

ACDA3.1 FI - Interface Homme Machine

Micro-projet : Hotel numérique

Année universitaire 2017 / 2018

L'objectif de ce micro-projet est de vous faire mettre en application ce qui a été vu dans le module d'ACDA 3.1. Vous allez donc devoir créer une **IHM** (Interface Homme Machine) basée sur l'approche **MVC** (Modèle Vue Contrôleur, *Model-View-Controller*).

1 Description de l'application

Ce projet consiste à créer un système simplifié de gestion des chambres d'un hôtel. On considère qu'un hôtel dispose de 100 chambres :

1. 50 chambre avec un lit double (index 1 à 50) ;
2. 25 chambre avec un lit simple (index 51 à 75) ;
3. 25 chambre avec deux lits simples (index 76 à 100).

Un système de réservation externe gère la prise de réservation des clients. Dans le cadre de ce projet, vous n'aurez pas à programmer ce système de réservation ! Vous devez juste considérer qu'il vous fournit une liste de réservations, accessible à partir d'une base de données externe au système de l'hôtel. On suppose de plus que les réservations ne peuvent pas créer des situations de *surbooking*¹.

Vous devez programmer les deux interfaces décrites dans la suite. Chaque interface étant destinée à des utilisateurs différents, vous devrez produire 2 programmes.

Allocation d'une chambre à un client. Les employés à la réception de l'hôtel doivent pouvoir attribuer une chambre à un client lorsqu'il se présente à l'hôtel. Dans le cadre de ce projet, on suppose que tout client a préalablement réservé une chambre avant de venir à l'hôtel. Lorsqu'un client se présente, le système doit tout d'abord permettre à l'employé de retrouver le client et sa réservation parmi la liste des réservations fournies par le système de réservation externe. Cette recherche peut être effectuée en se basant sur le numéro de réservation ou sur les nom et prénom du client. Ensuite, le système de gestion des chambres indique à l'employé le numéro d'une chambre disponible, qui correspond aux critères de réservation. Soit l'employé valide cette proposition et la chambre est allouée au client, soit il peut consulter la liste de toutes les chambres qui respectent ces critères, et l'employé peut dans ce cas échanger la chambre proposée automatiquement par le système avec une autre chambre. Vous devez aussi prévoir des notifications d'erreur en cas de problème avec les données client ou si la réservation n'existe pas. Pour des questions pratiques, nous vous conseillons de choisir une date fixe pour la date du jour, ce sera plus facile pour réaliser des tests.

1. pratique commerciale qui consiste à vendre en réservation un nombre de places supérieur à la quantité réellement disponible

Statistiques Le gérant de l'hôtel souhaite pouvoir consulter des statistiques concernant les nuitées déjà réalisées. Cette fonctionnalité sera accessible dans son bureau.

Il souhaite pouvoir :

- afficher le taux d'occupation de l'hôtel pour une date donnée (ex : le 20/10/2017) ;
- afficher le taux de non-présentation pour une date donnée (ex : le 24/12/2042) ;
- afficher le taux d'occupation de l'hôtel pour une période donnée (ex du 16/12/2016 au 2/1/2017) ;

2 Organisation

2.1 Informations techniques

Le langage à utiliser pour ce projet est obligatoirement **Java**.

La base de données externe fonctionne avec **MariaDB**. Elle est composée de plusieurs tables, dont *Client* et *Reservation*. Un exemple d'une telle base se trouve sur le serveur `dwarves.iut-fbleau.fr` et s'appelle `projetihm`. Vous pouvez la visualiser en vous rendant à l'adresse `http://dwarves.iut-fbleau.fr/phpmyadmin/`. Les identifiants pour se connecter sont :

- utilisateur : `projetihm`
- mot de passe : `whitejorp`

La connexion avec la base de donnée doit être effectuée en Java avec l'API **JDBC**.

Les IHM doivent respecter les critères **d'ergonomie**.

2.2 Rendu

Vous devez constituer des **groupes de 2** pour réaliser le projet.

Il est vivement conseillé de travailler sur l'analyse (cas d'utilisation, diagramme de classes, diagramme de séquences) avant de commencer à coder.

Deux livrables sont à réaliser :

1. Le **code source Java** organisé suivant le modèle MVC. Un **Makefile** doit permettre de compiler le programme, d'exécuter la suite de test et de nettoyer le projet.
2. Le **rapport** de micro-projet au format **PDF**. Il doit **au moins** contenir les diagrammes d'analyse et de conception, la description du produit final et des captures d'écran.

Une **archive** contenant le code source ainsi que le rapport final est à rendre le **vendredi 20 octobre 2017 à 23 heures au plus tard**. Tout retard sera pénalisé. Un lien de soumission sera créé sur le site du département.

Pour déposer votre travail, vous devez vous rendre sur la **page du département**, dans la partie **Ressources informatiques**, il faut cliquer sur **Rendre un devoir**.