



ALGORITHME ET PROGRAMMATION JAVA

*Rapport : Développement d'une application de gestion
des projets*



Nolwen DJUISSI

Ivan SIMO

22 mai 2025

Table des matières

Introduction	3
Structure du projet	3
Fonctionnalités développées	3
Approche technique	4
Améliorations et perspectives	5
Conclusion	7
Annexes	7
Lien vers le github :	7
https://github.com/angyDKA/gestion_des_projets.git	7

Introduction

Dans le cadre du module II.1102 du semestre 2 de l'année universitaire 2024–2025, ce projet a été conçu dans le but de développer une application JavaFX de gestion des emplois du temps au sein d'un établissement d'enseignement supérieur. L'application vise à centraliser la gestion des cours, des salles et des créneaux horaires, tout en offrant une interface interactive adaptée aux besoins spécifiques des trois profils utilisateurs : étudiants, enseignants et administrateurs. L'objectif principal était de fournir une solution efficace, intuitive et extensible pour organiser et consulter les plannings académiques.

Structure du projet

Le projet est structuré selon une architecture propre au modèle MVC (Modèle-Vue-Contrôleur), répartie en quatre grands packages :

- **model** : contient les entités métiers comme Cours, Utilisateur, Salle, Notification, avec leurs attributs et méthodes associées.
- **controller** : gère les interactions avec la base de données. Chaque entité dispose de son propre contrôleur (ex. : CoursController, NotificationController).
- **dao** : Ce package connecte l'application à la base de données. Les classes DAO gèrent les opérations principales : création, lecture, mise à jour et suppression des données.
- **view** : regroupe toutes les interfaces JavaFX spécifiques à chaque rôle utilisateur. Chaque vue est indépendante et assure la navigation intuitive dans l'application.
- **utils** : contient la classe Database pour la gestion sécurisée de la connexion JDBC à la base MySQL.

Fonctionnalités développées

Ce projet est une application JavaFX permettant de gérer les **utilisateurs**, **cours** et **séances** dans le cadre d'un emploi du temps universitaire.

Il a été conçu avec une architecture propre (MVC) et utilise une base de données MySQL pour la persistance.

Fonctionnalités effectivement développées

Gestion des Utilisateurs

- Ajouter, modifier, supprimer un utilisateur

- Afficher les informations détaillées d'un utilisateur
- **ComboBox des cours affichée, mais l'association à un cours est en cours de finalisation**
- Dissociation d'un cours non encore fonctionnelle

Gestion des Cours

- Ajouter, modifier, supprimer un cours (nom + dates)
- Affichage dans un tableau
- Génération de rapport CSV listant les utilisateurs associés à chaque cours
- Association/dissociation de séances partiellement implémentée
- Association/dissociation de membres à un cours partiellement fonctionnelle

Gestion des Séances

- Ajouter, modifier, supprimer une séance
- Champs : titre, description, statut, date limite, priorité
- Affichage dans une table fonctionnel
- Les vues Kanban et calendrier sont appelées mais **pas encore implémentées**
- Association à un cours en cours de développement

Fonctionnalités en développement

- Association/Dissociation utilisateurs ↔ cours
- Association/Dissociation séances ↔ cours
- Navigation vers vue Kanban ou calendrier
- Notification de rappel ou alerte en cas de priorité

Approche technique

Le projet repose sur une architecture modulaire en JavaFX, structurée selon le modèle MVC (Modèle-Vue-Contrôleur). Cette organisation garantit une séparation claire entre les données, la logique métier et la présentation graphique.

La couche Modèle comprend les entités métier telles que Cours, Salle, Utilisateur et Notification. Les rôles sont modélisés à travers des classes dérivées spécifiques (Etudiant, Enseignant, Administrateur) pour refléter les permissions et comportements distincts.

Les contrôleurs orchestrent l'interaction avec la base de données MySQL via JDBC. Ils permettent l'exécution sécurisée de requêtes SQL, notamment dans CoursController et NotificationController, pour la gestion des cours et des alertes.

La couche Vue est constituée de différentes interfaces JavaFX pour chaque type d'utilisateur (AdminDashboardView, EtudiantDashboardView, EnseignantDashboardView). Ces interfaces utilisent des composants modernes comme TextArea, ComboBox, TableView et PieChart pour une interaction visuelle optimale.

La base de données a été conçue sous MySQL et gérée via phpMyAdmin. Elle comprend les tables nécessaires à la gestion des utilisateurs, des cours, des salles, des inscriptions et des notifications, toutes reliées entre elles par des clés étrangères pour assurer l'intégrité référentielle.

Le développement a été réalisé en Java 23, avec le SDK JavaFX et les IDE IntelliJ IDEA et Eclipse selon les préférences.

Améliorations et perspectives

Fonctionnalités à finaliser ou corriger

Utilisateurs

- **Remplir automatiquement la ComboBox des cours** dans la vue utilisateur-view.fxml
- **Associer un utilisateur à un cours sélectionné** (avec mise à jour en BDD via CoursDAO)
- **Dissocier un utilisateur d'un cours** (en supprimant l'entrée dans la table Utilisateur_Cours)

Rafraîchir la TableView des utilisateurs après association/dissociation

Cours

- **Associer/dissocier des séances à un cours** (via ComboBox ou boîte de dialogue)

- **Associer/dissocier des utilisateurs à un cours** depuis la page des cours (pas seulement dans UtilisateurController)
- **Afficher les séances et membres associés** dans une nouvelle section ou pop-up
- **Corriger les champs pour bien afficher les utilisateurs dans le CSV**
- **Vérifier les doublons lors de l'ajout d'un utilisateur à un cours**

Séances

- **Associer une séance à un cours** au moment de sa création ou modification
- Ajouter une **fonction d'export CSV des séances**
- **Implémenter la vue Kanban**
- **Implémenter la vue Calendrier**
- **Ajout de filtres par statut ou priorité**

Ergonomie & Interface

- Centraliser les couleurs & styles dans un fichier style.css
- Rendre les vues adaptables (pas de dimensions figées)
- Grouper les boutons liés (CRUD, association, export) dans des HBox / VBox propres
- Ajout de **Tooltips** ou **icônes** pour améliorer la lisibilité

Structure & Code

- Séparer clairement les **DAO, Contrôleurs, Modèles**
- Ajouter **des logs ou messages console** pour faciliter les débogages
- Nettoyer les System.out.println inutiles
- S'assurer que **toutes les exceptions SQL sont capturées proprement**

Conclusion

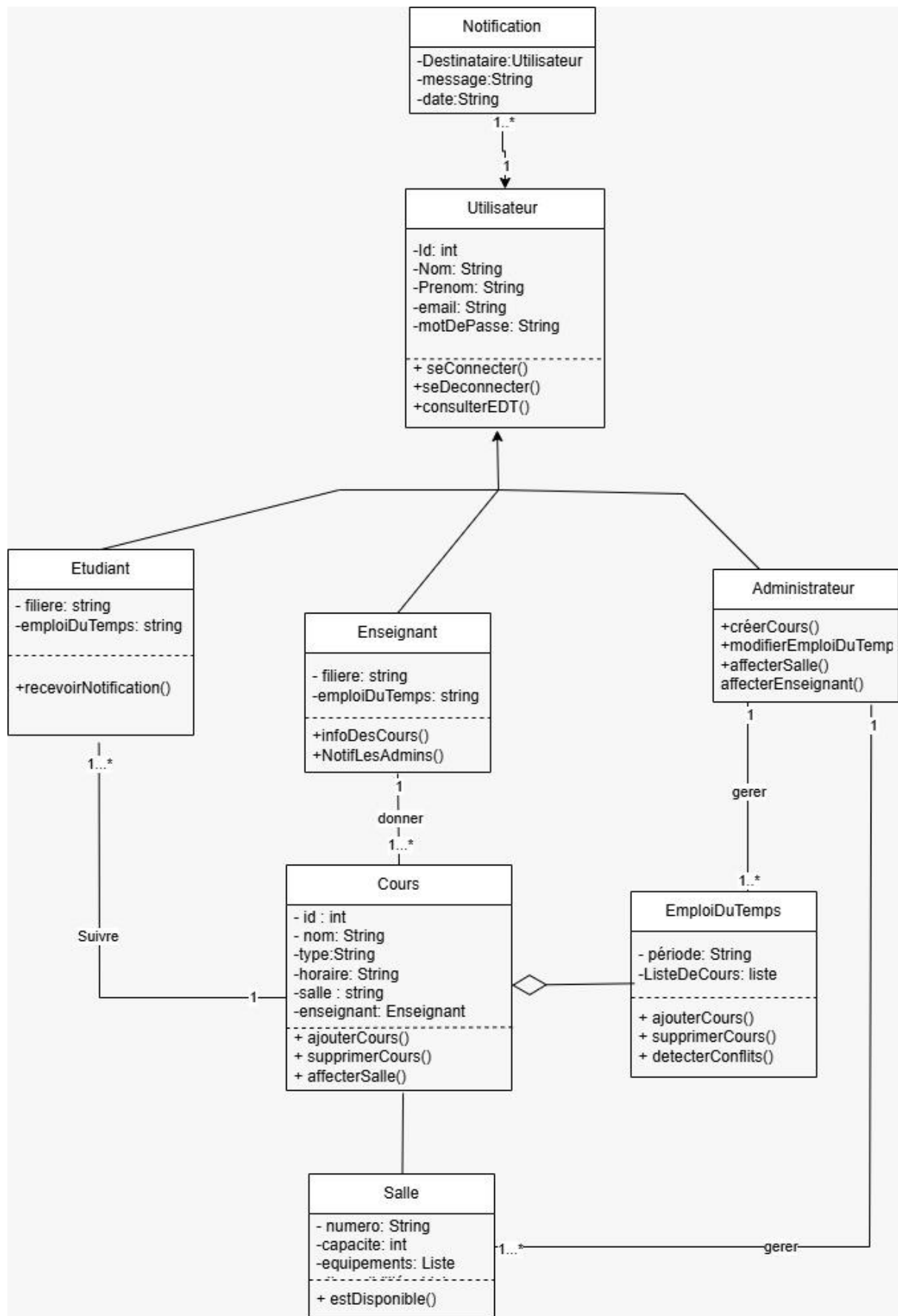
Ce projet a permis la conception d'une solution logicielle complète pour la gestion des emplois du temps, répondant fidèlement au cahier des charges initial. Les différentes fonctionnalités ont été développées dans une architecture robuste et extensible, en intégrant des aspects essentiels tels que la modularité, l'interopérabilité avec une base de données relationnelle et une interface utilisateur intuitive.

Les rôles utilisateurs sont clairement séparés, les fonctionnalités avancées telles que les notifications et les statistiques sont opérationnelles, et l'export des données garantit une ouverture vers d'autres formats. Ce projet représente une réalisation concrète mobilisant des compétences en programmation orientée objet, développement graphique JavaFX et gestion de données, tout en posant les bases pour des évolutions futures prometteuses.

Annexes

Lien vers le github :

https://github.com/angyDKA/gestion_des_projets.git





Plateforme de Gestion des Cours et Séances

Tableau de Bord - Gestion des Cours Universitaires

Navigation

Utilisateurs

Cours

Séances

Gestion des Séances

Titre	Statut	Date de la séance	Séance encadrée
TP	A_FAIRE		Oui

Titre

Description

Statut

☐ Séance encadrée ?

Date de la séance

Ajouter

Modifier

Supprimer

Vue Kanban

Calendrier

© 2025 - Gestion Universitaire - Tous droits réservés.

localhost / mysql wampserver /

github - Recherche

angyDKA/gestion_des_projets: Ap

localhost/phpmyadmin/db_sql.php?db=gestionprojets&table=&server=1&target=&token=49ae2283ef04939c9d750b6db561a641#PMAURL-1:db_structure...

phpMyAdmin

(Tables récentes) ...

Nouvelle base de données

findyourcourt

gestionprojets

Nouvelle table

cours

cours_utilisateur

seancecours

utilisateur

utilisateur_cours

Serveur: mysql wampserver » Base de données: gestionprojets

Structure

SQL

Rechercher

Requête

Exporter

Importer

Opérations

Privileges

Procédures stockées

Table	Action	Lignes	Type	Interclassement	Taille	Perte
<input type="checkbox"/> cours	<div><div>Afficher</div><div>Structure</div><div>Rechercher</div><div>Insérer</div><div>Vider</div><div>Supprimer</div></div>	~1	InnoDB	latin1_swedish_ci	16 Kio	-
<input type="checkbox"/> cours_utilisateur	<div><div>Afficher</div><div>Structure</div><div>Rechercher</div><div>Insérer</div><div>Vider</div><div>Supprimer</div></div>	~0	InnoDB	latin1_swedish_ci	32 Kio	-
<input type="checkbox"/> seancecours	<div><div>Afficher</div><div>Structure</div><div>Rechercher</div><div>Insérer</div><div>Vider</div><div>Supprimer</div></div>	~1	InnoDB	latin1_swedish_ci	48 Kio	-
<input type="checkbox"/> utilisateur	<div><div>Afficher</div><div>Structure</div><div>Rechercher</div><div>Insérer</div><div>Vider</div><div>Supprimer</div></div>	~1	InnoDB	latin1_swedish_ci	16 Kio	-
<input type="checkbox"/> utilisateur_cours	<div><div>Afficher</div><div>Structure</div><div>Rechercher</div><div>Insérer</div><div>Vider</div><div>Supprimer</div></div>	~0	InnoDB	latin1_swedish_ci	32 Kio	-
5 tables	Somme	3	InnoDB	latin1_swedish_ci	144 Kio	0 0

Tout cacher

Don la sélection

