CAHIER DE CHARGES

DOSSIER D’ANALYSE ET D’IMPLEMENTATION DU PROJET

Par Zafack Billy Ruffin

3GI

SOMMAIRE

**OBJECTIF**

**PRESENTATION DE LA METHODE IMPLEMENTEE**

**PRESENTATION DU DEVELOPPEMENT**

PRESENTATION DU CODE SOURCE

**AMELIORATIONS**

**CONCLUSION**

**BIBLIOGRAPHIE**

OBJECTIF DU PROJET

L’objectif de ce projet est la conception et l’implémentation d’un logiciel (programme) de gestion d’étudiants d’une institution.

PRESENTATION DE LA METHODE IMPLEMENTEE

Généralités

Le développement de ce projet a eu à implémenter plusieurs différentes notions du langage Java. Je vais pour une compréhension accentuée et précise de l’implémentation de ce projet diviser cette implémentation en deux parties principales notamment :

1. La Construction de l’interface graphique.
2. La Sauvegarde et l’accès aux données à travers la sérialisation.

La partie graphique a comportée l’utilisation de la classe **JFrame**  et de ses différentes composantes que nous allons examiner dans ce cahier de charges.

La partie sur la sauvegarde des données dans le fichier à consister la sérialisation des données dans données dans un fichier et la deserialisation éventuelle pendant les opérations de **Recherche**, **Suppression** et **Modification.**

PRESENTATION DU DEVELOPPEMENT

1. Partie Graphique

La partie graphique du projet consiste de deux fenêtres, une boite de dialogue et plusieurs boites de Message(JOptionPanes).

***Fenêtres :***

Les deux fenêtres dans ce projet sont tous des héritages de la classe JFrame. Nous avons notamment les fenêtres suivantes :

1. **Fenêtre Principale :**

La fenêtre principale est représentée par la classe Window du code source. Elle contient également les composantes graphiques suivantes :

1. **Une Barre De menu contenant les menus principaux :**

**- Fichier**

Elle contient les sous menus suivants :

**\* *Nouveau*** qui permet de créer un nouvel étudiant.

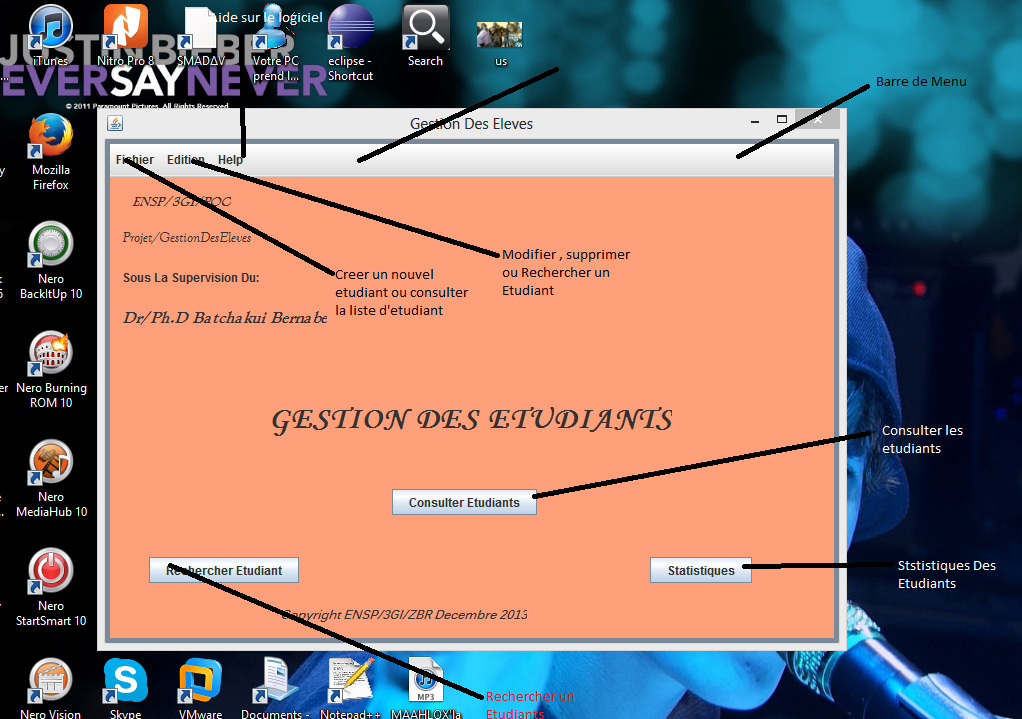
**\* *Consulter*** qui permet d’afficher la table contenant les informations sur les étudiants inscrits.

**\* *Quitter*** qui permet de fermer l’application.

-**Edition**

Elle contient les sous menus suivants :

* ***Rechercher*** qui permet de rechercher un étudiant dont on entrera le matricule.
* ***Modifier*** qui permet de modifier les informations sur un étudiant inscrits.
* ***Supprimer*** qui permet de supprimer un étudiant inscrit dont il faudra entrer le matricule.



**Fig. Fenêtre Principale**

**-Quitter**

Contient les sous menus suivants :

\****Mode Opératoire***

Affiche un mode opératoire brève du logiciel.

\****A propos***

\****Java DOC***

Donne un le chemin d’accès à la Java DOC c.à.d. …\Projet POO\doc

1. **Le Button Statistiques :**

Permet d’afficher les statistiques sur les étudiants enregistrées notamment

-Le nombre d’étudiants

-Le nombre d’étudiants masculins.

-Le nombre d’étudiants féminins.

-Le nombre d’étudiants parlant l’Anglais.

-Le nombre d’étudiants parlant le Français

1. **Button Rechercher**

Permet de rechercher un étudiant parmi les étudiants inscrit en entrant son matricule.

1. **Le Button ConsulterEtudiants**

Permet de consulter la liste des étudiants inscrits.

1. **Fenêtre Consulter**

Cette fenêtre contient une table des différents Etudiants inscrits ainsi que des informations personnelles sur eux. Nous avons notamment les colonnes suivantes :

**-Matricule**

**-Nom**

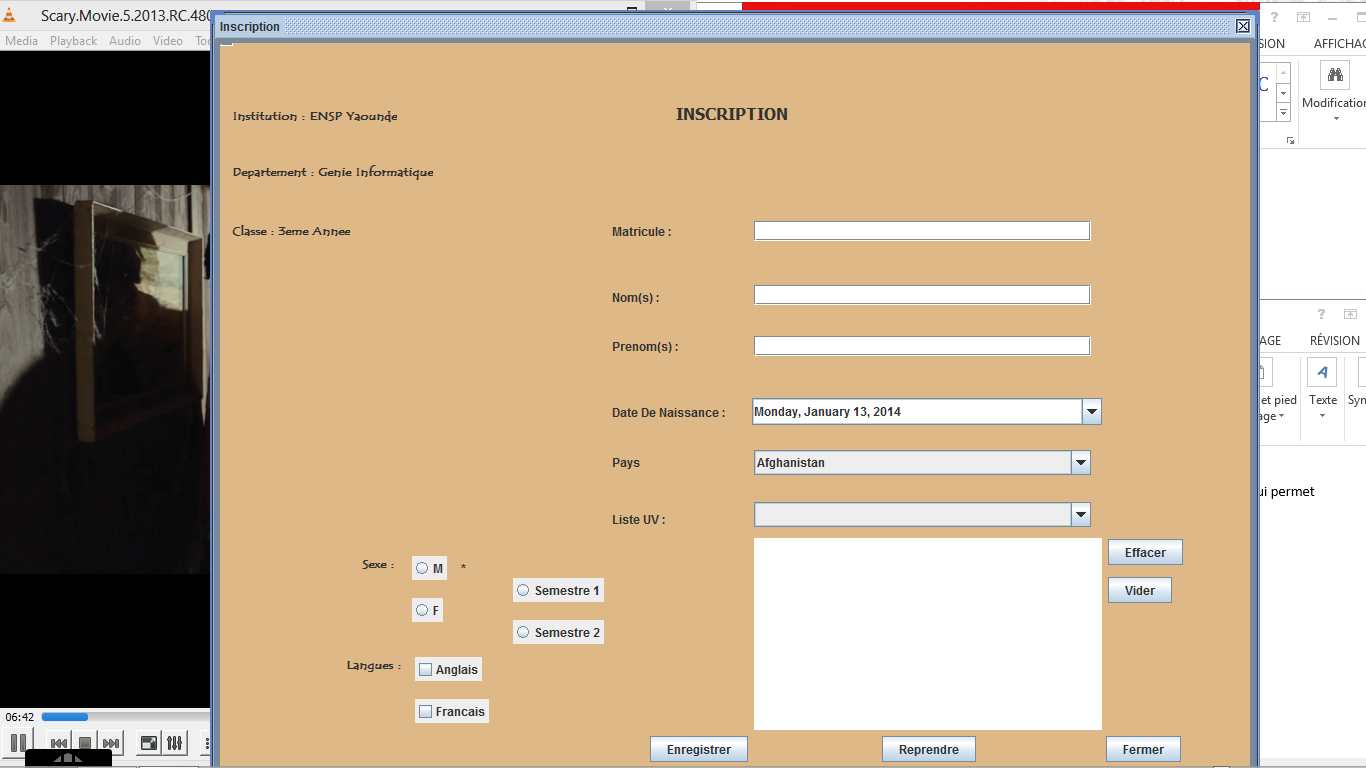
**-Prénom**

**-Pays**

**-Liste D’UVs**

***Boite De Dialogue***

Il y a une fenêtre de dialogue principale. Elle est la boite dialogue Inscription qui permet d’entrer les informations sur un étudiant pendant l’inscription.



**Fig. Inscription**

Elle contient les champs suivants :

-***Matricule :*** Ceci est un champ obligatoire. Elle est unique et permet d’identifier de façon unique chaque étudiant inscrit. C’est un champ de texte.

Il est alors rendu impossible d’inscrit plus d’un étudiant ayant le même matricule. Un mécanisme a été mis en place pour assurer cela.

***-Nom :***

Ceci est un champ de texte(**JTexteField**)

***-Prénom :***

Ceci est un champ de texte.

***-Date De Naissance :***

Ceci est un **JCalendarCombo** qui permet de sélectionner une date précise de naissance.

***-Pays***

Ceci est un **JComboBox** ayant la liste de pays et permet de choisir un pays d’origine pour l’étudiant à inscrire.

-***Liste d’UVs***

Ici il y a d’abord un **JComboBox** qui permet de choisir une des UVS de l’étudiant qui est ajouté directement à une liste en dessous.

***-Sexe :***

Ceci est également un champ obligatoire. C’est un ensemble de **JRadioButtons** à sélection exclusive.

***-Langues Parlées par l’étudiant***

Ceci est un ensemble de **JCheckBox** indiquant les langues possibles (Anglais, Français).

Il y a également un ensemble de buttons qui permet une inscription plus efficace.

Nous avons notamment :

***-Button Effacer***

Qui permet d’effacer la dernière UV sur la liste.

***-Button Vider***

***-Button Enregistrer***

Qui permet d’enregistrer l’étudiant.

***-Button Reprendre***

Qui permet d’initialiser la fiche d’inscription.

***-Button Quitter***

Qui permet de fermer la boite de dialogue.

1. Il y a également un très grand nombre de **JOPtionPane** qui permettent de gérer les transitions entre les différentes fenêtres et boite de dialogue.

2) Partie D’Access Aux Données

Comme nous allons le voir dans la partie code source, il y a un ArrayList table, un attribut **static** de la classe Inscription qui contient les différents étudiants et leurs information après que l’on les obtient dans le fichier **« projet1.txt ».** En effet ce tableau est successivement sérialisée pour l’ajout d’un étudiant dans le fichier et deserialisee pour permettre la modification ou la suppression d’un étudiant de cette liste.

PRESENTATION DU CODE SOURCE

Ce projet est constitué de cinq classes principales. Nous avons notamment :

-La Classe Window

-La Classe Inscription

-La Classe Etudiant

-La Classe Consult

-La Classe Modification

1) La Classe Window

Cette classe gère la fenêtre principale du projet. Elle hérite de la classe JFrame. Elle contient les différentes composantes principales de la classe comme attribut. Elle implémente également l’interface **ActionListener** qui permet de gérer les différents évènements sur les composantes de la fenêtre principale.

Il est également important de noter qu’elle contient la classe main du projet.

1. La Classe Inscription

Cette classe gère la fiche d’inscription d’un étudiant. Elle hérite également de **JFrame** pour gérer son interface graphique et implémente l’interface ActionListener pour gérer les différents évènements sur les différentes composantes de la boite de dialogue.

1. La Classe Etudiant

Cette classe est une représentation d’un étudiant. En effet elle contient les différents « attributs d’un étudiant ». Quand l’on enregistre un étudiant, cette classe est instanciée et ses différents champs (attributs) sont remplis avec les valeurs entrées dans la fiche d’inscription. Cet objet est ajouté a l’attribut static table et cette table est sérialisée.

1. La Classe Consult

Elle permet de construire le tableau des étudiants enregistrés. Elle hérite de la classe JFrame. Elle est en quelque sorte une représentation visuel de l’attribut static table.

1. La Classe Modification

Elle hérite de la classe Inscription et permet la modification d’un étudiant. Comme tout héritage de classe en Java elle hérite de tous les attributs et méthodes de la classe Inscription. Certaines de ses méthodes (surtout **actionPerformed**) on est redéfinies pour satisfaire le cas particulier de la modification d’un étudiant inscrit.

Pour une compréhension plus détaillée du code source vous pouvez accéder a la documentation Java DOC à travers les champs …\Project POO\doc. Egalement le code source est intensivement commenté.

AMELIORATIONS

D’autres améliorations possibles sur projet qui seront éventuellement implémentées

-L’intégration d’un système de notation pour les étudiants inscrit.

-L’intégration de différents enseignants pour les Unités de Valeurs.

Conclusion

Le logiciel résultant de ce projet est très efficace et permet de gérer les étudiants d’une institution. Un très grand nombre de fonctionnalités tel que :

-L’inscription de nouvel étudiant.

- La consultation des étudiants inscrit.

- La recherche d’un étudiant inscrit à travers son matricule.

- La suppression d’un étudiant inscrit.

- La modification des informations d’un étudiant inscrit.

Néanmoins, je continuerai à travailler sur ce projet pour intégrer un système de notes pour les étudiants. Il est nécessaire d’acclamer les différentes propriétés du langage de programmation Java qui a permis d’effectuer le projet à succès.