

# Cahier des charges

## 1) Contexte

Paris est une ville qui met à disposition de nombreux transports.

Nous avons conçu une application qui consiste à proposer une carte offrant un système de points avec plusieurs avantages.

Cela incite dans un premier temps les habitants de Paris à prendre d'avantage les transports en commun.

Cette initiative a pour but de diminuer la pollution et assurer une bonne circulation dans les métropoles.

## 2) Objectifs du projet

Les habitants de Paris pourront, une fois le projet mis en place, bénéficier de différents types de transport pour se déplacer au sein de l'agglomération parisienne.

5 différents types de transport seront à la disposition des usagers allant du transport en commun au transport individuel :

- Des bus
- Des tramways
- Des métros
- Des vélos
- Des trottinettes électriques

Pour utiliser ces transports, les usagers devront se munir d'une carte d'abonnement qui fonctionnera alors selon un système de points dépendant du type de transport utilisé.

Chaque point crédité sur la carte vaut la somme de 1.50€. Lors de l'acquisition du titre de transport, celui-ci est obligatoirement rechargé. Il pourra également être rechargé à tout moment depuis une borne de rechargement. Un système de points

bonus est pris en compte lors du rechargement d'une carte à partir de 10 points fonctionnant sur la base de paliers énoncés ci-dessous :

- Entre 10 et 15 points : 2 points
- Entre 15 et 20 points : 3 points
- Entre 20 et 30 points : 5 points
- Entre 30 et 45 points : 8 points
- Entre 45 et 75 points : 10 points
- Entre 75 et 100 points : 15 points
- Plus de 100 points : 20 points

De plus, chaque jour, pour fidéliser les usagers, 2 points bonus sont automatiquement crédités sur la carte de chacun d'entre-eux.

Chaque type de transport possède son propre tarif :

- Bus : 4 points
- Tram : 4 points
- Métro : 6 points
- Vélo : 3 points
- Trottinette : 2 points

Une option permettant de réserver tous les types de transports est également disponible pour offrir un prix plus avantageux : 15 points.

Le système de réservation fonctionne à la journée ; c'est-à-dire qu'un usager peut réserver un type de transport pour une journée seulement. Par exemple, s'il prend le bus tous les jours, il devra alors le réserver quotidiennement.

En cas de perte ou de vol d'une carte, un usager pourra la faire bloquer au sein du service client de notre centre de contrôle.

Pour ce qui est des employés qui s'occupent du fonctionnement des transports et notamment de la gestion de différents usagers, il pourront ajouter ou supprimer des

usagers. Ils auront également la possibilité d'afficher les informations d'un ou plusieurs usagers ou encore le bilan de la journée pour chaque type de transport.

### 3) Implémentation du projet

#### Chargement

Pour commencer, il faut d'abord charger un fichier pour pouvoir obtenir la liste des usagers avec leurs informations et pour pouvoir la mettre à jour. Nous partons du principe que chaque fois que le programme est lancé correspond à un nouveau jour. Cela signifie qu'à chaque fois que l'on lance le programme, les 2 points bonus quotidiens sont crédités sur toutes les cartes et le bilan de chacun des types de transport et remis à 0. En cas de problème avec le chargement du fichier ou encore si les tableaux des usagers sont pleins, un message d'erreur est alors affiché et la fonction retourne -1.

Les clients, dans le fichier dédié sont classés dans l'ordre croissant en fonction de leur numéro de carte composé de 5 chiffres.

#### Affichage

La fonction **affichage** parcourt les éléments des tableaux des usagers un par un afin de les afficher de façon ordonnée et esthétique en colonne. L'affichage est adapté en fonction de l'état de la carte (bloquée / débloquée). L'affichage obtenu à l'écran est fait sous forme de tableau, la taille du numéro de carte est verrouillée avec le "%5d" tout comme celui du nombre de points avec le "%3d" (qui sous-entend qu'on aura au maximum 999 points).

#### Affichage unique

La fonction **affichageUnique** fait exactement la même chose que la fonction **affichage** sauf qu'au lieu de parcourir les tableaux à l'aide d'une boucle, elle va se concentrer sur l'affichage des informations d'une seule personne en appelant la fonction **recherche** qui va transmettre via un retour l'indice de l'élément recherché

s'il a été trouvé. Si l'élément n'a pas été trouvé, la fonction **affichageUnique** va afficher un message d'erreur en se basant sur la valeur d'une variable "trouve" qui aura été transmise par **recherche** sous forme de pointeur.

## Recherche

La fonction **recherche** va parcourir le tableau `tnCarte[ ]` (correspondant au tableau dans lequel figure les numéros de carte). Elle va alors comparer chaque élément avec un numéro de carte transmis en paramètre. Si le numéro de carte recherché figure dans le tableau, la fonction renvoie sa position et fixe la valeur d'un pointeur "trouve" à 'O' afin de signaler que l'élément a été trouvé. Dans le cas contraire, la fonction renvoie sa position d'insertion et fixe la valeur de "trouve" à "N". Dans le cas où le tableau a été parcouru en entier sans résultat, la fonction renvoie -1.

## Insertion

La fonction **insertion** va d'abord commencer par demander à l'utilisateur de saisir le numéro de carte à insérer. Elle appelle ensuite la fonction **rechercher** via une variable `i` afin de savoir si la carte figure dans le tableau ou si elle peut être insérée. Si la carte en question est déjà présente dans le tableau ou qu'il n'y a plus de place dans les tableaux, la fonction affiche un message d'erreur et retourne -1. Dans le cas où l'on peut insérer la carte sans problème ; la fonction va, à partir du dernier élément des tableaux, décaler l'élément d'indice `j` à la position `j + 1` jusqu'à ce que la valeur de `j` (diminuant de 1 à chaque tour de boucle) soit égale à la valeur `i` (l'indice de la position d'insertion de la nouvelle carte). Ensuite, la fonction insère dans tous les tableaux la nouvelle carte. Le rechargement étant obligatoire lors de l'acquisition de la carte, la fonction **recharger** est appelée. La fonction retourne ensuite 0 et modifie grâce à un pointeur la taille logique des tableaux en l'incrémentant de 1.

## Suppression

La fonction **suppression** commence par demander à l'utilisateur le numéro de la carte à supprimer. Elle demande ensuite une confirmation pour vérifier que l'utilisateur n'a pas fait d'erreur. Comme la fonction **insertion**, la fonction **suppression** appelle la fonction **recherche** afin de vérifier que le numéro de carte saisi existe. Dans le cas où le numéro de carte n'est pas trouvé, la fonction affiche un message d'erreur et renvoie -1. Si la carte est trouvée, la fonction va alors, pour

tous les éléments des tableaux en partant de l'indice  $i$ , écraser l'élément d'indice  $i$  en le remplaçant par l'élément d'indice  $i + 1$  jusqu'à ce que  $i$  (qui augmente de 1 à chaque tour de boucle) soit égal à la taille logique des tableaux. La fonction, pour finir, diminue de 1 la taille logique via un pointeur.

## Afficher Bilan

La fonction nommée "**afficherBilan**" sert, comme son nom l'indique, à afficher le bilan de la journée c'est-à-dire voir le nombre d'entrée de chaque transports. Pour cela, la fonction a besoin des tableaux des transports et leurs tailles logiques que nous allons passer en paramètre. Nous allons aussi avoir besoin d'un compteur ( $i$ ) et de variables qui vont stocker le nombre total de réservation de transport. En effet, les tableaux des transports sont remplis de 0 ou de 1 indiquant si une personne donnée(via un index) a réserver ce type de transport ce jour.

On va donc utiliser une boucle pour répéter l'ajout de la valeur présente à l'index " $i$ " dans les variables stockant le nombre total de réservation de transport et cela pour chaque valeur dans le tableau puisqu'on incrémente  $i$  à la fin de cette boucle et que la condition d'arrêt est que  $i$  ne soit plus strictement inférieur à la taille logique des tableaux.

Enfin, on affiche les variables stockant le nombre d'entrée.

## Recharger

La fonction nommée "**recharger**" va servir à n'importe quel utilisateur de recharger sa carte.

Cette fonction peut être appelée de 2 fonctions différentes, soit par "**globale**" directement car l'utilisateur veut seulement recharger sa carte, soit par la fonction "**insertion**" puisque l'utilisateur doit mettre des points sur la carte créée. Pour différencier la provenance de l'appel, nous passons en paramètre de "**recharger**" un entier nommé "carte" qui soit contient le numéro de la carte créée si la fonction est lancée par "**insertion**", soit cette variable contient 0 si elle est appelée par "**globale**".

Dans le cas où "**recharger**" est appelée par globale, alors nous avons besoin de demander la carte que l'utilisateur veut recharger. Ce que l'on va demander si carte a la valeur 0 et nous allons écraser cette variable pour mettre le numéro de la carte.

Ensuite la fonction récupère l'index de la carte à recharger dans la variable "index" en appelant la fonction "**recherche**" qui va le retourner. Tant que la carte n'est pas

trouvée, c'est-à-dire que "**recherche**" renvoie "N" dans la variable "trouve", alors nous demandons à l'utilisateur de saisir une carte valide.

Ensuite, elle vérifie que la carte ne soit pas bloquée en regardant dans le tableau "tEtat" à l'index de la carte. Si la carte est bloquée alors la fonction s'arrête puisque l'utilisateur ne peut recharger une carte bloquée.

Sinon, elle affiche le nombre de points présents sur la carte et demande combien de points la personne souhaite ajouter. Elle stock la réponse dans la variable "ajout" et calcul le cout pour l'affecter dans la variable "cout". Elle appelle la fonction "**calculBonus**" pour calculer le nombre de points bonus. "recharger" affiche le coût et demande si l'utilisateur veut poursuivre son achat. Si la réponse est "N" alors l'achat est annulé. Sinon, on ajoute dans le tableau "tPoint" à l'index "index" les points achetés et les points bonus.

## Calcul du bonus

La fonction du nom de "**calculBonus**" permet d'attribuer des points bonus aux utilisateurs lors de leur achat de point.

Elle va utiliser l'entier "ajout" passé en paramètre, pour attribuer un certain nombre de points bonus par rapport au nombre de points achetés.

Si l'utilisateur achète moins de 10 points il ne reçoit aucun point bonus,

si ajout est supérieur à 10 mais inférieur à 15 alors il reçoit 2 points, si ajout est supérieur à 15 mais inférieur à 20 alors il reçoit 3 points, si ajout est supérieur à 20 mais inférieur à 30 alors il reçoit 5 points, si ajout est supérieur à 30 mais inférieur à 45 alors il reçoit 8 points, si ajout est supérieur à 45 mais inférieur à 75 alors il reçoit 10 points, si ajout est supérieur à 75 mais inférieur à 100 alors il reçoit 15 points, enfin si son achat dépasse les 100 points alors il recevra 20 points bonus.

## Réserver

Pour commencer, la fonction **réserver** demande à l'utilisateur de saisir le numéro de la carte à insérer. Elle fait ensuite appel à la fonction **recherche** via une variable "pos" qui permettra de récupérer la position de la carte dans le tableau "tnCarte" si elle existe. Si la carte n'existe pas, la fonction affiche un message d'erreur et retourne -1. Ensuite, la fonction vérifie si la carte n'est pas bloquée, si jamais elle l'est, elle affiche un message d'erreur et retourne -1. Dans le cas où tout est bon, la fonction affiche un menu de choix des types de transport. Elle traite ensuite la saisie

de l'utilisateur ; suivant le type de transport choisis, la fonction va débiter le nombre de point correspondant sur la carte en cours d'utilisation. Si jamais la carte utilisée ne possède pas assez de points, la fonction affiche alors un message d'erreur. De plus, lors de la demande de réservation d'un type de transport, la fonction vérifie que celui-ci n'a pas déjà été réservé, si c'est le cas, elle affiche également un message d'erreur. Le menu continue d'être affiché tant que l'utilisateur n'a pas demandé à en sortir.

## Enregistrement des tableaux

"**enregistrementTableaux**" permet d'écrire dans le fichier "clients.txt" tous les changements effectués au cours de la session. C'est-à-dire enregistrer les données des tableaux.

Pour cela, la fonction aura besoin des tableaux "tnCarte" "tPoint" et "tEtat" ainsi que leurs tailles logique en paramètre.

Elle aura besoin d'un compteur (i) et d'un pointeur sur le fichier clients.txt (flot). Si il y a une erreur d'ouverture du fichier alors la fonction affiche un message d'erreur et s'arrête.

Si, au contraire tout se passe bien alors elle commence une boucle qui tant que i est strictement inférieur à la taille logique alors elle écrit dans le fichier les données des tableaux à l'indice "i". Ensuite il y a une incrémentation de i et les instructions recommencent pour toutes les valeurs du tableaux.

## Gérer l'état de la carte

"**gererEtatCarte**" est la fonction qui, comme son nom l'indique va nous permettre de gérer l'état d'une carte. Elle aura besoin des tableaux "tnCarte" et "tEtat" ainsi que la taille logique de ses derniers qui seront mis en paramètre.

Tout d'abord, elle demande un numéro de carte à gérer, qui sera mis dans la variable "numeroC". Ensuite elle appelle "**recherche**" pour récupérer l'indice de la carte. La fonction regarde dans le tableau "tEtat" au même indice pour voir si la carte est bloquée.

Si la carte est bloquée alors un message est affiché pour le dire et demande si la personne veut débloquent la carte. La réponse sera stockée dans la variable "saisie". Si cette dernière équivaut à "O" alors la carte sera débloquée en changeant la valeur correspondante dans le tableau "tEtat".

Mêmes mécanismes si la carte est débloquée.

## **Globale**

La fonction globale s'occupe d'afficher un menu dans lequel l'utilisateur choisit une tâche qu'il veut faire. La fonction appelle alors les fonctions qui serviront à effectuer la tâche en question. Tant que l'utilisateur n'a pas décidé de quitter le programme, la fonction continue de lui proposer le menu.

## **Répartition du travail**

Chargement : Maël

Recherche : Johnny

Affichage : Malak

Affichage Unique : Malak

Insertion : Elsa

Suppression : Elsa

Afficher Bilan : Johnny

Réserver : Johnny

Calcul Bonus : Maël

Recharger : Maël

Enregistrement : Maël

Gérer état Carte : Johnny

Globale : Johnny