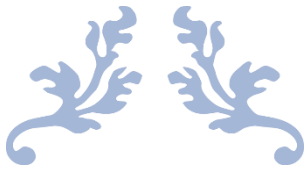

PROJET IHM

Florian CUNSOLO & Antoine MARCHAL DOMBRAT

28 NOVEMBRE 2021

Groupe 2

Table des matières

Lien du Projet : **https://dwarves.iut-fbleau.fr/gitiut/cunsolo/FIprojetIHM2021**
Répartition : Prototype1 (Florian) et Prototype2 (Antoine)

Présentation technique

Premier Prototype

Fonctionnement

Dans un premier temps, il y a le « *main* » qui lance la fenêtre d'accueil de notre application. Cette fenêtre ajoute l'image de fond pour toutes les autres pages. Avec une requête SQL, nous récupérons les tuples dans la base de données issues du système de réservation externe pour créer les clients avec l'API et ajoutons les pré-réservations afin de les récupérer dans un tableau. Ce tableau sera transmis au model suivant pour comparer ce que tape l'utilisateur avec ces pré-réservations. Si le nom et prénom du client ou la référence de sa pré-réservation est dans le tableau, la fenêtre affiche ses informations. Nous récupérons aussi les chambres occupées dans l'hôtel.

Dans un second temps, l'employé choisi la pré-réservation exacte du client en fonction de ce qui est affiché (une ou plusieurs), ce qui nous amène sur page récapitulative des informations du client avec un numéro de chambre attribué en fonction de sa catégorie. L'employé peut choisir de soit valider la chambre, soit choisir une autre chambre de la même catégorie ou soit de revenir en arrière pour chercher une autre pré-réservation.

Quand l'employé valide la chambre, le numéro de chambre est inscrit dans une base de données personnelle et quand il reviendra sur la réservation il ne pourra plus modifier le numéro de chambre et verra afficher ce numéro.

Diagramme de classe

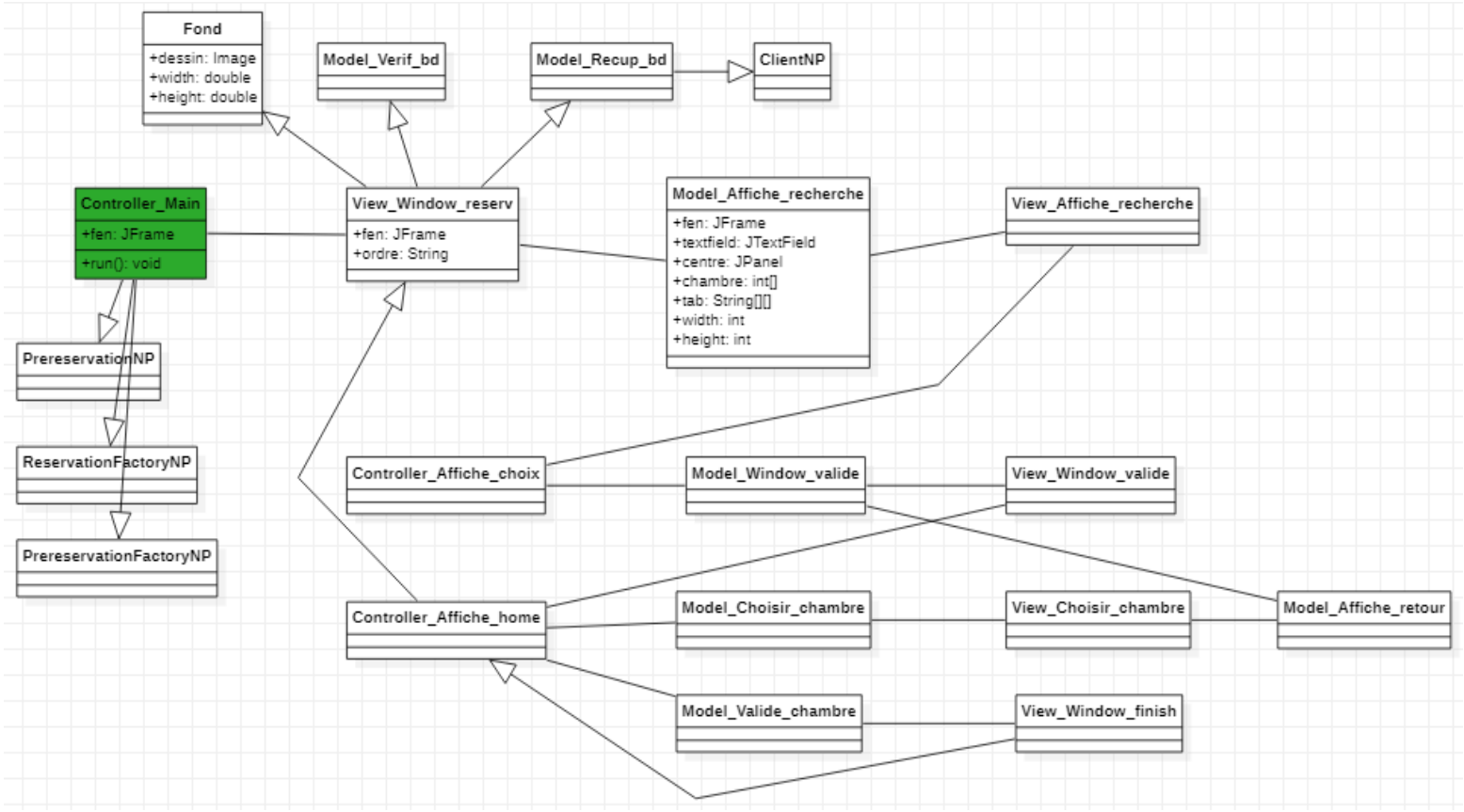

Diagramme de classe du second prototype

Second Prototype

Sélection de la date

Pour que l'utilisateur puisse sélectionner une date, un JSpinner est affiché à la fenêtre. Il s'agit d'un composant inclus dans le package javax.swing qui permet de sélectionner une date. Par défaut la date sélectionnée est le 09/11/2018. L'utilisateur peut changer la date sélectionnée par la date du jour précédant en cliquant sur la flèche supérieure ou par la date du jour suivant en cliquant sur la flèche inférieure. Il a également la possibilité de la modifier en cliquant sur le JSpinner puis en tapant au clavier.

Sélection de la durée

L'utilisateur peut sélectionner une durée. Il a le choix parmi 4 possibilités : 1 jour, 1 semaine, 1 mois, 3 mois. La sélection de la durée se fait par l'intermédiaire de boutons radios. C'est un composant inclus dans le package javax.swing. Les 4 boutons appartiennent au même ButtonGroup. Ainsi il n'est possible de sélectionner qu'une durée à la fois.

Afficher le diagramme

Après avoir sélectionné la date et choisi la durée, l'utilisateur peut afficher le diagramme en appuyant sur le bouton. Il s'agit d'un JButton. C'est un composant inclus dans le package javax.swing. Lorsqu'il est pressé, la base de données va être appelée à l'aide de la classe Jour. Pour chaque jour à partir de la date sélectionnée et pendant la durée sélectionnée, le taux d'occupation va être calculé. De plus, pour permettre à l'utilisateur de comparer ces données, la moyenne d'occupation des 3 ans années précédant la date choisie va être calculée. C'est un processus très gourmand qui demande alors plusieurs secondes d'attente.

Diagramme de classe

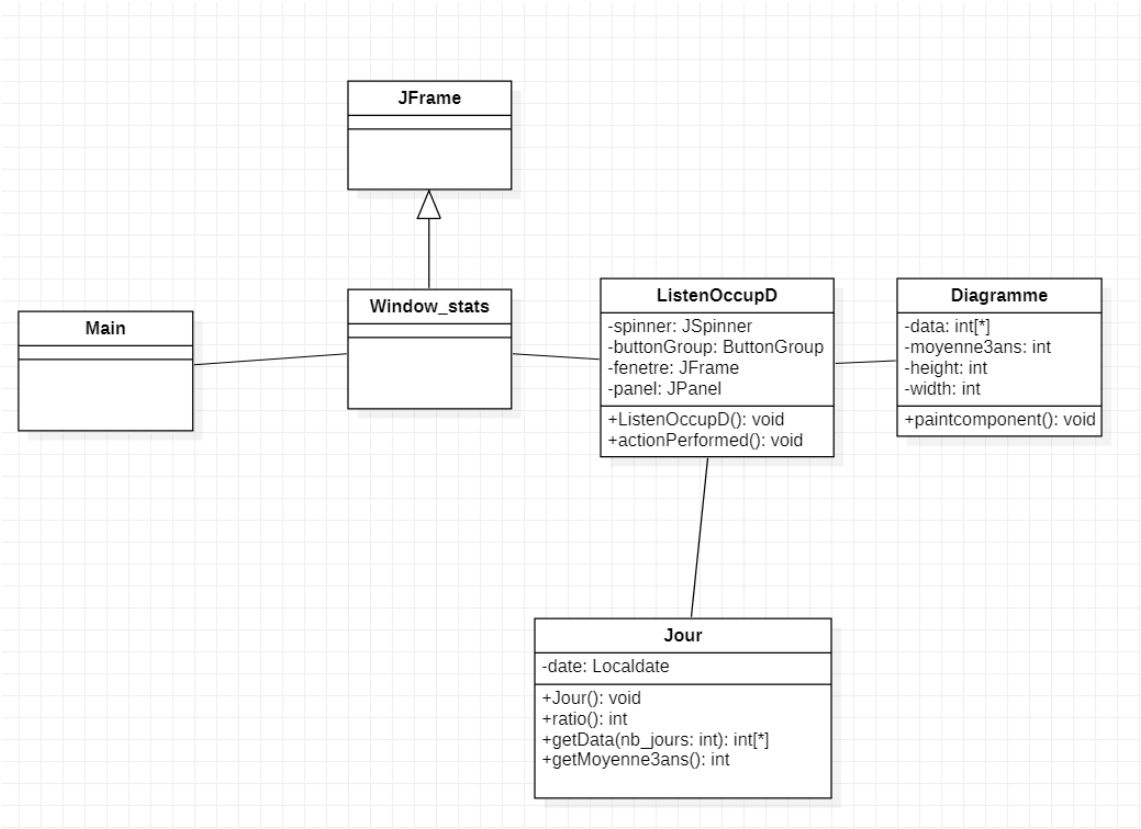

Diagramme de classe du second prototype

Présentation des tests

Premier Prototype

Test des pré-réservations

Comme dis plus haut, les clients et les pré-réservations créés avec l'API sont insérés dans un tableau qui est de type « String » or les données récupérées avec l'API ne sont pas tous de ce type. Il a donc fallu effectuer un transtypage de certaines valeurs et vérifier que les conversions avaient bien été effectué pour permettre une bonne lecture du tableau.

Test des numéros de chambre

La fenêtre de validation du numéro de chambre d'une réservation affiche un numéro de chambre aléatoire qui n'est pas déjà occupé. Si l'employé veut changer de chambre, la fenêtre lui affiche toutes les chambres non occupées moins la chambre qui avait été choisi à l'affichage de la fenêtre puisque l'employé peut appuyer sur le bouton « retour » et retrouver le numéro de base qui était affiché. Des tests ont été effectué pour vérifier que les bons numéros de chambre étaient affichés.

Test graphiques

Notre application est visuellement épurée et minimaliste, elle a été soumise a beaucoup de tests graphiques puisque nous utilisons plusieurs mises en page différentes pour les fenêtres en fonction des informations que nous voulions affichées. J'ai rencontré un gros problème graphique avec le JScrollPane qui ne voulait pas s'afficher puis dans un second ne voulait pas devenir opaque, mais qui a été réglé avant la fin du projet.

Second Prototype

Test des boutons radio

Il a été vérifié que les boutons radio appartiennent bien au même groupe et qu'il ne soit pas possible d'en sélectionner plusieurs à la fois.

Test du choix de la date

Plusieurs tests ont été effectués à propos du choix de la date.
Lorsque l'utilisateur tente de rentrer une valeur incohérente, le JSpinner gère cette entrée pour la transformer en une date conforme. Par exemple si "13" est rentré pour le mois, et x pour le jour et y pour l'année, alors le JSpinner va transformer cette date en x/01/y+1. Cela permet alors de filtrer d'éventuelles erreurs. De même, lorsque le nombre de jour est supérieur à 28, 30 ou 31 selon le mois en question.

Conclusion Florian

Ce projet en binôme m'a permis d'appliquer les connaissances d'analyses, conceptions et développements des applications que j'ai pu apprendre et renforcer les connaissances vues auparavant.

J'ai pris beaucoup de plaisir à coder cette application puisque la java est un langage que j'aime beaucoup. J'ai aussi apprécié mettre en place un model MVC ce qui permet d'avoir une meilleur répartition et compréhension de notre code.

Conclusion Antoine

Pour conclure, ce projet m'a permis d'approfondir et de consolider mes connaissances en JAVA notamment en mettant en application les connaissances acquises lors du module ACDA 3.1. J'ai aussi pu mettre en place des connections et des requêtes à une base de données ce qui m'a permis de m'améliorer en SQL.

Conclusion

Pour conclure, nous avons réussi à fournir un code fonctionnel avec la meilleur qualité possible même si cela n'a pas été toujours facile et nous sommes très fiers de ce projet.