

## Introduction

L'objectif de ce test technique est de modéliser une base de données pour la gestion des trajets de bus et de concevoir une interface de gestion simple en utilisant l'admin de Django. Ce document décrit les étapes prises pour atteindre cet objectif, les défis rencontrés et les solutions mises en place.

## Analyse et Planification

### Modèles à Créer

Pour répondre aux exigences métiers, il a été nécessaire de créer deux nouveaux modèles : **BusStop** et **BusShift**.

**BusStop** : représente chaque arrêt sur l'itinéraire.

- `bus_shift` : clé étrangère vers le modèle **BusShift**.
- `place` : emplacement géographique.
- `arrival_time` : heure d'arrivée.

**BusShift** : représente un trajet en bus.

- `bus` : clé étrangère, associée au modèle **Bus**.
- `driver` : clé étrangère, associée au modèle **Driver**.
- `stops` : représente les arrêts intermédiaires.
- `departure_time` : heure de début du trajet.
- `arrival_time` : heure de fin du trajet.

Ces relations entre les modèles assurent qu'un même bus et un même conducteur ne peuvent pas être affectés à d'autres trajets se chevauchant au même moment.

### Conception Modulaire

J'ai choisi de créer une nouvelle application nommée "transportation" pour gérer les aspects liés aux trajets de bus et aux arrêts. Cela permet de maintenir le projet modulaire et extensible.

## Implémentation

### Défis Rencontrés

#### 1. Gestion des Relations ManyToMany :

- **Problème** : Lors de la tentative de création d'une instance de **BusShift**, Django nécessite que l'objet soit enregistré et possède un ID avant de pouvoir gérer les relations **ManyToMany**.
- **Solution** : Réécrire la méthode `save` du modèle **BusShift** pour d'abord enregistrer l'instance, puis traiter la relation **ManyToMany**. Toutefois, une solution plus propre a été trouvée en déplaçant la logique de validation dans le formulaire.

#### 2. Validation des Données :

- **Problème** : Validation des horaires pour éviter les chevauchements de trajets pour un même bus ou conducteur.
- **Solution** : Déplacer la logique de validation dans le formulaire et utiliser ce formulaire dans l'admin de Django pour s'assurer que toutes les validations nécessaires sont effectuées avant l'enregistrement de l'instance.

### Validation et Enregistrement

Les étapes de validation incluent :

- Assurer qu'il y a au moins deux arrêts de bus dans chaque trajet.
- Vérifier que les horaires de travail des bus et des conducteurs ne se chevauchent pas.
- Déterminer les heures de départ et d'arrivée des trajets à partir des heures d'arrivée des arrêts.

### Tests et Validation

Pour vérifier le bon fonctionnement de l'application, plusieurs tests unitaires et fonctionnels ont été écrits :

1. **Création de BusStop** :
  - Vérification que les arrêts peuvent être créés correctement et affichés dans l'admin.
2. **Création de BusShift** :
  - Vérification que les trajets peuvent être créés sans chevauchement de bus ou de conducteurs.
  - Validation des heures de départ et d'arrivée des trajets.
3. **Validation des Conflits de Disponibilité** :
  - Assurer qu'un même bus ou conducteur ne peut être assigné à des trajets se chevauchant.
4. **Tests des Formulaires** :
  - Les formulaires ont été testés à 100% pour garantir la couverture de toutes les situations possibles.

Les tests des modèles, bien que complets, peuvent encore être améliorés pour une couverture optimale.

### Test manuel

L'objectif de la partie test manuel est de s'assurer que l'interface d'administration de Django permet de créer et de gérer correctement les instances de BusStop et de BusShift, et de vérifier l'intégrité des données ainsi que l'efficacité des règles métiers. Voici les points clés du test manuel :

#### Situation de vérification :

1. **Assurez-vous qu'il y a au moins deux arrêts de bus** :
  - Chaque changement de bus (BusShift) doit contenir au moins deux arrêts (BusStop).

Add bus shift

Please correct the errors below.

At least two stops are required

Bus: Bus: bus 1 (id: 1) ✎ +

Driver: Driver: user2 (id: 1) ✎ +

Stops: At least two stops are required

BusStop: Place 1 arrive at: 2024-06-07 18:00 (ID: 1)  
 BusStop: rue de Roussel arrive at: 2024-06-07 06:00 (ID: 2)  
 BusStop: rue Chantal Jacquet arrive at: 2024-06-07 19:00 (ID: 3)  
 BusStop: rue Augustin Mallet arrive at: 2024-06-08 06:00 (ID: 4)  
 BusStop: rue Chantal Jacquet arrive at: 2024-06-08 06:30 (ID: 5)

Hold down "Control", or "Command" on a Mac, to select more than one.

Save and add another Save and continue editing SAVE

## 2. Vérifiez que les horaires de travail ne se chevauchent pas :

- Le même bus (Bus) ne peut pas être affecté à plusieurs trajets dans la même période.

BUS	DRIVER	DEPARTURE TIME	ARRIVAL TIME	SHIFT DURATION
<input type="checkbox"/> Bus: bus 1 (id: 1)	Driver: user2 (id: 1)	June 8, 2024, 6 a.m.	June 8, 2024, 6:30 a.m.	0h 30m

Please correct the error below.

The bus Bus: bus 1 (id: 1) is already booked during this time period.

Bus: Bus: bus 1 (id: 1) ✎ +

Driver: Driver: lamychristelle (id: 4) ✎ +

Stops: BusStop: Place 1 arrive at: 2024-06-07 18:00 (ID: 1)  
 BusStop: rue de Roussel arrive at: 2024-06-07 06:00 (ID: 2)  
 BusStop: rue Chantal Jacquet arrive at: 2024-06-07 19:00 (ID: 3)  
 BusStop: rue Augustin Mallet arrive at: 2024-06-08 06:00 (ID: 4)  
 BusStop: rue Chantal Jacquet arrive at: 2024-06-08 06:30 (ID: 5)

Hold down "Control", or "Command" on a Mac, to select more than one.

Save and add another Save and continue editing SAVE

## 3. Vérifiez que les horaires de travail du conducteur ne se chevauchent pas :

- Un même conducteur (Driver) ne peut pas être affecté à plusieurs trajets au cours d'une même période.

BUS	DRIVER	DEPARTURE TIME	ARRIVAL TIME	SHIFT DURATION
<input type="checkbox"/> Bus: bus 1 (id: 1)	Driver: user2 (id: 1)	June 8, 2024, 6 a.m.	June 8, 2024, 6:30 a.m.	0h 30m

Add bus shift

Please correct the error below.

The driver Driver: user2 (id: 1) is already assigned to another shift during this time period.

Bus: Bus: JH-775-LT (id: 5) ✎ +

Driver: Driver: user2 (id: 1) ✎ +

Stops: BusStop: Place 1 arrive at: 2024-06-07 18:00 (ID: 1)  
 BusStop: rue de Roussel arrive at: 2024-06-07 06:00 (ID: 2)  
 BusStop: rue Chantal Jacquet arrive at: 2024-06-07 19:00 (ID: 3)  
 BusStop: rue Augustin Mallet arrive at: 2024-06-08 06:00 (ID: 4)  
 BusStop: rue Chantal Jacquet arrive at: 2024-06-08 06:30 (ID: 5)

Hold down "Control", or "Command" on a Mac, to select more than one.

Save and add another Save and continue editing SAVE

## 4. Vérifiez les heures de départ et d'arrivée :

- L'heure de départ du trajet doit être l'heure d'arrivée du premier arrêt.
- L'heure d'arrivée du trajet doit être l'heure d'arrivée du dernier arrêt.

BusStop: Place 1 arrive at: 2024-06-07 18:00 (ID: 1)  
BusStop: rue de Rousset arrive at: 2024-06-07 06:00 (ID: 2)  
BusStop: rue Chantal Jacquet arrive at: 2024-06-07 19:00 (ID: 3)  
BusStop: rue Augustin Mallet arrive at: 2024-06-08 06:00 (ID: 4)  
BusStop: rue Chantal Jacquet arrive at: 2024-06-08 06:30 (ID: 5)

Hold down "Control" or "Command" on a Mac to select more than one

<input type="checkbox"/>	BUS	DRIVER	DEPARTURE TIME	ARRIVAL TIME	SHIFT DURATION
<input type="checkbox"/>	Bus: bus 1 (id: 1)	Driver: user2 (id: 1)	June 8, 2024, 6 a.m.	June 8, 2024, 6:30 a.m.	0h 30m

## Résultats

Les tests ont confirmé que les principales fonctionnalités de l'application fonctionnent comme prévu :

- Les trajets de bus peuvent être créés et gérés sans chevauchement de bus ou de conducteurs.
- Les validations sont correctement appliquées pour assurer l'intégrité des données.
- L'interface d'administration de Django permet une gestion efficace des trajets