

Phase 1 - Ressources

Introduction

Bienvenue dans la documentation du projet *Ressources* du cours d'architecture logicielle. Notre équipe est constituée de quatre membres : Alexis Fiolet, Nathanaël Depauw, Louis Tonnevy et Adrien Taberner. L'objectif de cette documentation est de vous expliquer ce qui a été accompli au cours de la première phase du projet.

Procédure d'installation à partir du Code Source :

Clone du répertoire GIT :

```
$ git clone https://github.com/Akunesquik/TPArchi.git
```

Installation avec Maven :

```
$ cd TPArchi/  
$ mvn compile assembly:single
```

Procédure de désinstallation en ligne de commande :

Avec Maven

```
$ mvn clean
```

La création du programme est possible depuis un IDE comme Eclipse avec :

1. Run As -> Maven install
2. Run As -> Java Application

Environnement logiciel nécessaire

Pour l'installation du système, vous devez vous assurer d'avoir l'environnement logiciel nécessaire pour le projet.

1. Java Development Kit (JDK) 11 :

Assurez-vous d'avoir installé le JDK 11 sur votre système. Vous pouvez télécharger le JDK 11 depuis le site officiel d'[Oracle](https://www.oracle.com/technetwork/java/javase-downloads-2132671.html).

2. Apache Maven :

Installez Apache Maven sur votre système. Vous pouvez télécharger la dernière version de Maven depuis le site officiel de Maven. Les IDE comme Eclipse et IntelliJ IDEA intègrent déjà cet outil.

Description de l'interface de l'application

L'application se structure autour d'une fenêtre principale unique, où à droite un tableau affiche l'ensemble des réservations effectuées. Au-dessus de ce tableau, trois listes sont présentes, représentant respectivement les horaires, les personnes et les salles. Sous chaque liste, un bouton permet de supprimer après sélection un élément d'une liste. En dessous du tableau des réservations, une fonctionnalité permet la suppression d'une ou plusieurs réservations, selon le nombre sélectionné. Enfin, une autre option située sous le tableau donne la possibilité de modifier l'horaire d'une réservation. À gauche, on retrouve un menu déroulant permettant de créer, au choix, une salle, un horaire ou une personne. En dessous on a une option qui, après sélection d'un horaire, d'une personne et d'une salle, permet de créer une réservation.

Gestion des Ressources

Créer : Horaire ▼ Créer

Réserver

Horaire Personne Salle

Supprimer Horaire Supprimer Personne Supprimer Salle

Salle	Personne	Horaire
-------	----------	---------

Modifier Horaire Supprimer Réservation

Liste des exigences fonctionnelles, satisfaites :

- Req-res-01: L'utilisateur peut créer une salle.
- Req-res-02: L'utilisateur peut supprimer une salle.
- Req-res-03: L'utilisateur peut créer une personne.
- Req-res-04: L'utilisateur peut supprimer une personne.
- Req-res-05: L'utilisateur peut créer un créneau (temporel).
- Req-res-06: L'utilisateur peut supprimer un créneau (temporel).
- Req-res-07: L'utilisateur peut créer une réservation (personne - salle - créneau).
- Req-res-08: L'utilisateur peut supprimer une réservation (personne - salle - créneau).
- Req-res-09: L'utilisateur peut changer une réservation de créneau temporel.

Liste des exigences non fonctionnelles, satisfaites :

- Req-exp-01: Le système doit fonctionner sous Linux.
- Req-exp-02: Le système doit fonctionner sous Windows.
- Req-exp-04: Le système doit être développé en Java 11.
- Req-doc-05: La documentation doit décrire l'environnement logiciel nécessaire l'installation du système.
- Req-doc-10: La documentation doit décrire la procédure de compilation, depuis la récupération du code source jusqu'à la production du système packagé.
- Req-arc-07: Les requêtes vont uniquement du composant IHM vers le composant exposant les fonctions.
- Req-fct-01: Après re-démarrage du système il est dans le même état qu'avant son arrêt (données).

Développement

Vue physique

Le point d'entrée de l'application est le fichier Main.java, cette classe est responsable de l'instanciation des objets Model, View et Controller. Le code source s'engage à suivre les principes du modèle MVC.

IHM

Contient tous les objets swing nécessaire à l'interface graphique. Responsables des interfaces et du bon affichage des données.

Bibliothèque graphique : Java Swing

Fichiers : View.java

Modèle

Logiques métier de l'application, Définie les données et les structure sous-jacentes. Ces classes sont responsables de l'accès aux données, de leur traitement, et de leur gestion.

Fichiers : Personne.java, Salle.java, Horaire.java, Reservation.java, Model.java

Contrôleur

Responsables de la gestion des interactions entre la vue et le modèle et intervient également dans la mise à jour de la vue en fonction des changements dans le modèle.

Fichiers : Controller.java

Données

Classes nécessaires à la persistance des données utilisée par le contrôleur et le modèle.

API : JDBC + SQLite, le driver pour JDBC SQLite est chargé à l'exécution.

Fichiers : Database.java

Vue logique

Diagramme de classes

Dans cette représentation les éléments en jaune représentent des dépendances extérieures ou des héritages de modules externes.

