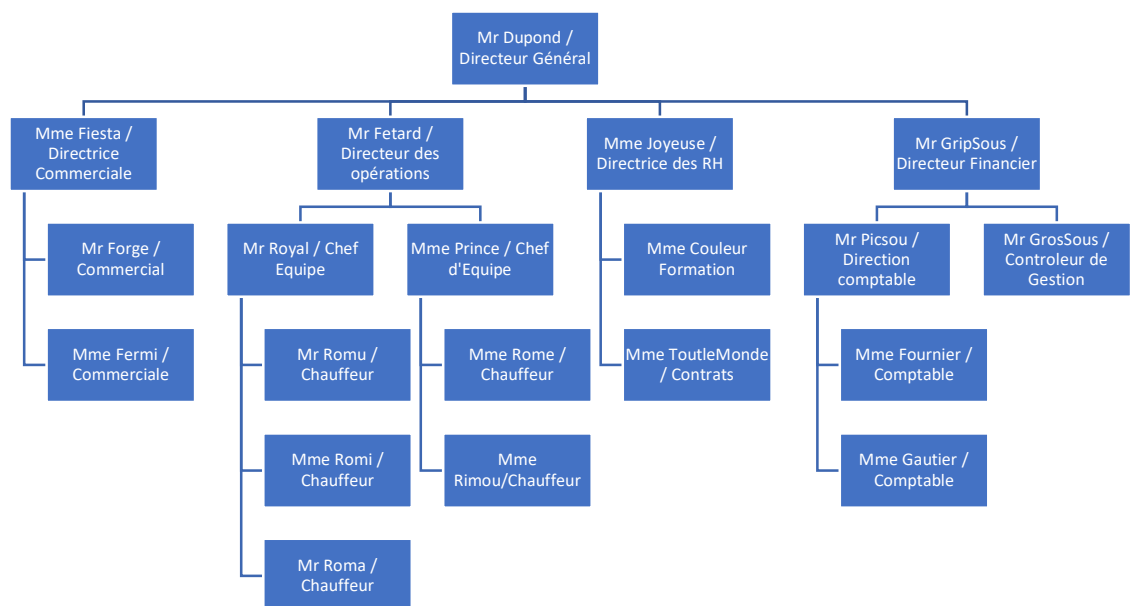


TransConnect

La société est une société de transports routiers qui possède

- Une flotte de véhicules qu'elle utilise pour livrer des marchandises de toutes sortes.
- Un personnel réparti selon l'organigramme suivant :



Cette société cherche à s'agrandir et a besoin de recruter massivement compte tenu des succès rencontrés ces derniers mois. Aussi, elle ouvre des postes de chef d'équipe et des postes de chauffeurs.

À tout moment, le directeur de l'entreprise, Mr Dupond veut connaître tous les membres de son entreprise avec les liens hiérarchiques et leur profil. Bien évidemment, il sera très utile de proposer un affichage simple et compréhensible de cet organigramme. Par ailleurs, étant plus visuel, il souhaite pour chaque départ ou licenciement ou chaque recrutement visualiser la structure de son entreprise.

Les **salariés** ont des caractéristiques précises.

Un salarié est donc une personne qui possède un N°SS, un nom, un prénom, une date de naissance, une adresse postale et une adresse mail ainsi qu'un téléphone. Le nom, l'adresse, le mail et le téléphone sont modifiables.

Le salarié possède encore une date d'entrée dans la société, un poste et un salaire. Le poste et le salaire sont modifiables également

TransConnect possède également une base de données de ses **clients** avec les mêmes caractéristiques que les personnes ci-dessus décrites. (On simulera ces données par une collection de clients en mémoire). Les clients commandent des services de livraisons de marchandises de toutes sortes et de tout volume aux quatre coins de la **France**.

Chaque **commande** met en relation un client, une livraison d'un point A à un point B, un prix, un véhicule, un chauffeur et une date.

On appliquera les contraintes suivantes,

- Un chauffeur ne peut faire qu'une seule livraison par jour
- Il ne peut livrer que, évidemment s'il est libre.
- Une livraison effectuée suppose le paiement du client dans notre application
- Le tarif sera établi en fonction du kilométrage et du véhicule loué

Une fois la livraison effectuée, on conservera l'historique des livraisons réalisées pour chacun des clients. Mr Dupond pourra donc estimer et faire la part des choses entre les bons et les moins bons clients et proposer des remises en fonction de leur statut.

Enfin la flotte de **véhicules** est répartie de la manière suivante :

- Voiture : pour transporter des passagers dont le nombre est à préciser
- Camionnette : dont l'usage est à préciser. Exemple les artisans dans la vitrerie ont besoin d'un cadre spécifique pour transporter les verres. Seul l'usage sera à définir et pas les accessoires mis en cause.
- Poids Lourds ou camions avec des volumes et des matières transportés différents.

Le camion-citerne

Le camion-citerne, avec sa cuve, est utilisé pour transporter des liquides ou encore des produits gazeux pour l'industrie chimique et agroalimentaire notamment. Le camion-citerne peut avoir une cuve très différente suivant les produits qu'il transporte.

Le camion benne

Ce camion porteur est utilisé pour tous types de travaux publics ou de voirie. Suivant son utilisation, les équipements peuvent être très variés. Il peut être doté d'une à trois bennes ainsi que d'une grue auxiliaire qui peut être installée sur la cabine du camion. Il est couramment utilisé pour transporter du sable, de la terre, du gravier...

Le camion frigorifique

Très utilisé dans le domaine du transport de marchandises périssables, le camion frigorifique dispose d'une caisse isotherme munie d'un ou plusieurs groupes électrogènes permettant la production de froid. Ce camion permet le transport sur de longues distances des marchandises périssables.

Recherche du plus court chemin

Les chefs d'équipe organisent l'emploi du temps des chauffeurs et les trajets qu'ils effectueront sachant qu'ils cherchent à optimiser le temps de chacun des chauffeurs.

Vous avez à votre disposition le fichier ci-joint **Distances.csv** où il est fourni un ensemble de distances en km entre 2 villes. En fonction des intempéries, des difficultés routières, des travaux, les distances peuvent évoluer dans le temps.

Ce fichier peut évoluer dans le temps, des villes peuvent être ajoutées, des distances peuvent être modifiées en fonction des intempéries, des travaux Vous pouvez vous aider du lien <http://www.distance2villes.com/> pour compléter la liste des villes possibles.

- Créer les classes Nœud, Lien, et Graphe et instancier le graphe en utilisant les 2 modes (liste d'adjacence et matrice d'adjacence)
- Implémentez les deux algorithmes classiques de parcours d'un graphe à partir d'un sommet quelconque : Parcours en Largeur d'abord et Parcours en Profondeur d'abord. Affichez les sommets dans l'ordre de visite pour chaque méthode.
- Déterminez si le graphe est **connexe**.
- Vérifiez si le graphe contient des **circuits** (cycles) ? Donnez des exemples si possibles

Vous proposerez ensuite un outil de visualisation du graphe en C#

Pour l'outil de visualisation, vous pouvez utiliser les librairies

- System.Drawing de Microsoft
- ou d'autres librairies comme SkiaSharp.(using SkiaSharp;)

en vous aidant des outils d'IA générative, vous devrez être en mesure d'expliquer le fonctionnement de l'un ou l'autre des modes.

- **Dijkstra**
- **Bellman-Ford**
- **Floyd-Warshall** (référez-vous à la documentation comme [Wikipédia](https://fr.wikipedia.org/wiki/Algorithme_de_Floyd-Warshall) pour implémenter cet algorithme).
- **Remarque :**
 - Déterminez le chemin le plus court entre **deux sommets** du graphe de votre choix.
 - De préférence, choisissez deux sommets situés aux **extrémités du graphe** et pour lesquels **plusieurs chemins** existent.
- **Comparez les résultats des trois algorithmes**
 - Évaluez leurs performances en termes de : complexité algorithmique, rapidité d'exécution et pertinence pour ce cas d'application.
 - Justifiez vos réponses.

Enfin, vous adapterez les outils graphiques pour être en mesure de visualiser le meilleur chemin à parcourir en affichant toutes les villes parcourues ainsi que toutes les données de temps et ou de kms utilisés.

Mr Dupont souhaite donc une application qui lui permettra de gérer au mieux à la fois, ses salariés, ses clients et ses commandes. Vous proposerez une application C# qui sous forme d'un menu principal permettra de traiter les clients, les salariés et les commandes

Livrables

Le projet peut se faire à 2 ou 3 au sein d'un même groupe

Il doit rendre compte du cahier des charges ci-dessus. Tout information non écrite peut être interprétée comme vous le souhaitez.

Vous joindrez un rapport (maximum 4 pages) à votre solution C# dans lequel vous noterez vos ajouts si besoin et l'analyse comparatives des algorithmes de recherche du plus court chemin.

Au moment de la remise qui se tiendra au dernier TD, vous déposerez un dossier au nom des binômes ou trinômes.

Vous êtes autorisés pour la partie visualisation de vous aider d'une IA en revanche, il sera considéré comme plagiat si pour une autre partie du projet vous usez de ces outils.

Un code fourni qui ne compile ou ne s'exécute pas ne fait pas l'objet d'une correction et obtient la note de 0/20

Module Client

L'outil doit pouvoir permettre d'entrer, supprimer ou modifier un nouveau Client depuis la console ou depuis un fichier

Il faut à tout moment pouvoir afficher l'ensemble des Clients selon plusieurs critères : (successivement ou simultanément)

- Par ordre alphabétique
- Par ville
- Par montant des achats cumulés, ce qui permettra de connaître les meilleurs clients
- ...

Module Salarié

Pour faciliter la simulation, il vous est demandé de pouvoir afficher l'organigramme de manière efficace en vous basant sur une construction d'arbre n-aire (vu en TD)

Il faut, par ailleurs, pouvoir embaucher ou licencier un salarié et l'inclure ou l'exclure de l'organigramme.

Module Commande

Pour faciliter la simulation, il vous est demandé de pouvoir partir d'une situation initiale où aucune commande n'existe et au fil de l'utilisation de l'application sauvegarder les commandes effectuées afin de créer une base qui s'auto-enrichit.

Il faut donc pouvoir créer une nouvelle commande ou la modifier et simuler ses différentes étapes

Une commande ne peut être réalisée que si le client existe dans la base ou sinon il faut le créer. Une commande doit mettre en jeu un parcours entre une ville de départ et une ville d'arrivée et établir le **chemin le plus court** pour la date déterminée.

Il faut ensuite assigner un chauffeur évidemment disponible.

Les tarifs sont fonction du km parcouru et du tarif horaire du chauffeur qui peut varier en fonction de son ancienneté.

Il faut pouvoir à tout moment

- Calculer le prix d'une commande et l'afficher moyennant son numéro
- Déterminer le chemin que devra parcourir le chauffeur afin de lui donner son plan de route.

Module Statistiques

Il faut également faire des bilans généraux

- Afficher par chauffeur le nombre de livraisons effectuées
- Afficher les commandes selon une période de temps
- Afficher la moyenne des prix des commandes
- Afficher la moyenne des comptes clients

- Afficher la liste des commandes pour un client

Module Autre qui sera le résultat de votre créativité sur la gestion de cette société de transport.

- 4 suggestions sont attendues

Bonus

- Solutions Forms ou WPF
- Utilisation d'une BDD (Mysql)
- Marquage des villes sur une GoogleMap

Dans tous les cas, il faut utiliser tous les concepts vus depuis le début du semestre, vous pouvez évidemment vous inspirer des TDs :

- POO avec héritage bien-sûr,
- Classe abstraite,
- Interface (utilisation d'interface C# et création d'interface),
- Délégation (utilisation de délégation C#)
- Au moins 2 collections génériques différentes
- Un arbre n-aire
- Algorithme de Dijkstra, Bellman-Ford, Floyd-Warshall