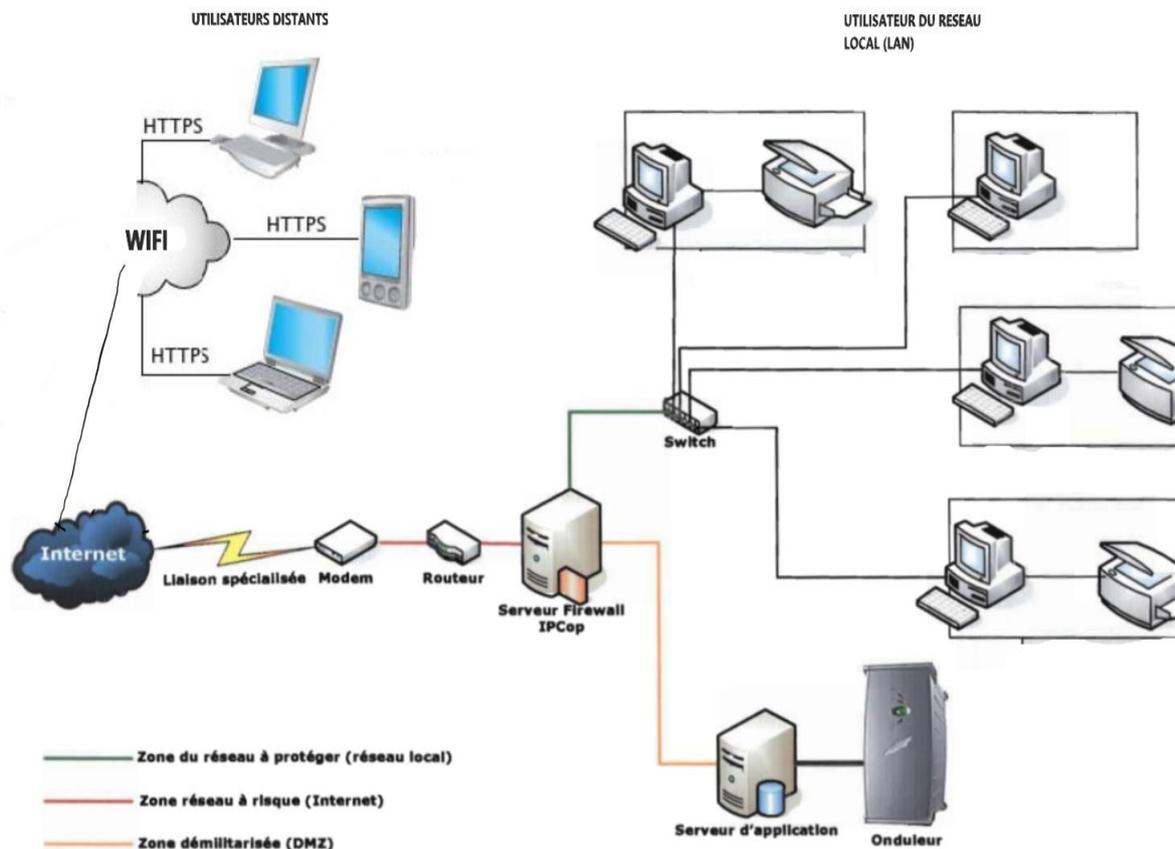


Figure 29 : Architecture réseau du système

L'architecture du réseau nous montre que l'application sera accessible de n'importe où et avec les terminaux les plus couramment utilisés (ordinateur portable, ordinateur de bureau, smartphone et tablette). Ainsi on aura deux types de connexions distincts :

- **Connexion à partir du réseau local (LAN)** : Cette connexion se fera à l'intérieur du réseau de l'université
- **Connexion à partir d'un réseau distant** : cette connexion se fait n'importe où dans le monde en accédant au nom de domaine de l'application et en ayant un navigateur qui prend en charge la connexion HTTPS.

## 4.2. Conception détaillée du système

La conception détaillée est la dernière phase dans la conception de notre système. Elle consiste à élaborer les différentes classes qui interviendront dans sa phase de développement (ou codage). Pour cela, nous abordons les différentes classes, le diagramme de classes et le dictionnaire de données du système.

### 4.2.1. Dictionnaire de données

Un dictionnaire des données est une collection de métadonnées ou de données de référence nécessaire à la conception d'une base de données relationnelle [39]. Il revêt une importance stratégique particulière, car il est le vocabulaire commun de l'organisation. Il décrit des données aussi importantes que les clients, les nomenclatures de produits et de services, les annuaires, etc. C'est donc le référentiel principal du système, sur lequel s'appuient les décisions de celui-ci. Il est souvent représenté par un tableau à quatre colonnes contenant le nom, le code et le type de donnée ainsi que des commentaires.

Un dictionnaire des données doit respecter les contraintes suivantes.

- Tous les noms doivent être monovalués et non décomposables.
- Il ne doit pas y avoir d'homonymes, ni de synonymes.
- Les données y sont regroupées par entité.
- Les identifiants sont complètement précisés.
- Les commentaires doivent être pertinents.

*Le dictionnaire de données de notre système est consultable à l'annexe 3 du document*

### 4.2.2. Diagramme de classe du système

Une classe est une description d'un groupe d'objets partageant un ensemble commun de propriétés (les attributs), de comportements (les opérations ou méthodes) et de relations avec d'autres objets (les associations et les agrégations). Une classe de conception est composée par des attributs (données de la classe) et des méthodes (comportement de la classe). Le diagramme de classes est un schéma utilisé pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Dans cette section, nous allons décrire le diagramme de classes de notre système.

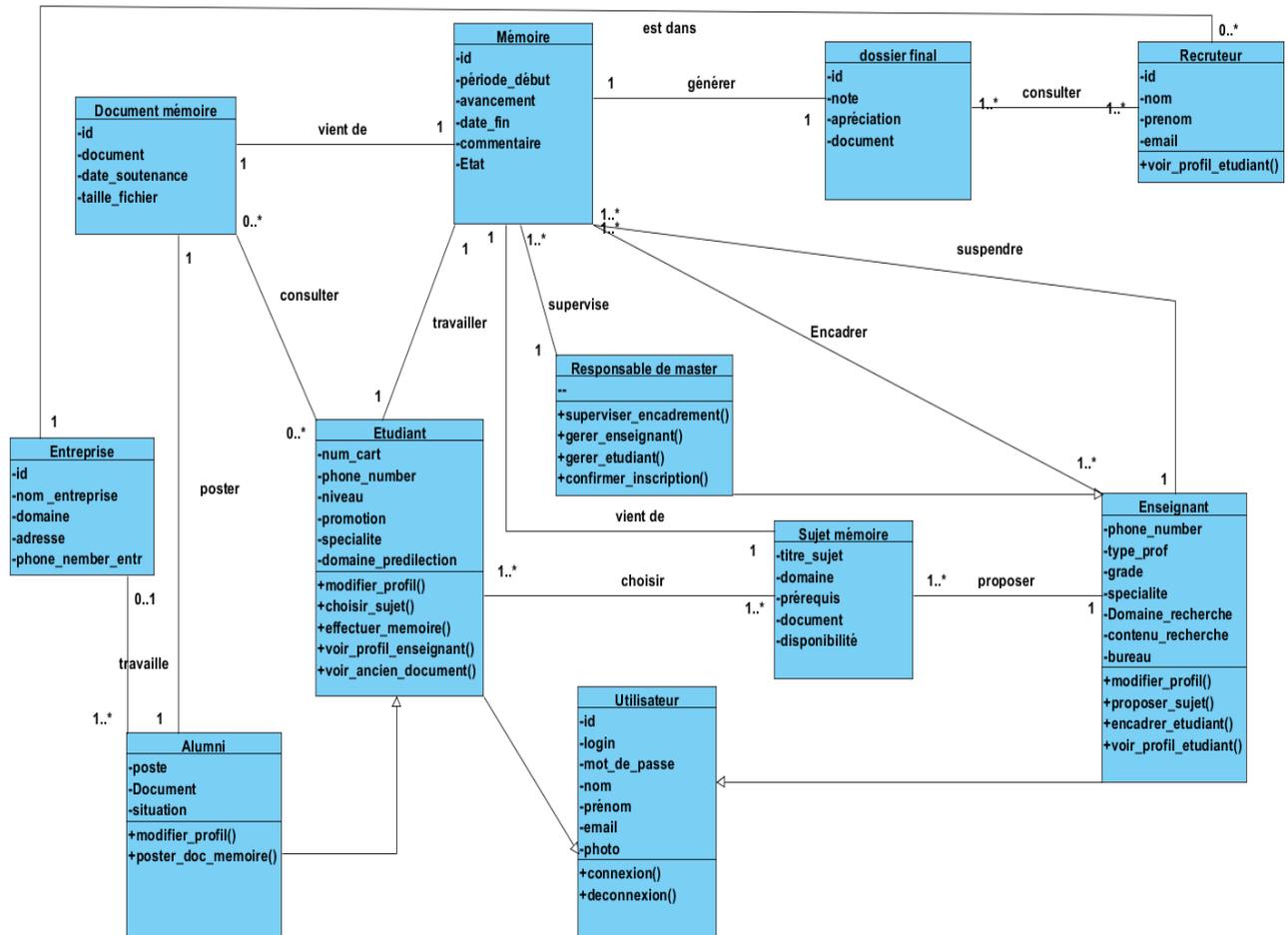


Figure 30 : Diagramme de classe du système

Nous avons onze (11) classes participantes dans notre système. Les classes étudiant et enseignant héritent de la classe utilisateur. La classe responsable master hérite de la classe enseignant car le responsable du master est avant tout un enseignant qui a des tâches supplémentaires. La classe alumni (ancien étudiant) hérite de la classe étudiant.

Un enseignant peut ajouter un sujet de mémoire qui peut être choisi par plusieurs étudiants. L'enseignant valide le choix d'un étudiant parmi ceux qui ont choisi son sujet. Un mémoire est fait par un étudiant et encadré par un ou plusieurs enseignants. A la fin du mémoire le profil est mis à jour et visible par de potentiels recruteurs

Un document de mémoire est posté par un alumni (ancien étudiant) après validation par son encadrant. Le document est consultable et téléchargeable par un étudiant, un enseignant, un recruteur, etc.

### **4.3. Conclusion**

Dans ce chapitre, nous avons pu faire la conception du système. Nous avons étudié la conception générale notamment l'architecture du système, l'architecture réseau du système et les diagrammes de composant, de package et de déploiement. Ensuite on a étudié la conception détaillée qui consistait à faire le diagramme de classe et le dictionnaire des données du système. La phase de conception est succédée par l'implémentation du système. Ce qui fera l'objet du chapitre 5.

# CHAPITRE 5 : IMPLEMENTATION ET PRESENTATION DE L'APPLICATION

Dans ce chapitre, nous parlons de l'implémentation et la présentation de l'application qui est la suite logique du chapitre précédent ou on a eu à faire la conception du système. Ainsi dans ce chapitre, on va analyser l'implémentation de l'application dans la section 5.1. Dans cette section, on présente les différents outils utilisés pour le développement ou encore les serveurs utilisés. Pour terminer on va présenter dans la section 5.2 l'application réalisée.

## 5.1. Implémentation de l'application

### 5.1.1. Outils de conception et de développement du système

#### a. Outils de conception : visual paradigm

Visual Paradigm for UML est, comme son nom le laisse supposer, un logiciel permettant aux programmeurs de mettre en place des diagrammes UML. Disposant d'un outil créant des rapports personnalisables aux formats PDF, Word ou HTML afin de les partager et les publier sur Internet, cette application est compatible avec de nombreuses applications, standards et environnements. Ainsi, il est possible de générer notamment des diagrammes de séquences ou de cas d'utilisation et ainsi produire du code source dans de nombreux langages comme le Java ou encore le C++, ou bien faire l'inverse, générer des diagrammes à partir de code déjà existant.

#### b. Environnement de développement : Visual Studio Code

Visual Studio Code [29] est un éditeur de code extensible développé par Microsoft pour Windows, Linux et MacOS. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré. Les utilisateurs peuvent modifier le thème, les raccourcis clavier, les préférences et installer des extensions qui ajoutent des fonctionnalités supplémentaires. Le code source de Visual Studio Code provient du projet logiciel libre et open source VSCode de Microsoft publié sous la licence MIT permissive. Cependant, les binaires compilés sont des logiciels gratuits pour toute utilisation.

#### c. Outil de versionning : git

Git est de loin le système de contrôle de version le plus largement utilisé aujourd'hui [34]. Git est un projet open source avancé, qui est activement maintenu. À l'origine, il a été développé en 2005 par Linus Torvalds, le créateur bien connu du noyau du système d'exploitation Linux.

De plus en plus de projets de développement reposent sur Git pour le contrôle de version, y compris des projets commerciaux et/ou open source. Les développeurs qui utilisent Git sont bien représentés dans le pool de talents disponibles, et la solution fonctionne bien sur une vaste gamme de systèmes d'exploitation et d'environnements de développement intégrés (IDE).

### **5.1.2. Serveurs et base de données**

#### **a. Serveur web : serveur local de Django**

Une application web nécessite un serveur web local pour permettre de lancer l'exécution du code en vue d'avoir un résultat sur le rendu du site. On peut citer WampServer comme Exemple de serveur web. Cependant pour gérer l'exécution de ces applications, Django utilise un serveur web local rapide et performant qui permettra d'exécuter les applications de Django.

#### **b. Base de données : SQLite et PostgreSQL**

Django est un Framework extrêmement souple qui embarque par défaut un fichier de base de données SQLite qui nous convient largement pour la phase de développement car . Ce type de configuration fonctionne parfaitement pour des applications peu gourmandes en ressource. Toutefois, utiliser un autre système de stockage peut aider à augmenter les performances en production.

Django tente d'activer autant de fonctionnalités que possible sur tous les types de base de données. Cependant, tous les types de base de données ne sont pas semblables, et nous avons dû prendre des décisions de conception sur les fonctionnalités à activer et les hypothèses sur lesquelles nous pouvions nous baser en toute sécurité. C'est ainsi qu'on a choisi de travailler avec PostgreSQL qui répond naturellement à tous les critères définis.

##### **i. SQLite**

SQLite est une bibliothèque en langage C qui implémente un petit, rapide, autonome, haute fiabilité, et complet, moteur de base de données SQL. SQLite est le moteur de base de données le plus utilisé au monde [35]. SQLite est intégré à tous les téléphones mobiles et à la plupart des ordinateurs et est intégré à d'innombrables autres applications que les gens utilisent chaque jour.

Le format de fichier SQLite est stable, multiplateforme et rétro compatible et les développeurs s'engagent à le maintenir ainsi tout au long jusqu'à l'an 2050. Les fichiers de base de données SQLite sont couramment utilisés comme conteneurs pour transférer un contenu riche entre les systèmes et comme format d'archivage à long terme pour les données. Il existe plus de 1 milliard de bases de données SQLite en cours d'utilisation [35].

## ii. PostgreSQL

PostgreSQL est un système de gestion de base de données relationnelle et objet (SGBDRO). C'est un outil libre disponible selon les termes d'une licence de type BSD. Ce système est un concurrent à d'autres systèmes de gestion de base de données, qu'ils soient libres (comme MariaDB et Firebird), ou propriétaires (comme Oracle, MySQL, Sybase, DB2, Informix et Microsoft SQL Server). Comme les projets libres Apache et Linux, PostgreSQL [28] n'est pas contrôlé par une seule entreprise, mais est fondé sur une communauté mondiale de développeurs et d'entreprises.

### 5.1.3. Technologies utilisées pour la partie Front-End

#### a. Html 5

Le HyperText Markup Language, généralement abrégé HTML ou dans sa dernière version HTML5, est le langage de balisage conçu pour représenter les pages web. Ce langage permet :

- D'écrire de l'hypertexte, d'où son nom ;
- De structurer sémantiquement des pages ;
- De mettre en forme le contenu d'une page ;
- De créer des formulaires de saisie ;
- D'inclure des ressources multimédias dont des images, des vidéos, et des programmes informatiques ;
- De créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du web.

#### b. CSS 3

Les feuilles de style en cascade<sup>1</sup>, généralement appelées CSS de l'anglais Cascading Style Sheets, forment un langage informatique qui décrit la présentation des documents HTML et XML. Les standards définissant CSS sont publiés par le World Wide Web Consortium (W3C). Introduit au milieu des années 1990, CSS devient couramment utilisé dans la conception de sites web et bien pris en charge par les navigateurs web dans les années 2000.

#### c. Bootstrap (version 4)

Bootstrap est une collection d'outils utiles à la création du design (graphisme, animation et interactions avec la page dans le navigateur, etc.) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. C'est l'un des projets les plus populaires sur la plate-forme de gestion de développement GitHub.

#### **d. JQuery**

JQuery [24] est une bibliothèque JavaScript rapide, petite et riche en fonctionnalités. Il simplifie beaucoup la traversée et la manipulation de documents HTML, la gestion des événements, l'animation et Ajax avec une API facile à utiliser qui fonctionne sur une multitude de navigateurs. Avec une combinaison de polyvalence et d'extensibilité, jQuery a changé la façon dont des millions de personnes écrivent du JavaScript.

#### **e. React js**

React (aussi appelé **React.js** ou **ReactJS**) est une bibliothèque JavaScript libre développée par Facebook depuis 2013. Le but principal de cette bibliothèque est de faciliter la création d'application web monopage, via la création de composants dépendant d'un état et générant une page (ou portion) HTML à chaque changement d'état.

React est une bibliothèque qui ne gère que l'interface de l'application, considéré comme la vue dans le modèle MVC. Elle peut ainsi être utilisée avec une autre bibliothèque ou un framework MVC comme AngularJS. La bibliothèque se démarque de ses concurrents par sa flexibilité et ses performances, en travaillant avec un DOM virtuel et en ne mettant à jour le rendu dans le navigateur qu'en cas de nécessité.

### **5.1.4. Technologies utilisées pour la partie Back-End**

#### **a. Langage de programmation : python (version 3.8.5)**

Python est un langage de programmation interprété, multiparadigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions ; il est ainsi similaire à Perl, Ruby, Scheme, Smalltalk et Tcl.

Le langage Python est placé sous une licence libre proche de la licence BSD4 et fonctionne sur la plupart des plates-formes informatiques, des smartphones aux ordinateurs centraux, de Windows à Unix avec notamment GNU/Linux en passant par MacOS, ou encore Android, iOS, et peut aussi être traduit en Java ou .NET. Il est conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser.

Il est également apprécié par certains pédagogues qui y trouvent un langage où la syntaxe, clairement séparée des mécanismes de bas niveau, permet une initiation aisée aux concepts de base de la programmation.

### **b. Framework Django (version 3.1.1)**

Django est un Framework Web Python de haut niveau qui encourage un développement rapide et une conception propre et pragmatique. Conçu par des développeurs expérimentés, il prend en charge une grande partie des tracas du développement Web, on peut donc se concentrer sur l'écriture de l'application sans avoir à réinventer la roue. C'est gratuit et open source.

- **Très rapide** : Django a été conçu pour aider les développeurs à faire passer les applications de la conception à la réalisation le plus rapidement possible.
- **Sécurisé de manière rassurante** : Django prend la sécurité au sérieux et aide les développeurs à éviter de nombreuses erreurs de sécurité courantes.
- **Extrêmement évolutif** : Certains des sites les plus fréquentés du Web tirent parti de la capacité de Django à évoluer rapidement et de manière flexible.

### **c. Memcached**

Memcached est un Système de mise en cache d'objets mémoire distribué gratuit et open source de haute performance. De nature générique, il est destiné à être utilisé pour accélérer les applications Web dynamiques en allégeant la charge de la base de données. Memcached est un magasin de valeurs-clés en mémoire pour de petits morceaux de données arbitraires (chaînes, objets) à partir des résultats d'appels de base de données, d'appels d'API ou de rendu de page. Memcached est simple mais puissant. Sa conception simple favorise un déploiement rapide, une facilité de développement et résout de nombreux problèmes auxquels sont confrontés les grands caches de données. Son API est disponible pour les langues les plus courantes.

### **d. RabbitMQ**

RabbitMQ est basé sur l'idée du Advanced Message Queuing Protocol (AMQP). Le principal avantage du AMQP est qu'il n'impose pas à l'émetteur et au destinataire de comprendre le même langage de programmation. Le message broker s'est aujourd'hui quelque peu détaché du AMQP et fonctionne également avec des protocoles de messages comme STOMP ou MQTT grâce à des plugins, mais l'idée reste la même : le producteur et le destinataire du message sont séparés par une file d'attente dans laquelle les messages sont stockés temporairement.

L'avantage de RabbitMQ réside dans le fait que le producteur du message n'a pas à effectuer personnellement l'envoi. Le message broker reçoit le message et donne ainsi la possibilité au producteur de commencer une nouvelle tâche. L'émetteur n'est pas obligé d'attendre que le destinataire reçoive le message. Dans le cadre de cette procédure, le message est placé dans la

file d'attente avant d'être récupéré par le consommateur. À cet instant, l'émetteur est déjà occupé à une nouvelle tâche. Il s'agit donc d'une procédure asynchrone : l'émetteur et le destinataire n'ont pas à agir au même rythme.

### 5.1.5. L'arborescence de l'application

L'organisation du projet en package permet de mieux scinder les sous parties du projet. Elle permet d'avoir une meilleure lisibilité et une meilleure approche de l'architecture 3 tiers tout en respectant le modèle MVC et de surcroit le modèle MVT. Les packages permettent d'avoir un code indépendant et assure la réutilisabilité du code sur d'autres projets. Django gère seul la division du code en considérant une fonctionnalité comme une application. la figure 31 illustre l'organisation de notre système

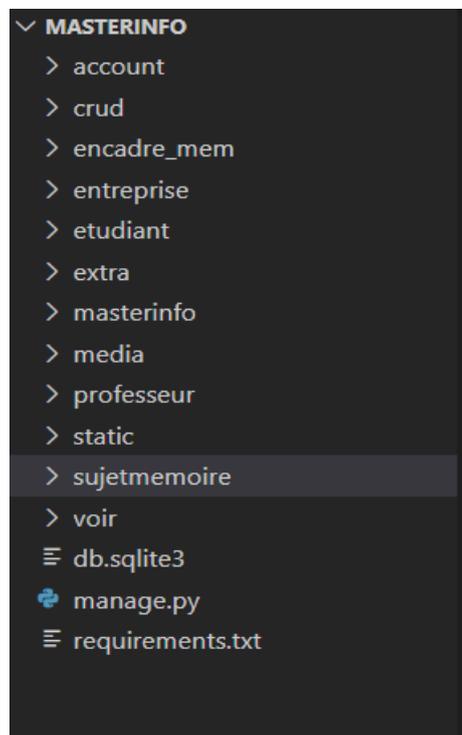


Figure 31 : structuration du projet sous Django

La figure 31 montre la structuration de notre système.

- **Requirement.txt** contient l'ensemble des dépendances du projet installés à l'aide de l'utilitaire pip.
- **Manage.py** est utilisé pour créer des applications, travailler avec des bases de données et démarrer le serveur Web de développement.

- **Db.sqlite3** est notre base de données embarquée par défaut par django
- Le terme **Application** décrit un paquet Python qui fournit un certain ensemble de fonctionnalités. Les applications peuvent être réutilisées dans différents projets. Cependant on distingue deux types d'application qui sont l'application principale du projet et les autres applications

#### a. L'application principale du projet (masterinfo)

L'application qui porte le même nom que le projet (ici masterinfo) est l'application principale du projet. Il permet de configurer le projet. La figure 32 illustre l'application principale masterinfo du projet.

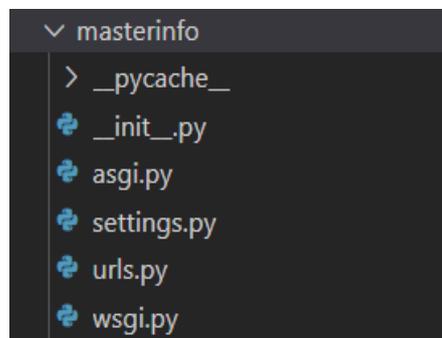


Figure 32 : application principale du projet

- **\_\_init\_\_.py** est un fichier vide qui demande à Python de traiter ce répertoire comme un package Python.
- **settings.py** contient tous les paramètres du site Web, y compris l'enregistrement des applications que nous créons, l'emplacement de nos fichiers statiques, les détails de configuration de la base de données, etc.
- **urls.py** définit les mappages URL-à-afficher du site. Bien que cela puisse contenir *tout* le code de mappage d'URL, il est plus courant de déléguer certains des mappages à des applications particulières, comme on le voit plus tard.
- **wsgi.py** est utilisé pour aider notre application Django à communiquer avec le serveur Web.
- **asgi.py** est une norme permettant aux applications Web et aux serveurs asynchrones Python de communiquer entre eux. ASGI est le successeur asynchrone de WSGI et

fournit une norme pour les applications Python asynchrones et synchrones (alors que WSGI a fourni une norme pour les applications synchrones uniquement). Il est rétro compatible avec WSGI et prend en charge plusieurs serveurs et cadres d'application.

### b. Les autres applications

Sur la figure 31 la liste allant de account à voir constituent des applications. Ces applications sont des packages regroupant un ensemble de fonctionnalité communes à une tâche. Ma figure 33 illustre l'application qui gère les comptes d'utilisateur account.

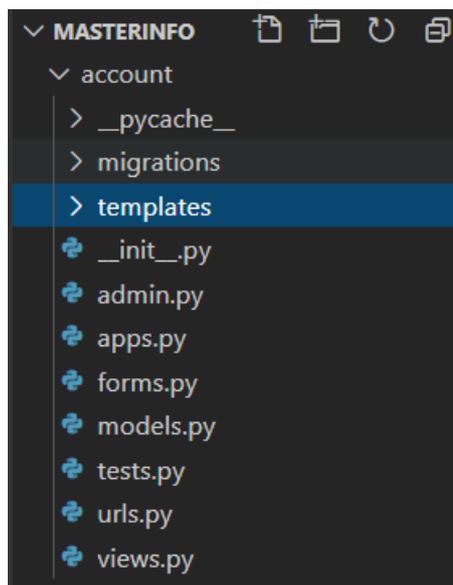


Figure 33 : l'application account

On remarque que par rapport à une application principale une application simple a les fichiers **forms.py**, **models.py**, **admin.py** et **views.py** en plus et le dossier **migrations** qui sera utilisé par django pour gérer les migrations et les modifications progressives apportées à la base de données quand des modifications seront faites dans les fichiers models.py.

- **Forms.py** : est le fichier qui permet de gérer les champs de formulaires.

```
account > forms.py > ...  
  
1  from django.contrib.auth.forms import UserCreationForm  
2  from django import forms  
3  from django.db import transaction  
4  from .models import User, Etudiant, Prof, Alumni  
5  from django.forms import MultiWidget, TextInput  
6  
7  class EtudiantSignUpForm(UserCreationForm):  
8      username = forms.CharField(label='Nom d\'utilisateur',  
9                                 widget=forms.TextInput)  
10     first_name = forms.CharField(label='Prénom',  
11                                 widget=forms.TextInput)  
12     last_name = forms.CharField(label='Nom',  
13                                 widget=forms.TextInput)  
14     password2 = forms.CharField(label='Confirmer votre mot de passe',  
15                                 widget=forms.PasswordInput)  
16  
17     class Meta(UserCreationForm.Meta):  
18         model = User  
19         fields = ('username', 'first_name', 'last_name', 'email')  
20  
21     @transaction.atomic  
22     def save(self):  
23         user = super().save(commit=False)  
24         user.is_etudiant = True  
25         user.save()
```

Figure 34 : fichier forms.py

- **Models.py** Un modèle est la source d'information unique et définitive à propos des données. Il contient les champs et le comportement essentiels des données que vous stockez. Généralement, chaque modèle correspond à une seule table de base de données.

```
account > models.py > ...  
  
1  from django.db import models  
2  from django.contrib.auth.models import AbstractUser  
3  from django.urls import reverse  
4  
5  
6  
7  
8  class User(AbstractUser):  
9      is_etudiant = models.BooleanField(default=False)  
10     is_prof = models.BooleanField(default=False)  
11     first_name = models.CharField(max_length=100, blank=True)  
12     last_name = models.CharField(max_length=100, blank=True)  
13  
14     class Etudiant(models.Model):  
15         CHOICE = (  
16             (u"Master 1", u'Master 1'),  
17             (u"Master 2", u'Master 2'),  
18         )  
19         user = models.OneToOneField(User, on_delete = models.CASCADE, p  
20         num_cart = models.CharField(max_length=20, blank=True)
```

Figure 35 : fichier models.py

- **Admin.py** est le fichier qui permet de définir les modèles qui seront visibles dans le site d'administration de django

```
from django.contrib import admin  
from .models import User, Etudiant, Prof, Alumni  
  
admin.site.register(User)  
admin.site.register(Etudiant)  
admin.site.register(Prof)  
admin.site.register(Alumni)
```

Figure 36 : fichier admin.py

- **Views.py** est une fonction Python acceptant une requête Web et renvoyant une réponse Web. Cette réponse peut contenir le contenu HTML d'une page Web, une redirection, une erreur 404, un document XML, une image... ou vraiment n'importe quoi d'autre. La vue elle-même contient la logique nécessaire pour renvoyer une réponse

```
account > views.py > etudiant_register

1  from django.contrib.auth import login, logout,authenticate
2  from django.shortcuts import redirect, render
3  from django.contrib import messages
4  from django.contrib.auth.decorators import login_required
5  from django.views.generic import CreateView
6  from .forms import EtudiantSignUpForm, ProfSignUpForm
7  from django.contrib.auth.forms import AuthenticationForm
8  from .models import User, Prof,Etudiant, Alumni
9  from .forms import UserEditForm, EtudiantEditForm, ProfEditForm, AlumniEditForm
10 import sweetify
11
12 def register(request):
13     return render(request, 'account/register.html')
14
15 class etudiant_register(CreateView):
16     model = User
17     form_class = EtudiantSignUpForm
18     template_name = 'account/etudiant_register.html'
19
20     def form_valid(self, form):
21         user = form.save()
22         login(self.request, user)
23         return redirect('register_done')
```

Figure 37 : fichier views.py

## 5.2. Présentation de l'application

A travers les différentes phases d'étude et de conception par lesquelles nous sommes passés, nous avons pu mettre en place un système informatique pour l'encadrement des étudiants de master 2. Ce système représente un prototype respectant les exigences qui permet de gérer numériquement un problème du secteur informel. Cependant avant de présenter quelques interfaces de l'application, nous présentons d'abord la vue globale de l'application composée des acteurs ainsi que des actions qui leurs sont possible de réaliser dans le système. La figure 38 illustre cette relation.

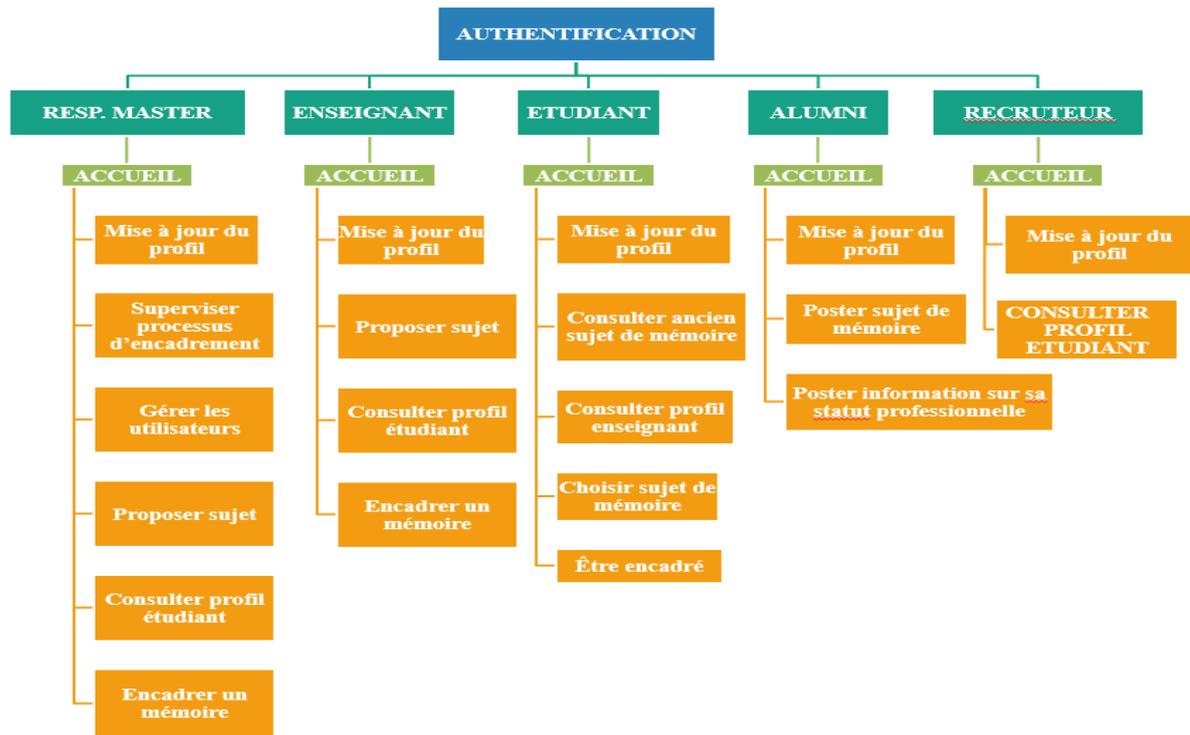


Figure 38 : vue globale de l'application

### 5.2.1. Inscription et Authentification

L'inscription est la première tâche à remplir pour avoir accès au système. Ainsi après avoir choisi le type d'utilisateur qu'on est (enseignant ou étudiant), il suffit de renseigner les informations demandées dans le formulaire d'inscription et d'attendre la validation du compte de la part du responsable du master.



a

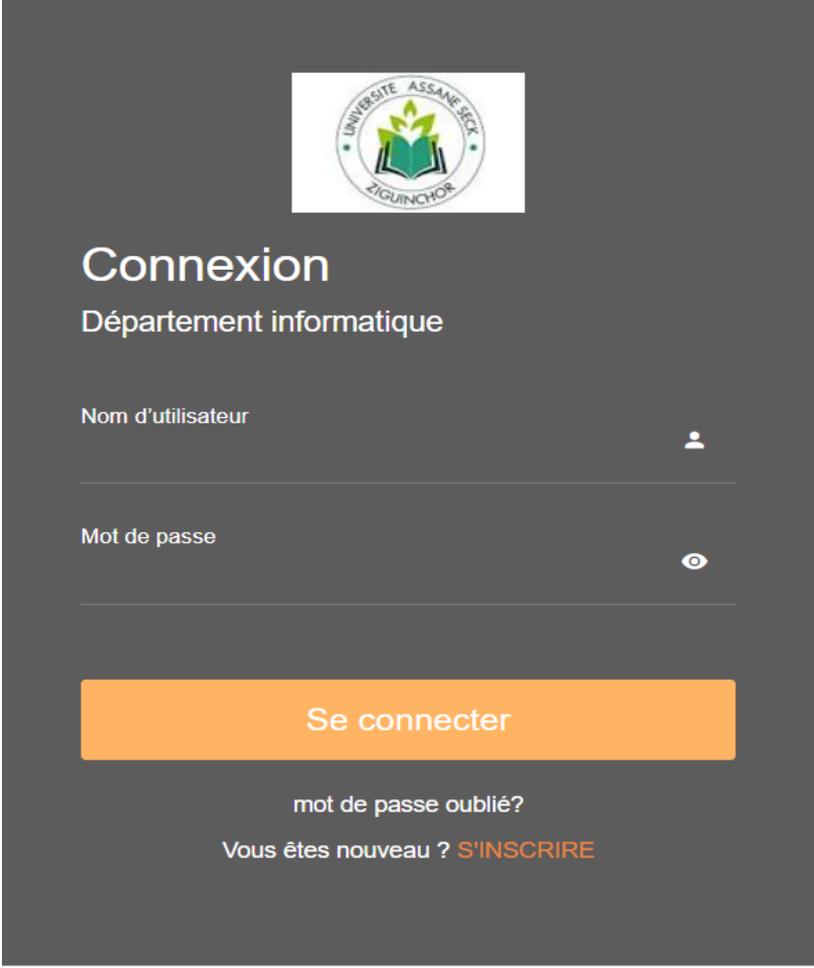
Figure 39.a : choix du profil



b

Figure 39.b : formulaire d'inscription

Après la validation du compte par le responsable de master, l'utilisateur peut à présent se connecter en renseignant son identifiant et son mot de passe qui lui donneront accès au système selon son profil (qu'il soit enseignant ou étudiant).



The image shows a login interface with a dark grey background. At the top center is the logo of Assane Seck University of Ziguinchor, featuring a green book and leaves inside a circular emblem with the text 'UNIVERSITE ASSANE SECK' and 'ZIGUINCHOR'. Below the logo, the title 'Connexion' is displayed in large white font, followed by 'Département informatique' in a smaller white font. There are two input fields: 'Nom d'utilisateur' with a user icon on the right, and 'Mot de passe' with an eye icon on the right. A prominent orange button with the text 'Se connecter' is centered below the fields. Underneath the button, there are two links: 'mot de passe oublié?' and 'Vous êtes nouveau ? S'INSCRIRE'.

Figure 40 : formulaire d'authentification du système

### 5.2.2. Menu d'administration (page d'accueil)

Le menu d'administration ou page d'accueil est la première page visible après la connexion. Pour le responsable du master il est composé de multiples informations d'aide à la décision ainsi que l'ensemble des fonctionnalités permettant de gérer les utilisateurs et de superviser l'encadrement.

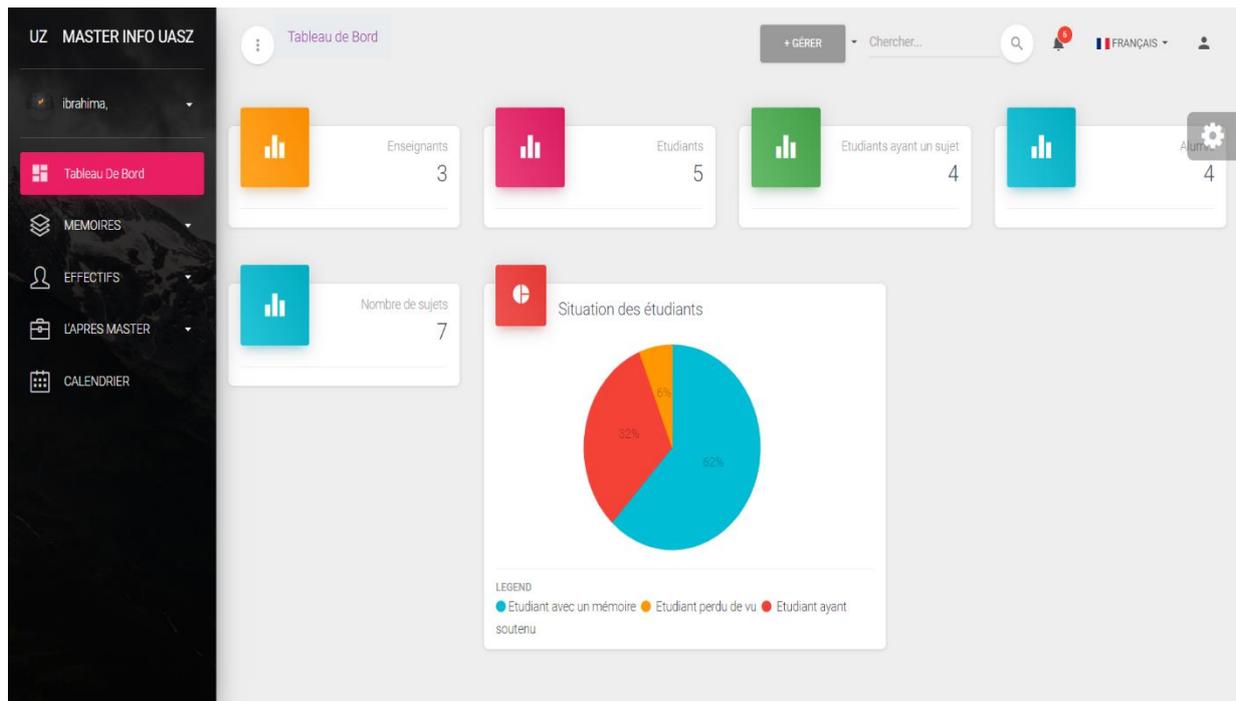


Figure 41 : page d'accueil du système pour le responsable de master

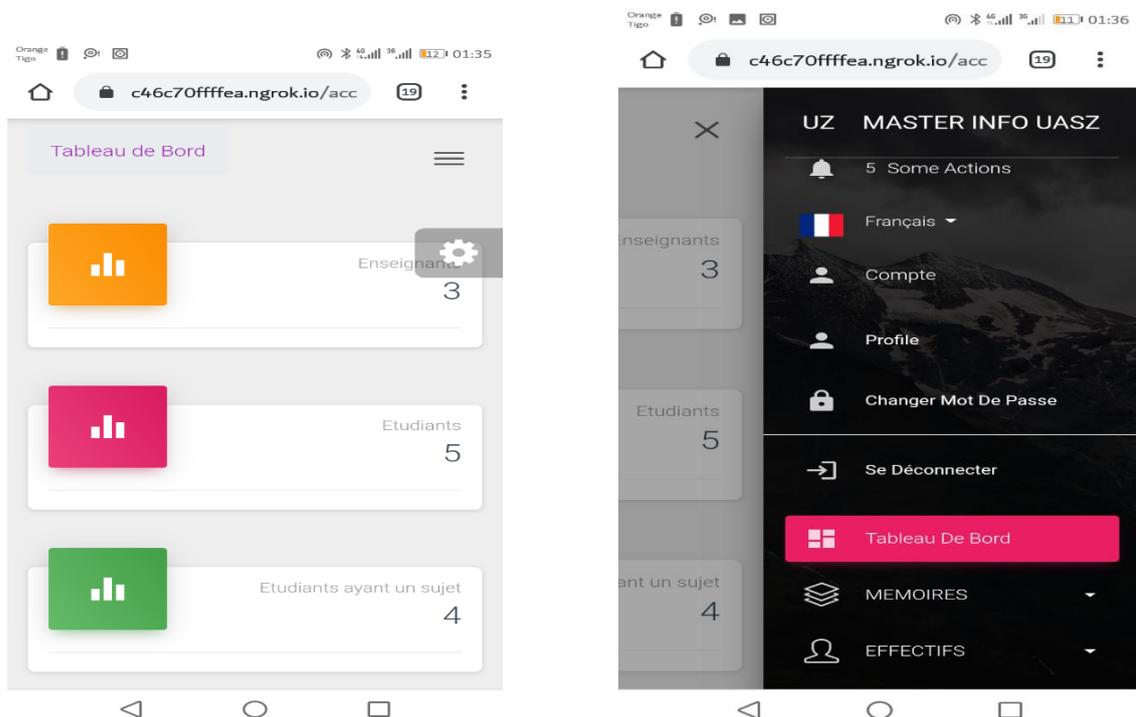


Figure 42 : page d'accueil sur terminal mobile

### 5.2.3. Présentation des profils des utilisateurs

Chaque utilisateur inscrit sur le site dispose d'un profil qu'il peut actualiser à sa guise. Ainsi l'utilisateur peut à tout moment mettre à jour les informations de son profil

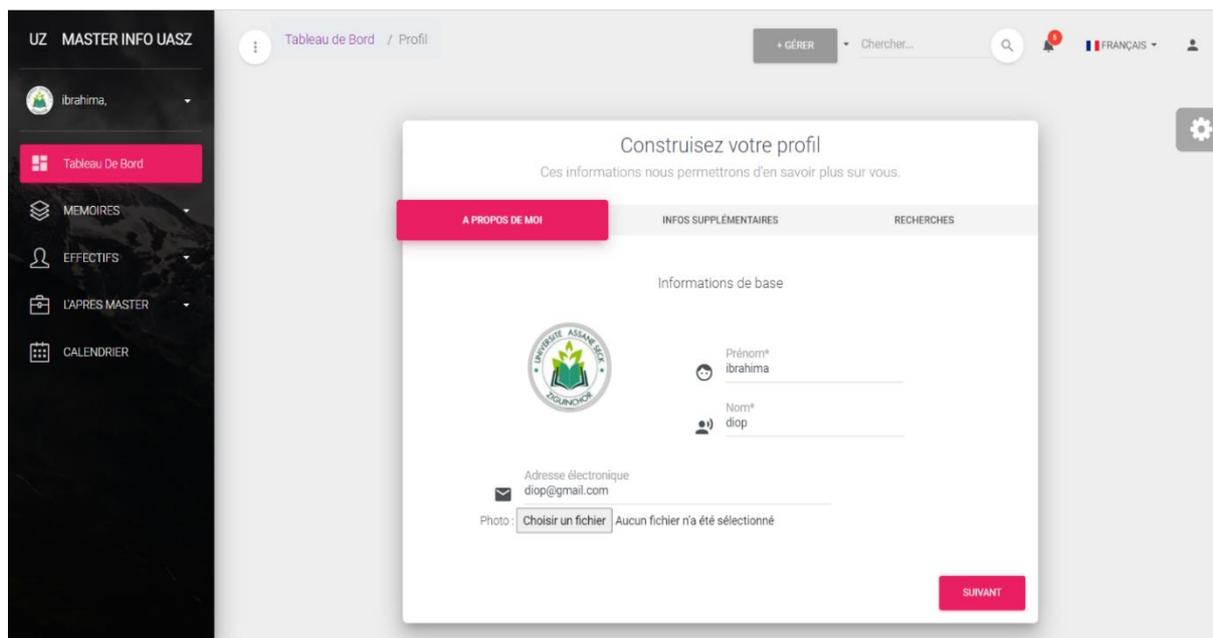


Figure 43 : actualisation du profil de l'utilisateur

#### 5.2.4. Liste des enseignants

L'application permet de voir la liste des enseignants qui se sont inscrits sur la plateforme ainsi que les étudiants.

| Ordre # | Prénom     | Nom    | Email                | type de prof         | Bureau | Actions |
|---------|------------|--------|----------------------|----------------------|--------|---------|
| 15      | louis      | dieme  | louis@gmail.com      | Enseignant chercheur | bp3    |         |
| 16      | serigne    | diagne | serigne@gmail.com    | Enseignant chercheur | bp4    |         |
| 17      | ibrahima   | diop   | ibrahima@gmail.com   | Intervenant externe  | bp5    |         |
| 18      | marie      | ndiaye | marie@gmail.com      | Enseignant chercheur | bp1    |         |
| 19      | khadim     | drame  | khadim@gmail.com     | Enseignant chercheur | bp3    |         |
| 20      | gorgoumack | samb   | gorgoumack@gmail.com | Intervenant externe  | bp6    |         |
| 21      | youssou    | faye   | youssou@gmail.com    | Enseignant chercheur | bp1    |         |
| 22      | youssou    | dieng  | youssoud@gmail.com   | Enseignant chercheur | bp6    |         |

Figure 44 : liste des enseignants

### 5.2.5. Affichage d'un enseignant

La vue d'affichage d'un enseignant comme celle d'un étudiant permet de consulter l'ensemble des informations concernant un enseignant, ses travaux de recherche ainsi que ses contacts (email et numéro de téléphone).

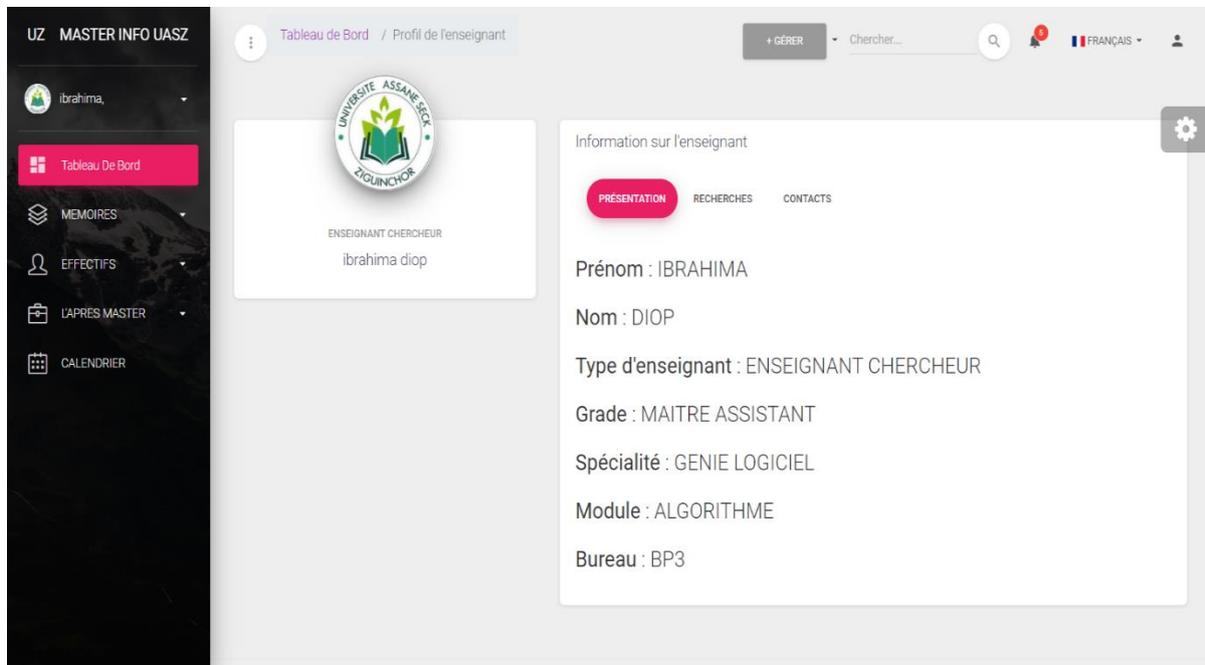


Figure 45 : affichage d'un enseignant

### 5.2.6. Ajout d'un document de mémoire

La fonctionnalité d'ajout d'un document de mémoire est réservée aux alumnis (anciens étudiants). Ainsi un alumni après sa soutenance peut poster son document de mémoire sur le système avec validation de son encadreur. L'ensemble des rapports de mémoire ajoutés par les alumnis sont consultables par les étudiants et tout utilisateur désireux de les consulter.

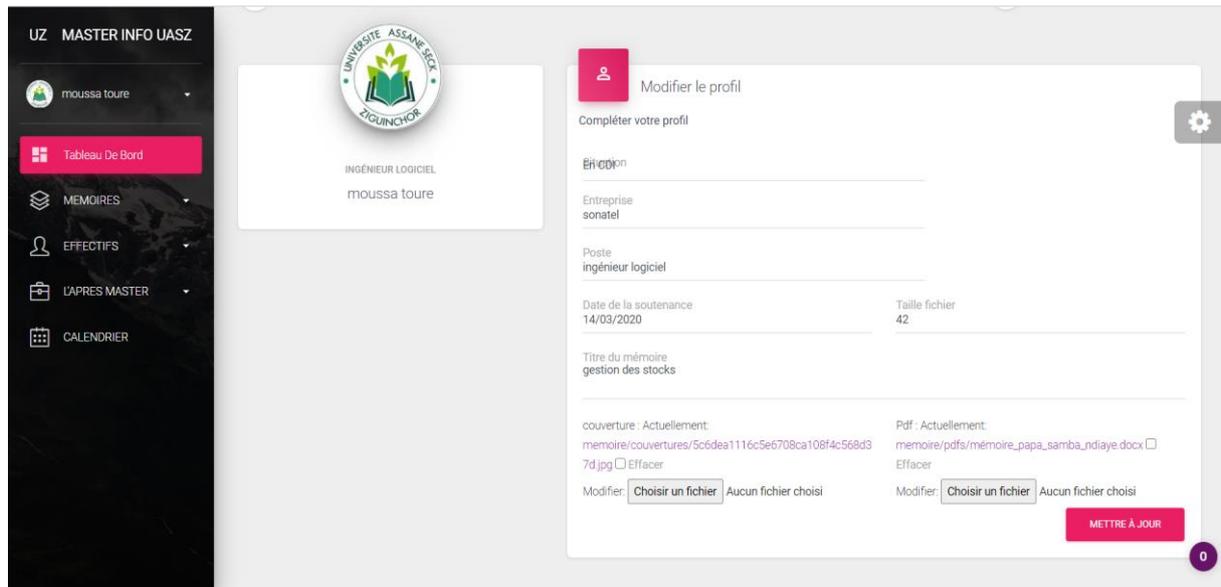


Figure 46 : ajout d'un document de mémoire

### 5.2.7. Bibliothèque des mémoires

La fonctionnalité bibliothèque des mémoires permet de lister l'ensemble des mémoires postés.

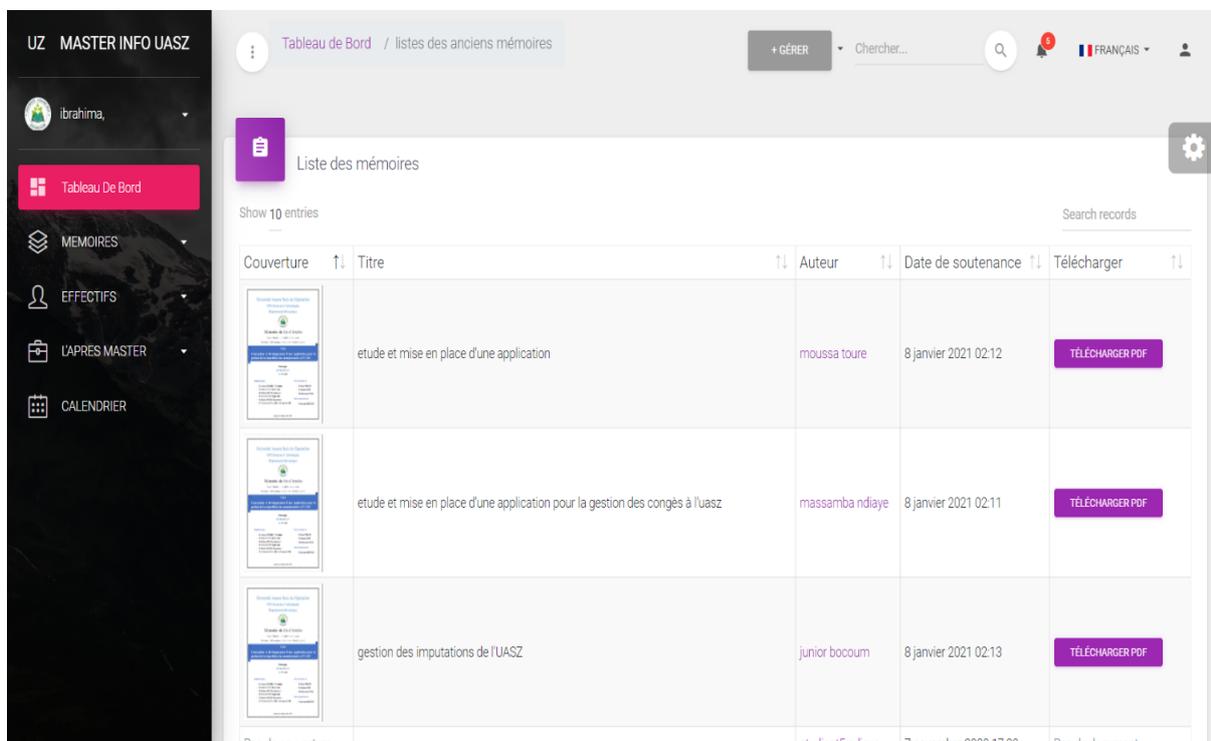


Figure 47 : liste des documents de mémoire

## 5.2.8. Ajout d'un sujet et liste des sujets de mémoire

Cette fonctionnalité est réservée aux enseignants. Ainsi l'enseignant peut poster des sujets de mémoire sur la plateforme qui se retrouveront dans la rubrique liste des sujets de mémoire et qui sera consultable par l'ensemble des étudiants. Par la suite un étudiant peut postuler aux sujets de son choix qui sera validé ou non par l'enseignant qui l'a posté.

Figure 48 : ajout d'un sujet de mémoire

| Domaine                            | Titre  | prérequis   | Auteür         | Date                 | Disponibilité | Télécharger          |
|------------------------------------|--|---|----------------|----------------------|---------------|----------------------|
| base de données                    | recherche sur les bases de données oracle et les contraintes d'intégrité   | maîtrise d'oracle                                   | serigne diagne | 8 janvier 2021 02:44 | DISPONIBLE    | TÉLÉCHARGER LE SUJET |
| base de données                    | mise en relation des bases de données  | maîtrise des bases de données, sérieux              | ibrahima diop  | 8 janvier 2021 02:36 | DISPONIBLE    | TÉLÉCHARGER LE SUJET |
| développement                      | étude et développement d'une application de vote électronique  | maîtrise d'un des langages du web                   | ibrahima diop  | 8 janvier 2021 02:43 | DISPONIBLE    | TÉLÉCHARGER LE SUJET |
| développement                      | développement d'une application pour la gestion du parc automobile de l'UASZ   | bonne maîtrise de java, et des méthodologies agiles | ibrahima diop  | 8 janvier 2021 02:38 | DISPONIBLE    | Pas de document      |
| intelligence artificielle          | étude et implémentation d'un réseau de neurone   | sérieux dans le travail                             | serigne diagne | 8 janvier 2021 02:45 | DISPONIBLE    | TÉLÉCHARGER LE SUJET |
| recherche sur l'internet des objet | les interconnexions du futur   | sérieux dans le travail                             | louis dieme    | 8 janvier 2021 02:46 | DISPONIBLE    | TÉLÉCHARGER LE SUJET |
| réseaux                            | découverte et implémentation du protocole IPv6 dans le réseau informatique de l'université Assane SECK de ZIGUINCHOR | sérieux dans le travail                             | ibrahima diop  | 8 janvier 2021 02:40 | DISPONIBLE    | Pas de document      |

Figure 49 : liste des sujets de mémoire

### 5.2.9. Rubrique suivie des mémoires

La rubrique suivie des mémoires permet à un enseignant d'encadrer un étudiant sur la plateforme. A la suite de la validation du sujet par l'enseignant, il est attribué dans l'application un espace qui permettra à l'enseignant de suivre le travail de l'étudiant pas à pas.

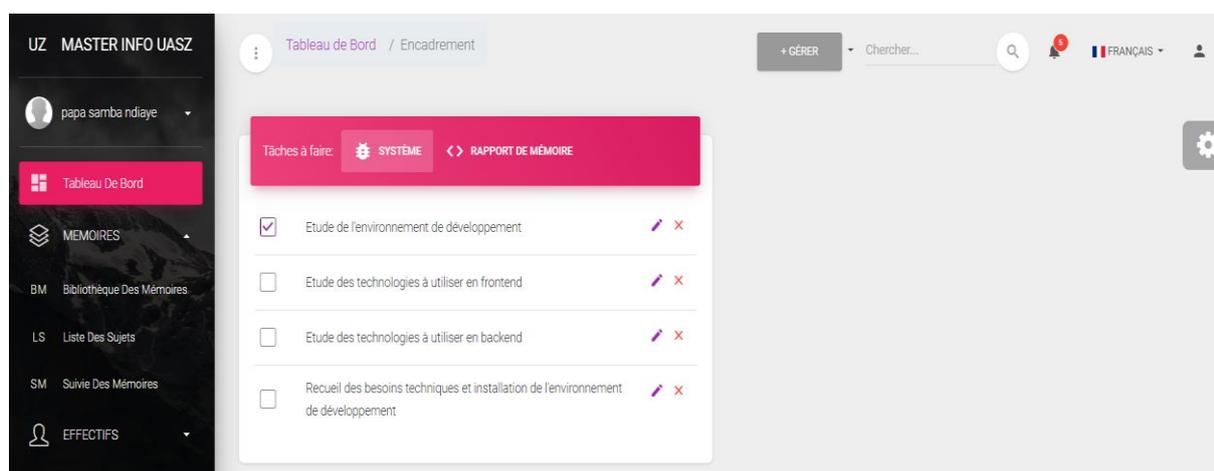


Figure 50 : liste des tâches à faire

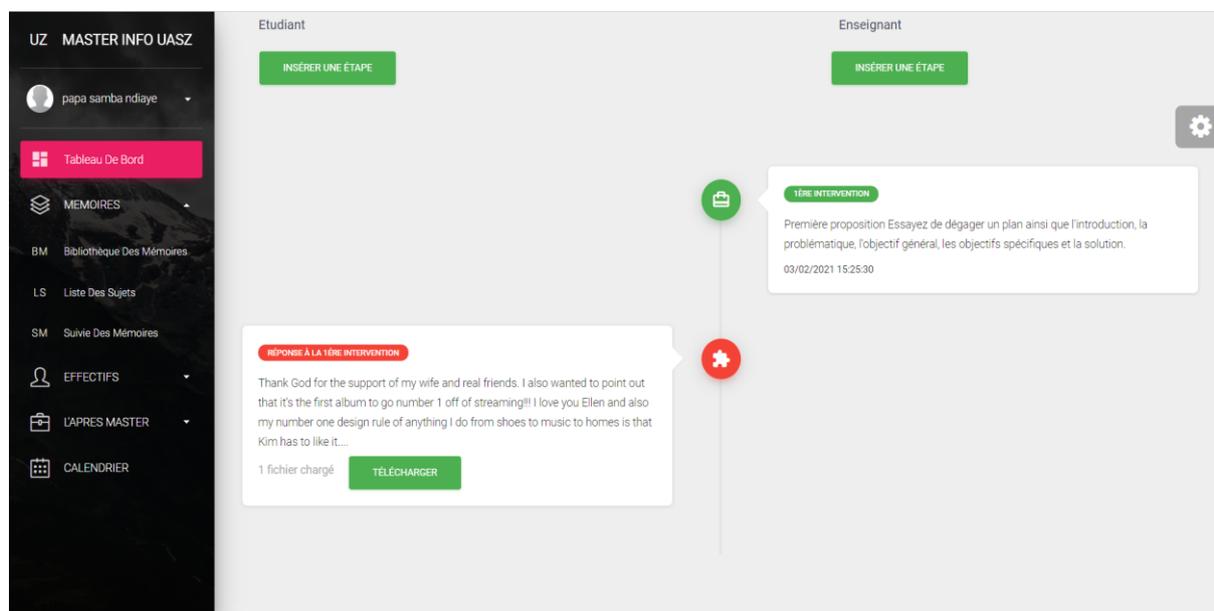


Figure 51 : timeline de l'encadrement

## 5.3. Mise en production sur Heroku

Lancé en 2007 [13], Heroku est une PaaS (plateforme en tant que service) permettant de déployer des applications sur le Cloud. A la façon dont un hébergeur web propose d'héberger

un site web sur ses propres serveurs, cette solution nous permet de déployer notre application sur le Cloud pour permettre aux internautes de l'utiliser.

Les applications sont exécutées dans des 'dynos', à savoir des ordinateurs virtuels dont la puissance peut être ajustée en fonction de l'envergure de l'application. Ainsi, les dynos peuvent être comparés à des blocs de construction.

Si nous avons besoin de traiter davantage de données ou d'effectuer des tâches plus complexes, il est possible d'ajouter davantage de blocs. C'est ce que l'on appelle le scaling horizontal. Il est aussi possible d'augmenter la taille des blocs, ce qu'on appelle le scaling vertical. Heroku facture ensuite ses clients en fonction du nombre de dynos utilisés et de leur taille

Nous avons fait le choix d'utiliser Heroku pour la mise en production de notre projet du fait de sa célébrité et son support actif [13] ainsi que l'existence d'une solution non payante qui nous a permis de mettre notre application en production.

*La mise en production de notre application sur Heroku est consultable à l'annexe 6*

## **5.4. Conclusion**

Dans ce chapitre, nous avons évoqué et traité l'implémentation et la présentation de l'application. D'abord on a expliqué l'implémentation du système en détaillant les technologies utilisées tant côté Front-End que côté Back-End, ensuite on a présenté l'application en montrant quelques-unes de ces interfaces.

Cependant pour nous assurer de la sécurité et de l'ergonomie du système on a réalisé un ensemble de tests qui feront l'objet du prochain chapitre.

# CHAPITRE 6 : SECURITE ET PERFORMANCE DU SYSTEME

## 6.1. Audit de la sécurité du système

L'audit de la sécurité d'un système informatique permet de jauger le niveau de sécurité du système en pratiquant des tests de pénétration, des balayages du réseau et la recherche de faille permettant de sécuriser au mieux le système. Pour notre application on a eu à pratiquer des tests de pénétration du système sous kali linux (distribution linux réservée à la sécurité informatique) allant de la recherche d'information sensible laissée au hasard pendant le codage jusqu'à l'audit des potentiels failles web. Nous avons eu des résultats concluants que nous avons développés dans les sections suivantes.

### 6.1.1. Sécurisation de la recherche d'information : Maltego

Maltego [32] est un logiciel utilisé pour l'intelligence open-source et la criminalistique, développé par Paterva de Pretoria, Afrique du Sud. Maltego se concentre sur la fourniture d'une bibliothèque de transformations pour la découverte de données à partir de sources ouvertes et la visualisation de ces informations dans un format graphique, adapté à l'analyse de liens et à l'exploration de données.

Maltego permet de créer des entités personnalisées, lui permettant de représenter tout type d'informations en plus des types d'entités de base qui font partie du logiciel. L'objectif de base de l'application est d'analyser les relations du monde réel (réseaux sociaux, API OSINT, nœuds de données privées et de réseaux informatiques auto-hébergés) entre les personnes, les groupes, les pages Web, les domaines, les réseaux, l'infrastructure Internet et les affiliations aux médias sociaux. Maltego étend sa portée de données avec des intégrations de divers partenaires de données. Parmi les sources de données sont les enregistrements DNS, whois dossiers, les moteurs de recherche, services de réseaux sociaux, divers API et diverses métadonnées.

**Application :** Maltego est installé par défaut sur kali linux donc on a pu faire nos tests sans grande difficulté sur la partie installation.

Les résultats obtenus sont affichés sur la figure 52 :

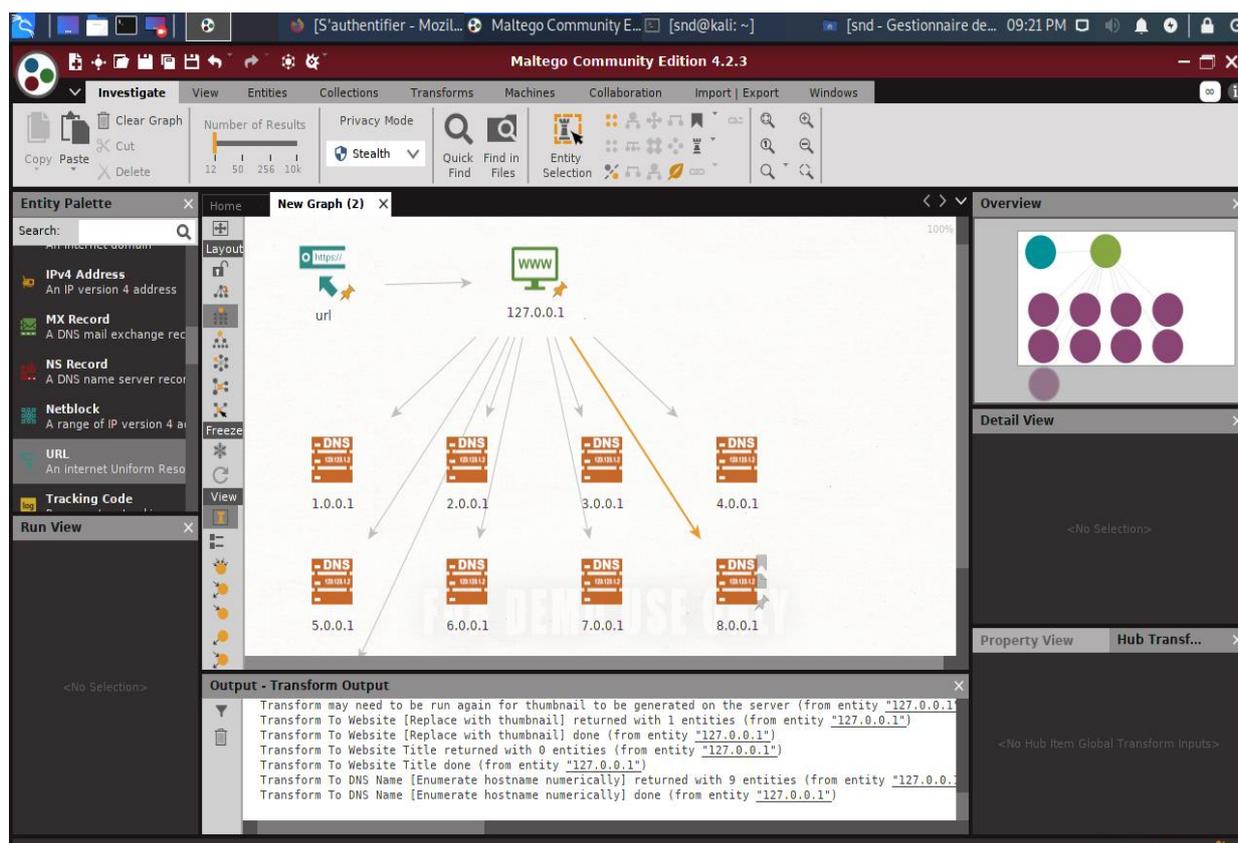


Figure 52 : audit système avec maltego

L'analyse de l'url local de notre application nous a permis de vérifier qu'il y a aucune information sensible (email numéro de téléphone ou lien) qui était accessible en dehors du système à part le DNS interne du serveur local de Django qu'essayait de joindre notre application dans chaque compartiment bien structuré du code.

## 6.1.2. Audit des failles web : Owasp Zap et Vega

### a. Owasp Zap

OWASP ZAP (abréviation de Zed Attack Proxy) est un scanner de sécurité d'applications Web open source. Il est destiné à être utilisé à la fois par les nouveaux utilisateurs de la sécurité des applications et par les testeurs de pénétration professionnels. C'est l'un des projets OWASP (Open Web Application Security Project) les plus actifs et il a reçu le statut Flagship. Lorsqu'il est utilisé comme serveur proxy, il permet à l'utilisateur de manipuler tout le trafic qui le traverse, y compris le trafic en utilisant https. Il peut également fonctionner en mode démon qui est ensuite contrôlé via une API REST.

**Application :** Zap est installé par défaut sur kali linux donc nous n'avons pas rencontré de difficulté pour son lancement. On a pu effectuer une analyse des failles web du système. La figure 53 nous donne le résultat.

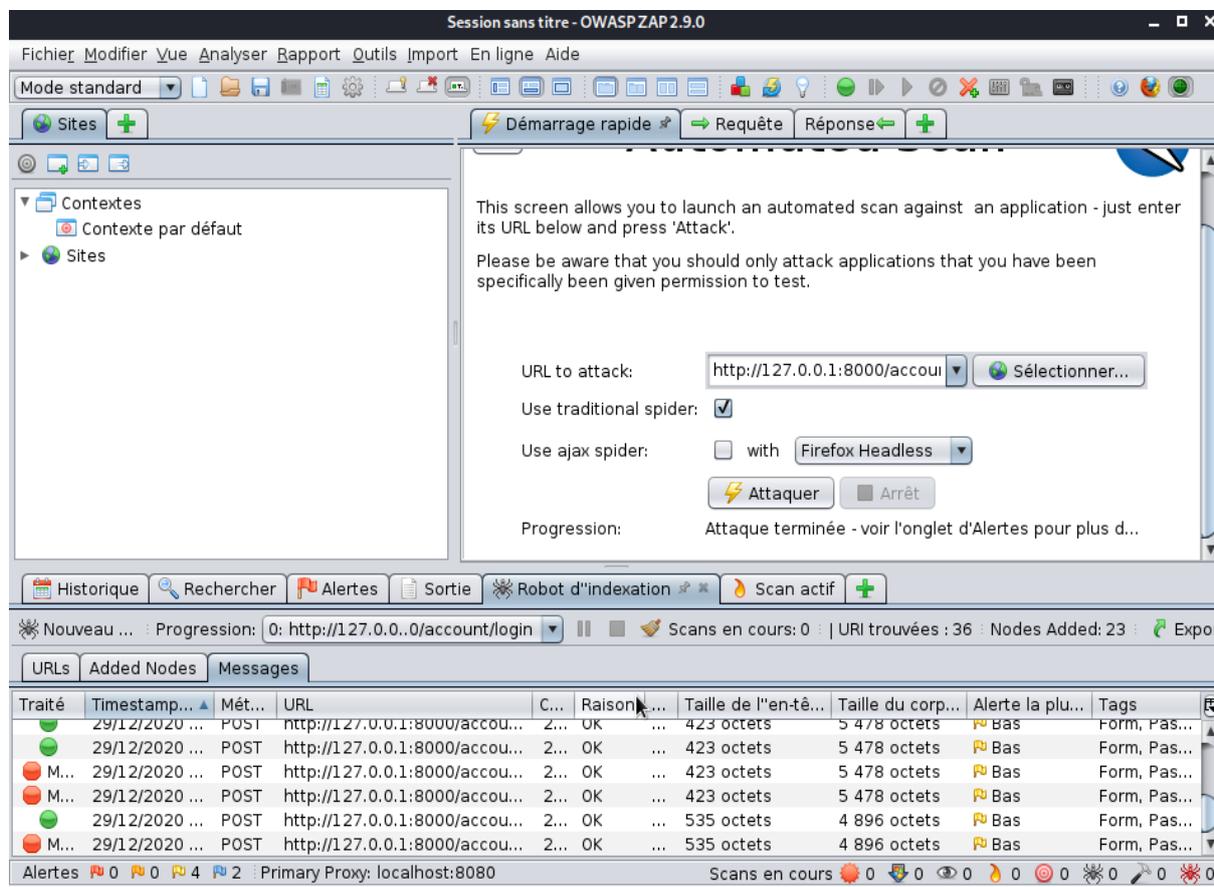


Figure 53 : audit sécurité web avec Zap

Nous remarquons que nous n'avons pas eu de faille critique à déplorer mais seulement des alertes le plus souvent dues au manque de certificat. Défaut que nous comblerons au déploiement avec l'obtention d'un certificat.

**Le rapport complet de l'audit avec une explication détaillée est disponible en annexe1.**

### b. Vega

Vega [31] est un scanner de sécurité Web gratuit et open source et une plate-forme de test de sécurité Web pour tester la sécurité des applications Web. Vega peut aider à trouver et à valider l'injection SQL, le Cross-Site Scripting (XSS), les informations sensibles divulguées par inadvertance et d'autres vulnérabilités. Il est écrit en Java, basé sur l'interface graphique et fonctionne sous Linux, OS X et Windows.

Vega peut aider à trouver des vulnérabilités telles que : les scripts intersites reflétés, les scripts intersites stockés, l'injection SQL aveugle, l'inclusion de fichiers distants, l'injection Shell, etc. Vega sonde également les paramètres de sécurité TLS / SSL et identifie les opportunités pour améliorer la sécurité des serveurs TLS.

**Application :** nous avons procédé à l'installation de Vega en allant sur le site officiel du projet et en téléchargeant la version 64bits pour linux. Ensuite nous avons procédé à un audit de notre système.



Figure 54 : Audit sécurité web avec Vega

Nous constatons qu'avec l'analyse Vega plus approfondie nous avons 7 menaces de niveau élevé, 1 menace moyenne, 3 menaces faibles et 2 alertes. La plupart des menaces sont dues à l'absence du Protocole http et du fait que l'application s'exécute sur un serveur local et par conséquent beaucoup de règles de sécurité sont négligées ou bloquées.

Les types de menaces restantes sont dus à l'activation du débogueur qui permet de retourner certaines informations sensibles aux développeurs.

## 6.2. Audit des performances du système : Lighthouse

Lighthouse est un outil automatisé open source pour améliorer les performances, la qualité et l'exactitude d'applications Web.

Lors de l'audit d'une page, Lighthouse exécute un barrage de tests sur la page, puis génère un rapport sur les performances de la page. À partir de là, il est possible d'utiliser les tests qui ont échoué comme indicateurs de ce qu'il est possible de faire pour améliorer application.

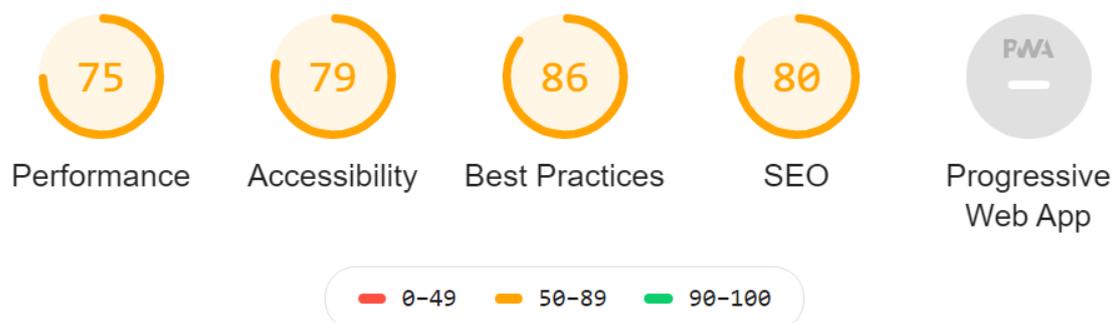


Figure 55 : Résultat des tests de performance

(Rapport de l'audit lighthouse voir Annexe 5).

## 6.3. Conclusion

Ce chapitre nous a permis d'auditer la sécurité et les performances de notre système. Tout d'abord on a testé la sécurité du site en vérifiant s'il n'y avait pas des informations qui sortaient du domaine notamment avec l'outil maltego. Ensuite on a pu auditer le site avec des outils comme Owasp Zap et Vega à la recherche de faille de sécurité qu'on a trouvé et corrigé avec succès. On a terminé ce chapitre par l'audit des performances du système notamment avec le système de google lighthouse ou le système a décroché un score de 80 sur un indice de performance de 100 au maximum.

# CONCLUSION GENERALE ET PERSPECTIVES

Ce travail a eu comme objectif de concevoir et d'implémenter un système informatique et décisionnel pour l'encadrement des étudiants en mémoire de master 2 et la traçabilité des anciens étudiants. (Cas pratique du département informatique de l'UASZ).

Ainsi nous pouvons affirmer que l'objectif est atteint car nous avons pu réaliser un système qui répond aux besoins exprimés. En effet, le système permet, entre autres de donner les commandes au responsable du master qui dispose d'un tableau de bord lui permettant de visualiser et de piloter l'écosystème de l'encadrement des étudiants en master 2 en ayant des informations clés comme le nombre d'étudiants ayant eu un sujet de mémoire, le nombre d'étudiants dans l'attente d'une soutenance, le nombre d'ancien étudiant en entreprise ainsi que leurs contacts ... le système permet aussi à un enseignant de proposer des sujets de mémoire et de suivre l'évolution des travaux des étudiants qu'il encadre sur la plateforme. De même pour les étudiants ils peuvent consulter les rapports de mémoire postés sur la plateforme, mais aussi les profils des enseignants. Les recruteurs sont aussi pris en charge avec une option pour consulter le profil d'un étudiant.

A cet effet, afin de bien présenter les objectifs et les résultats de notre travail, ce mémoire a été scindé en six principaux chapitres. Le premier chapitre a fait l'objet de la présentation du contexte justificatif du sujet ainsi que les solutions envisagées aux problèmes rencontrés lors du processus d'encadrement d'un étudiant en mémoire de master 2. Le deuxième chapitre a parlé du processus de développement dans lequel nous avons présenté la méthodologie Scrumban et l'avons adapté à notre projet. Le troisième chapitre s'est intéressé à la spécification et l'analyse des besoins fonctionnels du système. Le quatrième chapitre a abordé la conception de l'application. Le cinquième chapitre a présenté le système à savoir les technologies utilisées à l'implémentation et les différentes interfaces du système. Enfin, le sixième chapitre a présenté l'audit de la sécurité et des performances du système.

Nous avons aussi réalisé un formulaire d'enquête sur google Forms (**disponible en annexe 1 du document**) qui a permis aux étudiants de donner leurs avis sur l'existant et le système en place. Cette enquête permettra éventuellement au responsable de master et l'ensemble des

enseignants de redresser ou eclaircir certains points qui entravent le bon deroulement des memoires de master 2.

Dans les perspectives, nous envisageons d'ameliorer l'application de par les fonctionnalites suivantes :

- **Créer un profil pour le responsable pédagogique :** Cela permettra au responsable pédagogique de delivrer les certificats de soutenances directement dans le systeme apres l'approbation de l'encadreur ;
- **Elargir l'utilisation du systeme dans les autres departements de l'universite :** cela consistera a assister la scalabilite du systeme pour permettre aux autres departements de l'universite de pouvoir beneficier de l'application ;
- **Mettre en place un chat.** Cela consistera a mettre en place un service de discussion instantane qui permettra aux etudiants et enseignants d'echanger en temps reel a travers l'application.
- **Mettre en place un forum.** Cette fonctionnalite permettra aux enseignants et aux etudiants de poster des sujets de discussions ayant un rapport avec les memoires de master 2.

# BIBLIOGRAPHIE

## Livre

- (1) **MELE, Antonio.** Django 3 by example. Third Edition. 35 rue Livery  
Birmingham B3 2PB, Royaume-Uni. : Pavan Ramchandani, 2020. 569 pages.
- (2) **GABORY, Yohann.** Django avancé : pour des applications web puissantes en  
Python. 1<sup>ère</sup> édition. 61 bd Saint-Germain, France : Editions Eyrolles, 2013. 397 pages

## Mémoire

- (3) **DIALLO, Henry.** Conception et développement d'une application pour la gestion de  
la répartition des enseignements à l'UASZ. Mémoire de master 2. : Informatique,  
mention génie logiciel : Ziguinchor : Université Assane SECK de Ziguinchor : 2018.  
82 pages réparties en cinq chapitres.
- (4) **SEYE, ADAMA.** Conception et mise en œuvre d'un système de vote électronique :  
application aux élections organisées à l'UASZ. Mémoire de master 2. : Informatique,  
mention génie logiciel : Ziguinchor : Université Assane SECK de Ziguinchor : 2019.  
84 pages réparties en cinq chapitres.

# WEBOGRAPHIE

- [1] : Site de studyrama <https://www.studyrama.com/formations/diplomes/master/> consulté le 13/10/2020
- [2] : Site de simpleisbetterthancomplex <https://simpleisbetterthancomplex.com/tutorial/2018/01/18/how-to-implement-multiple-user-types-with-django.html> consulté le 15/10/2020
- [3] : Site de geeksforgeeks <https://www.geeksforgeeks.org/choicefield-django-forms/> consulté le 16/10/2020
- [4] : Site de github <https://github.com/Atrox/sweetify-django> consulté le 19/10/2020
- [5] : Site de attineos <https://www.attineos.com/blog/autres-sujets/les-methodes-de-gestion-de-projet-traditionnelles-vs-agiles> consulté le 19/10/2020
- [6] : Site de journal du net <https://www.journaldunet.fr/web-tech/guide-de-l-entreprise-digitale/1443832-kanban-une-methode-agile-de-gestion-de-projet-visuelle-et-continue/> consulté le 20/10/2020
- [7] : Site de <https://www.developpez.com/actu/203845/Agile-entre-Scrum-et-Kanban-laquelle-des-deux-methodologies-est-elle-la-meilleure-Le-point-dans-une-etude-comparative/> consulté le 29/10/2020
- [8] : Site de <https://www.blog-projet.fr/2016/06/20/approche-traditionnelle-agilite-differences/> consulté le 30/10/2020
- [9] : Site de <http://www.oeildecoach.com/scrum-vs-kanban-vs-agile-vs-cycle-en-v-comparatif/> consulté le 31/10/2020
- [10] : Site de <https://www.bocasay.com/fr/methode-scrumban-projets-developpement-informatique/> consulté le 01/11/2020
- [11] : Site de wikipédia les serveurs [https://fr.wikipedia.org/wiki/Serveur\\_informatique#:~:text=Un%20serveur%20informatique%20est%20un,clients%20\(parfois%20des%20milliers\).](https://fr.wikipedia.org/wiki/Serveur_informatique#:~:text=Un%20serveur%20informatique%20est%20un,clients%20(parfois%20des%20milliers).) 02/11/2020
- [12] : Site de pypi <https://pypi.org/project/django-bootstrap-modal-forms/> consulté le 03/11/2020
- [13] : Site de wikipédia Héroku <https://fr.wikipedia.org/wiki/Heroku> consulté le 3/11/2020
- [14] : Site de academia.edu [https://www.academia.edu/33346693/Chapitre\\_2\\_Sp%C3%A9cification\\_et\\_analyse\\_des\\_besoins](https://www.academia.edu/33346693/Chapitre_2_Sp%C3%A9cification_et_analyse_des_besoins) consulté le 08/11/2020
- [15] : Site de lucidchart <https://www.lucidchart.com/pages/fr/diagramme-de-sequence-uml> consulté le 09/11/2020
- [16] : Site de wikiversity [https://fr.wikiversity.org/wiki/Mod%C3%A9lisation\\_UML/Le\\_diagramme\\_d%27activit%C3%A9](https://fr.wikiversity.org/wiki/Mod%C3%A9lisation_UML/Le_diagramme_d%27activit%C3%A9) consulté le 10/11/2020
- [17] : Site de hardware [https://forum.hardware.fr/hfr/Programmation/Divers-6/utilisation-description-textuelle-sujet\\_146836\\_1.htm](https://forum.hardware.fr/hfr/Programmation/Divers-6/utilisation-description-textuelle-sujet_146836_1.htm) consulté le 11/11/2020
- [18] : Site de data-flair.training <https://data-flair.training/blogs/django-architecture/> consulté le 13/11/2020

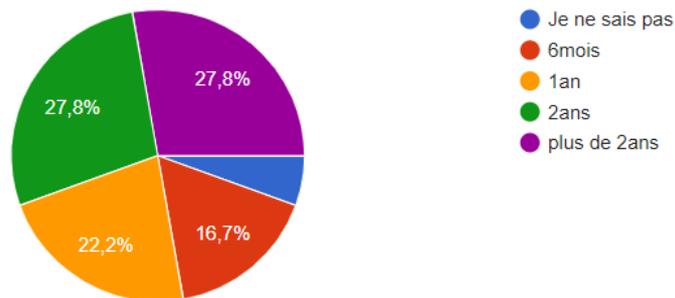
- [19] Site de zestedesavoir [https://zestedesavoir.com/tutoriels/598/developpez-votre-site-web-avec-le-framework-django/262\\_presentation-de-django/1517\\_le-fonctionnement-de-django/](https://zestedesavoir.com/tutoriels/598/developpez-votre-site-web-avec-le-framework-django/262_presentation-de-django/1517_le-fonctionnement-de-django/) consulté le 13/11/2020
- [20] Site de geeksforgeeks <https://www.geeksforgeeks.org/difference-between-mvc-and-mvt-design-patterns/> consulté le 18/12/2020
- [21] Site de explorweb.github <https://explorweb.github.io/cours2018A/cours/django/> consulté le 20/12/2020
- [22] Site de djangoproject <https://www.djangoproject.com/> consulté le 22/12/2020
- [23] Site de wikipedia <https://fr.wikipedia.org/wiki/React> consulté le 23/12/2020
- [24] Site de jquery <https://jquery.com/> consulté le 24/12/2020
- [25] Site de djangoproject [https://fr.wikipedia.org/wiki/Feuilles\\_de\\_style\\_en\\_cascade](https://fr.wikipedia.org/wiki/Feuilles_de_style_en_cascade) consulté le 24/12/2020
- [26] Site de djangoproject <https://docs.djangoproject.com/fr/3.1/ref/databases/#postgresql-notes> consulté le 26/12/2020
- [27] Site de wikipedia <https://fr.wikipedia.org/wiki/PostgreSQL> consulté le 28/12/2020
- [28] Site de postgresql <https://www.postgresql.org/> consulté le 28/12/2020
- [29] Site de wikipedia [https://fr.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://fr.wikipedia.org/wiki/Visual_Studio_Code) consulté le 29/12/2020
- [30] Site de phyks <https://pub.phyks.me/sdz/sdz/developpez-vos-applications-web-avec-django.html> consulté le 29/12/2020
- [31] Site de Vega <https://subgraph.com/vega/> consulté le 30/12/2020
- [32] Site de wikipedia (maltégo) <https://en.wikipedia.org/wiki/Maltego> consulté le 30/12/2020
- [33] Site de ludichart <https://www.lucidchart.com/pages/fr/diagramme-package-uml> consulté le 31/12/2020
- [34] Site de wikipedia recherche portant sur git <https://fr.wikipedia.org/wiki/Git> consulté le 03/01/2021
- [35] Site de wikipedia sqlite <https://fr.wikipedia.org/wiki/SQLite> consulté le 03/01/2021
- [36] Site de Iris <http://www.lsis.org/dea/M6optionD/Exp-GL-UML>. Consulté le 08/11/2020
- [37] Site de développez.com <https://laurent-audibert.developpez.com/Cours-UML/?page=diagrammes-composants-deploiement>. Consulté le 09/11/2020
- [38] Site de trello <https://blog.trello.com/fr/methode-agile-kanban> consulté le 10/11/2020
- [39] Site de Wikipédia dictionnaire de données <https://blog.trello.com/fr/methode-agile-kanban> consulté le 10/11/2020.

## Annexe

### Annexe 1 : Formulaire d'enquête google Forms sur le mémoire de master 2 au département informatique de l'UASZ.

A votre connaissance combien de temps dure le mémoire de master 2 au département informatique de l'UASZ ?

18 réponses



Préférez vous avoir un sujet au début du master 2 ou a la fin du premier semestre de master 2 ?

18 réponses



Quand avez vous su qu'il existe une alternative au mémoire qui est le stage en entreprise ?

18 réponses



disposez vous d'une solution digitale propre à l'UASZ pour consulter ou télécharger les documents de mémoire des anciens étudiants ?

18 réponses



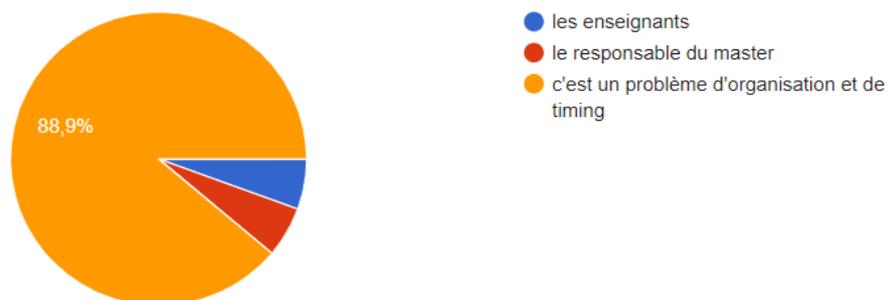
Comment un sujet de mémoire est proposé à un étudiant ?

17 réponses



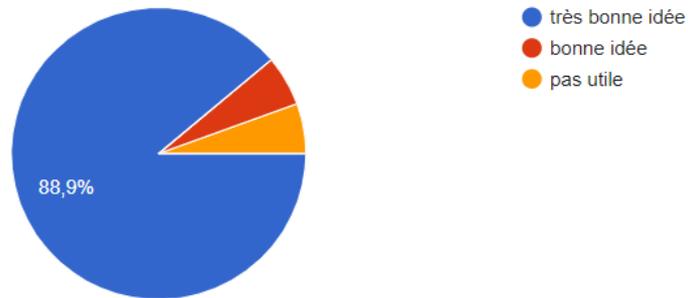
Selon vous qui est le responsable du retard qu'accuse les sujets de mémoire pour parvenir aux étudiants ?

18 réponses



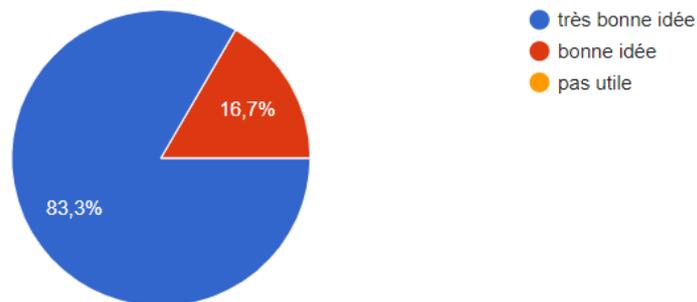
Que pensez vous d'un systeme en ligne pour consulter ou telecharger les sujets de memoire des anciens etudiants ?

18 reponses



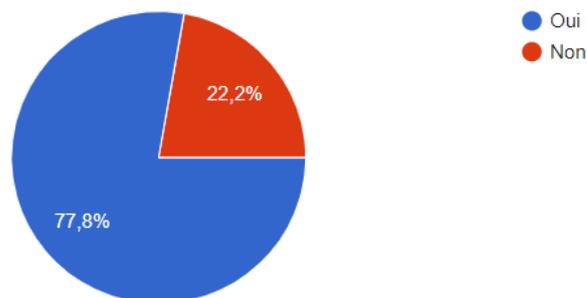
Que pensez vous d'un systeme en ligne qui permettra aux enseignants de proposer des sujets de memoire consultable en ligne en toute periode ?

18 reponses



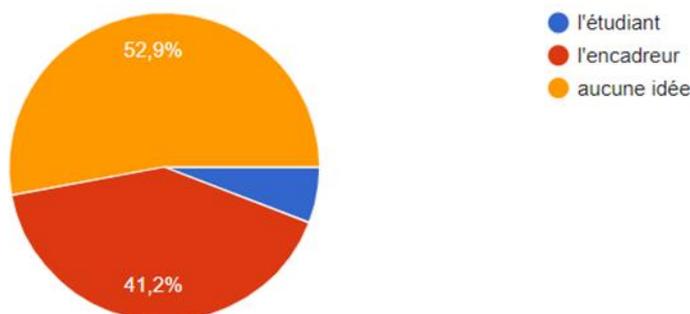
Pensez vous pouvoir faire votre memoire en 6 mois ?

18 reponses



Selon vous qui est le responsable du dépassement des délais d'encadrement du mémoire des étudiants de master 2 ?

17 réponses

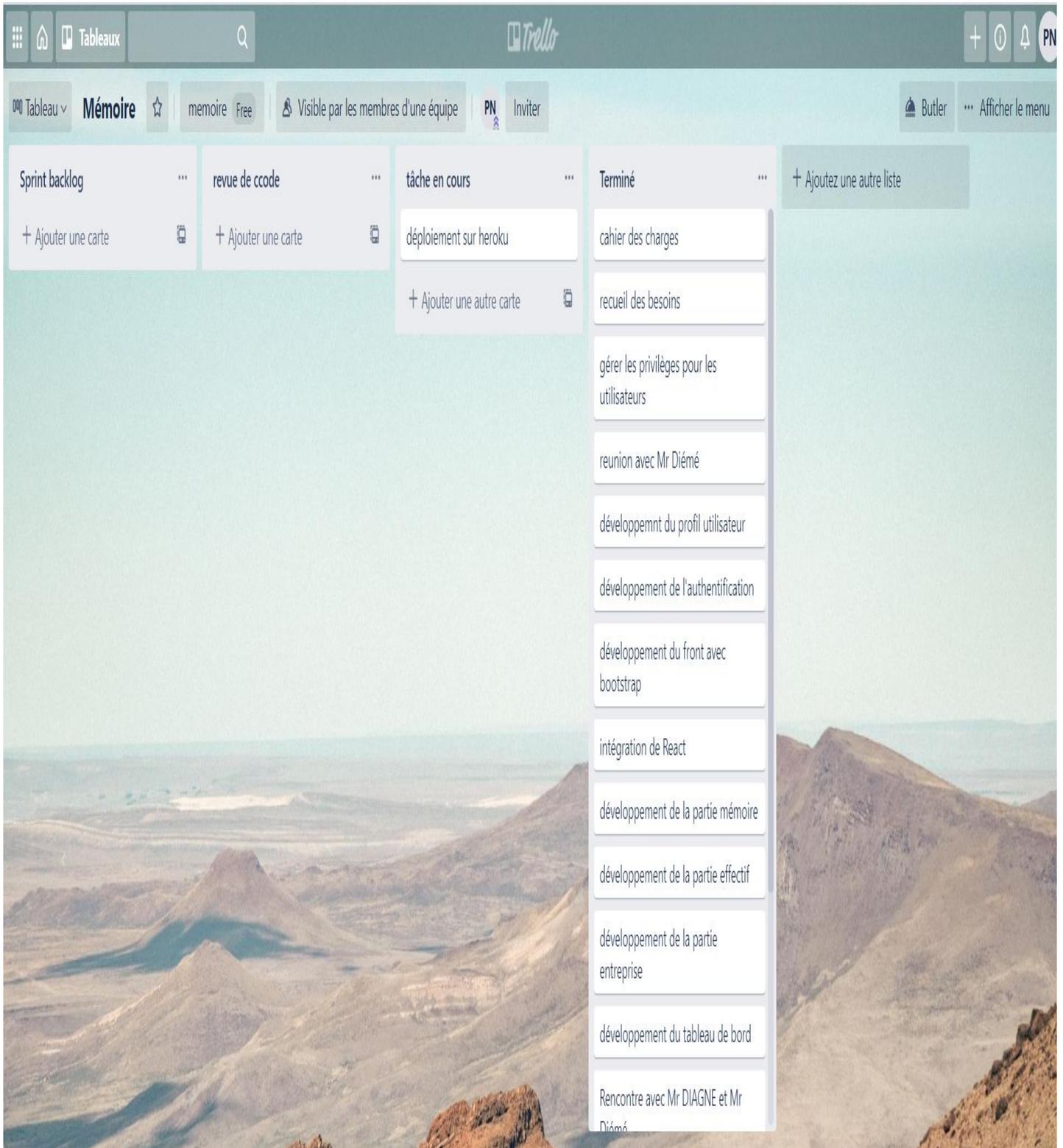


Quelles sont vos suggestions pour améliorer le processus d'encadrement des étudiants en mémoire au département informatique de l'UASZ ?

|  |
|--|
| L'encadrement est non satisfaisant   |
| Il est impératif de revoir l'organisation du processus d'encadrement. A mon avis, l'étudiant peut avoir un sujet depuis le Master 1 cela lui permettrait de gagner plus de temps. La mise en ligne des mémoires des anciens doit être une réalité car cela peut aider l'étudiant dans la rédaction de son mémoire          |
| Il faut des sujet à al fin du master 1 mais pour ce là il faut dés la seconde semestre du master 1. Et que les étudiants en master qu'est ce qu'on veut faire c'est à dire la spécialisation.  |
| Il faut les encadreurs se communiquent avec les étudiants  |
| Je pense il y'a beaucoup à améliorer pour l'avancement de notre université   |
| Mettre à la disposition de tous les documents pour les étudiants   |
| Les enseignants doivent faire preuve de disponibilité à l'égard des étudiants. Définir un plan de travail que les deux entités sont tenues de respecter  |
| aucune idée  |
| Mettre en place une plateforme dédiée au choix des sujets et des encdreurs.<br>- Mettre en place une plateforme de documentation.<br>- Inciter les étudiants à proposer des sujets.<br>- Commencer les démarches des mémoires au plus tard au début du master 2.<br>- Faire respecter les étudiants le délai des mémoires. |
| Les sujets de mémoire doivent être disponible le plus tôt possible.  |

|  |
|--|
| Il faut plus d'efforts pour diminuer la durée des mémoires chez les étudiants, responsable Master et enseignants. Le problème des mémoires ne commence pas au semestre 4 du M2 mais dès le début du Master car les trois semestres durent trop longtemps et c'est à cause des enseignants, responsable du Master et le chef de département informatique qui ne font pas d'efforts pour respecter l'emploi du temps annuel. |
| Je t'encourage à chercher des solutions pour le bien de tous.  |
| Noué des relations avec certaines structures informatiques pour que les étudiants puissent faire leurs mémoires en entreprise  |
| Plus d'entente, de cohérence et de la tolérance des enseignants envers les étudiants.  |
| Pour améliorer le processus, les sujets devront être disponible juste après le master 1, et que les enseignants exigent les étudiants qu'ils encadrent des rencontre à chaque quinzaine du mois pour savoir l'état d'avancement de l'étudiant.   |
| Choisir les sujets au premier semestre de M2.  |
| Je pense les enseignants devraient proposer les sujets à temps et aider les étudiants motivées   |
| avoir un système pour gérer les sujets ce qui permettra à chaque étudiant d'avoir un sujet à temps   |

## Annexe 2 : Tableau Trello de l'organisation du projet sous SCRUMBAN



### Annexe 3 : Dictionnaire de données du système

| Tables      | Champs            | Types                | Légendes                                       |
|-------------|-------------------|----------------------|--|
| Utilisateur | Id                | Entier               | Identifiant de l'utilisateur                   |
|             | Nom               | Chaîne de caractères | Nom de l'utilisateur                           |
|             | Prenom            | Chaîne de caractères | Prénom de l'utilisateur                        |
|             | Email             | Email                | Email de l'utilisateur                         |
|             | Photo             | Image                | Photo de profil de l'utilisateur               |
|             | Login             | Chaîne de caractères | Nom d'utilisateur                              |
|             | Password          | Chaîne de caractères | Mot de passe de l'utilisateur                  |
|             | Phone_number      | Entier               | Numéro de téléphone de l'utilisateur           |
| Enseignant  | Type_prof         | Chaîne de caractères | Type de l'enseignant (chercheur, vacataire...) |
|             | Grade             | Chaîne de caractères | Grade de l'enseignant                          |
|             | Specialite        | Chaîne de caractères | Specialité de l'enseignant                     |
|             | Domaine_recherche | Chaîne de caractères | Domaine de recherche de l'enseignant           |
|             | Contenu_recherche | Chaîne de caractères | Contenu de recherche de l'enseignant           |
|             | Bureau            | Chaîne de caractères | Numéro du Bureau de l'enseignant               |

|               |                      |                      |                                       |
|---------------|----------------------|----------------------|---------------------------------------|
|               | Num_matricule        | Entier               | Numéro de matricule de l'enseignant   |
| Etudiant      | Num_cart             | Entier               | Numéro de la carte d'étudiant         |
|               | Niveau               | Chaine de caractères | Niveau de l'étudiant                  |
|               | Promotion            | Entier               | Promotopn de l'étudiant               |
|               | Specialité           | Chaine de caractères | Nom du master suivi                   |
|               | Domaine_prédilection | Chaine de caractères | Domaine de predilection de l'étudiant |
| Alumni        | Situation            | Chaine de caractères | Situation de l'étudiant               |
|               | Poste                | Chaine de caractères | Poste d el'étudiant                   |
| Recruteur     | Id_rec               | Chaine de caractères | Identifiant du recruteur              |
|               | Nom_rec              | Chaine de caractères | Nom du recruteur                      |
|               | Prénom_rec           | Chaine de caractères | Prénom du recruteur                   |
|               | Email_rec            | Chaine de caractères | Email du recruteur                    |
| Sujet mémoire | Domaine              | Chaine de caractères | Domaine du sujet de mémoire           |
|               | Titre_sujet          | Chaine de caractères | Titre du sujet de mémoire             |
|               | Prérequis            | Chaine de caractères | Prérequis du sujet de mémoire         |
|               | Disponibilité        | Chaine de caractères | Disponibilité du sujet de mémoire     |
|               | Document_s           | Fichier              | Fichier détaillé du sujet de mémoire. |
| Mémoire       | Id_m                 | Entier               | Identifiant du mémoire                |

|                     |                 |                      |  |
|---------------------|-----------------|----------------------|--|
|                     | Période_début   | Date                 | Date de début de l'encadrement                         |
|                     | Avancement      | Chaine de caractères | Avancement de l'encadrement                            |
|                     | Date_fin        | Date                 | Date de fin de l'encadrement                           |
|                     | Commentaire     | Chaine de caractères | Commentaire de l'enseignant sur le trail de l'étudiant |
|                     | Etat            | Chaine de caractères | L'état d'avancement du mémoire                         |
| Dossier final       | Id_d            | Entier               | Identifiant du dossier                                 |
|                     | Note            | Chaine de caractères | Note attribuée à l'étudiant                            |
|                     | Appreciation    | Chaine de caractères | Appreciation attribuée à l'étudiant                    |
| Document de mémoire | Id_doc          | Entier               | Identifiant du document de mémoire                     |
|                     | Date_soutenance | Date                 | Date de la soutenance du mémoire                       |
|                     | Taille_fichier  | Chaine de caractères | Taille du document de mémoire                          |
|                     | Document        | Fichier              | Le fichier du document de mémoire                      |
| Entreprise          | Id_ent          | Entier               | Identifiant de l'entreprise                            |
|                     | Nom_entreprise  | Chaine de caractères | Nom de l'entreprise                                    |
|                     | Domaine         | Chaine de caractères | Domaine de l'entreprise                                |

|  |                   |                      |                                     |
|--|-------------------|----------------------|-------------------------------------|
|  | Adresse           | Chaine de caractères | Adresse de l'entreprise             |
|  | Phone_number_entr | Entier               | Numéro de téléphone de l'entreprise |

## Annexe 4 : Rapport de l'audit de la sécurité du système avec Owasp Zap

# ZAP Scanning Report

### Summary of Alerts

| Risk Level                    | Number of Alerts |
|-------------------------------|------------------|
| <a href="#">High</a>          | 0                |
| <a href="#">Medium</a>        | 0                |
| <a href="#">Low</a>           | 4                |
| <a href="#">Informational</a> | 2                |

### Alert Detail

| Low (Medium) | En-tête X-Content-Type-Options manquant   |
|--------------|---|
| Description  | L'en-tête X-Content-Type-Options contre le sniffing MIME n'est pas renseigné à 'nosniff'. Ceci permet à de vieilles versions d'Internet Explorer et de Chrome de pratiquer le sniffing MIME sur le corps de réponse, conduisant potentiellement à l'interprétation et l'affichage du contenu dans un autre type que celui déclaré. A l'heure actuelle (début 2014), les anciennes versions de Firefox utiliseront le type de contenu déclaré (au cas où un type est déterminé), plutôt qu'analyser le MIME. |
| URL          | <a href="http://127.0.0.1:8000/static/vendors/mdi/css/materialdesignicons.min.css">http://127.0.0.1:8000/static/vendors/mdi/css/materialdesignicons.min.css</a>   |
| Method       | GET   |
| Parameter    | X-Content-Type-Options  |
| URL          | <a href="http://127.0.0.1:8000/static/css/style.css">http://127.0.0.1:8000/static/css/style.css</a>   |
| Method       | GET   |
| Parameter    | X-Content-Type-Options  |
| URL          | <a href="http://127.0.0.1:8000/static/images/auth/images.jpg">http://127.0.0.1:8000/static/images/auth/images.jpg</a>   |
| Method       | GET   |
| Parameter    | X-Content-Type-Options  |
| URL          | <a href="http://127.0.0.1:8000/static/js/misc.js">http://127.0.0.1:8000/static/js/misc.js</a>   |
| Method       | GET   |
| Parameter    | X-Content-Type-Options  |

|                   |   |
|-------------------|---|
| URL               | <a href="http://127.0.0.1:8000/static/vendors/simple-line-icons/css/simple-line-icons.css">http://127.0.0.1:8000/static/vendors/simple-line-icons/css/simple-line-icons.css</a>   |
| Method            | GET   |
| Parameter         | X-Content-Type-Options  |
| URL               | <a href="http://127.0.0.1:8000/static/vendors/js/vendor.bundle.base.js">http://127.0.0.1:8000/static/vendors/js/vendor.bundle.base.js</a>   |
| Method            | GET   |
| Parameter         | X-Content-Type-Options  |
| URL               | <a href="http://127.0.0.1:8000/static/js/hoverable-collapse.js">http://127.0.0.1:8000/static/js/hoverable-collapse.js</a>   |
| Method            | GET   |
| Parameter         | X-Content-Type-Options  |
| URL               | <a href="http://127.0.0.1:8000/static/js/todolist.js">http://127.0.0.1:8000/static/js/todolist.js</a>   |
| Method            | GET   |
| Parameter         | X-Content-Type-Options  |
| URL               | <a href="http://127.0.0.1:8000/static/js/settings.js">http://127.0.0.1:8000/static/js/settings.js</a>   |
| Method            | GET   |
| Parameter         | X-Content-Type-Options  |
| URL               | <a href="http://127.0.0.1:8000/static/vendors/flag-icon-css/css/flag-icon.min.css">http://127.0.0.1:8000/static/vendors/flag-icon-css/css/flag-icon.min.css</a>   |
| Method            | GET   |
| Parameter         | X-Content-Type-Options  |
| URL               | <a href="http://127.0.0.1:8000/static/vendors/css/vendor.bundle.base.css">http://127.0.0.1:8000/static/vendors/css/vendor.bundle.base.css</a>   |
| Method            | GET   |
| Parameter         | X-Content-Type-Options  |
| URL               | <a href="http://127.0.0.1:8000/static/js/off-canvas.js">http://127.0.0.1:8000/static/js/off-canvas.js</a>   |
| Method            | GET   |
| Parameter         | X-Content-Type-Options  |
| Instances         | 12  |
| Solution          | <p>Assurez-vous que l'application ou le serveur internet renseigne l'en-tête Content-Type correctement, et que l'en-tête X-Content-Type-Options prenne la valeur 'nosniff' pour toutes les pages internet.</p> <p>Si possible, assurez-vous que l'utilisateur utilise un navigateur moderne et conforme aux standards, qui ne pratique pas du tout le sniffing MIME, ou qui puisse être commandé par l'application ou le serveur internet de manière à ne pas pratiquer le sniffing MIME.</p> |
| Other information | Ce problème s'applique toujours aux pages de type erreur (401, 403, 500, etc.), car ces pages sont encore souvent touchées par des problèmes d'injection, auquel cas il est encore possible que les navigateurs interprètent des pages autrement que selon leur type de contenu réel.   |
| Reference         | <p><a href="http://msdn.microsoft.com/en-us/library/ie/gg622941%28v=vs.85%29.aspx">http://msdn.microsoft.com/en-us/library/ie/gg622941%28v=vs.85%29.aspx</a></p> <p><a href="https://www.owasp.org/index.php/List_of_useful_HTTP_headers">https://www.owasp.org/index.php/List_of_useful_HTTP_headers</a></p>   |

|                     |   |
|---------------------|---|
| CWE Id              | 16  |
| WASC Id             | 15  |
| Source ID           | 3   |
| <b>Low (Medium)</b> | <b>Protection XSS du navigateur Internet non activée</b>  |
| Description         | La protection XSS du navigateur Internet n'est pas activée, ou est désactivée par la configuration de l'en-tête de réponse HTTP 'X-XSS-Protection' sur le serveur web |
| URL                 | http://127.0.0.1:8000/account/login/  |
| Method              | GET   |
| Parameter           | X-XSS-Protection  |
| URL                 | http://127.0.0.1:8000/account/login/  |
| Method              | POST  |
| Parameter           | X-XSS-Protection  |
| URL                 | <a href="http://127.0.0.1:8000/robots.txt">http://127.0.0.1:8000/robots.txt</a>   |
| Method              | GET   |
| Parameter           | X-XSS-Protection  |
| URL                 | http://127.0.0.1:8000/account/prof_register/  |
| Method              | POST  |
| Parameter           | X-XSS-Protection  |
| URL                 | <a href="http://127.0.0.1:8000/sitemap.xml">http://127.0.0.1:8000/sitemap.xml</a>   |
| Method              | GET   |
| Parameter           | X-XSS-Protection  |
| URL                 | http://127.0.0.1:8000/account/password_reset/done/  |
| Method              | GET   |
| Parameter           | X-XSS-Protection  |
| URL                 | http://127.0.0.1:8000/account/etudiant_register/  |
| Method              | GET   |
| Parameter           | X-XSS-Protection  |
| URL                 | http://127.0.0.1:8000/account/password_reset/   |
| Method              | GET   |
| Parameter           | X-XSS-Protection  |
| URL                 | http://127.0.0.1:8000/account/register/   |

|                     |  |
|---------------------|--|
| Method              | GET  |
| Parameter           | X-XSS-Protection   |
| URL                 | http://127.0.0.1:8000/account/etudiant_register/   |
| Method              | POST   |
| Parameter           | X-XSS-Protection   |
| URL                 | <a href="http://127.0.0.1:8000/account/login">http://127.0.0.1:8000/account/login</a>  |
| Method              | GET  |
| Parameter           | X-XSS-Protection   |
| URL                 | http://127.0.0.1:8000/account/prof_register/   |
| Method              | GET  |
| Parameter           | X-XSS-Protection   |
| Instances           | 12   |
| Solution            | Assurez-vous que le filtre XSS du navigateur internet est activé, en renseignant l'en-tête de réponse HTTP X-XSS-Protection à '1'.   |
| Other information   | <p>L'en-tête de réponse HTTP X-XSS-Protection autorise le serveur internet à activer ou désactiver le mécanisme de protection XSS du navigateur internet. Les valeurs suivantes tenteraient d'activer la protection:</p> <p>X-XSS-Protection: 1; mode = block</p> <p>X-XSS-Protection: 1; report = http://www.example.com/xss</p> <p>Les valeurs suivantes désactiveraient cette protection:</p> <p>X-XSS-Protection: 0</p> <p>L'en-tête de réponse HTTP X-XSS-Protection est actuellement supportée par Internet Explorer, Chrome et Safari (WebKit).</p> <p>Notez que cette alerte n'est déclenchée que si le corps de réponse pouvaient contenir une charge utile XSS (avec un type de contenu texte, d'une longueur différente de zéro).</p> |
| Reference           | <p><a href="https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet">https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet</a></p> <p><a href="https://blog.veracode.com/2014/03/guidelines-for-setting-security-headers/">https://blog.veracode.com/2014/03/guidelines-for-setting-security-headers/</a></p>  |
| CWE Id              | 933  |
| WASC Id             | 14   |
| Source ID           | 3  |
| <b>Low (Medium)</b> | <b>Cookie No HttpOnly Flag</b>   |
| Description         | Un cookie a été défini sans le flag HttpOnly, ce qui signifie que le cookie peut être accessible via JavaScript. Si un script malveillant peut être  |

|           |  |
|-----------|--|
|           | exécuté sur cette page, alors le cookie sera accessible et pourra être transmis à un autre site. Si il s'agit d'un cookie de session, alors le détournement de session pourra être possible. |
| URL       | <a href="http://127.0.0.1:8000/account/etudiant_register/">http://127.0.0.1:8000/account/etudiant_register/</a>  |
| Method    | POST   |
| Parameter | CsrfToken  |
| Evidence  | Set-Cookie: csrfToken  |
| URL       | <a href="http://127.0.0.1:8000/account/login">http://127.0.0.1:8000/account/login</a>  |
| Method    | GET  |
| Parameter | CsrfToken  |
| Evidence  | Set-Cookie: csrfToken  |
| URL       | <a href="http://127.0.0.1:8000/account/prof_register/">http://127.0.0.1:8000/account/prof_register/</a>  |
| Method    | GET  |
| Parameter | CsrfToken  |
| Evidence  | Set-Cookie: csrfToken  |
| URL       | <a href="http://127.0.0.1:8000/account/login/">http://127.0.0.1:8000/account/login/</a>  |
| Method    | GET  |
| Parameter | CsrfToken  |
| Evidence  | Set-Cookie: csrfToken  |
| URL       | <a href="http://127.0.0.1:8000/account/login/">http://127.0.0.1:8000/account/login/</a>  |
| Method    | POST   |
| Parameter | CsrfToken  |
| Evidence  | Set-Cookie: csrfToken  |
| URL       | <a href="http://127.0.0.1:8000/account/etudiant_register/">http://127.0.0.1:8000/account/etudiant_register/</a>  |
| Method    | GET  |
| Parameter | CsrfToken  |
| Evidence  | Set-Cookie: csrfToken  |
| URL       | <a href="http://127.0.0.1:8000/account/prof_register/">http://127.0.0.1:8000/account/prof_register/</a>  |
| Method    | POST   |
| Parameter | CsrfToken  |
| Evidence  | Set-Cookie: csrfToken  |
| URL       | <a href="http://127.0.0.1:8000/account/password_reset/">http://127.0.0.1:8000/account/password_reset/</a>  |
| Method    | GET  |
| Parameter | CsrfToken  |
| Evidence  | Set-Cookie: csrfToken  |
| Instances | 8  |
| Solution  | Veiller que le flag HttpOnly soit défini pour tous les cookies.  |

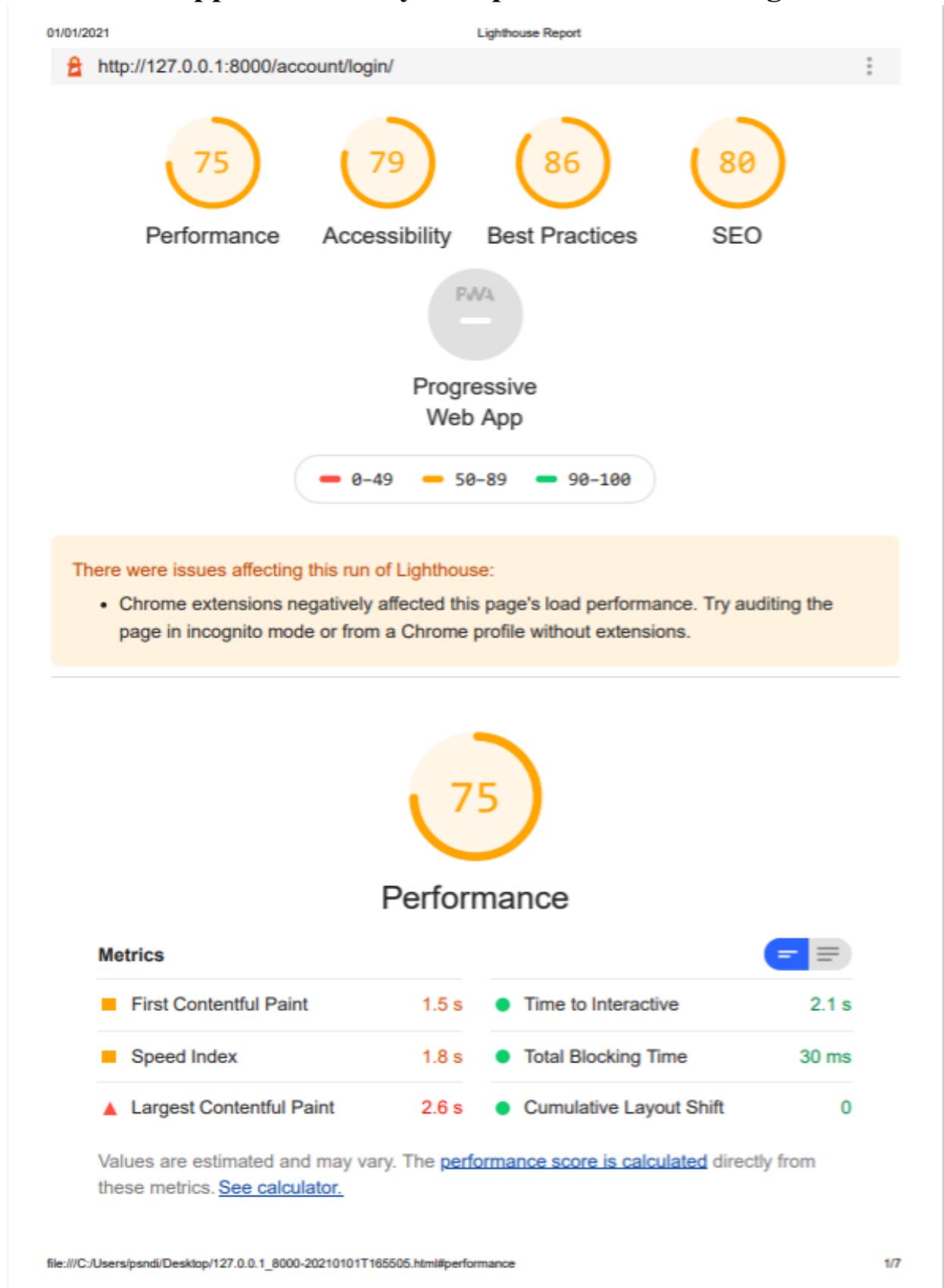
|                     |  |
|---------------------|--|
| Reference           | <a href="http://www.owasp.org/index.php/HttpOnly">http://www.owasp.org/index.php/HttpOnly</a>  |
| CWE Id              | 16   |
| WASC Id             | 13   |
| Source ID           | 3  |
| <b>Low (Medium)</b> | <b>Absence of Anti-CSRF Tokens</b>   |
| Description         | <p>No Anti-CSRF tokens were found in a HTML submission form.</p> <p>La contrefaçon de requête intersites (Cross Site Request Forgery - CSRF) est une attaque qui consiste à forcer une victime à envoyer une requête HTTP vers une destination cible, sans qu'elle n'en aie ni connaissance ni intention, afin d'effectuer une action en se faisant passer pour la victime. La cause originelle est que les fonctionnalités de l'application sont appelées à l'aide d'URL ou d'actions de formulaires prévisibles et reproductibles. La nature de l'attaque est que le CSRF exploite la confiance qu'un site internet accorde à un utilisateur. En revanche, le cross-site scripting (XSS) exploite la confiance que l'utilisateur porte à un site internet. Comme XSS, les attaques CSRF ne sont pas nécessairement multi-sites, mais elles peuvent l'être. La contrefaçon de requête intersite est également connue sous les noms CSRF, XSRF, attaque en un clic (one-click attack), session riding, confused deputy et sea surf.</p> <p>Les attaques CSRF sont efficaces dans de nombreuses situations, notamment:</p> <ul style="list-style-type: none"> <li>* quand la victime a une session active sur le site cible.</li> <li>* quand la victime est authentifiée via HTTP auth sur le site cible.</li> <li>* quand la victime est sur le même réseau local que le site cible.</li> </ul> <p>CSRF a d'abord été utilisée pour effectuer une action contre un site cible en utilisant les privilèges de la victime, mais des techniques récentes permettent d'avoir accès à des renseignements en accédant à la réponse. Le risque de divulgation d'informations est considérablement augmenté lorsque le site cible est vulnérable aux XSS, parce que XSS peut être utilisé comme une plateforme pour CSRF, permettant à l'attaque d'opérer dans les limites de la politique de même origine.</p> |
| URL                 | http://127.0.0.1:8000/account/register/  |
| Method              | GET  |
| Evidence            | <form >  |
| Instances           | 1  |
| Solution            | <p>Phase: Architecture et Design</p> <p>Utilisez une librairie ou un framework approuvé qui ne permet pas cette vulnérabilité, ou qui contient des fonctionnalités permettant d'éviter plus facilement cette vulnérabilité.</p> <p>Utilisez par exemple des bibliothèques anti-CSRF telles que CSRFGuard de l'OWASP.</p>   |

|                               |   |
|-------------------------------|---|
|                               | <p>Phase: Implémentation</p> <p>Assurez-vous que votre application soit exempte de problèmes de cross-site scripting, parce que la plupart des défenses contre le CSRF peuvent être contournées en utilisant des scripts contrôlés par le pirate.</p> <p>Phase: Architecture et Design</p> <p>Générez une valeur à usage unique pour chaque formulaire, placez-la dans le formulaire et vérifiez-la à la réception du formulaire. Assurez-vous que cette valeur unique ne soit pas prévisible (CWE-330).</p> <p>Notez que ceci peut aussi être contourné en utilisant XSS.</p> <p>Identifiez les opérations particulièrement dangereuses. Quand l'utilisateur exécute une opération dangereuse, envoyez une requête de confirmation distincte pour vérifier que l'utilisateur veut effectivement effectuer cette opération.</p> <p>Notez que ceci peut aussi être contourné en utilisant XSS.</p> <p>Utilisez la librairie de gestion de session ESAPI.</p> <p>Cette librairie comprend un composant pour le contrôle de CSRF.</p> <p>N'utilisez pas la méthode GET pour les requêtes entraînant un changement d'état.</p> <p>Phase: Implémentation</p> <p>Vérifiez l'en-tête HTTP Referer pour voir si la requête provient d'une page attendue. Ceci pourrait toutefois restreindre la fonctionnalité de l'application, car les utilisateurs ou les serveurs proxy pourraient avoir désactivé le renvoi du HTTP Referer pour des raisons de confidentialité.</p> |
| Other information             | No known Anti-CSRF token [anticsrf, CSRFToken, __RequestVerificationToken, csrfmiddlewaretoken, authenticity_token, OWASP_CSRFTOKEN, anoncsrf, csrf_token, _csrf, _csrfSecret] was found in the following HTML form: [Form 1: ].  |
| Reference                     | <a href="http://projects.webappsec.org/Cross-Site-Request-Forgery">http://projects.webappsec.org/Cross-Site-Request-Forgery</a><br><a href="http://cwe.mitre.org/data/definitions/352.html">http://cwe.mitre.org/data/definitions/352.html</a>  |
| CWE Id                        | 352   |
| WASC Id                       | 9   |
| Source ID                     | 3   |
| <b>Informational (Medium)</b> | <b>Information Disclosure - Suspicious Comments</b>   |
| Description                   | The response appears to contain suspicious comments which may help an attacker.   |
| URL                           | http://127.0.0.1:8000/account/prof_register/  |
| Method                        | GET   |

|                            |   |
|----------------------------|---|
| URL                        | http://127.0.0.1:8000/account/login/  |
| Method                     | POST  |
| URL                        | http://127.0.0.1:8000/account/login/  |
| Method                     | GET   |
| URL                        | <a href="http://127.0.0.1:8000/static/vendors/js/vendor.bundle.base.js">http://127.0.0.1:8000/static/vendors/js/vendor.bundle.base.js</a>         |
| Method                     | GET   |
| URL                        | http://127.0.0.1:8000/account/password_reset/done/  |
| Method                     | GET   |
| URL                        | http://127.0.0.1:8000/account/etudiant_register/  |
| Method                     | GET   |
| URL                        | http://127.0.0.1:8000/account/password_reset/   |
| Method                     | GET   |
| URL                        | <a href="http://127.0.0.1:8000/account/login">http://127.0.0.1:8000/account/login</a>   |
| Method                     | GET   |
| URL                        | http://127.0.0.1:8000/account/prof_register/  |
| Method                     | POST  |
| URL                        | <a href="http://127.0.0.1:8000/static/js/todolist.js">http://127.0.0.1:8000/static/js/todolist.js</a>   |
| Method                     | GET   |
| URL                        | http://127.0.0.1:8000/account/register/   |
| Method                     | GET   |
| URL                        | http://127.0.0.1:8000/account/etudiant_register/  |
| Method                     | POST  |
| Instances                  | 12  |
| Solution                   | Remove all comments that return information that may help an attacker and fix any underlying problems they refer to.                              |
| Other information          | <!-- Mirrored from www.urbanui.com/victory/pages/samples/login.html by HTTrack Website Copier/3.x [XR&CO'2014], Thu, 13 Feb 2020 14:28:09 GMT --> |
| Reference                  |   |
| CWE Id                     | 200   |
| WASC Id                    | 13  |
| Source ID                  | 3   |
| <b>Informational (Low)</b> | <b>Timestamp Disclosure – Unix</b>  |
| Description                | A timestamp was disclosed by the application/web server – Unix  |
| URL                        | <a href="http://127.0.0.1:8000/account/login">http://127.0.0.1:8000/account/login</a>   |
| Method                     | GET   |
| Evidence                   | 31449600  |

|                   |  |
|-------------------|--|
| URL               | http://127.0.0.1:8000/account/password_reset/  |
| Method            | GET  |
| Evidence          | 31449600   |
| URL               | http://127.0.0.1:8000/account/etudiant_register/   |
| Method            | GET  |
| Evidence          | 31449600   |
| URL               | http://127.0.0.1:8000/account/prof_register/   |
| Method            | POST   |
| Evidence          | 31449600   |
| URL               | http://127.0.0.1:8000/account/prof_register/   |
| Method            | GET  |
| Evidence          | 31449600   |
| URL               | http://127.0.0.1:8000/account/login/   |
| Method            | GET  |
| Evidence          | 31449600   |
| URL               | http://127.0.0.1:8000/account/etudiant_register/   |
| Method            | POST   |
| Evidence          | 31449600   |
| URL               | http://127.0.0.1:8000/account/login/   |
| Method            | POST   |
| Evidence          | 31449600   |
| Instances         | 8  |
| Solution          | Manually confirm that the timestamp data is not sensitive, and that the data cannot be aggregated to disclose exploitable patterns.  |
| Other information | 31449600, which evaluates to: 1970-12-31 00:00:00  |
| Reference         | <a href="https://www.owasp.org/index.php/Top_10_2013-A6-Sensitive_Data_Exposure">https://www.owasp.org/index.php/Top_10_2013-A6-Sensitive_Data_Exposure</a><br><a href="http://projects.webappsec.org/w/page/13246936/Information%20Leakage">http://projects.webappsec.org/w/page/13246936/Information%20Leakage</a> |
| CWE Id            | 200  |
| WASC Id           | 13   |
| Source ID         | 3  |

## Annexe 5 : rapport de l'analyse de performance avec lighthouse



01/01/2021 Lighthouse Report

**Opportunities** — These suggestions can help your page load faster. They don't [directly affect](#) the Performance score.

| Opportunity                           | Estimated Savings |
|---------------------------------------|-------------------|
| ▲ Eliminate render-blocking resources | 1.03 s            |
| ▲ Remove unused CSS                   | 0.92 s            |
| ■ Remove unused JavaScript            | 0.2 s             |

**Diagnostics** — More information about the performance of your application. These numbers don't [directly affect](#) the Performance score.

- ▲ Ensure text remains visible during webfont load
- ▲ Image elements do not have explicit [width](#) and [height](#)
- ▲ Serve static assets with an efficient cache policy — 14 resources found
- Avoid chaining critical requests — 12 chains found
- Keep request counts low and transfer sizes small — 20 requests • 1,999 KiB
- Largest Contentful Paint element — 1 element found
- Avoid large layout shifts — 2 elements found
- Avoid long main-thread tasks — 6 long tasks found

**Passed audits (24)**

79

file:///C:/Users/psndi/Desktop/127.0.0.1\_8000-20210101T165505.html#performance 2/7

01/01/2021

Lighthouse Report

## Accessibility

These checks highlight opportunities to [improve the accessibility of your web app](#). Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

**Contrast** — These are opportunities to improve the legibility of your content.

▲ Background and foreground colors do not have a sufficient contrast ratio. ▾

**Navigation** — These are opportunities to improve keyboard navigation in your application.

▲ Heading elements are not in a sequentially-descending order ▾

**Names and labels** — These are opportunities to improve the semantics of the controls in your application. This may enhance the experience for users of assistive technology, like a screen reader.

▲ Image elements do not have `[alt]` attributes ▾

**Additional items to manually check (10)** — These items address areas which an automated testing tool cannot cover. Learn more in our [guide on conducting an accessibility review](#). ▾

**Passed audits (9)** ▾

**Not applicable (29)** ▾

86

file:///C:/Users/psndi/Desktop/127.0.0.1\_8000-20210101T165505.html#performance

37

The image is a screenshot of a Lighthouse Report. At the top left, it shows the date '01/01/2021'. The main title is 'Best Practices' with a large orange circle around it. Below this, there are three sections: 'Trust and Safety', 'General', and 'Passed audits (13)'. The 'Trust and Safety' section has one item: 'Includes front-end JavaScript libraries with known security vulnerabilities — 3 vulnerabilities detected'. The 'General' section has one item: 'Browser errors were logged to the console'. The 'Passed audits (13)' section is collapsed. Below these sections, there is a large orange circle with the number '80' inside, labeled 'SEO'. A paragraph explains that these checks ensure the page is optimized for search engine results ranking. Below this, there is a section for 'Content Best Practices' with two items: 'Document does not have a meta description' and 'Image elements do not have [alt] attributes'. At the bottom left, the file path is shown: 'file:///C:/Users/psndi/Desktop/127.0.0.1\_8000-20210101T165505.html#performance'. At the bottom right, the page number '4/7' is visible.

01/01/2021

Lighthouse Report

# Best Practices

## Trust and Safety

- ▲ Includes front-end JavaScript libraries with known security vulnerabilities — 3 vulnerabilities detected

## General

- ▲ Browser errors were logged to the console

## Passed audits (13)

---

# 80

## SEO

These checks ensure that your page is optimized for search engine results ranking. There are additional factors Lighthouse does not check that may affect your search ranking. [Learn more.](#)

### Content Best Practices

Format your HTML in a way that enables crawlers to better understand your app's content.

- ▲ Document does not have a meta description
- ▲ Image elements do not have `[alt]` attributes

file:///C:/Users/psndi/Desktop/127.0.0.1\_8000-20210101T165505.html#performance

4/7

01/01/2021

Lighthouse Report

**Additional items to manually check (1)** — Run these additional validators on your site to check additional SEO best practices.

**Passed audits (8)**

**Not applicable (4)**



## Progressive Web App

These checks validate the aspects of a Progressive Web App. [Learn more.](#)

### Fast and reliable

Page load is not fast enough on mobile networks

▲ Your page loads too slowly and is not interactive within 10 seconds. Look at the opportunities and diagnostics in the "Performance" section to learn how to improve.

— Interactive on simulated mobile network at 11.2 s

▲ Current page does not respond with a 200 when offline

start\_url does not respond with a 200 when offline

▲ No usable web app manifest found on page.

### Installable

● Uses HTTPS

▲ Does not register a service worker that controls page and start\_url

file:///C:/Users/psndi/Desktop/127.0.0.1\_8000-20210101T165505.html#performance

5/7

01/01/2021

Lighthouse Report

▲ Web app manifest does not meet the installability requirements  
Failures: No manifest was fetched.

### ★ PWA Optimized

▲ Does not redirect HTTP traffic to HTTPS

▲ Is not configured for a custom splash screen Failures: No manifest was fetched.

Does not set a theme color for the address bar.  
▲ Failures: No manifest was fetched, No `

● Content is sized correctly for the viewport

● Has a `<meta name="viewport">` tag with `width` or `initial-scale`

● Contains some content when JavaScript is not available

▲ Does not provide a valid `apple-touch-icon`

▲ Manifest doesn't have a maskable icon No manifest was fetched

**Additional items to manually check (3)** — These checks are required by the baseline [PWA Checklist](#) but are not automatically checked by Lighthouse. They do not affect your score but it's important that you verify them manually.

### Runtime Settings

|                    |   |
|--------------------|---|
| URL                | http://127.0.0.1:8000/account/login/              |
| Fetch Time         | Jan 1, 2021, 4:55 PM GMT+1                        |
| Device             | Emulated Desktop                                  |
| Network throttling | 40 ms TCP RTT, 10,240 Kbps throughput (Simulated) |
| CPU throttling     | 1x slowdown (Simulated)                           |
| Channel            | devtools  |

file:///C:/Users/psndi/Desktop/127.0.0.1\_8000-20210101T165505.html#performance

6/7

## Annexe 6 : Documentation technique du système pour le déploiement local et sur le cloud (heroku)

- **Déploiement local**

**Prérequis :** - Avoir une connexion internet

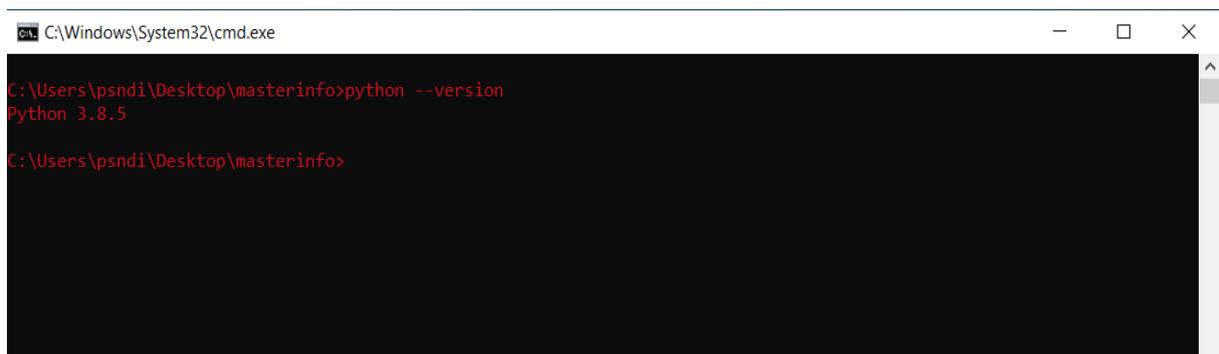
- Disposer du code source de l'application (dossier masterinfo)

**NB :** le déploiement a été réalisé sur un système d'exploitation Windows (pour linux et MacOS c'est le même procédé à l'exception des commandes qui permettent de se placer sur les répertoires).

Notre application est développée avec la version 3 de Django et python. Ainsi pour le déployer sur un environnement local, on a besoin d'installer l'ensemble des outils nécessaire à son bon fonctionnement. Tout d'abord on commence par installer python.

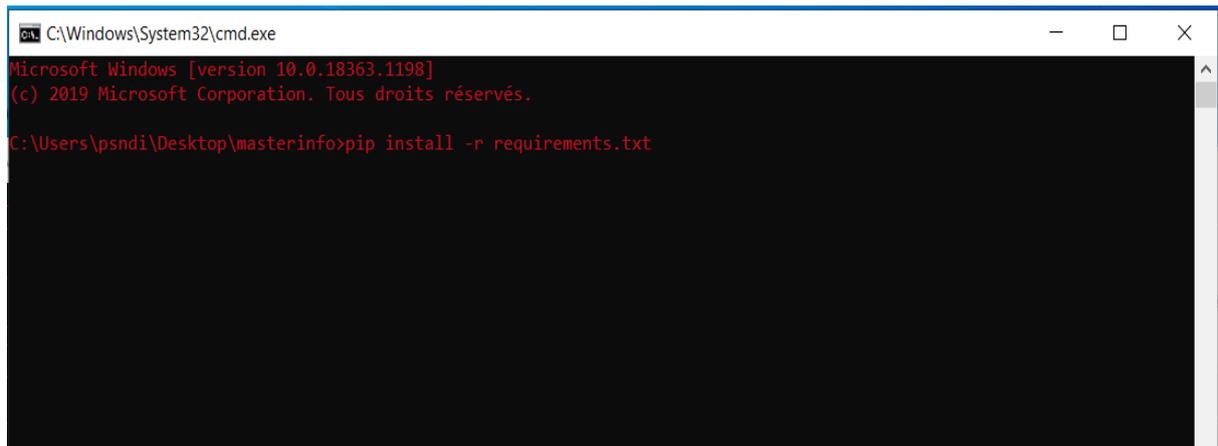
On se rend sur le site du projet python et on télécharge la version la plus récente de python 3

<https://www.python.org/downloads/>. On suit les étapes de l'installation en prenant soin de cocher la case **créer une variable d'environnement (ce qui permettra à l'invité de commande de reconnaître les commandes python)**. A la fin on saisit la commande *python --version* pour vérifier que l'installation s'est bien passée et la version de python installée est adéquate.



```
C:\Windows\System32\cmd.exe
C:\Users\psndi\Desktop\masterinfo>python --version
Python 3.8.5
C:\Users\psndi\Desktop\masterinfo>
```

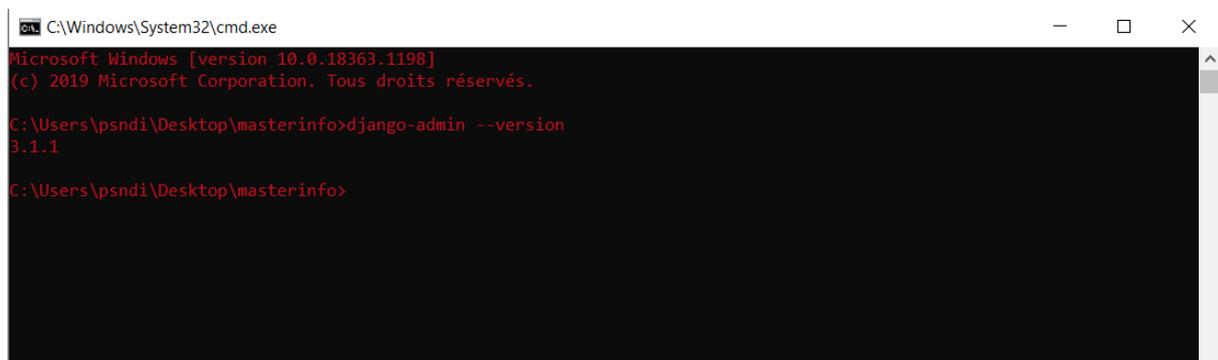
Ensuite on se place sur le répertoire du projet (master info) et on ouvre un invité de commande, on saisit la commande *pip install -r requirements.txt* qui exécute le fichier requirements.txt permettant d'installer l'ensemble des dépendances du projet y compris Django.



```
C:\Windows\System32\cmd.exe
Microsoft Windows [version 10.0.18363.1198]
(c) 2019 Microsoft Corporation. Tous droits réservés.

C:\Users\psndi\Desktop\masterinfo>pip install -r requirements.txt
```

Par la suite on peut vérifier la version de Django en saisissant la commande ***Django-admin --version***

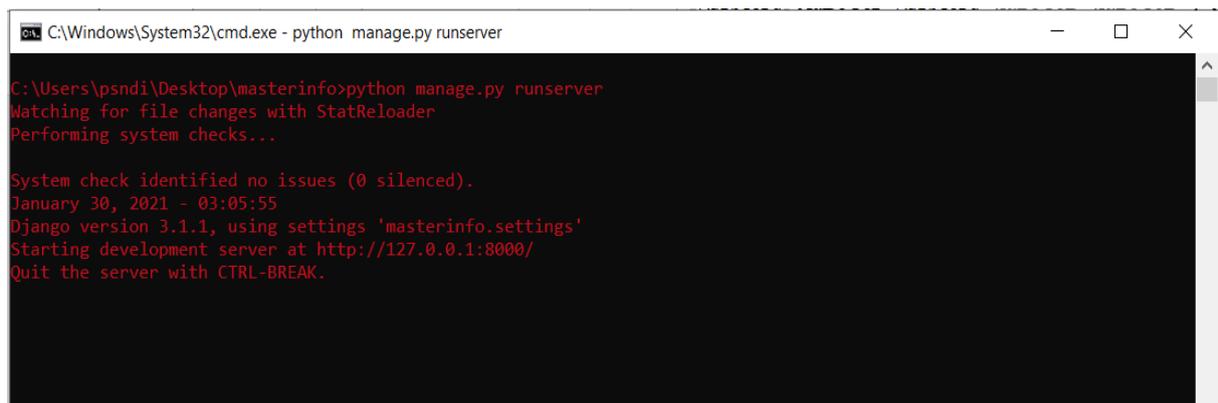


```
C:\Windows\System32\cmd.exe
Microsoft Windows [version 10.0.18363.1198]
(c) 2019 Microsoft Corporation. Tous droits réservés.

C:\Users\psndi\Desktop\masterinfo>django-admin --version
3.1.1

C:\Users\psndi\Desktop\masterinfo>
```

Après l'installation des dépendances on peut lancer la commande ***python manage.py runserver*** qui permet de démarrer le serveur local de Django



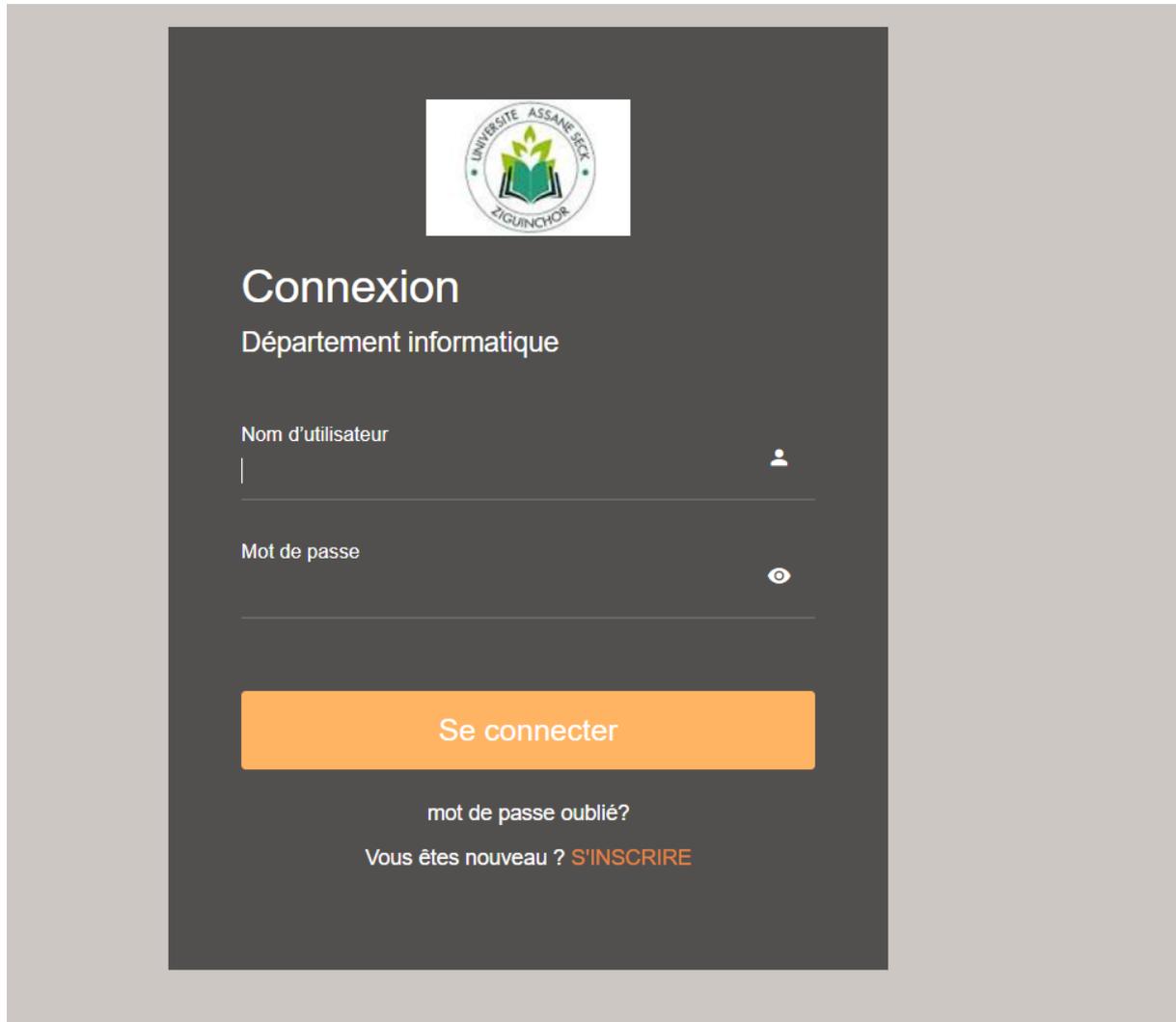
```
C:\Windows\System32\cmd.exe - python manage.py runserver
C:\Users\psndi\Desktop\masterinfo>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
January 30, 2021 - 03:05:55
Django version 3.1.1, using settings 'masterinfo.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

On constate que l'adresse du serveur local est ***127.0.0.1:8000***. L'adresse ***127.0.0.1:8000/account/login*** étant celle de la page de connexion. On accède à l'adresse sur un navigateur web et on se connecte avec les identifiants suivants pour visiter le système

***Login*** : admin

**Mot de passe** : admin.



UNIVERSITE ASSANE SECK  
ZIGUINCHOR

## Connexion

Département informatique

Nom d'utilisateur 

Mot de passe 

**Se connecter**

[mot de passe oublié?](#)

Vous êtes nouveau ? [S'INSCRIRE](#)

On peut aussi créer un compte enseignant ou étudiant et attendre la confirmation du responsable de master.

- **Déploiement sur le cloud (Hérouku)**

### Prérequis :

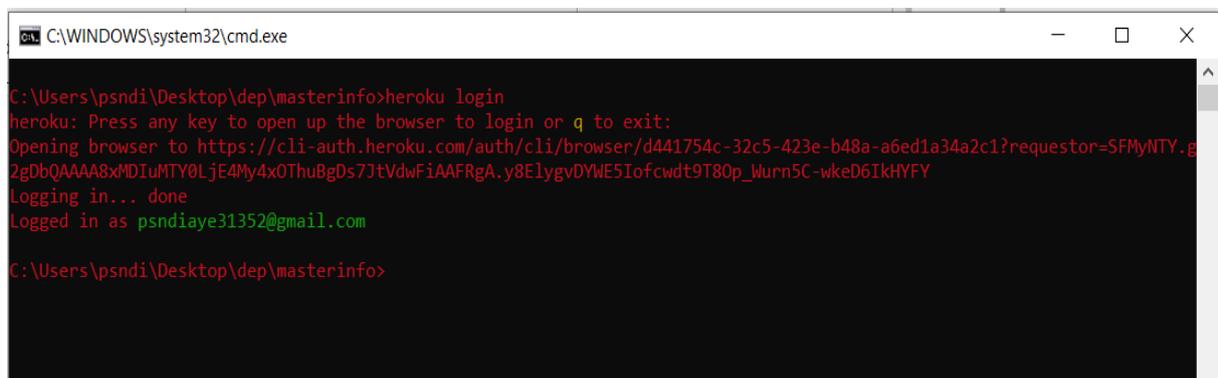
- Connexion internet
- Git
- Heroku CLI
- Avoir un compte d'utilisateur sur le site web de heroku

Tout d'abord nous devons nous rendre sur le site de Heroku <https://dashboard.heroku.com/> pour créer un compte d'utilisateur.

### Installer Heroku CLI (Heroku Toolbelt)

Heroku fournit une interface de ligne de commande facile à utiliser appelée Heroku Toolbelt. C'est un outil de ligne de commande pour gérer vos applications Heroku. Le site web <https://devcenter.heroku.com/articles/heroku-cli> permet de télécharger le CLI d'heroku.

Après l'installation, on ouvre un terminal et on se connecte à notre compte Heroku avec la commande *heroku login* en utilisant les informations d'identification avec lesquelles on s'est inscrit.



```
C:\WINDOWS\system32\cmd.exe
C:\Users\psndi\Desktop\dep\masterinfo>heroku login
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/d441754c-32c5-423e-b48a-a6ed1a34a2c1?requestor=SFMyNTY.g2gDbQAAAA8xMDIuMTY0LjE4My4x0ThuBgDs7JtVdwFIAAFRGA.y8ElygvDYWE5Iofcwdt9T80p_wurn5C-wkeD6IkHYFY
Logging in... done
Logged in as psndiaye31352@gmail.com
C:\Users\psndi\Desktop\dep\masterinfo>
```

Avant de déployer notre application sur Heroku, nous devons ajouter quelques fichiers de configuration et installer des packages pour exécuter notre application dans l'environnement de production.

- Ajoutons un **Procfile** dans le répertoire racine du projet pour définir les types de processus et déclarer explicitement la commande à exécuter pour démarrer votre application. *touch Procfile*

Ouvrons le Procfile et ajoutons la ligne ci-dessous :

*web: gunicorn masterinfo.wsgi --log-file -*

- Ajoutons un fichier **runtime.txt** dans le répertoire racine du projet et spécifiez la version Python correcte.

Ouvrons le fichier runtime.txt et ajoutez la ligne suivante, ou selon le cas, à votre application la version de python correspondante en veillant à ce qu'elle soit une version 3 de python.

*python-3.8.5*

- Installons les packages suivants dans l'environnement à l'aide de la commande

***pip install gunicorn dj-database-url whitenoise psycopg2-binary***

- Ajoutons les packages dans notre fichier requirements.txt situé à ma racine du projet

gunicorn==19.7.1

psycopg2-binary==2.8.3

whitenoise

dj-database-url

Ajoutons le middleware whitenoise sur la liste des middlewares dans settings.py du dossier principal du projet masterinfo

```
MIDDLEWARE = [  
    'django.middleware.security.SecurityMiddleware',  
    ✕ 'whitenoise.middleware.WhiteNoiseMiddleware',  
    'django.contrib.sessions.middleware.SessionMiddleware',  
    'django.middleware.common.CommonMiddleware',  
    'django.middleware.csrf.CsrfViewMiddleware',  
    'django.contrib.auth.middleware.AuthenticationMiddleware',  
    'django.contrib.messages.middleware.MessageMiddleware',  
    'django.middleware.clickjacking.XFrameOptionsMiddleware',  
]
```

Mettre à jour la configuration de la base de données dans settings.py (au bas du fichier)

```
import dj_database_url |  
prod_db = dj_database_url.config(conn_max_age=500)  
DATABASES['default'].update(prod_db)
```

Créons une application dans Heroku à partir du terminal avec la commande ***heroku create*** « ***app name*** »

```
C:\Users\psndi\Desktop\dep\masterinfo>heroku create memoireinfo  
Creating ☑ memoireinfo... done  
https://memoireinfo.herokuapp.com/ | https://git.heroku.com/memoireinfo.git  
C:\Users\psndi\Desktop\dep\masterinfo>
```

Ajoutons le nom de domaine de notre application à ALLOWED\_HOSTS dans le fichier settings.py.

```
ALLOWED_HOSTS = ['127.0.0.1' , 'localhost' , 'c46c70ffffea.ngrok.io', 'memoireinfo.herokuapp.com']
```

Initialisons Git et connectons notre nouvelle application au référentiel distant Heroku Git.

*git init*

*heroku git:remote -a herokudjangoapp*

```
C:\Users\psndi\Desktop\dep\masterinfo>git init
Reinitialized existing Git repository in C:/Users/psndi/Desktop/dep/masterinfo/.git/

C:\Users\psndi\Desktop\dep\masterinfo>heroku git:remote -a memoireinfo
set git remote heroku to https://git.heroku.com/memoireinfo.git

C:\Users\psndi\Desktop\dep\masterinfo>
```

Ajoutons des fichiers à la zone de préparation et validons les modifications.

*git add .*

*git commit -m "premier commit"*

```
C:\Users\psndi\Desktop\dep\masterinfo>git add .

C:\Users\psndi\Desktop\dep\masterinfo>git commit -m "Initial commit"
[master 627006b] Initial commit
 1 file changed, 1 insertion(+), 1 deletion(-)

C:\Users\psndi\Desktop\dep\masterinfo>
```

Poussons le projet vers le référentiel distant (déployer l'application sur Heroku)

*git push heroku master*

```
remote:          1038 static files copied to '/tmp/build_588iad15/collected_static'.
remote:
remote: -----> Discovering process types
remote:          Procfile declares types -> web
remote:
remote: -----> Compressing...
remote:          Done: 102.2M
remote: -----> Launching...
remote:          Released v5
remote:          https://memoireinfo.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/memoireinfo.git
 * [new branch]      master -> master

C:\Users\psndi\Desktop\dep\masterinfo>
```

Migrer la base de données avec la commande

*heroku run python manage.py migrate*

```
C:\Users\psndi\Desktop\dep\masterinfo>heroku run python manage.py migrate
Running python manage.py migrate on memoireinfo... up, run.9098 (Free)
Operations to perform:
  Apply all migrations: account, admin, auth, contenttypes, crud, sessions, sujetmemoire
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0001_initial... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying account.0001_initial... OK
  Applying account.0002_auto_20210127_1600... OK
  Applying account.0003_auto_20210127_1604... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying crud.0001_initial... OK
  Applying crud.0002_auto_20171013_1245... OK
  Applying crud.0003_auto_20201017_1924... OK
  Applying sessions.0001_initial... OK
  Applying sujetmemoire.0001_initial... OK

C:\Users\psndi\Desktop\dep\masterinfo>
```

À ce stade, la mise en production de notre système est terminée. On peut visiter [memoireinfo.herokuapp.com/account/login/](http://memoireinfo.herokuapp.com/account/login/) pour se connecter à notre système.

