



REPUBLIQUE DU CAMEROUN
Paix – Travail – Patrie

UNIVERSITE DE DOUALA

ECOLE NATIONALE SUPERIEURE
POLYTECHNIQUE DE DOUALA

B.P. 2701 Douala
Tél. (237) 697 542 240
Site web : www.enspd-udo.cm

REPUBLIC OF CAMEROON
Peace – Work – Fatherland

THE UNIVERSITY OF DOUALA

NATIONAL HIGHER POLYTECHNIC
SCHOOL OF DOUALA

P.O. Box :2701 Douala
Phone : (237) 697 542 240
Email : contact@enspd-udo.cm



PROJET DE POO

THEME : APPLICATION DE GESTION D'UNE BIBLIOTHEQUE

Rédigé par :

NOMS & PRENOMS	MATRICULE
FOHOM GAINGNE MERVEILLES DIANE	22G00136
FOUEJIO NGUEFACK NILIMA JODIE	24G01092
GUEMBU SIME GUY VALAIRE	22G00460
GUEMOGNE GERALDINE ESTHER	22G00153
MEKOUNDE ELAME GUSTAVE CYRIAC	22G00253
MOHAMMED EL BACHIR ISMAÏLA	22G00262
NKAMOLOUM TENE DAMIEN ANTHONY	22G00328
NONGNING LELE STEVE JORDAN	24G01122
NTAKEU BAPIAH LEANE YVANNA	22G00339
TIEMOUO DJIOLIO DORCAS	24G01134

Sous la supervision de :

Dr NOULAPEU

Année académique 2024/2025

SOMMAIRE

INTRODUCTION.....	3
CHAPITRE I : DEVELOPPEMENT DU PROJET	4
I.1 Choix des technologies	4
I.2 Répartition Des Taches	4
I.2.1 Développement Backend.....	4
I.2.2 Développement frontend	6
I.3 Tests fonctionnels	7
I.4 Intégration.....	7
CHAPITRE 2 : DIFFICULTÉS RENCONTRÉES, SOLUTIONS ADOPTÉES	9
CONCLUSION	10

INTRODUCTION

Dans le cadre de notre formation en Programmation Orienté-Objet, nous avons été amenés à concevoir et développer une application de gestion d'une bibliothèque. Ce projet s'inscrit dans une démarche pédagogique visant à renforcer nos compétences techniques et notre capacité à travailler en équipe sur un projet informatique structuré.

L'objectif principal était de mettre en place un système permettant de gérer efficacement les livres, les lecteurs, les emprunts, ainsi que les comptes des responsables. Pour ce faire, nous avons adopté une architecture basée sur le **modèle MVC (Modèle-Vue-Contrôleur)**, facilitant la séparation des responsabilités entre la logique métier, l'interface utilisateur et l'accès aux données.

Afin d'assurer une organisation claire et une progression harmonieuse, une **répartition rigoureuse des tâches** a été effectuée au sein de l'équipe, distinguant les volets **backend** et **frontend**. Chaque membre s'est vu confier un ensemble de modules spécifiques, selon ses compétences, avec pour objectif une intégration cohérente des différentes composantes.

Le choix technologique s'est porté sur **Java EE**, notamment l'utilisation de **Servlets**, **JSP**, **JDBC** et **MySQL**, pour répondre aux contraintes techniques et aux exigences de gestion en ligne multi-utilisateur. Le développement s'est appuyé sur l'environnement **Eclipse avec Tomcat** pour le déploiement et les tests locaux.

Ce rapport présente dans un premier temps le processus de développement du projet, en détaillant les technologies choisies, la répartition des tâches et les étapes d'intégration. Par la suite, nous reviendrons sur les **difficultés rencontrées**, les **solutions mises en place**, ainsi que les **enseignements tirés** de cette expérience formatrice, aussi bien sur le plan technique qu'organisationnel.

CHAPITRE I : DEVELOPPEMENT DU PROJET

Le développement du projet repose sur une planification rigoureuse et une répartition structurée des tâches entre les membres de l'équipe. Chaque membre de l'équipe s'est vu attribuer des responsabilités précises, que ce soit côté **backend** ou **frontend**, en suivant une architecture basée sur le **modèle MVC (Modèle-Vue-Contrôleur)**. Cette organisation avait pour objectif de favoriser une collaboration efficace, une intégration fluide des différentes composantes de l'application, et une progression harmonieuse du projet, en réponse aux besoins fonctionnels et techniques identifiés lors des phases précédentes.

I.1 Choix des technologies

Au départ, l'équipe avait envisagé d'utiliser **JavaFX** pour le développement de l'interface graphique. Toutefois, en raison de certaines contraintes techniques (intégration multi-utilisateur, gestion en ligne, etc.), le choix s'est finalement porté sur Java EE avec les technologies suivantes :

- **Servlets** pour la logique métier (Contrôleur)
- **JSP** (Java Server Pages) pour la couche vue
- **JDBC** pour la communication avec la base de données
- **MySQL** comme système de gestion de base de données

I.2 Répartition Des Taches

Les tâches ont été réparties de la manière suivante :

- Création des **classes Java** représentant les entités de la base
- Développement des **DAO** pour chaque entité (CRUD via JDBC)
- Implémentation des **Servlets**, chacune responsable d'un module (connexion, gestion des livres, lecteurs, emprunts, etc.)
- Création des vues **JSP** et intégration dans le projet

I.2.1 Développement Backend

Noms des developpeurs	Taches
GUEMOGNE GERALDINE ESTHER	Authentification <ul style="list-style-type: none"> Finaliser AuthServicelet.java Gérer connexion, création de session, redirection vers tableau de bord Messages d'erreur sur échec d'authentification
FOHOM GAINGNE MERVEILLES DIANE	Tableau de bord <ul style="list-style-type: none"> Développer DashboardServlet.java Appeler les DAO nécessaires pour afficher : Nombre d'emprunts en cours, Nombre d'emprunts clôturés, Derniers emprunts Stocker les données dans <code>request.setAttribute</code>
MOHAMMED EL BACHIR ISMAÏLA	Emprunt <ul style="list-style-type: none"> Implémenter EmpruntServlet.java Gestion de la consultation, ajout, modification, suppression Intégration avec EmpruntDAO, LecteurDAO, LivreDAO
GUEMBU SIME GUY VALAIRE	Livres <ul style="list-style-type: none"> Implémenter LivreServlet.java Consultation, ajout, modification, suppression Appeler LivreDAO, gérer liste des livres Appeler aussi ExempleDAO pour la gestion des exemplaires
MEKOUNDE ELAME GUSTAVE CYRIAC	Lecteurs <ul style="list-style-type: none"> Développer LecteurServlet.java CRUD des lecteurs (utiliser LecteurDAO) Vérifier les entrées utilisateur (nom, email, etc.)

NKAMOLOUM TENE DAMIEN ANTHONY	Gestion des exemplaires <ul style="list-style-type: none"> • Implémenter dans LivreServlet.java (ou Servlet dédiée si besoin) • Ajout / suppression / mise à jour d'un exemplaire d'un livre • Gérer les relations livre ↔ exemplaires avec ExemplaireDAO
FOUEJIO NGUEFACK NILIMA JODIE	Compte du responsable <ul style="list-style-type: none"> • Finaliser CompteServlet.java • Modifier les informations personnelles • Mettre à jour les infos en base avec ResponsableDAO • Mettre à jour les infos de session

I.2.2 Développement frontend

DEVELOPPEURS	TACHES
NONGNING LELE STEVE JORDAN (Chef)	Pages Authentification & Tableau de bord <ul style="list-style-type: none"> • index.jsp : page de login avec affichage conditionnel d'erreurs • dashboard.jsp : affichage des statistiques + navigation
NTAKEU BAPIAH LEANE YVANNA	Emprunts & Lecteurs <ul style="list-style-type: none"> • ListeEmprunts.jsp : tableau des emprunts, bouton modifier/supprimer • NewEmprunt.jsp, VoirEmprunt.jsp: formulaire de création/modification • ListeLecteurs.jsp + NewLecteur.jsp + VoirLecteur.jsp
TIEMOUO DJIOLIO DORCAS	Livres & Exemplaires + Mon compte

	<ul style="list-style-type: none"> • Listeivres.jsp, NewLivre.jsp, VoirLivre.jsp : gestion des livres • ListeExemplaires.jsp, NewExemplaire.jsp, VoirExemplaire.jsp : liste et édition des exemplaires d'un livre • Compte.jsp : formulaire de modification des infos personnelles
--	---

I.3 Tests fonctionnels

Afin de garantir la fiabilité et la conformité des différentes fonctionnalités développées, des **tests fonctionnels** ont été effectués sur chaque module de manière indépendante. Ces tests ont porté notamment sur :

- **L'authentification des utilisateurs** : vérification de la validité des identifiants, gestion des cas d'échec (mauvais mot de passe ou identifiant inexistant), redirection vers les vues appropriées, et affichage des messages d'erreur correspondants.
- **Les opérations de base de données** : insertion, modification et suppression des données ont été systématiquement testées pour s'assurer de leur bon fonctionnement et de la persistance correcte des informations.
- **L'affichage conditionnel dans les interfaces JSP** : chaque vue a été testée pour garantir l'apparition des éléments en fonction des rôles ou des états de l'application.

L'ensemble des tests a été effectué **localement sur le serveur Apache Tomcat** intégré à l'environnement **Eclipse**, avec une attention particulière portée à la **vérification des journaux de la console (logs)** pour détecter d'éventuelles anomalies ou exceptions.

I.4 Intégration

Après la validation fonctionnelle de chaque composant, l'équipe a procédé à une **intégration progressive** du système, en veillant à assurer la cohérence et l'interopérabilité entre les différentes couches de l'application.

Les principales étapes de cette phase d'intégration ont été :

- **Connexion entre le backend (Java, Servlets, JDBC) et le frontend (JSP, HTML, CSS)**, afin d'établir une communication fluide et dynamique entre la logique métier et les interfaces utilisateur.
- **Vérification de la navigation et de la cohérence des vues**, en s'assurant que l'utilisateur puisse parcourir l'ensemble des fonctionnalités de manière intuitive, sans rupture ni redondance.
- **Tests d'intégration globaux** sur le parcours utilisateur complet, de l'authentification à l'exécution des différentes actions (consultation, gestion des entités, déconnexion, etc.), afin de garantir une expérience fluide et exempte de bogues.

Cette démarche méthodique a permis de livrer une version stable de l'application, fonctionnelle aussi bien sur le plan technique qu'ergonomique.

CHAPITRE 2 : DIFFICULTÉS RENCONTRÉES, SOLUTIONS ADOPTÉES

La réalisation de ce projet, bien qu'enrichissante, n'a pas été exempte de défis, tant sur le plan technique qu'organisationnel. Toutefois, grâce à la cohésion du groupe et à une volonté collective de surmonter les obstacles, des solutions concrètes ont été mises en place pour garantir l'avancement du projet. Les principales difficultés rencontrées et les stratégies adoptées sont les suivantes :

- **Non-maîtrise des technologies utilisées** : Certains membres de l'équipe ne maîtrisaient pas suffisamment les technologies choisies, telles que les Servlets, JSP ou JDBC. Pour combler ces lacunes, nous avons mis en place des **séances de formation ciblées**, animées par les membres plus expérimentés. Des **sessions de partage de connaissances** ont également été organisées de manière régulière, favorisant l'apprentissage collaboratif et l'harmonisation des compétences au sein de l'équipe.
- **Pannes matérielles et problèmes techniques** : Des pannes d'ordinateurs ont affecté la productivité de certains membres. Face à cette contrainte matérielle, nous avons privilégié le **travail en binôme**, permettant aux membres concernés de poursuivre leurs tâches à l'aide des équipements des autres.
- **Difficultés de coordination et de collaboration en équipe** : Le travail en groupe a parfois été perturbé par un manque de communication ou une mauvaise coordination des tâches. Pour y remédier, nous avons instauré des **réunions hebdomadaires** visant à faire le point sur l'état d'avancement, identifier les blocages et ajuster la répartition des responsabilités. Ces rencontres ont favorisé une meilleure compréhension des rôles de chacun et ont renforcé la dynamique d'équipe.
- **Complexité de l'installation des outils et de la configuration de l'environnement** : L'installation des outils de développement (serveur Tomcat, configuration JDBC, etc.) a posé des problèmes à certains membres. Afin de faciliter cette étape, nous avons élaboré un **guide d'installation clair et détaillé**. Cette initiative a permis de gagner du temps et d'éviter les erreurs de configuration.

CONCLUSION

Ce projet de gestion d'une bibliothèque a constitué une étape significative dans notre parcours d'apprentissage en tant qu'élèves ingénieurs. En adoptant une approche structurée basée sur le modèle MVC et une répartition claire des responsabilités entre les membres de l'équipe, nous avons pu développer de manière collaborative et efficace les différentes composantes de l'application, tant côté backend que frontend.

Malgré les défis techniques (tels que la maîtrise des servlets, l'intégration des DAO, ou encore l'utilisation de JSTL) et organisationnels (comme les problèmes de communication ou les pannes de matériel), notre équipe a su s'adapter et progresser. La mise en place de règles de coordination, l'utilisation d'outils de gestion de projet, ainsi que des séances de formation ciblées ont permis de surmonter ces obstacles.

Ce projet nous a également permis de consolider des compétences clés : travail en équipe, gestion de projet, développement web structuré, et intégration d'un système complet dans un contexte réaliste. En se tournant vers l'avenir, les perspectives d'amélioration résident dans l'automatisation des tests, l'optimisation de l'interface utilisateur, et l'intégration de nouvelles fonctionnalités telles qu'un moteur de recherche avancé ou des statistiques graphiques.

Ainsi, cette expérience marque non seulement une avancée technique, mais également une maturation dans notre capacité à mener un projet informatique collaboratif à bien, nous préparant à relever de futurs défis professionnels avec plus de rigueur et de confiance.