



UNIVERSITÉ MOHAMMED V
ÉCOLE NATIONALE SUPÉRIEURE
D'INFORMATIQUE ET D'ANALYSE
DES SYSTÈMES
ENSIAS - Rabat



FILIÈRE GÉNIE LOGICIEL (GL)

RAPPORT DU PROJET DE L'INGÉNIERIE DU WEB

Conception et réalisation d'une plateforme estudiantine d'organisation et de révision des cours

Auteurs :

AARAB OUSSAMA
AABANE ABDERRAHIM
ANFAR ASMAA
LOUMEDENE SALMA

Sous la direction de :
Mr EL HAMLAOUI MAHMOUD

Remerciements

Avant de rentrer dans le vif du sujet, nous profitons de l'occasion pour remercier du fond du cœur tous ceux qui ont participé de près ou de loin à la réalisation de cet humble travail. Nos premiers remerciements vont tout d'abord à Mr Mahmoud EL HAMLAOUI qui nous a formé et accompagné tout au long de notre travail avec beaucoup de patience et de pédagogie.

Un merci bien particulier adressé également à nos professeurs pour leurs remarques, leurs directives et l'intérêt qu'ils portent aux étudiants. Nous les remercions sincèrement pour leur suivi et leur orientation.

Table des matières

Table des matières	2
Table des figures	4
1 Contexte général du projet	6
1.1 Présentation du projet :	7
1.2 Methodologie de travail :	7
1.3 Conclusion :	8
2 Analyse et Conception	9
2.1 Diagramme des cas d'utilisations :	10
2.2 Diagramme de séquences :	11
2.2.1 Accéder à un module :	11
2.2.2 Ajouter à un module :	12
2.3 Diagramme d'activité :	13
2.4 Diagramme de classes :	16
2.5 Maquettes du projet :	16
2.6 Conclusion :	19
3 Réalisation	20
3.1 Architecture du projet : Pattern MVC	21
3.2 Outils de développement :	22
3.2.1 Langage de programmation :	22
3.2.2 Outil de gestion de projet :	22
3.2.3 Environnement de développement :	22
3.2.4 Gestion de version :	22
3.2.5 Système de gestion de base de données :	23
3.2.6 Front-end :	23
3.2.7 Serveur de l'application :	24
3.2.8 Outils de test :	24
3.2.9 Outil de virtualisation :	24
3.2.10 Plateforme de déploiement :	25
3.3 Présentation de l'application :	25
3.3.1 Page d'accueil	26
3.3.2 Page d'inscription	26
3.3.3 Page d'authentification	27
3.3.4 Profil	28
3.3.5 Page de modules	28
3.3.6 Page de documents	29

TABLE DES MATIÈRES

3.3.7	Todo liste	30
3.3.8	Calendrier	31
3.3.9	Espace administrateur	31
3.4	Conclusion :	32
	Références	34

Table des figures

1	Cycle Agile	7
2	Utilisateur et Admin UC	11
3	Diagramme de séquences : Accéder à un module	12
4	Diagramme de séquences : Ajouter un module	13
5	Diagramme d'activité : Espace administrateur	14
6	Diagramme d'activité : Espace étudiant	15
7	Diagramme de classes	16
8	Maquette Ajouter Examen	19
9	Design Pattern MVC.	21
10	Page d'accueil.	26
11	Page d'inscription.	27
12	Page d'authentification.	27
13	Profil.	28
14	Page de modules.	29
15	Page de documents.	30
16	Todo liste.	30
17	Calendrier.	31
18	Espace admin.	32

Introduction générale :

Dans le cadre de sa formation innovante et solide, l'ENSIAS donne l'opportunité à ses élèves ingénieurs de s'engager dans des projets, favorisant le travail en groupe encadré par des enseignants chercheurs pour qu'ils puissent développer toute sorte de connaissances acquises en différents langages de programmation.

En deuxième année de notre filière génie logiciel nous sommes appelés à travailler sur un projet Java EE à travers lequel nous exploitons nos connaissances et compétences acquis durant notre formation Développement et Ingénierie Web afin d'aboutir à une application Web basé complètement sur Java EE bien construite.

Le présent document permettra de synthétiser les résultats de ce projet. Ce rapport sera donc articulé autour de trois grands chapitres : Le premier est consacré au contexte général du projet, le deuxième chapitre traitera à la partie conception et analyse où nous allons décortiquer le contenu des différents diagrammes des cas d'utilisation ,de séquence et de classe qui concernent notre projet, finalement pour le troisième chapitre nous allons entamer la réalisation, où nous allons préciser les outils de travail et présenter graphiquement les pages de notre site web à travers des captures d'écran.

Chapitre 1

Contexte général du projet

Dans ce chapitre nous allons présenter le contexte général de notre projet ainsi que ses objectifs. Nous aborderons par la suite la méthode adoptée pour le réaliser.

1.1 Présentation du projet :

Il s'agit d'une application web de gestion des documents des différents éléments de module dédié aux élèves ingénieurs de l'ENSIAS. L'application permet également de créer des Todo listes aidant à la mémorisation des tâches qu'un étudiant doit effectuer et qu'il peut visualiser à travers un calendrier réunissant les dates des examens des modules et les tâches précisées par cet étudiant.

Cette application vient pour faciliter la révision à ses utilisateurs en leur assurant un espace de centralisation des ressources de chaque élément (cours, td, tp...), ainsi qu'un meilleure organisation à partir des todo listes.

1.2 Methodologie de travail :

Nous avons choisi comme méthodologie de travail la méthode agile, puisqu'elle est plus efficace et moins rigide que les méthodes classiques. Elle place les besoins au centre des priorités du projet ce qui offre une plus grande flexibilité et une meilleure visibilité dans la gestion du projet à partir d'une décomposition de ce dernier en des cycles très courts.



FIGURE 1 – Cycle Agile

1.3 Conclusion :

Le long de cette partie on a présenté le contexte général du projet en introduisant sa description et en précisant ses objectifs. Dans le chapitre suivant nous allons étudier et analyser le projet et élaborer par la suite la conception du travail.

Chapitre 2

Analyse et Conception

Cette étape définit généralement les structures et les modèles à suivre lors de la phase d'implémentation de l'application. C'est la phase où nous préparons l'architecture du projet, et où nous définissons la structure de l'application. Par souci de clarté, nous débuterons par une phase préliminaire où sera présentés les diagrammes des cas d'utilisation. Ensuite, nous poursuivrons par quelques diagrammes de séquence du système suivis par le diagramme de classe qui illustrera le processus de fonctionnement du dit système.

2.1 Diagramme des cas d'utilisations :

Un diagramme de cas d'utilisation capture le comportement d'un système, d'un sous-système, d'une classe ou d'un composant tel qu'un utilisateur extérieur le voit. Les cas d'utilisation permettent d'exprimer le besoin des utilisateurs d'un système, ils sont donc une vision orientée utilisateur de ce besoin au contraire d'une vision informatique.

L'application tient en compte deux acteurs :

1. Utilisateur :

- S'inscrire
- Accéder à un module
- Consulter le calendrier
- Ajouter une tache dans la Todo liste

2. Admin :

- S'inscrire
- Accéder à un module
- Consulter le calendrier
- Ajouter une tache dans la Todo liste
- Créer un module
- Modifier un module

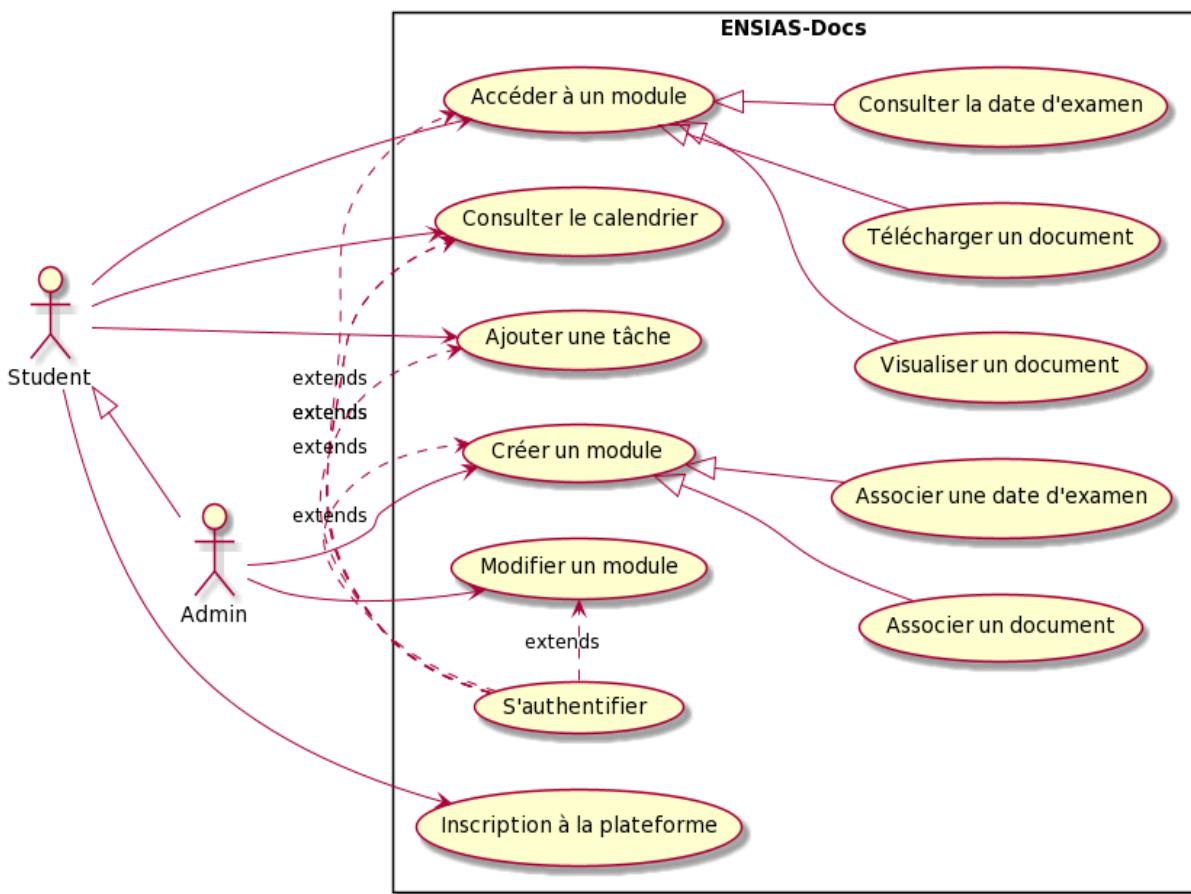


FIGURE 2 – Utilisateur et Admin UC

2.2 Diagramme de séquences :

Pour schématiser la vue comportementale de notre système informatique, nous faisons recours au diagramme de séquence d'UML. Ce diagramme permet de présenter les interactions entre l'acteur et le système avec des messages présentés dans un ordre chronologique. Ainsi nous établissons :

- Le diagramme de séquence 1 : "Accéder à un module"
- Le diagramme de séquence 2 : "Ajouter un module"

2.2.1 Accéder à un module :

- Description : Ce premier cas d'utilisation permet de chercher dans la liste des modules, en choisir un, et puis visualiser ou télécharger son contenu. au panier.
- Acteur : L'utilisateur de l'application
- Pré-conditions : L'utilisateur doit posséder un compte et doit s'authentifier

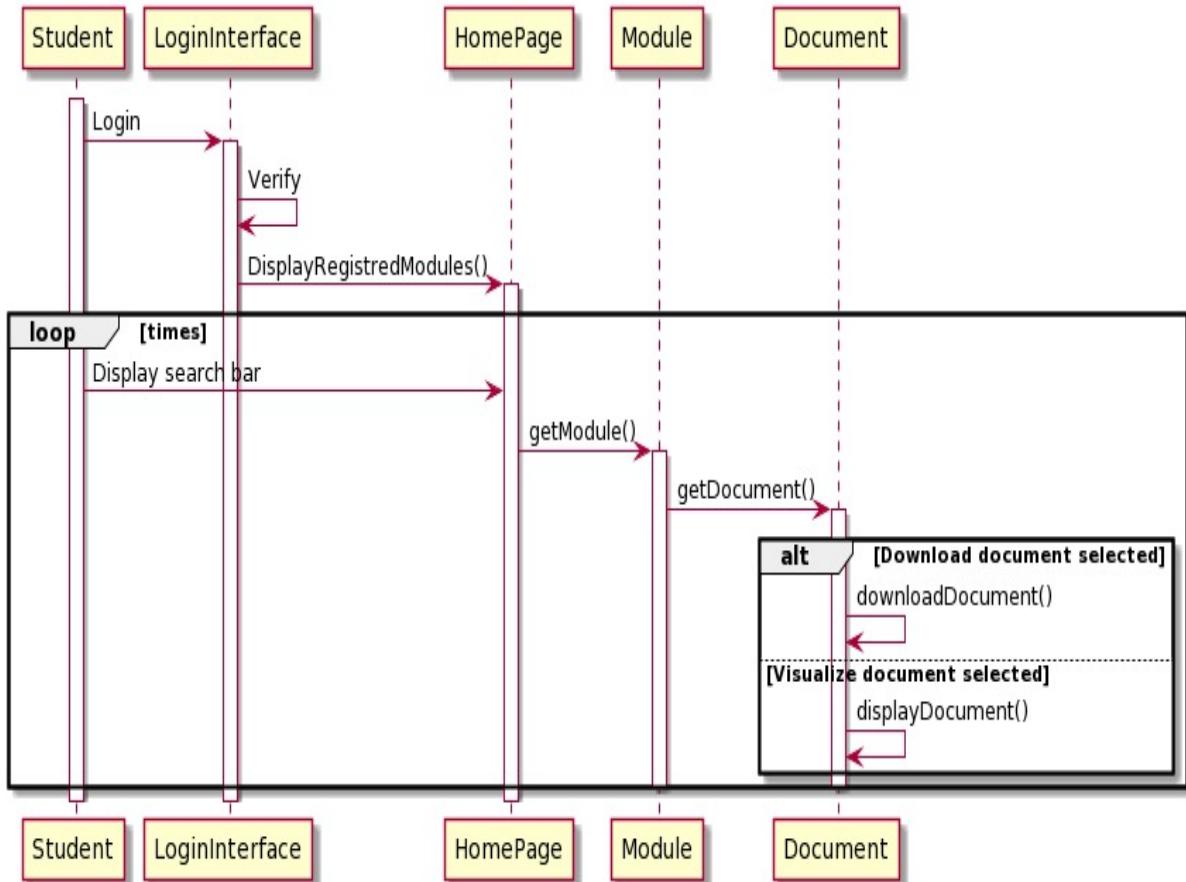


FIGURE 3 – Diagramme de séquences : Accèder à un module

2.2.2 Ajouter à un module :

- Description : Ce deuxième cas d'utilisation permet de chercher d'ajouter un module à l'application.
- Acteur : le ou les administrateurs de l'application
- Pré-conditions : L'utilisateur doit posséder un compte avec les droit d'administrateur et doit s'authentifier

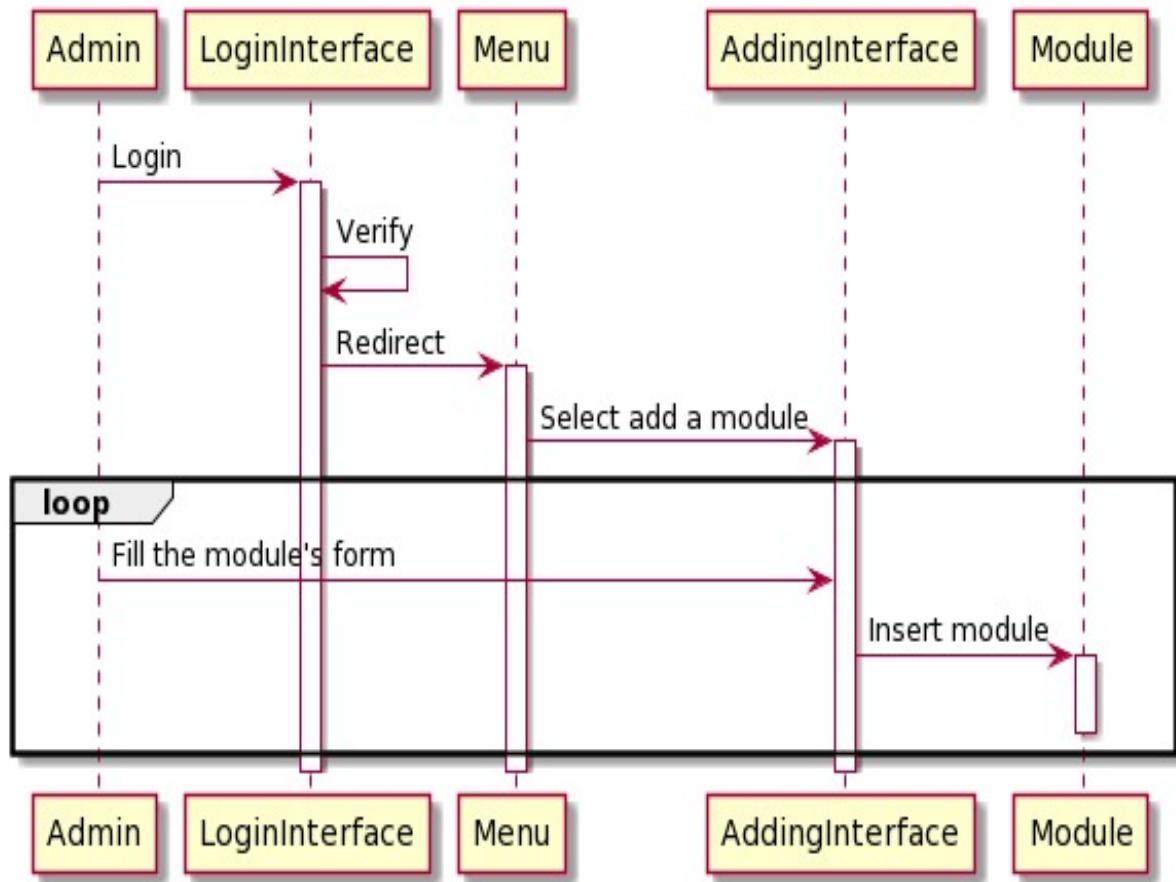


FIGURE 4 – Diagramme de séquences : Ajouter un module

2.3 Diagramme d'activité :

Les diagrammes d'activités sont considérés comme des diagrammes comportementaux, car ils décrivent ce qui doit arriver dans le système modélisé. Nous allons présenter dans cette partie les deux diagrammes d'activités dont notamment : "Espace administrateur" et "Espace étudiant".



FIGURE 5 – Diagramme d'activité : Espace administrateur

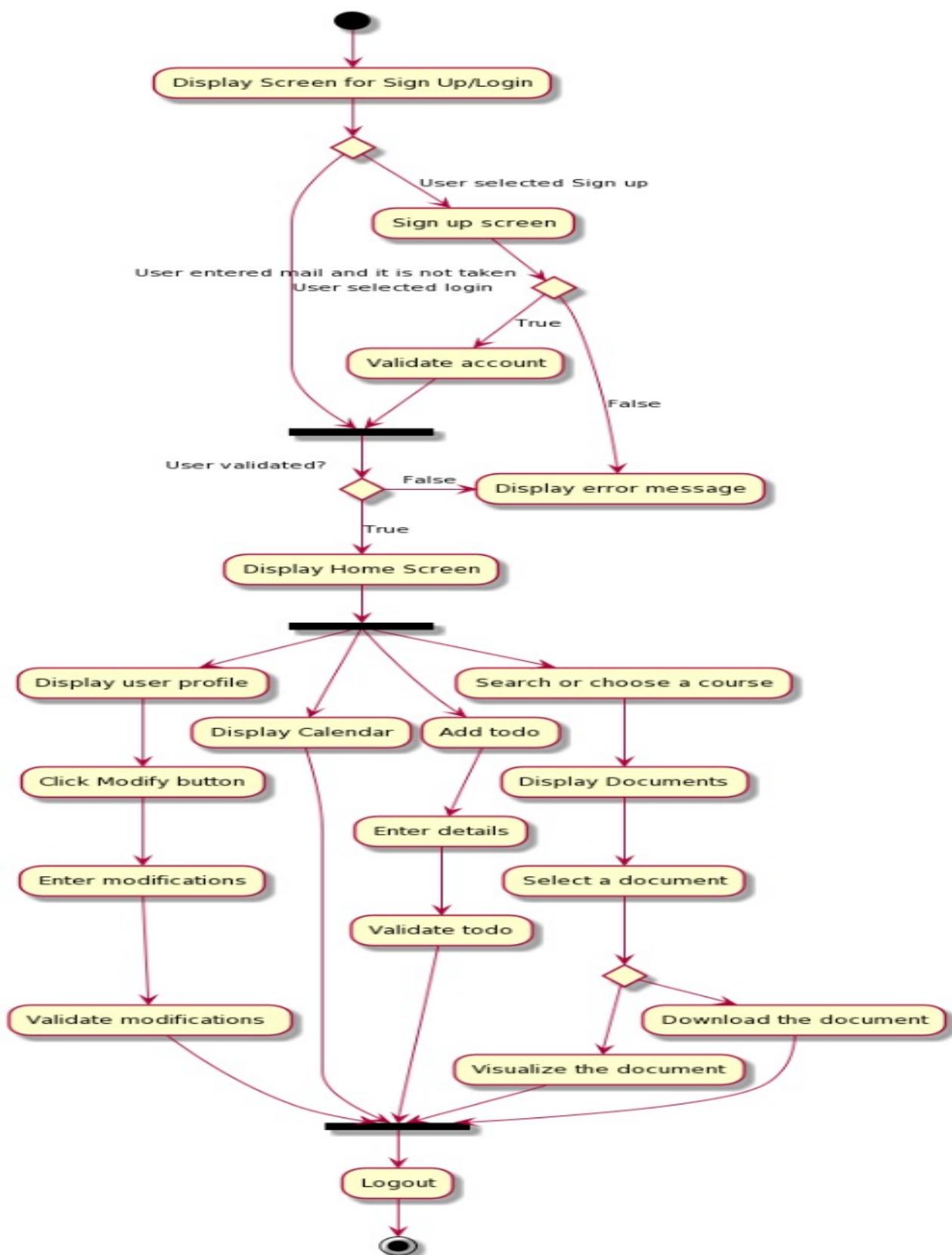


FIGURE 6 – Diagramme d'activité : Espace étudiant

2.4 Diagramme de classes :

Ce diagramme consiste à consolider et valider toute l'analyse des cas d'utilisation. Il s'agit d'un regroupement de l'ensemble des classes en un seul diagramme (diagramme de classe récapitulatif).

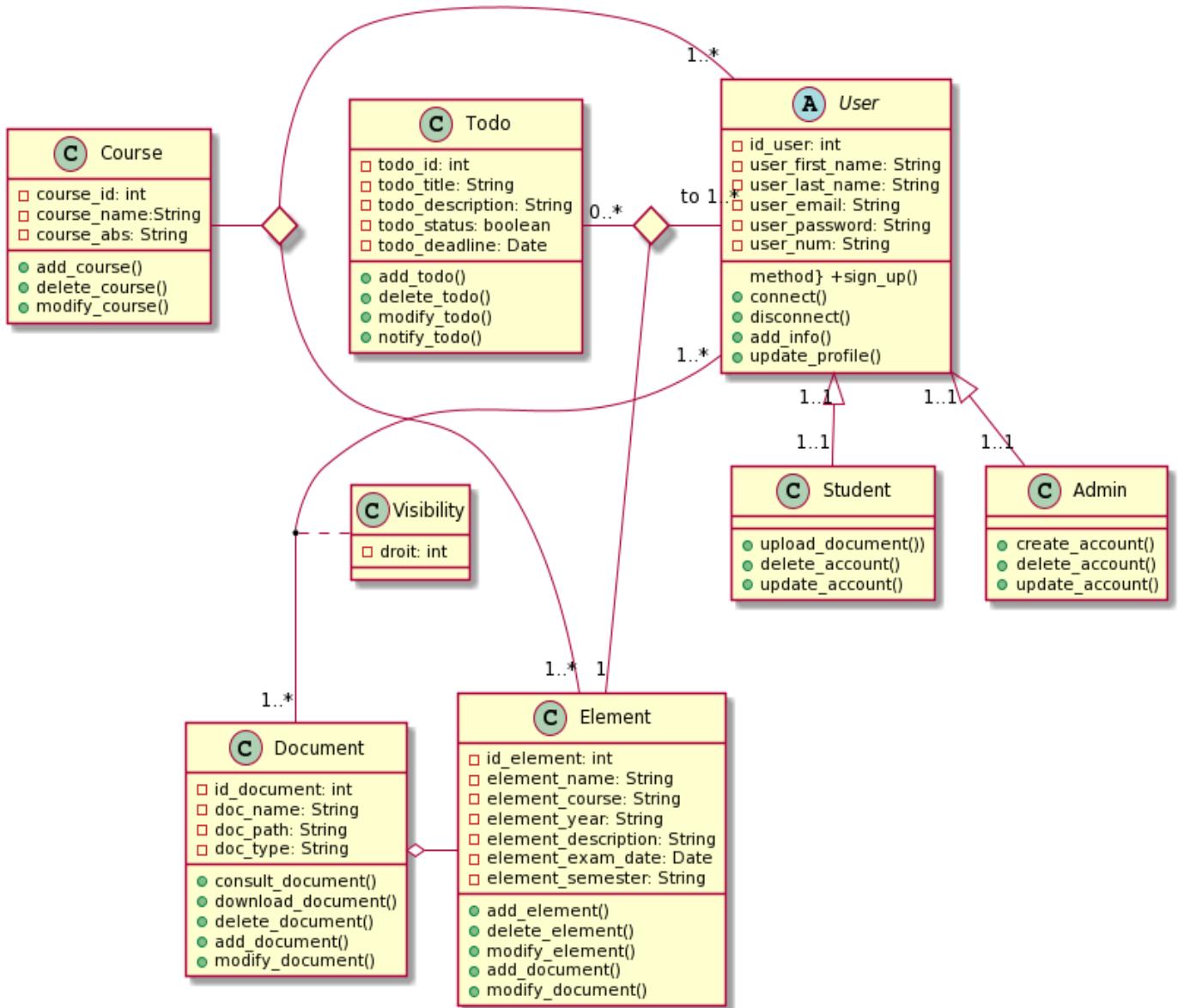
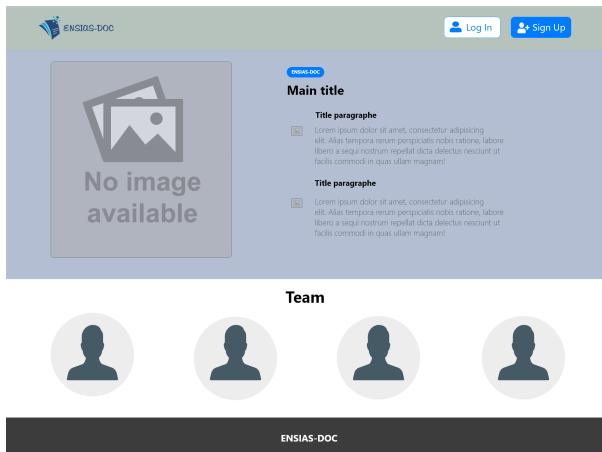


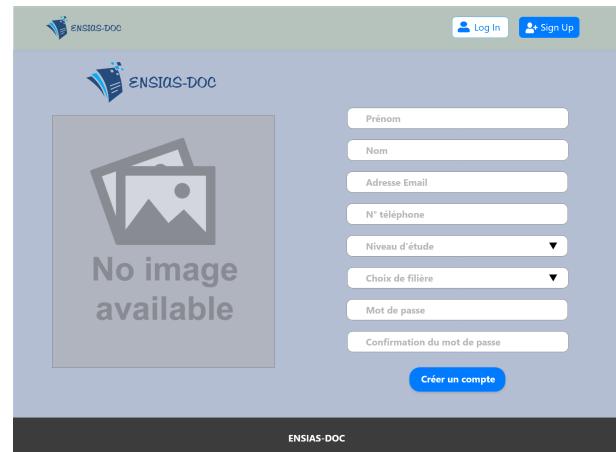
FIGURE 7 – Diagramme de classes

2.5 Maquettes du projet :

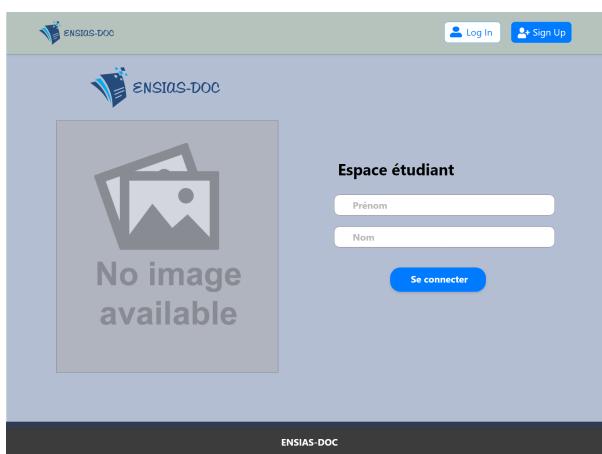
Le maquettage est une méthode de conception d'interface qui permet de proposer des interfaces conformes à nos attentes vis à vis de l'application. Dans cette partie nous allons présenter les maquettes de notre projet réalisées à l'aide de Adobe XD.



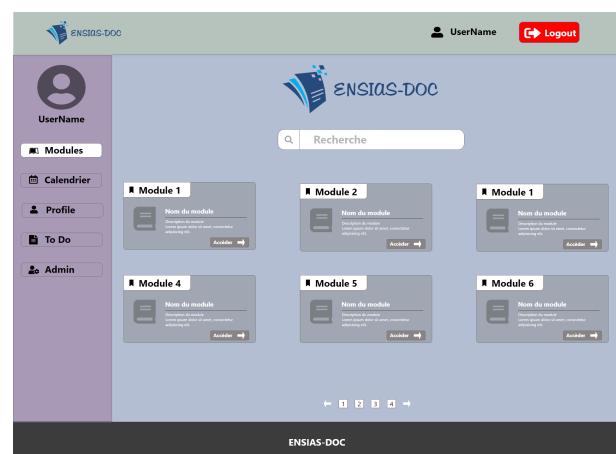
Maquette page d'accueil



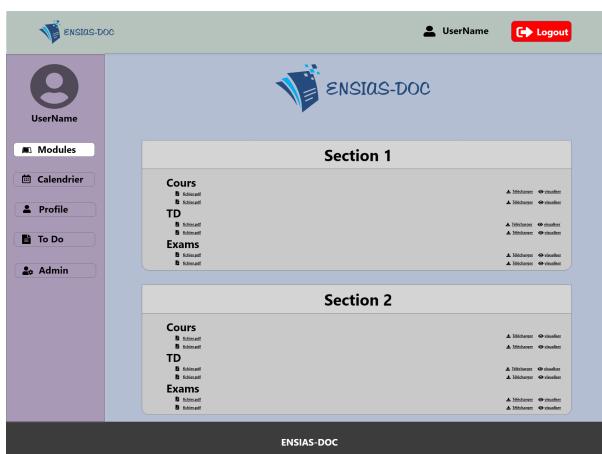
Maquette Sign Up



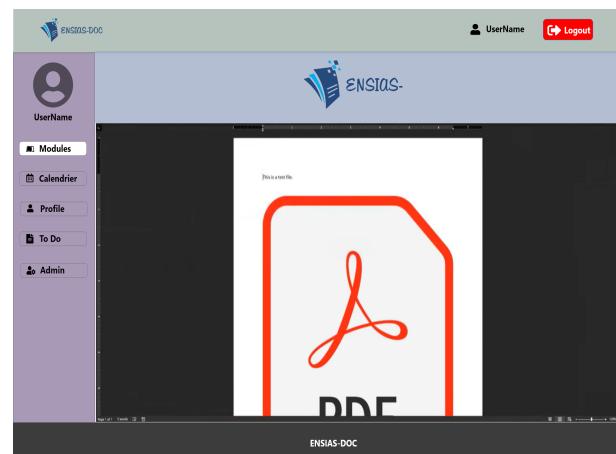
Maquette d'authentification



Maquette espace étudiant



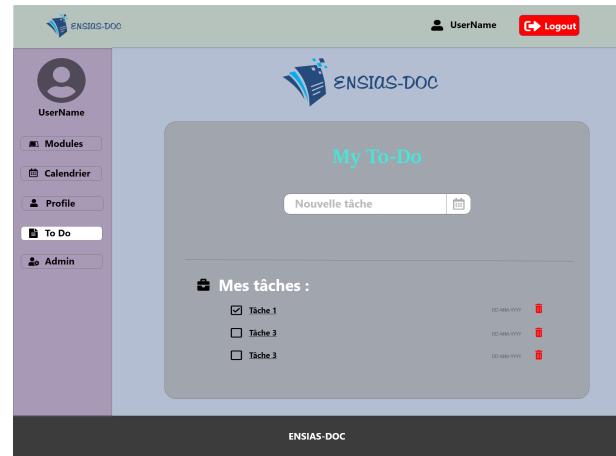
Maquette des modules



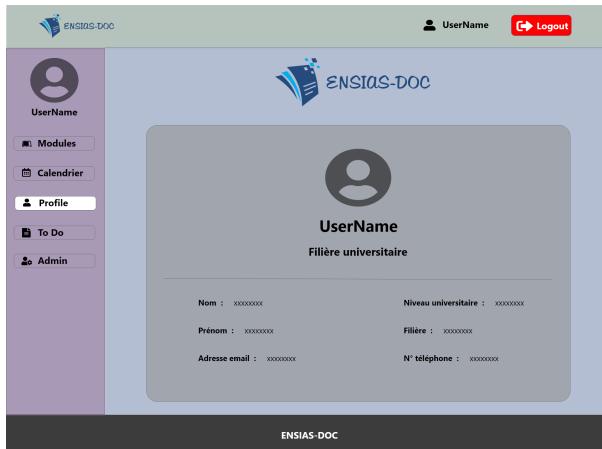
Maquette des documents



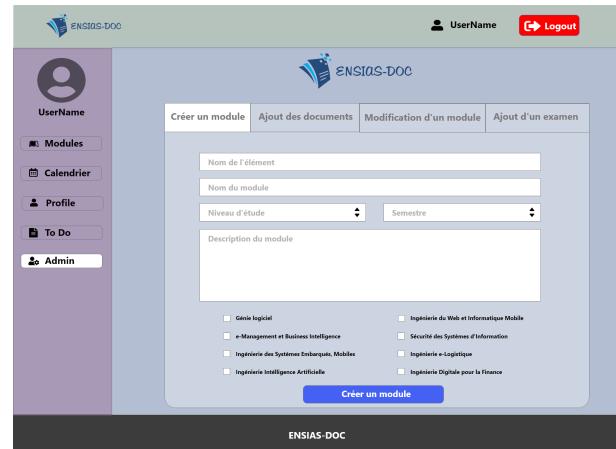
Maquette Calendrier



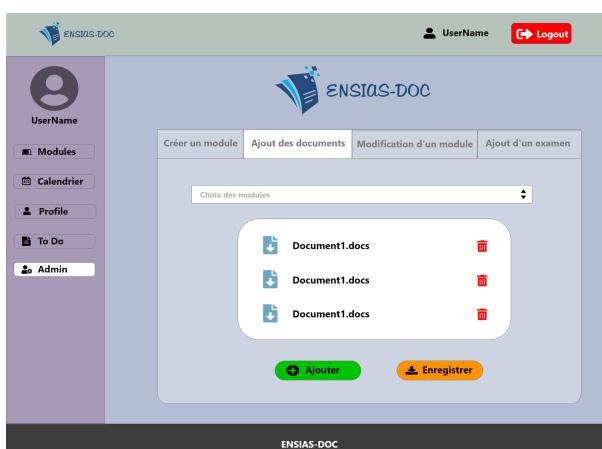
Maquette Todo



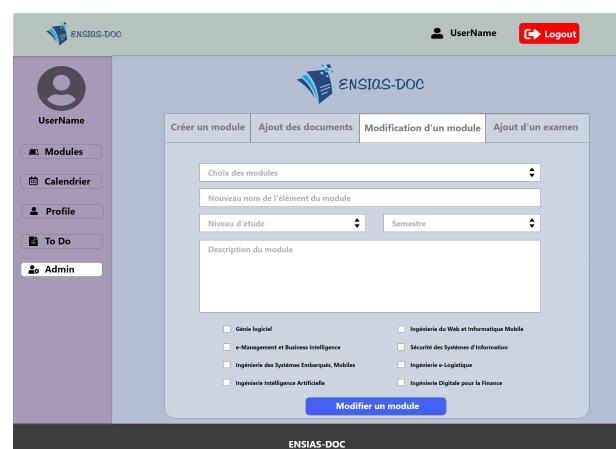
Maquette Profil utilisateur



Maquette Ajouter Module



Maquette Ajouter Document



Maquette Modifier Module

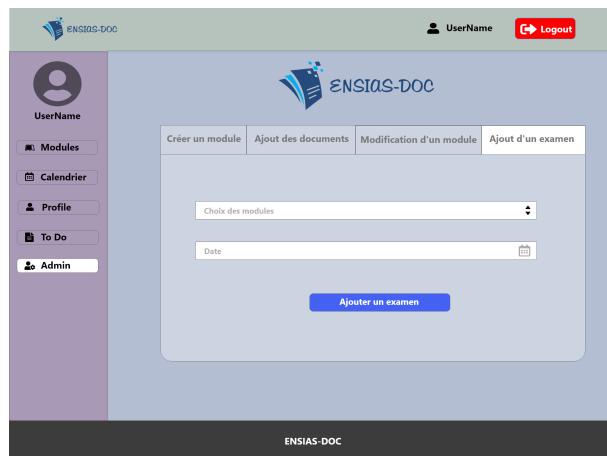


FIGURE 8 – Maquette Ajouter Examen

2.6 Conclusion :

Cette partie a été consacrée à l’élaboration des différents diagrammes selon le langage de modélisation unifié UML : cas d’utilisations, séquences et diagrammes de classes. Ces diagrammes sont nécessaires pour avoir un champ confortable lors de l’implémentation du code de l’application.

Chapitre 3

Réalisation

Dans ce chapitre, nous entamerons la phase de réalisation qui est l'étape où nous traduisons la conception et les règles par un langage de programmation afin d'aboutir à une automatisation des fonctionnalités de l'application. Nous allons préciser l'architecture du projet, les outils utilisés et finalement les captures d'écran du résultat final .

3.1 Architecture du projet : Pattern MVC

Le pattern d'architecture logicielle MVC (Modèle-Vue-Contrôleur) est un modèle destiné à répondre aux besoins des applications interactives en séparant les problématiques liées aux différents composants au sein de leur architecture respective. Ce paradigme regroupe les fonctions nécessaires en trois catégories :

- Un modèle : modèle de données.
- Une vue : interface utilisateur.
- Un contrôleur : logique de contrôle.

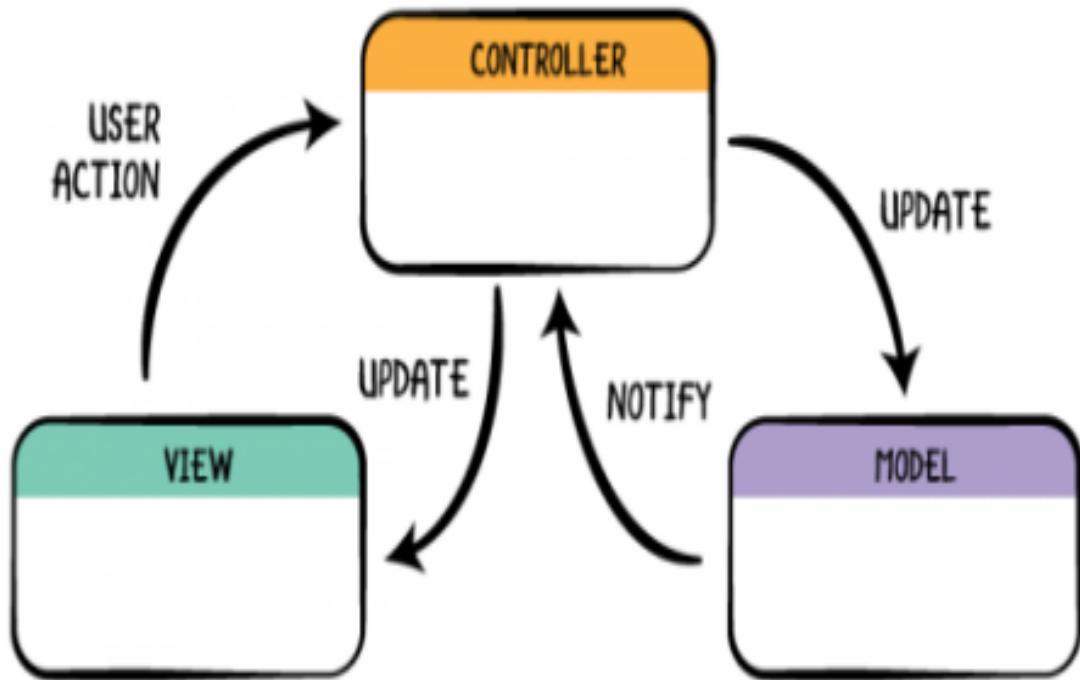


FIGURE 9 – Design Pattern MVC.

3.2 Outils de développement :

3.2.1 Langage de programmation :



Le terme « Java EE » signifie Java Enterprise Edition. En effet, la plate-forme Java EE est construite sur le langage Java et la plate-forme Java SE, et elle y ajoute un grand nombre de bibliothèques remplissant tout un tas de fonctionnalités que la plate-forme standard ne remplit pas d'origine. L'objectif majeur de Java EE est de faciliter le développement d'applications web robustes et distribuées. [1]

3.2.2 Outil de gestion de projet :



Apache Maven (couramment appelé Maven) est un outil de gestion et d'automatisation de production des projets logiciels Java en général et Java EE en particulier. Il est utilisé pour automatiser l'intégration continue lors d'un développement de logiciel. [2]

3.2.3 Environnement de développement :



Eclipse IDE est un environnement de développement intégré libre (le terme Eclipse désigne également le projet correspondant, lancé par IBM) extensible, universel et polyvalent, permettant potentiellement de créer des projets de développement mettant en œuvre n'importe quel langage de programmation.

3.2.4 Gestion de version :



Git est un système de contrôle de version open-source spécifique créé par Linus Torvalds en 2005. Concrètement, Git est un système de contrôle de version distribué, ce qui signifie que l'ensemble de la base du code et de l'historique est disponible sur l'ordinateur de chaque développeur, ce qui permet des branchements et une fusion faciles.

3.2.5 Système de gestion de base de données :



MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde³, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, Informix et Microsoft SQL Server. [4]

3.2.6 Front-end :



L'HyperText Markup Language, généralement abrégé HTML, est le langage de balisage conçu pour représenter les pages web. C'est un langage permettant d'écrire de l'hypertexte, d'où son nom.



Les feuilles de style en cascade, généralement appelées CSS de l'anglais Cascading Style Sheets, forment un langage informatique qui décrit la présentation des documents HTML et XML. Les standards dénissant CSS sont publiés par le World Wide Web Consortium.



JavaScript (qui est souvent abrégé en JS) est un langage de script léger, orienté objet, principalement connu comme le langage de script des pages web.

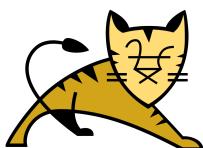


Bootstrap est une collection d'outils utile à la création du design (graphisme, animation et interactions avec la page dans le navigateur ... etc. ...) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. [7]



La JavaServer Pages Standard Tag Library est un composant de la plate-forme JEE de développement. Elle étend la spécification JSP en ajoutant une bibliothèque de balises pour les tâches courantes, comme le travail sur des fichiers XML, l'exécution conditionnelle, les boucles et l'internationalisation.

3.2.7 Serveur de l'application :



Tomcat est un conteneur web libre de servlets et JSP. Issu du projet Jakarta, c'est un des nombreux projets de l'Apache Software Foundation. [3]

3.2.8 Outils de test :

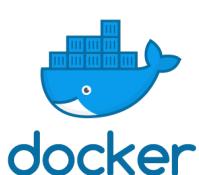


JUnit est un framework mature pour permettre l'écriture et l'exécution de tests automatisés unitaires. Les classes de tests JUnit 5 sont similaires à celles de JUnit 4. Cependant, JUnit 5 est une réécriture complète de l'API contenue dans des packages différents de ceux de JUnit 4. [5]



Dans un test d'intégration avec Arquillian, le code de test n'est pas dépendant du serveur ! Le code crée dynamiquement une archive et la déploie dans un conteneur cible en fonction de l'adaptateur de conteneur disponible dans le classpath de test.

3.2.9 Outil de virtualisation :



Docker permet d'embarquer une application dans un container virtuel qui pourra s'exécuter sur n'importe quelle machine. Disponibles pour les applications Linux et Windows, les "logiciels conteneurisés" fonctionneront toujours de la même façon, quel que soit l'environnement. Les conteneurs isolent le logiciel de son environnement. [6]

3.2.10 Plateforme de déploiement :



Amazon Web Services (AWS) est la plateforme cloud la plus complète et la plus largement adoptée au monde. Elle propose plus de 200 services complets issus de centres de données du monde entier. Des millions de clients y compris de très grandes entreprises et des agences fédérales de premier plan utilisent AWS pour réduire leurs coûts, gagner en agilité et innover plus rapidement. [6]

3.3 Présentation de l'application :

Cette partie dénombre la présentation des Scénarios applicatifs de l'application. Nous allons présenter dans ce qui suit, les imprimés-écran des principales interfaces réalisées dans notre site web.

3.3.1 Page d'accueil

C'est la page d'accueil qui s'affiche dès l'accès à notre application web. Elle contient deux options : se connecter ou s'inscrire.

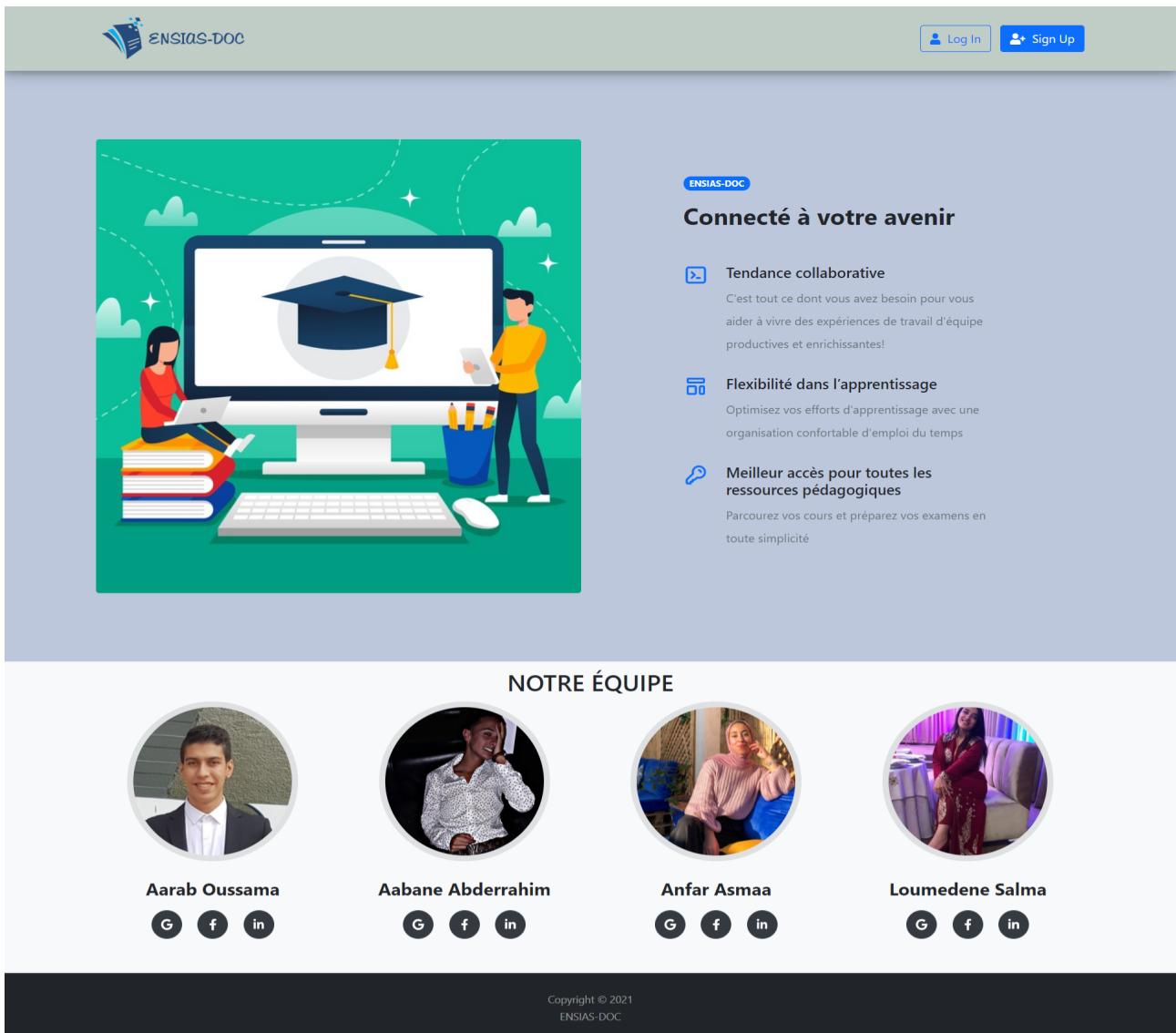


FIGURE 10 – Page d'accueil.

3.3.2 Page d'inscription

Comme dans toutes les plateformes en ligne, le visiteur ne peut profiter des activités qu'après la phase d'inscription, notre application web met à la disposition de ses visiteurs un formulaire d'inscription.

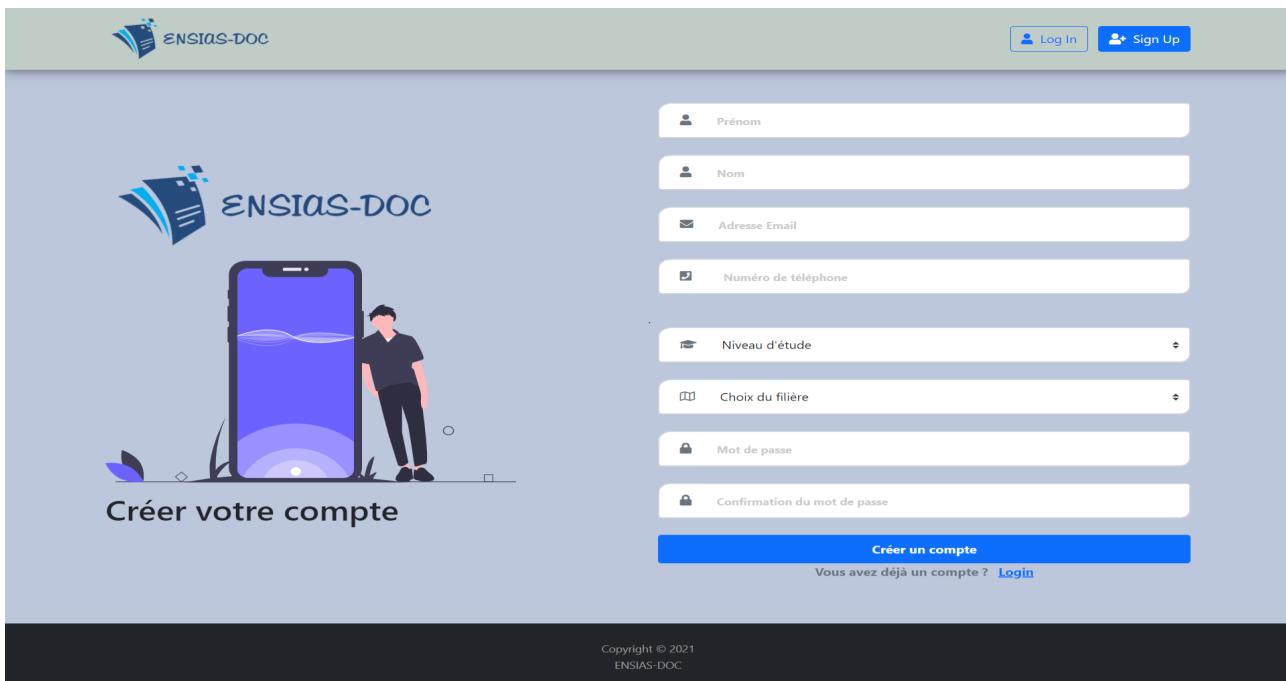


FIGURE 11 – Page d’inscription.

3.3.3 Page d’authentification

Après la phase d’inscription présentée dans la figure précédante l’utilisateur doit s’authentifier pour avoir accès à son compte.

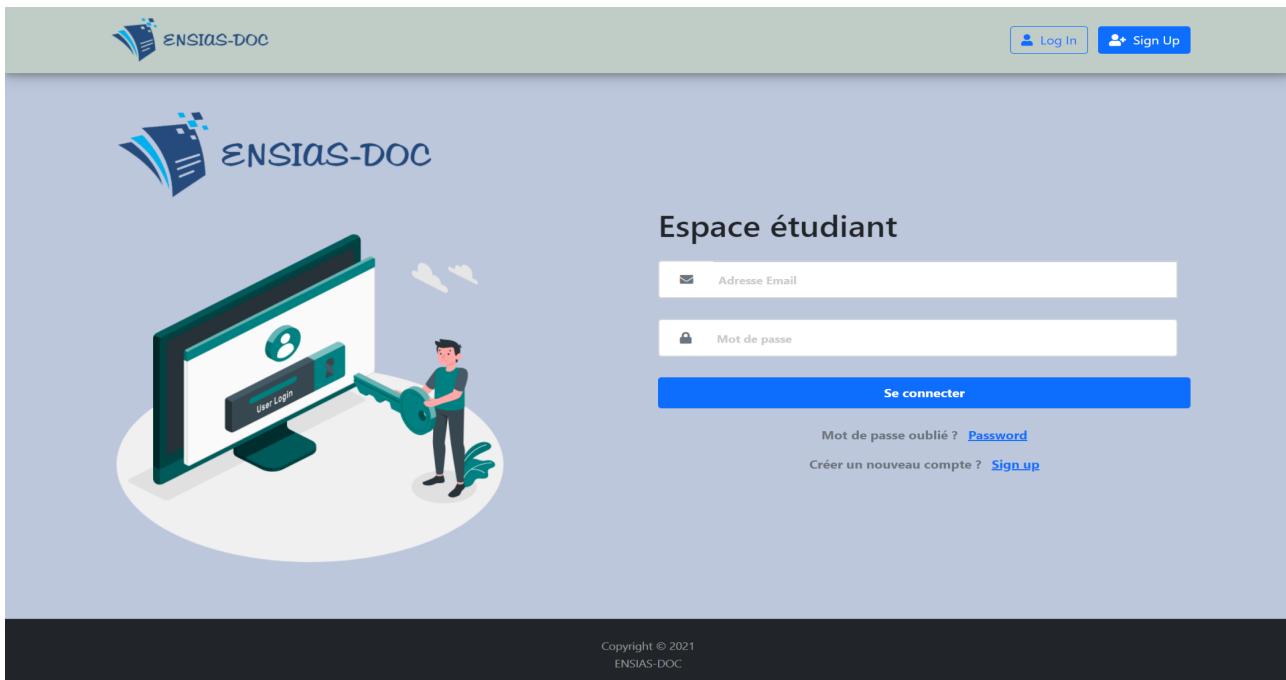


FIGURE 12 – Page d’authentification.

3.3.4 Profil

Le profil contient l'ensemble des informations concernant l'utilisateur inscrit sur l'application y compris : le nom, le prénom, le niveau d'étude, l'adresse mail, le numero de téléphone et la filière.

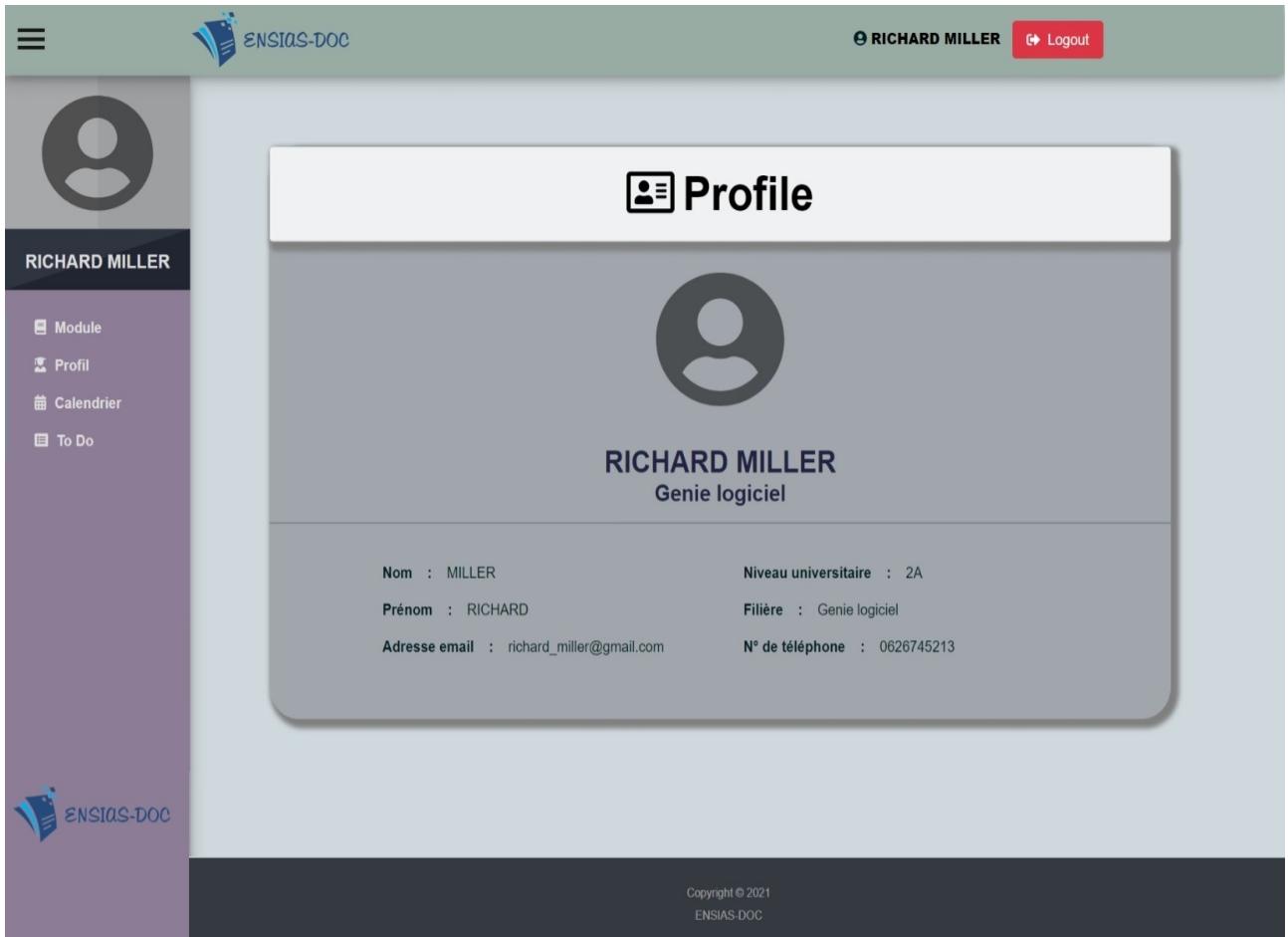


FIGURE 13 – Profil.

3.3.5 Page de modules

Cette page affiche l'ensemble des modules que contient l'application et dont l'utilisateur est inscrit, avec une barre de recherche facilitant la recherche d'un module quelconque.



FIGURE 14 – Page de modules.

3.3.6 Page de documents

Après avoir choisi le module concerné, la liste des document de ce dernier s'affiche. L'utilisateur pourra donc visualiser les documents ou même les télécharger.

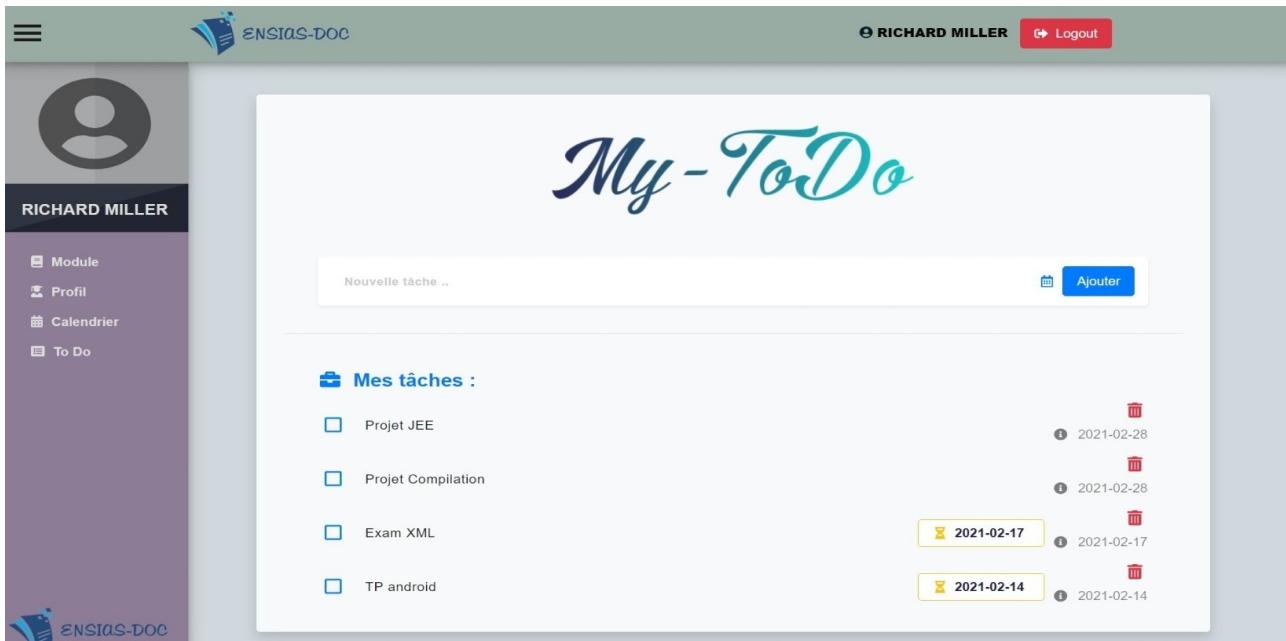


The screenshot shows the ENSIAS-DOC application interface. At the top, there is a header bar with the logo 'ENSIAS-DOC', the user name 'RICHARD MILLER', and a 'Logout' button. On the left, a sidebar displays a user profile icon and a menu with options: Module, Profil, Calendrier, and To Do. The main content area is titled 'Génie Logiciel' and contains a sub-section titled 'Documents'. This section is further divided into 'Cours' and lists five PDF files: '01 Génie Logiciel.pdf', '02 - Processus de Dev Logiciel.pdf', '03 - Exigences et Specification.pdf', '04 - Conception.pdf', and '05- Test & Verification.pdf'. Each file has two buttons: 'Télécharger' (Download) and 'Visualiser' (View).

FIGURE 15 – Page de documents.

3.3.7 Todo liste

Chaque utilisateur a sa propre todo liste lui permettant d'ajouter une tâche et son délai, la marquer comme terminée, ou la supprimer. Une alerte s'affiche près des tâches lorsque leurs délais s'approchent d'un jour afin de mieux rappeler l'utilisateur à les achever.



The screenshot shows the ENSIAS-DOC application interface with the 'My-Todo' section. The top navigation bar and sidebar are identical to Figure 15. The main content area is titled 'My-Todo' and features a input field 'Nouvelle tâche ..' with a calendar icon and a 'Ajouter' button. Below this, a section titled 'Mes tâches :' lists five tasks: 'Projet JEE', 'Projet Compilation', 'Exam XML', 'TP android', and 'TP android'. Each task has a checkbox, a due date (e.g., '2021-02-28' or '2021-02-14'), and a trash bin icon. A yellow box highlights the task 'TP android' with the due date '2021-02-14'.

FIGURE 16 – Todo liste.

3.3.8 Calendrier

Le calendrier affiche les dates des examens des éléments de modules ainsi que toutes les tâches de la todo listes ajoutées par l'utilisateur.

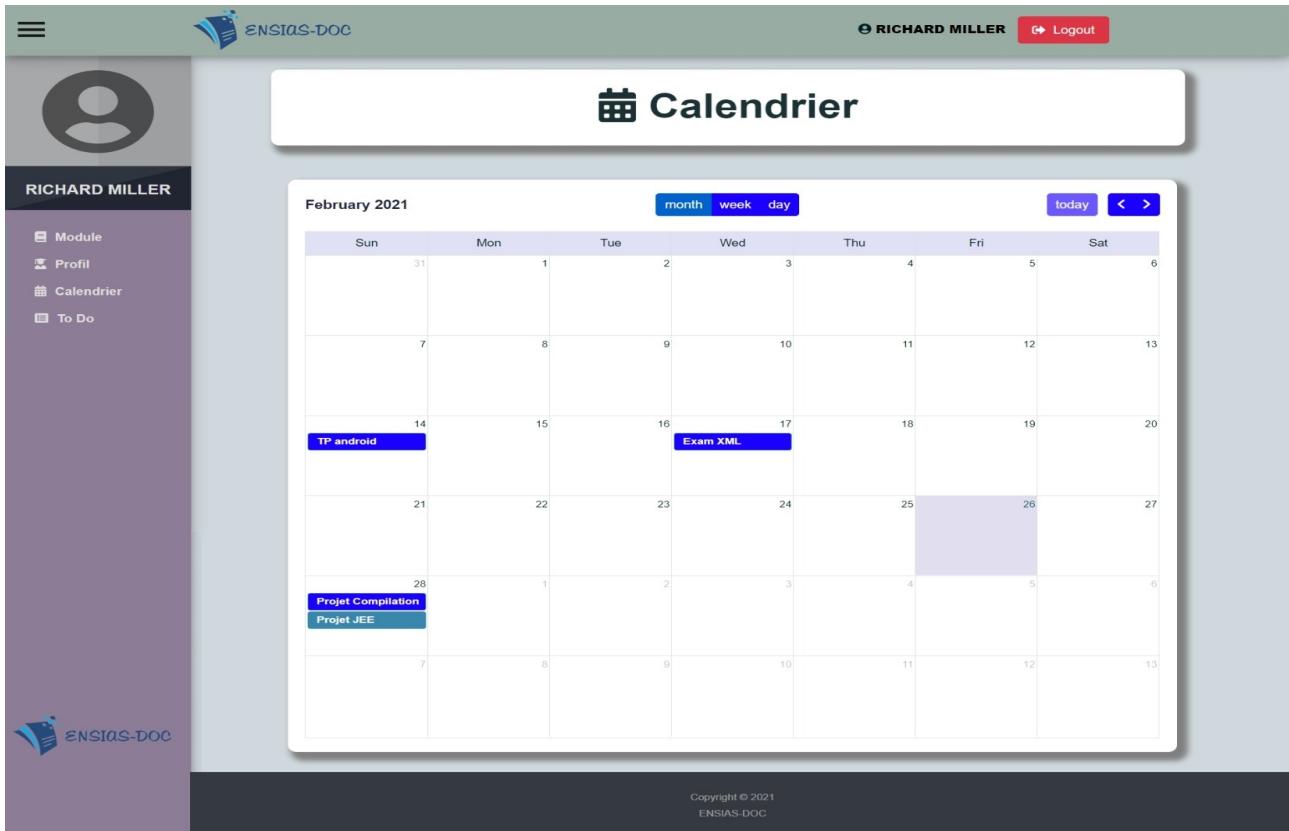


FIGURE 17 – Calendrier.

3.3.9 Espace administrateur

Cet espace est consacré seulement à l'administrateur de l'application où il a le droit d'ajouter un module, modifier les informations d'un module, ajouter un document à un module déjà existant ou ajouter la date d'un examen qui s'affichera par la suite aux utilisateurs.

3 RÉALISATION

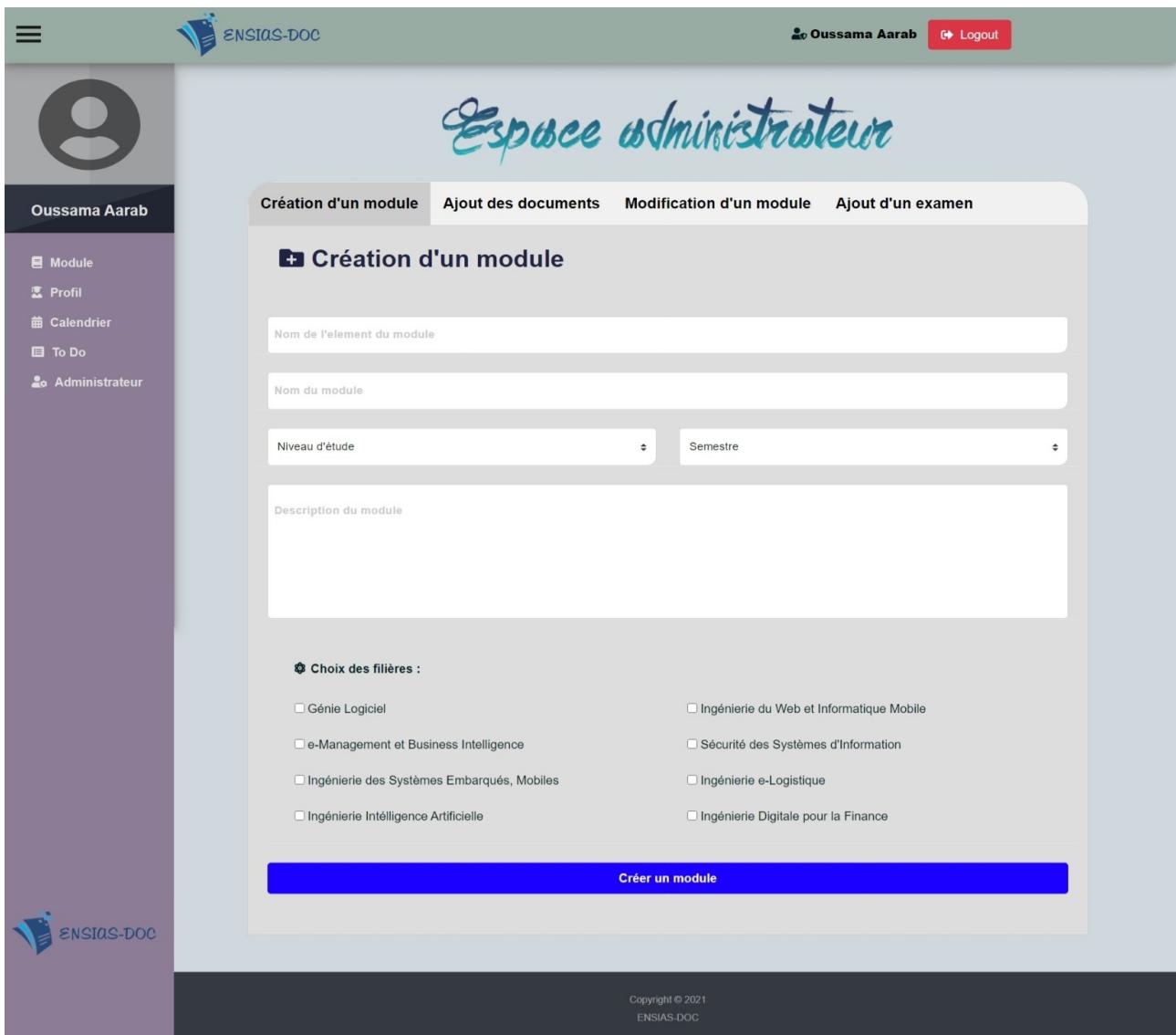


FIGURE 18 – Espace admin.

3.4 Conclusion :

Cette partie concerne la réalisation du projet. En effet, nous avons exposé tous nos outils de travail qui nous ont servi pour développer l'application. Nous avons ensuite exhibé nos copies d'écran pour montrer le résultat réel de notre application.

Conclusion générale :

Ce projet consistait en la réalisation d'une application WEB au service des étudiants de l'ENSIAS qui leur assure un espace parfait de révision. Pour ce, il fallait appliquer directement les connaissances acquises en cours : le langage UML[1] pour modéliser le système de notre application ainsi que celles acquises en cours de l'ingénierie du web pour la phase de l'implémentation. Afin de réaliser un travail aussi important, nous avons fait beaucoup de recherches pour atteindre nos objectifs. Or, notre réalisation est loin d'être parfaite voir terminée. Nous pensons certes être arrivés à un bon compromis entre réalisme et créativité. Enfin ce projet a été une bonne occasion d'apprendre de nouveaux outils que ce soit en JEE ou en JSTL. Le projet a été aussi une occasion pour améliorer notre aptitude pour travailler en groupe.

Références

- [1] <https://netbeans.org/kb/trails/java-ee.html>
- [2] <https://mkyong.com/maven/how-to-create-a-web-application-project-with-maven/>
- [3] <https://tomcat.apache.org/tomcat-7.0-doc/deployer-howto.html>
- [4] <https://netbeans.org/kb/docs/web/mysql-webapp.html>
- [5] <https://www.vogella.com/tutorials/JUnit/article.html>
- [6] <https://stackify.com/guide-docker-java/>
- [7] <https://getbootstrapstrap.com/>, 2014. [Online; accessed 27-December-2018]