

# Projet *Fast Pizza*<sup>1</sup>

---

## 1. INTRODUCTION

Le projet auquel vous allez participer a pour but de mettre en œuvre vos compétences en génie logiciel et en systèmes de gestion de bases de données relationnelles. Vous devrez concevoir et implanter pour une application permettant de gérer des pizzerias. De plus, vous vous essaieriez à la programmation d'applications utilisant une base de données relationnelles (à travers des transactions). Le développement sera fait en Java en utilisant l'API JDBC. Vous ne devez pas utiliser d'ORM.

Le projet est à faire en quadriômes ou trinômes, il comprend 6 séances de TD, 3 séances de TP et une soutenance en fin de projet. La constitution des équipes et la remise des livrables (rapport final, transparent de la soutenance et code source) se fera sur l'application *Chamilo*. L'évaluation se fera sur les aspects génie logiciel et bases de données du projet en l'état au moment de la soutenance.

## 2. DESCRIPTION DE L'APPLICATION

La société *FastPizza* désire que vous l'aidiez à réaliser une application permettant la gestion de l'ensemble de ses pizzerias.

Chaque pizzeria est identifiée par un numéro unique, a un nom et une adresse postale (numéro de rue, rue, code postal et ville). Une pizzeria est gérée par un responsable qui est un employé de la pizzeria. Un responsable ne gère qu'une seule pizzeria.

Chaque pizzeria propose plusieurs variétés de pizzas (Reine, Marguerite, Calzone...) qui comprennent un ensemble d'ingrédients de base. Les pizzas ont une taille (grande, moyenne ou mini) et un prix. Chaque pizza peut être complétée avec des ingrédients optionnels (identifiés par leur nom). Les mini-pizzas ne peuvent pas avoir plus de deux ingrédients optionnels, les pizzas moyennes pas plus de 4. Pour les grandes pizzas, il n'y a pas de limite pour le nombre d'ingrédients optionnels.

Des livreurs sont affectés à chaque pizzeria. Un livreur est un employé qui travaille dans une seule pizzeria. Il a un badge numéroté (numéro unique pour la pizzeria), un nom, un prénom, une date de naissance, un numéro de téléphone, une adresse email et une adresse postale (numéro de rue, rue, code postal et ville). Il a également une ou plusieurs habilitations pour certains types de permis de conduire (par exemple : permis B pour conduire des voitures, BSR pour des mobylettes).

Un client, identifié par un numéro unique pour une pizzeria, a un nom, un prénom, un numéro de téléphone, une adresse mail et une adresse postale (numéro de rue, rue, code postal et ville). Il passe commande de pizzas à une pizzeria (numéro de commande unique) en précisant pour chaque ligne de sa commande la variété de pizza, la taille, les éventuels ingrédients optionnels, ainsi que le nombre de pizzas commandées. Chaque ingrédient optionnel ajouté coûte 2€. En fonction du contenu de la commande, le prix total de la

---

<sup>1</sup> A noter : ce sujet a été réalisé par l'équipe Base de Données de l'Ensimag (Stéphanie Chollet, Christophe Bobineau et Nagham Alhadad).

commande est calculé et stocké. Lors de la commande, le client doit préciser l'heure à partir de laquelle il souhaite être livré.

A chaque livraison sont associés un livreur, une ou plusieurs commandes et un véhicule. Les véhicules des pizzerias sont identifiés par leur plaque d'immatriculation et décrits par leur type correspondant aux habilitations des livreurs. Pour chaque livraison, il faut préciser l'heure de départ et de retour à la pizzeria.

### 3. TRAVAIL A REALISER

#### 3.1. Modélisation du problème

La modélisation se décompose en deux temps :

- dans un premier temps, vous devez analyser le problème pour en extraire les concepts, les mots-clés, les différents acteurs et les cas d'utilisation. Cette analyse doit donner lieu à la rédaction d'un cahier des charges qui comprendra, entre autres, votre planning, le choix de votre cycle de vie, les exigences fonctionnelles et non-fonctionnelles... Vous noterez que l'énoncé vous indique la problématique à résoudre mais c'est à vous de définir les fonctionnalités que vous allez mettre en place. Vous devrez faire valider votre document par l'enseignant présent en séance de TP. Vous devrez utiliser à bon escient les outils vus en cours de Génie Logiciel pour présenter votre solution : diagramme(s) de cas d'utilisation, diagramme(s) d'états, diagramme(s) de séquence, diagramme(s) de classes, diagrammes d'objets...
- dans un deuxième temps, vous devrez réaliser le schéma de la base de données en respectant les contraintes d'une base de données relationnelles comme vues en cours de Système d'information. Le schéma de la base de données devra être extrait de l'analyse des dépendances fonctionnelles ainsi que de tous les autres types de contraintes. Durant cette phase, vous devrez fournir l'analyse de l'énoncé (dépendances fonctionnelles et contraintes), puis le schéma de la base de données, l'analyse des formes normales, puis le schéma physique obtenu par transformation de modèles. Les choix que vous aurez faits, devront être expliqués et justifiés. Toute cette documentation doit être présentée dans le dossier de conception.

#### 3.2. Implantation de l'application et de la base de données

Vous devrez implémenter votre application en langage Java et utiliser l'API JDBC pour manipuler et interroger votre base de données. Vous n'utiliserez pas d'ORM. Vos développements devront être faits sur le dépôt Git fourni par l'école et utilisable de l'extérieur via le VPN. Les enseignants pourront suivre l'évolution de vos développements dans votre dépôt Git !

Vous devrez implémenter le schéma Entités/Associations en un **schéma relationnel** sur le SGBD Oracle disponible à l'ESISAR. Vous devrez insérer suffisamment de données pertinentes pour la suite du projet.

#### 3.3. Analyse des fonctionnalités

Vous devrez définir les requêtes SQL nécessaires pour réaliser les fonctionnalités suivantes en les regroupant en transactions, ce qui permettra d'assurer la cohérence globale de la base de données, même en cas d'accès concurrents :

- planification d'une nouvelle livraison (affectation d'une commande à un livreur avec un véhicule) ;
- création d'une nouvelle commande de pizza pour un nouveau client et pour un habitué de la pizzeria (attention au calcul des numéros de commandes) ;
- calcul de statistiques :
  - nombre total de pizzas commandées par jour, pizza préférée des clients ;
  - véhicules les plus utilisés ;
  - valeur de la commande avec le montant le plus petit, par pizzeria ;
  - pour une pizzeria donnée et un jour donné : recette du jour (somme du montant de toutes les commandes), avec ventilation de la recette par variété de pizzas ;
  - temps moyen de livraison des commandes ;
  - total des recettes des différentes pizzerias.

Ces requêtes et transactions peuvent (doivent !) être testées sur Oracle (SQL\*Plus) pour en vérifier leur bon fonctionnement, y compris pour des exécutions concurrentes.

### 3.4. Implantation des fonctionnalités

Les fonctionnalités précédemment étudiées devront être implantées en Java/JDBC. Vous réaliserez une interface en mode texte.

## 4. DEROULEMENT DU PROJET

Le projet est constitué de 6 séances de TD/TP, 3 séances de TP de 3 heures et d'une soutenance de 30 minutes.

### 4.1. Séances encadrées

Lors de ces séances de TP, les quadrinômes/trinômes devront avancer sur le projet et pourront poser des questions à leur encadrant. Attention : posez bien vos questions. L'encadrant jouera soit le rôle du client (et, dans ce cas, ne pourra traiter les aspects Génie Logiciel ou Base de Données), soit le rôle d'un expert en génie logiciel ou en bases de données (mais, dans ce cas, il ne connaît rien à l'application). En fin de chaque séance, chaque équipe devra déposer une version actualisée de la documentation du projet dans Chamilo. Ceci afin de pouvoir suivre le déroulement du projet et de pouvoir intervenir au plus tôt en cas de grosses erreurs/anomalies.

### 4.2. Outils

Vous utiliserez l'outil Chamilo pour le bon déroulement du projet ainsi que votre dépôt Git.

### 4.3. Livrables

- **Documentation du projet** : Vous devez réaliser un cahier des charges et un dossier de conception. Dans le dossier de conception, vous devez inclure l'analyse du problème d'un point de vue base de données, la conception Entités/Associations, sa traduction en relationnel en précisant les formes normales des relations, l'analyse des différentes requêtes, leur implantation en SQL et l'utilisation des transactions. D'autres éléments issus du cours de génie logiciel sont aussi attendus, comme la structure de votre application, les explications et justifications de vos choix techniques... Les transparents de la présentation finale font également partie des livrables.

- **Sources Java et SQL :** Vous devrez rendre en fin de projet le code source Java, que vous aurez exporté depuis Eclipse en utilisant l'outil d'export au format **ZIP**. Votre application comprendra un fichier `param.xml` permettant de configurer l'accès à la base de données (voir format en fin de document). Votre application devra permettre la création du schéma SQL sur une base vide, par une entrée de menu dédiée (avec également l'insertion des données de test).

#### 4.4. Soutenance

Le projet se termine par une soutenance. La soutenance se compose de la façon suivante :

- 20 minutes pendant lesquelles vous devrez présenter votre projet (conception, implantation, fonctionnalités), faire une démonstration convaincante de votre prototype et faire un bilan du projet ;
- 10 minutes de questions pour le jury.

Les soutenances sont courtes, vous devez donc bien les préparer (scénario pour la démonstration, répétitions, etc.)

#### 4.5. Format du fichier *param.xml*

```
<FastPizza>
  <Parameter name="database.url" value="jdbc:xxx" />
  <Parameter name="database.user" value="XX" />
  <Parameter name="database.password" value="YY" />
</FastPizza>
```