

DOCUMENTATION PROJET

ARCHITECTURE LOGICIELLE

Réalisé par:

AZZAHRAOUI Najah
CHERIF Jihene
HIBBI Kenza
HOFFER Max

Encadré par:

M. Marc PETIT

Année 2021-2022

Environnement matériel et logiciel :	3
Procédure d'installation:	3
Changement de path de la base de données.	12
Diagramme de classes	12
Exigences	13
Anomalies	14
Problème sous certaines distributions linux avec javaFx. En effet la compilation fonctionne sur Ubuntu mais nous n'avons pas réussi à le faire tourner sous Arch. Nous n'avons pas réussi à identifier le problème.	14
Procédure de compilation	15
Annexe	19
Script SQL	19

Environnement matériel et logiciel :

- Pour pouvoir installer et utiliser l'application, aucun matériel spécifique n'est nécessaire, il suffit de se munir d'un ordinateur sous système d'exploitation Windows ou Linux.
- Afin de pouvoir utiliser les éléments permettant la génération de l'application, vous aurez besoin d'un JDK 11.0.8 (Java Development Kit), l'application étant réalisée avec Java 11 ainsi que JavaFx
- Eclipse est conseillé comme IDE.

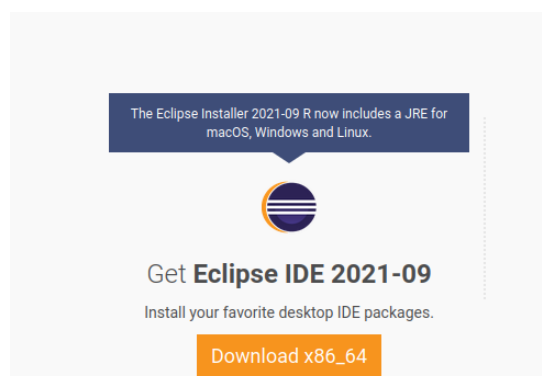
Procédure d'installation:

- INSTALLATION ECLIPSE:

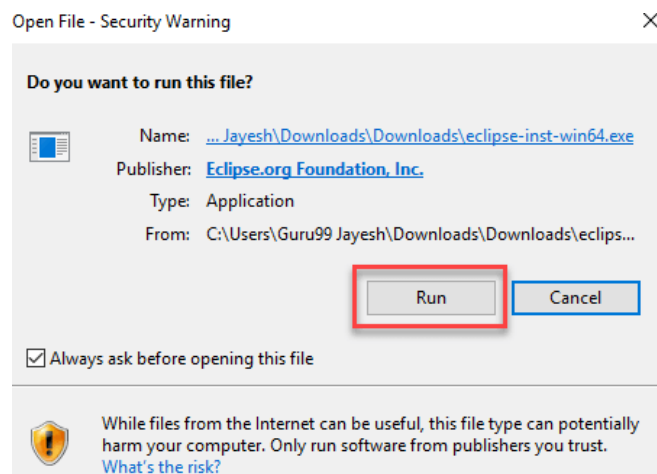
Dans cette partie du projet on a installé eclipse, pour cela on a suivie les étapes suivantes:

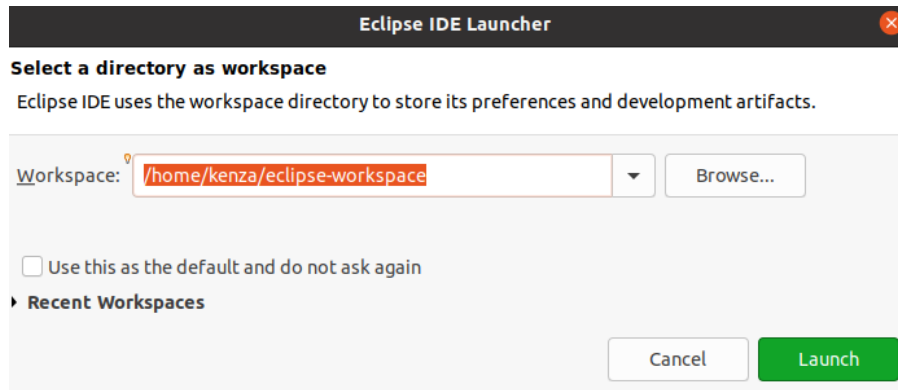
- Télécharger Eclipse :

Afin de télécharger eclipse , on a récupéré l'application sur le site <https://www.eclipse.org/>



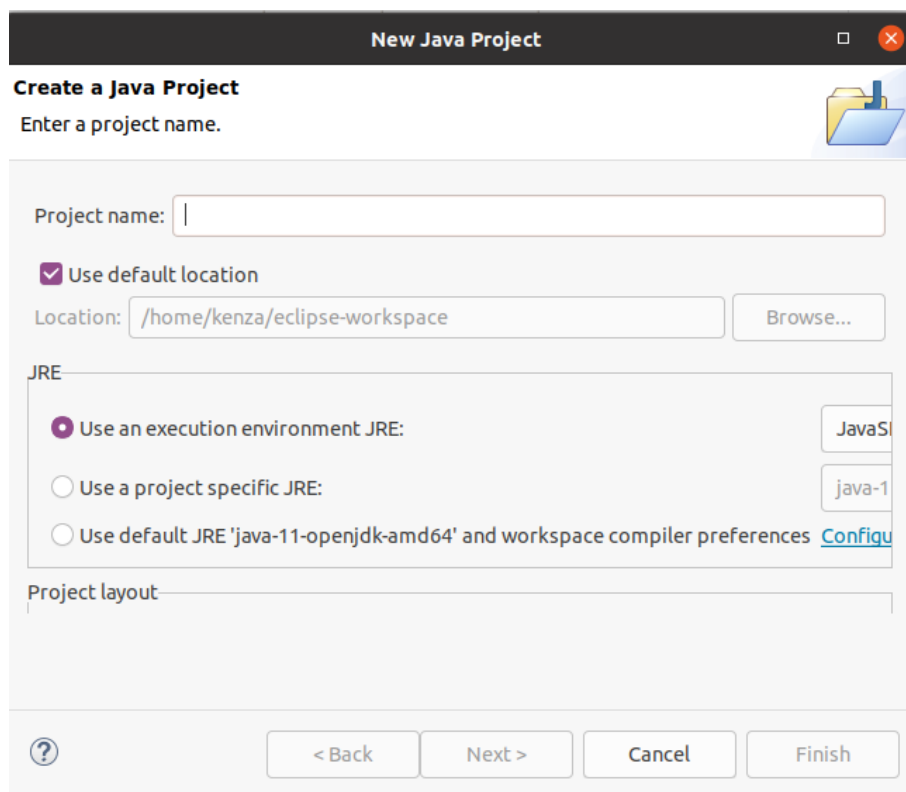
Ensuite on l' a installé comme le montre la figure suivante:

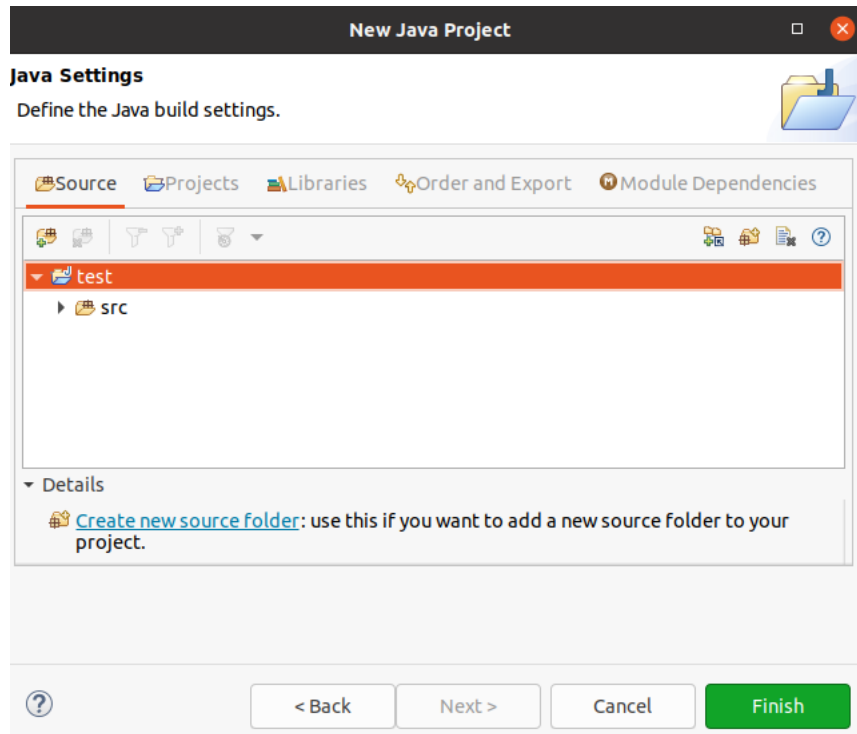




- Création d'un nouveau projet:

Après avoir installé Eclipse, pour ouvrir un nouveau projet on a cliqué sur **File > New > Java Project**, ensuite on a choisi le nom du projet comme le montre l'image ci-dessous.





- INSTALLATION JDK11 :

pour windows:

- Télécharger et installer JDK11

Pour ce projet nous avons utilisé **la version 11.0.8** que nous avons téléchargé de ce site

lien :

https://www.openlogic.com/openjdk-downloads?field_java_parent_version_target_id=All&field_operating_system_target_id=426&field_architecture_target_id=All&field_java_package_target_id=All&fbclid=IwAR3c-Dl-heDTvhx2wvFmAXPXacTVNeAq0bxkJK6z81ATkQUTkzRGYZFT99w

Pour linux:

- Avant de commencer, on peut vérifier la version actuelle de java avec la commande suivante : `java --version`
- Pour installer JDK 11, nous utilisons les commandes suivantes :
`sudo apt update`
`sudo apt install openjdk-8-jdk -y`
- Changer la version de java avec la commande suivante : `sudo update-alternatives --config java`

```
ilyes@jihene-HP:~$ sudo update-alternatives --config java
[sudo] password for ilyes:
There are 3 choices for the alternative java (providing /usr/bin/java).

  Selection    Path                                            Priority  Status
  -----
*  0            /usr/lib/jvm/java-11-openjdk-amd64/bin/java    1111     auto mode
    1            /opt/jdk/jdk1.8.0_231/bin/java                 100      manual mode
    2            /usr/lib/jvm/java-11-openjdk-amd64/bin/java    1111     manual mode
    3            /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java  1081     manual mode

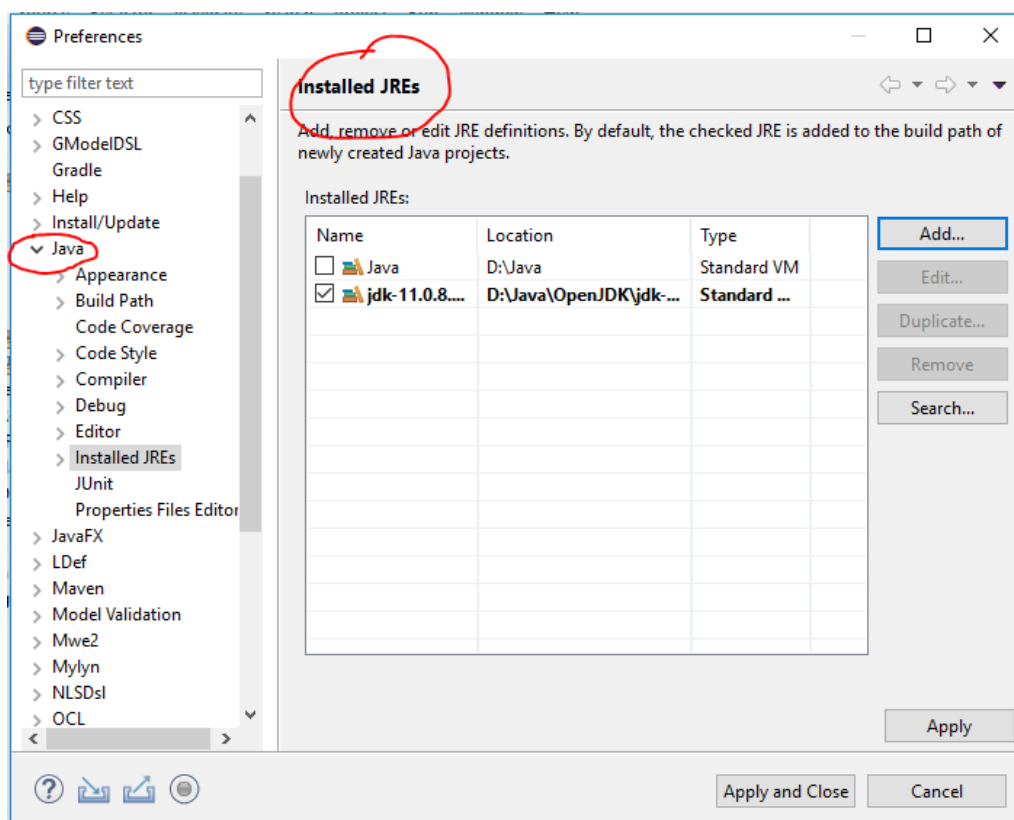
Press <enter> to keep the current choice[*], or type selection number:
```

Choisissez le numéro associé à la version de java-11-openjdk pour l'utiliser comme valeur par défaut, ou appuyez sur ENTRÉE pour laisser les paramètres actuels en place

- Configurez le JDK dans Eclipse:

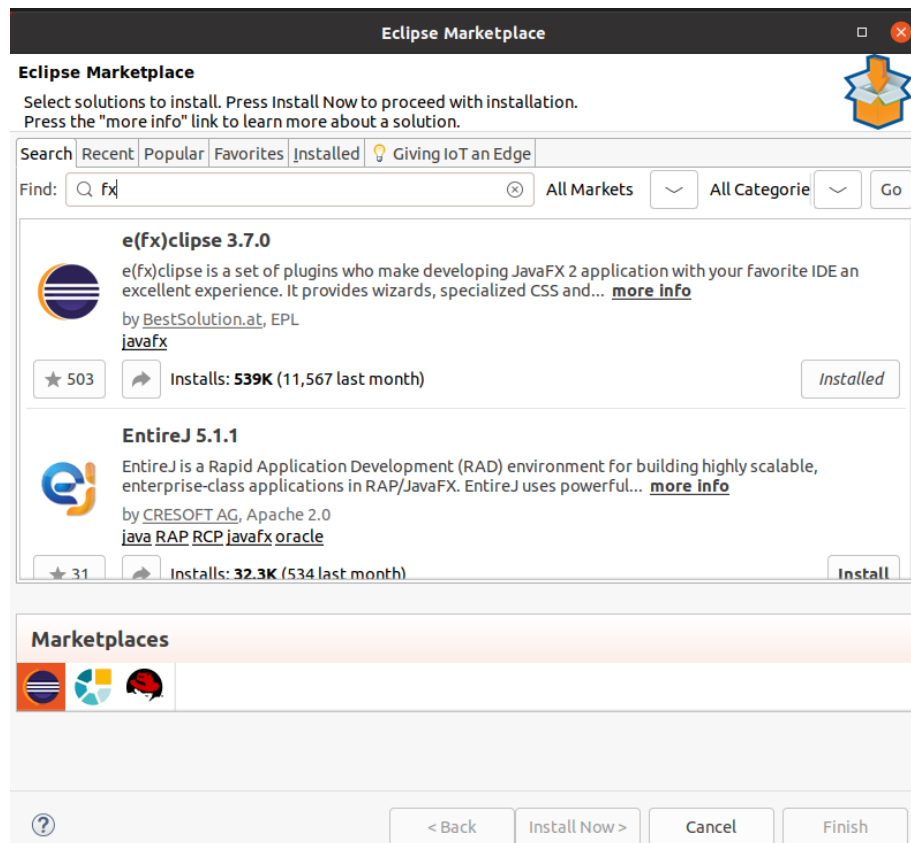
Ensuite on a ajouté java 11 à Eclipse, de la manière suivante:

Eclipse -> Window -> Preferences -> Java -> Installed JREs -> Add.



- INSTALLATION JavaFX :

Pour installer javaFX, on ouvre Eclipse ensuite on clique sur Help > Eclipse Marketplace, après on cherche



Nous devons installer aussi le SDK pour javaFX, en effet avant le JDK 11, javaFX était nativement intégré dans le JDK, mais à partir du JDK 11 une installation et une configuration sont nécessaires pour faire fonctionner le projet. Pour cela nous avons téléchargé la version javaFX17 de ce lien (qui nécessite au moins un JDK 11 pour fonctionner)

Lien :

https://gluonhq.com/products/javafx/?fbclid=IwAR2ky7jCDsKD6x_-S6qxlUcHqidWLlpbCqk3hnqgl8_XxQ0ugB5oqqO4

Nous terminons avec la configuration dans eclipse, nous allons en effet, ajouter le path du fichier téléchargé dans une configuration de la manière suivante :

, de la manière suivante:

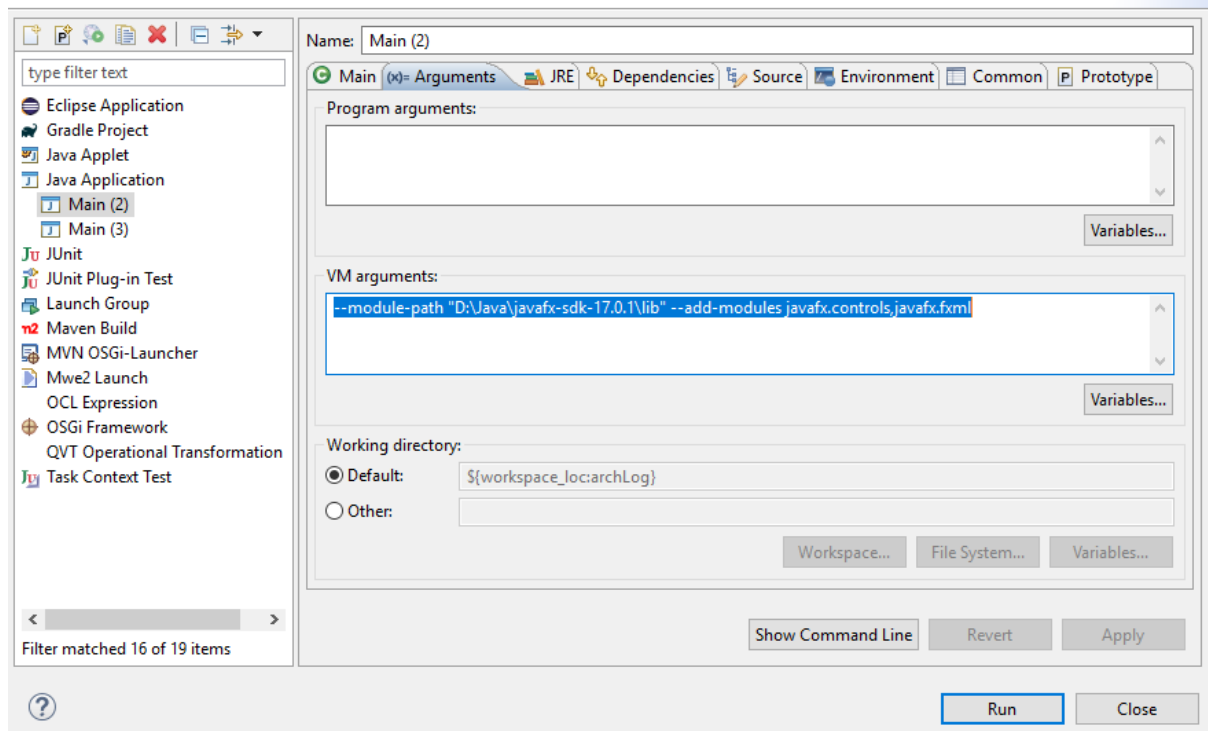
bouton run -> run configurations -> arguments -> ajouter dans Vm arguments :

```
--module-path "Path\lib" --add-modules javafx.controls,javafx.fxml
```

Run Configurations

Create, manage, and run configurations

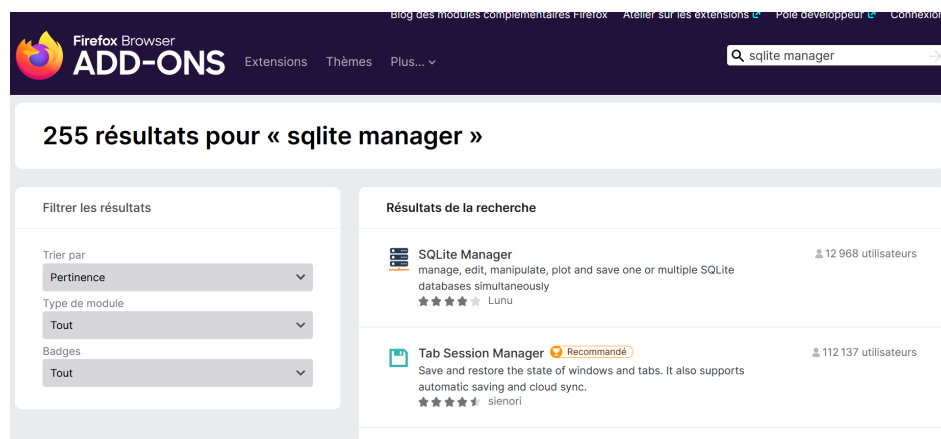
Run a Java application



- INSTALLATION SQLite:

Pour utiliser SQLite on a opté pour une solution simple qui est l'add-ons de sqlite dans le navigateur Firefox. Pour cela on a suivi ces étapes:

- Rechercher l'add-ons SQLite manager dans firefox



- Ajouter à Firefox



SQLite Manager

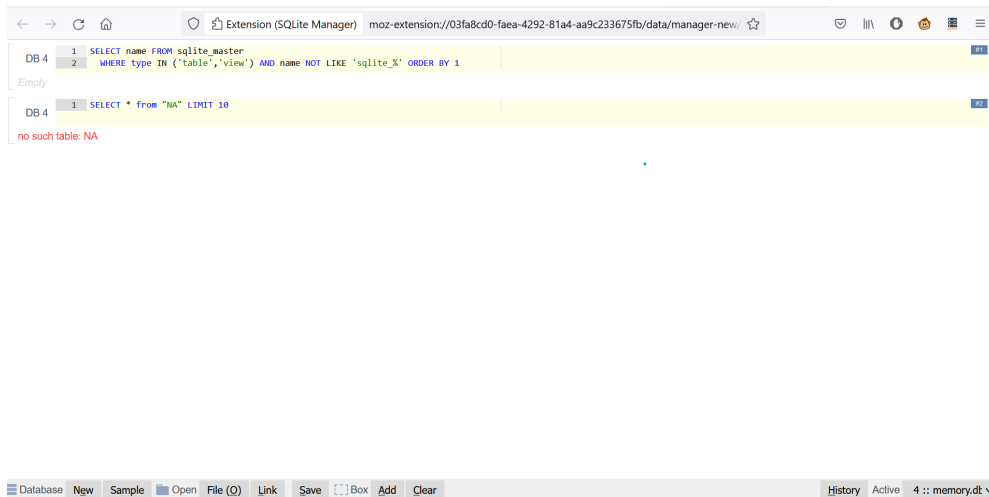
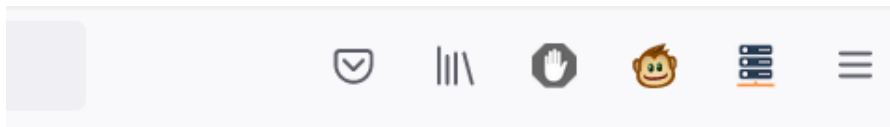
par Lunu

manage, edit, manipulate, plot and save one or multiple SQLite databases simultaneously

Ajouter à Firefox

- Redémarrer Firefox:

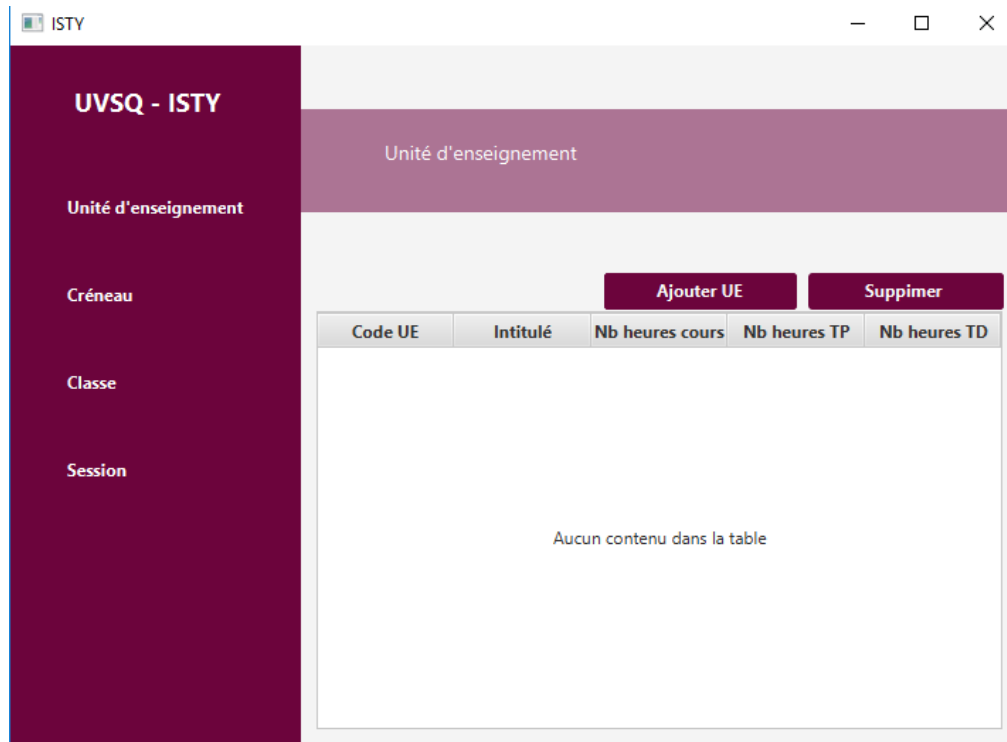
Après avoir redémarrer Firefox on trouve l'icône SQLite dans la barre du Firefox, et pour le lancer il suffit de cliquer sur l'icône.



- **Installation du projet:**

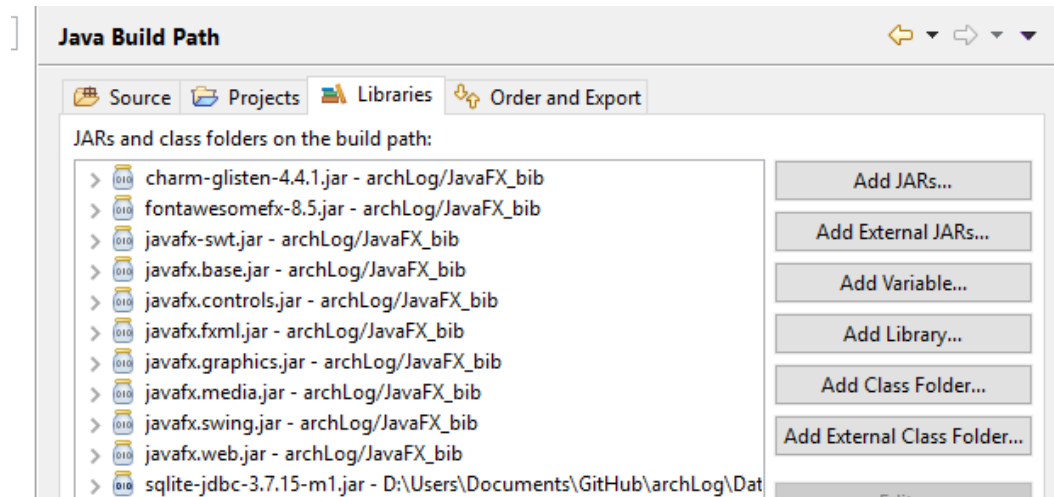
Une fois l'environnement mis en place, il suffit de télécharger le fichier zip du projet, de l'extraire à l'emplacement désiré puis suivre les étapes suivantes:
 démarrer un nouveau projet Eclipse -> Import from existing projects -> Select root directory -> sélectionner le dossier extrait -> Finish

En compilant le main sur Eclipse, l'utilisateur aura accès à l'interface suivante de notre projet:



ATTENTION:

- **Vérifiez le chemin de construction** : assurez-vous que "sqlite-jdbc-3.7.15-m1.jar" ainsi que les librairies dans le fichier javaFX_bib sont dans les bibliothèques référencées comme l'image ci-dessous



Si ce n'est pas le cas:

aller sur project -> properties-> java build path -> add External jars -> ajouter la bib
 sqlite-jdbc-3.7.15-m1.jar (dans le dossier Database_bib) ainsi que toutes les bib dans le
 dossier javafx_bib

- **Changement de path de la base de donn es:** Pour le changement de base de donn es. Il nous suffit de changer le path vers la nouvelle base. Le changement de path se fait directement dans la classe Connexiondb, dans la fonction connexiondb().

```
public Connexiondb() {
    DBPath = "Your/path/to/the/dataBase";
}
```

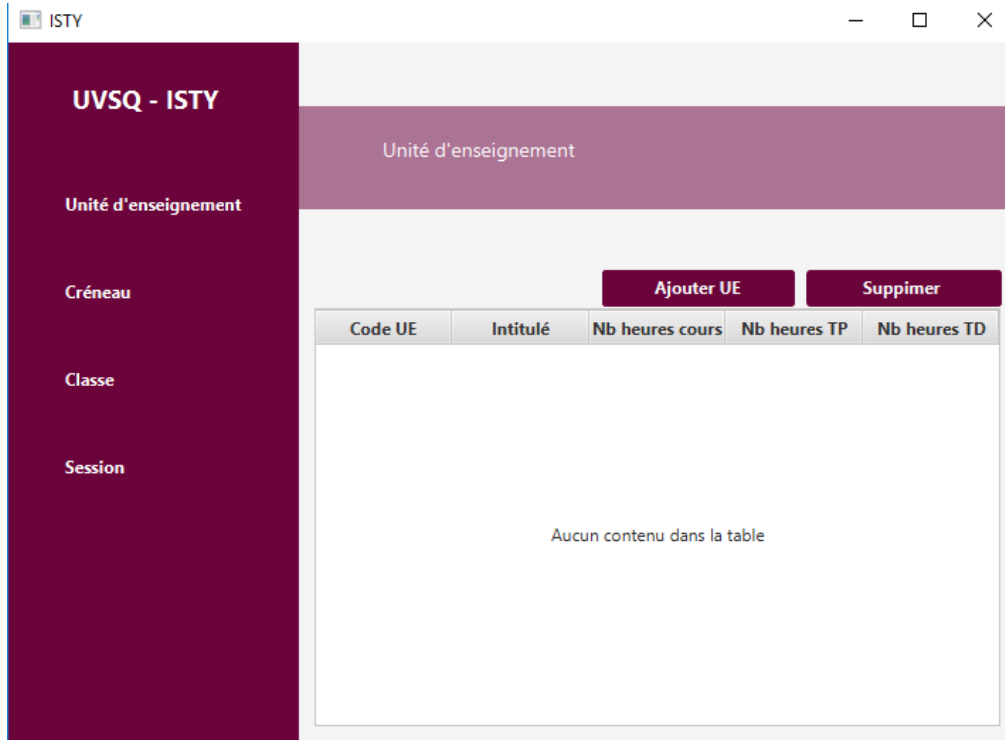
Comme on peut le voir nous avons un path qui est pris par la fonction connexiondb. Il nous suffit de le changer vers la nouvelle base de donnee.

ATTENTION:

Nous vous conseillons fortement d'utiliser la base que nous vous fournissons:

- La base est correctement agenc e pour pouvoir accueillir les donn es.
- En cas de cr ation d'une nouvelle base veuillez   utiliser le script fourni dans l'archive. Si vous ne l'avez pas vous pouvez utiliser sqllite browser afin de visualiser les tables de la base actuelle et  ventuellement de les recrer.

Utilisation de la GUI:



Pour l'ajout, il suffit de cliquer sur Ajouter [nom de l'option].

Pour supprimer, il faut sélectionner la ligne que l'on souhaite supprimer et la cliquer sur le bouton supprimer.

Pour changer d'option à afficher, il faut cliquer sur celle souhaiter sur le panel de gauche.

Pour mettre à jour, il suffit de cliquer sur le bouton de l'option courante.

Procédure de désinstallation:

Pour désinstaller le projet il suffit de supprimer le projet d'Eclipse (suppression dans le workspace d'Eclipse).

Exigences

Exigences fonctionnelles	Etat
L'utilisateur peut créer/supprimer une unité d'enseignement	satisfaite
L'utilisateur peut créer/supprimer une classe	satisfaite
L'utilisateur peut créer/supprimer une créneau	satisfaite

L'utilisateur peut créer/supprimer une session	satisfaite
L'utilisateur peut changer une session de créneau temporel.	non satisfaite
L'utilisateur peut créer des sessions sur un ensemble de créneaux.	non satisfaite

Exigences d'architecture	Etat
Le système est constitué d'un seul exécutable	satisfaite
Le programme principal instancie deux objets : l'un implémentant l'IHM et l'autre exposant les fonctionnalités	non satisfaite
Les communications entre IHM et fonctions passent par une unique interface Java	satisfaite
Les requêtes vont uniquement de l'IHM vers l'objet exposant les fonctions	satisfaite
Les opérations exposées par l'interface sont de 4 types(lecture/écriture/édition/suppression)	pas de modification
Les classes implémentant l'IHM et les fonctionnalités sont packagées dans des JAR distincts	satisfaite
L'identifiant des objets créés est attribué par l'IHM	satisfaite

Exigences liées au système d'exploitation	Etat
Le système doit fonctionner sous Linux	satisfaite
Le système doit fonctionner sous Windows	satisfaite
Le système doit fonctionner sous Mac OS	Non testé
Le système doit être développé en Java 11	satisfaite

Exigences liées à la persistance des données	Etat
Après redémarrage du système il est dans le même état qu'avant son arrêt (données)	satisfaite

Il est possible de mettre à jour l'IHM (automatiquement ou à la demande de l'utilisateur)

satisfaite

Anomalies

Problème sous certaines distributions linux avec javaFx. En effet la compilation fonctionne sur Ubuntu mais nous n'avons pas réussi à le faire tourner sous Arch. Nous n'avons pas réussi à identifier le problème.

Tous les tests n'étaient pas réalisés : l'utilisateur peut sélectionner une heure de fin inférieur à l'heure de début

Procédure de compilation

Pour la procédure de compilation nous utilisons Eclipse. Il suffit de faire un run.

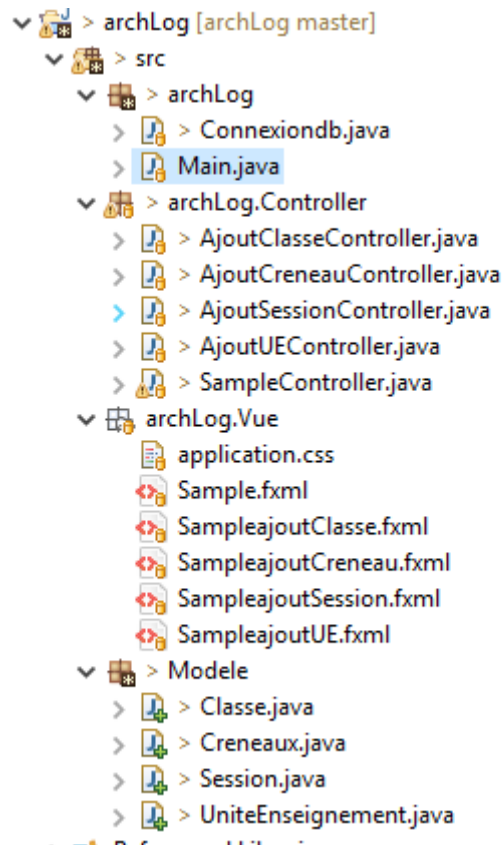
Pour exécuter le jar de notre application, il faut lancer le jar avec la commande suivante:

```
java --module-path "D:\Java\javafx-sdk-17.0.1\lib" --add-modules  
javafx.controls,javafx.fxml -jar rendu.jar
```

Fonctionnement de l'application :

Architecture:

L'application a été réalisée avec Java 11 et la bibliothèque javaFX pour l'interface.



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

Le package modèle contient les classes entités liées à la base de données de mon application.

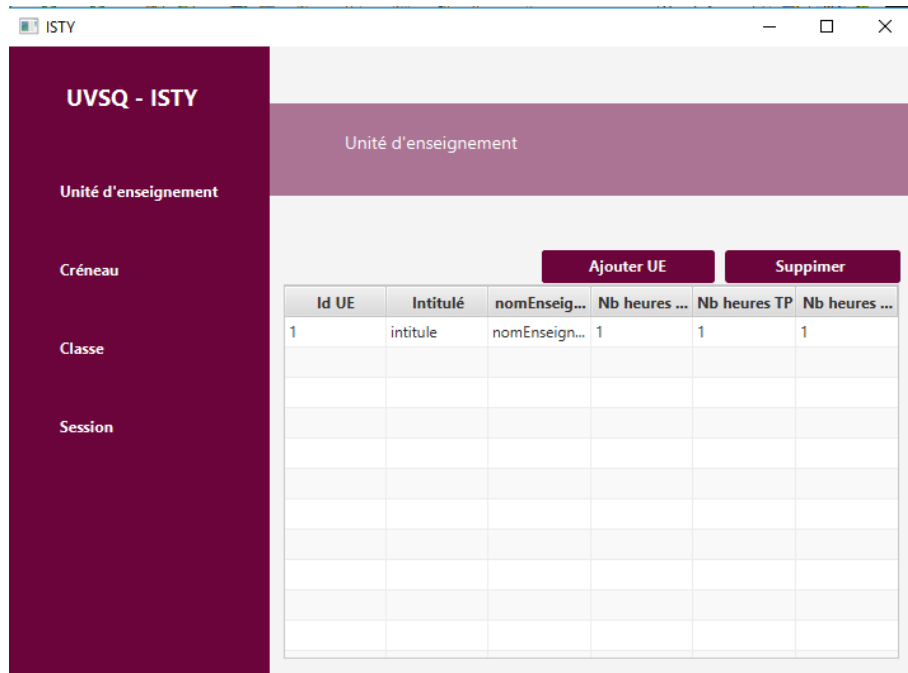
Le package Vue concerne l'implémentation de l'interface graphique.

Le package Controller contient tous les contrôleurs de notre application.

Interfaces de l'application:

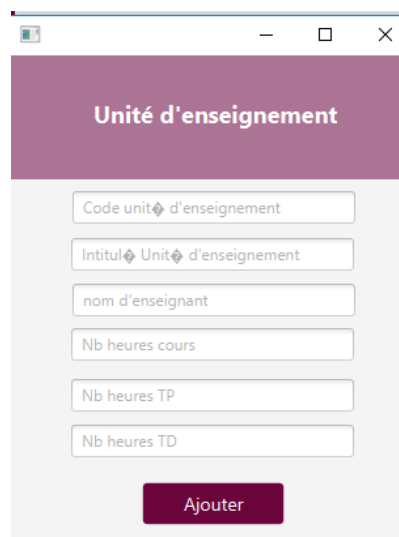
L'interface de l'application est composée de 4 écrans, chaque écran gère une entité à savoir les unités d'enseignements, les créneaux temporels, les promotions/classes et les sessions.

1- écran unité d'enseignement:



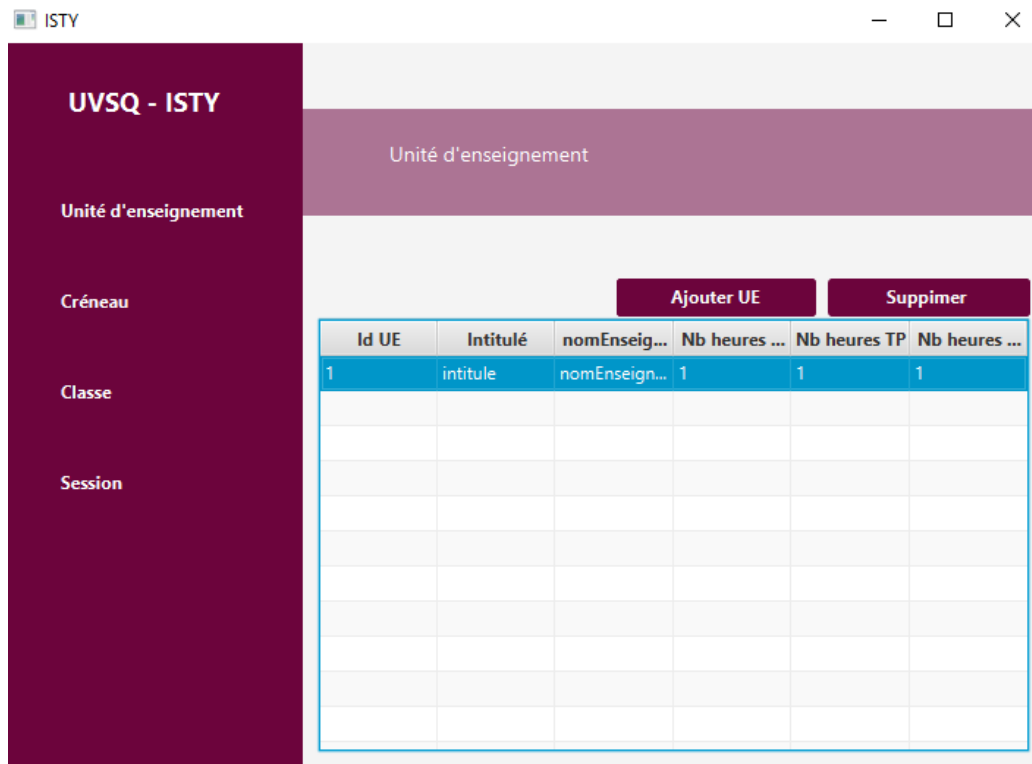
Id UE	Intitulé	nomEnseign...	Nb heures ...	Nb heures TP	Nb heures ...
1	intitule	nomEnseign...	1	1	1

L'écran unité d'enseignement permet à l'utilisateur d'afficher, de créer ou supprimer une UE. A l'initialisation, L'IHM fait appel au constructeur pour récupérer toutes les salles et les afficher dans le tableau.
 Pour ajouter une UE il suffit de cliquer sur le bouton **Ajouter UR**, on obtient par la suite l'écran affiché ci-dessous.



Code unité d'enseignement
 Intitulé Unité d'enseignement
 nom d'enseignant
 Nb heures cours
 Nb heures TP
 Nb heures TD
 Ajouter

Cette interface est composée de plusieurs textField permettant d'entrer les informations d'une unité d'enseignement. Le bouton Ajouter fait appel à l'interface java pour la création d'une unité d'enseignement et son ajout dans la base de données.

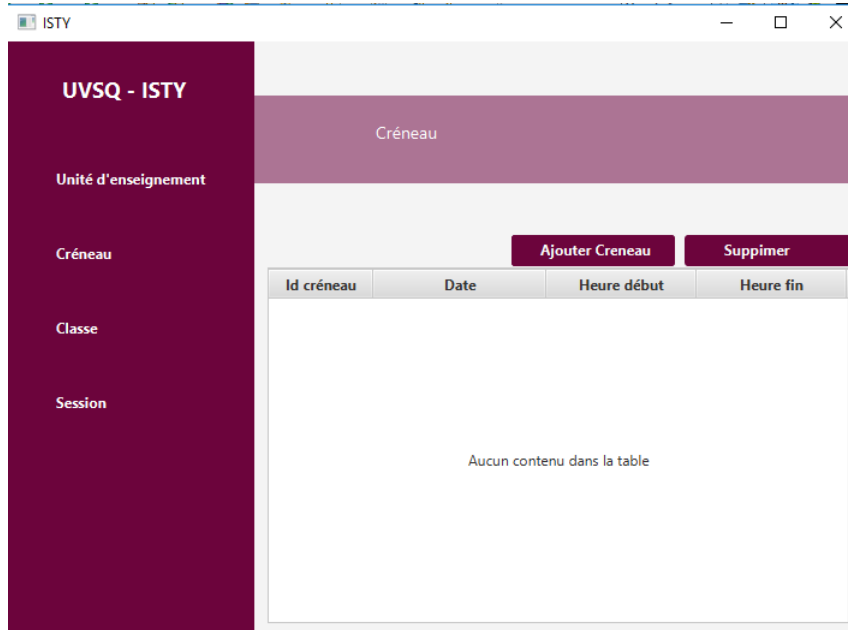


The screenshot shows a web application window titled 'ISTY'. On the left is a dark red sidebar with navigation links: 'Unité d'enseignement' (selected), 'Créneau', 'Classe', and 'Session'. The main content area has a header 'Unité d'enseignement' and two buttons: 'Ajouter UE' and 'Supprimer'. Below these is a table with the following data:

Id UE	Intitulé	nomEnseign...	Nb heures ...	Nb heures TP	Nb heures ...
1	intitule	nomEnseign...	1	1	1

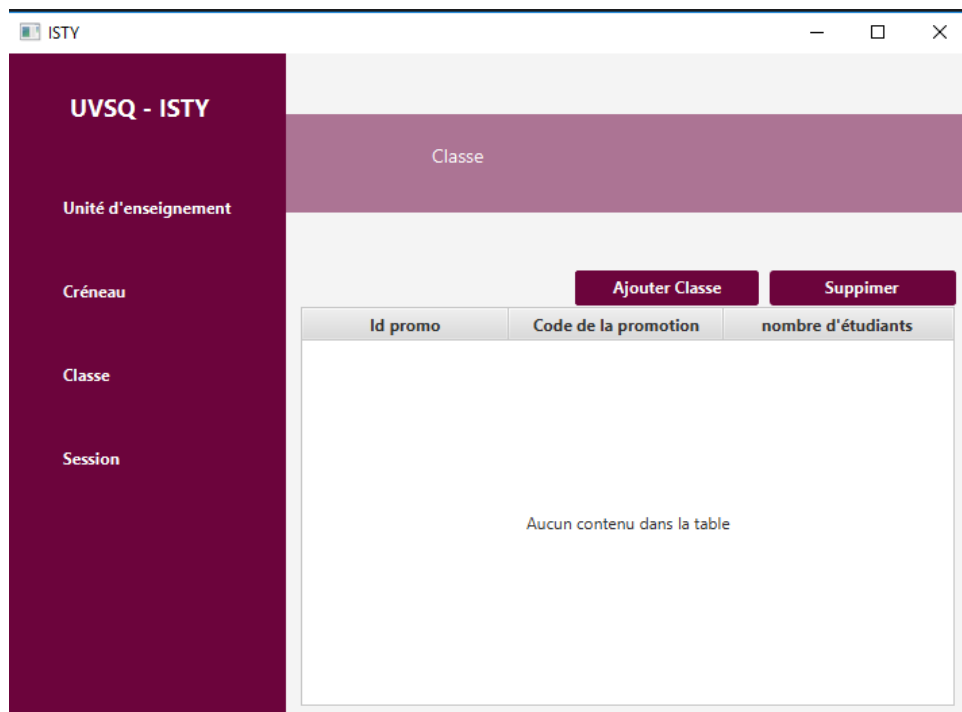
Pour supprimer une unité d'enseignement, nous devons sélectionner la ligne correspondante et cliquer sur le bouton supprimer. l'IHM fait donc un appel à la fonctionnalité de suppression de l'interface Java pour supprimer la salle de la base de données.

2- Ecran créneau



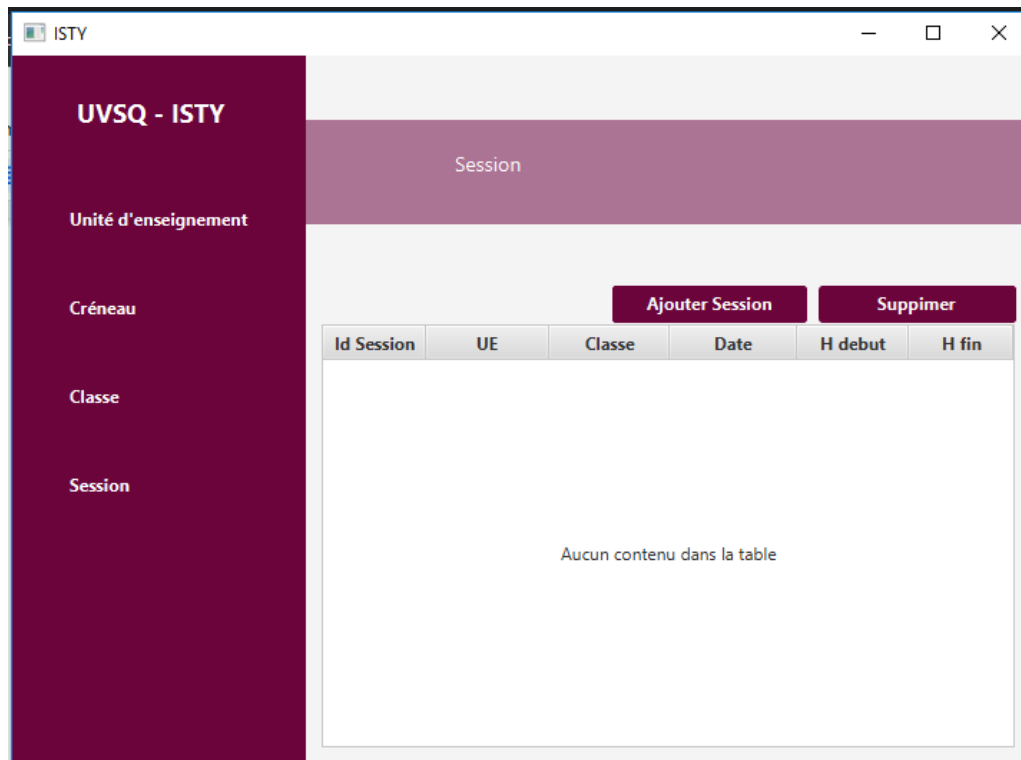
En cliquant sur le bouton créneau, nous affichons l'écran de gestion des créneaux. Cet écran nous permet d'afficher tous les créneaux stockés dans la base de données. Comme précédemment le bouton supprimer fait appel à la fonctionnalité de suppression. Le bouton ajouter créneau nous redirige vers une autre interface dans laquelle nous réaliserons la saisie des différentes informations du créneau à ajouter.

3-Écran classe



L'écran classe permet de gérer les classes stockées dans la base de données en les affichant dans le tableau, il permet aussi la suppression et l'ajout de nouvelles classes/promotions à travers les boutons ajouter classe et supprimer.

3-Écran Session



L'écran Session permet aussi l'affichage, l'ajout et la suppression des sessions. Il fonctionne de la même façon que les autres écrans.

Annexe

Script SQL

```
CREATE TABLE promo(  
    idPromo      integer      primary key autoincrement,  
    codePromo    text,  
    Etudiant     int  
);  
  
CREATE TABLE crenau(  
    idPromo      integer      primary key autoincrement,  
    date         text,  
    heureDebut   text,  
    heureFin     text  
);  
  
CREATE TABLE session(  
    idSession    integer      primary key autoincrement,  
    codePromo    text,  
    nbEtudiant   int  
);  
  
CREATE TABLE uniteEns(  
    idUE         integer      primary key autoincrement,  
    intitule     text,  
    nomEnseignement varchar(20),  
    nbHeuresCours int,  
    nbHeuresTP   int,  
    nbHeuresTD   int  
);
```