# Chapitre 2 : Travaux réalisés au cours du projet de fin de cycle

Dans ce deuxième chapitre, nous allons décrire de manière détaillée l’ensemble des travaux réalisés durant le projet de conception et de développement de la plateforme de gestion des mémoires à l’Institut Supérieur d’Informatique (ISI). Le déroulement du projet s’est articulé autour de cinq grandes phases : l’analyse des besoins, la conception de la solution, la maquettisation des interfaces, l’implémentation technique, et enfin le déploiement de la plateforme.

## Travail 1 : Analyse des besoins

### 1.1. Besoins fonctionnels détaillés

Les besoins fonctionnels décrivent l’ensemble des fonctionnalités que la plateforme doit impérativement offrir aux utilisateurs pour répondre aux objectifs pédagogiques et administratifs du processus de gestion des mémoires. Ces besoins traduisent les actions concrètes que chaque type d’utilisateur (étudiant, encadreur, administrateur) doit pouvoir réaliser sur le système.

* **Dépôt des sujets par les enseignants** : chaque encadreur a la possibilité de proposer des sujets de mémoire en ligne. Ces propositions sont enregistrées et classées automatiquement, facilitant leur consultation ultérieure par les étudiants.
* **Consultation et choix des sujets par les étudiants** : les étudiants ont accès à un espace sécurisé où ils peuvent parcourir l’ensemble des sujets disponibles, trier les propositions selon différents critères (domaine, encadreur, spécialité), puis sélectionner le sujet qui les intéresse.
* **Choix libre de l’encadreur** : contrairement à certaines pratiques figées, l’étudiant conserve la liberté de choisir son encadreur indépendamment de l’auteur du sujet proposé. Ce choix permet une meilleure adéquation entre les aspirations de l’étudiant et les compétences de l’encadreur.
* **Validation des choix par les encadreurs et l’administration** : une fois le choix effectué, l’encadreur et l’administration interviennent pour valider respectivement la faisabilité pédagogique et la conformité administrative de la proposition.
* **Gestion des paiements de soutenance** : la plateforme intègre un module de suivi des paiements des frais de soutenance. Cette gestion numérique garantit la transparence et l’automatisation du processus de délivrance des fiches de suivi.
* **Suivi des séances d’encadrement** : chaque encadrement fait l’objet d’un enregistrement précis sur la plateforme. On y note la date de la séance, les travaux réalisés, les observations de l’encadreur ainsi que les visas respectifs.
* **Dépôt du mémoire final** : avant la soutenance, l’étudiant dépose son document finalisé directement sur la plateforme, garantissant ainsi sa traçabilité et sa conservation numérique.
* **Composition automatique des jurys** : la plateforme permet de générer automatiquement des jurys de soutenance tout en respectant les contraintes pédagogiques et institutionnelles. En plus de vérifier que l’encadreur de l’étudiant ne fasse pas partie du jury, le système tient également compte du domaine du sujet de mémoire pour proposer des jurys pertinents.
* **Archivage numérique des mémoires** : les mémoires soutenus sont archivés dans une bibliothèque numérique consultable par les étudiants, les enseignants et l’administration selon des droits d’accès prédéfinis.
* **Gestion des utilisateurs et rôles** : la plateforme gère différents profils (administrateur, encadreur, étudiant), chacun disposant d’un espace de travail personnalisé et de droits d’accès spécifiques.

### 1.2. Besoins non fonctionnels détaillés

Les besoins non fonctionnels traduisent les exigences qualitatives du système. Ils n’expriment pas ce que le système doit faire, mais **comment** il doit le faire. Ils concernent des aspects transversaux comme la performance, la sécurité, la disponibilité ou encore l’ergonomie. Ces besoins garantissent la fiabilité, la robustesse et la facilité d’utilisation de la solution à long terme.

* Sécurité et confidentialité des données
* Accessibilité et ergonomie
* Disponibilité et performance
* Traçabilité des actions
* Sauvegarde et récupération des données

## Travail 2 : Conception de la solution

### 2.1. Choix du langage de modélisation UML

L’UML (Unified Modeling Language) a été retenu pour sa capacité à représenter les interactions, entités et relations du système de manière claire et standardisée.

Pourquoi UML ?

* **Langage universel et standardisé** : utilisé dans la majorité des projets logiciels professionnels, UML garantit une compréhension partagée entre les membres de l’équipe technique et les parties prenantes non techniques.
* **Adapté à l’approche objet** : le projet étant basé sur une architecture orientée objet, UML permet de représenter avec précision les relations entre classes, objets et responsabilités.
* **Richesse des représentations** : UML propose plusieurs types de diagrammes (cas d’utilisation, classes, séquences, activités, etc.), offrant ainsi une couverture complète du cycle de vie d’un système.
* **Outil de documentation** : les diagrammes UML servent également de documentation technique pour les futurs développeurs et facilitateurs de maintenance.

### 2.2. Diagramme de cas d’utilisation détaillé

* Authentification des utilisateurs
* Consultation et choix de sujet de mémoire par les étudiants
* Validation des sujets par les encadreurs et l’administration
* Paiement des frais de soutenance et génération des fiches de suivi
* Planification et gestion des séances d’encadrement
* Dépôt final du mémoire
* Génération automatique des jurys de soutenance
* Consultation de la bibliothèque numérique
* Gestion des rôles et des profils utilisateurs
* Suivi des actions et notifications

(image à insérer)

### 2.3. Diagramme de classes approfondi

Entités principales : Utilisateur, Étudiant, Encadreur, Sujet, Mémoire, Séance, Jury, MembreJury, Paiement, FicheSuivi.

(image à insérer)

## Travail 3 : Maquettisation de la solution

La phase de maquettisation a permis de simuler les interfaces avant leur développement. Outil utilisé : **Figma**. Interfaces créées : tableau de bord, espace étudiant, encadreur, bibliothèque numérique.

(image à insérer)

# Travail 4 : Implémentation

## 4.1. Environnement technique

### 4.1.1. Les outils utilisés durant le projet de fin de cycle

#### Matériel :

* Ordinateur portable personnel
* Connexion Internet stable

Logiciels :

* **Visual Studio Code** : puissant éditeur de code open-source, polyvalent pour le développement frontend (React) et backend (Node.js).
* **PostgreSQL** : SGBD relationnel robuste, fiable et performant, utilisé pour stocker toutes les données de la plateforme.
* **Postman** : outil de test d’API REST, utile pour valider les endpoints backend de manière rapide et visuelle.

### ****4.1.2. Technologies utilisées****

La plateforme a été construite à l’aide d’un ensemble cohérent de technologies modernes, facilitant à la fois le développement, la maintenabilité et la sécurité du système :

* **React.js + TypeScript** : pour le développement du frontend avec une structure typée, robuste, et facilitant la lisibilité et la maintenance du code.
* **Tailwind CSS + shadcn/ui** : pour la mise en place rapide d’interfaces utilisateurs modernes, responsives et esthétiques, avec une base de composants réutilisables.
* **Node.js + Express.js** : pour la construction d’une API backend REST performante et évolutive.
* **TypeScript** : utilisé côté backend également pour bénéficier de la vérification statique des types, réduisant les erreurs à l'exécution.
* **Prisma ORM** : outil de mapping objet-relationnel moderne permettant une interaction simplifiée, sécurisée et typée avec la base PostgreSQL.
* **JWT (jsonwebtoken) + bcrypt** : utilisés pour la gestion sécurisée de l’authentification et du chiffrement des mots de passe.
* **CORS (Cross-Origin Resource Sharing)** : configuration serveur permettant de sécuriser les échanges entre le frontend (hébergé séparément) et l’API backend.
* **Git + GitHub** : pour le versionnement, la collaboration et le déploiement continu.
* **Vercel** : pour le déploiement du frontend avec un CDN, SSL intégré, et gestion CI/CD.
* **Render** : pour l’hébergement du backend Node.js et la gestion automatisée de la base PostgreSQL.

## 4.2. Réalisation des travaux applicatifs

### Création de la base de données

La base de données PostgreSQL a été créée intégralement à partir d’un modèle conceptuel conçu durant la phase de conception. Toutes les entités (utilisateur, mémoire, encadreur, sujet, paiement, séance, jury) et leurs relations ont été implémentées avec les contraintes d’intégrité nécessaires (clés primaires, étrangères, unicité).

### Création des interfaces

Les interfaces utilisateur ont été conçues intégralement à l’aide de React.js et Tailwind CSS **shadcn/ui**. Elles sont organisées selon les profils utilisateurs :

* Tableau de bord administrateur
* Espace encadreur
* Espace étudiant
* Accès à la bibliothèque numérique  
  Chaque interface a été testée pour assurer sa réactivité, son accessibilité et sa fluidité d’utilisation.

### Création de CRUDs des services choisis

Les modules développés couvrent tous les aspects fonctionnels du système :

* **Gestion des utilisateurs** : authentification, rôles (admin, encadreur, étudiant), sécurisation des accès.
* **Gestion des sujets** : création, modification, suppression, consultation.
* **Gestion des séances d’encadrement** : ajout de séances, mise à jour des remarques et visas, traçabilité des suivis.
* **Gestion des paiements** : enregistrement des paiements de soutenance, génération des fiches de suivi.
* **Dépôt des mémoires** : formulaire de dépôt, gestion des fichiers PDF, archivage.
* **Gestion des jurys** : génération automatique des jurys en respectant les contraintes (pas d’encadreur dans le jury de son étudiant, cohérence pédagogique).
* **Notifications dynamiques** : alertes automatiques à chaque changement de statut ou action attendue.

## Travail 5 : Déploiement

### 5.1. Étapes de déploiement

* Déploiement frontend sur Vercel
* Déploiement backend sur Render
* Sécurisation de la base PostgreSQL

### 5.2. Validation utilisateur

Des tests ont été réalisés avec des encadreurs et administrateurs afin de s’assurer de la conformité fonctionnelle de la plateforme. Les retours ont permis d’ajuster l’expérience utilisateur et de confirmer la mise en production.

Ainsi s’achève la présentation des travaux réalisés durant ce projet de fin de cycle. Le chapitre suivant mettra en lumière les résultats obtenus et les apports de cette solution.