

Analyse

NAVIGATION DES VEHICULES (PROJET GPS)

Equipe du projet:

WEN Han

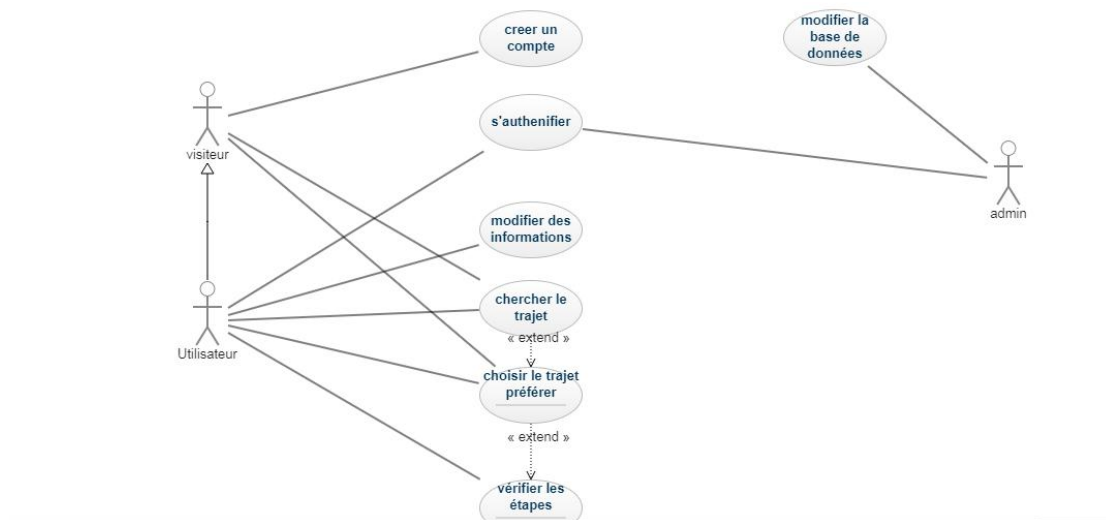
YANG Yu

Zhu Tingting

Sommaire

1. Cas d'utilisateur.....	3
2. Diagramme de classe.....	6
2.1 Utilisateur.....	6
2.2 Ville.....	8
2.3 Tronçon.....	9
2.4 Route.....	11
2.5 Trajet.....	12
2.6 Enumeration : TypeVille & TypeRoute.....	13
2.7 Les association en UML.....	13
3. Diagramme de sequence.....	15
3.1 Inscription.....	15
3.2 S'authentifier.....	16
3.3 Modifier informations personnel.....	17
3.4 Les attributs.....	18
3.5 Choisir les préférences.....	19
3.6 Vérifier les étapes.....	20
3.7 Finir le trajet.....	21
3.8 Générer la base de données.....	22
3.9 Générer l'histoire.....	23

1. Cas d'utilisateur



1. Les visiteurs doivent s'inscrire pour profiter de nombreuses fonctions, notamment leur nom, leur numéro de téléphone, leur adresse électronique et choisir leur sexe. Après avoir entré le nom d'utilisateur, le système vérifiera qu'il en est de même dans la base de données. Si le nom d'utilisateur existe déjà, vous devez le ressaisir. Entrez le mot de passe une fois la vérification terminée et ressaisissez-le pour confirmer.

2. Les visiteurs doivent s'inscrire pour profiter de nombreuses fonctions, notamment leur nom, leur numéro de téléphone, leur adresse électronique et choisir leur sexe. Après avoir entré le nom d'utilisateur, le système vérifiera qu'il en est de même dans la base de données. Si le nom d'utilisateur existe déjà, vous devez le ressaisir. Entrez le mot de passe

une fois la vérification terminée et ressaisissez-le pour confirmer.

3. Les visiteurs doivent s'inscrire pour profiter de nombreuses fonctions, notamment leur nom, leur numéro de téléphone, leur adresse électronique et choisir leur sexe. Après avoir entré le nom d'utilisateur, le système vérifiera qu'il en est de même dans la base de données. Si le nom d'utilisateur existe déjà, vous devez le ressaisir. Entrez le mot de passe une fois la vérification terminée et ressaisissez-le pour confirmer.

4. Les visiteurs peuvent également utiliser la fonction de navigation, mais ne peuvent pas modifier les informations personnelles, ne peuvent pas consulter l'historique et ne peuvent obtenir que l'intégralité de l'itinéraire, ne peuvent pas déterminer étape.

5. Lors de la connexion, la base de données déterminera automatiquement l'attribut du compte. Si l'utilisateur se connecte, la fonction utilisateur peut être utilisée. S'il s'agit d'un compte administrateur, la fonction administrateur est utilisée.

6. Les comptes d'utilisateur peuvent modifier leurs informations personnelles telles que leur nom, leurs préférences, leur numéro de téléphone et leur adresse électronique.

7. Les utilisateurs peuvent visualiser leurs propres enregistrements

historiques. Le système répertorie les 10 derniers enregistrements de voyage, par ordre chronologique de près ou de loin. L'utilisateur peut sélectionner l'enregistrement de voyage à supprimer ou à collecter, ou utiliser directement à nouveau l'enregistrement de voyage pour voyager à nouveau.

8. Lorsque les utilisateurs naviguent, ils doivent entrer leurs propres points de départ et de destination, le système calculera la distance et la durée de la demande, et répertoriera tous les itinéraires, en indiquant la distance la plus proche et la distance la plus rapide. L'utilisateur sélectionne l'itinéraire dont il a besoin. Après le démarrage de la navigation, le système indique la première étape de l'itinéraire et, lorsque l'utilisateur atteint la première étape, choisit de confirmer l'étape. Le système commencera la prochaine étape de la navigation. Si l'utilisateur est perdu, vous pouvez entrer à nouveau le point de départ et la destination et lancer la navigation.

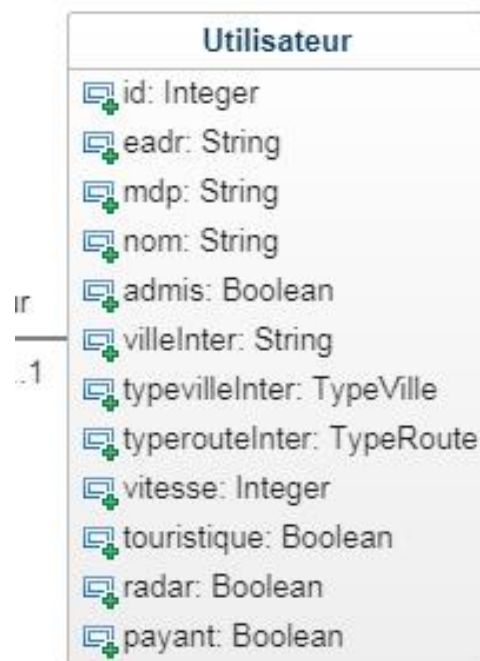
9. La connexion administrateur peut gérer les données de la base de données.

2. Diagramme de classe

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci.

2.1 Utilisateur

La classe de l'Utilisateur est pour enregistrer les informations du client , et le <id> est pour lier avec la classe < Trajet>.



Les attributs	
id	<ul style="list-style-type: none"> * De type primitif uniquement . * Considerer plusieurs clients ont même nom , donc quand l'utilisateur s'inscrit , ce logiciel va lui distribuer

	un id automatiquement . Et chaque utilisateur est identifié par l'id pour éviter de se tromper les informations personnelles .
eadr	* De type simp (String) * L'adresse email de l'utilisateur pour se connecter , consulter et gérer ses informations personnelles .
mdp	* De type simp(String) * Le mot de passe pour se connecter , consulter et gérer ses informations personnelles .
nom	* De type simple (String) * Le nom de l'utilisateur
admis	* De type simple (Boolean) *
villeInter	* De type simple (String) * Les villes intermédiaires à éviter ou à suivre
typevilleInter	* De type simple (TypeVille) * Le type de villes intermédiaires à éviter ou à suivre
typerouteInter	* De type simple (TypeRoute) * Le type de routes intermédiaires à éviter ou à suivre
vitesse	* De type simple (Integer) * La max vitess de routes à suivre

touristique	<ul style="list-style-type: none"> * De type simple (Boolean) * La max vitess de routes à suivre
radar	<ul style="list-style-type: none"> * De type simple (Boolean) * Les routes à suivre avec radar ou sans radar
payant	<ul style="list-style-type: none"> * De type simple (Boolean) * Les routes à suivre sont payant ou pas

2.2 Ville

La classe de Ville est pour spécifier ses caractères, le <cp> est code postal pour distinguer les villes de même nom et lier avec la classe <Tronçon> .

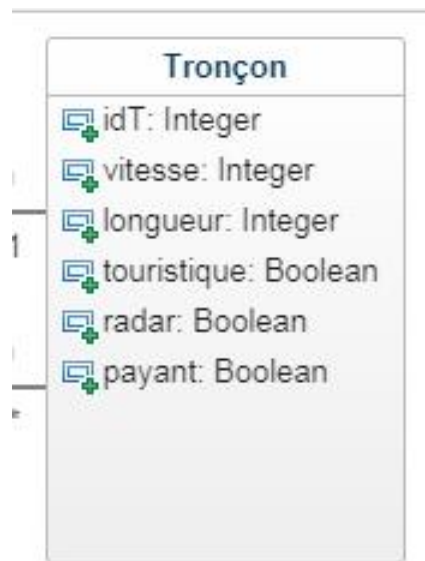


Les attributs	
cp	<ul style="list-style-type: none"> * De type primitif uniquement . * Considerer il existe plusieurs villes ont même

	nom , donc on distinguer les villes par son code postal .
nom	* De type simple (String) * Le nom de la ville , utilisé par l'utilisateur pour le sélectionner
typeVille	* De type simple (TypeVille) * Pour indiquer le type de cette ville(parmi départementale, nationale, autoroute, chemin de campagne)
touristique	* De type simple (Boolean) * Pour indiquer cette ville est touristique ou pas
latitude	* De type simple (Integer) * La position précise de la ville
longitude	* De type simple (Integer) * La position précise de la ville

2.3 Tronçon

La classe <Tronçon> est pour composer la route, il relie deux villes et a des caractère spéciaux. Le <idT> pour identifier chaque tronçon et pour lier avec la classe <Route> et la classe <Utilisateur> .



Les attributs	
idT	<ul style="list-style-type: none"> * De type primitif uniquement * Pour faciliter de gérer les tronçon , chaque tronçon a son <idT> uniquement .
vitesse	<ul style="list-style-type: none"> * De type simple (Integer) * Fixer la max vitesse du tronçon
longueur	<ul style="list-style-type: none"> * De type simple (Integer) * La taille du tronçon pour trouver le plus court chemin
Touristique	<ul style="list-style-type: none"> * De type simple (Boolean) * Pour indiquer ce tronçon est touristique ou pas
radar	<ul style="list-style-type: none"> * De type simple (Boolean) * Pour indiquer ce tronçon avec radar ou sans radar

payant	<ul style="list-style-type: none"> * De type simple (Boolean) * Pour indiquer ce tronçon est payant ou pas
--------	--

2.4 Route

L'utilisateur choisi des routes à suivre ou à éviter par son nom , et le <idR> pour identifier chaque route et pour lier avec la classe <Tronçon> et la classe <Trajet>



Les attributs	
idR	<ul style="list-style-type: none"> * De type primitif uniquement * Pour faciliter de gérer les routes , chaque route a son <idR> uniquement
nom	<ul style="list-style-type: none"> * De type simple (String) * Le nom de la route
typeRoute	<ul style="list-style-type: none"> * De type simple (TypeRoute) * Pour indiquer le type de cette route (parmi

	départementale, nationale, autoroute, chemin de campagne)
--	---

2.5 Trajet

Après l'utilisateur bien choisi la ville départ , la ville destination et les préférences, ce logiciel va proposer 3 trajets(un plus court , un plus rapide et un selon les choix de l'utilisateur). Chaque trajet nécessite un identifiant unique, donc chaque trajet est identifié par <idTra>.



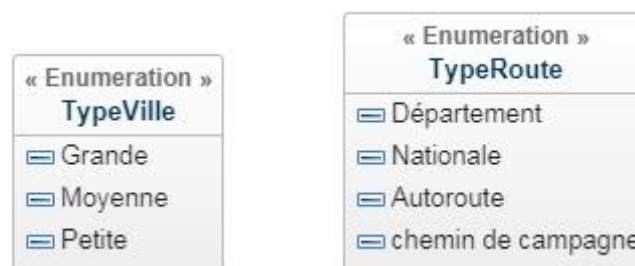
Les attributs	
idTra	<ul style="list-style-type: none"> * De type primitif uniquement * Pour faciliter de gérer les trajets , chaque trajet a son <idTra> uniquement
depart	<ul style="list-style-type: none"> * De type simple (String) * La ville de départ , tapé par l'utilisateur
destination	<ul style="list-style-type: none"> * De type simple (String) * La ville de destination , tapé par l'utilisateur

duree	<ul style="list-style-type: none"> * De type simple (Integer) * Le temps calculé pour aller de la ville de départ à la ville de destination
tempsDebut	<ul style="list-style-type: none"> * De type simple (DataType) * L'heure de début de ce trajet

2.6 Enumeration : TypeVille & TypeRoute

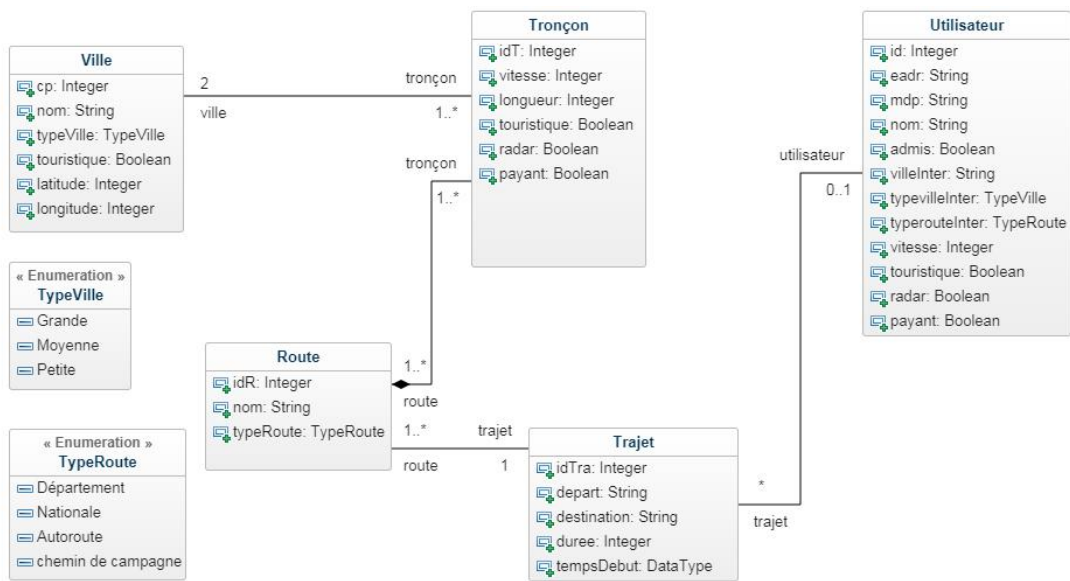
Pour spécifier les types de ville et les types de route , on énumérer les types .

Les villes ont type de Grand, type de Moyenne et type de Petite. Les routes ont type de Departement , type de Nationale, type d'Autoroute et type de chemin de campagne.



2.7 Les association en UML

Les liens avec les autres classes d'intérêt sont représentés par les associations.



Pour v: Ville ,

v.tronçon correspond à une collection d'instance de Tronçon.

Pour t: Tronçon ,

t.ville correspond à une collection de deux Ville

t.toute correspond à une collection d'instance de Route

Pour r: Route ,

r.tronçon correspond à une collection d'instance de Tronçon

r.trajet correspond à une instance de Trajet

Pour tra: Trajet ,

tra.route correspond à une collection d'instance de Route

Tra.utilisateur correspond à une instance de 0 ou 1 Utilisateur

Pour u: Utilisateur ,

u.trajet correspond à une collection d'instance d'utilisateur

3. Diagramme de sequence

3.1 Inscription

CU : Inscription
ID : 1
Description brève : Visiteur saisit son information et inscrit dans le système
Acteurs primaires : visiteur
Acteurs secondaires : Aucun
Préconditions : n'est pas utilisateur
Enchaînement Principal 1. Le visiteur saisit les informations , nom d'utilisateur et mot de passe. 2. Les œuvres correspondant à la recherche sont affichées. 3. Si le nom d'utilisateur n'existe pas , le système enregistre les informations et rend une message "ok", si l'utilisateur existant déjà , rendre une message "changez le nom d'utilisateur".
Post-condition :
Enchainements alternatifs :

3.2 S'authentifier

CU :S'authentifier
ID : 2
Description brève : Permet à l'utilisateur de s'authentifier et accéder son compte
Acteurs primaires : visiteur
Acteurs secondaires : Aucun
Préconditions : <ul style="list-style-type: none"> Le login et le mot de passe doivent être corrects.
Enchaînement Principal : <ol style="list-style-type: none"> Le visiteur s'authentifie son compte Le visiteur saisit le nom d'utilisateur et mot de passe Si les informations sont invalides. <ol style="list-style-type: none"> Afficher un message d'erreur Aller à l'étape suivant . Le système affiche « Le login ou le mot de passe n'est pas correct ! » Si les informations sont corrects ,accéder son compte
Post-condition :

3.3 Modifier informations personnel

CU :ModifierInformationPersonnel
ID : 3
Description brève : Permet a l'utilisateur de modifier les informations personnelles
Acteurs primaires : utilisateur
Acteurs secondaires : Aucun
Préconditions : <ul style="list-style-type: none"> • L'utilisateur doit s'authentifier.
Enchaînement Principal : <ol style="list-style-type: none"> 1. L'utilisateur vérifie les nouvelles informations personnelles. 2. L'utilisateur modifie les informations relatives un compte déjà existant.
Post-condition :
Enchainement alternatifs :

3.4 Les attributs

CU : Les attributs
ID : 4
Description brève : Permet a l'utilisateur de chercher son trajet
Acteurs primaires : Les attributs
Acteurs secondaires : Aucun
Préconditions : <ul style="list-style-type: none"> • L'utilisateur doit s'authentifier. • L'utilisateur doit saisir la ville debut et la ville fini
Enchaînement Principal : <ol style="list-style-type: none"> 1. L'utilisateur saisit la ville debut et la ville fini 2. le système rend les informations des trajets <ol style="list-style-type: none"> 2.1. le système calcule les temps boson 2.2. afficher les trajets avec les commentaire (plus court, moins de temps)
Post-condition :
Enchainement alternatifs :

3.5 Choisir les préférences

CU :ChoisirPreference
ID : 5
Description brève : Permet a l'utilisateur de choisir sont preference
Acteurs primaires : Utilisateur
Acteurs secondaires : Aucun
Préconditions : <ul style="list-style-type: none"> • Les informations sont deja rendu par le système
Enchaînement Principal : <ol style="list-style-type: none"> 1. Le système affiche plusieurs choix comme le trajet plus court, moins de temps 2. L'utilisateur choisit son preference
Post-condition :
Enchainement alternatifs :

3.6 Vérifier les étapes

CU : VerifierEtape
ID : 6
Description brève : Permet a l'utilisateur de verifier son étape et commence la nouvelle étape
Acteurs primaires : Utilisateur
Acteurs secondaires : Aucun
Préconditions <ul style="list-style-type: none"> • L'utilisateur a commencé l'étape déjà .
Enchaînement Principal : <ol style="list-style-type: none"> 1. L'utilisateur clique verifier l'étape. Si l'utilisateur est dans la ville correcte affiche l'étape suivant Si l'utilisateur est perdu, rendre une nouvelle étape
Post-condition :
Enchainement alternatifs :

3.7 Finir le trajet

CU : FiniTrajet
ID : 7
Description brève : Permet de l'utilisateur de finir son trajet
Acteurs primaires : Utilisateur
Acteurs secondaires : Aucun
Préconditions <ul style="list-style-type: none"> • L'utilisateur a commencé l'étape déjà .
Enchaînement Principal : <ol style="list-style-type: none"> 1. L'utilisateur clique la bouton "fini" 2. Enregistrer se trajet dans l'historie 3. Retour a main page
Post-condition :
Enchainement alternatifs :

3.8 Générer la base de données

CU : GenererBasedonnee
ID : 8
Description brève : Permet a l'administrateur génère la base des données
Acteurs primaires : Administrateur
Acteurs secondaires : Aucun
Préconditions <ul style="list-style-type: none"> • L'attribut de son compte de "Administrateur" est vrais.
Enchaînement Principal : <ol style="list-style-type: none"> 1. Générer la base des données comme les informations de cartes, de tronçon ,de ville etc.
Post-condition :
Enchaînement alternatifs :

3.9 Générer l'histoire

CU : GenerHistoire
ID : 9
Description brève : Permet a l'utilisateur de générer son histoire.
Acteurs primaires : Utilisateur
Acteurs secondaires : Aucun
Préconditions : Utilisateur a deja s'authentifie
Enchaînement Principal <ol style="list-style-type: none"> 1. L'utilisateur clique le bouton "histoire" 2. le système affiche les histoires 3. Utilisateur peut supprimer ou accéder a l'histoire qui est choisi
Post-condition :
Enchainements alternatifs :