**Collège Rosemont**

**Sciences Informatiques et Mathématiques**

**Développement de programmes dans un environnement graphique**

***TP3***

***GESTINOTES***

**Professeure** : Mme Niar Wafaa

**Cours** : 420-203-RE

**Fait par** : *Dominguès Patrick et Duchaîne Raphaël*

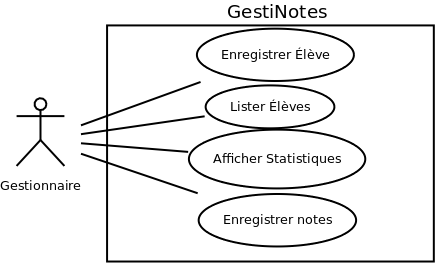
**Session** : Automne 2016

**1-** Consignes :

(Ci-jointes)

**2- ANALYSE :**

* 1. **Diagrammes des cas d’utilisation :**



* 1. **Description narrative des cas d’utilisation**

**Cas d’utilisation1**: Enregistrer Élèves

**Acteurs** : Gestionnaire

**But**: Enregistrer des Élèves

**Cours typique d’événements :**

|  |  |
| --- | --- |
| **Actions Acteur** | **Réponses système** |
| 1-Le **gestionnaire** choisit cette option | 2-Le système lui demande les données de l'élève |
| 3-Il rentre le **nom**, le **prénom** et la **date de naissance** de l'**élève**. | 4-Le système vérifie les informations, affiche le code permanent et le numéro de groupe de l'élève et demande si on désire enregistrer un autre élève. |
| 5-Il quitte ou enregistre un autre élève [retour 1] |  |

**Cours alternatifs :**

Ligne 4:

4- Si données invalide, Le système affiche le message d'erreur adéquat et les redemande

Ligne 4:

4- Si groupe plein, Le système crée un nouveau groupe et mets l’élève dedans

**Cas d’utilisation2**: Lister Élèves

**Acteurs** : Gestionnaire

**But**: Lister les Élèves d'un groupe

**Cours typique d’événements :**

|  |  |
| --- | --- |
| **Actions Acteur** | **Réponses système** |
| 1-Le gestionnaire choisit cette option | 2-Le système demande le groupe |
| 3- Il choisit/écrit le numéro de **groupe**. | 4-Le système affiche les informations des élèves (nom, prenom, date, note finale, code permanent) et demande si l'usager veut **lister** un autre groupe |
| 5-Le gestionnaire quitte ou demande un autre groupe [retour ligne 1] |  |

**Cours alternatifs :**

Ligne 2 :

2 S'il n'y a pas d'élèves enregistrés, le système affiche qu’il n’y a aucun groupe.

**Cas d’utilisation3**: Afficher Statistiques

**Acteurs** : Gestionnaire

**But**: Afficher des Statistiques

**Cours typique d’événements :**

|  |  |
| --- | --- |
| **Actions Acteur** | **Réponses système** |
| 1-Le gestionnaire choisit cette option | 2-Le système affiche des **statistiques (**moyenne, écart-type, variance) |
| 3-Le gestionnaire quitte quand il est satisfait. |  |

**Cours alternatifs :**

Ligne 2 :

2 S'il n'y a pas d'élèves enregistrés, le système affiche qu’il n’y a aucun groupe.

**Cas d’utilisation4**: Enregistrer Notes

**Acteurs** : Gestionnaire

**But**: Enregistrer les Notes d’un Eleve

**Cours typique d’événements :**

|  |  |
| --- | --- |
| **Actions Acteur** | **Réponses système** |
| 1-Le **gestionnaire** choisit cette option | 2-Le système lui propose des élèves |
| 3-Il rentre selectionne l’élève | 4-Le système affiche les données de l’élève et laisse le Gestionnaire les modifier. |
| 5-Il rentre le(s) note(s) à modifier/enregistrer pour cet **élève**. | 6-Le système vérifie les informations, affiche les notes de l'élève et demande si on désire enregistrer d’autres notes. |
| 7-Il quitte ou enregistre d’autres notes [retour ligne 1] |  |

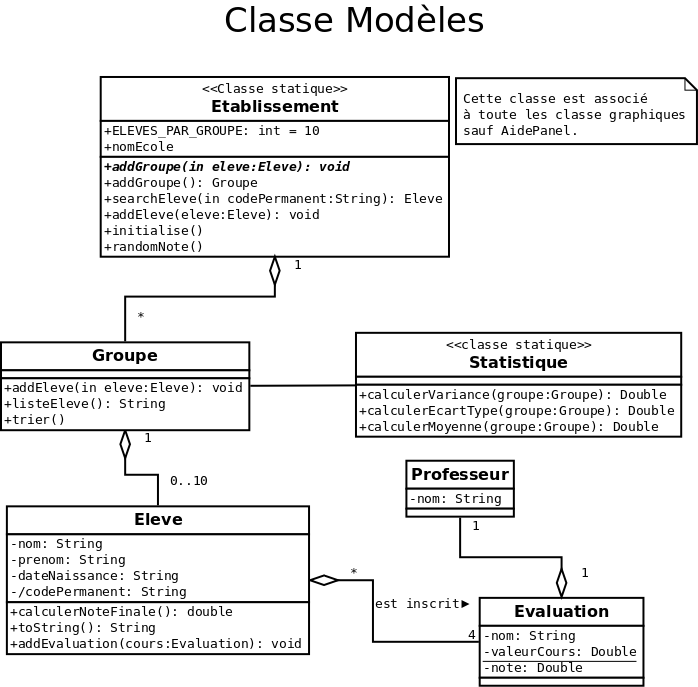
**Cours alternatifs :**

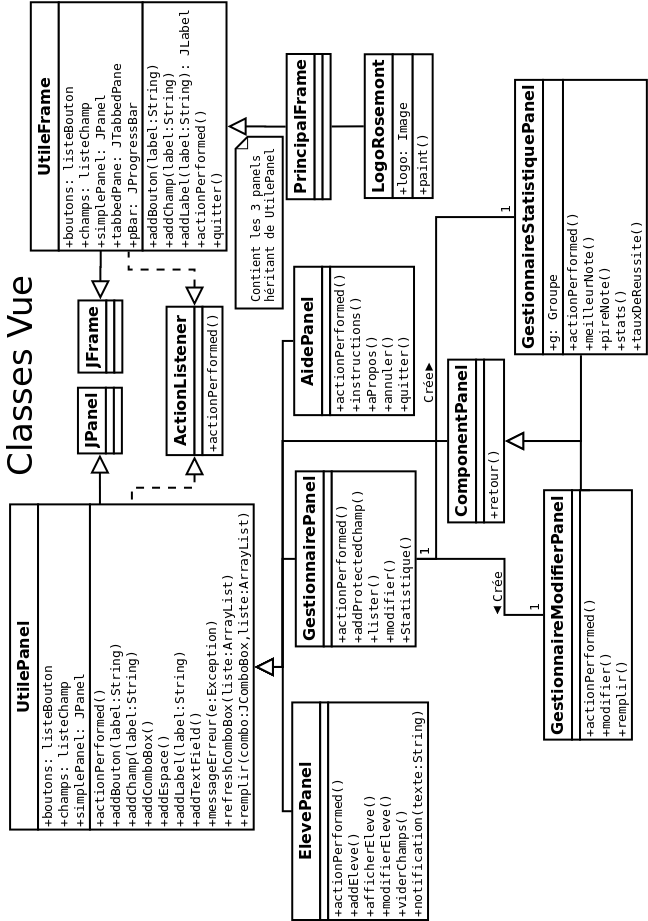
Ligne 2 :

2 S'il n'y a pas d'élèves enregistrés, le système affiche qu’il n’y a aucun élève.

Ligne 4:

4- Si donnée invalide, le Système affiche le message d'erreur adéquat et la redemande





**Grille d’évaluation :** **Code source** - 80 %

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Noms :** | **Très satisfaisant** | **Satisfaisant** | **Assez satisfaisant** | **Peu satisfaisant** | **Insatisfaisant** | |
| Correction du code :   * Absence de bugs * Résultats justes * Validations appropriés | **20** | **15** | **10** | **5** | **0** | |
| Respect des concepts orientés objet :   * Modularité (découpage classes) * Encapsulation * Héritage et polymorphisme | **20** | **15** | **10** | **5** | **0** | |
| Convivialité des interfaces graphiques :   * Choix judicieux des interfaces graphiques * Application des bonnes pratiques | **20** | **15** | **10** | **5** | **0** | |
| Application des normes de programmation :   * Commentaires * Indentations * Nomenclature * Organisation MVC | **15** | **12** | **8** | **4** | **0** | |
| Qualité de la langue   * dans les interfaces et les messages * dans les commentaires | **5** | **4** | **3** | **2** | **0** | |
| **Total** | **………………./ 80** | | | | |  |

**Grille d’évaluation :**  **Rapport** – 20%

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Très satisfaisant** | **Satisfaisant** | **Assez satisfaisant** | **Peu satisfaisant** | **Insatisfaisant** | |
| Organisation et structure du rapport | **5** | **4** | **3** | **2** | **0** | |
| Description précise des Cas d’utilisation | **5** | **4** | **3** | **2** | **0** | |
| Description précise des Classes | **5** | **4** | **3** | **2** | **0** | |
| Qualité de la langue | **5** | **4** | **3** | **2** | **0** | |
| **Total** | **………………. / 20** | | | | |  |

**Grand total :** :

**Remarques : code source**

|  |  |
| --- | --- |
| **Remarques** |  |
| Correction du code :   * Absence de buggs * Résultats justes * Validations appropriés |  |
| Respect des concepts orientés objet :   * Modularité (découpage classes) * Encapsulation * Héritage et polymorphisme |  |
| Convivialité des interfaces graphiques :   * Choix judicieux des interfaces graphiques * Application des bonnes pratiques |  |
| Application des normes de programmation :   * Commentaires * Indentations * Nomenclature * Organisation MVC |  |
| Qualité de la langue   * dans les interfaces et les messages * dans les commentaires |  |
| **AUTRES** |  |

**Grille Remarques :**  **Rapport**

|  |  |
| --- | --- |
| **Remarques** |  |
| Organisation et structure du rapport |  |
| Description précise des Cas d’utilisation |  |
| Description précise des Classes |  |
| Qualité de la langue |  |
| **AUTRES** |  |