Système de gestion de transport au Faculté



Par

**Mohamad CHALHOUB**

Sous-direction de

Dr **Ali JABER**

Sommaire

[Object du document : 1](#_Toc534903145)

[A - Introduction : 3](#_Toc534903146)

[B - Application : 3](#_Toc534903147)

[B.1 - S’identifier : 3](#_Toc534903148)

[B.2 - Création du compte : 3](#_Toc534903149)

[B.2.a - Préciser le propriétaire du compte : 4](#_Toc534903150)

[B.2.b - Entrer les informations personnelles (Etudiant) : 4](#_Toc534903151)

[B.2.c - Entrer son programme : 4](#_Toc534903152)

[B.2.d - Confirmation de l’enregistrement : 4](#_Toc534903153)

[B.2.e - Profil d e l’étudiant : 4](#_Toc534903154)

[B.2.f - Entrer les informations personnelles (Chauffeur) : 5](#_Toc534903155)

[B.2.g - Entrer les informations du bus : 5](#_Toc534903156)

[B.2.h - Confirmation de l’enregistrement : 6](#_Toc534903157)

[B.2.i - Profil du chauffeur : 6](#_Toc534903158)

[C - Use case : 7](#_Toc534903159)

[D - Diagramme d’activité : 7](#_Toc534903160)

[E - Base des données : 8](#_Toc534903161)

[E.1 - Modèle de donnée physique (Physical data model): 8](#_Toc534903162)

[E.2 - Code de la base des données : 8](#_Toc534903163)

[F - Diagram des classes: 10](#_Toc534903164)

# Object du document :

Ce site sert à résoudre les difficultés rencontrées par les étudiants à l’université au niveau de choisir le bus et connaitre l’horaire des bus et la gestion du processus du payement.

# Introduction :

La gestion traditionnelle des bus par les chauffeurs consiste à enregistrer sur une feuille les horaires de chaque étudiant chaque jour.

* Les fonctionnalités du site :
* Ce site est vu en deux perspective :

1. Le chauffeur (qui a 1 bus ou plusieurs).
2. L’étudiant de l’université.

* Organiser la manière de payement, par carte ou directe (si direct le payement doit être confirmer par le chauffeur), et déterminer le plan de payement mensuelle ou annuelle.
* Afficher le trajet de bus sur la carte (aller et retour) et localiser le bus à des temps définie.
* Localiser les maisons des étudiants sur la carte et les afficher pour le chauffeur affin de faciliter leurs collectes.
* Au départ de la collection des étudiants : afficher à chaque étudiant le temps approximer de l’arrive de bus.
* Au retour : afficher la location du bus pour chaque étudiant et afficher le temps de retour.
* Collection des programmes des étudiant (à quel heurs chaque étudiant fini) et les diviser en groups (pour organiser l’horaire de retours des étudiant qui finissent tôt et tard.
* L’heure de retour est modifiable (si l’étudiant fini tôt ou s’il a fini en retard) et notifier le chauffeur.
* Faire une méthode qui propose à chaque chauffeur de remplacer les étudiants entre eux (ex : Deux bus ont des élevés qui finit très tard au lieu que les deux bus attendent ils peuvent remplacer les étudiant entre eux d’une manière qu’un seul bus attend.
* Faire des contraint sur le bus (s’il vient en retard ou oublie un étudiant on débit du montant payer et en même temps si l’étudiant vient en retard on crédit au montant payer.
* Chaque nouveau étudiant doit être associe au bus le plus convenable.
* Groupement des étudiants qui fini au même temp pour aller dans le même bus.
* Détermination par le chauffeur le montant a payé pour les étudiants.

# Application :

## S’identifier :

Si l’utilisateur a déjà créé un compte il suffit de se connecter en entrant le **Username** et le **Password** et le site prend soin de reconnaitre si le compte appartient à un étudiant ou un chauffeur et ensuite ouvre le profil.

## Création du compte :

Si l’utilisateur n’a pas encore crée un compte il doit créer en appuyant sur le petit mot [**Sign up ?**](http://localhost:8080/Tp/Transport_final/choose.php) Au coin.

### Préciser le propriétaire du compte :

Une autre page s’ouvre pour préciser le propriétaire du compte s’il est un étudiant ou bien un chauffeur.

### Entrer les informations personnelles (Etudiant) :

S’il a choisi de s’enregistrer sous un compte d’étudiant il doit entrer ses informations personnelles. Il doit tout d’abord choisit un **User Name** c’est un mot unique qui identifie l’utilisateur puis entrer son nom **First Name** et le nom de sa famille **Last Name** et entrer un mot de passe **Password** puis le confirmer **Confirm Password** ensuite il doit entrer le numéro du téléphone **Phone Number** et enfin i doit choisir une **Location** des options disponible, Puis appuyer le bouton Next.

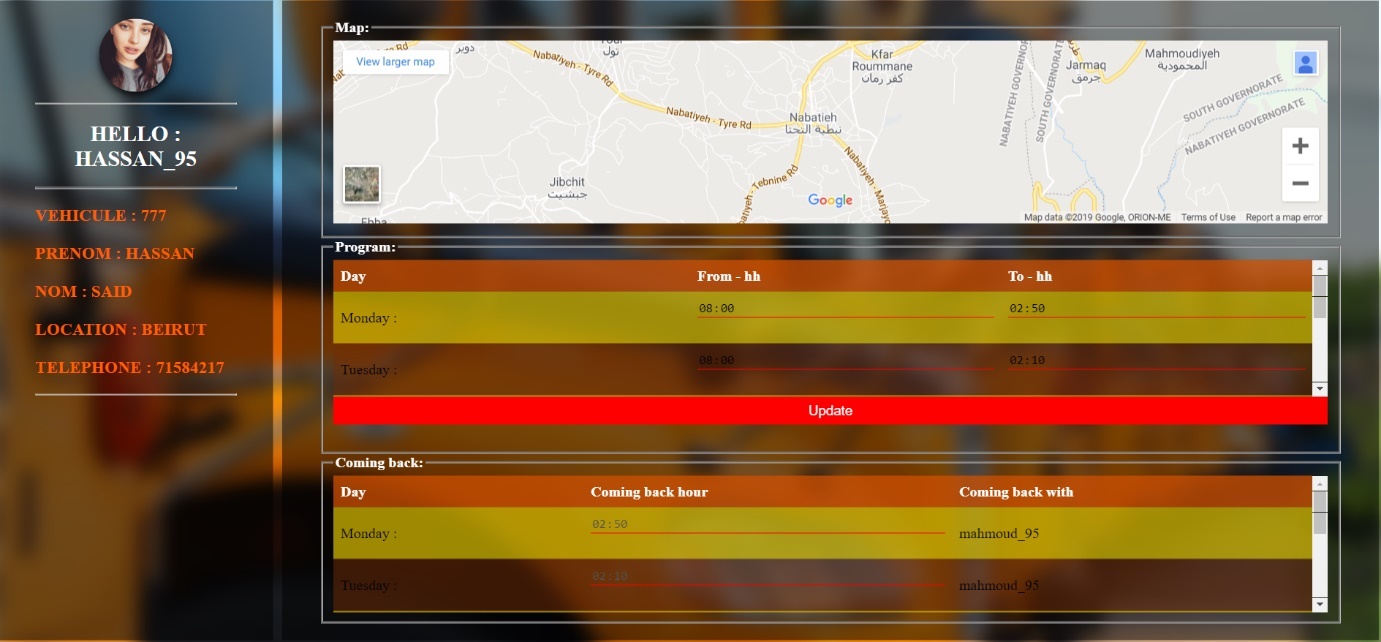
### Entrer son programme :

Après avoir remplir ses informations personnelles il doit entrer son programme hebdomadaire (de quelle heure à quel heure). Enfin il appuie sur OK.

### Confirmation de l’enregistrement :

Après avoir entré son programme une page de Confirmation de l’enregistrement apparait avec le **User Name** de l’utilisateur. Cette page assure que le processus de l’enregistrement est complet, Enfin on appuie sur Go to your profile.

### Profil d e l’étudiant :

Et Enfin le profil de l’étudiant apparait. A droit il voit ses informations personnelles et à gauche il voit la carte, et son programme et l’heure de retour et avec quel chauffeur, et il peut appuyer changer son programme et appuyer sur UPDATE pour le changer et il peut appuyer sur la photo du profil pour se déconnecter.

### Entrer les informations personnelles (Chauffeur) :

S’il a choisi de s’enregistrer sous un compte de chauffeur il doit entrer ses informations personnelles (STEP1). Il doit tout d’abord choisit un **User Name** c’est un mot unique qui identifie l’utilisateur puis entrer son nom **First Name** et le nom de sa famille **Last Name** et entrer un mot de passe **Password** puis le confirmer **Confirm Password** ensuite il doit entrer le numéro du téléphone **Phone Number** et le mode de payement **Payment Mode (PER MONTH ou PER YEAR)** et enfin le **Montant**.

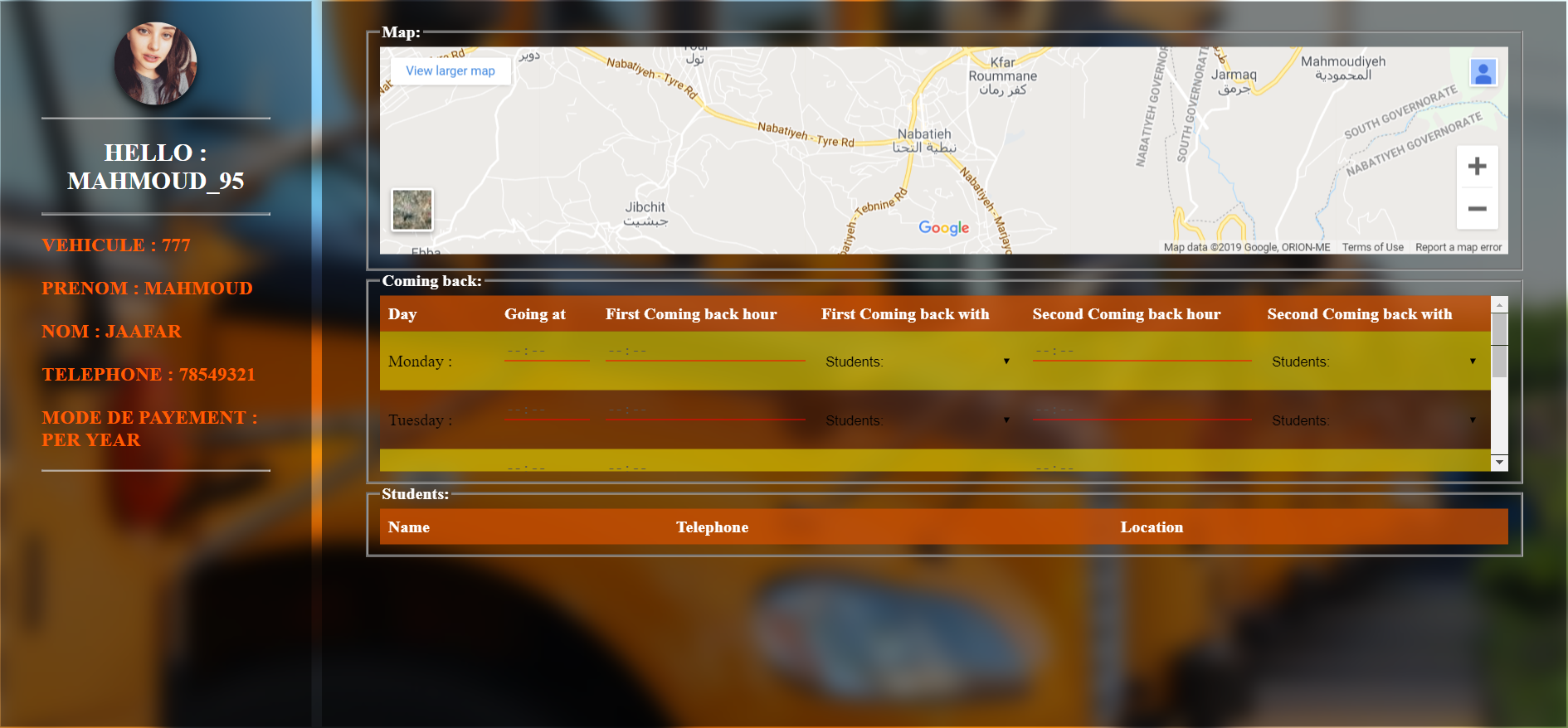
### Entrer les informations du bus :

Après avoir remplir ses informations personnelles il doit entrer les informations du bus qu’il possède le NUMERO, MODEL et LA CAPACITE DU VEHICULE et la LOCATION où il veut collecter les étudiant, il peut ajouter plusieurs locations en appuyant sur le bouton ADD et chacune des LOCATION sera ajouter dans le tableau en bas. Enfin il appuie sur OK.

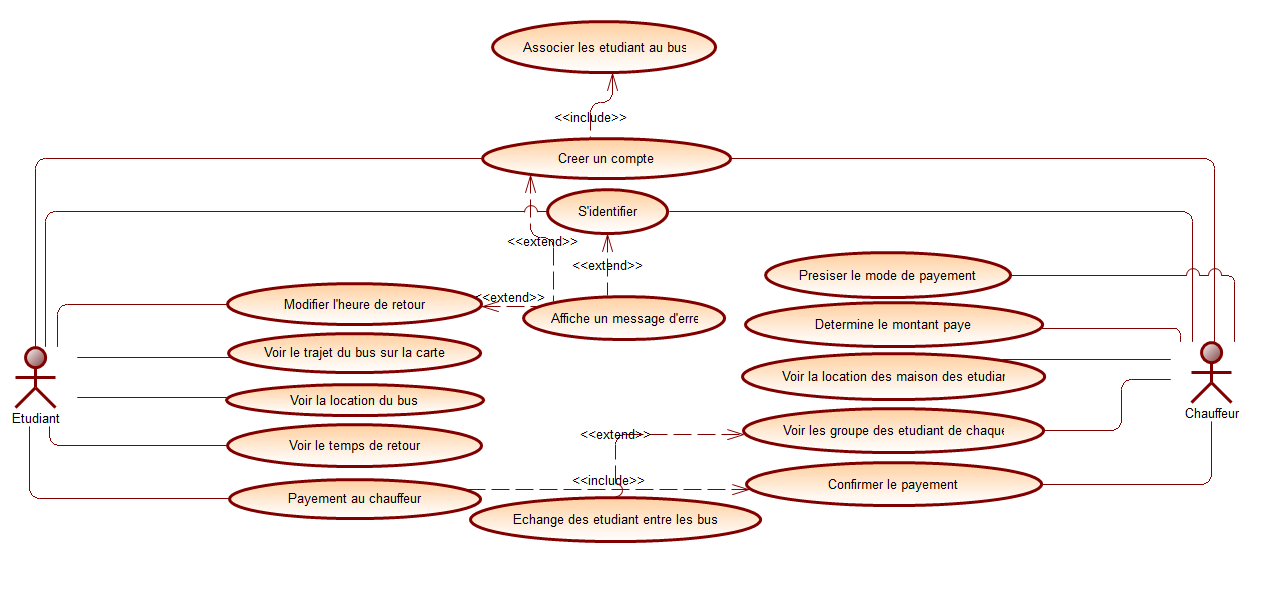
### Confirmation de l’enregistrement :

Après avoir entré tous les information du bus  une page de Confirmation de l’enregistrement apparait avec le **User Name** de l’utilisateur. Cette page assure que le processus de l’enregistrement est complet, Enfin on appuie sur Go to your profile.

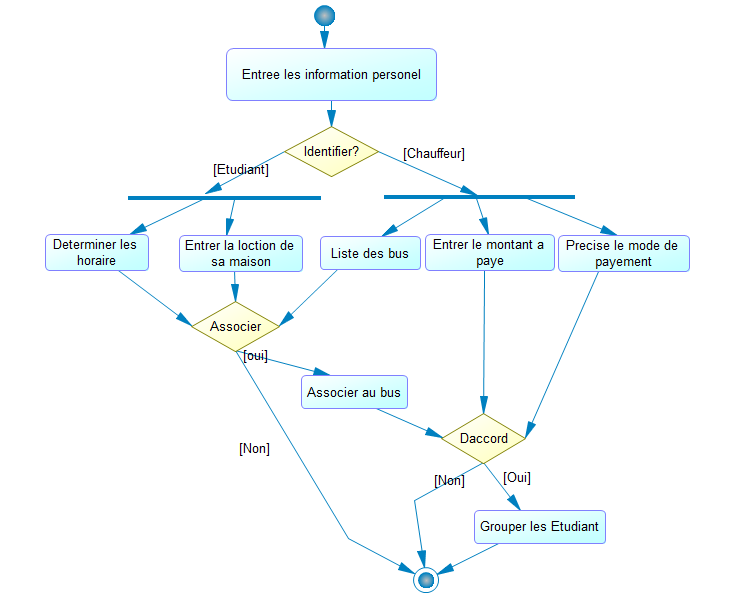
### Profil du chauffeur :

Et Enfin le profil du chauffeur apparait. A droit il voit ses informations personnelles et à gauche il voit la carte, et son programme et les étudiant enregistrer dans son bus, et il peut appuyer sur la photo du profil pour se déconnecter. 

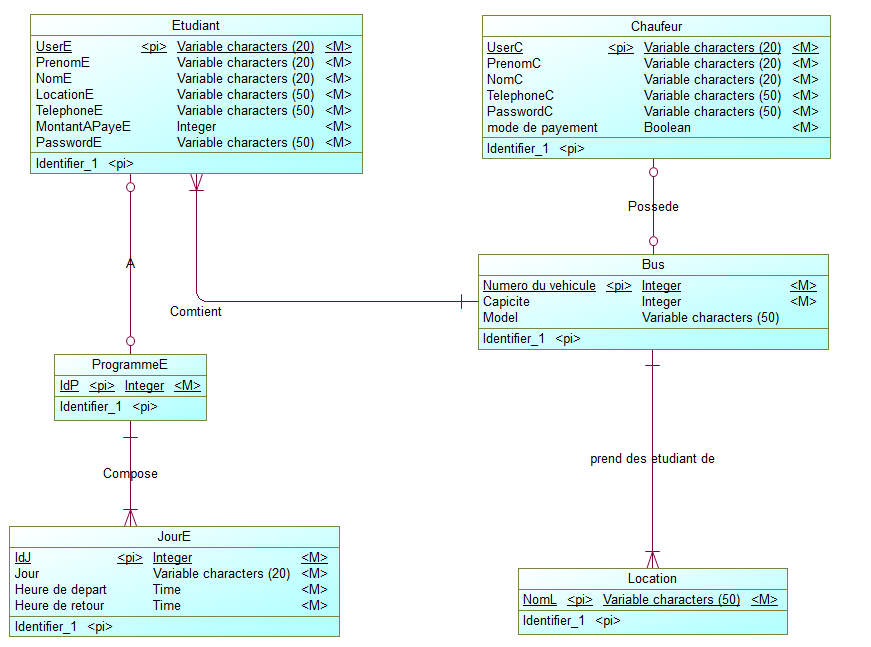
# Use case :



# Diagramme d’activité :

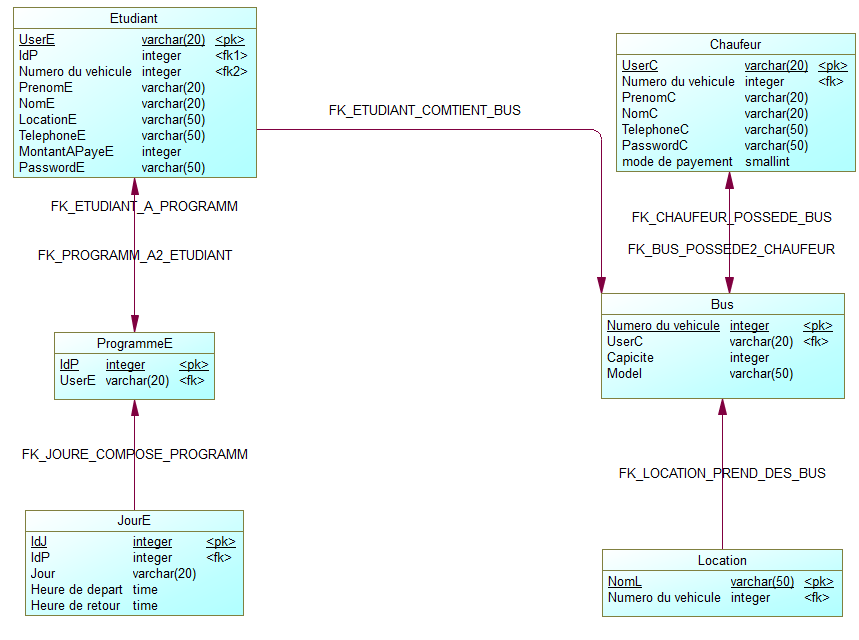


# Modèle entité association (ER Model) :



# Base des données :

## Modèle de donnée physique (Physical data model):



## Code de la base des données :

**create** **table** BUS

**(**

NUMERO\_DU\_VEHICULE **integer** **not** **null,**

USERC **varchar(**20**)** **null,**

IDP **integer** **null,**

CAPICITE **integer** **not** **null,**

MODEL **varchar(**50**)** **null,**

**constraint** PK\_BUS **primary** **key** clustered **(**NUMERO\_DU\_VEHICULE**)**

**);**

**create** **table** CHAUFEUR

**(**

USERC **varchar(**20**)** **not** **null,**

NUMERO\_DU\_VEHICULE **integer** **null,**

PRENOMC **varchar(**20**)** **not** **null,**

NOMC **varchar(**20**)** **not** **null,**

TELEPHONEC **varchar(**50**)** **not** **null,**

PASSWORDC **varchar(**50**)** **not** **null,**

MODE\_DE\_PAYEMENT **integer** **not** **null,**

**constraint** PK\_CHAUFEUR **primary** **key** clustered **(**USERC**)**

**);**

**create** **table** ETUDIANT

**(**

USERE **varchar(**20**)** **not** **null,**

IDP **integer** **null,**

NUMERO\_DU\_VEHICULE **integer** **not** **null,**

PRENOME **varchar(**20**)** **not** **null,**

NOME **varchar(**20**)** **not** **null,**

LOCATIONE **varchar(**50**)** **not** **null,**

TELEPHONEE **varchar(**50**)** **not** **null,**

MONTANTAPAYEE **integer** **not** **null,**

PASSWORDE **varchar(**50**)** **not** **null,**

**constraint** PK\_ETUDIANT **primary** **key** clustered **(**USERE**)**

**);**

**create** **table** JOURE

**(**

IDJ **integer** **not** **null,**

IDP **integer** **not** **null,**

HEURE\_DE\_DEPART **time** **not** **null,**

HEURE\_DE\_RETOUR **time** **not** **null,**

HEURE\_DE\_RETOUR1 **time** **null,**

**constraint** PK\_JOURE **primary** **key** clustered **(**IDJ**)**

**);**

**create** **table** LOCATION

**(**

NOML **varchar(**50**)** **not** **null,**

NUMERO\_DU\_VEHICULE **integer** **not** **null,**

**constraint** PK\_LOCATION **primary** **key** clustered **(**NOML**)**

**);**

**create** **table** PROGRAMMEE

**(**

IDP **integer** **not** **null,**

USERE **varchar(**20**)** **null,**

NUMERO\_DU\_VEHICULE **integer** **null,**

**constraint** PK\_PROGRAMMEE **primary** **key** clustered **(**IDP**)**

**);**

**alter** **table** BUS

**add** **constraint** FK\_BUS\_A1\_PROGRAMM **foreign** **key** **(**IDP**)**

**references** PROGRAMMEE **(**IDP**)**

**on** **update** **restrict**

**on** **delete** **restrict;**

**alter** **table** BUS

**add** **constraint** FK\_BUS\_POSSEDE2\_CHAUFEUR **foreign** **key** **(**USERC**)**

**references** CHAUFEUR **(**USERC**)**

**on** **update** **restrict**

**on** **delete** **restrict;**

**alter** **table** CHAUFEUR

**add** **constraint** FK\_CHAUFEUR\_POSSEDE\_BUS **foreign** **key** **(**NUMERO\_DU\_VEHICULE**)**

**references** BUS **(**NUMERO\_DU\_VEHICULE**)**

**on** **update** **restrict**

**on** **delete** **restrict;**

**alter** **table** ETUDIANT

**add** **constraint** FK\_ETUDIANT\_A\_PROGRAMM **foreign** **key** **(**IDP**)**

**references** PROGRAMMEE **(**IDP**)**

**on** **update** **restrict**

**on** **delete** **restrict;**

**alter** **table** ETUDIANT

**add** **constraint** FK\_ETUDIANT\_COMTIENT\_BUS **foreign** **key** **(**NUMERO\_DU\_VEHICULE**)**

**references** BUS **(**NUMERO\_DU\_VEHICULE**)**

**on** **update** **restrict**

**on** **delete** **restrict;**

**alter** **table** JOURE

**add** **constraint** FK\_JOURE\_COMPOSE\_PROGRAMM **foreign** **key** **(**IDP**)**

**references** PROGRAMMEE **(**IDP**)**

**on** **update** **restrict**

**on** **delete** **restrict;**

**alter** **table** LOCATION

**add** **constraint** FK\_LOCATION\_PREND\_DES\_BUS **foreign** **key** **(**NUMERO\_DU\_VEHICULE**)**

**references** BUS **(**NUMERO\_DU\_VEHICULE**)**

**on** **update** **restrict**

**on** **delete** **restrict;**

**alter** **table** PROGRAMMEE

**add** **constraint** FK\_PROGRAMM\_A2\_ETUDIANT **foreign** **key** **(**USERE**)**

**references** ETUDIANT **(**USERE**)**

**on** **update** **restrict**

**on** **delete** **restrict;**

**alter** **table** PROGRAMMEE

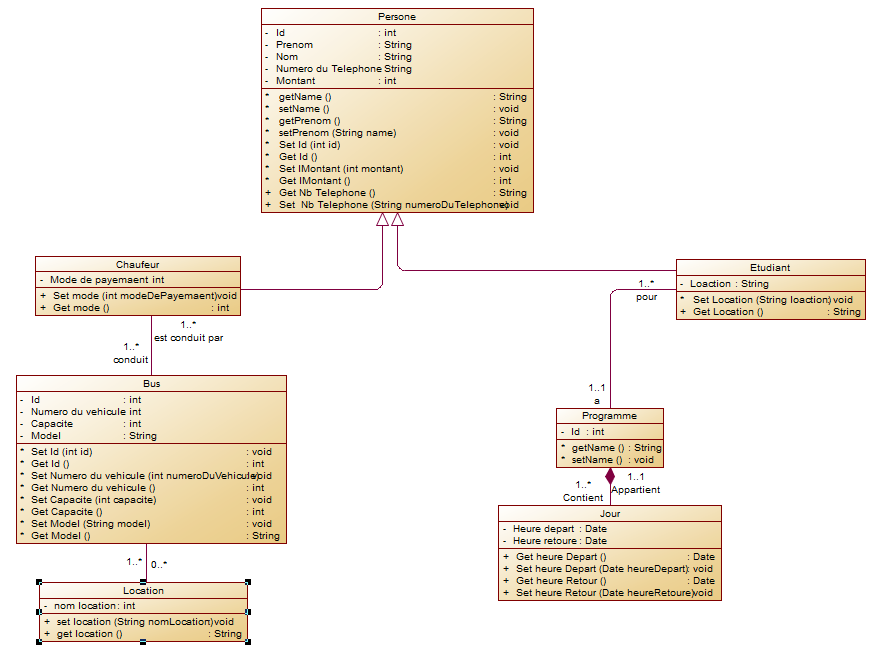
**add** **constraint** FK\_PROGRAMM\_A3\_BUS **foreign** **key** **(**NUMERO\_DU\_VEHICULE**)**

**references** BUS **(**NUMERO\_DU\_VEHICULE**)**

**on** **update** **restrict**

**on** **delete** **restrict;**

# Diagram des classes:



# Diagram de sequence: