

**Compte Rendu Projet IC**

Réalisation d’une petite application d’étude de système expert

**CS2**

**Groupe** **1**

**Auteurs** :

* **Benmessabih Toufik**
* **Lousra Issam**

*Table des matières*

**Présentation du projet 3**

**1 Travail demandé**

**Travail effectué 3**

**1 Choix de l’langage de programmation Java :**

**2 Choix du Framework JavaFx :**

**3 Choix de l’environnement de développement : Intellij IDE**

**4 Manuelle d’utilisation**

**Difficultés et inconvenants 4**

**Webographie 5**

Présentation du projet

**1.2 Travail demandé**

Bien expliqué dans la fiche : Mini-projet: Systèmes experts.pdf.

Travail effectué

1. **Choix de l’langage de programmation Java :**

**Java** est un langage de programmation orienté objet créé par James Gosling et Patrick Naughton présenté officiellement le 23 mai 1995 au **SunWorld**. La société Sun a été ensuite rachetée en 2009 par la société **Oracle** qui détient et maintient désormais Java.

1. **Choix du Framework JavaFx :**

**JavaFX** est un Framework et une bibliothèque d'interface utilisateur issue du projet **OpenJFX**, qui permet aux développeurs Java de créer une interface graphique pour des applications de bureau, des applications internet riches et des applications smartphones et tablettes tactiles.

1. **Choix de l’environnement de développement : Intellij IDE**

**Intellij IDEA** est un environnement de développement intégré écrit en Java pour le développement de logiciels informatiques.

Il est développé par **JetBrains.**

1. **Manuelle d’utilisation :**

Vous voulez voir Comment fonctionne l’application ? Fonctions disponible ? Et plus !

tous est expliqué dans le document « «*manuelle d’utilisation* »  avec des illustrations et exemples ainsi que la gestion des erreurs.

Difficultés et inconvénients

**JavaFX** n’est plus intégré dans le **JDK** depuis la version **9**, ce qui rend plus complexe l’export de ses bibliothèques ainsi que l’export de l’application en fin de développement.

1. **Intégration du projet sous JavaFX**

Comme *javaFX* n’est plus une partie intégrante du *JDK* depuis la version *9* (le projet utilise la version *jdk11* la plus stable et la plus utilisée),

il a fallu en plus de la création du projet *javaFX* sur *intellij*, télécharger *javaFX SDK* et importer *javaFX/lib*.

Certaine opérations graphiques comme *CheckComboBoxe* qui permet la sélection multiple dans une liste *ComboBox* nécessite une manipulation spécifique comme ***controlsfx***

1. **l’exportation de l’application en fichi.er exécutable:**

Apres avoir exporté l’application en fichier .*jar*, il faut prendre en considération les architectures des machine qui vont l’exécuté*, x64, x32*

Pour cela il faut réaliser *2 versions* de l’application, une pour chaque architecture car les bibliothèques importées dans chaque version ne peuvent être disponibles sur certaine architectures,

*Exemple* : pour *x64* on utilise *jdk11* avec *Javafx SDK11* qui est puissant et plus performant, alors que *jdk11* n’est pas supporté par les machines *x32*, donc on utilise *jdk8* avec *javafx* ce qui est limité et pour cela, il faudra réalisée *2 versions x64, x32.*

L’exécutable envoyée fonctionne sur machine *x64*,

*Si* votre machine est d’architecture *x32*, veiller nous le notifier et on fera notre mieux pour vous envoyer la version *32bit*.

Webographie

• Site officiel framework JavaFX

<https://openjfx.io/index.html>

• JavaFX tutorials

<http://tutorials.jenkov.com/javafx/index.html>

• introduction JavaFx

[https://openjfx.io/openjfx-docs/#install-java](https://openjfx.io/openjfx-docs/%23install-java)

• controlfx open source documentation

<https://github.com/controlsfx/controlsfx>

• Forum d’entraide

<http://stackoverflow.com/>