

# Secours 2030

## Détail de la partie «Développement d’une application avec IHM» (R2.02)

A. BELMOUHCINE et H. EVENAS

2024 - 2025

### 1 Architecture Modèle-Vue-Contrôleur (MVC)

Comme spécifié dans la partie **R2.01**, l’application finale devra suivre une architecture logicielle **MVC** (Modèle-Vue-Contrôleur), qui divise l’application en trois composants relativement indépendants :

- Le **modèle**, qui contient les données de l’application et les traitements associés. Il ne possède aucune logique concernant la présentation de ces données à l’utilisateur.
- La **vue**, qui est responsable de la présentation des données du modèle à l’utilisateur. Elle accède aux données mais ne traite ni leur signification ni les interactions possibles que l’utilisateur peut avoir avec ces données.
- Le **contrôleur**, qui agit comme intermédiaire entre la vue et le modèle. Il réagit aux événements initiés par la vue, généralement en déclenchant des opérations sur le modèle. Étant donné que la vue et le modèle sont souvent connectés par un mécanisme de notification, les modifications apportées au modèle sont automatiquement reflétées dans la vue.

### 2 Travail à réaliser

Un package **model** regroupant les classes **POJO** correspondant à votre BDD a été réalisé ou est en cours de réalisation dans le cadre de **R2.01**. Vous allez ajouter deux autres packages : **view** et **controller**. De plus, vous ajouterez au package **model** deux sous-packages : **data** et **dao**. Tous ces packages pourront contenir plusieurs classes.

1. Dans le package **model**, déplacez toutes les classes créées en **R2.01** vers le sous-package **data**. Dans le package **dao**, placez toutes les classes d’accès aux données qui serviront d’intermédiaire entre les objets du modèle et la BDD (voir le CM5 de **R2.02**). Votre application se connectera à une BDD **MySQL** ou **Oracle** via **JDBC** (voir le CM5 de **R2.02**).
2. Votre **contrôleur**, réalisé en **JavaFX**, devra permettre au minimum d’effectuer toutes les opérations **CRUD** sur votre BDD, de mettre à jour le modèle et d’exporter les données au format **CSV**. Votre contrôleur pourra également modifier la vue si nécessaire.
3. La **vue**, à réaliser en **JavaFX**, devra permettre d’activer les opérations fournies par le contrôleur et d’afficher les données. Il est recommandé que votre vue suive la maquette réalisée en **P3** de **R2.02**, bien que des modifications justifiées soient possibles.

Il est obligatoire de réaliser votre application en **JavaFX**. L’utilisation de classes **AWT** ou **Swing** n’est pas autorisée.

### 3 Livrable

Chaque groupe devra remettre une archive contenant :

- le code source de l'application, commenté en français ou en anglais.
- Un fichier texte contenant les commandes de compilation et d'exécution de l'application. Ces commandes devront placer les fichiers compilés dans un dossier nommé *classes* et les fichiers sources doivent être placés dans un dossier nommé *sources*.
- Un PDF justifiant les changements appliqués à la maquette, s'il y en a.
- Une courte vidéo de démonstration de l'application, montrant un exemple pour chacune des opérations CRUD, une démonstration de l'export de fichier et, éventuellement, une démonstration de l'affichage d'un graphique.

Cette archive devra être remise au plus tard le **jeudi 19 juin 2025 à 23h59** sur l'espace Moodle de la **SAÉ S2.01**, en un exemplaire par groupe.

Les noms de tous les membres du groupe doivent être indiqués sur le rendu.

Le travail sera à réaliser lors des quatre dernières séances de **R2.02** et pendant les créneaux de **SAÉ**.

En plus de l'archive, chaque étudiant devra rendre **une page** contenant une description des différentes tâches réalisées ou en cours de réalisation, au plus tard **vendredi 13 juin 2025 à 23h59**.