Table des matières

[1. Cahier des charges 2](#_Toc419315881)

[1.1. Descriptif de l’applicatif 2](#_Toc419315882)

[1.1.1. Objectifs de base 2](#_Toc419315883)

[1.1.2. Contraintes 2](#_Toc419315884)

[1.1.3. Utilisation de l’applicatif 2](#_Toc419315885)

[1.1.4. Règles du jeu 2](#_Toc419315886)

[1.2. Partage des responsabilités 5](#_Toc419315887)

[1.2.1. Client 5](#_Toc419315891)

[1.2.2. Serveur 6](#_Toc419315892)

[1.3. Cas d’utilisation 7](#_Toc419315893)

[1.3.1. Diagramme général de contexte 7](#_Toc419315895)

[1.3.2. Description des acteurs 7](#_Toc419315896)

[1.3.3. Scénario « Se connecter au serveur avec un compte utilisateur » 8](#_Toc419315897)

[1.3.4. Scénario « Créer un compte utilisateur » 8](#_Toc419315898)

[1.3.5. Scénario « Editer son compte utilisateur » 9](#_Toc419315899)

[1.3.6. Scénario « Se déconnecter du compte utilisateur » 9](#_Toc419315900)

[1.3.7. Scénario « Gérer une partie » 10](#_Toc419315901)

[1.3.8. Scénario « Créer une partie » 10](#_Toc419315902)

[1.3.9. Scénario « Choisir une carte » 10](#_Toc419315903)

[1.3.10. Scénario « Fixer le nombre de joueurs » 10](#_Toc419315904)

[1.3.11. Scénario « Supprimer une partie » 11](#_Toc419315905)

[1.3.12. Scénario « Rejoindre une partie » 11](#_Toc419315906)

[1.3.13. Scénario « Initialiser une partie » 11](#_Toc419315907)

[1.3.14. Scénario « Disputer une partie » 11](#_Toc419315908)

[1.3.15. Scénario « Quitter une partie » 12](#_Toc419315909)

[1.3.16. Scénario « Arrêter une partie » 12](#_Toc419315910)

[1.3.17. Scénario « Jouer un tour » 12](#_Toc419315911)

[1.3.18. Scénario « Jouer un tour en tant que détective » 12](#_Toc419315912)

[1.3.19. Scénario « Jouer un tour en tant que Mister X » 13](#_Toc419315913)

[1.3.20. Scénario « Se déplacer » 13](#_Toc419315914)

[1.3.21. Scénario « Choisir un moyen de déplacement » 13](#_Toc419315915)

[1.3.22. Scénario « Bloquer un utilisateur » 14](#_Toc419315916)

[1.3.23. Scénario « Débloquer un utilisateur » 14](#_Toc419315917)

[1.3.24. Scénario « Supprimer un utilisateur » 14](#_Toc419315918)

[1.3.25. Scénario « Ajouter une carte » 14](#_Toc419315919)

[1.3.26. Scénario « Supprimer une carte » 15](#_Toc419315920)

[1.3.27. Scénario « Editer une carte » 15](#_Toc419315921)

[1.4. Protocole d’échange entre le client et le serveur 16](#_Toc419315922)

[1.5. Ebauche des modèles de domaine 19](#_Toc419315923)

[1.5.1. Modèle de domaine serveur 19](#_Toc419315926)

[1.5.2. Modèle de domaine côté client 21](#_Toc419315927)

[1.6. Données persistantes 22](#_Toc419315928)

[1.6.1. Technologies 22](#_Toc419315930)

[1.6.2. Base de données 22](#_Toc419315931)

[1.7. Règles de définition de compte utilisateur 23](#_Toc419315932)

[1.7.1. Définition de nom d’utilisateur 23](#_Toc419315933)

[1.7.2. Définition de mot de passe 23](#_Toc419315934)

Table des figures

[Figure 1 : Diagramme général de contexte 7](#_Toc419315546)

[Figure 2 : Ebauche du modèle de domaine côté serveur 19](#_Toc419315547)

[Figure 3 : Ebauche du modèle de domaine côté client 21](#_Toc419315548)

[Figure 4 : Modèle conceptuel de la base de données 22](#_Toc419315549)

1. Cahier des charges

Cette partie du document présente le cahier des charges du projet. Celui-ci a pour but de décrire dans un premier temps le contenu du projet : objectifs, utilisation et règles y sont notamment présentés. Cette description est suivie de la mise en évidence des partages des responsabilités entre l’application cliente et l’application serveur. Des cas d’utilisation sont ensuite introduits en détails. Des informations relatives au protocole d’échange, à la base de données utilisées et au modèle de l’applicatif concluent cette partie avec la planification du projet.

* 1. Descriptif de l’applicatif

Cette section fait office de descriptif de l’applicatif. Les objectifs de base du projet y sont présentés. Quelques contraintes notables y sont mentionnées. Une explication quant à l’utilisation de l’applicatif y est décrite. Les règles du jeu constituant le sujet de l’application y sont détaillées.

* + 1. Objectifs de base

Les points suivants constituent les objectifs de base du projet :

* Disputer une partie de Scotland Yard.
* Gérer un ensemble de parties de Scotland Yard.
* Communiquer par le biais d’un tchat durant une partie.
  + 1. Contraintes

Ci-dessous figure une liste de contraintes préliminaires fonctionnelles :

* Une partie doit avoir au minimum deux joueurs et au maximum six joueurs.
* Un joueur doit endosser le rôle de Mister X, les autres joueurs constituent l’équipe de détectives.
* Chaque joueur doit jouer sur une machine différente.
  + 1. Utilisation de l’applicatif

L’applica tif se présente sous deux formes :

* Une version cliente.
* L’application cliente se présentera sous la forme d’un fichier jar. L’exécution de celui-ci ouvrira une fenêtre de connexion. Celle-ci permettra au client de se connecter au serveur puis d’accéder à la plateforme du jeu.
* Une version serveur
* L’application serveur se présentera sous la forme d’un fichier jar. L’exécution de celui-ci ouvrira une fenêtre de gestion du serveur dans laquelle le serveur à proprement parler pourra être démarré. L’application disposera également d’une partie lui permettant d’interagir et de manipuler une base de données locale devant stocker des données persistantes.
  + 1. Règles du jeu

Cette partie se veut être une présentation détaillée des règles du jeu Scotland Yard allant faire l’objet d’une implémentation dans l’application client-serveur.

#### Préparatifs du jeu

Les joueurs décident de celui qui endossera le rôle de Mister X. Il leur est fort conseiller de choisir de préférence un participant ayant les nerfs solides. Les joueurs restants constitueront l’équipe des détectives.

#### Distribution du matériel

Les préparatifs du jeu seront suivis par la distribution du « matériel » du jeu. Ainsi, chaque type de joueur se verra distribuer d’une certaine quantité de ressources allant être utilisées durant la partie :

* Mister X se verra attribuer de :
* Le pion incolore.
* Le tableau de parcours.
* 4 tickets de taxi.
* 3 tickets de bus.
* 3 tickets de métro.
* Les 2 cartes « coup double ».
* Autant de tickets noirs qu’il y a de détectives participant au jeu.
* Chaque détective se verra attribuer de :
  + Un pion de couleur.
  + 10 tickets de taxi.
  + 8 tickets de bus.
  + 4 tickets de métro.

#### Placement de départ des pions des joueurs

Afin de déterminer les points de départ des joueurs, les cartes de départ sont mélangées. Chaque joueur, Mister X y compris, tire ensuite une carte de départ. Puis les détectives placent leurs pions sur les cases portant le numéro de la carte de départ obtenue.

Il est important de noter que seul Mister X s’abstient de tout placement de pion. Il conserve donc le secret le plus absolu quant à son point de départ.

#### Déroulement du jeu

Mister X ouvre le bal, puis c’est aux détectives de jouer à tour de rôle, ce qui constitue un tour complet. Chacun doit payer sa course en remettant le ticket correspondant. Mister X inscrit ses déplacements sur le tableau de parcours. Quant aux détectives, ils font avancer leurs pions sur le plateau de jeu.

**Progression des pions**

Chaque coup correspond à un trajet en taxi, en métro ou en bus d’un arrêt à un autre le long du tracé en couleur. La course doit être payée avec le ticket de couleur correspondant. Chaque point représente un arrêt indiquant le(s) moyen(s) de transport y faisant halte. Les points à plusieurs couleurs représentent les arrêts de plusieurs moyens de transport. Là-bas, les joueurs peuvent, au coup suivant, changer de moyen de transport.

**Les coups de Mister X**

Mister X joue généralement ses coups dans le secret le plus absolu. Pour cela, il note en cachette, dans la fenêtre adéquate du tableau de parcours, le numéro de la case sur laquelle il voudrait se rendre. Il cache le numéro sous le ticket correspondant. A présent, les détectives connaissent le moyen de transport emprunté par Mister X, mais ignorent sa destination.

**Premier coup**

Au premier coup, Mister X choisit un arrêt relié à son point de départ. Il note le numéro de cet arrêt dans la première fenêtre de son tableau de parcours et le cache au moyen du ticket correspondant. La couleur de ce ticket doit en effet correspondre à la ligne de couleur reliant le point de départ à ce premier arrêt.

Au prochain tour, il avance son pion en se rendant de l’arrêt noté au prochain arrêt de son choix, et ainsi de suite. Pour connaître les autres possibilités offertes à Mister X pour échapper à ses poursuivants, se reporter au paragraphe « Coups spéciaux de Mister X ».

**Les coups des détectives**

Après Mister X, c’est aux détectives de jouer à tour de rôle, toujours dans le sens des aiguilles d’une montre. Chacun remet un ticket et place son pion sur l’arrêt suivant du moyen de transport choisi par lui. Mister X se voit remettre les tickets utilisés (autrement dit le détective doit donner son ticket à Mister X).

Il est important de noter que, pour chaque moyen de transport, les détectives ne disposent que d’un nombre limité de tickets. Dès qu’un détective a consommé sa provision de tickets, il ne peut plus emprunter le moyen de transport en question. Mister X doit avoir connaissance des tickets que chaque détective a en sa possession, afin qu’il sache à tout moment où il peut encore être traqué.

**Apparition de Mister X**

Mister X se manifeste à intervalles réguliers, plus précisément à son 3ème, 8ème, 13ème, et 18ème tour. Les endroits où il doit faire son apparition sont indiqués par une fenêtre plus grande sur le tableau de parcours. Il continue à enregistrer ses déplacements sur le tableau de parcours en cachant chaque fois le numéro au moyen du ticket correspondant. Puis il place son pion à l’arrêt où il est arrivé.

**Coups spéciaux de Mister X**

Les points suivants constituent les coups spéciaux pouvant être portés par Mister X :

* Coup double
  + Mister X peut, en un seul et même coup, parvenir deux arrêts plus loin en combinant astucieusement les moyens de transport. Il note les deux arrêts et dépose deux tickets. La carte « coup double » est mise de côté. Si le premier coup le conduit à une station d’apparition, il doit se manifester, mais disparaît dès le deuxième coup
* Tickets noirs
  + Mister X peut à tout moment remettre un ticket noir à la place d’un ticket normal, ce qui lui permet d’emprunter n’importe quel moyen de transport. Les détectives ignorent alors le moyen de transport utilisé par Mister X.

#### Fin du jeu

Mister X doit se manifester au moment où un détective arrive, avec son pion, sur l’arrêt servant momentanément de repaire à Mister X. La victoire revient alors aux détectives, autrement dit à l’équipe.

Si, au contraire, Mister X parvient à rester caché jusqu’au 24ème coup ou que les détectives ne peuvent plus progresser, c’est lui qui remporte la partie.

**Règles complémentaires**

Les points suivants constituent des règles complémentaires ne devant pas être négligées :

* Il ne peut jamais y avoir deux pions de détective sur un seul et même arrêt.
* Il est possible de faire marche arrière au prochain coup, à condition bien sûr de posséder le ticket correspondant.
* Les détectives n’ont pas le droit de s’échanger des tickets entre eux. Ils doivent chacun se satisfaire de leur propre réserve.
* L’ordre du jeu des détectives doit toujours être le même (c’est-à-dire à tour de rôle dans le sens des aiguilles d’une montre). Dans certaines situations délicates, le respect de cet ordre peut sauver la vie de Mister X.
* Si un détective ne peut plus avancer, les autres détectives sont toujours aux trousses de Mister X. Le détective en question deviendra un spectateur de la partie (donc il ne joue plus) jusqu’à qu’elle soit terminée.
* Un déplacement est obligatoire aussi longtemps que le joueur soit en mesure de se déplacer de la situation où il se trouve i.e. en usant d’un ticket adéquat.
  1. Partage des responsabilités

Cette section présente la répartition des responsabilités entre l’application cliente et l’application serveur. Il s’agit de décrire le rôle de chacune, leur interconnexion et leur interaction dans le déroulement du jeu.

1. 2. 1. Client

La version cliente de l’application permet à tout un chacun de se connecter à une machine distante faisant office de serveur. Pour ce faire, un client doit spécifier les données du serveur distant d’une part i.e. le numéro de port et son adresse ip. D’autre part, il doit fournir ses informations de login i.e. nom d’utilisateur et mot de passe. La tentative de connexion sera réalisée par l’envoi d’un message au serveur. Dans l’éventualité où il ne disposerait pas de compte utilisateur, il devra en créer un en établissant une connexion avec le serveur distant. Cet établissement de connexions se fera là aussi par le biais d’un message.

L’accès à la plateforme de parties n’est possible que par la disposition d’un compte utilisateur. Cette plateforme permet à tout utilisateur de créer une partie, de rejoindre une partie, de consulter ses statistiques de jeu ou d’éditer son compte utilisateur. Elle tâchera d’afficher sous la forme d’une liste l’ensemble des parties crées ou en cours de jeu. Cette liste sera mise à jour par la réception de messages en provenance du serveur.

La création d’une partie consiste en la limitation du nombre de joueurs variant de deux à six et le choix de la carte de jeu. Les choix disponibles auront été communiqués au préalable par le serveur qui aura interrogé une base de données. Il incombe à l’utilisateur hôte de la partie de définir ces informations. La création de cette dernière n’est possible que si le serveur n’est pas plein. L’application cliente enverra donc un message au serveur lui signalant une demande de création de partie. Celui-ci répondra positivement ou négativement suivant son état. L’utilisateur peut soit tenter sa chance à nouveau soit revenir à la plateforme des parties.

Une partie crée, l’utilisateur hôte se met en attente d’utilisateurs jusqu’à ce qu’un nombre suffisant convienne. Chaque utilisateur rejoignant une partie est appelé à désigner Mister X par le biais d’un vote. Son choix est communiqué au serveur. Dès le moment où chaque utilisateur a voté, l’hôte peut lancer la partie. Un message est communiqué au serveur. Avant de faire débuter le jeu, le serveur comptabilise les votes de chaque joueur devant désigner Mister X. En cas d’égalité dans les votes, il désignera lui-même de manière aléatoire, parmi les finalistes des votes, l’utilisateur allant endosser le costume du fameux bandit. Cette information est ensuite communiquée aux utilisateurs. Le jeu peut ensuite débuter.

La partie se déroulera selon les règles du jeu établies. Chaque joueur jouera à tour de rôle. Ceux-ci seront en mesure de communiquer par le biais d’un chat durant toute la partie indépendant de leur tour de jouer. Les communications entre joueurs seront gérées par le serveur. Celui-ci veillera à transmettre les messages aux bons destinataires dans l’éventualité où certains auraient été explicitement exclus par l’émetteur.

Durant tout le déroulement du jeu, le client devra veiller au bon respect des règles du jeu. Lors de chaque message envoyé par un joueur, il sera attendu de lui qu’il contrôle que celui-ci aille dans la logique du jeu.

La main du jeu sera donnée par le serveur. Il veillera à contrôler l’état du jeu, à transmettre les informations de chaque tour à chaque joueur et de traiter les événements particuliers du jeu selon les spécificités des règles de celui.ci. En cas de fin de partie, le serveur communiquera en informera les joueurs. Il leur communiquera le camp victorieux. Ceux-ci pourront quitter la partie et revenir dans la plateforme des parties.

La possibilité qu’un joueur quitte la partie en cours de jeu devra être gérée par le serveur. Celui-ci en informera les autres joueurs. Dans le cas où il s’agirait d’un détective, la partie se poursuivra. Dans l’hypothèse où le joueur en question serait Mister X, la partie serait terminée. Le départ de Mister X serait considéré comme étant un abandon et la victoire reviendrait donc à l’équipe de détectives. Il serait également envisageable que l’hôte de la partie décide de quitter la partie. Le serveur en prendrait note et en informerait les joueurs restants. Cet événement conduirait à arrêter automatiquement la partie.

La plateforme des parties permettra non seulement de disposer d’un inventaire des parties en cours mais également de pouvoir accéder aux statistiques de jeu. Un utilisateur pourra donc disposer d’informations sur le nombre de parties qu’il aura disputé, le nombre qu’il aura remporté et le nombre qu’il aura perdu. La plateforme permettra également à tout utilisateur de pouvoir éditer son compte utilisateur. Cette édition ne portera que sur son mot de passe. Ce dernier pourra être redéfini à condition bien sûr de respecter les règles de définition de mot de passe.

La déconnexion d’un client du serveur aura pour conséquence l’envoi d’un message à celui-ci. Le serveur en prendra note et réagira de manière adéquate selon l’engagement d’un utilisateur dans une partie.

* + 1. Serveur

La version serveur de l’application permettra à tout un chacun de disposer de son propre serveur du jeu. Une personne quelconque aura donc la possibilité d’héberger un certain nombre de parties sur sa machine. Celle-ci devra néanmoins disposer de configurations matérielles suffisamment poussées de manière à pouvoir traiter une charge de travail conséquente.

Contrairement à l’accoutumée, l’application serveur disposera d’une interface graphique utilisateur. Celle-ci, destinée à un administrateur du serveur, permettra de définir le port sur lequel le serveur écoutera les requêtes des éventuels clients. Il aura également la possibilité de bloquer ou de supprimer des utilisateurs, d’ajouter et de supprimer des cartes de jeu. L’application serveur interagira étroitement avec une base de données qui lui sera locale. La machine hôte devra donc disposer en conséquence d’outils lui permettant de mettre en service une base de données MySQL. Ainsi, les utilisateurs de l’application serveur voulant mettre en service un serveur du jeu le pourront en faisant de leur propre machine une machine dédiée au serveur. Il sera donc tout à fait possible que plusieurs serveurs soient disponibles de par le monde. La connaissance de ces serveurs nécessite de connaître la machine dédiée.

Les points suivants constituent les différents rôles qu’endossera le serveur :

* Gérer les connexions clientes.

Le serveur devra être en mesure de gérer les connexions provenant de clients distants. Concrètement, il s’agira de traiter les authentifications de client avant tout accès à la plateforme de parties et à réagir de manière cohérente à toute déconnexion de client lors de n’importe quelle situation, aux conséquences qu’elle peut impliquer

* Transmettre la liste de parties hébergées sur le serveur.

La liste de parties en cours pouvant évoluer à tout moment suivant la création, la suppression ou l’arrêt d’une partie, son nouvel état devra être communiqué régulièrement aux clients distants. Il incombe au serveur de l’application d’en prendre la responsabilité.

* Gérer les parties inter clients.

Le serveur devra gérer minutieusement les différentes parties pouvant se dérouler simultanément. Il portera la casquette de gestionnaire de parties. Il devra veiller à la bonne création d’une partie, à la bonne la synchronisation des joueurs pour une partie donnée et à la bonne initialisation du jeu.

* Gérer les communications inter clients.

Durant une partie, les clients pourront s’échanger des messages. Le serveur fera office de passerelle de communication. Chaque message envoyé par un client sera transmis au serveur. Celui-ci se chargera de le retransmettre à tout le monde. Il veillera toutefois à respecter les éventuelles contraintes émises par l’émetteur. En effet, un client pouvant spécifier les destinataires du message, le serveur devra en conséquence le remettre aux utilisateurs concernés.

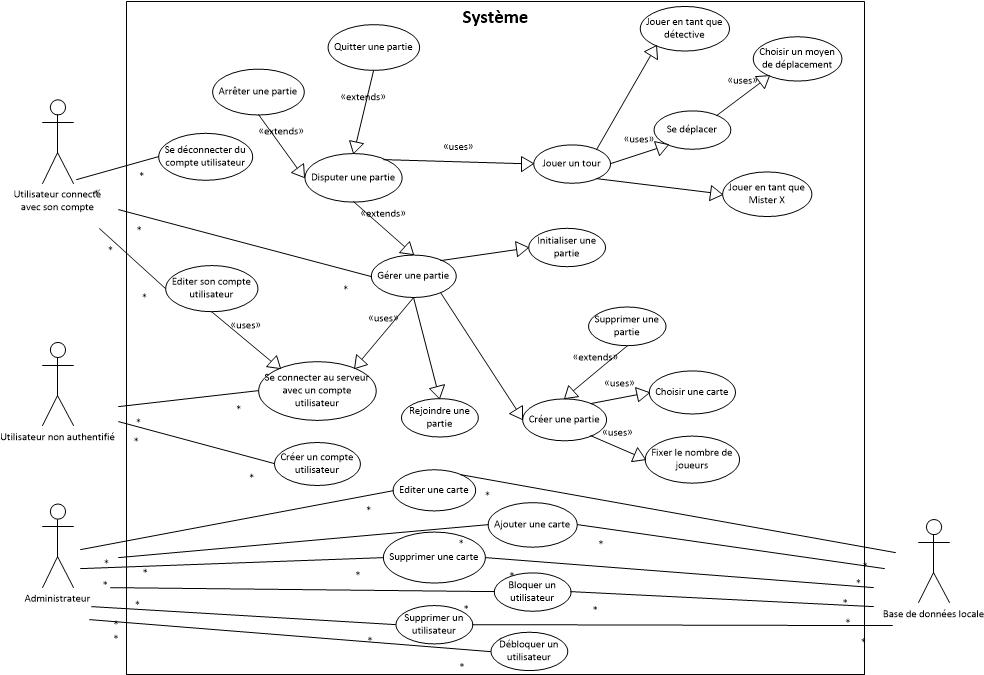
* Traiter les requêtes clientes destinées à la base de données.

Certaines actions exécutées par les clients nécessiteront des manipulations ou des interrogations de la base de données du jeu. Ces traitements seront réalisés par le bais de requête à destination du serveur. Ce dernier les analysera, les traitera et les transmettra à la base de données. La réponse de celle-ci sera communiquée aux clients par le serveur.

De manière générale, la tâche du serveur sera d’appliquer et de faire appliquer le protocole d’échange avec chaque client. Son rôle portera donc dans la vérification des messages envoyés par un client, par une réaction de sa part correspondant à la procédure du protocole et par l’envoi d’une réponse adéquate allant mettre à jour l’état dans lequel se trouve le client.

* 1. Cas d’utilisation

Cette section fait office de conception préliminaire du projet sous la forme de cas d’utilisation. Le diagramme général de contexte du système Scotland Yard y est exposé. Les acteurs interagissant avec le système y sont énumérés et présentés. Les scénarios des cas d’utilisations y sont détaillés.

* 1. 1. Diagramme général de contexte

Ci-dessous figure le diagramme général de contexte du système Scotland Yard :

Figure 1 : Diagramme général de contexte

* + 1. Description des acteurs

Ci-dessous figure la liste des acteurs du système auxquels sont associés des objectifs :

|  |  |
| --- | --- |
| **Acteur** | **Objectifs** |
| Utilisateur authentifié | * Gérer une partie. * Disputer une partie. |
| Utilisateur non authentifié | * Se connecter avec son compte utilisateur. |
| Administrateur | * Gérer une carte. * Gérer un utilisateur. |

* + 1. Scénario « Se connecter au serveur avec un compte utilisateur »

Connexion au serveur avec un compte utilisateur

**Scénario principal :**

1. L’utilisateur saisit son nom d’utilisateur.
2. L’utilisateur saisit son mot de passe.
3. L’utilisateur spécifie l’adresse ip du serveur.
4. L’utilisateur spécifie le numéro de port du serveur.
5. L’utilisateur se connecte au serveur avec son compte utilisateur.

**Scénarios d’erreur :**

Opération 5 : *Si le serveur est injoignable*

1. L’utilisateur en est informé.
2. L’utilisateur vérifie les paramètres de connexion au serveur.
3. L’utilisateur tente une nouvelle connexion.

Opération 5 : *Si le login est incorrecte*

1. L’utilisateur en est informé.
2. L’utilisateur vérifie les données de son login.
3. L’utilisateur tente une nouvelle connexion.

Opération 6 : *Si l’utilisateur est bloqué*

1. L’utilisateur en est informé.
   * 1. Scénario « Créer un compte utilisateur »

Création d’un compte utilisateur

**Scénario principal :**

1. L’utilisateur spécifie son nom d’utilisateur.
2. L’utilisateur spécifie son mot de passe.
3. L’utilisateur confirme son mot de passe.
4. L’utilisateur crée son compte.

**Scénarios d’erreur :**

Opération 1 : *Si le nom d’utilisateur ne respecte pas les règles de définition de nom d’utilisateur*

1. L’utilisateur en est informé.
2. L’utilisateur modifie son nom d’utilisateur.
3. L’utilisateur tente une nouvelle création de compte.

Opération 2 : *Si le mot de ne passe ne respecte pas les règles de définition de mot de passe*

1. L’utilisateur en est informé.
2. L’utilisateur modifie son mot de passe.
3. L’utilisateur tente une nouvelle création de compte.

Opération 4 : *Si les mots de passe ne concordent pas*

1. L’utilisateur en est informé.
2. L’utilisateur vérifie son mot de passe.
3. L’utilisateur tente une nouvelle création de compte.

Opération 5 : *Si le serveur est injoignable*

1. L’utilisateur en est informé.
2. L’utilisateur tente une nouvelle création.

Opération 5 : *Si le nom d’utilisateur est déjà utilisé*

1. L’utilisateur en est informé.
2. L’utilisateur change de nom d’utilisateur.
3. L’utilisateur tente une nouvelle création.
   * 1. Scénario « Editer son compte utilisateur »

Edition du compte utilisateur

**Scénario principale :**

1. L’utilisateur doit être authentifié : **include Se connecer au serveur avec un compte utilisateur**
2. L’utilisateur modifie son mot de passe.
3. L’utilisateur confirme le nouveau mot de passe.
4. L’utilisateur édite son compte.
5. Le serveur met à jour la base de données.
6. Le serveur confirme les modifications.

**Scénarios d’erreur :**

Opération 1 : *Si le mot de passe ne respecte pas les règles de définition de mot de passe*

1. L’utilisateur modifie le mot de passe.
2. L’utilisateur tente d’éditer son compte.

Opération 2 : *Si les mots ne passe ne concordent pas*

1. L’utilisateur reconfirme le mot de passe.
2. L’utilisateur tente d’éditer son compte.

Opération 4 : *Si la base de données est indisponible*

1. L’utilisateur en est informé.
   * 1. Scénario « Se déconnecter du compte utilisateur »

Déconnexion du compte utilisateur

**Scénario principale :**

1. L’utilisateur se déconnecte de son compte utilisateur.
2. Le serveur en est informé.
3. L’utilisateur est redirigé vers l’interface de connexion.
   * 1. Scénario « Gérer une partie »

Gestion d’une partie

**Scénario principale :**

1. L’utilisateur doit être authentifié : **include Se connecter au serveur avec un compte utilisateur**
2. L’utilisateur peut :

* créer une partie : **include Créer une partie.**
* rejoindre une partie : **include Rejoindre une partie.**
* initialiser une partie : **include Initialiser une partie.**

**Scénario alternatif :**

\*1.a *La gestion d’une partie peut découler sur la dispute d’une partie.*

Un utilisateur disputer une partie : **include Disputer une partie.**

* + 1. Scénario « Créer une partie »

Création d’une partie

**Scénario principale :**

1. L’utilisateur choisit la carte du jeu : **include Choisir une carte.**
2. L’utilisateur fixe le nombre de joueurs : **include Fixer le nombre de joueurs.**
3. L’utilisateur peut :

* supprimer une partie : **include Supprimer une partie.**
* créer la partie.

**Scénario d’erreur :**

Opération 3 : *Si le nombre de parties maximales gérées par le serveur est atteint.*

1. L’utilisateur en est informé.
2. L’utilisateur tente de créer une partie.
   * 1. Scénario « Choisir une carte »

Choix d’une carte

**Scénario principale :**

1. L’utilisateur demande à choisir une carte.
2. Le serveur lui transmet les cartes disponibles.
3. L’utilisateur choisit une carte.
   * 1. Scénario « Fixer le nombre de joueurs »

Fixation du nombre de joueurs

**Scénario principale :**

1. L’utilisateur fixe un nombre de joueurs.
2. L’utilisateur valide le nombre.

**Scénario d’erreur :**

Opération 1 : *Si le nombre de joueurs est insuffisant selon la règle*

1. L’utilisateur en est informé
2. L’utilisateur fixe un nouveau nombre d’utilisateurs.

Opération 1 : *Si le nombre de joueurs est trop grand selon la règle*

1. L’utilisateur en est informé
2. L’utilisateur fixe un nouveau nombre d’utilisateurs.
   * 1. Scénario « Supprimer une partie »

Suppression d’une partie

**Scénario principale :**

1. L’utilisateur supprime la partie.
2. Le serveur en est informé.
   * 1. Scénario « Rejoindre une partie »

Rejoint une partie

**Scénario principale :**

1. L’utilisateur sélectionne une partie.
2. L’utilisateur rejoint la partie.
3. Le serveur en est informé.
4. Le serveur informe les autres utilisateurs de la partie.

**Scénarios d’erreur :**

Opération 2 : *Si le nombre de joueurs maximal est atteint*

1. L’utilisateur en est informé.

Opération 2 : *Si la partie a déjà débuté*

1. L’utilisateur en est informé.
   * 1. Scénario « Lancer une partie »

Initialisation d’une partie

**Scénario principal :**

1. L’utilisateur hôte attend des utilisateurs.
2. L’utilisateur hôte lance la partie.

**Scénario d’erreur :**

Opération 2 : *Si le nombre de joueurs est insuffisant*

1. L’utilisateur hôte en est informé.
2. L’utilisateur hôte attend des joueurs supplémentaires.
3. L’utilisateur hôte lance la partie.
   * 1. Scénario « Disputer une partie »

Dispute d’une partie

**Scénario principale :**

1. L’utilisateur reçoit les données d’initialiation du jeu :
   1. La carte.
   2. Les stations.
   3. Les liens entre stations
   4. Les tickets de chaque joueur.
   5. La position de chaque joueur sauf celle de Mister X
2. L’utilisateur joue son tour : **include Jouer un tour**.

**Scénario alternatif :**

\*2.a *A tout moment, durant la dispute de la partie*

Un utilisateur tchate : **include Tchater.**

\*2.c *A tout moment, durant la dispute de la partie*

Un utilisateur quitte la partie : **include Quitter une partie.**

\*2.c *A tout moment, durant la dispute de la partie*

L’utilisateur hôte arrête la partie : **include Arrêter une partie.**

* + 1. Scénario « Quitter une partie »

Sortie de la partie

**Scénario principale :**

1. Le détective quitte la partie.
2. Le serveur en est informé.
3. Le serveur informe les autres joueurs.

**Scénario alternatif :**

1. Mister X quitte la partie.
2. Le serveur en est informé.
3. Le serveur informe les détectives.
4. La partie est terminée.
   * 1. Scénario « Arrêter une partie »

Arrêt de la partie

**Scénario principale :**

1. Le joueur hôte arrête la partie.
2. Le serveur en est informé.
3. Le serveur informe les autres joueurs.
4. La partie est arrêtée.
   * 1. Scénario « Jouer un tour »

Tour de jeu d’un joueur

Scénario principal :

1. Le joueur se déplace : **include Se Déplacer**
2. Le joueur joue son tour :

* En tant que détective : **include Jouer un tour en tant que détective.**
* En tant que Mister X : **include Jouer un tour en tant que Mister X.**

**Scénarios d’erreur :**

Opération 1 : *Si le joueur n’a pas la main*

1. Le joueur en est informé.
2. Le joueur attend son tour.

Opération 1 : *Si le joueur ne peut plus se déplacer*

1. Le joueur en est informé.
2. Le joueur passe son tour.
   * 1. Scénario « Jouer un tour en tant que détective »

Tour de jeu d’un détective

**Scénario principal :**

1. Le détective joue son tour.
2. Le détective met à jour ses tickets.
3. Le serveur en est informé.
4. Le serveur communique le jeu du détective aux autres joueurs.
5. Mister X reçoit le ticket dépensé par le détective.
   * 1. Scénario « Jouer un tour en tant que Mister X »

Tour de jeu de Mister X

**Scénario principal :**

1. Mister X se déplace : **include Se déplacer**
2. Mister X joue son tour.
3. Mister X met à jour ses tickets.
4. Le serveur en est informé.
5. Le serveur communique le moyen de déplacement de Mister X aux détectives.
6. Le ticket utilisé est perdu.

**Extensions :**

5a. Apparition de Mister X

Le serveur communique la position de Mister X aux détectives aux tours 3, 8, 13 et 18.

5b. Pas de communication

Mister X a joué le ticket demandant de cachant son moyen de déplacement.

5c. Double tour

Mister X a joué le ticket lui permettant de jouer un tour consécutif.

* + 1. Scénario « Se déplacer »

Déplacement dans la carte

**Scénario principale :**

1. Le joueur choisit un moyen de déplacement : **include Choisir un moyen de déplacement.**
2. Le joueur choisit une station.
3. Le joueur confirme ses choix.
4. Le serveur en est informé.

**Scénario d’erreur :**

Opération 2 : *Si la station n’est pas accessible*

1. Le joueur en est informé.
2. Le joueur choisit une autre station.
   * 1. Scénario « Choisir un moyen de déplacement »

Choix d’un moyen de déplacement

**Scénario principale :**

1. Le joueur choisit un moyen de déplacement.

Opération 1 : *Si le ticket relatif au moyen de déplacement est épuisé*

1. Le joueur en est informé.
2. Le joueur choisit un autre moyen de déplacement.
   * 1. Scénario « Bloquer un utilisateur »

Blocage d’un utilisateur

**Scénario principale :**

1. L’administrateur sélectionne un utilisateur.
2. L’administrateur bloque l’utilisateur sélectionné.

**Scénario d’erreur :**

Opération 2 : *Si la base de données est indisponible*

1. L’administrateur en est informé.
   * 1. Scénario « Débloquer un utilisateur »

Blocage d’un utilisateur

**Scénario principale :**

1. L’administrateur sélectionne un utilisateur.
2. L’administrateur débloque l’utilisateur sélectionné.

**Scénario d’erreur :**

Opération 2 : *Si la base de données est indisponible*

1. L’administrateur en est informé.
   * 1. Scénario « Supprimer un utilisateur »

Suppression d’un utilisateur

**Scénario principale :**

1. L’administrateur sélectionne un utilisateur.
2. L’administrateur supprime l’utilisateur sélectionné.

**Scénario d’erreur :**

Opération 2 : *Si la base de données est indisponible*

1. L’administrateur en est informé.

Opération 2 : *Si l’utilisateur est connecté.*

1. L’administrateur en est informé.
   * 1. Scénario « Ajouter une carte »

Edition d’une carte

**Scénario principale :**

1. L’administrateur crée une carte.

**Scénario d’erreur :**

Opération 2 : *Si les règles de création de carte ne sont pas respectées*

1. L’administrateur en est informé.

Opération 2 : *Si la base de données est indisponible*

1. L’administrateur en est informé.
   * 1. Scénario « Supprimer une carte »

Suppression d’un utilisateur

**Scénario principale :**

1. L’administrateur sélectionne une carte.
2. L’administrateur supprime la carte sélectionnée.

**Scénario d’erreur :**

Opération 2 : *Si la base de données est indisponible*

1. L’administrateur en est informé.
   * 1. Scénario « Editer une carte »

Edition d’une carte

**Scénario principale :**

1. L’administrateur sélectionne une carte.
2. L’administrateur édite la carte sélectionnée.

**Scénario d’erreur :**

Opération 2 : *Si les règles d’édition de carte ne sont pas respectées*

1. L’administrateur en est informé.

Opération 2 : *Si la base de données est indisponible*

1. L’administrateur en est informé.
   1. Protocole d’échange entre le client et le serveur

Voici le protocole d’échange qui sera utilisé pour la communication entre le client et le serveur.

La syntaxe de base est : **COMMANDE#**param1**#**param2**#**...**#**paramN

| **Commande** | **Direction** | **Description** |
| --- | --- | --- |
| PLAY\_DETECTIVE\_TURN  #pseudo  #numeroStation  #transport  #numeroPartie | Client 🡪 Serveur | Commande qui permet au client de dire qu’il effectue un déplacement (donc joue un coup) d’un détective dans la partie ayant le numéro **#numeroPartie**.  Dans ce message se trouve également le numéro de la station où le détective s’est déplacé, le transport qu’il a utilisé (afin que le serveur puisse comptabiliser les tickets) ainsi que le pseudo du joueur qui a joué.  **Exemple :** PLAY\_DETECTIVE\_TURN#toto123#156#metro#2  *« Le détective qui a le pseudo toto123 s’est déplacer sur la station 156 en prenant le métro, sur la partie possédant l’identifiant 2 »* |
| PLAY\_MISTER\_X\_TURN  #pseudo  #numeroStation  #transport  #ticketDouble  #numeroPartie | Client 🡪 Serveur | Commande qui permet au client de dire qu’il effectue un déplacement (donc joue un coup) de Mister X dans la partie ayant le numéro **#numeroPartie**.  Dans ce message se trouve également le numéro de la station où Mister X s’est déplacé, le transport qu’il a utilisé (afin que le serveur puisse comptabiliser les tickets) ainsi que le pseudo du joueur qui a joué.  Le message contient aussi une information qui dit si Mister X a utilisé un ticket pour jouer 2 fois ou bien pas.  **Exemple :** PLAY\_MISTER\_X\_TURN#boulette#32#noir#true#3  *« Mister X qui a le pseudo boulette s’est déplacé sur la station 32 en prenant un ticket noir (donc n’importe quel moyen de transport) signifiant qu’il veut cacher son moyen de transport, sur la partie possédant l’identifiant 3 »* |
| NOTIFY\_DETECTIVE\_PLAYED  #pseudo  #prochainJoueurPseudo  #ancienJoueurPseudo  #transport  #station  #numeroPartie | Serveur 🡪 Client | Commande qui permet au serveur de dire au client qu’un joueur détective s’est déplacé dans la partie qui a l’identifiant #numeroPartie.  Dans ce message, on y trouve le pseudo a qui le message est envoyé (comme ça le client peut vérifier que c’est bien pour lui), moyen de transport que le détective précédant a utilisé, le pseudo de celui qui vient de jouer, la station sur laquelle s’est rendue le détective, le pseudo du prochain joueur qui doit jouer (comme ça le client peut savoir s’il doit jouer ou non).  **Exemple :**  NOTIFY\_DETECTIVE\_PLAYED#bouboule#bouboule#yakahoma#taxi#127#3  *« Cette notification est destinée à bouboule. Bouboule doit maintenant jouer son tour et le joueur yakahoma vient de se déplacer en taxi vers la station 127 dans la partie ayant l’identifiant 3 »* |
| NOTIFY\_MISTER\_X\_PLAYED  #pseudo  #prochainJoueurPseudo  #ancienJoueurPseudo  #transport  #station  #numeroPartie | Serveur 🡪 Client | Commande qui permet au serveur de dire au client que Mister X a joué son coup et s’est déplacé dans la partie qui a l’identifiant #numeroPartie.  Dans ce message, on y trouve le pseudo a qui le message est envoyé (comme ça le client peut vérifier que c’est bien pour lui), moyen de transport que le détective précédant a utilisé, le pseudo de celui qui vient de jouer, la station sur laquelle s’est rendue le détective, le pseudo du prochain joueur qui doit jouer (comme ça le client peut savoir s’il doit jouer ou non).  A noter que le client doit afficher l’emplacement de Mister X si le paramètre station est différent de -1. Dans le cas échéant, la position de Mister X ne doit pas être affichée.  **Exemple :**  NOTIFY\_MISTER\_X\_PLAYED#mickey#donald#georges#noir#-1#5  *« Cette notification est destinée à Mickey. Donald doit maintenant jouer son tour et le joueur Georges vient de se déplacer (en cachant son moyen de transport car il utilise un ticket noir) vers une station inconnu (dû au -1), dans la partie ayant l’identifiant 3 »* |
| LAUNCH\_GAME  #pseudo  #nombreJoueurs  #carte | Client 🡪 Serveur | Commande qui permet à un client de lancer une partie sur le serveur.  Elle contient le nombre de joueurs de la partie ainsi que la carte qui auront été choisi par celui qui lance la partie, ainsi que le pseudo de l’utilisateur qui veut lancer une partie.  **Exemple :**  LAUNCH\_GAME#aladdin#5#londres  *« Demande de lancement d’une nouvelle partie de la part de Aladdin avec 5 joueurs et pour jouer sur la carte de Londres »* |
| ACCEPT\_CREATION\_GAME  #pseudo  #numeroPartie | Serveur 🡪 Client | Suite à une demande de lancement d’une partie (commande LAUNCH\_GAME) le serveur renverra ce message à l’utilisateur ayant le pseudo #pseudo et le numéro de la partie qui a été créer sur le serveur, afin que le client connaisse le numéro de cette partie.  **Exemple :**  ACCEPT\_CREATION\_GAME#bouboule#4  *« Notification à destination de bouboule. Le serveur a accepté la demande d’une nouvelle partie bouboule et cette partie a le numéro 4 »* |
| REJECT\_CREATION\_GAME  #pseudo | Serveur 🡪 Client | Permet simplement de dire au client ayant le pseudo #pseudo que la création de la partie est refusée.  **Exemple :**  REJECT\_CREATION\_GAME#tintin  *« Notification pour Tintin. La demande de création d’une partie a été refusée »* |
| IMAGESIZE  #taille | Serveur 🡪 Client | Permet au serveur de transmettre la taille d’une image à un client afin qu’il connaisse la taille de l’image à recevoir. |
| STARTGAME  #nompartie | Client 🡪 Serveur | Permet à l’hôte de la partie de dire au serveur de démarrer la partie car elle est prête au niveau des joueurs. |
| AUTHENTICATE  #username  #password | Client 🡪 Serveur | Permet à un client de s’authentifier sur le serveur avec un **username** et un **password** |
| UNAUTHENTICATE | Client 🡪 Serveur | Permet à un client de de déconnecter de son compte sur le serveur. |
| CREATEACCOUNT  #username  #password | Client 🡪 Serveur | Permet à un client de créer un compte utilisateur sur le serveur avec un **username** et un **password**. |
| REQUESTMAPLIST | Client 🡪 Serveur | Permet à un client de demander au serveur la liste des noms des cartes disponibles |
| REQUESTGAMELIST | Client 🡪 Serveur | Permet à un client de demander au serveur la liste des parties en cours |
| REQUESTPLAYERLIST  #nompartie | Client 🡪 Serveur | Permet à un client de demander la liste des joueurs d’une partie (portant le nom **nompartie**) |
| EDITACCOUNT  #newUsername  #newPassword | Client 🡪 Serveur | Permet à un client d’éditer les informations de son compte soit sont username et son password. |

Pour le reste des informations utiles à transmettre (cartes, stations, etc…) ceci se fait par sérialisation JSon (voir le chapitre concerné).

* 1. Ebauche des modèles de domaine

Cette partie du document présente les ébauches des modèles de domaine devant représenter l’application. Il se divise en deux modèles de domaine distincts. Le premier a pour but de représenter le serveur. Le second a pour vocation d’illustrer le client.

* 2. 1. Modèle de domaine serveur

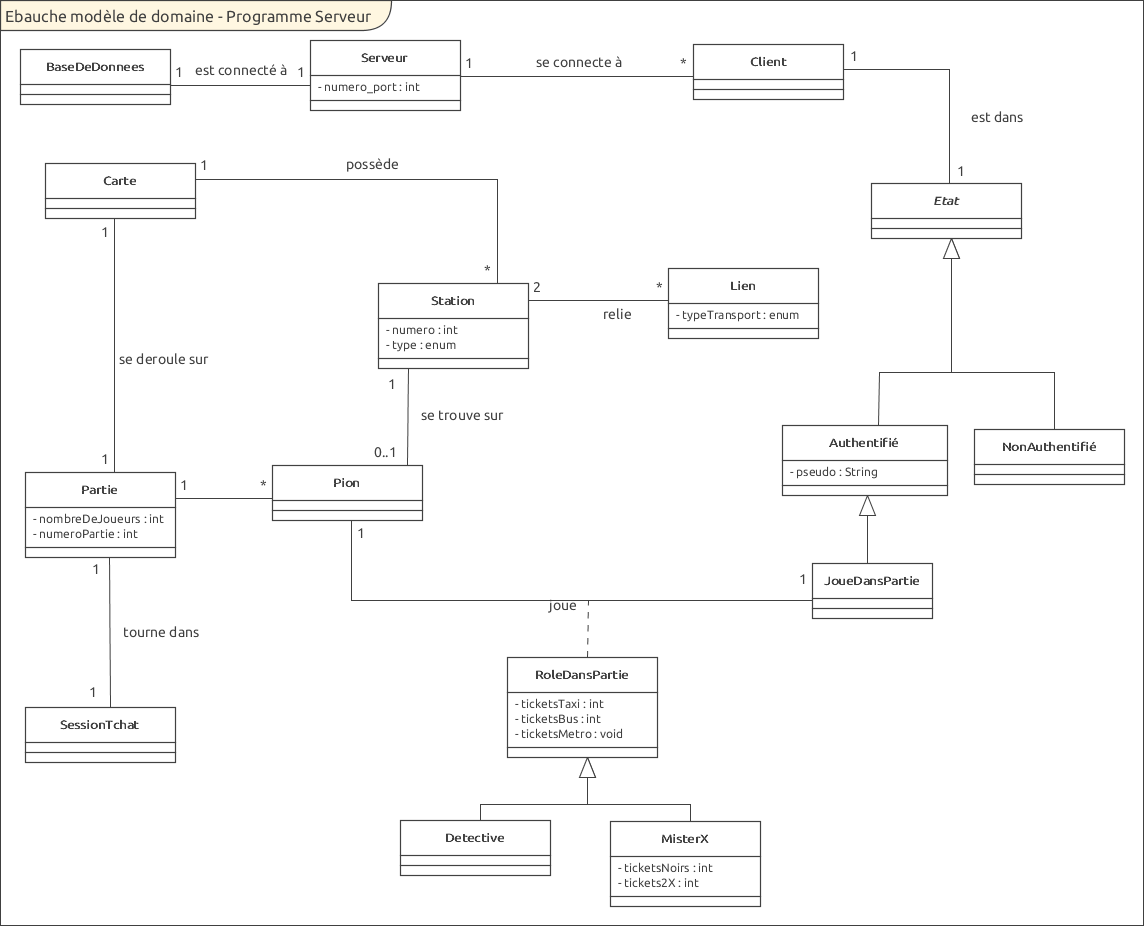
Ci-dessous figure une ébauche du modèle de domaine côté serveur pour lequel aucune contrainte d’intégrité n’a été définie temporairement :

Figure 2 : Ebauche du modèle de domaine côté serveur

*Le serveur est connecté à une base de données. C’est elle qui lui fournira les informations sur des cartes ou bien des comptes utilisateurs.*

*Plusieurs clients peuvent se connecter à un serveur. Chaque client connecté aura un état vis-à-vis du serveur, qui peut être l’un des états suivants :*

* *Non authentifié*
  + *Signifie que le client n’est pas authentifié sur un compte utilisateur*
* *Authentifié*
  + *Signifie que le client est authentifié sur un compte utilisateur*
* *En train de jouer dans une partie*
  + *Signifie que le client est authentifié est qu’il est en train de jouer dans une partie*

*Lorsqu’un client se trouve dans une partie, il est rattaché à un pion qui le caractérise comme jouant soit le rôle de MisterX, soit le rôle d’un détective. Que ça soit un détective ou Mister X, il possède un certain nombre de tickets de taxi, bus, métros pendant la partie.*

*Lors du déroulement de la partie, un pion se trouve sur l’une des stations se trouvant sur la carte. Une partie se déroule sur une seule carte.*

*Entre deux stations A et B, il peut y avoir un lien qui symbolise le fait qu’il est possible d’aller de A à B ou de A à B. Ce lien est caractérisé par un type de moyen de transport (par exemple Bus qui signifierait que ce lien permet d’aller de A à B en bus…).*

*Finalement, il y a une session de tchat qui est lancée dans chaque partie, afin que les joueurs puissent discuter. Chaque partie possède sa propre session de tchat.*

*Comme il s’agit d’une ébauche d’un modèle de domaine, ceci ne constitue en rien le diagramme de classes définitif, où certaines classes nécessaires pourraient être ajoutées lors de l’implémentation.*

* + 1. Modèle de domaine côté client

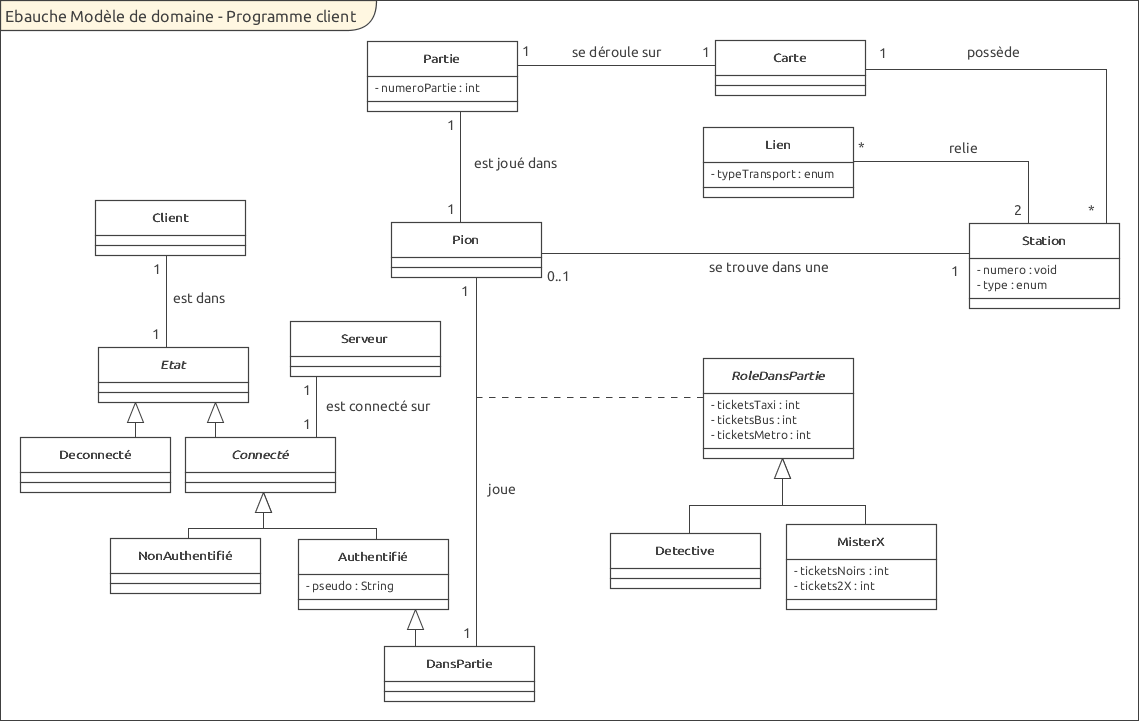
Ci-dessous figure une ébauche du modèle de domaine côté client pour lequel aucune contrainte d’intégrité n’a été définie temporairement :

Figure 3 : Ebauche du modèle de domaine côté client

*Un client a plusieurs états possibles qu’il peut prendre à la fois. Ces différents états sont les suivants :*

* *Déconnecté*
  + *Signifie que le client n’est pas du tout connecté à un serveur.*
* *Connecté*
  + *Signifie que le client est connecté à un serveur.*
* *Connecté et non authentifié*
  + *Signifie que le client est connecté mais qu’il n’est pas authentifié sur son compte utilisateur.*
* *Connecté et authentifié*
  + *Signifie que le client est connecté et qu’il est authentifié sur son compte utilisateur.*
* *Dans une partie*
  + *Signifie que le client est connecté, authentifié et qu’il est en train de jouer une partie.*

*Concernant les autres entités, voir les explications du modèle de domaine du serveur.*

*Comme il s’agit d’une ébauche d’un modèle de domaine, celui-ci ne constitue en rien le diagramme de classes définitif, où certaines classes nécessaires pourraient être ajoutées lors de l’implémentation.*

* 1. Données persistantes

L’application manipulera des données persistantes. Celles-ci pourront représenter d’une part des informations liées à des comptes utilisateurs et aux statistiques leurs étant associées. D’autre part, des données relatives aux cartes du jeu seront également stockées.

* 1. 1. Technologies

Le stockage des informations persistantes sera réalisé par le biais d’une base de données. La technologie MySql sera utilisée à cet effet. La communication des données de la base de données du serveur au client sera réalisée à l’aide de Json.

* + 1. Base de données

Cette section introduit la base de données allant être utilisées par l’application. Celle-ci interagira directement avec le serveur.

L’application cliente désirant interroger la base de données devra passer par le serveur. Celui-ci se verra confier la tâche de transmettre la requête du client et la réponse de la base de données.

Les réponses de la base de données que tâchera de transmettre le serveur au client seront envoyées par sérialisation.

#### Objectifs

Les points suivants constituent les objectifs que la base de données devra remplir en termes de stockage de données persistantes :

* Comptes utilisateurs
  + Les informations relatives à un compte utilisateur comprennent nom d’utilisateur et mot de passe.
* Statistiques d’utilisateurs
  + Les informations relatives aux statistiques de joueur comprennent le nombre de parties disputées, le nombre de parties remportées et le nombre de parties perdues.
* Cartes du jeu
  + Les informations relatives aux cartes du jeu comprennent une image constituant l’arrière-plan de la carte et un ensemble de stations.

#### Modèle conceptuel

Ci-dessous figure le modèle conceptuel de la base de données de l’application :

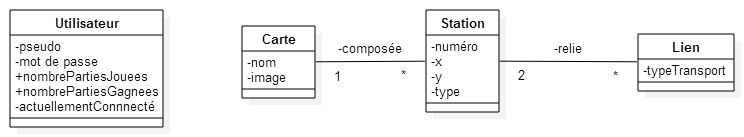


Figure 4 : Modèle conceptuel de la base de données

#### Contraintes d’intégrité

Les points suivants constituent les contraintes d’intégrité du modèle :

* Le pseudo d'un utilisateur ne doit pas contenir de # (pour ne pas corrompre le protocole)
* Le mot de passe d'un utilisateur ne doit pas contenir de # (même raison)
* Le nom d'une carte ne doit pas contenir de # (même raison)
* Un lien doit relier 2 stations différentes, et en plus ces deux stations doivent se trouver sur la même carte.
* Si un lien existe entre une station A et une station B, son type de transport doit être différents des autres liens reliant A et B
* Il ne peut pas y avoir deux stations avec le même numéro dans une carte
* Les coordonnées x et y d'une station sont positives ou nulles
* Le type d'une station peut prendre uniquement les valeurs : TAXI TAXIBUS TAXIBUSMETRO
* Le type de transport d'un lien peut prendre uniquement les valeurs : TAXI BUS METRO
* Un lien de type BUS peut relier uniquement deux stations ayant comme type : TAXIBUS ou TAXIBUSMETRO
* Un lien de type METRO peut relier uniquement deux stations ayant comme type : TAXIBUSMETRO

Il est à noter que la grande majorité des contraintes énumérées sera gérée dans l’implémentation de la solution plutôt que du côté de la base de données.

#### Commentaires

Les points suivants constituent quelques explications à titre indicatives relatives au modèle conceptuel de la base de données :

* Un utilisateur est caractérisé par un pseudo et un mot de passe. Il dispose également de statistiques dès le moment où il a disputé une partie. Ces statistiques contiennent des informations sur le nombre de parties disputées et le nombre de parties remportées.
* Une carte est caractérisée avant tout par un nom et par une image. Celle-ci constitue l’arrière-plan de la carte. Une carte est également composée par un ensemble de stations. Ces dernières sont caractérisées par un numéro et des coordonnées x et y. Une station est définie par un type : StationTaxi, StationBusTaxi ou StationMetroiBusTaxi.
  1. Règles de définition de compte utilisateur

Cette partie présente les règles de définition de compte utilisateur. Il s’agit de présenter en détails les règles qui régissent la définition d’un nom d’utilisateur et d’un mot de passe utilisateur.

* + 1. Définition de nom d’utilisateur

Un nom d’utilisateur doit respecter les contraintes suivantes :

* Doit contenir entre six et vingt caractères.
* Doit contenir aucun caractère #.
* Doit contenir aucun espace.
* Peut contenir des caractères numériques.
* Peut contenir des caractères alphabétiques.
* Peut contenir des caractères spéciaux.

L’expression régulière suivante veille à faire respecter ces contraintes :

^(?=.\*.)(?!.\*[#|\s]).{6,20}$

* + 1. Définition de mot de passe

Un mot de passe utilisateur doit respecter les contraintes suivantes :

* Doit contenir entre six et vingt caractères.
* Doit contenir aucun caractère #.
* Doit contenir aucun espace.
* Doit contenir au moins un caractère numérique.
* Doit contenir au moins un caractère alphabétique minuscule.
* Doit contenir au moins un caractère alphabétique majuscule.
* Doit contenir au moins un caractère spécial.

L’expression régulière suivante veille à faire respecter ces contraintes :

^(?=.\*\d)(?=.\*[a-z])(?=.\*[A-Z])(?!.\*[#|\s]).{6,20}$