



المعهد الوطني للبريد والمواصلات
المعهد الوطني للبريد والمواصلات
Institut National des Postes et Télécommunications



الوكالة الوطنية لتنظيم المواصلات
Agence Nationale de Réglementation des Télécommunications

RAPPORT

SYSTEME DE GESTION D'UNE

BIBLIOTHEQUE

eLibrary



Java™



eclipse

Réalisé par :

HAJJY Ismail

Elkihel Oussama

GHRIS Adam

Encadré par :

Mr El HAMLAOUI

Mahmoud

REMERCIEMENTS

Je tiens à exprimer ma profonde gratitude envers toutes les personnes qui ont contribué à la réalisation de ce projet académique. Leur soutien et leur engagement ont été essentiels à la réussite de ce travail.

En premier lieu, je tiens à remercier notre professeur Mr **EL HAMLAOUI Mahmoud** pour sa guidance précieuse et ses conseils éclairés tout au long de ce projet. Sa disponibilité, sa patience et son expertise ont grandement enrichi mon expérience et ont été des facteurs déterminants dans la qualité de ce travail.

Je tiens à exprimer ma gratitude envers l'ensemble du corps professoral et du personnel de **INPT** pour leur contribution à mon parcours académique. Leur dévouement à l'excellence a été une source d'inspiration constante.

Merci à tous ceux qui ont participé de près ou de loin à la réalisation de ce projet. Votre soutien a été inestimable et a contribué de manière significative à son aboutissement.

Sommaire

INTRODUCTION GENERAL.....	
Chapitre I : Cahier de charges	
1. Objectif du Projet	
2. Fonctionnalités Principales.....	
a. Gestion des Utilisateurs	
b. Catalogue des Livres	
c. Gestion des Emprunts	
d. Tableau de Bord Administratif	
e. Interface Utilisateur Intuitive	
3. Spécification des besoins	
CHAPITRE II : Conception et modélisation	
1. Introduction.....	
2. Définition STAR UML.....	
3. Diagramme de cas d'utilisation	
3.1 Acteur	
3.2 Cas d'utilisation	
4. Diagramme de cas d'utilisation pour Les utilisateurs	
5. Diagramme de cas d'utilisation pour l'Administrateur	
6. Diagramme de cas d'utilisation final.....	
7. Diagramme de classe	
8. Conception Graphique	
9. Conclusion	
CHAPITRE III : Intégration et test du système	
1. Calendrier de développement.....	
2. Détails de la mise en œuvre	
a. Aperçu du système.....	
b. JAVA	
c. Eclipse.....	
d. Base de données	
e. JDBC.....	
f. Tableaux de bases de données	

3.	Résultat du Mise en œuvre	
a.	Espace administration	
b.	Espace Utilisateur	
CHAPITRE IV : Résultats et discussion.....		
1.	Résultats des tests	
1.1	Tests unitaires	
1.2	GitHub Link	

Liste des Figure

Figure 1 : LOGO du StarUML	
Figure 2: Acteur	
Figure 3 : Cas d'utilisation	
Figure 4 : Use-Case du Client	
Figure 5 : Use-case d'administrateur	
Figure 6 : Use-case général	
Figure 7 : Diagramme des classes.....	
Figure 8 : Diagramme des interfaces.....	
Figure 9 : Calendrier de développent	
Figure 10: LOGO du JAVA.....	
Figure 11 : LOGO du Eclipse	
Figure 12 : Hello world avec Eclipse	
Figure 13 : Eclipse Description de la fenêtre.....	
Figure 14 : Code de connexion au MySQL	
Figure 15 : Fenêtre principale	
Figure 16 : Fenêtre principale d'administration.....	
Figure 17 : Fenêtre de gestion des livres	
Figure 18 : base de données des livres	
Figure 19 : Fenêtre de gestion des utilisateurs	
Figure 20 : base de données des admins.....	
Figure 21 : Fenêtre principale du client.....	
Figure 22 : Genre : Horror	
Figure 23 : Genre : Action	
Figure 24 : Favorites books	
Figure 25 : Base de données des livres.....	

INTRODUCTION GENERAL

La gestion moderne des bibliothèques requiert une approche technologique afin d'optimiser les opérations, d'améliorer l'accessibilité aux ressources et de fournir une expérience utilisateur transparente. Ce rapport présente en détail le processus de conception, de développement et de mise en œuvre d'une application de système de gestion de bibliothèque, élaborée en utilisant la plateforme Java Enterprise Edition (JEE).

Objectif :

L'application web eLibrary Java a été créée comme une plateforme numérique visant à proposer aux utilisateurs un vaste répertoire en ligne de livres numériques. Dotée d'une interface conviviale, elle permet aux utilisateurs de rechercher, d'accéder et de gérer aisément une collection de livres, de consulter en détail chaque ouvrage, et éventuellement de lire directement son contenu depuis la plateforme. Les utilisateurs peuvent s'enregistrer, se connecter de manière sécurisée, et télécharger gratuitement des livres tout en ayant la possibilité d'évaluer et de commenter ces ouvrages. Les administrateurs bénéficient de fonctionnalités de gestion de la bibliothèque, incluant l'ajout et la suppression de livres ainsi que la gestion des comptes utilisateurs. L'application assure la sécurité des données grâce au chiffrement, à l'authentification sécurisée et à la gestion des sessions.

Structure :

Ce rapport est structuré pour documenter de manière exhaustive les différentes phases du développement, depuis la phase de conception architecturale jusqu'à la réalisation pratique de l'application. Chaque section abordera les choix de conception, les défis rencontrés, les solutions apportées, et fournira des insights sur la manière dont ces choix ont contribué à atteindre les objectifs du projet.

En conclusion, cette application de gestion de bibliothèque développée en Java EE représente une étape significative vers la modernisation des services de bibliothèque, en intégrant les meilleures pratiques de développement logiciel avec les exigences spécifiques du domaine de la gestion documentaire.

Chapitre I : cahier de charges

Objectif du Projet :

Le projet vise à concevoir, développer et déployer une application de système de gestion de bibliothèque permettant une administration efficace des ressources, des prêts et des retours, tout en offrant une expérience utilisateur conviviale.

Fonctionnalités Principales :

1. Gestion des Utilisateurs :

- Enregistrement des utilisateurs (étudiants, enseignants, administrateurs).
- Authentification sécurisée.
- Profils d'utilisateurs avec des autorisations spécifiques.

2. Catalogue des Livres :

- Ajout, modification et suppression de livres.
- Catégorisation des livres (par genre, auteur, etc.).
- Recherche avancée dans le catalogue.

3. Gestion des Emprunts :

- Enregistrement des emprunts et des retours.
- Suivi des dates d'échéance.
- Notifications automatiques pour les retards.

4. Tableau de Bord Administratif :

- Statistiques sur l'utilisation de la bibliothèque.
- Rapports sur les livres les plus empruntés, les utilisateurs les plus actifs, etc.
- Gestion des utilisateurs et des privilèges.

5. Interface Utilisateur Intuitive :

- Interface conviviale pour la recherche de livres.
- Processus de prêt et de retour simplifié.
- Notifications pour les utilisateurs.

Spécification des besoins

Cette phase consiste à comprendre le contexte du système. Il s'agit de déterminer les fonctionnalités et les acteurs les plus pertinents, de préciser les risques les plus critiques et d'identifier les cas d'utilisation initiaux.

- Le bibliothécaire est un employé de la bibliothèque. Le système à concevoir doit assister le bibliothécaire dans sa tâche.
- Le bibliothécaire communique avec les emprunteurs.
- Le bibliothécaire prête des livres à des emprunteurs
- Il doit gérer l'achat de nouveaux titres.
- Les titres les plus demandés sont achetés en plusieurs exemplaires.
- Un emprunteur doit s'enregistrer pour être adhérent avant d'emprunter les livres .
- Un visiteur peut consulter un livre sur place.
- Un emprunteur peut réserver un livre qui est indisponible (déjà prêté ou Répertorié mais non encore acheté).
- Un emprunteur ne peut emprunter que pas plus de trois livres.
- Lorsqu'un livre devient disponible (rendu ou acheté), un emprunteur qui l'avait réservé doit être averti.
- La réservation est annulée quand le livre est emprunté.
- Une réservation peut être annulée à tout moment.
- La mise à jour (ajout, suppression et modification) des informations relatives aux exemplaires, emprunteurs et fond bibliothèque doit être aisée.
- De même que la mise à jour des informations relatives aux prêts et réservations.

CHAPITRE II : Conception et modélisation

1. Introduction :

Dans ce chapitre on va étudier la modélisation de cahier de charges, on utilise des logiciels pour la modélisation afin de construire un diagramme de cas d'utilisation ainsi qu'un diagramme de classe.

Dans notre travail nous avons utilisé le logiciel STAR UML pour faire la modélisation, Alors c'est quoi ce logiciel et à quoi sert ?

2. Définition STAR UML :

StarUML est un logiciel de modélisation UML (Unified Modeling Language) open source qui peut remplacer dans bien des situations des logiciels commerciaux et coûteux. Étant simple d'utilisation, nécessitant peu de ressources système, supportant UML 2, ce logiciel constitue une excellente option pour une familiarisation à la modélisation. Cependant, seule une version Windows est disponible.



Figure 1 : LOGO du StarUML

3. Diagramme de cas d'utilisation :

3.1 Acteur

Un acteur est l'idéalisation d'un rôle joué par une personne externe, un processus ou une chose qui interagit avec un système. Il se représente par un petit bonhomme avec son nom (son rôle) inscrit dessous.

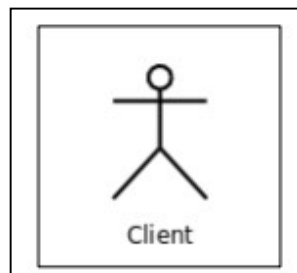


Figure 2: Acteur

3.2 Cas d'utilisation

Un cas d'utilisation est un service rendu à un acteur : c'est une fonctionnalité de son point de vue. Un cas d'utilisation se représente par une ellipse contenant le nom du cas (un verbe à l'infinitif), et optionnellement, au-dessus du nom, un stéréotype.



Figure 3 : Cas d'utilisation

4. Diagramme de cas d'utilisation pour Les Utilisateurs :

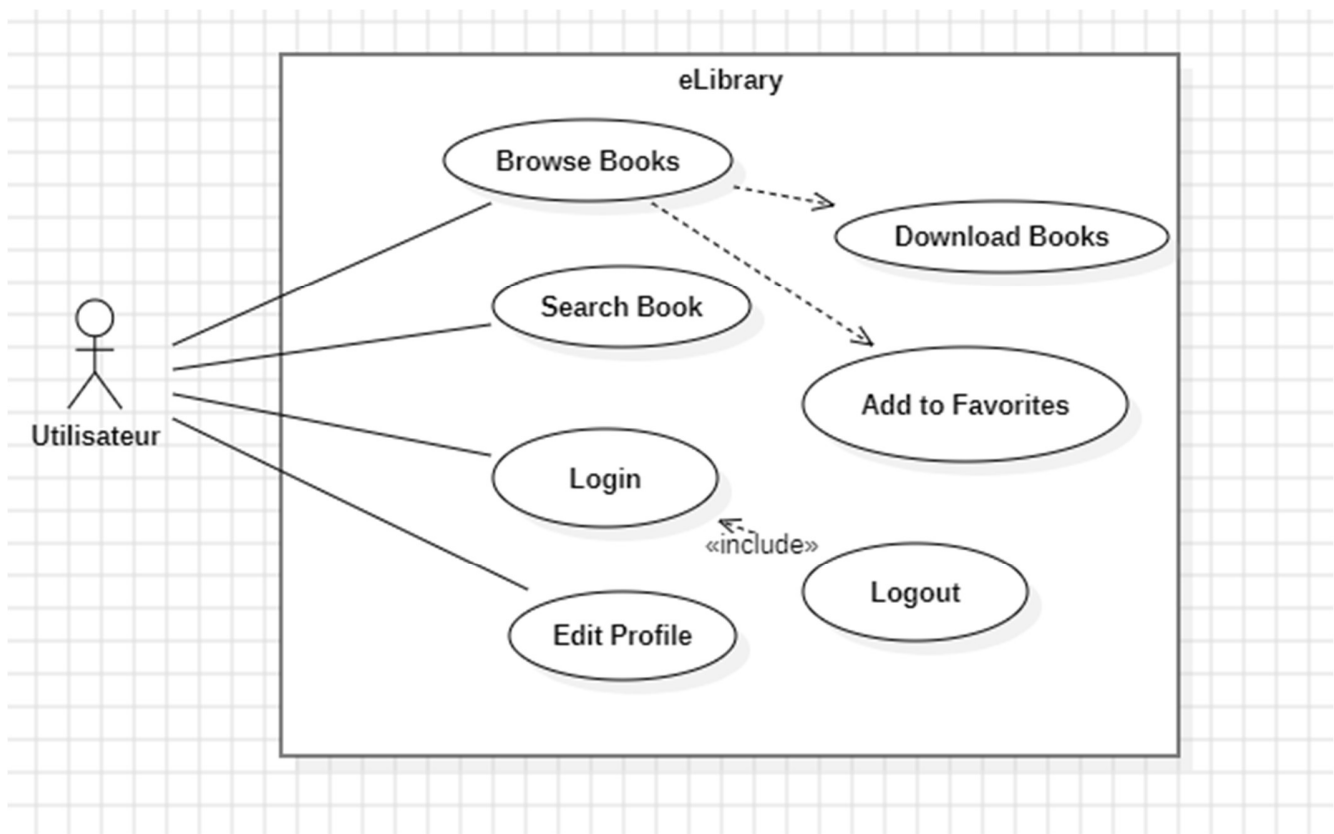


Figure 4 : Use-Case du Client

5. Diagramme de cas d'utilisation pour Admin :

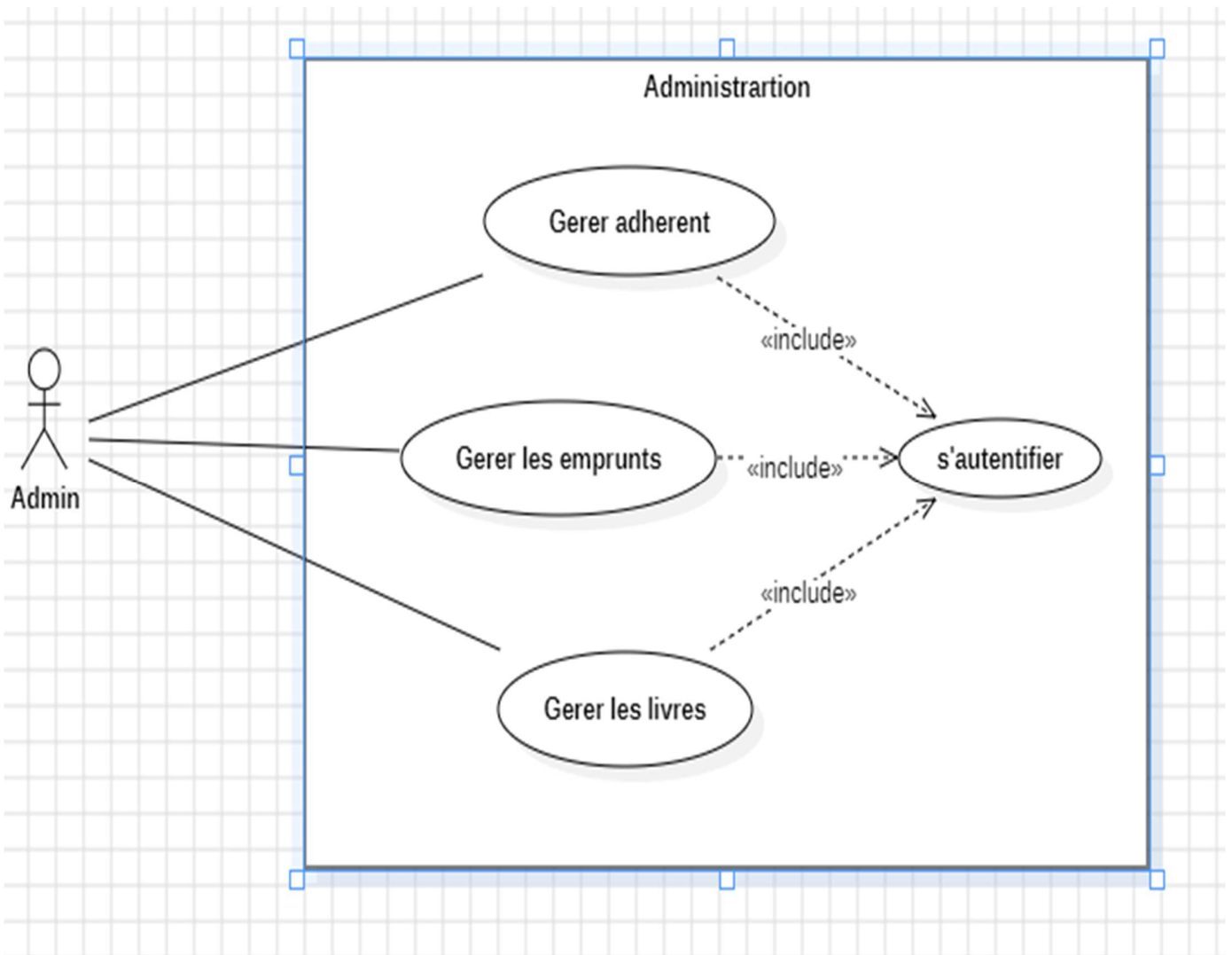


Figure 5 : Use-case administrateur

6. Diagramme de cas d'utilisation final :

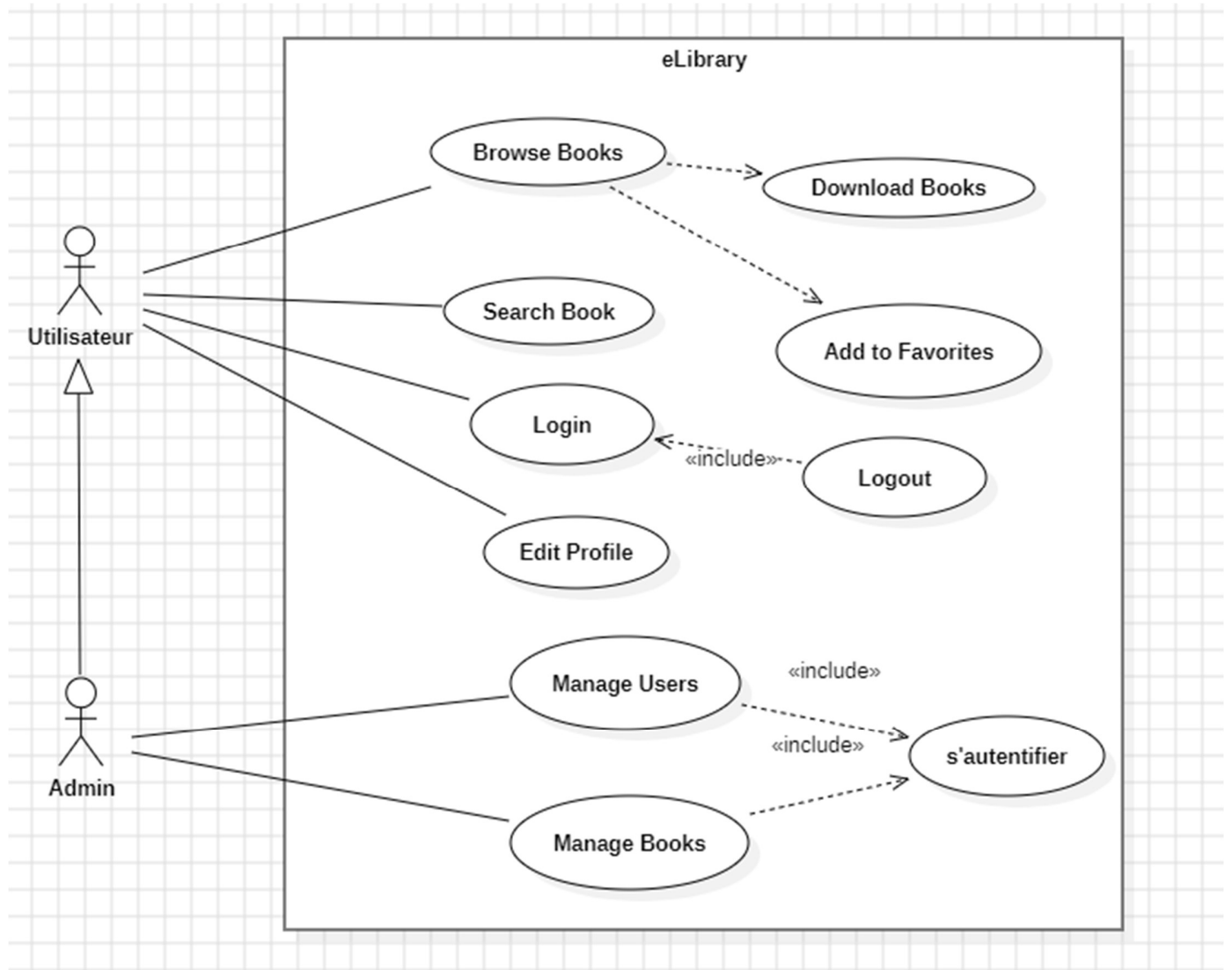


Figure 6 : Use-case général

6. Diagramme de classe :

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que leurs relations. Ce diagramme fait partie de la partie statique d'UML, ne s'intéressant pas aux aspects temporels et dynamiques.

Une classe décrit les responsabilités, le comportement et le type d'un ensemble d'objets. Les éléments de cet ensemble sont les instances de la classe.

Une classe est un ensemble de fonctions et de données (attributs) qui sont liées ensemble par un champ sémantique. Les classes sont utilisées dans la programmation orientée objet. Elles permettent de modéliser un programme et ainsi de découper une tâche complexe en plusieurs petits travaux simples.

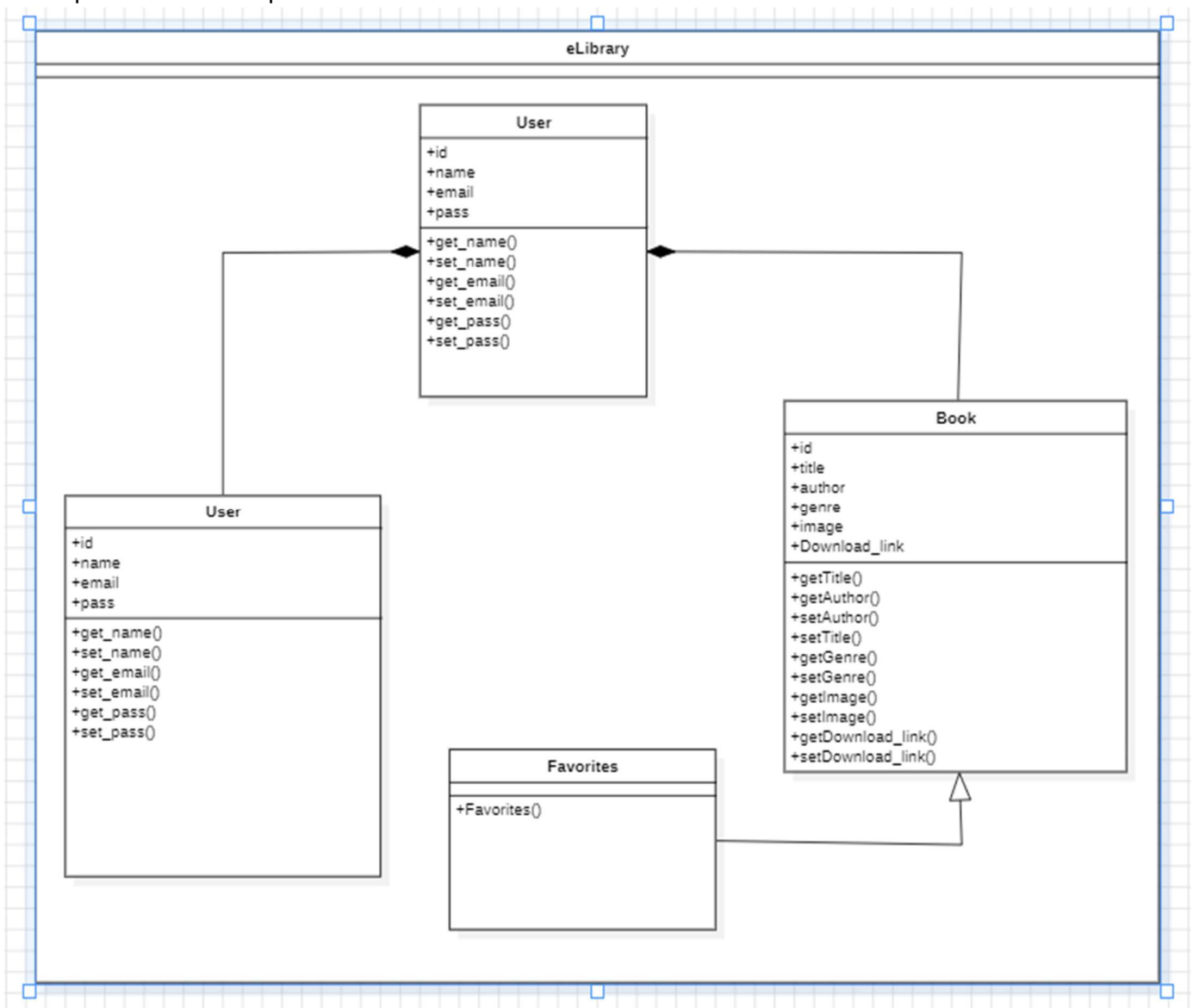


Figure 7 : Diagramme des classes

7. Conclusion :

Dans ce chapitre, nous avons conçu et documenté le code que nous devons produire. Dans cette phase, toutes les questions concernant la manière de réaliser le système à développer ont été élucidées. Le produit obtenu est un modèle graphique (ensemble de diagrammes) prêt à être codé. Dans le chapitre suivant nous allons étudier en détails les outils et les langages utilisés durant la phase de construction.

CHAPITRE III : Intégration et test du système

1. Calendrier de développement

Le travail sur Le SYSTÈME DE gestion d'une bibliothèque eLibrary a commencé le 6 Novembre 2023 et devrait être terminé le 26 Novembre 2023. Le diagramme suivant estime la durée des différentes phases du travail de développement :

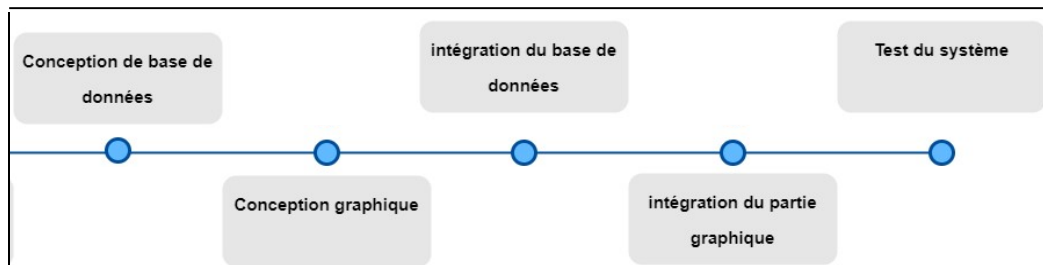


Figure 8 : Calendrier de développement

2. Détails de la mise en œuvre

a. Aperçu du système

Le système eLibrary est une application Desktop. Elle a été développée en utilisant JAVA. Une base de données MySQL composée de plusieurs tables est utilisée pour le stockage des données.

b. JAVA

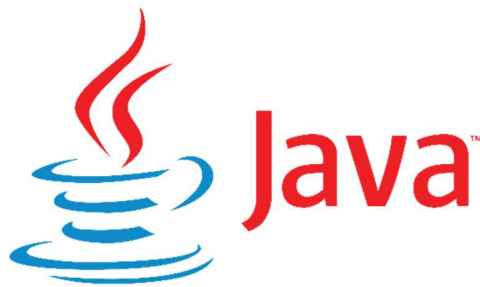


Figure 10: LOGO du JAVA

JAVA est un langage de programmation et une plate-forme informatique qui ont été créés par Sun Microsystems en 1995. Beaucoup d'applications et de sites Web ne fonctionnent pas si Java n'est pas installé et leur nombre ne cesse de croître chaque jour. Java est rapide, sécurisé et fiable.

Java est un langage de programmation inspiré du langage C++, avec un modèle de programmation orienté objet. Java permet de créer des applications complètes. Il peut également servir à créer un petit module d'application, dit applet, à intégrer dans une page Web.

Les principales caractéristiques de Java sont les suivantes :

- Les programmes créés sont portables. Le programme source est compilé dans un « code », qui peut être exécuté sur un serveur ou un client doté d'une machine virtuelle Java.
- Le code est robuste.
- Java est orienté objet.

c. Eclipse



Figure 10 : LOGO du Eclipse

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. Eclipse IDE est principalement écrit en Java (à l'aide de la bibliothèque graphique SWT, d'IBM), et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions.

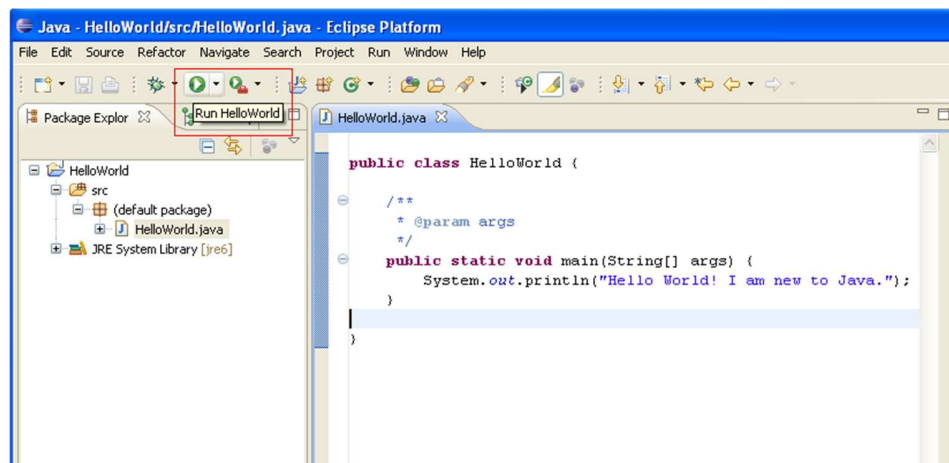


Figure 11 : Hello world avec Eclipse

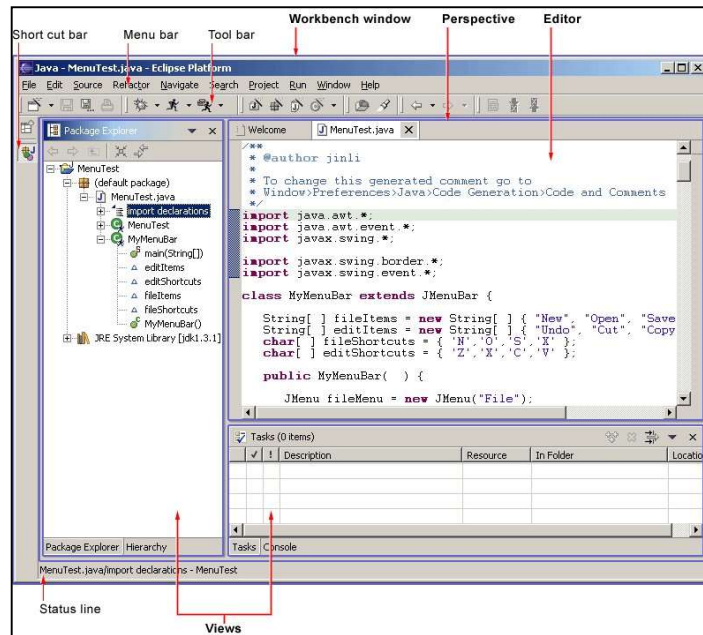


Figure 12 : Eclipse Description de la fenêtre

d. Base de données

La base de données (BDD, ou DB en anglais) est un système qui enregistre des informations, mais pas n'importe comment : ces informations sont toujours classées. Et c'est ça qui fait que laBDD est si pratique : c'est un moyen extrêmement simple de ranger des informations ! Grossièrement, une BDD peut être vue comme un ensemble de tableaux, des structures contenant donc des lignes et des colonnes et dans lesquelles nos données sont rangées.

Il existe un vocabulaire spécifique pour désigner les différents éléments composant une BDD :

- La base désigne le volume englobant l'ensemble, la boîte qui contient tous les tableaux ;
- Une table désigne un tableau de données, elle contient des lignes et des colonnes ;
- Une entrée désigne une ligne ;
- Un champ désigne une colonne.

Il n'existe pas qu'une seule solution pour créer et gérer des bases de données. Voici une liste des principaux acteurs de ce marché :

- MySQL : solution libre et gratuite, c'est le SGBD le plus répandu. C'est d'ailleurs celui que nous allons utiliser dans ce cours !
- PostgreSQL : solution libre et gratuite, moins connue du grand public mais proposant des fonctionnalités inexistantes dans MySQL ;
- Oracle : solution propriétaire et payante, massivement utilisée par les grandes entreprises. C'est un des SGBD les plus complets, mais un des plus chers également ;
- SQL Server : la solution propriétaire de Microsoft.
- DB2 : la solution propriétaire d'IBM, utilisée principalement dans les très grandes entreprises sur des *Mainframes*.

e. JDBC

C'est une API qui fait partie intégrante de la plate-forme Java, et qui est constituée de classes permettant l'accès depuis vos applications Java à des données rangées sous forme de tables. Dans la très grande majorité des cas, il s'agira bien entendu de bases de données stockées dans un SGBD ! Les actions rendues possibles par cette API sont :

- La connexion avec le SGBD ;
- L'envoi de requêtes SQL au SGBD depuis une application Java ;
- Le traitement des données et éventuelles erreurs retournées par le SGBD lors des différentes étapes du dialogue (connexion, requête, exécution, etc.).

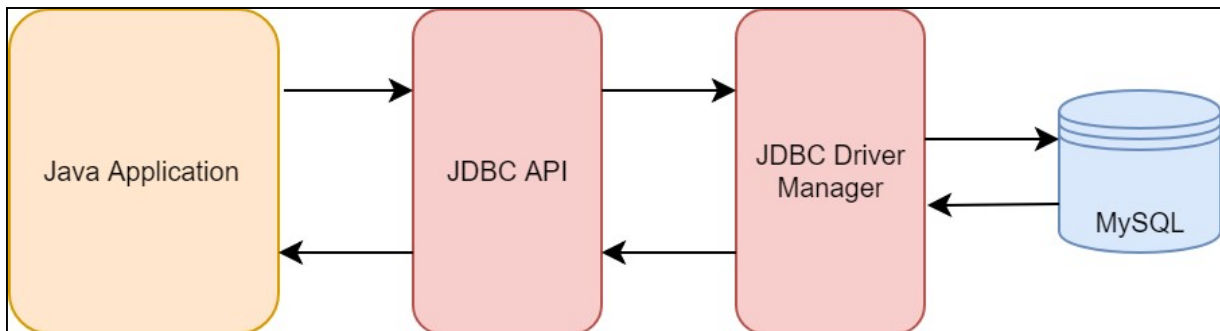


Figure 13 : utilisation de l'API JDBC pour accéder aux MySQL

En utilisant le code suivant, nous pouvons nous connecter avec succès à la base de données MySQL

```
package e_library_connection;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DbCon {

    private static Connection connection = null;

    public static Connection getConnection() throws ClassNotFoundException, SQLException {
        if(connection == null) {
            Class.forName("com.mysql.cj.jdbc.Driver");
            connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/employee_service", "root", "Breath130@");
            System.out.println("connected");
        }
        return connection;
    }
}
```

Figure 14 : Code de connexion au MySQL

f. Tableaux de bases de données

Nom de la table	Description
books	Enregistrer les détails de chaque livre
Favorites books	Enregistrer les détails de chaque favorite book
Users	Enregistrer les détails de chaque user

3. Résultat du Mise en œuvre

Après le démarrage de l'application, la Fenêtre principale s'affiche, sur laquelle l'utilisateur peut cliquer pour accéder à l'espace client ou à l'espace d'administration.

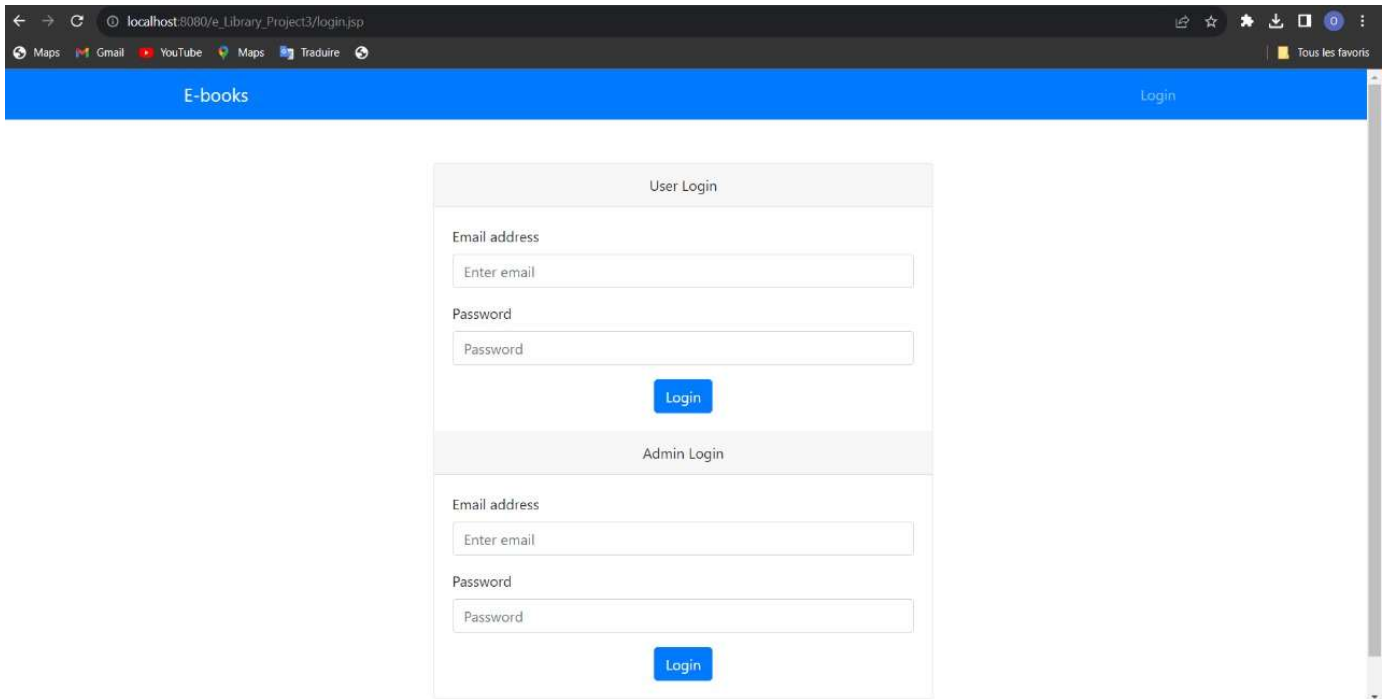


Figure 15 : Fenêtre principale

Si l'utilisateur se connecter avec un compte d'admin, Une fenêtre d'identification sera affichée. Chaque employée a son propre compte, Ici on va essayer d'accès come un employé appeler anass avec un admin email «benlakhali47@email.com » et un Mot de passe « 123456@ ».

a. Espace administration

Dans la fenêtre d'administration principale on peut modifier certaines options liées aux la compagnie telles que :

- Gestion des utilisateurs :
 - Suppression des utilisateurs.
- Gestion des livres :
 - Ajoute des livres.
 - Suppression des livres.

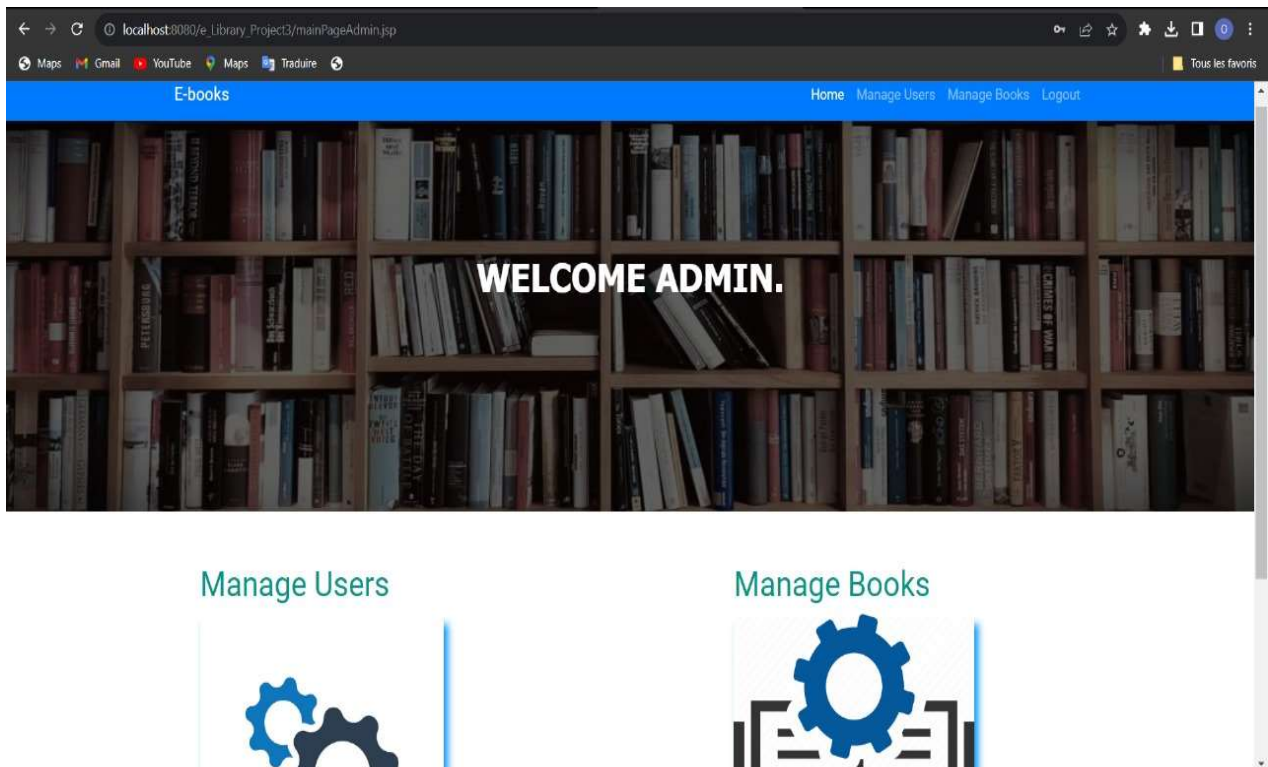


Figure 16 : Fenêtre principale d'administration

The screenshot shows a web browser at localhost:8080/e_Library_Project3/ManageBooks.jsp. The page has a blue header with 'E-books' on the left and navigation links 'Home', 'Manage Users', 'Manage Books', and 'Logout' on the right. The main content area is titled 'Manage Books' and contains a table with the following data:

Title	Author	Genre	Delete
Action Front	Boyd Cable	action	Remove
Adventures of Huckleberry Finn	Mark Twain	adventure	Remove
The Horror in the Museum	H. P. Lovecraft	horror	Remove
A Princess Of Mars	Edgar Rice Burroughs	action	Remove
Living History	Ben Essex	history	Remove

Figure 17 : Fenêtre de gestion des livres

Toutes les données saisies dans cette fenêtre sont directement enregistrées dans la base de données

	id	title	author	genre	image	Download_link
▶	1	Action Front	Boyd Cable	action	Action_Front.jpg	https://drive.google.com/file/d/1nsPyyrTrRhOf86m3VhLdNUPiICJDL9ip/
	2	Adventures of Huc...	Mark Twain	adventure	Adventures_of_Huckleberry_...	https://drive.google.com/file/d/1uU3h5xJ5VWP52PGpaSP8G_kXPART4cb/
	3	The Horror in the ...	H. P. Lovecraft	horror	The_Horror_in_the_Museum.jpg	https://drive.google.com/file/d/1mxPNtraKCIRik79iMGxyHiAb7fQT25u/v
	4	A Princess Of Mars	Edgar Rice Bu...	action	A_Princess_Of_Mars.jpg	https://drive.google.com/file/d/13uWaOR_NG3y5zIT4ckj4B0Xx9X6e9o_R
	5	Living History	Ben Essex	history	Living_History.jpg	https://drive.google.com/file/d/132zOH8tOKBst9O3CWJg1.1QYGcbU-SzQ
✱	NULL	NULL	NULL	NULL	NULL	NULL

Figure 18 : base de données des livres

← → ↻

localhost:8080/e_Library_Project3/ManageUsers.jsp

Maps

Gmail

YouTube

Maps

Traduire

Tous les favoris

E-books

[Home](#)
[Manage Users](#)
[Manage Books](#)
[Logout](#)

Manage Users :

Name	Email	Password	Delete
benlakhali	benlakhali47@email.com	123456	Remove

Figure 19 : Fenêtre de gestion des utilisateurs

	id	admin_name	admin_email	admin_pass
	1	anass	benlakhali47@email.com	123456@
	NULL	NULL	NULL	NULL

Figure 20 : base de données des admins

b. Espace Utilisateur

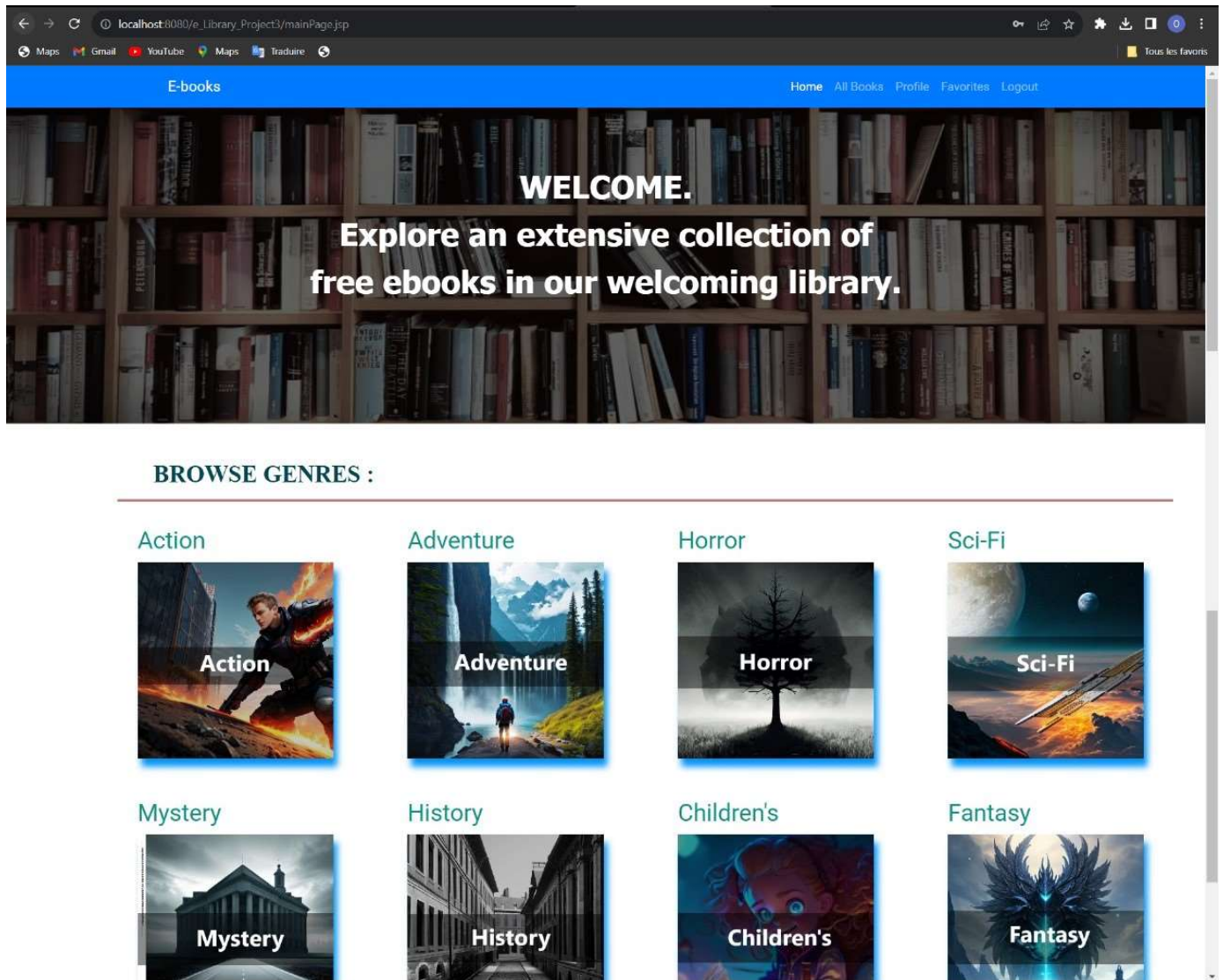
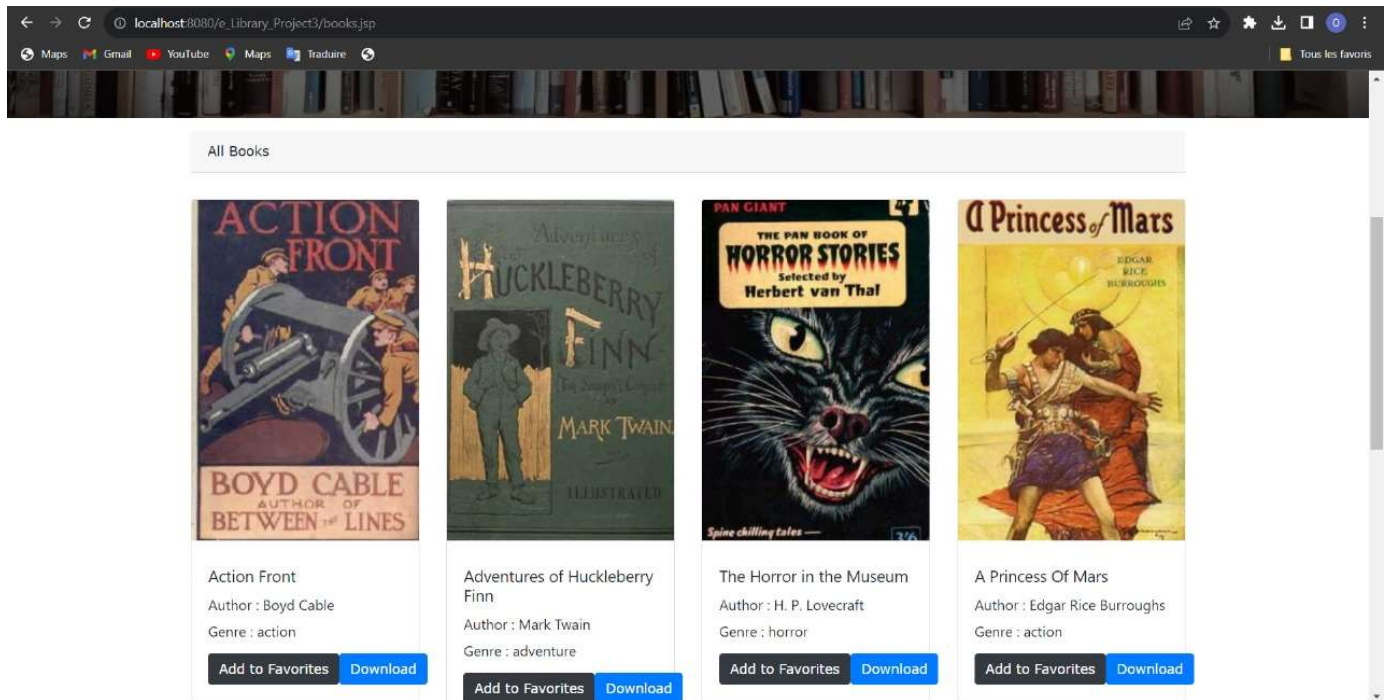
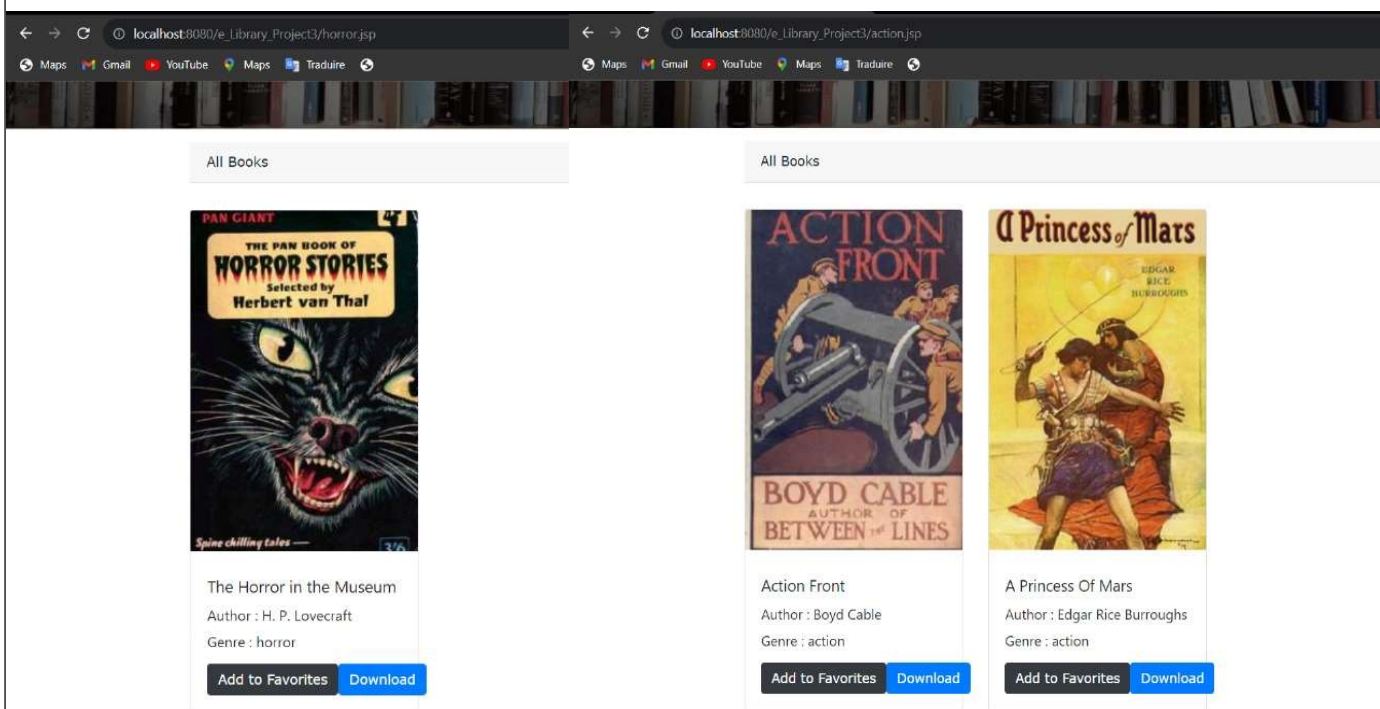


Figure 21 : Fenêtre principale du client

Si l'utilisateur a choisi la fenêtre de toutes les livres, Utilisateur peut télécharger et ajouter au favorites les livres.



L'utilisateur peut aussi filtrer sa recherche avec genre des livres :



L'Utilisateur peut consulter et supprimer des livres dans fenêtre Favorites :

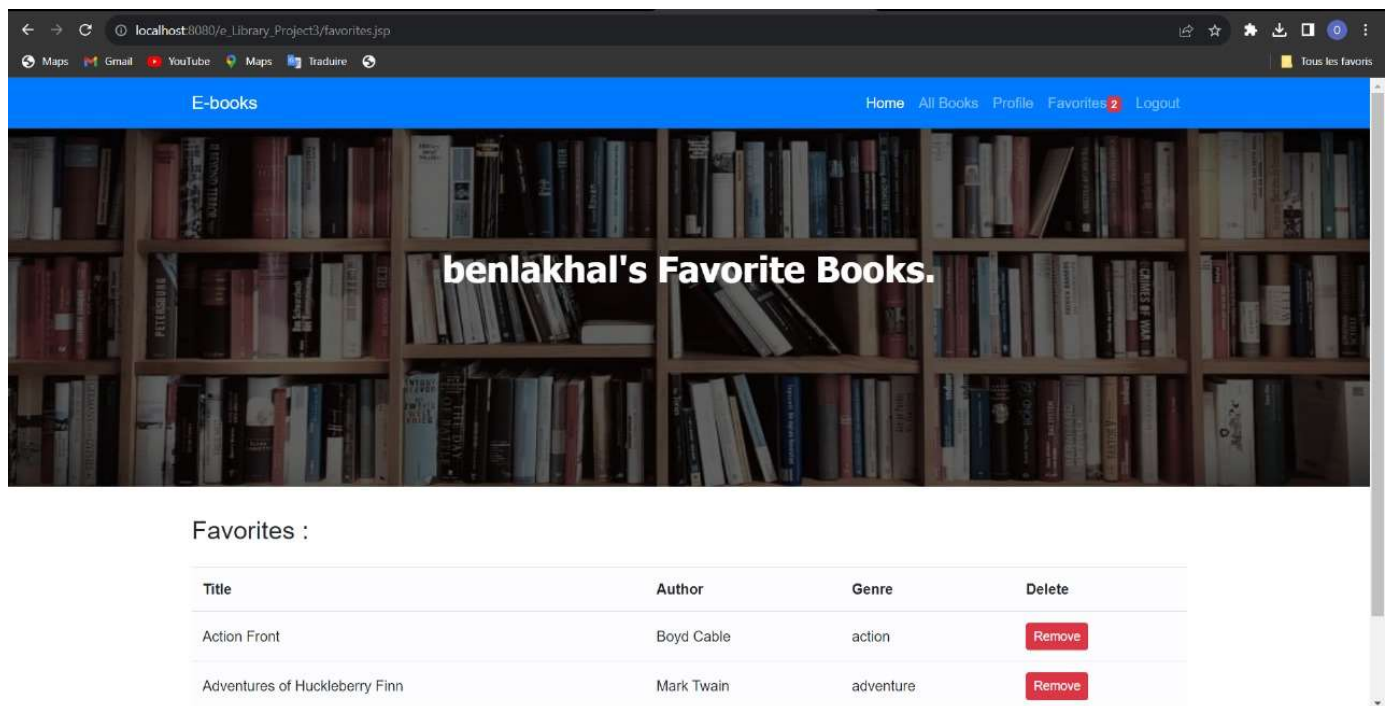


Figure 24 : Favorites books

Les données de cette fenêtre sont totalement sauvegardées dans notre base de données.

	id	title	author	genre	image	Download_link
▶	1	Action Front	Boyd Cable	action	Action_Front.jpg	https://drive.google.com/file/d/1nsPyyrTrRhOf...
	4	A Princess Of Mars	Edgar Rice Burroughs	action	A_Princess_Of_Mars.jpg	https://drive.google.com/file/d/13uWaOR_NG3...

Figure 25 : Base de données des favorites livres

CHAPITRE IV : Résultats et discussion




1. Résultats des tests

1.1 Tests unitaires





En partant du bas, le premier niveau de test est le "test unitaire". Il s'agit de vérifier que chaque spécifiée dans la "Conception du composant" a été mise en œuvre dans le composant. En théorie, un testeur indépendant devrait le faire, mais en pratique, le développeur fait généralement car ce sont les seules personnes qui comprennent le fonctionnement d'un composant. Le problème avec un composantest qu'il n'assure qu'une petite partie des fonctionnalités d'un système, et il repose sur la coopération avec d'autres parties du système, qui n'ont peut-être pas encore été construites. Pour surmonter ce problème, le développeur construit ou utilise un logiciel spécial pour tromper le composant de croire qu'il fonctionne dans un système pleinement fonctionnel.

Le résumé des tests unitaires est fourni ci-dessous

- Pour l'administration :

UNITÉ	BUT	VÉRIFIÉE
Connexion de l'administrateur	Cette unité détecte l'autorisation d'accès l'administrateur	OUI 
Ajouter et modifier les détails des livres	Cette unité permet à l'administrateur de modifier les détails des livres.	OUI 
Ajouter et modifier les détails des utilisateurs.	Cette unité permet à l'administrateur de modifier les détails des utilisateurs.	OUI 

- Pour le Client :

UNITÉ	BUT	VÉRIFIÉE
Rechercher un livre	Cette unité fouille les livres avec des Filtres.	OUI 
Télécharger un livre	L'utilisateur peut télécharger un livre.	OUI 
Ajouter les livres au favorites	L'utilisateur peut ajouter les livres au favorites	OUI 
Supprimer un livre	L'utilisateur peut supprimer les livres dans favorites.	OUI 

CHAPITRE V : Résumé et conclusions

1. Résumé des résultats

Les réalisations suivantes ont été faites au cours du projet :

- Nous avons eu l'occasion d'apprendre la technologie JAVA et MySQL
- A appris à travailler dans Eclipse.
- Nous avons appris à gérer un projet de manière efficace et correcte.
- Nous avons appris à faire face à diverses situations adverses tout en gérant et en développant des logiciels.
- Apprendre les différentes phases du développement de logiciels et les processus de génie logiciel impliqués dans le développement de logiciels.

GitHub Link: <https://github.com/OussamaElkihel/ProjectS3>