

ULTIMATE TRAIN TRAFFIC CONTROLLER

<https://gitlab.com/uttc-team/ultimate-train-traffic-controller>

Dans le cadre de ma formation en BTS SIO option SLAM, j'ai participé à un projet intitulé **Ultimate Train Traffic Controller (UTTC)**, une application Java destinée à gérer les trafics de trains. Cette réalisation a été effectuée en équipe, entre **novembre 2024 et avril 2025**, au sein du **Lycée de l'Hautil**.

L'objectif principal de ce projet était de concevoir une solution permettant la **gestion des trajets ferroviaires**, l'**affectation des conducteurs** ainsi que la **planification des horaires**. Pour atteindre ces objectifs, nous avons adopté une démarche structurée, en commençant par la **modélisation UML** de l'application à l'aide de Draw.io, puis en développant l'application en Java avec une interface graphique réalisée en **FXML via Scene Builder**.

J'ai conçu l'interface FXML avec **Scene Builder** pour permettre l'ajout et la suppression d'un utilisateur ainsi qu'un trajet. Cette interface est reliée à une classe contrôleur Java (**Ajout_supp_utili**) qui gère les interactions. Lors du développement de l'application Java (**UTTCApplication**), j'ai dû modifier la classe **Utilisateur** en ajoutant un constructeur par défaut, afin de pouvoir l'utiliser dans la classe **MainDAO**, créée par mon collègue.

J'ai également amélioré les classes **MainDAO** et **UTTCDAO** en :

- ajoutant une **méthode d'insertion** d'un utilisateur via une requête préparée en Java avec **try-catch** et JDBC, pour prévenir les **injections SQL**. Cette méthode est appelée dans le contrôleur Java ;
- créant la **méthode `getUtilisateurByLogin`**, permettant de récupérer un utilisateur via son identifiant ;
- implémentant une **méthode de suppression d'un utilisateur**, en vérifiant d'abord son existence avec **`getUtilisateurByLogin`**.

Pour ce qui est de la base de données, j'ai apporté plusieurs améliorations. Afin d'éviter les doublons, j'ai ajouté une **contrainte d'unicité** sur la table **Utilisateur** :

```
ALTER TABLE Utilisateur ADD UNIQUE KEY `unique_login` (`login`);
```

Par ailleurs, pour renforcer la **sécurité des mots de passe**, j'ai mis en place le **hachage** de ces derniers à l'aide de la fonction **`PASSWORD()`**.

Lors du développement, plusieurs difficultés ont été rencontrées. Notamment, des divergences entre les noms d'attributs en Java et ceux de la base de données ont généré des erreurs de communication. Ces problèmes ont été résolus par une harmonisation des noms dans le code. L'intégration des contrôleurs Java dans la structure globale de l'application a également demandé des ajustements pour assurer la bonne transmission des données entre l'interface et la logique métier.

En plus du développement technique, j'ai participé à la **gestion du projet** : élaboration d'un **diagramme de Gantt**, suivi de l'avancement, répartition des tâches, et travail collaboratif via GitLab. L'intelligence artificielle, notamment ChatGPT, a été utilisée comme soutien ponctuel pour mieux comprendre certains problèmes techniques.

En conclusion, ce projet m'a permis de consolider mes compétences en **programmation orientée objet**, en **gestion de base de données**, et en **travail collaboratif**. L'application obtenue est fonctionnelle, avec une interface intuitive, une base de données sécurisée, et un code structuré. Cette expérience m'a offert une vision concrète du cycle de développement d'une application, de la conception à la mise en œuvre.

Table utilisateur de la bdd uttc_app :

The top screenshot shows the 'utilisateur' table in the 'uttc_app' database. The table has 4 rows of data. The columns are: id, login, mdp, prenom, nom, dateEmbauche, and role.

#	id	login	mdp	prenom	nom	dateEmbauche	role
1	1	Axel	*792E16DF76725EE5AF8FA522B1C28089DCB7F1	Axel	Auger	2025-03-12	ADMIN
2	2	Benoit	*C1F5474779920054AD52083B4EC6039C23727B3A	Benoit	Dupont	2025-03-12	CONDUCTEUR
3	3	Valentin	*E61A8B3E11FA6CA643C8082DCE2B6530BEFAD6A	Valentin	Deschamps	2025-03-12	ADMIN
4	4	Agip123	*6C08BBDEA9E500F5F0B26AD26881719AF2B3C92	Agip	Shalik	2025-03-19	MANAGER

The bottom screenshot shows the table structure of the 'utilisateur' table. The table has 7 columns: id, login, mdp, prenom, nom, dateEmbauche, and role.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	login	varchar(50)	utf8mb4_unicode_520_ci		Yes	NULL			Change Drop More
3	mdp	varchar(50)	utf8mb4_unicode_520_ci		Yes	NULL			Change Drop More
4	prenom	varchar(255)	utf8mb4_unicode_520_ci		Yes	NULL			Change Drop More
5	nom	varchar(50)	utf8mb4_unicode_520_ci		Yes	NULL			Change Drop More
6	dateEmbauche	date			Yes	NULL			Change Drop More
7	role	enum('ADMIN', 'MANAGER', 'CONDUCTEUR')	utf8mb4_unicode_520_ci		Yes	NULL			Change Drop More

Interaction avec la bdd lors d'ajout d'un utilisateur

The image displays two screenshots of a web application and phpMyAdmin, illustrating the process of adding a user to a database.

Top Screenshot:

- Left Panel (Web Application):** The 'Ajout employé' form is shown. The 'Prénom' field is filled with 'Agip'. The 'Login' field is filled with 'Agip123'. The 'Mot de passe' field is filled with '*****'. The 'Créer' button is visible.
- Right Panel (phpMyAdmin):** The 'utilisateur' table is selected. The query results show 3 rows (0 - 3) with columns 'id', 'login', 'mdp', and 'prenom'. The data is as follows:

id	login	mdp	prenom
1	AXel	*792E16DF767250EE5AF0FA522B1C28089DCBA7F1	Axel
2	Benoit	*C1F5474779920954AD52083BAC6839C23727B3A	Benoit
3	Valentin	*E61A8B3E11FA6CA643C882DCE2BE630BEFAD7A6	Valentin

Bottom Screenshot:

- Left Panel (Web Application):** The 'Ajout employé' form is shown. The 'Prénom' field is filled with 'Agip'. The 'Login' field is filled with 'Agip123'. The 'Mot de passe' field is filled with '*****'. The 'Créer' button is visible.
- Right Panel (phpMyAdmin):** The 'utilisateur' table is selected. The query results show 2 rows (0 - 2) with columns 'id', 'login', 'mdp', and 'prenom'. The data is as follows:

id	login	mdp	prenom
1	AXel	*792E16DF767250EE5AF0FA522B1C28089DCBA7F1	Axel
2	Benoit	*C1F5474779920954AD52083BAC6839C23727B3A	Benoit

Diagramme des classes

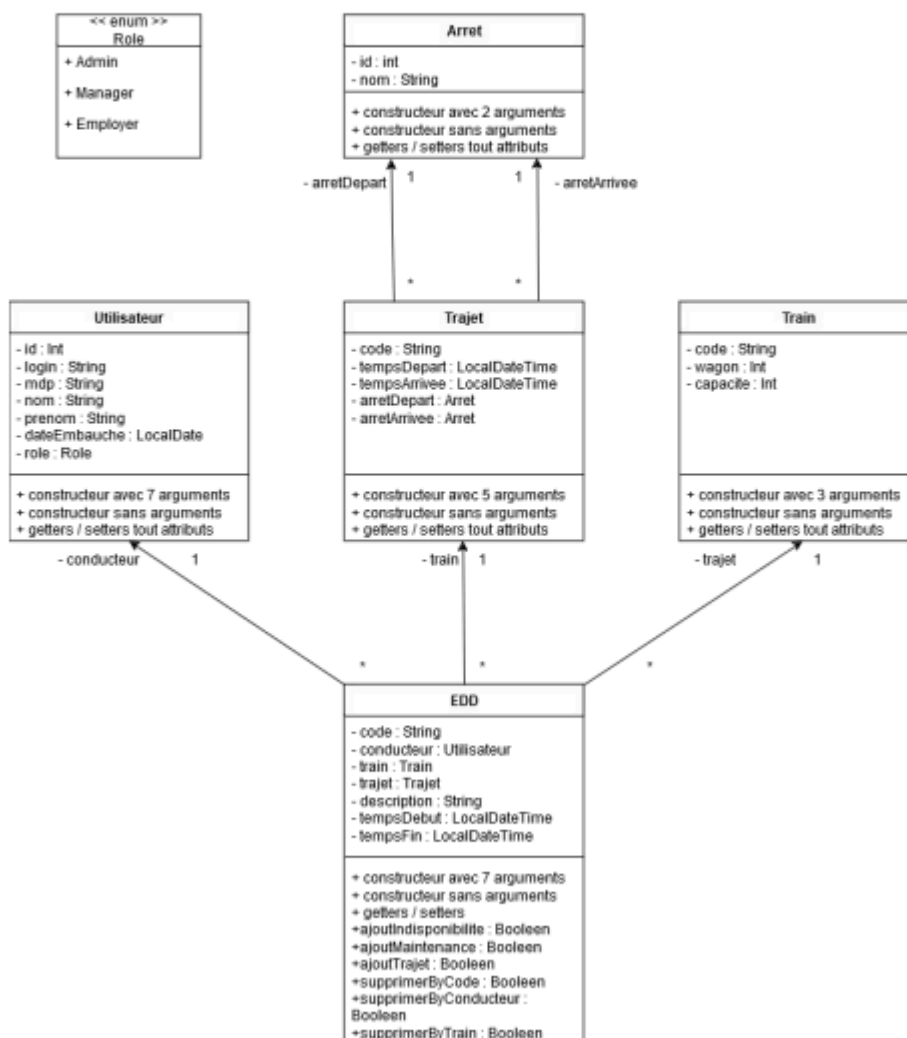


Diagramme de gantt

