FICHIER D’AIDE

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. Contrairement à la version antérieure (Classique-Produit au premier semestre), cette version gère les étudiants à travers une base de données.

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 l’interface JDBC.

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 une base de données à consiste la connexion a une base de données et la récupération éventuelle pendant les opérations de **Recherche**, **Suppression** et **Modification.**

PRESENTATION DU DEVELOPPEMENT

1. Partie Graphique

La partie graphique du projet consiste d’une fenêtre, quatre boites de dialogue et plusieurs boites de Message (JOptionPanes).

***Démarrage :***

1. **Choix De L’Interface :**

Ce projet(ou encore « mini-projet ») est liée a un autre projet de gestion des étudiants avec sérialisation des objets et interface classique. Il est alors proposé de ici de choisir l’interface. Saisir « Professionnelle » et valider.

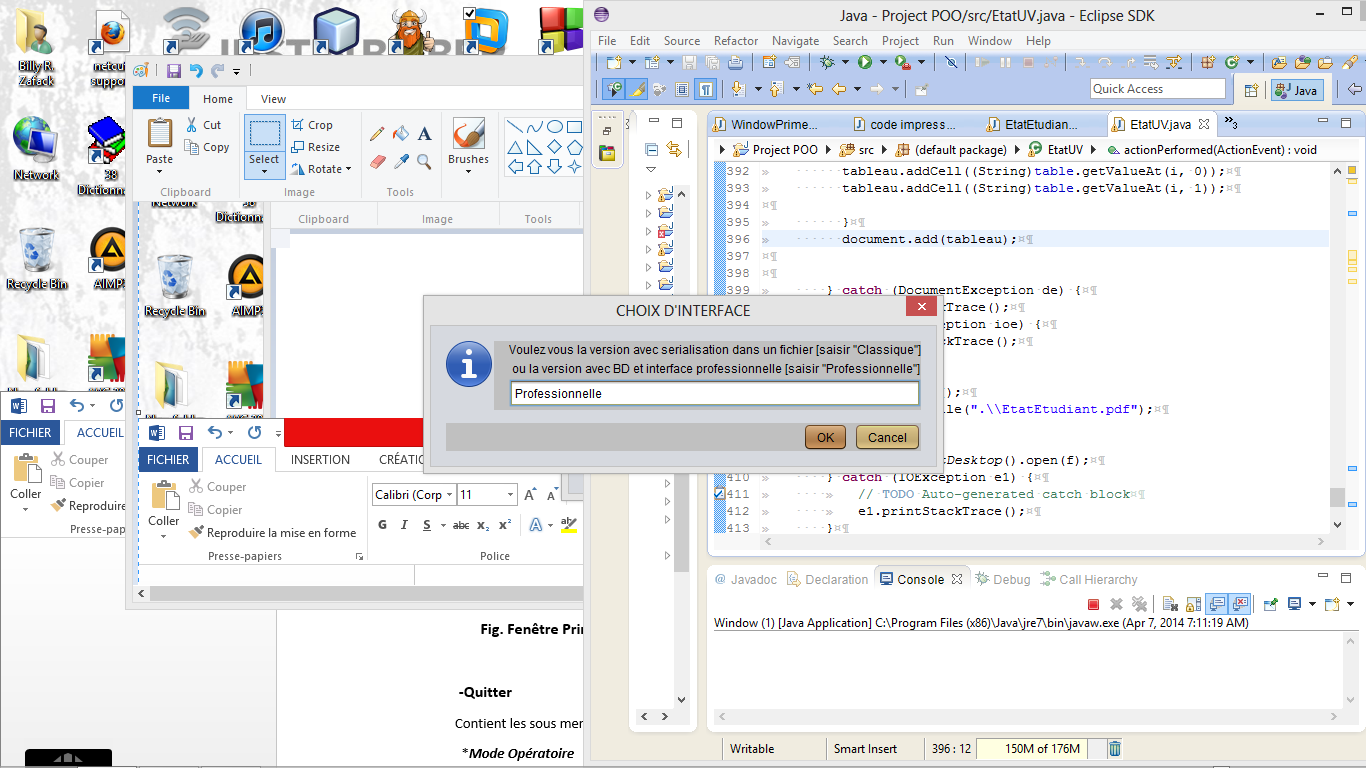
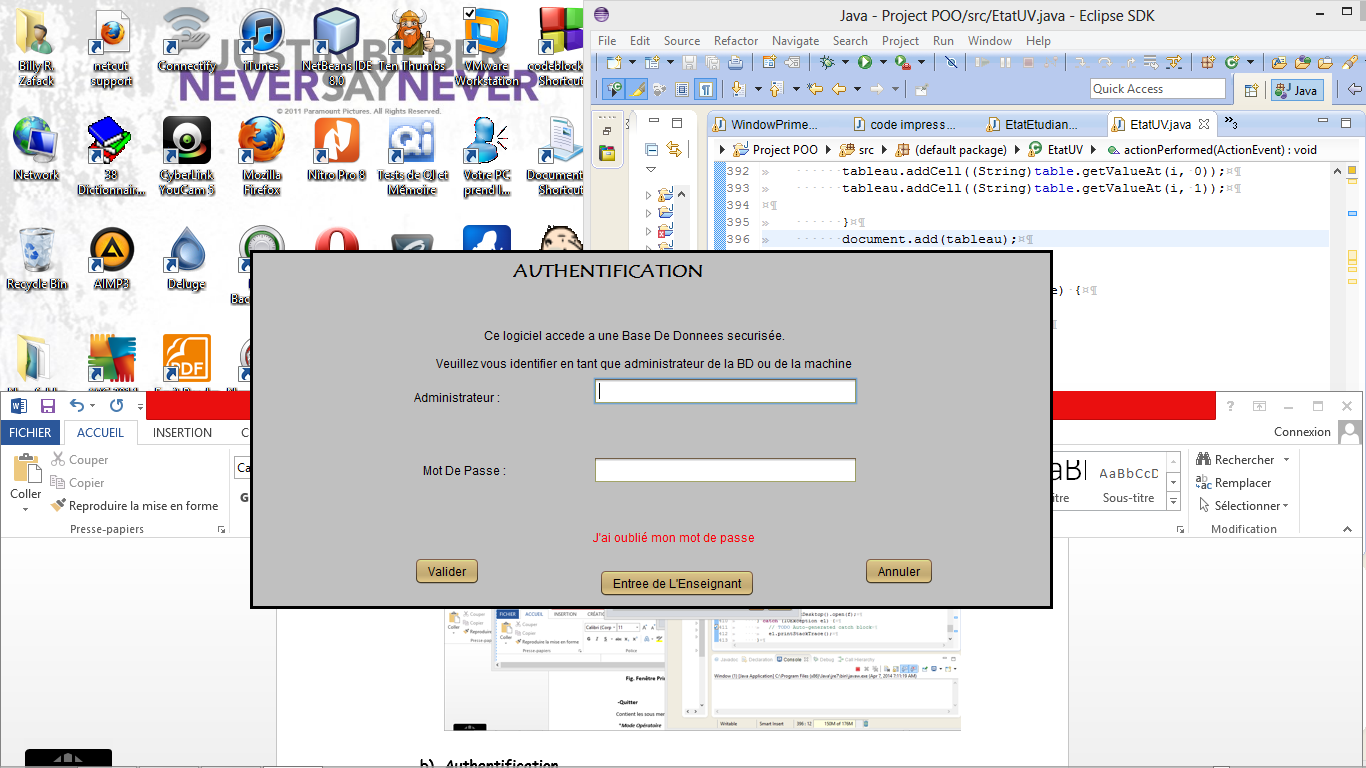


Fig CHOIX DE L’INTERFACE

1. **Authentification**

Cliquer sur **« Entrée de l’enseignant »** pour accéder au logiciel sans avoir à s’authentifier. En temps normale, une authentification sera requis puisque le logiciel accède a un base de données sécurisée dans son déroulement.



**FIG Authentification**

***Fenêtres :***

La fenêtre dans ce projet est un héritage de la classe JFrame.

1. **Fenêtre Principale :**

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

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

**- Fichier**

Elle contient le sous menu:

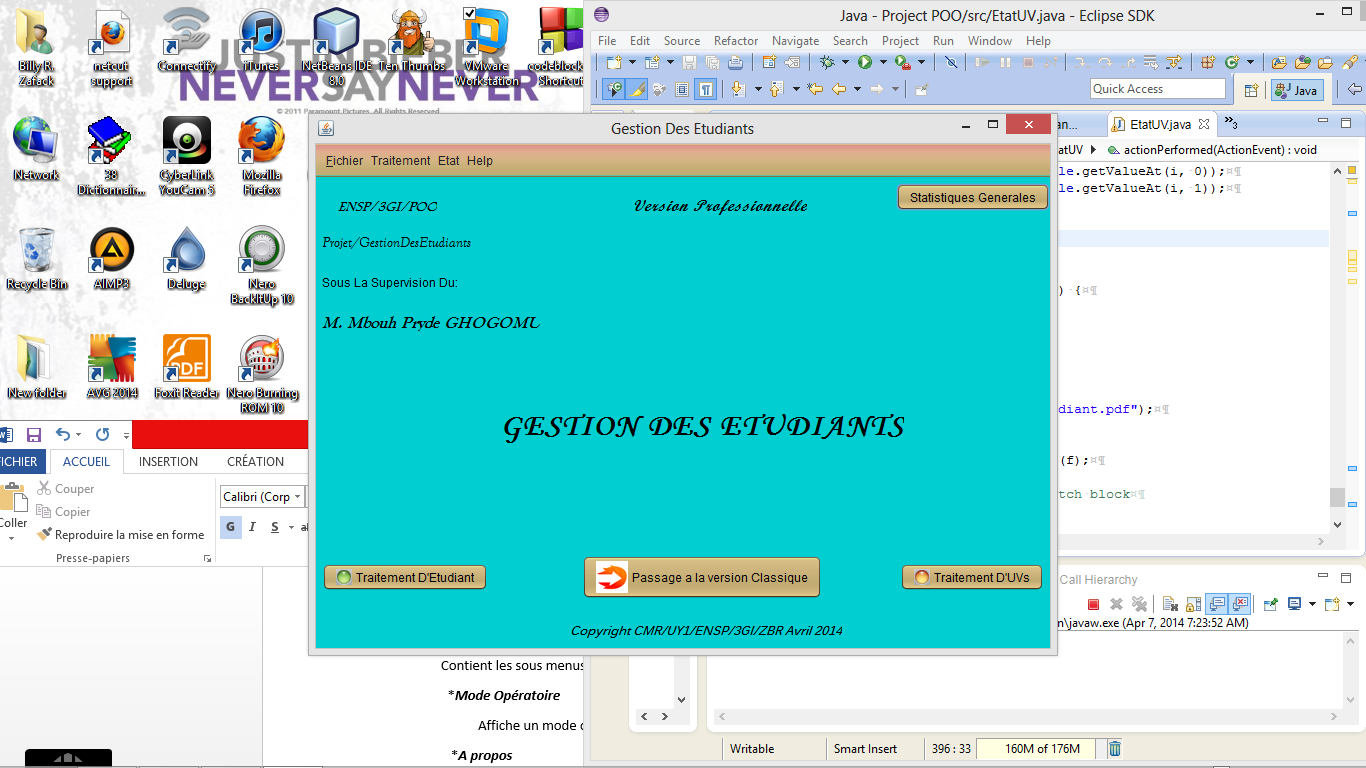
**\* *Quitter*** qui permet de fermer l’application.

-**Traitement**

Elle contient les sous menus suivants :

* ***Etudiant*** qui permet de rechercher un étudiant dont on entrera le matricule.
* **UV** qui permet de modifier les informations sur un étudiant inscrits.

**Fig. Fenêtre Principale**



**Fig Fenetre Principale**

**-Aide**

Contient les sous menus suivants :

\****Mode Opératoire***

Affiche un mode opératoire brève du logiciel.

\****A propos***

Une description brève du projet et son auteur

\****Java DOC***

Donne plusieurs chemins d’accès aux documentations JAVADOC des différentes classes.

\***Aide Imagée**

Donne accès à ce document-ci

\***Aide Html**

Donne accès à une aide html du logiciel

1. **Le Bouton Statistiques Générales :**

Permet d’afficher les statistiques sur les étudiants enregistrées notamment

-Le nombre d’étudiants inscrits

-Le nombre d’UV crée

1. **Bouton « Passage à la version classique »**

Permet de passer à la version classique du projet avec sérialisation de fichier (produit au 1er Semestre).

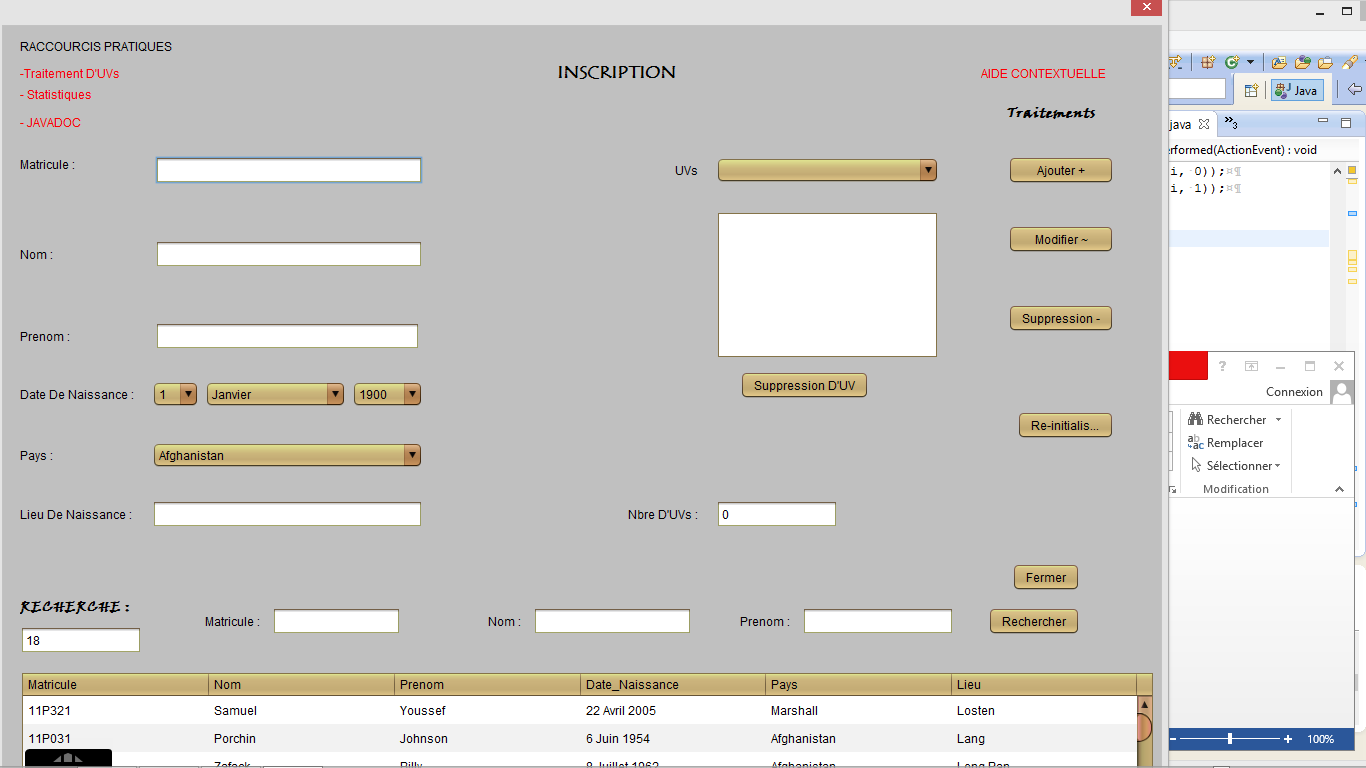
1. **Les Boutons « Traitement d’Etudiants » et « Traitement d’UVs »**

Permettent l’accès aux interfaces de traitements des étudiants et des UVs respectivement.

***Boite De Dialogue***

Il y a 4 fenêtres de dialogues principales

1. **Traitements des Etudiants**



**Fig. Traitement Des Etudiants**

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.

-***Lieu De Naissance***

Un **JTextField** permettant de renseigner le lieu de naissance de l’étudiant.

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

Nous avons notamment :

***-Button Suppression d’UV***

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

***-Button Ajouter***

Qui permet d’ajouter l’étudiant.

***-Button Réinitialiser***

Qui permet d’initialiser la fiche d’inscription.

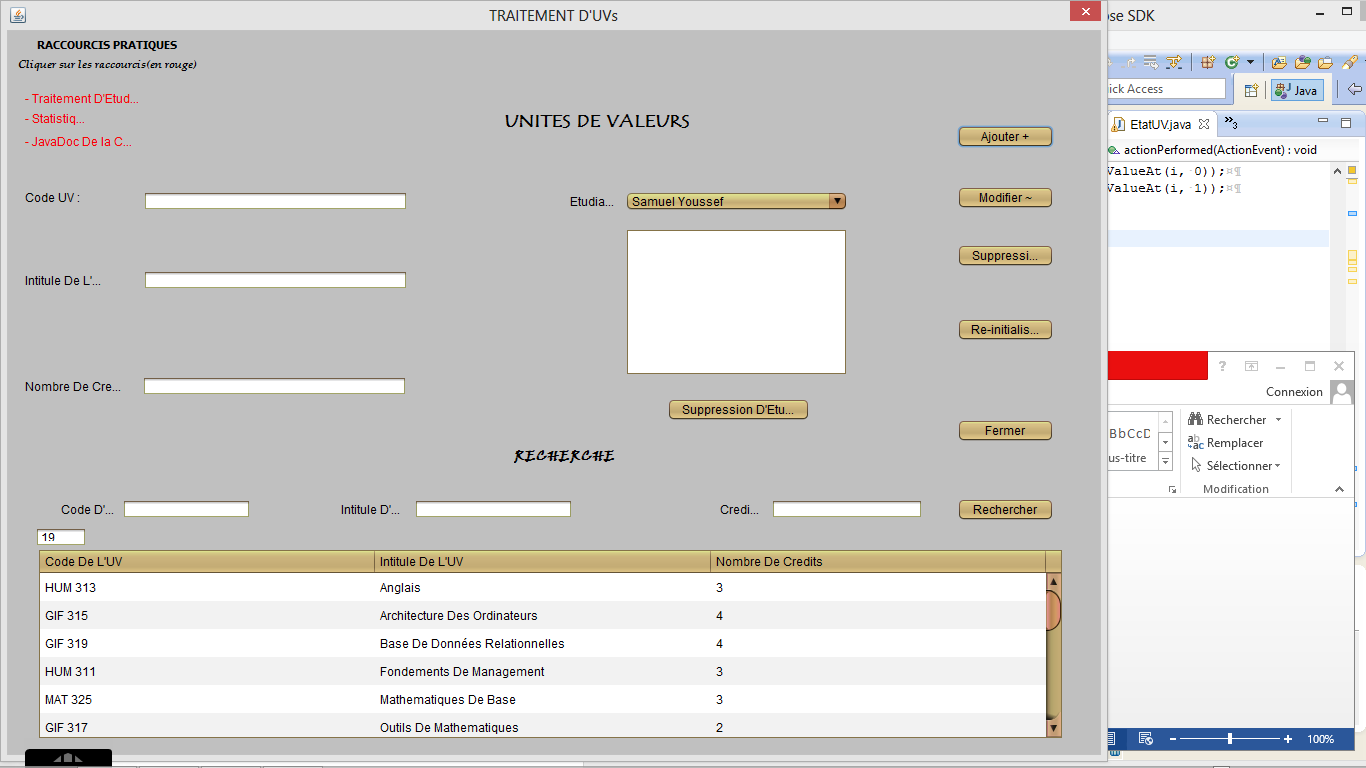
***-Button Fermer***

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. **Tableau**

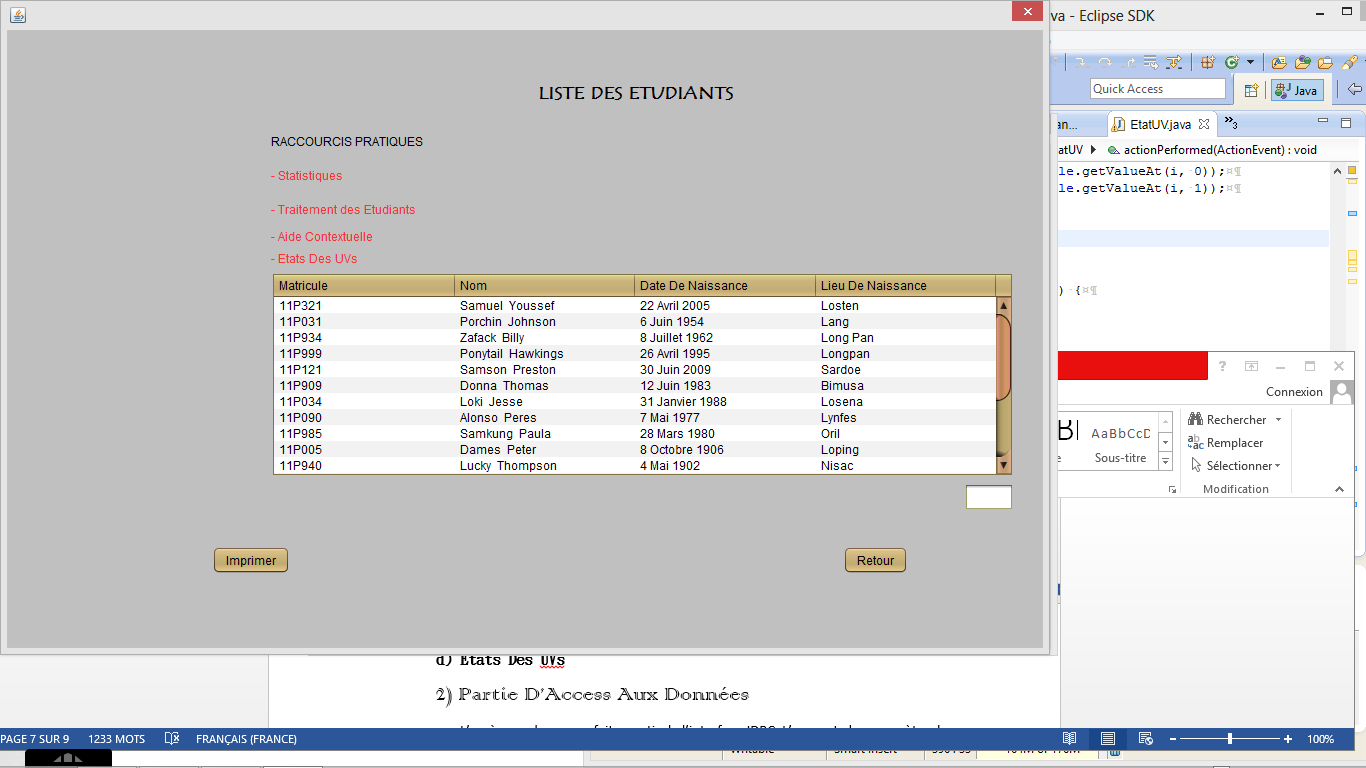
Il y a également un tableau d’étudiant et un système de recherche qui permet de rechercher un étudiant sur la de son matricule, nom ou prénom.

1. **Traitement Des UVs**



**Fig. Interface de traitement des UVs**

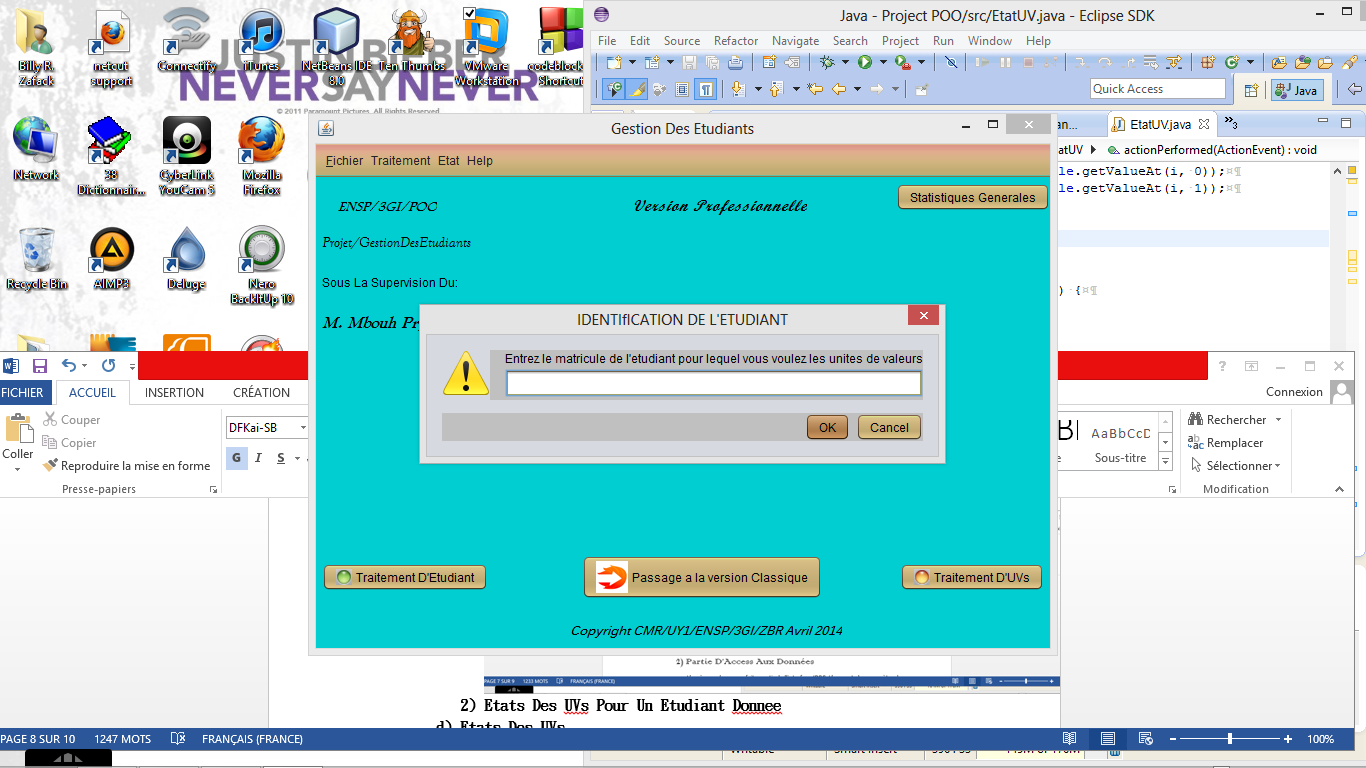
1. **Etats Des Etudiants**
2. **Etats Des Etudiants En Générale**



**Fig Interface Des Etat Des Etudiants(en generale)**

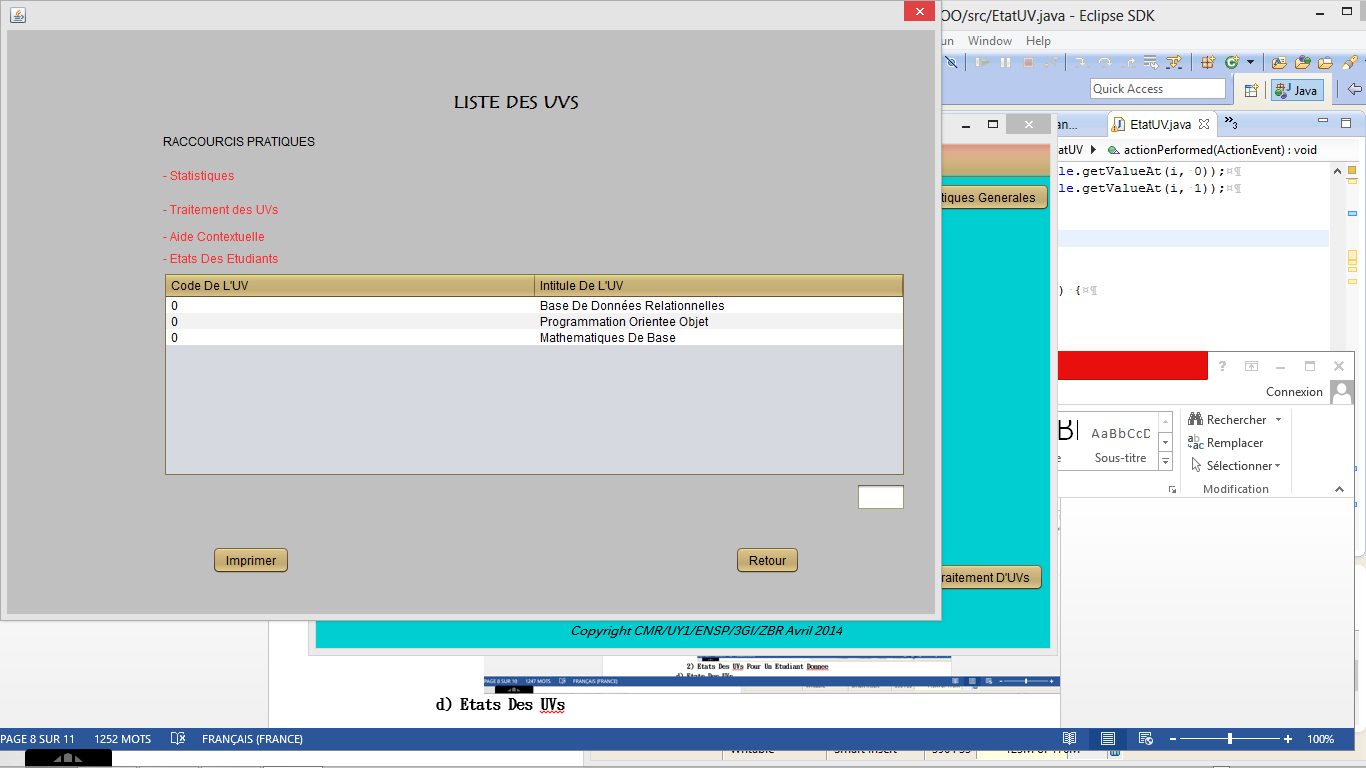
1. **Etats Des UVs Pour Un Etudiant Donné**

**-Entrer Le Matricule De L’Etudiant**



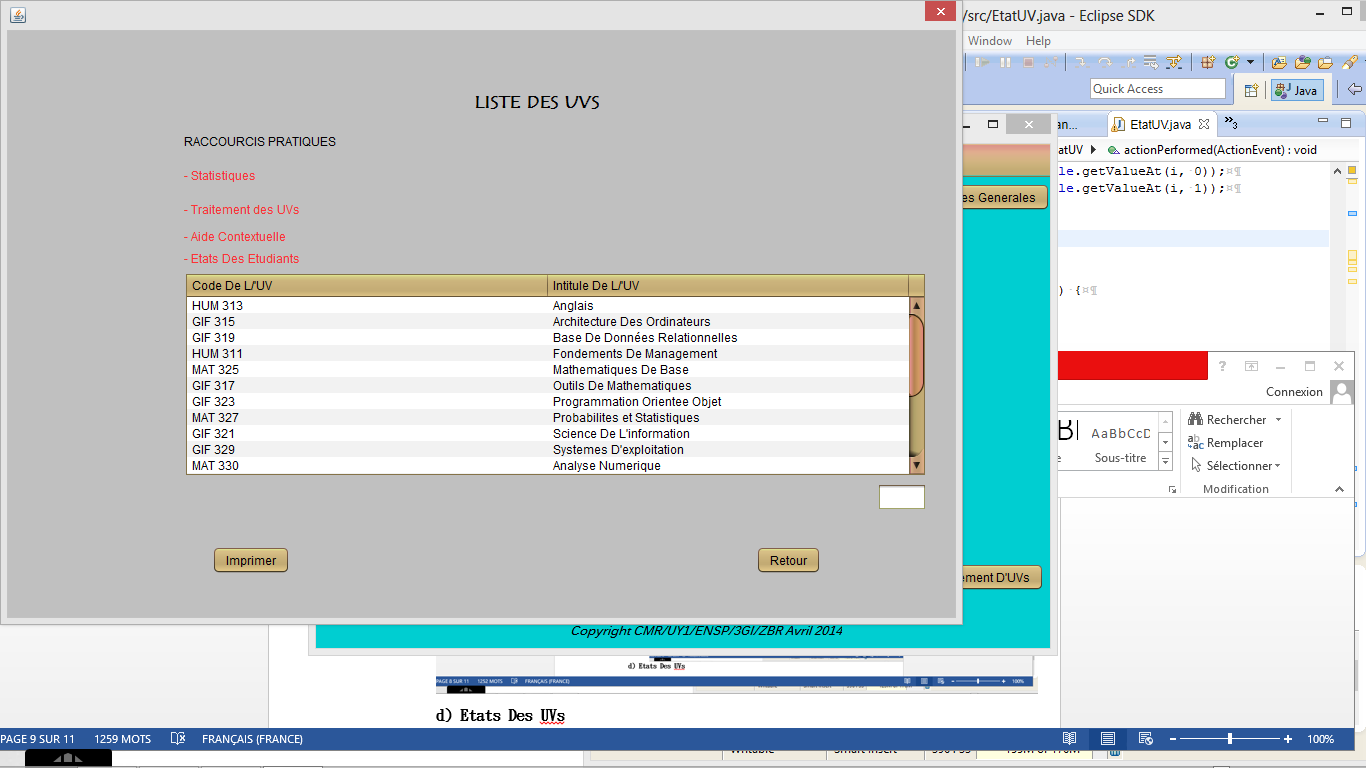
***Fig. Identification de l’étudiant***

**- Accès à l’état de ses UVs**



***Fig. accès a l’état des UVS***

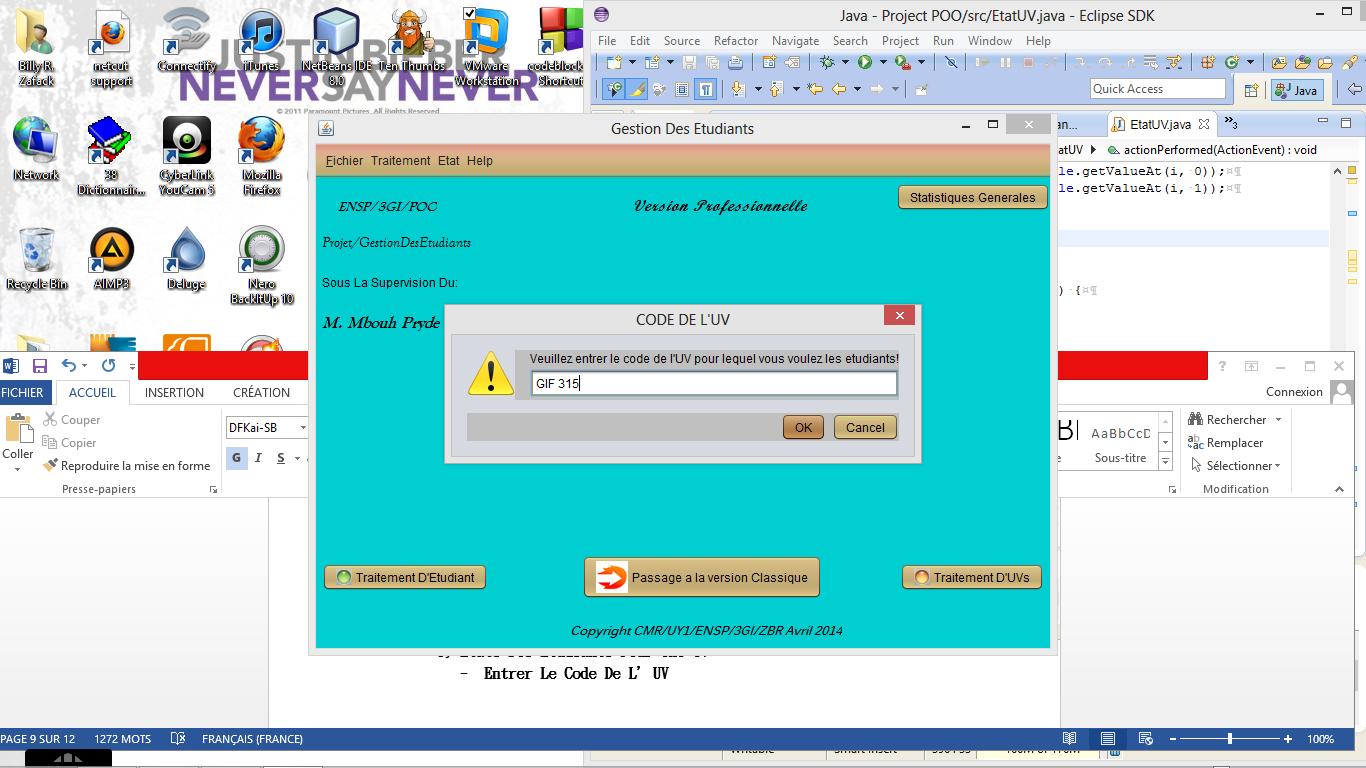
1. **Etats Des UVs**



***Fig. Interface des UVs(en Générale)***

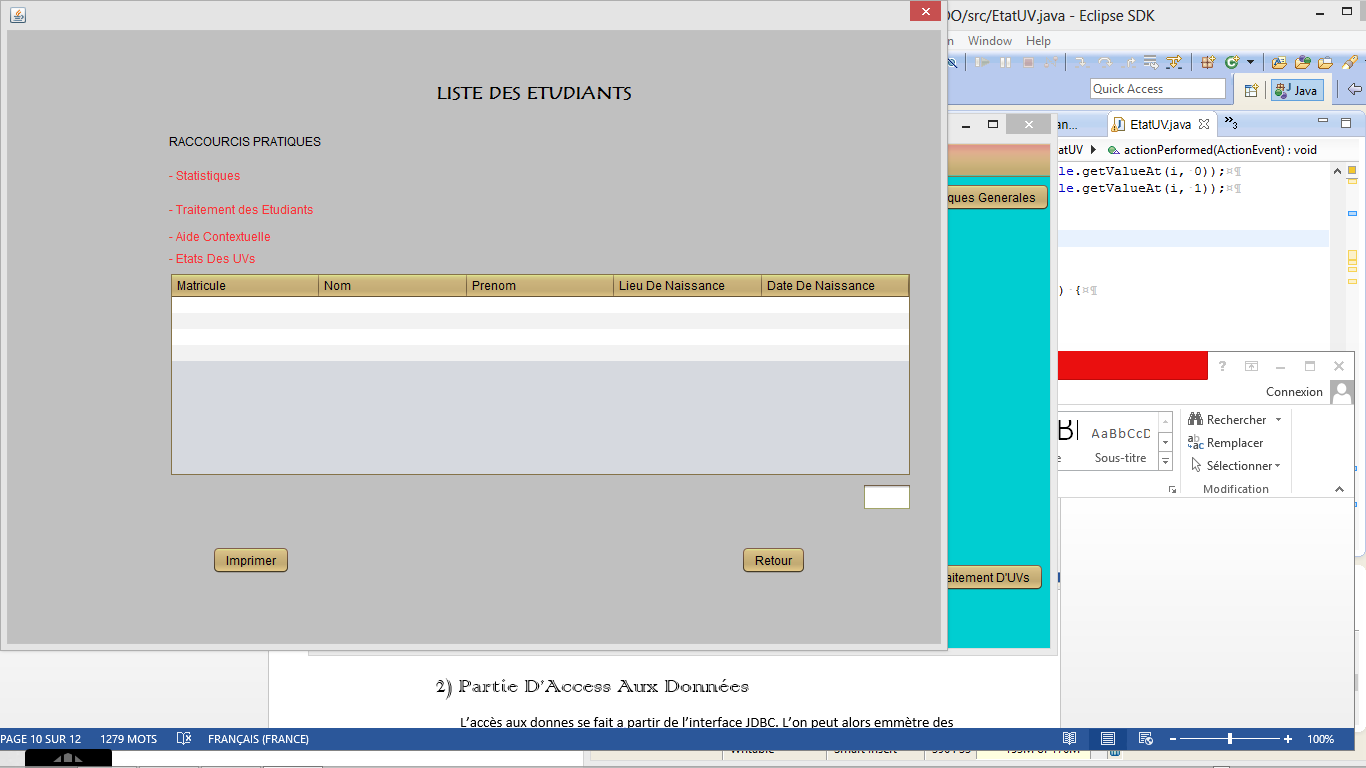
1. **Etats Des Etudiants Pour Une UV**

* **Entrer Le Code De L’UV**



**Fig. Identification De L’UV**

**- Acces a l’etat de ses etudiants**



2) Partie D’Access Aux Données

L’accès aux donnes se fait a partir de l’interface JDBC. L’on peut alors emmètre des requetés de sélection(**Statement.executeQuery**) ou de mise-a-jours(**PreparedStatement.executeUpdate**).

PRESENTATION DU CODE SOURCE

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

-La Classe WindowPrime

-La Classe InscriptionPrimeeee

-La Classe EtudiantPrime

-La Classe Authentification

-La Classe EtatEtudiant

-La Classe EtatUV

-La Classe UVPrimee

1) La Classe WindowPrime

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 InscriptionPrimeeee

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 EtudiantPrime

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 Authentification

Elle permet de construire l’interface d’authentification.

1. La Classe EtatEtudiant

Construit les interfaces sur les états des étudiants.

1. La Classe EtatUV

Construit les interfaces sur les états des UVs.

1. La Classe UVPrimee

Permet de gerer les traitements sur les UVs ;

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.

-Des traitements equivalents sur les UVS.

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.