

# Transportation 2018-2019

L'objectif de ce TP est de développer un logiciel de gestion de tickets de transport dans une région. Ainsi il y aura les concepts de ticket, mais également de déplacement, de carte de transport et enfin de banque. Que du bonheur.

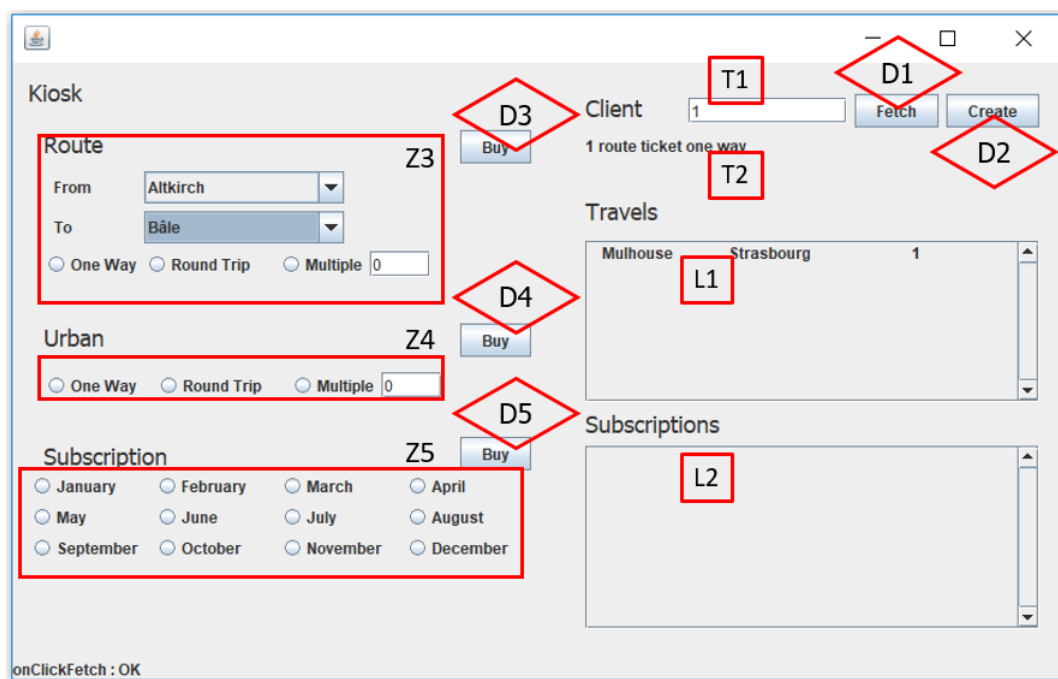
## 1. Concepts

Il y aura plusieurs applications réseaux qui auront pour vocation de proposer des services aux autres applications.

- Le kiosk représente l'application qu'un consommateur va consulter afin d'acheter ou d'approvisionner sa carte de transport.
- Le terminal représente la borne qui est dans un transport en commun et qui va décompter une unité de déplacement sur ceux présents dans la carte de transport.
- Le serveur nommé ainsi par manque d'imagination, n'a pas d'interface. Il reçoit les requêtes du kiosk et du terminal pour mettre à jour les déplacements achetés et consommés. Lors d'un achat, il est chargé de contacter la banque pour finaliser les transferts financiers.
- La banque, n'a pas d'interface, reçoit les demandes du server et valide ou invalide les achats en fonction de critères simples.

## 2. L'interface Homme Machine du kiosk

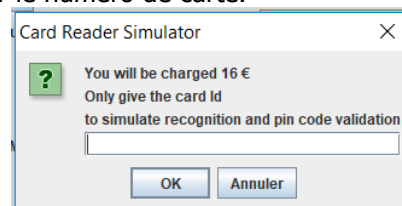
La copie d'écran ci-dessous montre l'interface du kiosk avec les éléments dynamiques marqués d'un symbole géométrique.



Décrivons le cas d'utilisation principal :

- L'utilisateur présente une carte de transport sur le lecteur, la lecture de l'identification est en principe automatique, ici l'utilisateur peut saisir l'identifiant de la carte dans la zone T1.
- Un click sur le diamant D1 va lancer une requête sur le serveur pour afficher la description de la carte, la liste des transports et des abonnements qui sont sur cette carte. Ainsi, les champs L1, L2 et T2 seront mis à jour.

- Un click sur le diamant D2 va créer une carte vierge qui pourra être utilisée par la suite, le numéro de la carte sera affiché dans T1.
- Puis notre usager souhaite charger sa carte avec des déplacements. Pour cela il va utiliser la partie gauche de l'interface. L'utilisateur aura le choix entre trois types de déplacements :
  - Libre : Z3 est la zone qui permet de fabriquer un ticket de transport avec la gare de début et d'arrivée et le nombre de trajets qu'il pourra effectuer.
  - Urbain : Z4 est un ticket avec un nombre limité de trajets dans une zone géographique restreinte.
  - Abonnement: Z5 permet d'avoir un ticket valable 1 mois avec des trajets illimités.
- Pour chacun de ces types de déplacement il y a un bouton achat spécifique : D3, D4, D5. Tous vont déclencher un processus de paiement qui débute par :
  - Une première demande au serveur qui va initier une transaction et obtenir le prix du transport demandé.
  - Le prix connu, une lecture de la carte bleue de l'utilisateur est proposée. Comme nous n'avons pas de lecteur, nous allons faire apparaître le popup ci-dessous qui simule cette lecture. Nous devrions voir le prix et nous voyons un champ texte T3 qui permet de saisir le numéro de carte.

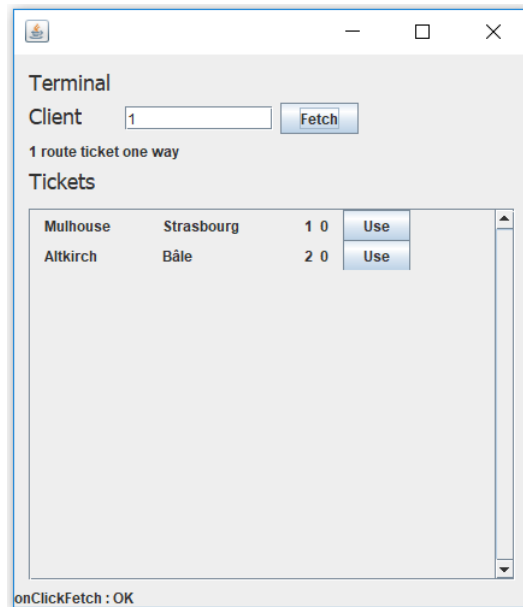


- Le bouton OK permet de cacher la saisie du code pin et sa validation et donc il permet de confirmer le paiement et de faire une validation (locale) de la transaction. La validation finale est du ressort de la relation serveur/banque.
  - Le bouton Annuler permet de simuler l'abandon ou l'échec de la transaction.
- Par la suite, il l'on souhaite 'vérifier' le contenu de la carte de transport, il suffit de refaire les deux premiers points de ce cas d'utilisation.
- Si l'utilisateur ne présente pas de carte mais achète un déplacement alors il aura une carte automatiquement. Votre difficulté sera de trouver son numéro car rien ne permet de connaître, mais je pourrais vous offrir la mise à jour du champ T1.

Après une première utilisation de l'interface, vous pouvez avoir cet aspect sur l'écran.

From	To	Count
Mulhouse	Strasbourg	1
Altkirch	Bâle	2

Lorsque l'utilisateur se déplace dans un moyen de transport, il présente sa carte à une borne qui va enregistrer le déplacement. Comme, il peut y avoir plusieurs types de déplacement sur une carte, je vous propose de choisir le bon à l'aide de l'interface ci-dessous:



A nouveau, vous donnez le numéro de la carte dans le champ texte et le click sur le bouton fetch permet de récupérer la liste des déplacements sur la carte. L'affichage est comparable à celui du kiosk mais différent (ou est-ce différent mais comparable ?)

Un click dans le bouton Use associé à un déplacement réalise l'opération d'utiliser un déplacement sous réserve qu'il y en a encore suffisamment. Les deux nombres dans chaque ligne donnent respectivement le total autorisé et le nombre de déplacement actuellement réalisé. Après utilisation le 0 devrait passer à 1. Wahou...

### 3. Les aides

Dans le modèle métier du serveur se trouve toutes les informations sympas pour faciliter le développement. Vous y trouverez plusieurs cartes de transport déjà créées et approvisionnées. Ce sont celles dont le numéro commence par 1 à 7.

Il y aura tout l'éco système indispensable pour créer des cartes vierges et des tickets, les mettre en attente de paiement, et les attacher à une carte.

Le serveur pourra aussi interroger la banque (il n'y en a qu'une avec des comptes différents pour quelques consommateurs qui ont tous une carte de crédit). Comme seul la carte de crédit est importante, il suffit de savoir qu'il y a plusieurs comptes de créés (il n'y a pas de facilité pour créer de nouveaux comptes ni de nouvelles cartes bleues) avec des comportements très spéciaux.

La carte de crédit :

- 1 : est une carte qui refusera toujours la transaction
- 2 : est une carte qui acceptera toujours la transaction
- 3 : est un compte qui traite la transaction normalement mais le solde est gigantesque
- 4, 5 et 6 : sont des comptes qui traitent la transaction normalement mais le solde est réduit, cela permet de vérifier le refus après quelques transactions

### 4. Les exigences

Au moins, derrière chacun des diamants il y aura une méthode qui va devoir interroger le serveur et récupérer les données, pour cela il faudra échanger via le réseau et donc un protocole que vous devez définir. Si je compte bien, il y a plusieurs méthodes à définir tant au niveau du serveur qu'au niveau du client.

Techniquement, il faudra compter :

Kiosk : les 5 de l'interface principales et les 2 du popup.

Terminal : la recherche et les trois issues du 'use' chaque type (trois types de ticket de transport)

Bank : une requête de débit

## 5. Développement du protocole

Il faudra développer un protocole qui donne tous les aspects indispensables :

- transport(s)
- port(s)
- formatage(s) de bas niveau des différentes requêtes (trames et la succession des informations que vous prévoyez, encodage des types de base et des types structurés, tableaux, enregistrement, etc.).
- séquence(s) des requêtes : pour 1 requête du client quelles sont les réponses du serveurs et quelles traitements déclenchent-elles.

Ce protocole a une vocation d'être unique quel que soit le langage d'implantation.

## 6. Développement de l'application

A ce stade, il n'est pas demandé de réaliser l'application. Avant les TP, vous recevrez un zip contenant cinq projets (bank, kiosk, server, commun et terminal) qu'il faudra **directement importer** dans un espace de travail et une brève notice explicative qui sera alors le cadre de développement de l'application.