

Conception

NAVIGATION DES VEHICULES (PROJET GPS)

Equipe du projet:

Zhu Tingting

WEN Han

YANG Yu

目录

1. Introduction.....	1
1.1 Objectif.....	1
1.2 Portée du projet.....	2
1.3 Vue d'ensemble.....	2
2. Cas d'utilisation.....	3
2.1 Inscription.....	4
2.2 Login.....	5
2.3 Oublier le mot de passe.....	7
2.4 Modifier le mot de passe.....	8
2.5 Modifier les informations personnelles.....	10
2.6 Choisir les préférences.....	12
2.7 Chercher le trajet.....	15
2.8 Verifier les étapes	17
2.9 Finir la simulation.....	18
2.10 Déconnection	19
2.11 Gestion la base de données de map.....	20
3. Digramme de séquence.....	21
3.1 Inscription.....	21
3.2 Login.....	22
3.3 Oublier le mot de passe.....	22
3.4 Modifier le mot de pase.....	23
3.5 Modifier les informations personnelles.....	24
3.6 Choisir les préférences.....	25
3.7 Chercher le trajet.....	26
3.8 Finir la simulation.....	28
3.9 Déconnexion.....	28
3.10 Gestion la base de données	29
4. Digramme de clase.....	30
4.1 Exigences fonctionnelles.....	31
4. 1. 1. Classe de <Utilisateur>.....	31
4. 1. 2. Classe de <Ville>.....	37
4. 1. 3. Classe de <Tronçon >.....	38
4. 1. 4. Classe de <Route>.....	39
4. 1. 5. Classe de <Trajet>.....	40
4. 2. Exigence spécifique.....	44
4. 2. 1. Contrainte de chercher un plus court trajet.....	44
4. 2. 2. Algorithme de A Etoile.....	45
4. 2. 3. Pseudocode de chercherTrajet (Aetoile)	48

1. Introduction

GPS-Like est la simulation d'un système de navigation, qui joue un rôle indispensable dans le processus de conduite du conducteur, car elle permet aux utilisateurs de disposer de plusieurs itinéraires rapidement et efficacement, et les utilisateurs peuvent même définir leurs propres préférences d'itinéraire (telles que: passer pas par un tronçon avec un radar, en passant par une ville touristique, etc.)

Ce système possède 2 côtés, le côté administration permet aux administrateurs de gérer les informations de la carte, autre côté permet aux utilisateurs (les clients) d'effectuer une navigation avec les préférences qu'ils choisissent.

1.1 Objectif

Avec le développement de la technologie, les gens peuvent se rendre de plus en plus en voiture, mais la route entre les villes est de plus en plus compliquée, le conducteur ne peut pas déterminer comment atteindre la destination ni déterminer lequel est le plus court chemin. Afin d'améliorer la lisibilité de cette conception, nous analyserons les exigences d'utilisation de l'utilisateur du point de vue de l'utilisateur, ainsi que la manière de mettre en œuvre les besoins de

l'utilisateur en code, du point de vue du développeur du logiciel. Cette conception est à la base de l'implémentation du code et du logiciel de test final pour répondre aux attentes.

1.2 Portée du projet

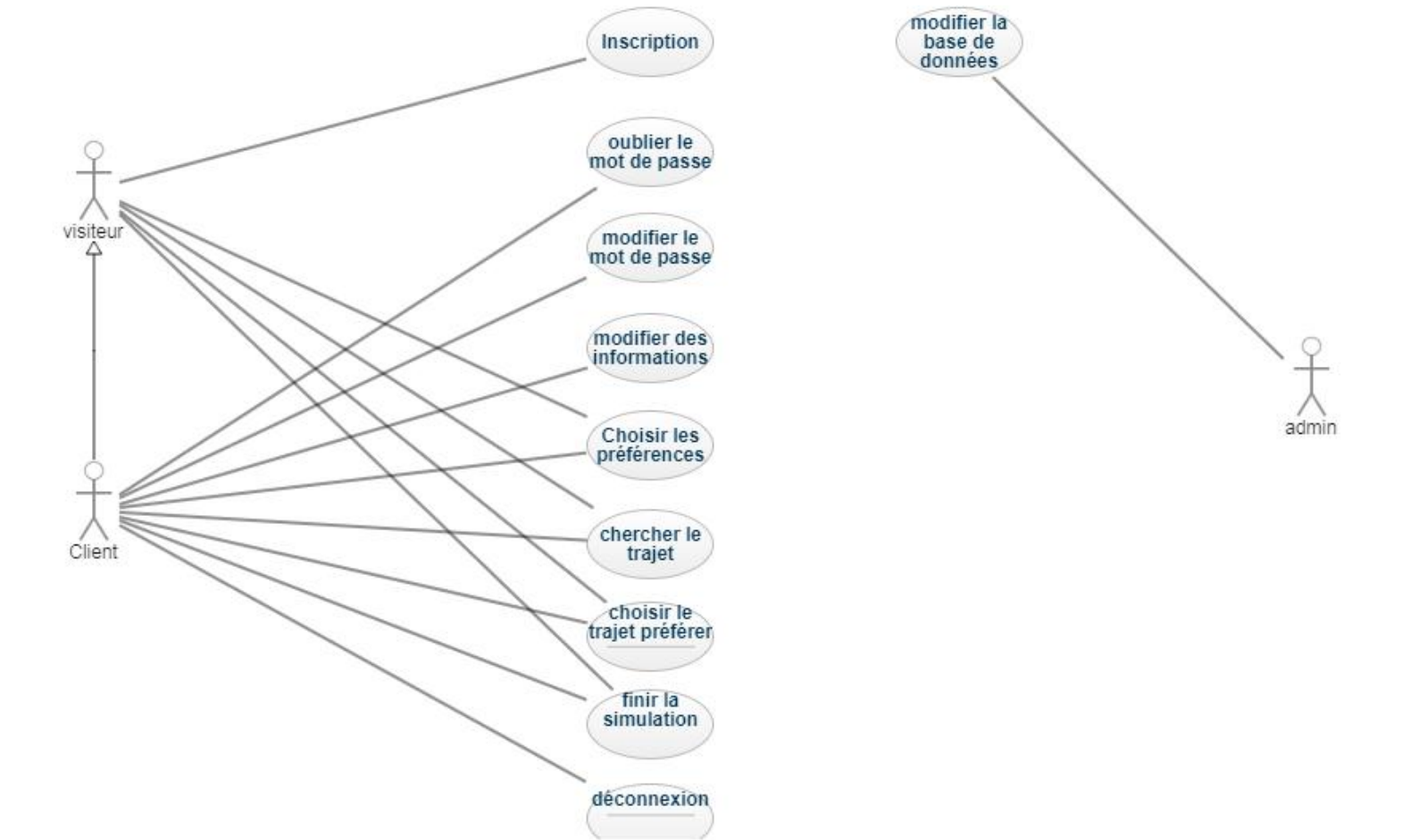
Ce logiciel permettra la navigation dont les gens ont besoin tous les jours. Il est principalement divisé en deux interfaces principales, l'une est l'interface utilisateur et l'autre, l'interface administrateur.

- L'interface utilisateur peut uniquement rechercher des chemins et modifier ou consulter des informations personnelles et des préférences.
- L'interface administrateur peut modifier diverses informations de la carte(tronçon ,route ,ville).

1.3 Vue d'ensemble

Cette conception est organisée en trois parties, la première est la description de cas d'utilisation , deuxième section, la diagramme de sequence, et la troisième partie est le diagramme de classe et les fonctions.

2. Cas d'utilisation



2.1 Inscription

Ce n'est que lorsqu'un nouvel utilisateur peut enregistrer un nouveau compte (c'est-à-dire qu'un adresse email ne peut enregistrer qu'un seul compte), afin de s'assurer que les informations de la base de données entrante sont exactes, il faut confirmer les informations par les points suivants.

1. Lorsque l'utilisateur clique sur le bouton "Enregistrer", il entre dans "l'interface d'enregistrement". L'interface d'enregistrement comporte quatre zones de texte: Nom d'utilisateur, Mot de passe de messagerie, Confirmer le mot de passe. Vous devez vous assurer que chaque zone de texte n'est pas vide pour cliquer sur le bouton "confirmer".
2. Lorsque l'utilisateur clique sur le bouton "confirmer", le logiciel va confirmer si le nom d'utilisateur entré par l'utilisateur est légal et si l'adresse email est conforme au format de celle-ci. Si le format de l'adresse email entré par l'utilisateur n'est pas valide, l'interface affichera "Erreur d'adresse email saisie, veuillez la saisir à nouveau après confirmation." Le mot de passe du compte étant défini par l'utilisateur lui-même, nous avons ajouté le champ de saisie «Confirmer le mot de passe» afin d'éviter que celui-ci ne le

saisisse en raison d'un problème de saisie. Lorsque les mots de passe saisi par l'utilisateur ne sont pas le même, l'interface affichera «Les mots de passe saisis sont différent, tapez-les encore une fois après la confirmation , s'il vous plaît ». " Lorsque l'adresse email entrée par l'utilisateur correspond les conditions requises et que les deux mots de passe sont identiques, le logiciel recherchera si l'adresse email existe déjà dans la base de données ou pas. Si elle existe déjà, le message "L'adresse email a été enregistrée, veuillez entrer à nouveau, s'il vous plaît ".

Ce n'est que lorsque l'utilisateur répond aux les deux détections ci-dessus que vous pouvez enregistrer un nouveau compte, le logiciel génère l'identifiant de l'utilisateur en arrière-plan et stocke les informations utilisateur obtenues et l'identifiant de l'utilisateur dans la base de données.

2.2 Login

Lorsque l'utilisateur a déjà eu un compte, il est nécessaire de s'assurer que l'utilisateur est connecté ou que l'utilisateur est autorisé à se connecter au compte afin d'éviter tout vol de ses informations (sécurité). Nous devons donc vérifier si la connexion de l'utilisateur est correcte en suivant les étapes ci-dessous.

1. Lorsque l'utilisateur non connecté clique sur le bouton «Connexion» de l'interface principale, l'interface de «Connexion» s'affiche. L'interface de connexion a principalement deux zones de texte: l'adresse email et le mot de passe. Lorsque l'utilisateur entre son adresse email et son mot de passe, cliquez sur le bouton «confirmer». (Si l'une des zones de texte est vide, l'utilisateur ne peut pas cliquer sur le bouton «confirmer»).
2. Lorsque l'utilisateur clique sur le bouton «confirmer», le logiciel associe l'adresse email saisie par l'utilisateur à la "adresse email" enregistrée dans la base de données. Si les informations de compte correspondantes ne sont pas trouvées, l'interface affichera "Cet utilisateur n'existe pas". Si l'adresse email correspondante est mise en correspondance, il est déterminé si le mot de passe saisi par l'utilisateur est cohérent avec le mot de passe de la base de données. Si les informations sont cohérents, la connexion est réussie. Si elles sont incompatibles, entrez le troisième jugement.
3. Afin de garantir que le compte de l'utilisateur ne soit pas connecté de manière malveillante par d'autres personnes, si le même compte entre continuellement le mot de passe incorrect plus de trois fois, le logiciel envoie automatiquement un code de vérification à l'adresse email enregistrée. En plus, chaque fois avoir saisi un mot de passe incorrect,

l'interface affichera "Le mot de passe saisi est incorrect et il vous reste x chance. Si vous entrez plus de trois fois, votre courrier électronique recevra un code de vérification". Si l'opération est fait par l'utilisateur soi-même , il peut entrer le code de vérification reçu par l'adresse email pour entrer dans l'interface permettant de modifier le mot de passe. Après avoir modifié le mot de passe, l'utilisateur retourne à l'interface de connexion pour se reconnecter.

L'utilisateur peut consulter les informations d'utilisateur et l'historique de recherche historique, il peut aussi modifier des informations personnelles uniquement après que l'utilisateur s'est connecté.

2.3 Oublier le mot de passe

1. Étant donné que presque tous les logiciels nécessitent désormais un compte de connexion, l'utilisateur peut confondre le mot de passe du compte ou oublier le mot de passe. Afin de permettre aux utilisateurs de se connecter à tout moment, les étapes suivant permettant à l'utilisateur de récupérer le mot de passe sont plus concises afin d'assurer la fiabilité. Par conséquent, nous avons ajouté le bouton "Mot de passe oublié" sous la zone de texte de l'interface "Connexion". Lorsque l'utilisateur clique sur le bouton "Mot de passe oublié", il peut passer à l'interface "Mot de passe

oublié". Une fois que l'utilisateur a saisi l'interface «Mot de passe oublié», il vous suffit de suivre les étapes ci-dessous pour terminer rapidement l'opération de récupération du mot de passe et vous connecter.

1. Dans l'interface “Mot de passe oublié”, il y a une zone de texte.

L'utilisateur entre l'adresse email dans la zone de texte et clique sur le bouton "confirmer". Le logiciel recherchera dans la base de données une adresse email correspondant à la saisie de l'utilisateur. Si l'adresse email n'est pas trouvée, l'interface affichera "L'adresse email entrée est incorrecte, veuillez la saisir à nouveau après avoir vérifié ". Sinon, le logiciel enverra automatiquement un code de vérification aléatoire à l'adresse email saisie par l'utilisateur. Après avoir saisi le code de vérification reçu sur l'interface, l'utilisateur peut entrer dans l'interface pour modifier le mot de passe. Après la modification du mot de passe, l'utilisateur passe automatiquement à l'interface de connexion et peut se reconnecter avec le nouveau mot de passe.

2.4 Modifier le mot de passe

Il y a un bouton “Change le mot de passe” dans l'interface des paramètres. Lorsque l'utilisateur constate que le mot de passe

actuellement utilisé n'est pas sécurisé, l'utilisateur peut cliquer sur le bouton "Changer le mot de passe" dans l'interface de paramétrage pour accéder à l'interface "Modifier le mot de passe". La page Modifier le mot de passe comporte trois zones de texte: Mot de passe actuel, Nouveau mot de passe et Confirmer le mot de passe à nouveau. Afin de confirmer le mot de passe de modification utilisé par l'utilisateur soi-même, il faut confirmer les étapes suivantes.

2. L'utilisateur doit remplir les trois zones de texte de l'interface. Lorsqu'une des zones de texte est vide, l'utilisateur ne peut pas cliquer sur le bouton "confirmer la modification".
3. Une fois que l'utilisateur a renseigné les trois zones de texte, le logiciel va rechercher le mot de passe correspond de cet utilisateur dans la base de données. Si le mot de passe actuel entré par l'utilisateur ne correspond pas à celui de la base de données, l'interface affichera "Le mot de passe entré est incorrect. Entrez encore une fois après la confirmation , s'il vous plaît." Si le mot de passe actuel entré par l'utilisateur correspond au mot de passe de la base de données, continue à l'étape suivante.
4. Lorsque le logiciel confirme que le mot de passe actuel est correct, il vérifie les deux nouveaux mots de passe entrés. Si le nouveau mot de passe entré par l'utilisateur ne correspond pas aux

spécifications, l'interface affiche "Le nouveau mot de passe entré contient des caractères spéciaux, veuillez vérifier et ressaisir le nouveau mot de passe, s'il vous plaît ". Si le nouveau mot de passe entré par l'utilisateur est en bonne forme, le logiciel va comparer les deux entrées "nouveau mot de passe" et "re-confirmer le mot de passe". Si les deux sont incohérentes, l'interface affiche "Les mots de passe entrés deux fois sont incohérents, veuillez les vérifier et les entrer à nouveau, s'il vous plaît ." Si les deux sont cohérents, l'utilisateur modifie avec succès le mot de passe. il va passer automatiquement à l'interface de connexion et l'utilisateur peut se connecter sur cet interface .

2.5 Modifier les informations personnelles

Chaque compte peut stocker les informations personnelles de l'utilisateur: nom d'utilisateur, préférence par défaut, positions plus utilisés, etc. Lorsque l'utilisateur est en état de connexion, il peut accéder à l'interface d'informations personnelles en cliquant sur le bouton "Paramètres" sur l'interface principale. Dans l'interface d'informations personnelles, l'utilisateur peut modifier le nom d'utilisateur et modifier les préférences etc.

1. Lorsque l'utilisateur entre un nouveau nom d'utilisateur dans la zone "Nouveau nom d'utilisateur", le logiciel doit vérifier si les

caractères entrés par l'utilisateur sont illégaux. S'il y a des caractères illégaux, l'interface affiche "Le nom d'utilisateur que vous avez entré ne répond pas aux exigences, veuillez confirmer à nouveau et entrer, s'il vous plaît ."

2. Chaque utilisateur peut définir ses propres préférences par défaut. Les paramètres préférés sont les mêmes que ci-dessous(1.6 Choisir les préférences).
3. Afin de rendre l'opération plus facile et plus rapide, l'utilisateur peut définir les raccourcis par défaut du système pour enregistrer Les emplacements que la plupart des utilisateurs utilisent souvent. Par exemple, "home" et "company", l'utilisateur peut également nommer une touche de raccourci pour enregistrer les autres emplacements fréquents de l'utilisateur, l'utilisateur peut également supprimer une touche de raccourci auto-nommée dans l'interface de modification.
4. Une fois que l'utilisateur a modifié le contenu qu'il souhaite modifier, l'utilisateur peut cliquer sur le bouton «confirmer». À ce stade, l'interface affiche «Cette modification a été enregistrée» et le logiciel met à jour les informations modifiées par l'utilisateur dans la base de données. Les informations mises à jour seront affichées lors de la prochaine visualisation par l'utilisateur.

2.6 Choisir les préférences

Lorsque les utilisateurs utilisent un logiciel de navigation pour naviguer, ils peuvent vouloir définir des préférences spéciales, telles que l'espoir de ne pas avoir de surveillance radar dans la trajectoire, d'espérer passer ou éviter un certain itinéraire, etc. Pour répondre aux besoins des clients, nous ajoutons un bouton pour sélectionner les préférences dans l'interface principale. (Mais pas la partie de la navigation qui faut être soumise.)

- Quand l'utilisateur est un visiteur, si la préférence n'est pas sélectionnée, le logiciel lance avec le plus court chemin et le plus rapide chemin par défaut.
- Lorsque l'utilisateur est connecté, s'il ne sélectionne pas de préférence, le logiciel utilise la préférence définie dans les informations personnelles pour indiquer le chemin. Si l'utilisateur souhaite définir une autre préférence pour cette recherche, il peut cliquer sur le bouton "Sélectionner les préférences" pour entrer dans l'interface de cette sélection de préférence, mais les paramètres définis ici ne modifieront pas les préférences dans les informations personnelles.

Les utilisateurs peuvent choisir entre les préférences suivantes:

1. Définir la ville de l'itinéraire: Lors de la définition de la ville de l'itinéraire, une zone de liste déroulante apparaît dans l'interface. L'utilisateur peut sélectionner «passer» ou «éviter», ainsi qu'une zone de texte dans laquelle l'utilisateur peut entrer le nom de la ville. Lorsque l'utilisateur entre le nom de la ville et confirme, le logiciel cherche dans la base de données l'existence de la ville. Si le nom de la ville ne peut pas être recherché dans la base de données, l'interface affiche "Désolé, la ville que vous avez entrée n'existe pas, veuillez le vérifier à nouveau et le saisir à nouveau, s'il vous plaît ." Si l'utilisateur n'a pas de ville spécifique, l'utilisateur peut également utiliser la case à cocher multi-sélection pour sélectionner le type de ville que vous souhaitez passer ou à éviter. (Type de ville: grande ville, ville moyenne, petite ville, touristique).
2. Spécifier les caractéristiques des routes: Lors du réglage de la route , il existe une liste déroulante dans l'interface, l'utilisateur peut sélectionner «passer» ou «éviter» et plusieurs cases à cocher permettant à l'utilisateur de cliquer pour définir les caractéristiques de la route que vous souhaitez passer ou éviter. Par exemple, spécifiez le type de route (département, nationale, autoroute, chemin de campagne), s'il

y a un radar sur la route, s'il s'agit d'une vitesse limitée, etc.

3. Les utilisateurs peuvent également choisir le chemin avec la priorité de chemin le plus court ou la priorité de chemin le plus rapide par un bouton sur l'interface.
- Une fois l'utilisateur connecté, l'utilisateur peut modifier les préférences de l'utilisateur par défaut dans les paramètres ou directement sur la page d'accueil. Les préférences de l'utilisateur sont également enregistrées dans l'historique des informations de l'utilisateur et stockées dans la base de données. Mais, à part l'utilisateur, les autres utilisateurs ne peuvent pas consulter son historique de recherche (protection des informations utilisateur) ni divers autres paramètres.
 - Lorsque l'utilisateur ne s'est pas connecté (statut du visiteur), il peut définir des préférences directement sur la page d'accueil. L'historique de la recherche et l'historique de conception des préférences sont stockés dans la base de données. Cependant, il n'y a pas de sécurité, tout le monde peut consulter les historique de recherche et conception des préférences dans le navigateur (local).

- Le logiciel ne peut enregistrer que 10 historiques de recherche, que l'utilisateur soit connecté ou non.

2.7 Chercher le trajet

Lorsque l'utilisateur entre dans l'interface principale du logiciel, il y a deux zones de texte de "point de départ" et "point d'arrivée", et l'utilisateur recherche l'itinéraire en saisissant le nom de la ville.

1. Lorsque l'utilisateur clique sur la zone de texte "Point de départ", les dix enregistrements de recherche de l'utilisateur sont automatiquement affichés (triés par ordre chronologique). L'utilisateur peut également entrer un nouveau nom de ville "point de départ" et, lorsqu'il entre "point de départ" et "point d'arrivée", il peut cliquer sur le bouton de confirmation. À ce stade, le logiciel va rechercher dans la base de données les deux villes entrées par l'utilisateur. Si les deux villes entrées par l'utilisateur ne sont pas trouvées dans la base de données, l'interface affiche "Ville que vous avez entrée n'existe pas , veuillez confirmer et entrer à nouveau, s'il vous plaît ". Si le logiciel a trouvé les deux villes dans la base de données, passez à l'étape suivante.
2. Si l'utilisateur a des exigences particulières pour définir des préférences (partie facultative), lorsque l'utilisateur clique sur le

bouton "Lancer la recherche", le logiciel lit les préférences sélectionnées par l'utilisateur. Si l'utilisateur ne les définit pas, il est transmis sous forme de paramètre vide.

3. Lorsque le logiciel lit le "point de départ" et le "point d'arrivé" saisis par l'utilisateur et les paramètres de préférence, " le cheminle plus court" et " le chemin plus rapide" sont respectivement introduits en fonction des paramètres de préférence. Si toutes les préférences sélectionnées par l'utilisateur ne sont pas respectées, l'ordre est trié en fonction du nombre de préférences satisfaites. Si la ville et la route passant par le "point de départ" au "point d'arrivé " ne peuvent pas satisfaire les préférences de l'utilisateur, le logiciel lancera deux chemins " le cheminle plus court" et " le chemin plus rapide" sans les préférences , et en même temps, il affichera aussi "Désolé,on ne trouve aucun chemin correspondant à vos préférences, revenez en arrière et sélectionnez à nouveau vos préférences ou choisissez le chemin par défaut ci-dessus , s'il vous plaît ."
4. Après l'utilisateur sélectionne un chemin, le logiciel entre dans l'interface de navigation. Et le logiciel passera l'enregistrement de recherche de l'utilisateur à la base de données pour une utilisation ultérieure.

2.8 Verifier les étapes

Lorsque le logiciel entre dans l'état de navigation, afin de s'assurer que l'utilisateur ne décale pas l'itinéraire guidé par la navigation, l'utilisateur peut atteindre avec précision le "destination", le logiciel doit ajouter la fonction permettant de confirmer si l'utilisateur conduit dans le bon tronçon et atteint la ville désignée.

1. L'interface affichera la prochaine section du tronçon à conduire, le temps de trajet restant et le bouton permettant de confirmer si "atteindre la ville connectée au tronçon". Lorsque le prochain tronçon que l'utilisateur va suivre appartient au type de route que l'utilisateur souhaite passer, l'interface affiche également "le segment de route actuel est le type de route que vous souhaitez passer" et toutes les informations de la ville actuelle.
2. Lorsque l'utilisateur clique sur le bouton «atteindre xx ville», si la ville actuellement arrivée par l'utilisateur est la ville que l'utilisateur souhaite passer ou le type de ville qu'il souhaite passer, l'interface affiche "la ville actuelle est le type de ville que vous souhaitez passer" et toutes les informations de la ville. Il y a un bouton "aller à la section suivante" ci-dessous, lorsque l'utilisateur clique sur ce bouton, l'interface affiche le tronçon devant être conduit ensuite et un bouton "confirmer pour atteindre xx ville".

Si la ville d'arrivée actuelle n'est pas la ville que l'utilisateur veut passer ou le type de ville qui veut passer, l'interface affiche directement le tronçon à conduire ensuite et le bouton "confirmer l'arrivée à xx ville"(sans les informations de la ville).

3. Si l'utilisateur n'arrive pas dans la ville indiquée dans la navigation, il peut cliquer sur "Ne pas atteindre la ville xx indiquée". L'interface affichera la prochaine ville qui demandera à l'utilisateur d'arriver. Lorsque l'utilisateur entre le nom de la ville à laquelle le tronçon est sur le point d'arriver, le logiciel recalculera l'itinéraire en fonction des préférences et des destinations préalablement définies par l'utilisateur. Si vous ne trouvez pas de nouvel itinéraire en fonction de vos préférences, le logiciel lancera deux chemins "le plus court" et le "le plus rapide" , tout en indiquant "Désolé, impossible de trouver un chemin correspondant à vos préférences , veuillez revenir en arrière pour sélectionner à nouveau les préférences ou sélectionner un chemin de navigation à partir du chemin par défaut ci-dessus. "
4. Lorsque l'utilisateur atteint la destination , le logiciel se ferme automatiquement et quitte l'interface de navigation.

2.9 Finir la simulation

Lorsque le logiciel est dans l'interface de navigation, si l'utilisateur souhaite sélectionner un autre itinéraire ou modifier le point de départ ou le point d'arrivée, il peut quitter de force l'état de navigation en cliquant sur le bouton "Quitter la navigation"

1. Il y a un bouton "Quitter la Navigation" dans "l'interface de navigation". L'utilisateur peut cliquer sur le bouton "Quitter la navigation" pour revenir à l'interface principale.
2. Après l'utilisateur clique sur le bouton "Quitter la Navigation". Dans l'interface principale, les zones de texte du point de départ et du point d'arrivée conservent le nom de la ville précédemment saisi par l'utilisateur. L'utilisateur peut aussi changer le point de départ ou le point d'arrivée par le nom de la ville.

2.10 Déconnexion

Quand l'utilisateur souhaite se déconnecter, l'utilisateur peut cliquer sur le bouton "se déconnecter" pour se déconnecter dans l'interface des paramètres afin de forcer la déconnexion.

1. Au bas de l'interface des paramètres, il y a un bouton «Déconnexion». Lorsque l'utilisateur clique pour se déconnecter, le logiciel entre l'état du visiteur. Les utilisateurs ne peuvent pas consulter toutes les histoires de recherche et tous les

informations personnelles de tous les utilisateurs ayant un compte sur ce potable.

2.11 Gestion la base de données de map

Dans la réalité, peut-être le gouvernement va changer le nom de la ville ou le nom de la route , certaines routes sont en cours de réparation ou un nouveau tronçon est ajouté entre deux villes. Afin de rendre le système de navigation plus précis et plus en temps réel , un administrateur est tenu de mettre à jour la base de données de la carte de gestion.

L'administrateur dispose d'un compte spécial: lorsqu'il se connecte au compte, il peut visualiser toutes les villes et leurs informations, consulter tout le tronçon et toutes ses informations, ainsi que tous les routes et leurs informations. L'administrateur peut sélectionner un objet et entrer dans l'interface "Modifier la base de données de cartes".

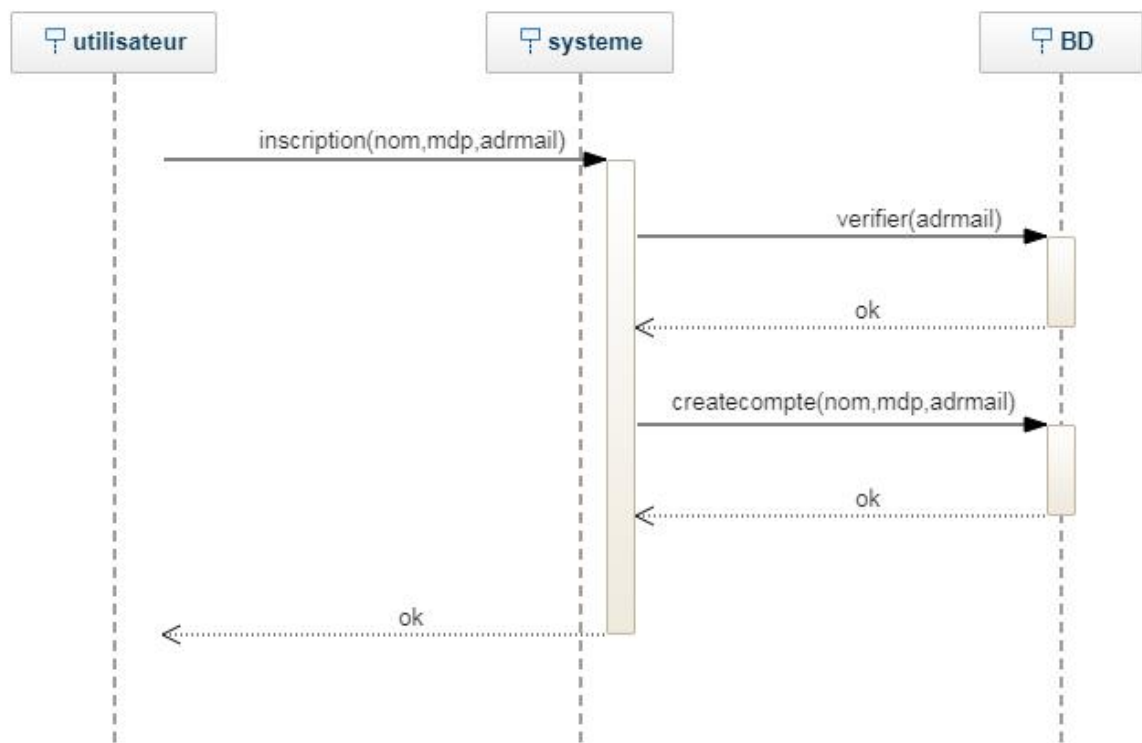
1. Ville: l'administrateur peut modifier le nom de la ville, qu'il soit touristique ou non, également le code postal et la position coordonnée de la ville. L'administrateur peut ajouter ou supprimer une ville.
2. Troncon: l'administrateur peut modifier la limite de vitesse, la longueur, le nombre de radars, la nécessité de payer, il peut aussi

modifier l'identification du tronçon . L'administrateur peut également ajouter ou supprimer un tronc.

3. Route: l'administrateur peut modifier le nom et le type de la route , il peut aussi modifier l'identification de la route . L'administrateur peut ajouter ou supprimer une route

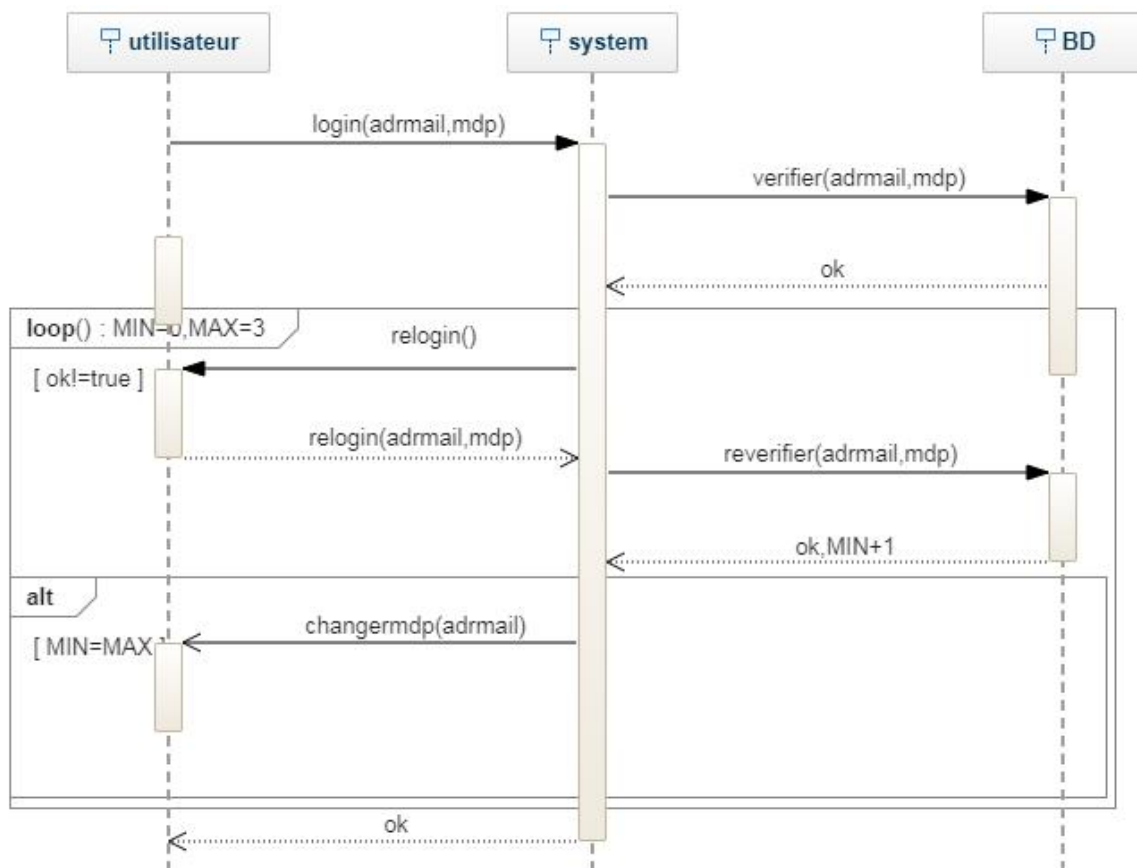
3. Diagramme de séquence

3.1 Inscription



3.2 Login

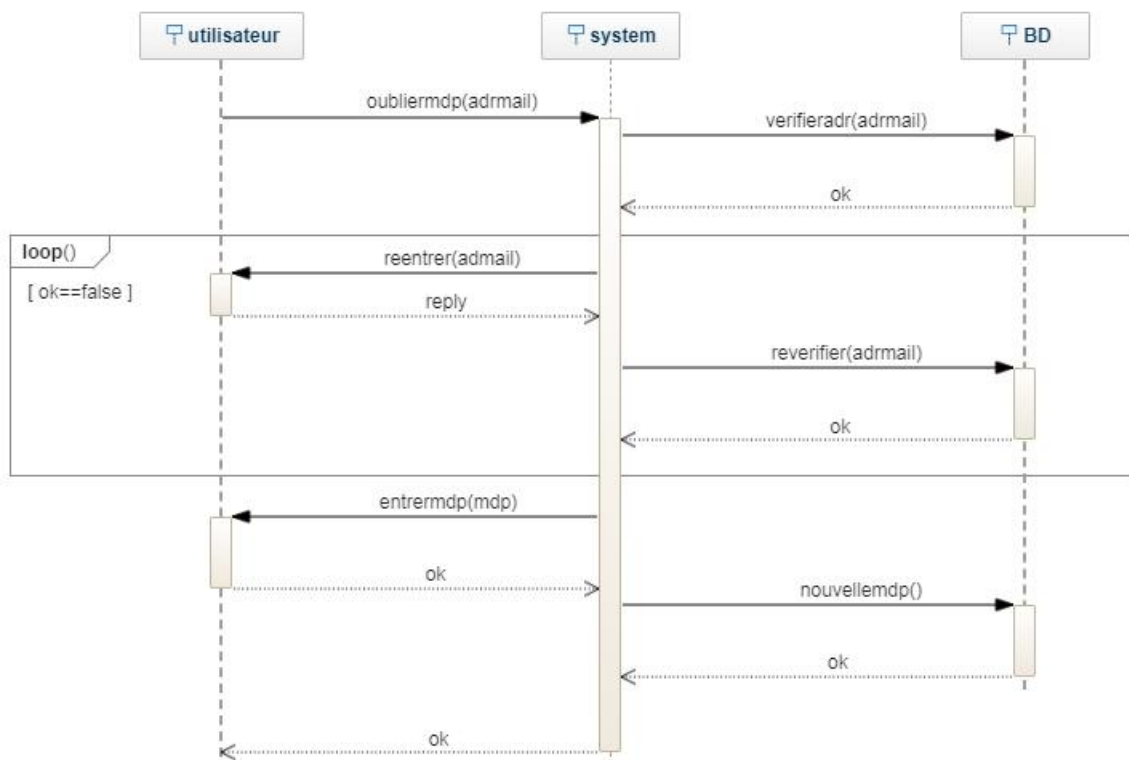
- Ajoutez une boucle pour s'assurer que l'utilisateur est autorisé à entrer au plus trois fois le mot de passe incorrect. Si le mot de passe incorrect est saisi trois fois, le système envoie automatiquement un message demandant à l'utilisateur de modifier le mot de passe.



3.3 Oublier le mot de passe

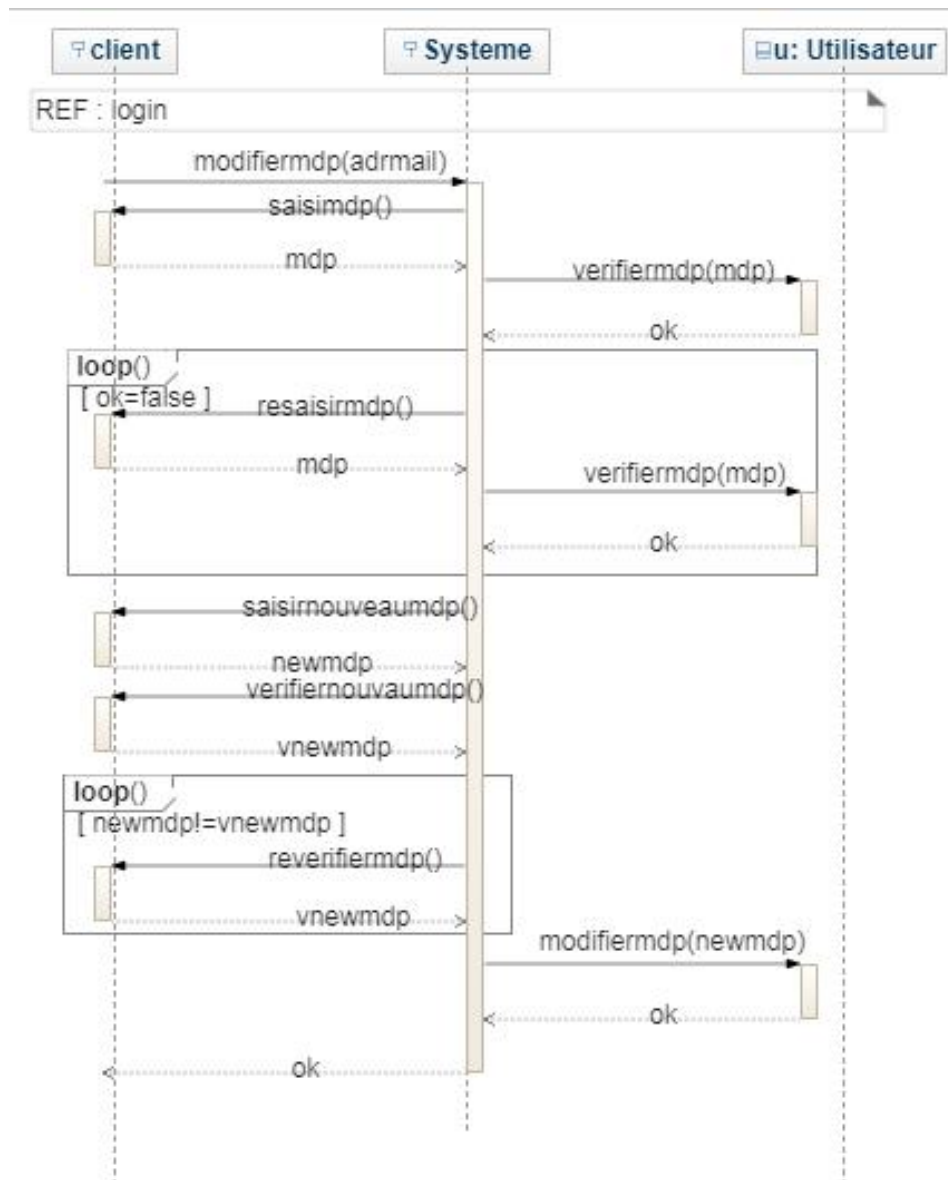
- La boucle est pour que l'utilisateur entre la mauvaise adresse email , puis puisse le saisir à nouveau jusqu'à ce que l'adresse

email correcte soit entrée.



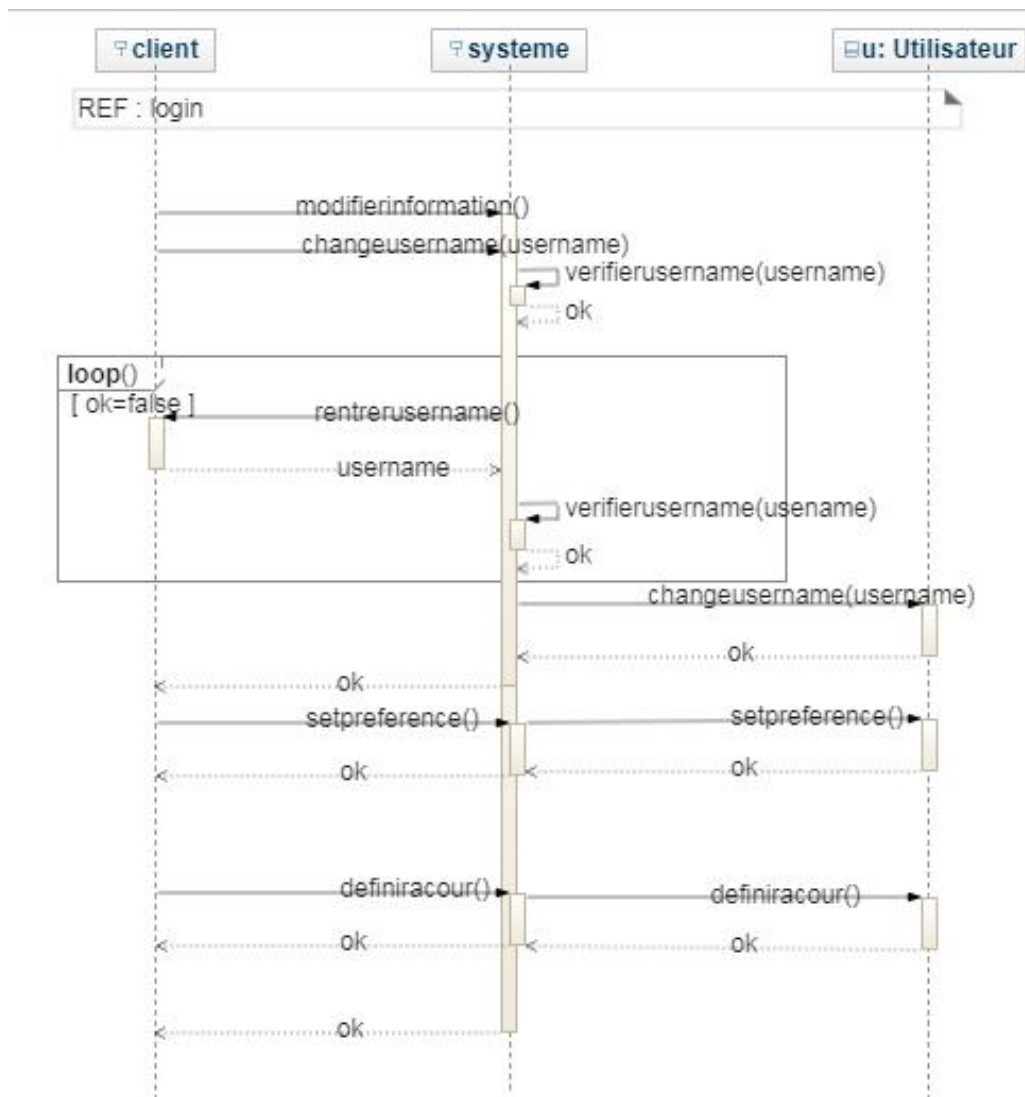
3.4 Modifier le mot de passe

- La première boucle consiste à vérifier que l'utilisateur connaît le mot de passe d'origine, au cas où quelqu'un vole le compte pour changer le mot de passe.
- La deuxième boucle consiste à vérifier que les deux mots de passe d'entrée sont cohérents, ce qui empêche l'utilisateur de saisir la mauvaise entrée.



3.5 Modifier les informations personnelles

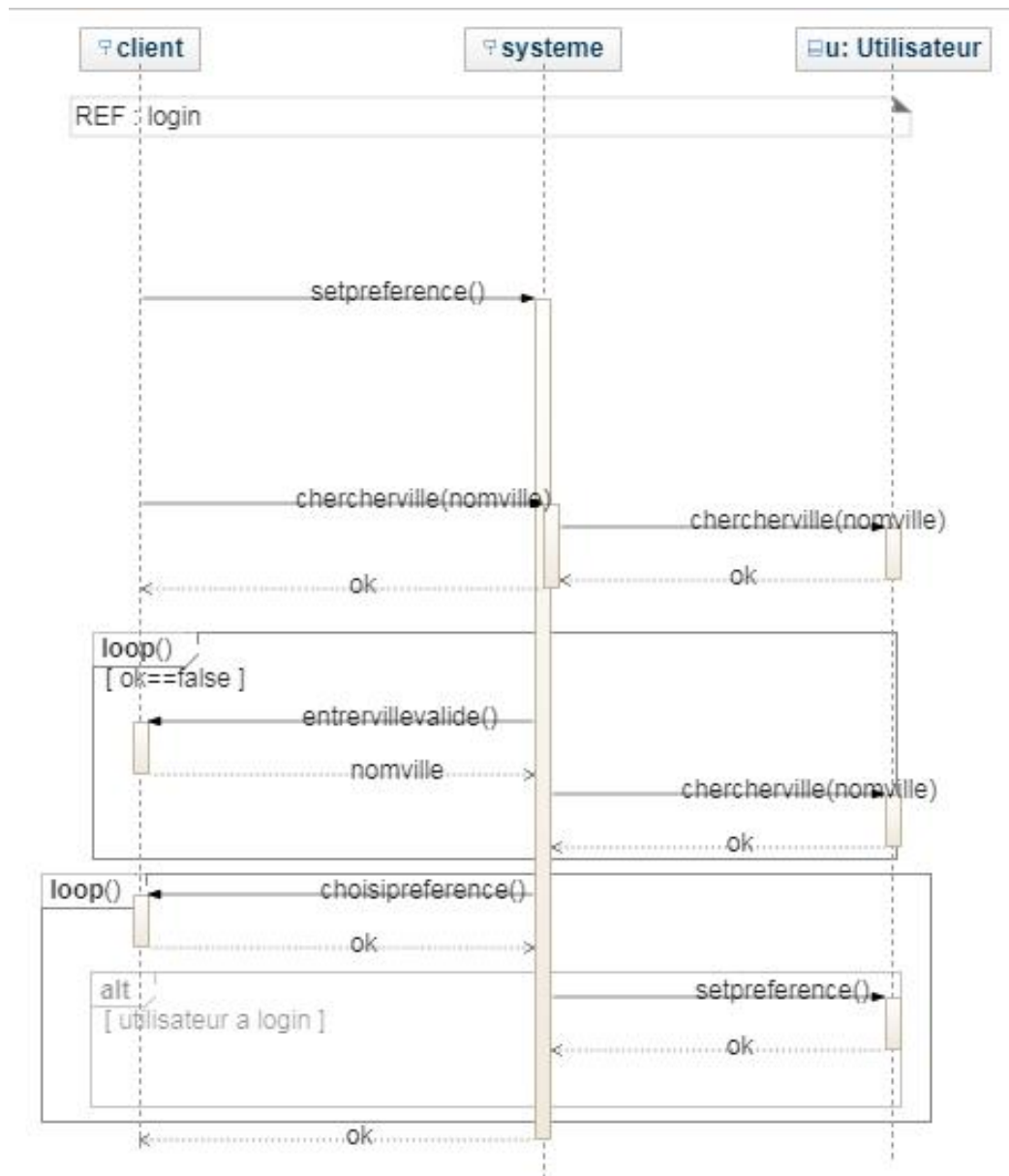
- Ajouter cette boucle pour s'assurer que le nom d'utilisateur peut être ressaisi chaque fois que l'utilisateur entre un caractère illégal jusqu'à ce que tous les caractères soient légaux



3.6 Choisir les préférences

- La première boucle assure que la ville entrée par l'utilisateur existe dans la base de données
- La deuxième boucle sert à sélectionner la préférence que l'utilisateur souhaite définir.
- alt permet de garantir que seules les préférences définies par l'utilisateur connecté peuvent être stockées dans la base de

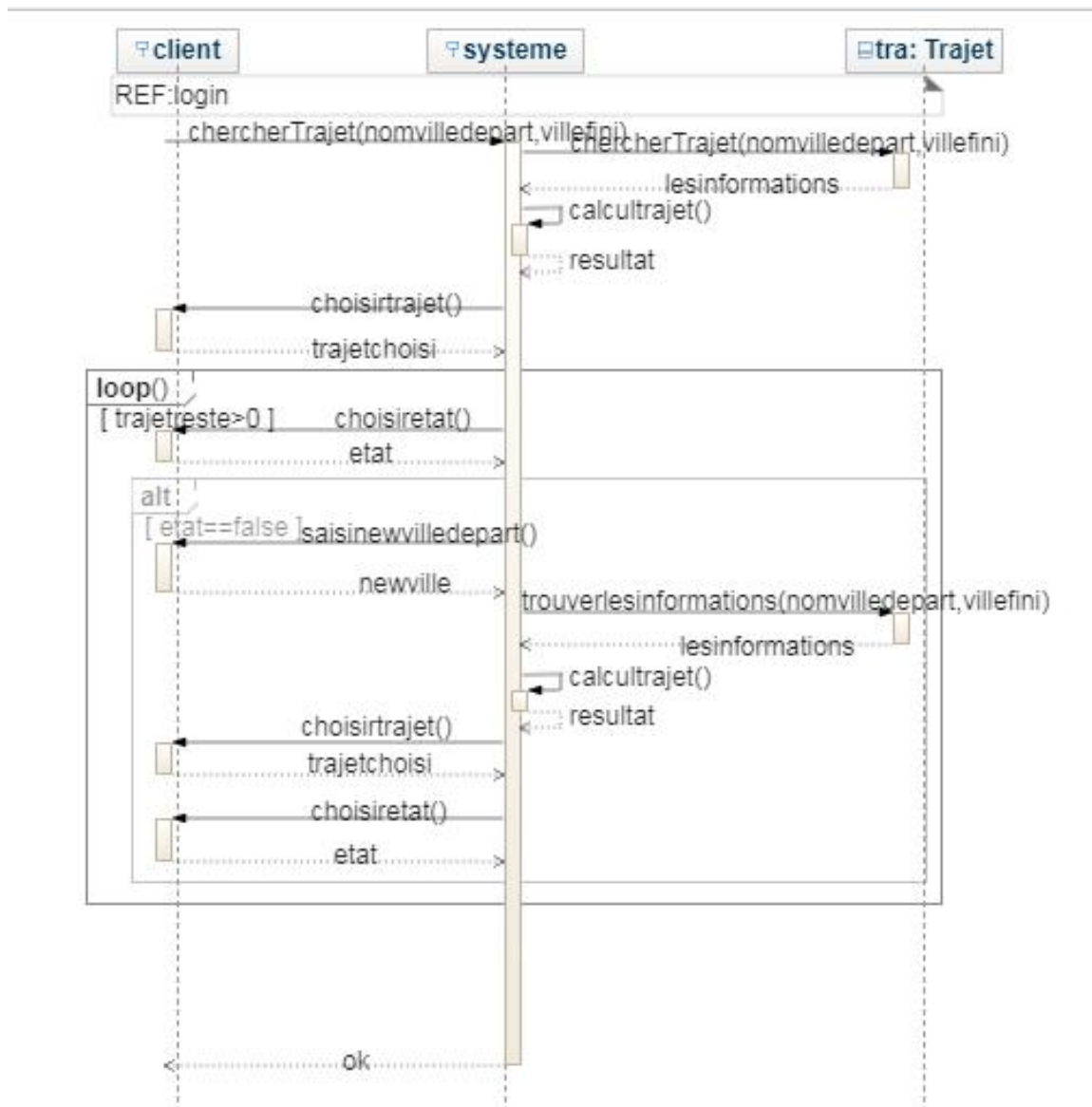
données.



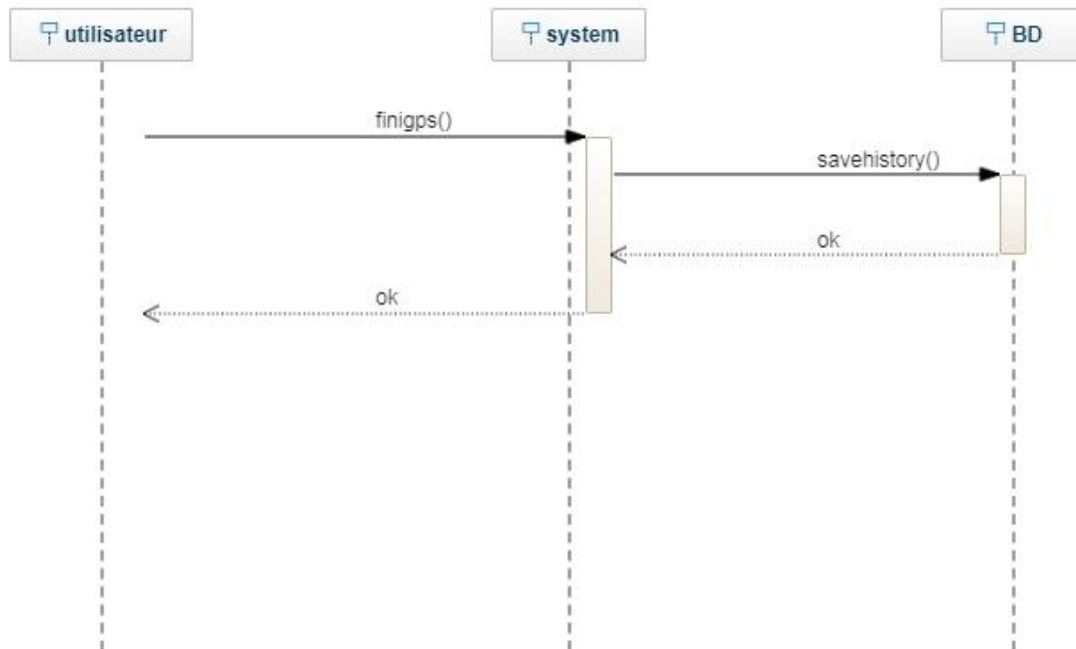
3.7 Chercher le trajet

- La boucle garantit que, lorsque la simulation n'est pas terminée, le système demande toujours à l'utilisateur s'il est arrivé à la ville

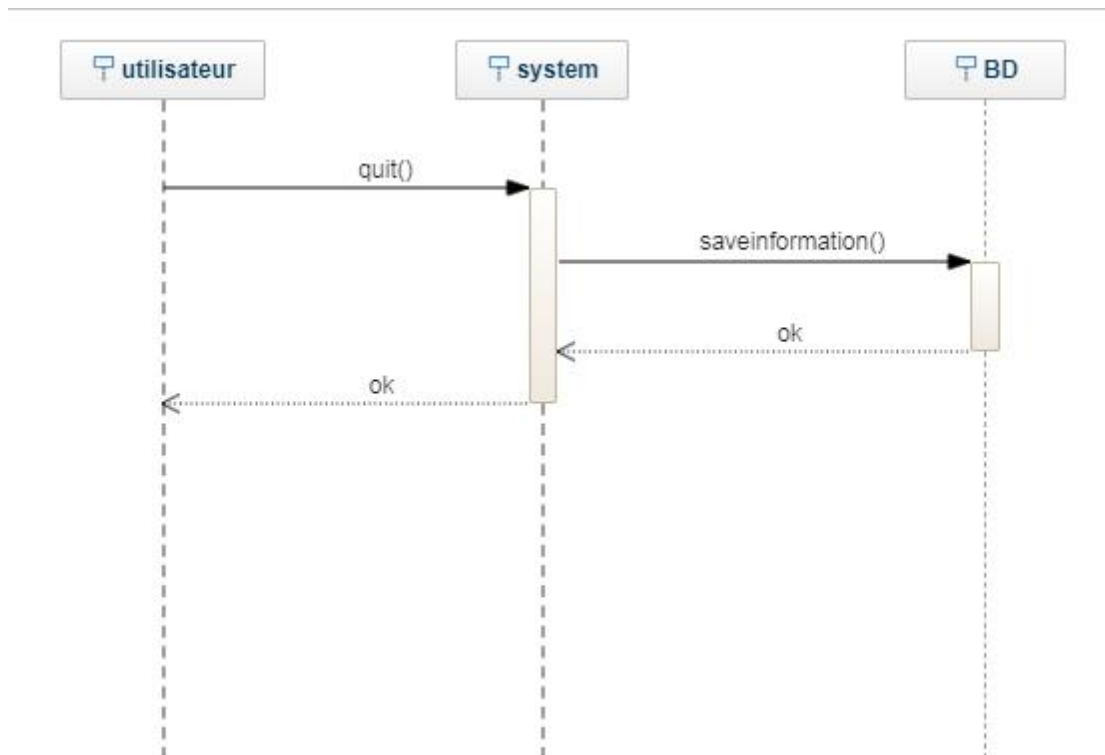
indiquée et obtient l'état de l'utilisateur. Si l'utilisateur se trompe, la ville sur le point d'arriver peut être saisie à nouveau comme point de départ. La destination et les préférences sont préalablement définies par l'utilisateur.



3.8 Finir la simulation

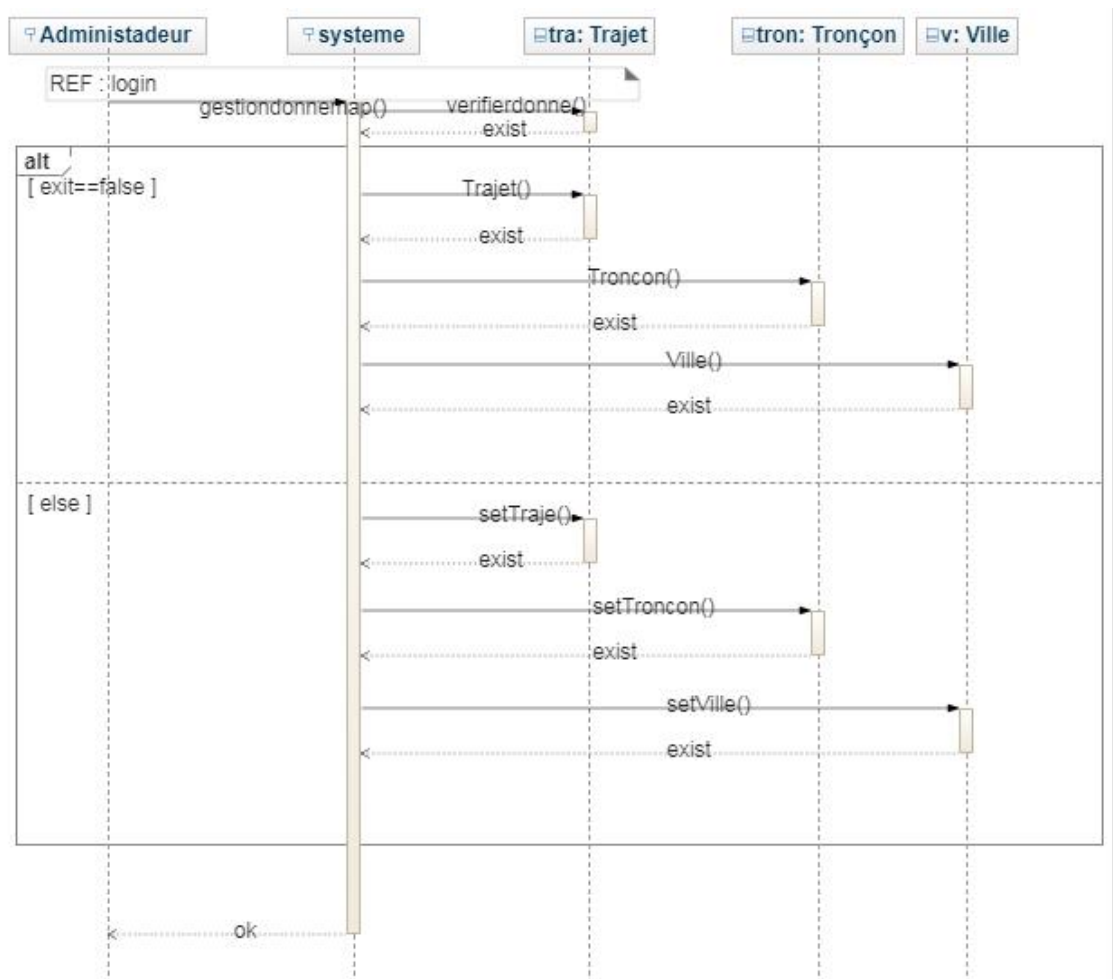


3.9 Déconnexion

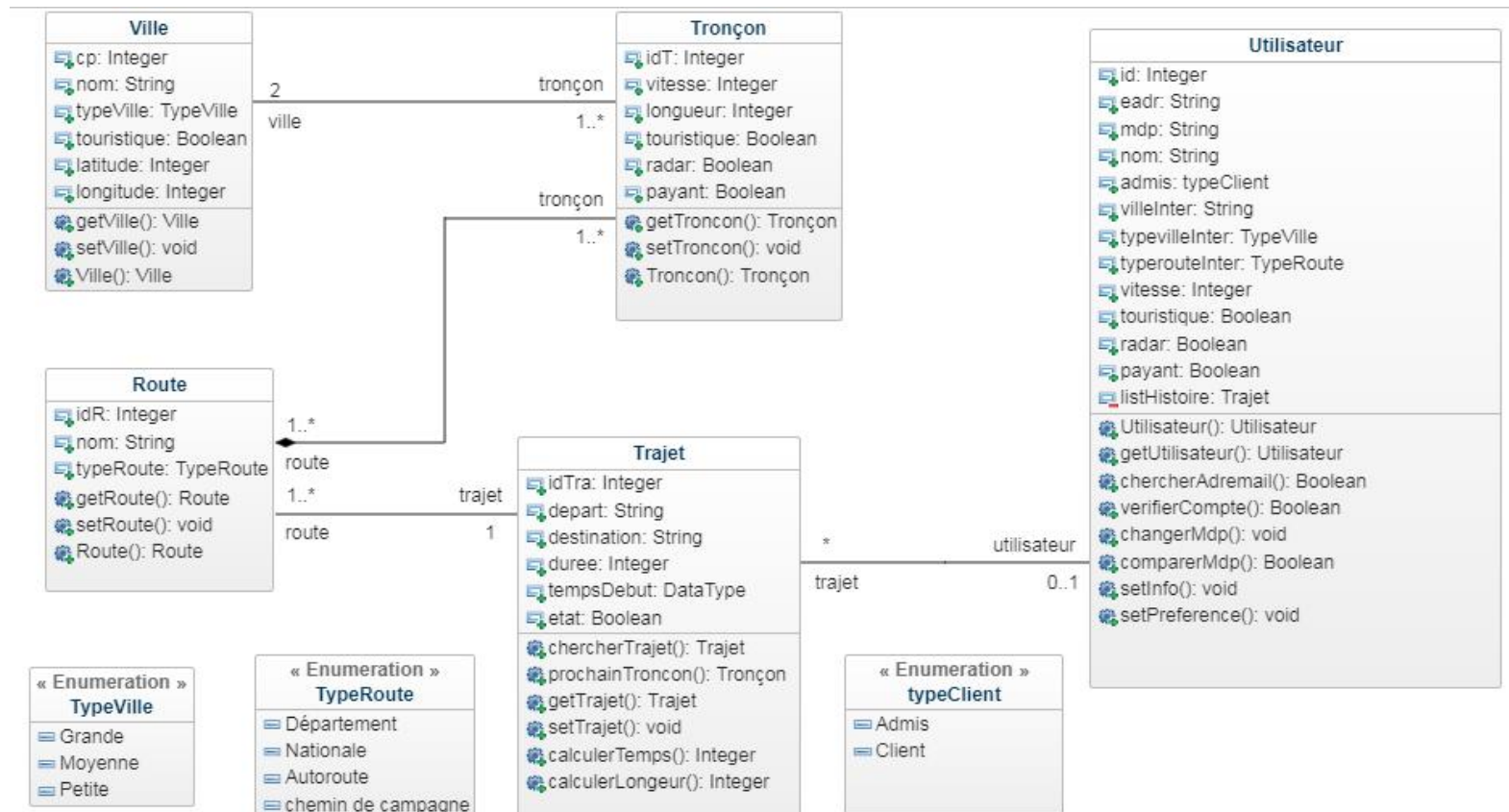


3.10 Gestion la base de données

- Cette boucle permet de s'assurer que les informations modifiées par l'administrateur sont mises à jour synchroniquement dans la base de données.



4. Digramme de classe



4.1 Exigences fonctionnelles

4.1.1. Classe de <Utilisateur>

- L'attribut <id> dans cette classe pour identifier chaque utilisateur.

Constructeur 1	Utilisateur
Entrée	eadr,mdp,nom,admis,
Sortie	Utilisateur
Description	<p>Lorsqu'un utilisateur enregistre un compte, il doit ajouter l'utilisateur et ses informations aux données, auquel cas un nouvel objet Utilisateur doit être créé. Vous devez donc ajouter un constructeur dans la classe Utilisateur.</p> <p>Pour pouvoir instancier cette classe Utilisateur, il faut créer un constructeur. Un constructeur est un bloc de code qui initialise un objet immédiatement après sa création.</p>
Moal	definition <pre>preUtilisateur(nom:String,adr:String,pwd:String)</pre> $\equiv \text{nom} \neq \text{null} \wedge \text{adr} \neq \text{null} \wedge \text{pwd} \neq \text{null}$ definition

	$\text{postUtilisateur}(\text{nom:String}, \text{eadr:String}, \text{pwd:String}, \text{r:Boolean}) \equiv \forall x \in \text{Utilisateur}, \{x x.\text{eadr} \neq \text{eadr}\} \wedge r = \text{true}$
--	---

Fonction 1	getUtilisateur
Entrée	void
Sortie	id
Description	<p>Pour empêcher que les attribues de la classe Utilisateur soient modifiées par d'autres classes, le type de données de la classe Utilisateur est privé. En d'autres termes, les données des membres de ces types privés ne peuvent être modifiées qu'au moment de la définition, aucune autre option ne peut être utilisée. Les objets de cette classe ne sont pas non plus autorisés. Afin de rendre les données de la classe Utilisateur disponibles dans d'autres classes, nous pouvons également définir une fonction publique qui lit les données de type privé dans la classe. Il faut donc ajouter getUtilisateur () pour obtenir l'identifiant de l'utilisateur.</p>

Fonction 2	chercherAdremail
Entrée	adremail

Sortie	boolean
Description	<p>Quand l'utilisateur se connecte ou récupère le mot de passe, il est nécessaire de confirmer si l'adresse email entrée par l'utilisateur existe réellement dans la base de données. En obtenant l'adresse email entrée par l'utilisateur, par rapport à tous les adresses e-mail des données, s'il existe une correspondance, il va renvoyer true, sinon renvoyer false.</p>
Moal	<p>definiion $\text{preChercherAdremail}(\text{adr}:\text{String}) \equiv \text{adr} \neq \text{null}$</p> <p>definition $\text{postChercherAdremail}(\text{adr}:\text{String}, r:\text{Boolean}) \equiv \exists x \in \text{Utilisateur}, \{x x.\text{eadr} = \text{eadr}\} \wedge r = \text{true}$</p>

Fonction 3	verifierCompte
Entrée	adremail,mdp
Sortie	boolean
Description	<p>Lorsque l'utilisateur souhaite se connecter au compte, il doit entrer son adresse email et le mot de passe. L'adresse email d'utilisateur, le mot de passe et d'autres informations de l'utilisateur sont stockés dans la base de données. Afin de protéger la sécurité des informations de l'utilisateur, l'adresse email et le mot de</p>

	<p> passe saisis lors de la connexion de l'utilisateur doivent correspondre au nom du compte et au mot de passe figurant dans la base de données. Si l'utilisateur (adresse email) n'est pas trouvé dans la base de données, il retournera false. Une fois l'adresse email correspondante trouvée dans la base de données, vérifiez si le mot de passe est le même. Si les mots de passe ne sont pas les mêmes, il renvoie également false. Renvoie true uniquement si le nom d'utilisateur est trouvé et que le mot de passe est correct. </p>
Moal	<p> definition $\text{preVerifierCompte}(\text{adr:String}, \text{mdp:String}) \equiv \text{adr} \neq \text{null} \wedge \text{mdp} \neq \text{null}$ definition $\text{postVerifierCompte}(\text{adr:String}, \text{mdp:String}, \text{r:boolean}) \equiv \exists x \in \text{Utilisateur}, \{x \mid x.\text{eaddr} = \text{adr} \wedge x.\text{mdp} = \text{mdp}\} \wedge \text{r} = \text{true}$ </p>

Fonction 4	changerMdp
Entrée	mdp, nmdp1, nmdp2
Sortie	void
Description	Lorsque l'utilisateur souhaite modifier le mot de

	<p> passe, il doit être réinitialisé. Dans la classe Utilisateur, tous les paramètres sont privés et les objets de cette classe ne peuvent pas modifier les données de manière arbitraire. Pour que modifier les attributs dans la classe Utilisateur, nous devons ajouter la méthode changeMdp à la classe Utilisateur pour définir le mot de passe dans la classe Utilisateur. </p> <p> Le mdp du paramètre doit être comparé au mdp de l'utilisateur obtenu dans la base de données. Lorsque les deux sont identiques, les paramètres nmdp1 et nmdp2 sont comparées. Si les deux sont identiques, le mot de passe est bien modifié. Sinon, la modification du mot de passe a échoué. </p>
Moal	<p> definition </p> <p> $\text{preChangerMdp}(\text{mdp}:\text{string}, \text{nmdp1}:\text{string}, \text{nmdp2}:\text{string}) \equiv \text{mdp} \neq \text{null} \wedge \text{nmdp1} \neq \text{null} \wedge \text{nmdp2} \neq \text{null}$ </p> <p> definition </p> <p> $\text{postChangerMdp}(\text{mdp}:\text{string}, \text{nmdp1}:\text{string}, \text{nmdp2}:\text{string}, r:\text{boolean}) \equiv \exists x \in \text{Utilisateur}, \{x.\text{mdp} = \text{mdp}\} \wedge \text{nmdp1} = \text{nmdp2} \wedge r = \text{true}$ </p>

Fonction 5	comparerMdp
Entrée	mdp1,mdp2
Sortie	Boolean
Description	<ul style="list-style-type: none"> ● Lorsque l'utilisateur enregistre un nouveau compte ou qu'il modifie le mot de passe, il est nécessaire de confirmer que les deux nouveaux mots de passe entré sont identique. ● Lorsque l'utilisateur se connecte, il doit confirmer que le mot de passe entré par l'utilisateur est identique à celui de la base de données. <p>Nous devons donc ajouter une méthode pour comparer si les deux mots de passe sont identiques. Si les deux mots de passe sont identiques, renvoyez true, sinon renvoyez false.</p>

Fonction 6	setInfo
Entrée	nom,villeInter,typevilleInter,typerouteInter,vitesse,radar,payant,position
Sortie	void
Description	Étant donné que les informations de l'utilisateur sont de type <private>, lorsque l'utilisateur souhaite modifier

	les informations personnelles, il faut utiliser la méthode setInfo pour modifier les informations de l'utilisateur, les paramètres seront enregistrés dans la base de données.
--	--

Fonction 7	setPreference
Entrée	villeInter, typevilleInter, typerouteInter, vitesse, radar, payant
Sortie	void
Description	Étant donné que les informations de l'utilisateur sont de type <private>, lorsque l'utilisateur souhaite choisir ses préférences, il faut utiliser la méthode setPreference pour choisir les préférences de l'utilisateur, les paramètres seront enregistrés dans la base de données.

4. 1. 2. Classe de <Ville>

Fonction 8	setVille
Entrée	cp, nom, typeVille, touristique, latitude, longitude
Sortie	void
Description	Pour que l'administrateur puisse modifier la ville sur la

	carte ou ajouter une ville à la carte, il doit ajouter la méthode setVille pour mettre à jour la ville dans la base de données.
--	---

Fonction 9	getVille
Entrée	void
Sortie	Vile
Description	Pour permettre aux autres classes de consulter des informations sur une ville, mais comme chaque attribut dans ville est privé, nous ajoutons la méthode getVille pour obtenir les détails de la ville.

4. 1. 3. Classe de <Tronçon >

Fonction 10	setTroncon
Entrée	Vitesse,longueur,touristique,radar,payant
Sortie	void
Description	Pour que l'administrateur puisse modifier le troncon sur la carte ou ajouter un tronçon à la carte, il doit ajouter la méthode setTroncon pour mettre à jour le tronçon dans la base de données.

Fonction 11	getTroncon
Entrée	void
Sortie	Troncon
Description	Pour permettre aux autres classes de consulter et utiliser des informations sur un troncon (par exemple ,pour calculer le longueur d'un trajet) , mais comme chaque attribut dans Troncon est privé, nous ajoutons la méthode getTroncon pour obtenir les détails du tronçon .

4. 1. 4. Classe de <Route>

Fonction 12	setRoute
Entrée	nom,typeRoute
Sortie	Void
Description	Pour que l'administrateur puisse modifier la route sur la carte ou ajouter une route à la carte, il doit ajouter la méthode setRoute pour mettre à jour la ville dans la base de données.

Fonction 13	getRoute
Entrée	void

Sortie	Route
Description	Pour permettre aux autres classes de consulter des informations sur une route(par exemple ,pour éviter une toute de type “chemin de campagne”), mais comme chaque attribut dans la classe Route est privé, nous ajoutons la méthode getRoute pour obtenir les détails de la route.

4. 1. 5. Classe de <Trajet>

Fonction 16	getTrajet
Entrée	void
Sortie	liste de troncon
Description	Afin de permettre aux autres classes de consulter les troncons de ce trajet, nous devons définir la méthode de getTrajet.

Fonction 14	chercherTrajet
Entrée	depart,destination,preferences
Sortie	liste de trajet
Description	Afin de trouver un chemin répondant aux besoins de l'utilisateur, il faut d'abord déterminer les points de

	<p>début et d'arrivé de l'utilisateur. Et trouvez le chemin dans tous les chemins correspondant aux préférences définies par l'utilisateur.</p> <ul style="list-style-type: none"> ● Renvoie null s'il n'y a pas de chemin entre les points de départ et d'arrivée ● Si un chemin répondant à toutes les exigences de l'utilisateur n'est pas trouvé, le chemin le plus satisfait de toutes les préférences de l'utilisateur est renvoyé. ● Si aucun chemin correspondant aux préférences de l'utilisateur n'est trouvé, un chemin sans préférence est renvoyé et print "Impossible de trouver un chemin répondant aux critères".
Moal	<p>definition</p> <p>preChercherTrajet(depart:ville,destination:ville,preferences:liste d'object)≡</p> <p>depart!=null/\destination!=null/\preferences!=null</p> <p>definition</p> <p>postChercherTrajet(depart:ville,destination:ville,preferences:liste d'object,r:boolean) ≡ ∃ v1,v2 ∈ Ville,{v1.nom=depart/\v2.nom=destination}/\r=true</p>

Fonction 15	prochainTroncon
Entrée	liste de troncon
Sortie	Tronçon
Description	<p>Afin de s'assurer que l'utilisateur roule sur le bon chemin, il est nécessaire de confirmer que chaque troncon piloté par le conducteur est correct.</p> <p>Cette méthode est utilisée pour rappeler au pilote que le prochain troncon est celui-ci. Chaque fois que l'utilisateur clique pour confirmer pour atteindre une ville, le troncon correspondant de la liste est supprimé et le troncon suivant devant être piloté est renvoyé.</p> <p>Si le troncon suivant est vide, retourne null et affiche "Vous avez atteint la destination"</p>

Fonction 18	calculerTemps
Entrée	liste de troncon
Sortie	integer
Description	<p>Afin de faciliter la sélection du chemin le plus rapide , il est nécessaire de calculer le temps total nécessaire pour parcourir chaque trajet.</p> <ul style="list-style-type: none"> ● Dans la liste du troncon, nous pouvons utiliser les

	<p>attributs du troncon (limite de vitesse et longueur)</p> <p>pour calculer le temps nécessaire pour parcourir chaque troncon et additionner le temps pris par chaque troncon, qui correspond au temps total de trajet.</p>
Pseudocode	<pre> fonction calculertemps(trajet:liste de tronçon){ int temps=0; pour tous t dans trajet faire temps=temps+t.longueur/t.vitesse; fpour return temps; } </pre>

Fonction 19	calculerLongueur
Entrée	liste de troncon
Sortie	Integer
Description	<p>Afin de permettre à l'utilisateur de sélectionner le chemin le plus court, la méthode de calcul de la longueur totale du chemin est nécessaire.</p> <ul style="list-style-type: none"> ● À travers la liste du troncon entrant, nous pouvons calculer la longueur totale du chemin en accumulant

	les attributs de chaque tronçon (longueur).
Pseudocode	<pre> fonction calculerlongueur(trajet:liste de tronçon){ int longueur=0; pour tous t dans trajet faire temps=temps+t.longueur; fpour return longueur; }</pre>

4. 2. Exigence spécifique

Afin de trouver le chemin rapidement et complètement, nous devons écrire un algorithme efficace pour chercher le trajet -- A Etoile

4. 2. 1. Contrainte de chercher un plus court trajet

- La base de données de la carte est volumineuse et il faut beaucoup de temps pour parcourir chaque tronçon afin de trouver le chemin le plus court. Il est nécessaire de ne pas perdre de temps à parcourir chaque tronçon et à trouver avec précision le chemin le plus court. L'algorithme doit donc être efficace.
- Dans notre logiciel GPS-Like, les utilisateurs peuvent définir leurs

propres préférences de chemin et le logiciel doit trouver le chemin en fonction de ses préférences. Par exemple, si l'utilisateur ne veut pas passer par le tronçon avec radar, il doit supprimer le tronçon avec radar pendant chercher le trajet, etc. Cela nécessite que l'algorithme soit suffisamment flexible.

4. 2. 2. Algorithme de A Etoile

4.2.2.1. Concept de base

- **dejaDev liste:** Enregistrez les villes qui ne seront plus considérées (le meilleur chemin qu'on a déjà trouvé)
- **Frontiere liste:** Enregistrer toutes les villes considérées pour trouver le chemin le plus court
- **unAcces liste:** Enregistrez les villes que l'utilisateur ne veut pas passer et les villes lient les tronçon que l'utilisateur ne veut pas les passer
- **G:** La longueur de la ville de départ à la ville actuelle. Ainsi, la longueur de la ville de départ à la ville voisine est la longueur qui relie les deux villes au tronçon et la valeur de G augmente à mesure que l'on s'éloigne de plus en plus du point de départ.
- **H:** La longueur estimée de la ville actuelle à la ville d'arrivée .

Mais comme cette route n'a pas encore parcourue, ce n'est qu'une estimation.

- F: $F = G + H$, la longueur totale du point de départ au point final

4.2.2.2. Commencer à chercher

1. Ajoutez la ville de départ A à la liste de frontière
2. Consulter les villes voisines de la ville de départ A et ajoutez les villes accessibles à la liste de frontière. Et définissez la ville de départ A comme le père de ces villes.
3. Supprimez la ville de départ A de la liste frontière et ajoutez-la à la liste déjàDev.

4.2.2.3. Continuer la recherche

Afin de continuer la recherche, nous répétons ce qui suit:

4. Sélectionnez Ville B avec la valeur F la plus basse dans la liste de frontière, puis supprimez-la de la liste de frontière et ajoutez-la à la liste déjàDev.
5. Consulter toutes les villes(noté : ville T) proches de la ville B
 - 5.1. Si la ville T est dans la liste unAcces ou déjà dans la liste déjàDev, ignorez-la

5.2. Si la ville T n'est pas dans la liste de frontière, ajoutez-la à la liste de frontière. Et définissez la ville B sélectionnée comme le père de cette nouvelle ville.

5.3. Si la ville T est déjà dans la liste de frontière, vérifiez si le nouveau chemin est meilleur avec la valeur G, c'est-à-dire si la ville T a une valeur G plus petite via la ville actuelle B (la ville que nous avons sélectionnée). Sinon, ne faites rien.

Inversement, si la valeur G est plus petite, le père de la ville T est défini sur la ville actuelle B (la ville sélectionnée), puis les valeurs F et G de la ville T sont recalculées.

4.2.2.4. Arrêtez de chercher

Arrêtez quand

- Ajouter la ville de destination à la liste de frontière, à ce moment là, le meilleur chemin a été trouvé, ou
- Le point final de recherche a échoué et la liste de frontière est vide. Il n'y a pas de chemin pour l'instant.

6. Enregistrer le chemin. Partant de la ville de destination, chaque ville se déplace le long du nœud parent vers la ville de départ, qui est le chemin le plus court.

4. 2. 3. Pseudocode de chercherTrajet (Aetoile)

Function rechercheAEtoile(départ :ville,destination :ville){

 Ville dejaDev=[]; //la liste de ville qu'on a déjà trouvé le meilleur chemin

 Ville frontiere=[départ] ; //la liste de ville qu'on n'a pas déjà trouvé le meilleur chemin

 g(départ)=0 ; // g(n) est la plus petite somme de longueur de tronçon entre départ et n

 f(départ)=h (départ); //h(n) est la distance entre n et destination; f(n) est une heuristique et f(n)=g(n)+h(n)

 tant que (frontiere !=null) faire

 n=choixFMin(frontiere) // choixFMin(frontiere): on choisi le minimum f dans toutes les villes on n'a pas déjà trouvé le meilleur chemin

 si estTerminal(n) alors retourner construireSolution(n,pere)

 sinon

 supprimer(n,frontiere) // cette chemin est le meilleur chemin entre départ et n

```

ajouter(n,dejadev) // cette chemin est le meilleur chemin entre
départ et n

pour tous  $s \in \text{successeurs}(n)$  faire

    si  $s \notin \text{dejaDev}$  et  $s \notin \text{frontiere}$  alors // si c'est une ville on a
    jamais calculé sa heuristique, on mis a jour son g, h et f

        pere(s)=n;

         $g(s)=g(n)+\text{longueur}(n,s); f(s)=g(s)+h(s);$ 

        ajouter(s,Frontiere);

    sinon

        si  $g(s) > g(n)+\text{longueur}(n,s)$  alors

            //si c'est une ville on a déjà calculé sa heuristique, on compare le chemin
            qu'on trouve maintenant et le meilleur chemin ancien.

            // Si le chemin maintenant est mieux que le meilleur chemin ancien,on
            mis a jour son g, h et f,sinon on fait rien

            //longueur(n,s) retourne le longueur de trouçon entre la ville n et s

            pere(s)=n;  $g(s)=g(n)+\text{longueur}(n,s); f(s)=g(s)+h(s);$ 

retourner echec;//si le boucle dessus ne peut pas trouver le chemin,
c'est pas possible on conduire entre le départ et le destination

```

}