

ARLO Ressources Phase 1 - Documentation

Côme Annocque, Elisa Conin, Robin Gaillard

2019-2020



Sommaire

1	Installation	3
1.1	Environnement matériel et logiciel nécessaire	3
1.2	Installation supplémentaire pour Linux	3
1.3	Importation et compilation sous Eclipse	3
1.4	Désinstallation	4
2	Exigences satisfaites	4
2.1	Non Fonctionnelles	4
2.1.1	Export	4
2.1.2	Livrable	4
2.1.3	Architecture	4
2.2	Fonctionnelles	4
3	Exigences non satisfaites	4
4	Anomalies	5
5	Fonctionnement de l'application	5
5.1	Architecture	5
5.2	Interface et cheminement des appels de fonctions	6
5.2.1	Écran Salle	6
5.2.2	Écran Personne	7
5.2.3	Écran Créneau	8
5.2.4	Écran Réservation	9
5.3	Description des entrées/sorties des composants	9



1 Installation

1.1 Environnement matériel et logiciel nécessaire

Pour pouvoir installer et utiliser l'application, vous aurez besoin d'un ordinateur sous système d'exploitation Windows ou MAC OS ou Linux.

Pour pouvoir lancer l'application vous aurez besoin d'un JRE (Java Runtime Environment), permettant l'utilisation d'application Java.

Afin de pouvoir utiliser les éléments permettant la génération de l'application, vous aurez besoin d'un JDK 8 (Java Development Kit), l'application étant réalisée avec Java 8.

Eclipse est conseillé comme IDE.

1.2 Installation supplémentaire pour Linux

Pour exécuter notre logiciel depuis un environnement Linux, il faut utiliser la commande :

```
$ java -jar ARLO_Ressources.jar
```

Si cette commande ne fonctionne pas, il vous est conseillé d'installer le paquet nécessaire grâce à la commande :

```
$ sudo apt install openjdk-8-jre-headless
```

Si une erreur subsiste il faut modifier le fichier `/etc/java-8-openjdk/accessibility.properties` grâce à la commande :

```
$ sudo vim /etc/java-8-openjdk/accessibility.properties
```

et mettre la ligne `assistive_technologies=org.GNOME.Accessibility.AtkWrapper` en commentaire avec un `#` devant.

Si cette commande ne fonctionne pas, veuillez installer vim grâce à la commande :

```
$ sudo apt install vim
```

1.3 Importation et compilation sous Eclipse

Pour télécharger le projet, rendez-vous sur le lien suivant :

<https://github.com/WaRobby78/Ressources/tree/CôteBranch>

La procédure pour installer le projet sous Eclipse est la suivante.

Aller dans File->Import->Existing Projects Into Workspace->Select Root Directory->Browse...

Indiquer le chemin où vous avez enregistré le projet téléchargé.

Cliquer sur Finish et vous aurez importé le projet dans votre environnement Eclipse avec succès.

Vous pourrez ensuite l'exécuter avec le raccourci Ctrl+F11 ou en cliquant sur Run.

Pour créer un exécutable .jar de l'application sous Eclipse, faites un clic droit sur votre projet -> Export -> Java -> Runnable JAR File -> Next -> Finish



1.4 Désinstallation

Supprimer le projet.

2 Exigences satisfaites

2.1 Non Fonctionnelles

2.1.1 Export

Le système doit fonctionner sous Linux.

Le système doit fonctionner sous Windows.

Le système doit fonctionner sous Mac OS.

Le système doit être développé en Java 8

2.1.2 Livrable

La livraison doit contenir tous les éléments nécessaires à la génération de la version binaire.

La livraison doit contenir la version binaire du système.

La livraison doit contenir toute la documentation.

2.1.3 Architecture

Le système est constitué d'un seul exécutable.

Le programme principal instancie deux objets : l'un implémentant l'IHM et l'autre exposant les fonctionnalités.

Les communications entre IHM et fonctions passent par une unique interface Java.

Les requêtes vont uniquement de l'IHM vers l'objet exposant les fonctions.

Les opérations exposées par l'interface sont de 4 types : lecture, création, modification, suppression d'un objet.

Les classes implémentant l'IHM et les fonctionnalités sont packagées dans des jar distincts.

L'identifiant des objets créés est attribuée par l'IHM.

2.2 Fonctionnelles

Après redémarrage du système il est dans le même état qu'avant son arrêt (données).

L'utilisateur peut créer une salle.

L'utilisateur peut supprimer une salle.

L'utilisateur peut créer une personne.

L'utilisateur peut supprimer une personne.

L'utilisateur peut créer un créneau (temporel).

L'utilisateur peut supprimer un créneau (temporel).

L'utilisateur peut créer une réservation (personne – salle - créneau).

L'utilisateur peut supprimer une réservation (personne – salle – créneau).

L'utilisateur peut changer une réservation de créneau temporel.

3 Exigences non satisfaites

L'utilisateur peut créer des réservations sur un ensemble de créneaux.



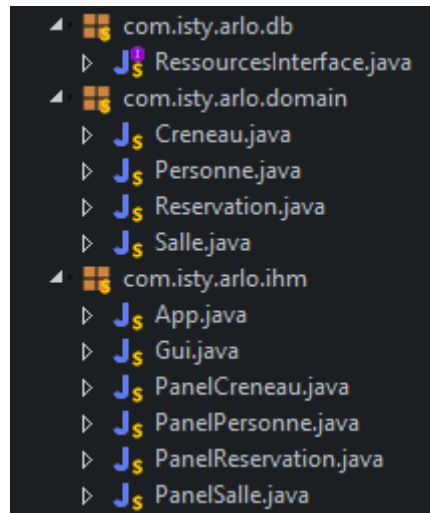
4 Anomalies

On peut citer comme anomalie que on peut supprimer une salle, une personne ou un créneau qui a une réservation qui lui est lié.

5 Fonctionnement de l'application

5.1 Architecture

L'application a été réalisée avec Java 8 et la bibliothèque graphique Swing pour l'interface.



Les fichiers .java de l'application sont structurés en fonction de leur rôle.

Le package **domain** contient les classes entités liées à la base de données de mon application. Elles permettent de définir ce qu'est une salle, une personne, un créneau et une réservation.

Une salle est définie par son nom qui est une chaîne de caractères.

Une personne est définie par son nom, prénom et job qui sont des chaînes de caractères.

Un créneau est défini par une date de début et une date de fin.

Une réservation est définie par une salle, une personne et un créneau.

Le package **ihm** concerne l'implémentation de l'interface graphique.

Elle contient App.java qui est le programme principal qui permet de lancer l'application.

Elle contient Gui.java qui concerne l'implémentation de la fenêtre contenant les sous-composants Panel.

Les Panel correspondent à la gestion de chacune des entités.

L'interface graphique fait des appels à l'interface java contenu dans le package **db**.

L'interface java contenue dans ce package contient les fonctionnalités permettant les opérations CRUD sur les entités et les appels à la base de données.

C'est donc là qu'on y trouve les opérations de lecture, de création, de modification et de suppression d'objet.

5.2 Interface et cheminement des appels de fonctions

L'interface de l'application est composée de 4 écrans, chaque écran gérant son entité/objet à savoir les salles, les personnes, les créneaux temporels et les réservations.

Il y a aussi un bouton disponible pour formater la base de données et recommencer avec une nouvelle.

Chaque écran est implémenté dans une classe java : Panel<nom de l'entité>.java.

Ils sont chacun divisé en 2 parties : une partie représentant la table des objets qu'ils doivent gérer et une partie pour la gestion de ces objets.

5.2.1 Écran Salle

id	nom
1	S11
2	S12
3	103

Nom

Ajouter la salle

Supprimer la salle

Reset database

Le Panel salle est donc composé d'un tableau représentant toutes les salles stockées dans la base de données.

A l'initialisation, l'IHM fait un appel en lecture à l'interface java pour récupérer toutes les salles et les afficher dans le tableau.

La partie gestion est composée d'un TextField permettant d'entrer le nom de la salle à ajouter. Le nom de la salle ne peut pas être null et il est unique, il ne peut pas y avoir de doublons.

Il y a un bouton d'ajout qui va faire un appel à l'interface java pour la création d'une salle et l'ajouter à la base de données.

Lorsqu'une salle est sélectionnée dans la table des salles, le bouton Supprimer la salle devient actif. Il est alors possible de supprimer la salle sélectionnée. Au clic du bouton, l'IHM va faire appel à la fonctionnalité de suppression de l'interface Java pour supprimer la salle de la base de données.

5.2.2 Écran Personne

id	nom	prenom	job
1	Annocque	Côme	Etudiant
2	Conin	Elisa	Etudiant
3	Gaillard	Robin	Etudiant
4	M	Petit	Professeur

Nom

Prénom

Job

Ajouter la personne

Supprimer la personne

Reset database

Le Panel personne permet de gérer les personnes stockées dans la base de données.

A l'initialisation, l'IHM fait un appel en lecture à l'interface java pour récupérer toutes les personnes et les afficher dans le tableau.

Pour la partie gestion, il y a 3 TextField permettant d'entrer le nom, le prénom et le job de la personne à ajouter. Le nom et le prénom d'une personne ne peut pas être null.

Idem que précédemment, il y a un bouton d'ajout et de suppression qui font tous les 2 appels respectivement à la fonctionnalité de création et de suppression.

5.2.3 Écran Créneau

id	date début	date fin
1	04/10/2019 09:00	04/10/2019 12:00

DEBUT

2019
Octobre
4
9h

FIN

2019
Octobre
4
12h

Ajouter le créneau
Supprimer le créneau

Reset database

Le Panel créneau permet de gérer les créneaux stockés dans la base de données.

A l'initialisation, l'IHM fait un appel en lecture à l'interface java pour récupérer tous les créneaux et les afficher dans le tableau.

Afin de définir le créneau à créer, on a 8 ComboBox, 4 pour la date de début de créneau et 4 pour la date de fin. Elles permettent de définir l'année, le mois, le jour et l'heure d'une date.

La date de fin ne peut être inférieure à la date de début.

Idem que précédemment, il y a un bouton d'ajout et de suppression qui font tous les 2 appels respectivement à la fonctionnalité de création et de suppression.

5.2.4 Écran Réservation

id	personne	salle	créneau début	créneau fin
1	M Petit	103	04/10/2019 0...	04/10/2019 1...

Personne: [1] Annocque Côte

Salle: [1] S11

Créneau: [1] 04/10/2019 09:00 -> 04/10/2019 12:00

Ajouter la réservation

[1] 04/10/2019 09:00 -> 04/10/2019 12:00

Modifier le créneau

Supprimer la réservation

Reset database

Le Panel réservation permet de gérer les réservations stockées dans la base de données.

A l'initialisation, l'IHM fait un appel en lecture à l'interface java pour récupérer toutes les réservations et les afficher dans le tableau.

La partie gestion, est composé de 3 ComboBox qui font respectivement un appel en lecture à l'interface java afin de récupérer toutes les personnes, salles et créneaux stockés dans la base de données.

Le bouton d'ajout de réservation va faire un appel en création à l'interface java avec les paramètres définis dans les ComboBox décrits précédemment.

A la sélection d'une réservation dans la table, le ComboBox pour modifier le créneau de la réservation s'active ainsi que le bouton de modification et de suppression.

Au clic sur le bouton de suppression, un appel de suppression d'entité est envoyé à l'interface java qui va s'occuper de supprimer la réservation sélectionnée de la base de données.

Au clic sur le bouton de modification de créneau, un appel de type update va être envoyé à l'interface java, et va remplacer le créneau de la réservation par le créneau sélectionné dans la ComboBox prévu à cet effet.

5.3 Description des entrées/sorties des composants

