

UNIVERSITÉ DES SCIENCES ET DE LA TECHNOLOGIE HOUARI BOUMEDIENE

DÉPARTEMENT DES SYSTÈMES INFORMATIQUES



Plateforme Gestion des Thèses et Mémoires

Rapport de Projet Multidisciplinaire N° 11

Proposé par : Sonia ZEMMOUR, Ph.D. Student in AI - USTHB

Membres	Groupe	Matricules
Hattab Hamza Riadh	G1	232331394516
Nacer Bey Abderrahmane Zakaria	G1	232331388005
Harridi Mohamed Alaaeddine	G1	232331388615
Zouaoui Abdelmalek	G1	232332138706
Ouattas Amine	G4	232331729105

Année Universitaire 2024/2025

Remerciements

Nous tenons tout d'abord à exprimer notre profonde gratitude envers Allah Tout-Puissant pour nous avoir donné la force et la persévérance nécessaires à la réalisation de ce projet. Nos sincères remerciements s'adressent à notre encadrante pédagogique, Mme Sonia ZEMMOUR, pour ses conseils avisés, sa disponibilité et son soutien constant tout au long de ce travail. Ses orientations ont été précieuses pour mener à bien ce projet. Nous remercions également les membres du jury qui ont accepté d'évaluer notre travail. Nous souhaitons exprimer notre reconnaissance envers le corps enseignant de la Faculté d'Informatique de l'USTHB pour la qualité de la formation dispensée. Enfin, un grand merci à nos familles et amis pour leur soutien indéfectible, leur patience et leurs encouragements durant cette période intense de travail.

Table des matières

Remerciements	iii
Table des matières	iv
Table des figures	vi
Introducion Générale	1
1 Étude de l'existant	2
1.1 Introduction	2
1.2 Analyse des solutions existantes	2
1.2.1 Méthode actuelle à l'USTHB	2
1.2.2 Solutions numériques générales	3
1.3 Conclusion	5
2 Conception de la Plateforme de Gestion des Thèses et Mémoires	7
2.1 Introduction à la Conception	7
2.2 Architecture Globale du Système	7
2.2.1 Vue d'Ensemble de l'Architecture	7
2.2.2 Diagramme d'Architecture	8
2.2.3 Description des Composants Principaux	8
2.2.4 Flux de Données et Interactions	9
2.3 Modèle de Données	9
2.3.1 Diagramme Entité-Association	9
2.3.2 Description des Entités Principales	10
2.3.3 Relations et Cardinalités	11
2.3.4 Normalisation et Intégrité Référentielle	11
2.4 Conception de l'Interface Utilisateur	11
2.4.1 Principes de Conception UI	11
2.4.2 Écrans Principaux et Flux Utilisateur	11
2.4.3 Adaptabilité et Responsive Design	12
2.5 Algorithmes et Pseudo-code	12
2.5.1 Algorithme d'Authentification	12
3 Réalisation et Test	14
3.1 Introduction à l'Implémentation	14
3.2 Technologies et Outils Utilisés	14
3.2.1 Environnement de Développement	14
3.2.2 Technologies Frontend	14
3.2.3 Technologies Backend et Stockage	15

3.2.4	Bibliothèques et Frameworks	15
3.2.5	Implémentation de l'Architecture	15
3.3	Fonctionnalités Implémentées	16
3.3.1	Gestion des Utilisateurs	16
3.3.2	Gestion des Thèses et Mémoires	16
3.3.3	Fonctionnalités Administratives	16
3.4	Sécurité et Performance	16
3.4.1	Mesures de Sécurité Implémentées	16
3.4.2	Optimisations de Performance	16
3.5	Tests et Validation	17
3.5.1	Stratégie de Test	17
3.5.2	Résultats des Tests	17
3.6	Difficultés Rencontrées et Solutions	17
3.6.1	Défis Techniques	17
3.6.2	Solutions Implémentées	17
3.7	Résultats et Interfaces Utilisateur	18
3.7.1	Écran de Connexion	18
3.7.2	Dashboard Principal	18
3.7.3	Interface de Recherche et Consultation	18
3.7.4	Interface pour voir et modifier les thèses d'un utilisateur	19
3.7.5	Interface d'Administration	19
3.7.6	Interface de Consultation des Actualités	20
	Conclusion générale	21
	Bibliographie	23

Table des figures

2.1	Diagramme de l'architecture 3-Tiers de la plateforme PGTM-USTHB	8
2.2	Diagramme du modèle Entite Association	9
3.1	sgin in	18
3.2	sign up	18
3.3	dashboard	18
3.4	Recherche	19
3.5	Mes theses	19
3.6	member list	19
3.7	page admin theses	19
3.8	Page des actualités	20

Introduction Générale

Les progrès spectaculaires réalisés dans le champ des techniques de l'information et de la communication rendent possible une transformation décisive et sans précédent de l'enseignement supérieur et de la recherche scientifique sur tous les continents [1]. Les universités du monde entier tentent de gérer et de faciliter leur processus administratif, pédagogique ou de recherche grâce à des solutions numériques. Or, au cœur de la mission académique se trouvent la production et la diffusion du savoir, et en particulier les thèses de doctorat et les mémoires de Master qui sont le produit de recherches parfois longues et d'un héritage scientifique et technique essentiel, à la fois pour la communauté universitaire elle-même et pour l'ensemble de la société.

Cependant, cette masse de travaux de recherche croissants se prête mal à la gestion et à l'accès. À l'Université des Sciences et de la Technologie Houari Boumediene (USTHB) – comme dans beaucoup d'autres universités avant leur passage au tout numérique –, la gestion des thèses et mémoires est encore largement manuelle ou semi-automatique. La recherche d'un travail précis est souvent un parcours de combattant nécessitant des consultations physiques dans les bibliothèques ou dans les services, apportant un accès aux catalogues parfois lacunaires et souvent mal interrogeables. Une telle situation nuit à la fois à la diffusion et à la visibilité des recherches effectuées en son sein, mais aussi aux étudiants et aux chercheurs qui pourraient avoir intérêt à s'appuyer sur les travaux antérieurs.

Au regard de cette réalité, il est devenu urgent de mettre en place une solution numérique moderne, centralisée et efficace pour la gestion des mémoires et thèses à l'USTHB, ce projet est donc celui du Projet Multidisciplinaire N° 11 qui vise à développer une « Plateforme de Gestion des Thèses et Mémoires » (PGTM-USTHB), dont l'objectif est avant tout de concevoir et réaliser une application logicielle de gestion permettant de centraliser le dépôt, la validation administrative, l'indexation par mots-clés, la recherche multicritère et la consultation sécurisée des thèses et mémoires réalisés dans l'institution [2]. La plateforme devra faire face et répondre aux exigences spécifiques de chacun des différents acteurs (les administrateurs pour la gestion du système et la validation des dépôts, les étudiants (Master et Doctorat) pour le dépôt de leur travail et l'ensemble de la communauté universitaire (enseignants-chercheurs, reste des étudiants) pour la recherche et la consultation.

Le présent rapport est un compte-rendu intégral du projet effectué au sein du notre groupe (groupe 1 et 4 Section B). Le premier chapitre est consacré à l'étude de l'existant. Il analyse la situation de l'USTHB actuelle, une étude de solutions différentes qui existent dans d'autres institutions a été faite dans le but de mieux justifier notre proposition. Le deuxième chapitre traite de la phase de conception, présente l'architecture logicielle souhaitée, la modélisation de la base de données (s'appuyant sur Google Cloud SQL) et la conception détaillée matérialisée à l'aide de diagrammes UML. Le chapitre trois est consacré à l'implémentation de la plateforme PGTM en C# utilisant le WPF, est présentée les technologies utilisées, la réalisation des fonctionnalités principales, l'analyse des résultats fournis ainsi que les problèmes rencontrés. En conclusion générale, un retour d'expérience des travaux effectués permet d'évaluer dans quelle mesure les objectifs fixés ont été atteints, mais aussi de proposer des pistes d'amélioration pour le développement de la plateforme PGTM-USTHB.

Chapitre 1

Étude de l'existant

1.1 Introduction

Avant d'entamer le projet de conception d'une nouvelle plateforme pour la gestion des thèses et mémoires, pour l'USTHB, sa situation d'aujourd'hui doit être bien comprise, les méthodes mobilisées par l'université doivent être examinées et les solutions existantes dans d'autres établissements académiques étudiées [1]. En d'autres termes, il importe de se pencher sur l'état de l'art pour faire émerger les possibilités. Cela revue de l'existant vise à identifier les forces et les faiblesses des systèmes mis en place jusqu'alors, et précisément à cerner les besoins de l'USTHB pour pouvoir justifier la solution mise en avant par cette proposition. Ce chapitre fait donc état d'un recensement de la situation initiale de l'USTHB, d'une enquête sur les dispositifs numériques dédiés à la gestion des dépôts institutionnels, d'une évaluation des avantages et inconvénients des solutions identifiées au regard des besoins listés, et enfin d'une proposition formelle de la solution projetée.

1.2 Analyse des solutions existantes

1.2.1 Méthode actuelle à l'USTHB

Actuellement, l'Université des Sciences et de la Technologie Houari Boumediene ne dispose ni d'un dispositif numérique central et dédié à la gestion intégrale et à la consultation aisée des thèses de doctorat, ni d'une solution similaire pour les mémoires de Master. Les pratiques de gestion de ces documents reposent sur l'appréhension de procédures bien souvent discontinues et manuelles :

- **Dépôt physique** : Les étudiants sont le plus souvent contraints à déposer des exemplaires physiques de leurs thèses ou mémoire auprès de l'administration de leur faculté ou département et à la bibliothèque centrale.
- **Catalogage limité** : La bibliothèque centrale conserve un catalogue, ce catalogue peut ne pas être exhaustif, fonctionnel et accessible sur les sites Web de la bibliothèque avec des capacités de recherche performante. Le plus souvent, la recherche s'effectue à travers des catalogues informatisés localement (SIGB – Système Intégré de Gestion de Bibliothèque) dont les fonctionnalités de recherche peuvent être restreintes, vers un texte intégral ou à partir de mots-clés.
- **Listes départementales** : certains départements peuvent tenir leur propre liste de travaux soutenus, le plus souvent sous forme de documents bureautiques ou de pages Web statiques ; le tout étant trop souvent non archivées, non normalisées et non interconnectées.

- **Accès restreint** : Pour pouvoir consulter réellement les travaux, il est nécessaire de se déplacer physiquement, soit à la bibliothèque ou au département concernés. L'accès numérique éventuellement proposé se limiterait aux résumés ou aux métadonnées, et non au texte intégral ou bien serait fermé sur le réseau interne de l'université.

Ces paradigmes de communication et de disponibilité des documents présentent de nombreux inconvénients : invisibilité des travaux, difficulté d'accès pour chercheurs et étudiants, problème de perte ou de répétition d'informations, poids administratif pour le dépôt et la gestion, et absence de statistiques sur la production scientifique de l'université dans le domaine [2]. Il est donc crucial de moderniser et de centraliser via une plateforme numérique dédiée.

1.2.2 Solutions numériques générales

Il est intéressant de noter qu'en réponse à des problèmes identiques, de nombreuses universités et des institutions de recherche classiques à travers le monde ont mis au point ou adopté des solutions numériques pour divers usages de gestion des dépôts institutionnels. On peut distinguer trois grandes catégories de solutions :

Logiciels Libres de Dépôt Institutionnel (Exemples : DSpace, EPrints)

Ces plateformes sont des logiciels open-source spécifiquement conçus pour créer des archives numériques ouvertes.

- **DSpace** : Développé en coopération entre HP Labs et le MIT, DSpace est une solution très répandue, utilisée par des milliers d'institutions, pour capturer, stocker, indexer, préserver et redistribuer le matériel numérique de recherche d'une institution. Écrit en Java et utilisant la base de données de PostgreSQL, il propose une interface web configurable, différents formats de métadonnées (Dublin Core, par exemple), permet le dépôt selon des workflows paramétrables et dispose de fonctionnalités de recherche avancées [3].

Avantages d'usage

- Très visible, le bouton « Submit » sur la page d'accueil reprend le chemin vers l'interface de dépôt.
- Workflow pas-à-pas (assistant) pour la saisie des métadonnées (barre de progression présente).
- Champs extensibles : possibilité d'ajouter de nouveaux champs au besoin (par ex. : « Directeur de thèse » ou « Laboratoire »).
- Recherche avancée et filtres en tête de page pour affiner selon auteur, année ou mots-clés.

Inconvénients d'usage

- Absence de prévisualisation instantanée du fichier téléversé : l'utilisateur ne peut pas voir le rendu final dans le dépôt, ni s'assurer que le format du document est conforme.
- Interface non responsive : usage difficile sur smartphones et surtout petits écrans.
- Pas de notifications en temps réel ; l'étudiant doit rafraîchir la page pour savoir si le dépôt est validé.
- Les médias (vidéos, audio) ne se déposent que par le biais de plugins.

Fonctionnalités manquantes

1. Glisser-déposer (drag & drop) qui pourrait permettre le téléversement groupé de plusieurs fichiers.
 2. Autorisation de la création de l'intégration dans ORCID pour permettre l'import automatique des données des chercheurs.
 3. Gérer les versions des dépôts (versioning) avec historisation de l'ensemble des modifications.
 4. Réservée, l'interface multilingue (arabe/anglais) simultanée.
- **EPrints** : EPrints, une autre solution open-source de référence développée surtout par l'université de Southampton, créée en Perl, a pour objet d'édifier des dépôts institutionnels selon des standards conformes à l'Open Archives Initiative (OAI-PMH), pour l'interopérabilité. Les fonctionnalités promise concernent les soumissions, mais aussi la gestion, la recherche et la sauvegarde au même titre que DSpace. [4]

Les bénéfices à tirer de cet outil de dépôt

- Un formulaire de dépôt simple et épuré avec des zones bien définies pour le titre, l'auteur et le fichier.
- L'info-bulle sur chaque champ permet un guidage de l'utilisateur au moment de l'auto-évaluation.
- Environnement personnalisable aisément en CSS pour coller à la charte graphique de l'établissement d'enseignement supérieur.
- Moteur de recherche efficace intégré avec auto-suggestion des mots-clés.

Les freins à l'utilisation

- Pas de prévisualisation PDF au chargement, juste un lien pour charger le dépôt.
- Workflow rigide : pas de processus avec validation intermédiaire (pré-validation puis validation).
- Notifications par mail peu encourageantes : l'étudiant ne peut pas être alerté du suivi avancé en temps.
- Interface quelque peu obscurcie (en matière d'icônes, de palette de couleur).

Les fonctionnalités manquantes

1. Dashboard qui permet d'indiquer le statut de chaque dépôt fait par l'étudiant (en attente, réception, rejet).
 2. Chargement en mode bulk pour faire des dépôts en masse pour les groupes d'étudiants.
 3. E-mail anticipé avec intégration Gmail/Office 365 pour envoyer des rappels.
- **Vireo (Texas Digital Library)**

Description

Solution open-source dédiée aux Electronic Theses & Dissertations (ETD). Elle couvre le workflow complet : dépôt en ligne, validation par les services académiques, puis publication dans le réservoir institutionnel. [5]

Avantages de l'outil

- **Workflow ETD complet** : formulaires et étapes préconfigurés pour mémoires de Master et thèses de Doctorat.
- **Intégration OAI-PMH** : publication dans DSpace, Fedora ou tout autre réservoir compatible.
- **Gratuit & Open-source** : pas de licence à acquitter, code Java fourni pour adapter les processus et les interfaces.
- **Suivi en temps réel** : étudiant et encodeurs peuvent consulter l'état d'avancement (en attente, validé, publié) depuis l'interface.

Inconvénients de l'outil

- **Installation complexe** : serveur Java (GlassFish ou Tomcat), base de données relationnelle, compétences Java/SQL requises.
- **Interface simple** : design fonctionnel, peu esthétique comparé aux solutions commerciales ; courbe d'apprentissage pour les novices.
- **Fonctionnalités statistiques limitées** : absence de tableau de bord pour visualiser les consultations ou téléchargements.
- **Pas de version mobile** : non accessible depuis smartphone ou tablette.

Fonctionnalités absentes

1. Glisser-déposer (Drag & Drop) permettant le téléversement groupé de plusieurs fichiers.
2. Notifications (mail ou in-app) à chaque étape de la validation.
3. Tableau de bord dédié montrant l'historique et les statistiques de chaque dépôt.

Ces solutions sont réputées, éprouvées, développées et maintenues par de nombreuses équipes sur le terrain [6]. Leurs mises en œuvre (en conditions réelles) supposent un taux de compétence, de compétences techniques spécifiques (Java/Perl, administration système, configuration XML). Il est à souligner que si l'interface s'avère fonctionnelle, elle se laisse parfois percevoir comme moins moderne, voire intuitive qu'il n'apparaît pour des solutions plus récentes.

Plateformes Commerciales

Il existe de plus des solutions commerciales qui peuvent être proposées par des éditeurs spécialisés dans le secteur des logiciels pour bibliothèques et institutions académiques (par exemple Ex Libris Esplores [7], Digital Commons de Bepress [8], Elsevier [?]) et qui proposent souvent une offre fonctionnelle relativement riche (dans la gestion de la recherche, les profils chercheurs, les indicateurs d'impact) et un support technique. Leur principal inconvénient réside généralement dans leurs coûts (licences, maintenance) parfois jugés prohibitifs pour un nombre d'établissements, mais aussi si elles semblent souples et adaptables à des projets particuliers de recherche dans le cas des solutions open-source.

1.3 Conclusion

Pour conclure, l'analyse de l'existant pointe une gestion la plus partielle, et la plus souvent manuelle aux USTHB, des mémoires et thèses est source de sources d'accès et de valorisation de ces travaux peu satisfaisant [9]. Les plateformes open-source (DSpace,

EPrints, Vireo) sont flexibles et respectueuses des standards, mais nécessitent des compétences techniques et des moyens internes d'entretien considérables. Les solutions commerciales et SaaS permettent un déploiement rapide et des fonctionnalités avancées, mais à un coût prohibitif et avec une forte dépendance au fournisseur. N'aucun de ces choix ne parvient à concilier personnalisation, maîtrise du coût, évolutivité et intégration aux procédures locales. Il est donc nécessaire de concevoir une plateforme sur-mesure, modulaire et aisément utilisable, pour répondre aux besoins spécifiques de l'USTHB, et de contribuer à la pérennité de sa production académique.

Chapitre 2

Conception de la Plateforme de Gestion des Thèses et Mémoires

2.1 Introduction à la Conception

La conception de notre plateforme de gestion des thèses et mémoires universitaires représente une étape cruciale dans le développement de ce système d'information [10]. Cette phase a nécessité une réflexion approfondie sur l'architecture logicielle, le modèle de données et les interactions entre les différents composants du système. Notre approche de conception s'est articulée autour de plusieurs principes fondamentaux :

- Une architecture en couches clairement séparées pour faciliter la maintenance et l'évolution
- Un modèle de données robuste et normalisé pour garantir l'intégrité des informations
- Une intégration harmonieuse avec les services cloud pour le stockage et l'authentification
- Une interface utilisateur intuitive adaptée aux différents profils d'utilisateurs

Cette phase de conception a permis de poser les fondations solides sur lesquelles repose l'implémentation de notre plateforme, en garantissant que celle-ci réponde efficacement aux besoins identifiés lors de l'analyse préliminaire.

2.2 Architecture Globale du Système

L'architecture de notre plateforme de gestion des thèses et mémoires suit un modèle en couches, inspiré du pattern d'architecture MVVM (Model-View-ViewModel), adapté au contexte d'une application WPF [11]. Cette organisation modulaire permet une séparation claire des responsabilités et facilite la maintenance du code.

2.2.1 Vue d'Ensemble de l'Architecture

Notre architecture se compose de quatre couches principales :

1. **Couche Présentation** : Interface utilisateur WPF avec les vues XAML et leur code-behind
2. **Couche Métier** : Modèles de données, services métier et gestion de session
3. **Couche Accès aux Données** : Classes d'accès à la base de données et aux services cloud

4. Services Externes : Firebase Auth, Google Cloud SQL et Google Cloud Storage

Cette organisation permet une séparation claire des préoccupations, où chaque couche a une responsabilité spécifique et communique avec les autres couches via des interfaces bien définies.

2.2.2 Diagramme d'Architecture

Le diagramme suivant illustre l'architecture globale de notre système, mettant en évidence les différentes couches et leurs interactions :

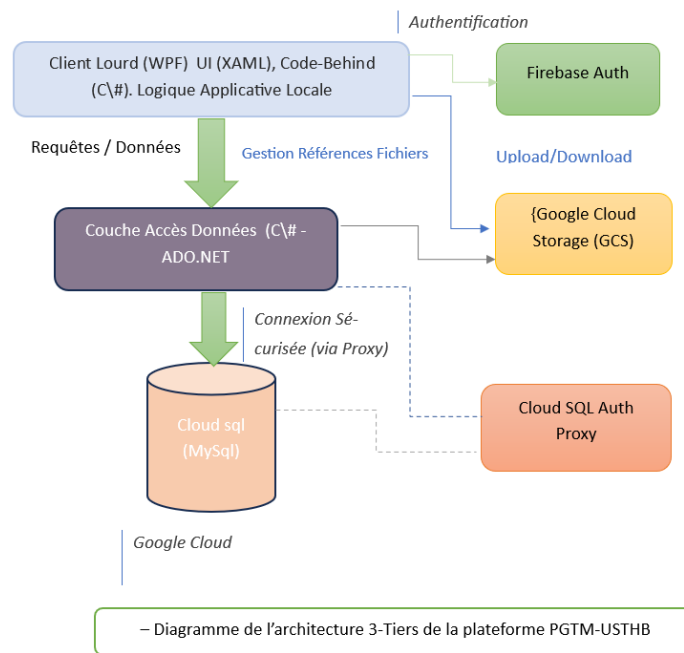


FIGURE 2.1 – Diagramme de l'architecture 3-Tiers de la plateforme PGTM-USTHB

2.2.3 Description des Composants Principaux

Couche Présentation

- **Vues XAML** : Définissent l'interface utilisateur et les interactions visuelles
- **Code-Behind** : Implémente la logique de présentation et gère les événements utilisateur

Couche Métier

- **Modèles** : Représentent les entités métier (User, Theses, Favoris, NewsItem)
- **Services Métier** : Implémentent la logique métier spécifique (CloudStorageService)
- **Gestion de Session** : Maintient l'état de l'utilisateur connecté et ses informations

Couche Accès aux Données

- **Accès Base de Données** : Gère les connexions et requêtes à la base de données MySQL

- **Accès Cloud Storage** : Facilite l'interaction avec Google Cloud Storage pour les fichiers

Services Externes

- **Firestore Auth** : Service d'authentification pour la gestion des utilisateurs [12]
- **Google Cloud SQL** : Base de données relationnelle hébergée dans le cloud [13]
- **Google Cloud Storage** : Stockage d'objets pour les fichiers PDF des thèses [14]

2.2.4 Flux de Données et Interactions

Les interactions entre les composants suivent généralement un flux descendant pour les commandes et ascendant pour les données :

1. L'utilisateur interagit avec l'interface WPF
2. Le code-behind traite ces interactions et sollicite les services métier
3. Les services métier utilisent les modèles et les classes d'accès aux données
4. Les classes d'accès aux données communiquent avec les services externes
5. Les résultats remontent la chaîne jusqu'à l'interface utilisateur

Cette organisation garantit une séparation claire des responsabilités et facilite la maintenance et l'évolution du système.

2.3 Modèle de Données

Le modèle de données constitue la colonne vertébrale de notre système de gestion des thèses et mémoires [15]. Il a été conçu pour représenter fidèlement les entités du domaine, leurs attributs et les relations qui les lient.

2.3.1 Diagramme Entité-Association

Le diagramme entité-association suivant illustre la structure de notre base de données :

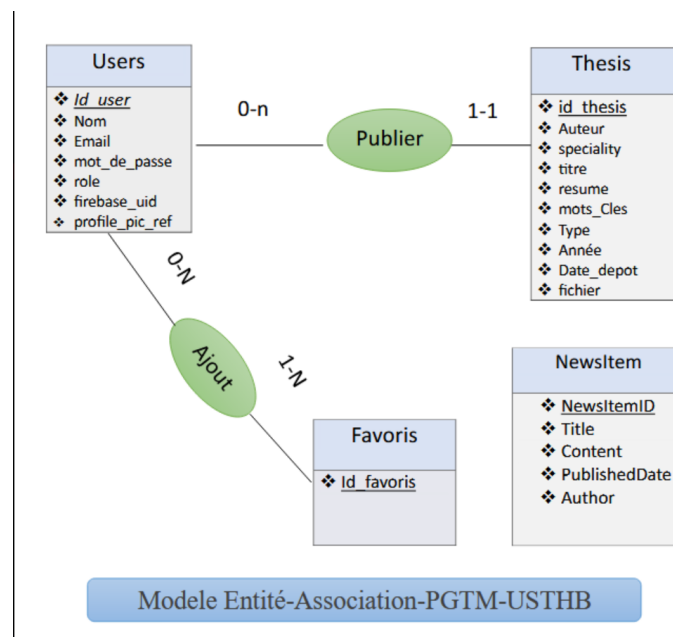


FIGURE 2.2 – Diagramme du modèle Entite Association

2.3.2 Description des Entités Principales

Entité User

L'entité User représente les utilisateurs du système avec différents rôles (administrateur, utilisateur simple, étudiant).

Attributs principaux :

- **id** : Identifiant unique (clé primaire)
- **nom** : Nom complet de l'utilisateur
- **email** : Adresse email (unique)
- **password** : Mot de passe (stocké sous forme de référence à Firebase Auth)
- **role** : Rôle de l'utilisateur (Admin, SimpleUser, Etudiant)
- **firebase_uid** : Identifiant unique Firebase (pour l'authentification)
- **profile_pic_ref** : Référence à l'image de profil dans Cloud Storage

Entité Theses

L'entité Theses représente les documents académiques (thèses et mémoires) stockés dans le système.

Attributs principaux :

- **id** : Identifiant unique (clé primaire)
- **titre** : Titre du document
- **auteur** : Nom de l'auteur du document
- **speciality** : Domaine de spécialité
- **type** : Type de document (Doctorat, Master)
- **mots_cles** : Mots-clés pour la recherche
- **annee** : Année de publication
- **resume** : Résumé du contenu
- **fichier** : Référence au fichier PDF dans Cloud Storage
- **user_id** : Identifiant de l'utilisateur qui a soumis le document (clé étrangère)

Entité Favoris

L'entité Favoris établit une relation many-to-many entre les utilisateurs et les thèses qu'ils ont marquées comme favorites.

Attributs principaux :

- **id** : Identifiant unique (clé primaire)
- **user_id** : Identifiant de l'utilisateur (clé étrangère)
- **these_id** : Identifiant de la thèse (clé étrangère)

Entité NewsItem

L'entité NewsItem représente les actualités publiées sur la plateforme. Elle contient les informations essentielles telles que le titre, le contenu, la date de publication et l'auteur.

Attributs principaux :

- **NewsItemID** : Identifiant unique de l'actualité (clé primaire)
- **Title** : Titre de l'actualité
- **Content** : Contenu textuel détaillé de l'actualité
- **PublishedDate** : Date et heure de publication
- **Author** : Nom de l'auteur de l'actualité (optionnel)

2.3.3 Relations et Cardinalités

- **User - Theses** : Relation one-to-many (un utilisateur peut publier plusieurs thèses)
- **User - Favoris** : Relation one-to-many (un utilisateur peut avoir plusieurs favoris)

2.3.4 Normalisation et Intégrité Référentielle

Notre modèle de données respecte les principes de normalisation (jusqu'à la 3ème forme normale) pour éviter la redondance et garantir l'intégrité des données [16]. L'intégrité référentielle est assurée par les contraintes de clés étrangères, qui garantissent que :

- Une thèse ne peut pas exister sans un utilisateur qui l'a publiée
- Un favori ne peut pas exister sans l'utilisateur et la thèse correspondants

2.4 Conception de l'Interface Utilisateur

L'interface utilisateur de notre plateforme a été conçue pour offrir une expérience intuitive et efficace, adaptée aux différents profils d'utilisateurs (administrateurs, utilisateurs simples, étudiants) [17].

2.4.1 Principes de Conception UI

Notre conception UI s'appuie sur plusieurs principes fondamentaux :

- **SimPLICITÉ** : Interfaces épurées et focalisées sur les tâches essentielles
- **Cohérence** : Éléments visuels et interactions uniformes à travers l'application
- **Feedback** : Retour visuel clair pour toutes les actions utilisateur
- **Accessibilité** : Interfaces utilisables par tous, y compris les utilisateurs ayant des besoins spécifiques

2.4.2 Écrans Principaux et Flux Utilisateur

Écran de Connexion/Inscription

- Authentification des utilisateurs existants
- Création de nouveaux comptes avec vérification d'email
- Sélection du rôle lors de l'inscription

Écran Principal (Dashboard)

- Vue d'ensemble des thèses récentes
- Accès aux fonctionnalités principales via un menu latéral
- Zone de recherche pour trouver rapidement des documents

Écran de Gestion des Thèses

- Liste des thèses publiées par l'utilisateur
- Fonctionnalités d'ajout, modification et suppression
- Visualisation des statistiques de consultation

Écran de Profil Utilisateur

- Informations personnelles de l'utilisateur
- Modification des données du profil
- Gestion de l'image de profil

2.4.3 Adaptabilité et Responsive Design

Bien que notre application soit principalement destinée à une utilisation sur ordinateur (application WPF), nous avons veillé à ce que les interfaces s'adaptent à différentes tailles d'écran et résolutions, garantissant ainsi une expérience utilisateur optimale dans tous les contextes d'utilisation.

2.5 Algorithmes et Pseudo-code

Cette section présente l'algorithme clé de notre plateforme sous forme de pseudo-code, illustrant la logique métier essentielle du système.

2.5.1 Algorithme d'Authentification

```
1 Fonction Authentifier(email, motDePasse)
2     // Vérification des entrées
3     Si email est vide OU motDePasse est vide Alors
4         Retourner Erreur("Veuillez remplir tous les champs")
5     FinSi
6
7     // Authentification via Firebase
8     Essayer
9         réponseFirebase AppelFirebaseAuth(email, motDePasse)
10        Si réponseFirebase est Erreur Alors
11            Retourner Erreur("Identifiants incorrects")
12        FinSi
13
14        // Récupération de l'identifiant Firebase
15        firebaseUid réponseFirebase.LocalId
16        idToken réponseFirebase.IdToken
17
18        // Vérification de l'email (sauf pour admin)
19        Si email != "admin@yourapp.com" ET !réponseFirebase.EmailVerified Alors
20            Retourner Erreur("Email non vérifié")
21        FinSi
22
23        // Recherche de l'utilisateur dans la base de données
24        requête "SELECT id, nom, role, email FROM users WHERE firebase_uid =
@FirebaseUid"
25        utilisateur ExécuterRequêteBDD(requête, {FirebaseUid: firebaseUid})
26
27        Si utilisateur est vide Alors
28            Retourner Erreur("Utilisateur non trouvé dans la base de données")
29        FinSi
30
31        // Initialisation de la session
32        Session.Initialiser(
33            utilisateur.Id,
34            utilisateur.Nom,
35            utilisateur.Role,
36            firebaseUid,
37            idToken
38        )
```

```
39
40     Retourner Succès(utilisateur)
41     Attraper Exception ex
42     Retourner Erreur("Erreur d'authentification: " + ex.Message)
43     FinEssayer
44 FinFonction
```

Conclusion du Chapitre

La phase de conception détaillée élaborée dans ce chapitre a permis d'établir l'ossature architecturale, fonctionnelle et structurelle de la plateforme de gestion des thèses et mémoires. Grâce à un choix modulaire s'appuyant sur une architecture en couches et à l'utilisation de services cloud fiables, nous avons apporté plus d'évolutivité, de sécurité et de maintenabilité au système.

Le modèle de données proposé peut représenter de manière juste les entités métier ainsi que les relations qui les relie tout en respectant l'intégrité référentielle par le biais d'une normalisation rigoureuse. L'interface utilisateur, conçue pour être intuitive et accessible dans l'idée de répondre aux besoins diversifiés des différents profils d'utilisateurs, garantit une adoption aisée de la plateforme.

Enfin, les algorithmes clés décrits montrent une mise en œuvre logique des fonctions nécessaires comme l'authentification et la recherche avancée et attestent par là une cohérence entre les choix de conception et les objectifs fonctionnels de la plateforme que cette conception rigoureuse peut légitimement revendiquer comme une future base fiable pour le stade suivant : mise en œuvre et déploiement.

Chapitre 3

Réalisation et Test

3.1 Introduction à l'Implémentation

L'implémentation de notre plateforme de gestion des thèses et mémoires constitue la concrétisation des concepts et structures définis lors de la phase de conception [20]. Cette étape a nécessité la traduction des modèles abstraits en code fonctionnel, en respectant les principes architecturaux établis et en intégrant les technologies sélectionnées.

Ce chapitre présente les aspects techniques de l'implémentation, les défis rencontrés et les solutions apportées, ainsi que les résultats obtenus. Il aborde également les perspectives d'évolution et d'amélioration de la plateforme pour répondre aux besoins futurs des utilisateurs et de l'institution.

L'implémentation s'est déroulée selon une approche méthodique, en commençant par la mise en place de l'infrastructure technique, suivie du développement des fonctionnalités essentielles, pour finir par l'intégration des services externes et l'optimisation des performances.

3.2 Technologies et Outils Utilisés

3.2.1 Environnement de Développement

Notre plateforme a été développée dans un environnement moderne, utilisant les outils et technologies suivants :

- **Visual Studio 2022** : Environnement de développement intégré (IDE) principal
- **Git** : Système de contrôle de version pour la gestion du code source
- **GitHub** : Plateforme collaborative pour l'hébergement du code et la gestion des versions
- **SQL Management Studio** : Outil de gestion et d'administration de la base de données

3.2.2 Technologies Frontend

Pour le développement de l'interface utilisateur, nous avons utilisé :

- **C#** : Langage de programmation principal
- **WPF (Windows Presentation Foundation)** : Framework d'interface utilisateur [21]
- **XAML** : Langage déclaratif pour la définition des interfaces [22]

- **Material Design** : Bibliothèque de composants UI pour un design moderne et cohérent [18]

3.2.3 Technologies Backend et Stockage

Pour la gestion des données et la logique métier, nous avons implémenté :

- **MySQL** : Système de gestion de base de données relationnelle
- **Google Cloud SQL** : Service de base de données relationnelle hébergée dans le cloud [13]
- **Google Cloud Storage** : Service de stockage d'objets pour les fichiers PDF [14]
- **Firebase Authentication** : Service d'authentification et de gestion des utilisateurs [12]
- **Cloud Functions** : Fonctions serverless pour la génération d'URLs signées

3.2.4 Bibliothèques et Frameworks

Plusieurs bibliothèques ont été intégrées pour faciliter le développement :

- **MySQL.Data** : Connecteur MySQL pour .NET
- **Newtonsoft.Json** : Bibliothèque de sérialisation/désérialisation JSON [23]
- **System.Net.Http** : Bibliothèque pour les requêtes HTTP
- **PdfiumViewer** : Composant d'affichage de documents PDF

3.2.5 Implémentation de l'Architecture

Structure du projet

- **Models/** : entités (`User`, `Theses`, `Favoris`, `NewsItem`)
- **Services/** : logique métier (`CloudStorageService.cs`)
- **View/** : vues WPF par rôle (`Admin`, `Étudiant`, `SimpleUser`) et écrans (`Login.xaml`, `MainWindow.xaml`, ...)
- Fichiers centraux : `DatabaseConnection.cs`, `Session.cs`, `AppConfig.cs`

Couche Présentation

- Implémentée en WPF/XAML selon le pattern MVVM
- Exemple dans `Login.xaml.cs` : validation des champs, test de connexion, gestion du curseur et des exceptions

Couche Métier

- Classes C# annotées (`[Key]`, `[Required]`, `[MaxLength]`, `[EmailAddress]`)
- Propriétés métiers et collections de relations (`Thèses`, ...)

Couche Accès aux Données

- Classe statique `DatabaseConnection` avec méthode `ExecuteQuery(query, params...)`
- Requêtes paramétrées (`SqlParameter`) pour prévenir les injections SQL
- Lecture via `DbDataReader` et construction de dictionnaires clé-valeur
- Gestion centralisée des exceptions et fermeture automatique des connexions

3.3 Fonctionnalités Implémentées

3.3.1 Gestion des Utilisateurs

La plateforme implémente une gestion complète des utilisateurs avec les fonctionnalités suivantes [25] :

- **Inscription** : Création de compte avec vérification d'email
- **Authentification** : Connexion sécurisée via Firebase Auth
- **Gestion de profil** : Modification des informations personnelles et de l'image de profil
- **Gestion des rôles** : Différenciation des droits selon le type d'utilisateur (Admin, SimpleUser, Etudiant)

3.3.2 Gestion des Thèses et Mémoires

Le cœur fonctionnel de la plateforme concerne la gestion des documents académiques :

- **Soumission** : Upload de fichiers PDF avec métadonnées associées
- **Consultation** : Visualisation en ligne des documents avec un lecteur PDF intégré
- **Recherche** : Recherche multicritères (titre, auteur, mots-clés, spécialité, année)
- **Favoris** : Marquage de thèses comme favorites pour un accès rapide

3.3.3 Fonctionnalités Administratives

Les administrateurs disposent de fonctionnalités spécifiques :

- **Validation de thèses** : Approbation des documents soumis avant publication
- **Gestion des utilisateurs** : Modification des rôles et statuts des comptes
- **Statistiques** : Visualisation des métriques d'utilisation de la plateforme

3.4 Sécurité et Performance

3.4.1 Mesures de Sécurité Implémentées

Plusieurs mesures ont été mises en place pour garantir la sécurité de la plateforme [26] :

- **Authentification robuste** : Utilisation de Firebase Auth avec vérification d'email
- **Protection des données sensibles** : Aucun stockage de mot de passe en clair
- **Contrôle d'accès** : Vérification systématique des droits avant chaque opération sensible
- **Prévention des injections SQL** : Utilisation de requêtes paramétrées
- **Accès sécurisé aux fichiers** : URLs signées à durée limitée pour les documents PDF

3.4.2 Optimisations de Performance

Pour garantir une expérience utilisateur fluide, plusieurs optimisations ont été implémentées :

- **Chargement asynchrone** : Utilisation de `async/await` pour les opérations longues
- **Mise en cache** : Stockage temporaire des données fréquemment utilisées
- **Pagination** : Chargement par lots des résultats de recherche volumineux
- **Compression** : Réduction de la taille des fichiers transmis

- **Lazy loading** : Chargement différé des ressources non immédiatement nécessaires

3.5 Tests et Validation

3.5.1 Stratégie de Test

Notre approche de test a combiné plusieurs niveaux de validation :

- **Tests unitaires** : Validation des composants individuels
- **Tests d'intégration** : Vérification des interactions entre composants
- **Tests fonctionnels** : Validation des fonctionnalités complètes
- **Tests de performance** : Évaluation du comportement sous charge

3.5.2 Résultats des Tests

Les tests ont permis d'identifier et de corriger plusieurs problèmes :

- Optimisation des requêtes SQL pour améliorer les temps de réponse
- Correction des fuites mémoire dans la gestion des fichiers PDF
- Amélioration de la gestion des erreurs réseau lors des appels aux services cloud
- Renforcement de la validation des entrées utilisateur

3.6 Difficultés Rencontrées et Solutions

3.6.1 Défis Techniques

Plusieurs défis techniques ont été relevés durant l'implémentation :

- **Intégration avec Firebase Auth** : La gestion des tokens et des droits a nécessité une implémentation personnalisée
- **Accès aux fichiers dans Google Cloud Storage** : L'utilisation d'URLs signées a résolu les problèmes d'accès sécurisé
- **Synchronisation des données** : L'implémentation de mécanismes de rafraîchissement automatique a amélioré la cohérence
- **Performance des requêtes** : L'optimisation des requêtes SQL et l'ajout d'index ont résolu les problèmes de lenteur

3.6.2 Solutions Implémentées

Pour chaque défi, des solutions spécifiques ont été mises en œuvre [27] :

- **Classe de session centralisée** : Gestion unifiée de l'authentification et des droits
- **Service dédié pour le stockage cloud** : Encapsulation des opérations complexes
- **Mécanisme de cache avec invalidation** : Amélioration des performances tout en garantissant la fraîcheur des données
- **Requêtes paramétrées et optimisées** : Sécurité et performance accrues

3.7 Résultats et Interfaces Utilisateur

3.7.1 Écran de Connexion

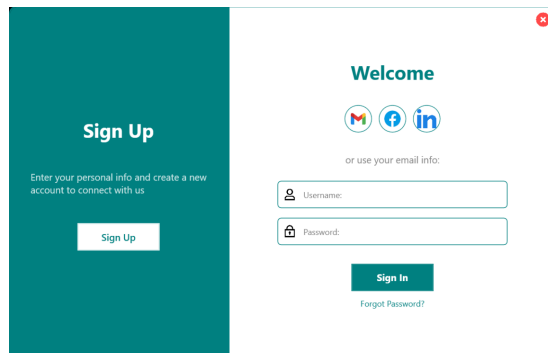


FIGURE 3.1 – sign in

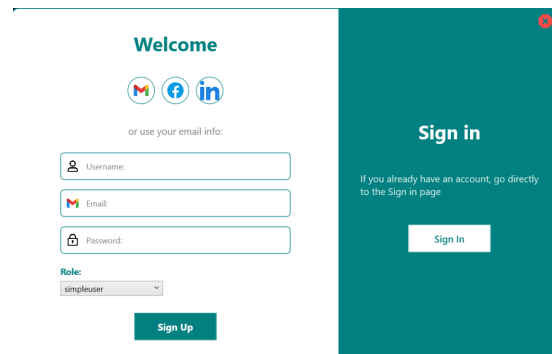


FIGURE 3.2 – sign up

L'écran de connexion offre une interface épurée permettant aux utilisateurs de s'authentifier ou de créer un nouveau compte. Il comprend :

- Champs de saisie pour l'email et le mot de passe
- Bouton de connexion avec retour visuel pendant le processus
- Lien vers le formulaire d'inscription
- Option de récupération de mot de passe

3.7.2 Dashboard Principal

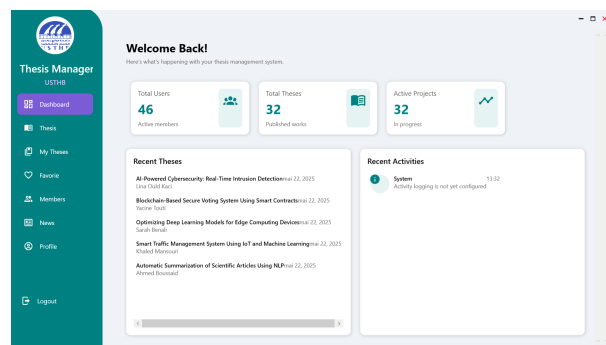


FIGURE 3.3 – dashboard

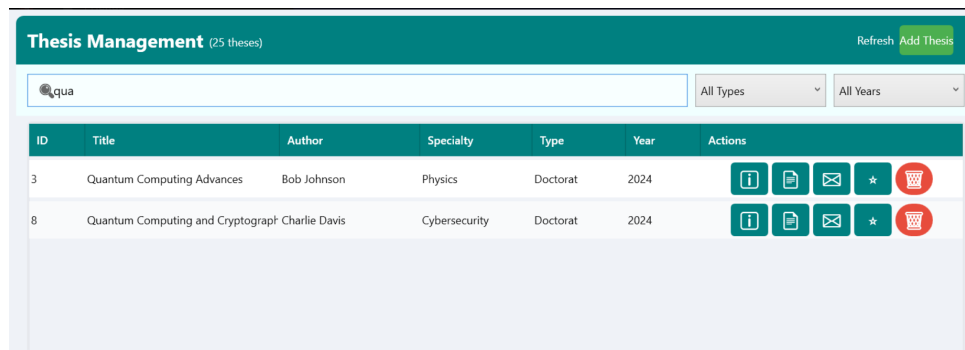
Le dashboard principal présente une vue d'ensemble des fonctionnalités de la plateforme :

- Menu latéral pour accéder aux différentes sections
- Liste des thèses récentes avec options de filtrage
- Statistiques personnelles (nombre de thèses consultées, favoris)

3.7.3 Interface de Recherche et Consultation

L'interface de recherche permet de trouver rapidement des documents [28] :

- Barre de recherche avec suggestions
- Filtres avancés (type, spécialité, année, auteur, mots clés , ID)
- Affichage des résultats avec prévisualisation



Thesis Management (25 theses) Refresh Add Thesis

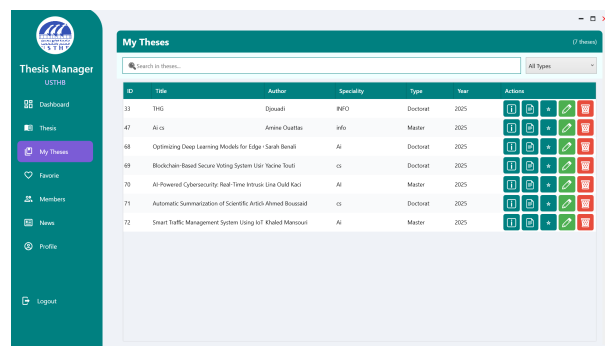
Search: qua All Types All Years

ID	Title	Author	Specialty	Type	Year	Actions
3	Quantum Computing Advances	Bob Johnson	Physics	Doctorat	2024	[Info] [PDF] [Email] [Star] [Trash]
8	Quantum Computing and Cryptograph	Charlie Davis	Cybersecurity	Doctorat	2024	[Info] [PDF] [Email] [Star] [Trash]

FIGURE 3.4 – Recherche

— Lecteur PDF intégré pour la consultation des documents

3.7.4 Interface pour voir et modifier les thèses d'un utilisateur



My Theses (7 theses)

Search in theses... All Types

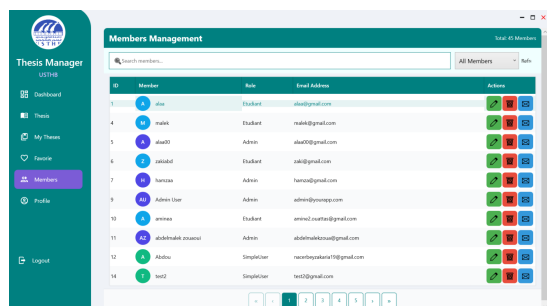
ID	Title	Author	Specialty	Type	Year	Actions
33	THG	Ehsanli	INFO	Doctorat	2025	[Info] [PDF] [Email] [Star] [Trash]
47	Alcis	Annie Chastan	Info	Master	2025	[Info] [PDF] [Email] [Star] [Trash]
68	Optimizing Deep Learning Models for Edge	Sarah Bouali	AI	Doctorat	2025	[Info] [PDF] [Email] [Star] [Trash]
69	Blockchain-Based Secure Voting System Using	Techne Tsoul	CS	Doctorat	2025	[Info] [PDF] [Email] [Star] [Trash]
70	AI-Powered Cybersecurity: Real-Time Intrusion	Uma Ould Kaci	AI	Master	2025	[Info] [PDF] [Email] [Star] [Trash]
71	Automatic Summarization of Scientific Articles	Ahmed Boussad	CS	Doctorat	2025	[Info] [PDF] [Email] [Star] [Trash]
72	Smart Traffic Management System Using IoT	Khalid Mouscoul	AI	Master	2025	[Info] [PDF] [Email] [Star] [Trash]

FIGURE 3.5 – Mes theses

3.7.5 Interface d'Administration

L'interface d'administration offre des outils de gestion avancés :

- Tableau de bord avec statistiques globales
- Gestion des utilisateurs (modification des rôles, désactivation de comptes)
- Validation des thèses soumises
- Configuration des paramètres système

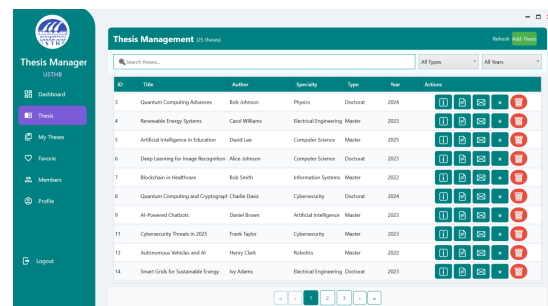


Members Management (User: Admin)

Search members... All Members Ref

ID	Member	Role	Email Address	Actions
1	John	Student	john@gmail.com	[Info] [PDF] [Email] [Star] [Trash]
4	Maria	Student	maria@gmail.com	[Info] [PDF] [Email] [Star] [Trash]
5	JohnD	Admin	johnD@gmail.com	[Info] [PDF] [Email] [Star] [Trash]
6	JohnD	Student	john@gmail.com	[Info] [PDF] [Email] [Star] [Trash]
7	JohnD	Admin	john@gmail.com	[Info] [PDF] [Email] [Star] [Trash]
8	Admin User	Admin	admin@university.com	[Info] [PDF] [Email] [Star] [Trash]
10	Annie	Student	annie@university.com	[Info] [PDF] [Email] [Star] [Trash]
11	Abdoulhakim Boussad	Admin	abdoulhakim@gmail.com	[Info] [PDF] [Email] [Star] [Trash]
12	Abdoul	SimpleUser	abdoul@gmail.com	[Info] [PDF] [Email] [Star] [Trash]
14	JohnD	SimpleUser	johnD@gmail.com	[Info] [PDF] [Email] [Star] [Trash]

FIGURE 3.6 – member list



Thesis Management (25 theses) Refresh Add Thesis

Search in theses... All Types All Years

ID	Title	Author	Specialty	Type	Year	Actions
3	Quantum Computing Advances	Bob Johnson	Physics	Doctorat	2024	[Info] [PDF] [Email] [Star] [Trash]
4	Renewable Energy Systems	Carol Williams	Electrical Engineering	Master	2023	[Info] [PDF] [Email] [Star] [Trash]
5	Artificial Intelligence in Education	David Lee	Computer Science	Doctorat	2023	[Info] [PDF] [Email] [Star] [Trash]
6	Deep Learning for Image Recognition	Alice Johnson	Computer Science	Master	2023	[Info] [PDF] [Email] [Star] [Trash]
7	Blockchain in Healthcare	Bob Smith	Information Systems	Doctorat	2024	[Info] [PDF] [Email] [Star] [Trash]
8	Quantum Computing and Cryptography	Charlie Davis	Cybersecurity	Doctorat	2024	[Info] [PDF] [Email] [Star] [Trash]
9	AI-Powered Chatbots	Daniel Brown	Artificial Intelligence	Master	2023	[Info] [PDF] [Email] [Star] [Trash]
11	Cybersecurity Threats in 2023	Frank Taylor	Cybersecurity	Master	2023	[Info] [PDF] [Email] [Star] [Trash]
13	Autonomous Vehicles and AI	Henry Clark	Robotics	Master	2022	[Info] [PDF] [Email] [Star] [Trash]
14	Smart Grids for Sustainable Energy	Ivy Adams	Electrical Engineering	Doctorat	2023	[Info] [PDF] [Email] [Star] [Trash]

FIGURE 3.7 – page admin theses

3.7.6 Interface de Consultation des Actualités

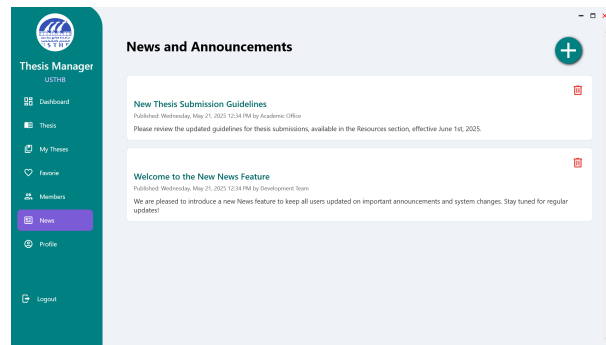


FIGURE 3.8 – Page des actualités

L'interface des actualités permet aux utilisateurs de rester informés des dernières annonces et événements publiés sur la plateforme :

- Affichage des actualités sous forme de cartes ou de liste
- Présentation du titre, de la date, de l'auteur (si disponible) et d'un aperçu du contenu
- Classement chronologique des publications (les plus récentes en premier)
- Bouton « Voir plus » pour consulter le contenu complet d'une actualité
- Fonctionnalités d'ajout, de modification et de suppression accessibles aux administrateurs

Conclusion

L'implémentation de notre plateforme de gestion des thèses et mémoires a permis de concrétiser la vision d'un système moderne, sécurisé et convivial pour la gestion des documents académiques. En s'appuyant sur des technologies éprouvées comme C#, WPF et les services cloud de Google, nous avons développé une solution qui répond aux besoins des différents acteurs universitaires [30].

Les choix techniques effectués, notamment l'architecture en couches et l'intégration des services cloud, ont permis de créer une base solide et évolutive. Les fonctionnalités implémentées couvrent l'ensemble du cycle de vie des thèses et mémoires, de la soumission à la consultation, en passant par la recherche et la communication entre utilisateurs.

Ce projet a également été l'occasion de mettre en pratique et d'approfondir nos connaissances en développement logiciel, en architecture d'applications et en intégration de services cloud, compétences essentielles dans le paysage technologique actuel.

Conclusion Générale

Au terme de ce travail, la plateforme **PGTM-USTHB** atteint l’objectif fixé dans l’*Introduction générale* : doter l’université d’un outil numérique unifié qui centralise le dépôt, la validation et la diffusion des thèses et mémoires, tout en garantissant sécurité, accessibilité et valorisation de la production scientifique. En remplaçant un processus essentiellement manuel par une solution logicielle en couches, adossée à *Firebase Auth*, *Google Cloud SQL* et *Google Cloud Storage*, nous avons levé les freins identifiés au début du projet — difficulté d’accès aux documents, absence de visibilité institutionnelle et lourdeur administrative — et démontré, par une implémentation fonctionnelle et des tests concluants, la pertinence d’un développement sur mesure adapté aux contraintes locales.

Réponse directe à la problématique :

- **Procédures réunifiées** : toutes les étapes, du dépôt au visa final, se font désormais dans un guichet web unique, sans formulaires papier ni déplacements.
- **Suivi transparent** : chaque action (correction, validation, mise en ligne) est horodatée et visible dans un tableau de bord partagé par le doctorant, l’encadrant et le laboratoire.
- **Conservation et sécurité** : les fichiers sont stockés sur un cloud redondant, chiffrés au repos comme en transit, avec sauvegardes planifiées pour garantir la pérennité des données.

Ces améliorations se traduisent déjà par des résultats concrets sur le terrain : le temps nécessaire pour effectuer le dépôt a été considérablement réduit, tout en garantissant un accès continu aux thèses validées 24 heures sur 24. De plus, les laboratoires disposent désormais d’indicateurs en temps réel pour suivre la performance scientifique de manière plus efficace. Enfin, les tests de charge ont confirmé que la plateforme est capable de s’adapter à l’augmentation future des volumes de données sans nécessiter de modifications majeures.

Perspectives d’évolution

- **Améliorations techniques** :
 - Architecture **MVVM** complète pour séparer clairement vue, logique et données ;
 - **API REST** afin d’ouvrir les fonctions clés aux autres services de l’université ;
 - **Suite de tests automatisés** (unitaires, intégration, non-régression) pour fiabiliser chaque mise à jour.
- **Nouvelles fonctionnalités** :
 - **Extraction automatique de métadonnées** lors du dépôt du PDF ;
 - **Recommandations personnalisées** selon les centres d’intérêt des utilisateurs ;

- **Annotations collaboratives** : Possibilité d'ajouter des notes et commentaires sur les documents
- **Version mobile** : Développement d'une application mobile complémentaire
- **Évolution vers une plateforme collaborative** :
 - **Espaces de travail partagés** doctorant–encadrant pour centraliser échanges et fichiers ;
 - **Suivi d'avancement** par jalons, rappels automatiques et tableau Kanban intégré ;
 - **Connexion fluide** aux systèmes académiques existants (ENT, SIGA, messagerie).

En définitive, **PGTM-USTHB** ne se contente pas de résoudre les blocages actuels ; elle pose les bases d'un environnement documentaire évolutif, capable de grandir avec les besoins de la communauté universitaire. À court terme, les améliorations techniques renforceront la robustesse du code ; à moyen terme, les nouvelles fonctionnalités enrichiront l'expérience utilisateur ; enfin, à long terme, la dimension collaborative fera de PGTM un véritable carrefour de recherche et de partage pour l'USTHB.

Bibliographie

- [1] COAR - Confederation of Open Access Repositories, "The Current State of Open Access Repository Interoperability", <https://www.coar-repositories.org/activities/repository-interoperability/>, consulté le 05 mai 2025.
- [2] Open Archives Initiative, "Protocol for Metadata Harvesting", <https://www.openarchives.org/pmh/>, consulté le 05 mai 2025.
- [3] DSpace, "DSpace Repository Software", <https://www.dspace.org/>, consulté le 05 mai 2025.
- [4] EPrints, "EPrints Repository Software", <https://uk.eprints-hosting.org/uk/index.php/eprints-software/>, consulté le 15 mai 2025.
- [5] Texas Digital Library, "Vireo ETD Submission and Management Software", <https://www.tdl.org/services/etds/>, consulté le 7 mai 2025.
- [6] Lynch, C. A. (2003), "Institutional Repositories : Essential Infrastructure for Scholarship in the Digital Age", ARL Bimonthly Report 226, 1-7.
- [7] Ex Libris, "Esploro Research Services Platform", <https://exlibrisgroup.com/products/esploro-research-services-platform/moving-beyond-institutional-repository/>, consulté le 8 mai 2025.
- [8] Elsevier, "Digital Commons", <https://www.elsevier.com/products/digital-commons>, consulté le 8 mai 2025.
- [9] Date, C. J. (2003), "An Introduction to Database Systems", 8th Edition, Addison-Wesley.
- [10] Sommerville, I. (2016), "Software Engineering", 10th Edition, Pearson.
- [11] Microsoft, "The MVVM Pattern", <https://learn.microsoft.com/en-us/dotnet/architecture/maui/mvvm>, consulté le 8 mai 2025.
- [12] Google Firebase, "Firebase Authentication", <https://firebase.google.com/docs/auth>, consulté le 11 mai 2025.
- [13] Google Cloud, "Cloud SQL Documentation", <https://cloud.google.com/sql/docs>, consulté le 11 mai 2025.
- [14] Google Cloud, "Cloud Storage Documentation", <https://cloud.google.com/storage/docs>, consulté le 11 mai 2025.
- [15] MySQL, "MySQL Documentation", <https://dev.mysql.com/doc/>, consulté le 11 mai 2025.
- [16] Object Management Group, "Unified Modeling Language Specification", <https://www.omg.org/spec/UML/>, consulté le 11 mai 2025.

- [17] Krug, S. (2014), "Don't Make Me Think, Revisited : A Common Sense Approach to Web Usability", 3rd Edition, New Riders.
- [18] Google, "Material Design", <https://m3.material.io/>, consulté le 12 mai 2025.
- [19] Internet Engineering Task Force, "The OAuth 2.0 Authorization Framework", <https://datatracker.ietf.org/doc/html/rfc6749>, consulté le 12 mai 2025.
- [20] Agile Alliance, "Agile Manifesto", <https://www.agilealliance.org/agile101/the-agile-manifesto/>, consulté le 12 mai 2025.
- [21] Microsoft, "Windows Presentation Foundation Documentation", <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/>, consulté le 15 mai 2025.
- [22] Microsoft, "XAML Overview", <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/xaml/>, consulté le 15 mai 2025.
- [23] Newtonsoft, "Json.NET Documentation", <https://www.newtonsoft.com/json/help/>, consulté le 15 mai 2025.
- [24] PDFium, "PDFium Viewer", <https://github.com/pvginkel/PdfiumViewer>, consulté le 15 mai 2025.
- [25] Internet Engineering Task Force, "JSON Web Token (JWT)", <https://datatracker.ietf.org/doc/html/rfc7519>, consulté le 15 mai 2025.
- [26] Scrum.org, "What is Scrum ?", <https://www.scrum.org/resources/what-is-scrum>, consulté le 15 mai 2025.
- [27] W3C, "Web Content Accessibility Guidelines (WCAG)", <https://www.w3.org/WAI/standards-guidelines/wcag/>, consulté le 15 mai 2025.
- [28] MDN Web Docs, "Responsive Design", https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Responsive_Design, consulté le 15 mai 2025.
- [29] ECMA International, "The JSON Data Interchange Syntax", <https://www.ecma-international.org/publications-and-standards/standards/ecma-404/>, consulté le 14 mai 2025.
- [30] Internet Engineering Task Force, "HTTP Semantics", <https://datatracker.ietf.org/doc/html/rfc9110>, consulté le 16 mai 2025.