

Projet Java B3 Alternant Groupe 1

Le projet du cours de Java permet de mettre en œuvre les notions de base vues en cours et d'appréhender la réalisation d'une première application en Java. Il doit être réalisé par groupe de 2 étudiants.

I Calendrier

Le calendrier du projet est le suivant :

- 5 janvier : choix des binômes et tirage au sort des sujets
- **12 janvier** : rendu **pour le début de la séance de cours au plus tard** des livrables suivants
 - Dossier de conception au format Word ou PDF comportant au moins :
 - Un diagramme de classe
 - Un diagramme de cas d'utilisation
 - Un ou plusieurs diagrammes de séquence
- **9 février** : rendu des livrables suivants
 - Code source du projet au format d'une archive ZIP incluant
 - Les sources Java
 - Un script de création et d'initialisation de base de données si nécessaire
 - Un bilan du projet (max 1 page) au format Word ou PDF décrivant
 - Les points de satisfaction
 - Les difficultés rencontrées
 - Les axes de progrès

Tout retard dans le respect des jalons précédents pourra entraîner des pénalités.

Les livrables doivent être rendus par mail à l'adresse suivante : **jerome.dassonville@worldline.com**

II Barème d'évaluation

Le projet est évalué de la manière suivante, la note de projet compte pour 50% de l'évaluation totale du module.

Conception	20
Mise en œuvre du langage Java	40
Couverture fonctionnelle	30
Qualité des livrables	10
Total	100

NB : La couverture fonctionnelle est évaluée en « boîte noire » c'est-à-dire en utilisant uniquement l'interface fournie sans regarder le code.

III Expression du besoin

La description du besoin ci-dessous décrit le besoin tel qu'il aurait pu être recueilli de manière brute auprès du client. Certaines propositions peuvent vous paraître confuse ou peu précise, n'hésitez pas alors à demander des précisions.

L'entreprise **Poils aux Pattes** est une société spécialisée dans le gardiennage d'animaux de compagnie. Elle possède plus de 50 sites de gardiennage en France et a réalisé en 2014 un chiffre d'affaire de plus de 10M€.

Chez *Poils aux pattes*, les réservations pour les animaux sont aujourd'hui faites par téléphone ce qui n'est pas très flexible (on ne peut faire des réservations qu'en heure ouvrée) et ce qui occupe un standard téléphonique à temps plein.

Poils aux pattes vous a donc sollicité afin que vous réalisiez un outil de réservation permettant à ses clients de réserver à toute heure. L'outil doit offrir les fonctionnalités suivantes :

Les clients devront pouvoir lister les sites de gardiennage afin de sélectionner le plus proche des chez eux. Après s'être authentifié, ils devront pouvoir effectuer une réservation pour leur animal de compagnie et pour une période donnée.

Les clients doivent pouvoir au moment de la réservation laisser un commentaire afin d'indiquer des informations supplémentaires au personnel (ex. allergies, trait de caractère...)

La réservation doit notamment inclure le type de l'animal (chien, chat, lapin...), son prénom, sa race, sa catégorie de poids. Le système doit être capable de déterminer si une place est disponible en fonction :

- Du type d'animal
- De la catégorie de poids
- De la période souhaitée
- Du site souhaité

Si une place est disponible le client devra pouvoir valider sa réservation. Si malheureusement aucune place n'est pas disponible sur le site choisi, le système devra en chercher une dans un des sites aux alentours et le proposer à l'utilisateur.

Les employés de Poils aux pattes devront pouvoir consulter la liste des animaux actuellement en cours de gardiennage sur leur site, ainsi que la liste des départs et des arrivés pour la semaine en cours.

Les employés de Poils aux pattes devront pouvoir mettre à jour la liste des cages de leur site. Une cage a un nombre de place limité et ne peut contenir qu'un seul type d'animal. En fonction de sa catégorie de poids un animal peut occuper une ou plusieurs places.

Les animaux actuellement pris en charge par Poils aux pattes sont des chiens, des chats et des lapins. Un accueil futur des furets et des chinchillas est cependant en cours d'étude.

L'application doit être sécurisée à l'aide d'un système d'authentification basique (nom d'utilisateur et mot de passe).

L'interface envisagée par le client est une interface web, cependant les maquettes ayant pris du retard vous devez réaliser une première version de l'application avec une interface Console ce qui permettra de tester et démontrer les fonctionnalités métiers.

IV Objectif

Votre objectif est de réaliser une application en Java répondant aux besoins exprimés ci-dessus. Notez bien que l'interface doit être suffisamment fonctionnelle pour démontrer les fonctionnalités de l'application.

IV Exercices

Les exercices suivants permettent de mettre directement en application les notions vues en cours dans le cadre du projet. **A chaque étape réussie il est conseillé de versionner sa copie de travail à l'aide d'un outil tel que GIT ou SVN.**

Exercice 1 : Premiers pas (séance 2)

1. Prenez quelques minutes pour modéliser de manière simple (sans héritage) le module d'authentification de l'application à l'aide d'un ou plusieurs diagrammes UML
2. Dans Eclipse (ou l'IDE de votre choix) créez un nouveau projet de type « Java Project »
3. Implémentez la ou les classes correspondant à votre modélisation en tenant compte des contraintes suivantes :
 - a. La méthode main ne doit pas contenir plus de 10 lignes
 - b. Chaque classe doit posséder un constructeur par défaut et un constructeur recevant l'ensemble des attributs en paramètres
 - c. Aucun attribut ne doit être public
 - d. La méthode toString doit être redéfinie
 - e. La méthode equals et hashCode doit être redéfinie
 - f. Vous pouvez utiliser les possibilités de génération de code de votre IDE
4. Implémentez et testez une interface console permettant de s'authentifier
 - a. Utilisez une liste de l'API Collection pour gérer la liste des utilisateurs
5. Gérez les erreurs suivantes avec des classes d'exception distinctes :
 - a. Compte utilisateurs inexistant
 - b. Mauvais couple nom d'utilisateur / mot de passe

NB : Les exceptions métier doivent être placées dans un package dédié et doivent être gérées par l'application (elles ne doivent pas provoquer l'arrêt du programme)

Exercice 2 : Réalisation du modèle (séance 2)

1. Réalisez le diagramme de classe pour l'ensemble de l'application avec les contraintes suivantes :
 - a. Il doit comporter au moins une relation d'héritage
2. Listez les cas d'erreur « métiers » pouvant être rencontrés lors de l'utilisation de l'application, chaque cas d'erreur devra correspondre à une classe d'exception.
3. Implémentez l'ensemble des classes du modèle ainsi que les classes d'exception

Exercice 3 : Persistance sur disque (séance 3)

1. Définissez une interface permettant de sauvegarder et de charger des utilisateurs depuis une source de données
2. Implémentez cette interface à l'aide d'objets sérialisés et de fichier plats

Exercice 4 : Persistance en base de donnée (séance 3)

1. Si ce n'est pas déjà le cas, installez MySQL sur votre poste
2. A l'aide d'un outil graphique (phpMyAdmin, MySQL Workbench) ou en ligne de commande créez une base de données correspondant à votre projet.
3. Dans votre projet, importez le Driver JDBC se trouvant également dans le dépôt
 - a. Clic droit sur le projet puis *Build Path > Add external archive*
4. Testez le bon fonctionnement du Driver
 - a. Chargement du driver
 - b. Récupération des informations d'une table
5. Proposez une nouvelle implémentation pour l'interface IUserDataAccess s'appuyant sur la base de données plutôt que sur un fichier plat

Exercice 5 : Réalisation des modules fonctionnels (séance 4 et 5)

1. Implémentez et testez la persistance de votre choix (base de données ou disque) pour l'ensemble des classes du modèle
2. Implémentez et testez chacun des cas d'utilisations