

**Collège Rosemont**  
**Sciences Informatiques et Mathématiques**

**Développement de programmes dans un environnement graphique**  
***TP3***

***GESTINOTES***

**Professeure** : Mme Niar Wafaa

**Cours** : 420-203-RE

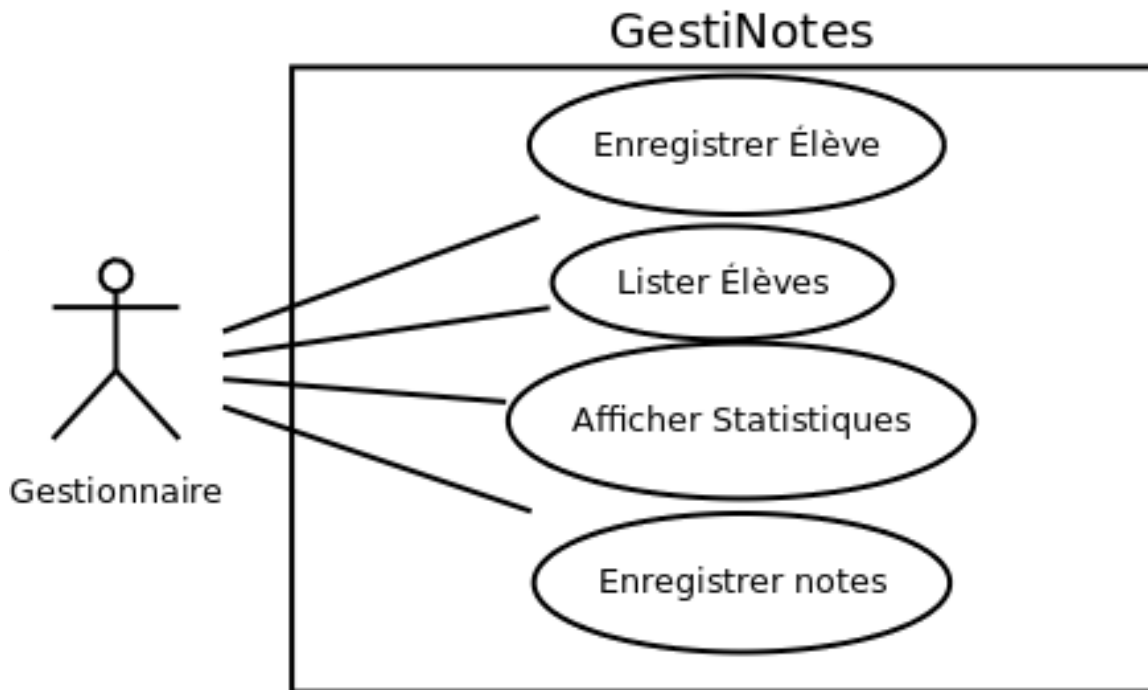
**Fait par** : *Dominguès Patrick et Duchaîne Raphaël*

**Session** : Automne 2016

1- Consignes :  
(Ci-jointes)

## 2- ANALYSE :

### 2.1. Diagrammes des cas d'utilisation :



Actions Acteur	Réponses système
1-Le <b>gestionnaire</b> choisit cette option	2-Le système lui demande les données de l'élève
3-Il rentre le <b>nom</b> , le <b>prénom</b> et la <b>date de naissance</b> 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 <b>groupe</b> .	4-Le système affiche les informations des élèves (nom, prénom, date, note finale, code permanent) et demande si l'utilisateur veut <b>lister</b> 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 <b>statistiques</b> (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 <b>gestionnaire</b> 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 <b>élève</b> .	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 :**

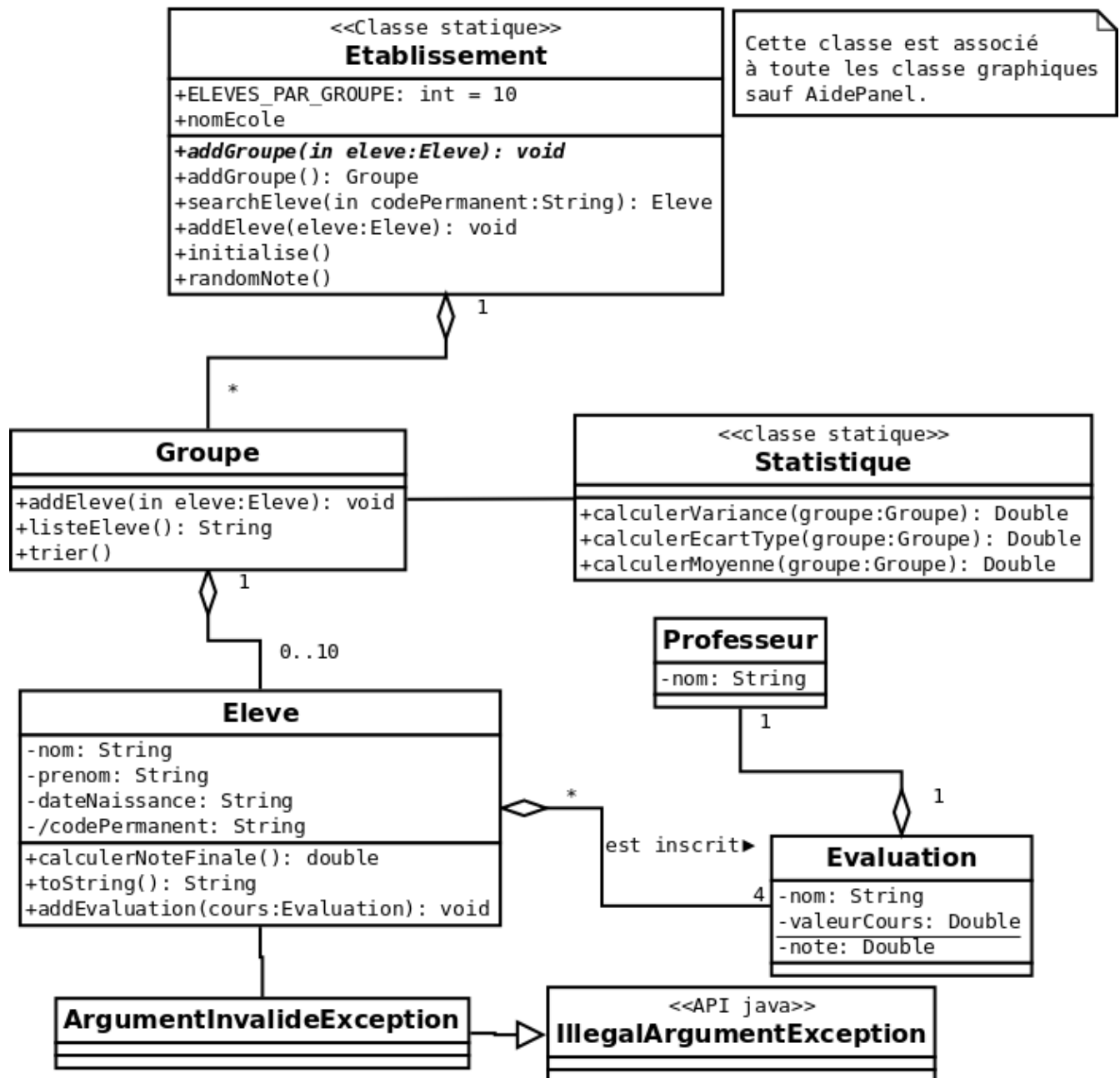
Ligne 2 :

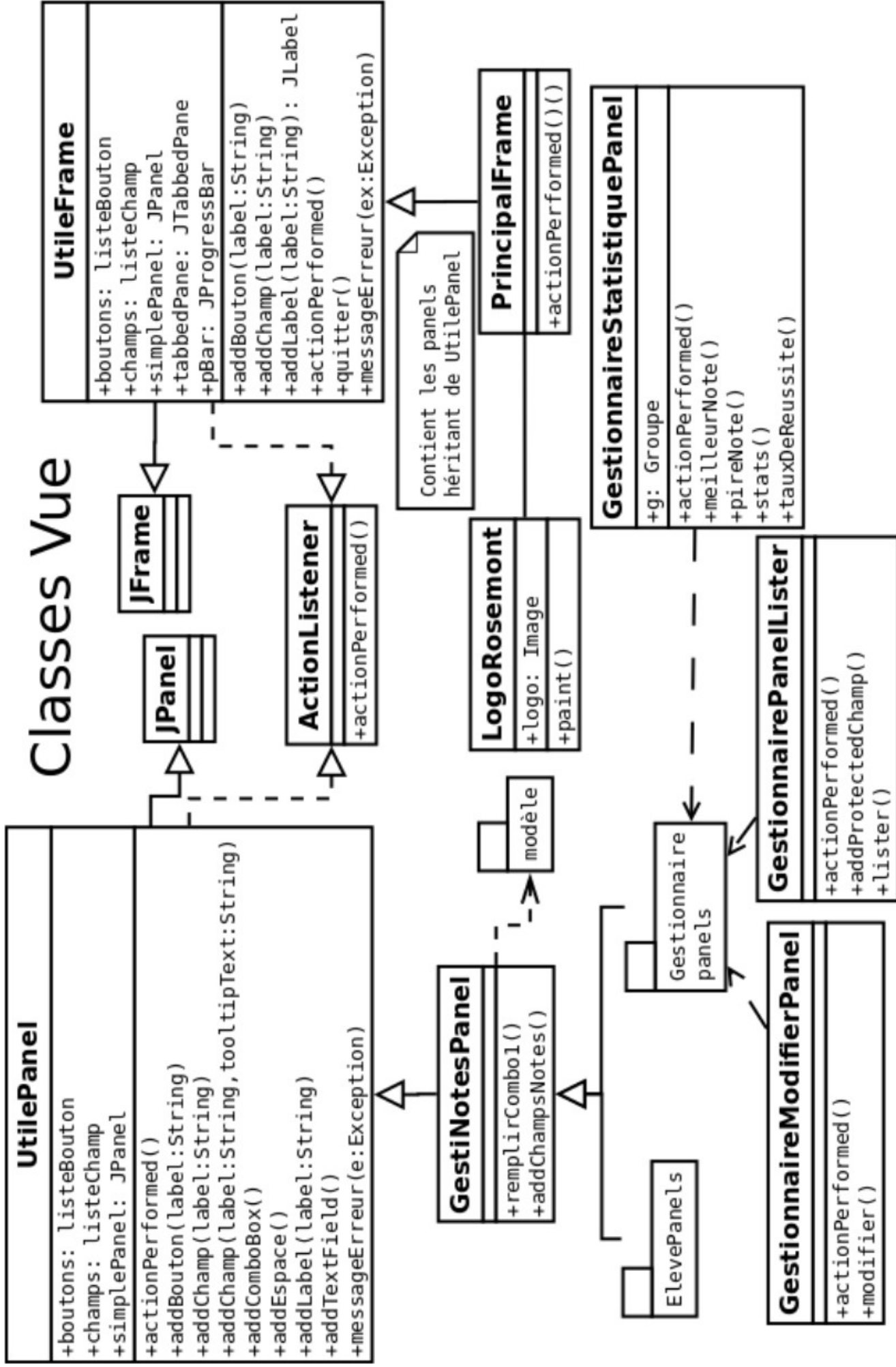
2 S'il n'y a pas d'élèves enregistrés, le système affiche qu'il n'y a aucun élève.

Ligne 4:

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

# Classe Modèles





**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
<b>Total</b>	...../ 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
<b>Total</b>	..... / 20				

**Grand total : :**

### Remarques : code source

<b>Remarques</b>	
Correction du code :  - Absence de bugs - 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	
<b>AUTRES</b>	

### Grille Remarques : Rapport

<b>Remarques</b>	
Organisation et structure du rapport	
Description précise des Cas d'utilisation	
Description précise des Classes	
Qualité de la langue	
<b>AUTRES</b>	

## **Autocritique**

Nous avons apprécié plusieurs choses dans ce projet. Par exemple, la rédaction des cas d'utilisations et des diagrammes de classe, quoique parfois longue à faire, a été très utile pour implémenter le projet. En effet, avoir un bon plan initial et une idée claire des ont permis non seulement d'accélérer l'écriture du code, mais également d'augmenter la clarté de la tâche à faire. Faire un plan sera très utile pour le projet intégrateur, car il permettra à tous les membres de l'équipe d'avoir la même vision de la tâche à accomplir. De plus, l'utilisation de GitHub lors du projet nous a permis de bien travailler en équipe, puisqu'il n'y avait pas de confusion lorsqu'on essayait de modifier du code fait par l'autre coéquipier. Également, nous avons découvert une technique de programmation nommée test-driven development (TDD), qui consiste à faire les tests unitaires en parallèle et JUSTE AVANT de commencer à écrire le code source. Ainsi, au fur et à mesure, on peut repérer les erreurs, ce qui est notre faiblesse, et les corriger avant de faire une remise. ce qui évite leurs accumulations à la fin de l'écriture du code source. Cette méthode nous sera très utile lors du projet intégrateur, pour ainsi corriger les erreurs beaucoup plus facilement. Cependant, certaines choses ont été faites, qui ont malheureusement nuis au progrès du projet. Entre autres, lors de la 2ème partie du projet, nous avons eu beaucoup de problèmes lorsque nous avons essayé d'utiliser un JTabbedPane à la place de JMenuBar, et cela a causer une dégradation considérable de la qualité du travail, ainsi qu'une augmentation du temps requis terminer le TP2 (au moins une dizaine d'heure de travail de plus pour un résultat équivalent aux autre équipes). Par ailleurs, une meilleure communication au sein de l'équipe aurait été nécessaire. En effet, il y avait parfois des délais de réponse trop grands entre les membres de l'équipe, parfois dû à des absences, ou à un système de communication inefficace. C'est pour cela que dorénavant, nous allons utiliser beaucoup plus Discord, logiciel assez similaire sur le fonctionnement que Skype, pour pouvoir communiquer sans délai.



**GRILLE D'ÉVALUATION DES INTERFACES GRAPHIQUES**

<b>Thème</b>	<b>Règles</b>	<b>note</b>
<b><u>ORGANISATION VISUELLE</u></b>		<b>/5</b>
<b><u>Dialogue</u></b>	<ol style="list-style-type: none"> <li>1. Présenter les commandes et les données dans l'ordre de leur utilisation</li> <li>2. Regrouper les informations relatives à une même activité sur la même fenêtre</li> <li>3. Laisser l'initiative du dialogue à l'utilisateur</li> <li>4. Guider l'utilisateur pour faciliter la navigation</li> </ol>	
	<b>Note</b>	<b>4</b>
<b><u>Agencement Écran</u></b>	<ol style="list-style-type: none"> <li>1. Organiser les zones de manipulation dans le sens la lecture: Gauche-&gt; Droite et Haut -&gt; Bas et en fonction de la fréquence d'utilisation</li> <li>2. Les zones de manipulation doivent être les plus compactes possibles</li> <li>3. Regrouper les informations en relation entre elles</li> <li>4. Rassembler dans la même fenêtre les données nécessaires à l'accomplissement de la même tâche (ou sous-tâche).</li> </ol>	
	<b>Note</b>	<b>5</b>
<b><u>GRAPHISME</u></b>		
<b><u>Codage couleurs</u></b>	<ol style="list-style-type: none"> <li>1. Associer chaque couleur à un sens précis</li> <li>2. Respecter sens couleurs de l'utilisateur</li> <li>3. Minimiser le nombre de couleurs: 7+-2</li> <li>4. Couleurs doivent être identifiables facilement</li> </ol>	
	<b>Note</b>	<b>3</b>
<b><u>Choix couleurs</u></b>	<ol style="list-style-type: none"> <li>1. Bleu recommandé pour les encadrements</li> <li>2. Choisir couleur faiblement saturée (pâle) pour afficher l'information</li> <li>3. Choisir couleur très peu saturée pour information de moindre importance (Gris)</li> <li>4. Utiliser mélange de jaune-vert pour plus d'effet sensibilité</li> <li>5. Utiliser couleurs peu contrastées pour similarité</li> <li>6. Utiliser couleurs très contrastées pour différences</li> </ol>	
	<b>Note</b>	<b>3</b>
<b><u>Couleurs de fond</u></b>	<ol style="list-style-type: none"> <li>1. Utiliser une couleur neutre et claire pour fond</li> <li>2. Éviter le fond gris ou coloré pour petits objets</li> <li>3. Éviter les fonds marrons et verts</li> <li>4. Le noir devrait faire ressortir les petits objets (risque de fatigue)</li> <li>5. Utiliser les couleurs peu saturées (Pastels) pour les grandes surfaces,</li> </ol>	
	<b>Note</b>	<b>3</b>
<b><u> Icônes</u></b>	<ol style="list-style-type: none"> <li>1. Utiliser icônes pour objets et commandes fréquemment employés</li> <li>2. Lien entre icône et référent doit être le plus direct possible</li> <li>3. Toujours accompagner icône par nom</li> <li>4. Limiter nombre icône : min 12 - max 20</li> </ol>	

## AUTO-ÉVALUATION - ERGONOMIE INTERFACES GRAPHIQUES

	5. S'assurer que l'utilisateur distingue bien les différentes icônes	
	<b>Note</b>	<b>2</b>
<b>Mises en évidence</b>	<ol style="list-style-type: none"> <li>1. Utiliser exceptionnellement pour efficacité</li> <li>2. Limiter à cinq moyens différents</li> <li>3. Éviter plusieurs moyens sur même objet:</li> <li>4. Ne pas entraver perception utilisateur</li> </ol>	
	<b>Note</b>	<b>5</b>
<b>Moyens de mise en évidence</b>	<ol style="list-style-type: none"> <li>1. Couleur: c'est le changement de couleur qui attire: saturés</li> <li>2. Souligné: À éviter pour texte long</li> <li>3. Proximité: Utiliser pour ressortir informations connexes</li> <li>4. Gras: Utiliser pour ressortir chaîne dans texte</li> <li>5. Taille: À utiliser pour comparaisons relatives</li> <li>6. Police: À utiliser pour ressortir longs textes</li> <li>7. Clignotement: Est le moyen le plus efficace mais Offrir possibilité d'interrompre</li> </ol>	
	<b>Note</b>	<b>3</b>
<b><u>LANGAGE</u></b>		
<b>Libellé des commandes</b>	<ol style="list-style-type: none"> <li>1. Utiliser un seul et unique libellé par commande</li> <li>2. Utiliser une syntaxe homogène</li> <li>3. Éviter les formes nominales</li> <li>4. Éviter les abréviations</li> </ol>	
	<b>Note</b>	<b>4</b>
<b>Messages affichés</b>	<ol style="list-style-type: none"> <li>1. Utiliser un langage familier à l'utilisateur</li> <li>2. Éviter les messages longs lorsque la charge de travail est importante</li> <li>3. Produire des messages clairs, concis et adaptés au niveau de connaissance de l'utilisateur</li> <li>4. Adopter un vocabulaire homogène</li> <li>5. Construire des messages affirmatifs et Employer la forme active</li> <li>6. Employer une fenêtre de message lorsque le message doit être lu</li> </ol>	
	<b>Note</b>	<b>4</b>
<b><u>FENÊTRES</u></b>		
<b>Fenêtres</b>	<ol style="list-style-type: none"> <li>1. Faciliter activation et ouverture</li> <li>2. Agencer fenêtre pour faciliter mémorisation</li> <li>3. Augmenter nombre fenêtres si utilisation peu fréquente / Diminuer si fréquente</li> <li>4. Minimiser quantité informations à mémoriser d'une fenêtre à l'autre</li> </ol>	
	<b>Note</b>	<b>4</b>
<b>Fenêtres de dialogue</b>	<ol style="list-style-type: none"> <li>1. Présenter composants dans ordre utilisation</li> <li>2. Minimiser déplacements souris</li> <li>3. Faciliter accès composants fréquemment utilisés</li> <li>4. Mettre en évidence éléments importants</li> <li>5. Utiliser toujours titres courts et non ambigus</li> </ol>	
	<b>Note</b>	<b>4</b>
<b><u>Composant fenêtres</u></b>		
<b>Boutons</b>	<ol style="list-style-type: none"> <li>1. À employer pour commandes fréquemment utilisées</li> <li>2. Utiliser libellé explicite et non générique</li> </ol>	
	<b>Note</b>	<b>5</b>
<b>Champs de</b>	<ol style="list-style-type: none"> <li>1. Choisir des libellés les plus courts possibles</li> </ol>	

## AUTO-ÉVALUATION - ERGONOMIE INTERFACES GRAPHIQUES

<b>saisie</b>	<ol style="list-style-type: none"> <li>2. Proposer par défaut la valeur la plus courante</li> <li>3. Préciser le format de saisie lorsque possible</li> <li>4. Indiquer les champs facultatifs / obligatoires</li> <li>5. Aligner les champs de manière à minimiser la charge de travail.</li> </ol>	
	<b>Note</b>	<b>3</b>
<b>Composants de choix</b>	<ol style="list-style-type: none"> <li>1. Utiliser boutons sélection pour choix fréquents et peu nombreux</li> <li>2. Utiliser liste déroulante lorsque place réduite et choix peu fréquents</li> <li>3. Utiliser liste simple lorsque le nombre de choix est variable</li> <li>4. Présenter toujours verticalement les composants de sélection</li> </ol>	
	<b>Note</b>	<b>5</b>
<b>MODES DE DIALOGUE (INTERACTIVITÉ)</b>		
<b>Par boutons</b>	<ol style="list-style-type: none"> <li>1. Minimiser leur nombre dans la fenêtre</li> <li>2. Utiliser pour les commandes très fréquentes</li> <li>3. Bien choisir leur libellé et leur position</li> <li>4. Rester cohérent dans libellé et position</li> </ol>	
	<b>Note</b>	<b>5</b>
<b>Par menus</b>	<ol style="list-style-type: none"> <li>1. Ajuster la structure du menu selon l'activité de l'utilisateur</li> <li>2. Minimiser la taille des menus</li> <li>3. Organiser les menus selon leur utilisation</li> <li>4. Isoler les commandes à risque</li> <li>5. Faire correspondre le libellé des commandes avec le titre de la fenêtre actionnée</li> <li>6. Permettre l'accès rapide et direct aux commandes fréquentes</li> </ol>	
	<b>Note</b>	<b>4.5</b>
<b>Par Souris</b>	<ol style="list-style-type: none"> <li>1. Maintenir un rôle constant pour les boutons</li> <li>2. Ne pas déclencher les fonctions importantes ou risquées par simple clic</li> <li>3. Utiliser le survol souris (ou balayage) pour guider l'utilisateur</li> </ol>	
	<b>Note</b>	<b>5</b>
<b>PRÉVENTION ERREURS</b>		
<b>Erreurs</b>	<ol style="list-style-type: none"> <li>1. Guider utilisateur</li> <li>2. Fournir visibilité fonctionnement système</li> <li>3. Fournir retour aux actions utilisateur</li> <li>4. Rendre possible exploration logiciel</li> <li>5. Fournir aide en ligne adaptée à la tâche</li> </ol>	<b>5</b>
	<b>Note</b>	
<b>Prévention</b>	<ol style="list-style-type: none"> <li>1. Signaler message confirmation si risque de perte données ou si commande irréversible</li> <li>2. Tester pour identifier erreurs</li> <li>3. Éviter de placer côte à côte les commandes à risque avec les autres</li> </ol>	
	<b>Note</b>	<b>5</b>
<b>CORRECTION/SIGNALEMENT ERREURS</b>		
<b>Correction erreurs</b>	<ol style="list-style-type: none"> <li>1. Signaler les erreurs immédiatement</li> <li>2. Placer message dans la direction du regard utilisateur</li> <li>3. Offrir possibilité de modifier facilement les données suite à erreur</li> </ol>	

## AUTO-ÉVALUATION - ERGONOMIE INTERFACES GRAPHIQUES

	<ul style="list-style-type: none"> <li>4. Permettre le retour arrière</li> <li>5. Permettre en tout temps l'accès à l'aide en ligne</li> </ul>	
	<b>Note</b>	<b>4</b>
<b>Signalement erreurs</b>	<ul style="list-style-type: none"> <li>1. Le message doit préciser la nature du problème et offrir des moyens pour y remédier</li> <li>2. Le message doit être coopératif</li> <li>3. Le message doit fournir une description du problème</li> <li>4. Adapter le message au niveau de connaissances de l'utilisateur</li> <li>5. Ne pas dramatiser ni culpabiliser l'utilisateur</li> <li>6. Préférer une syntaxe homogène dans le message même</li> </ul>	
	<b>Note</b>	<b>5</b>
<b>AIDE EN LIGNE</b>		
<b>Aide en ligne</b>	<ul style="list-style-type: none"> <li>1. Fournir un accès rapide à l'aide en ligne</li> <li>2. Décrire l'utilisation du système en s'appuyant sur le vocabulaire du métier de l'utilisateur</li> <li>3. Permettre l'exploration</li> <li>4. Utiliser des titres explicites</li> <li>5. Illustrer les descriptions avec des exemples</li> <li>6. Rédiger clairement</li> </ul>	
	<b>Note</b>	<b>5</b>
<b>Rédaction aide en ligne</b>	<ul style="list-style-type: none"> <li>1. Phrases courtes et simples</li> <li>2. Pas de subordonnées ni relatives</li> <li>3. Forme active</li> <li>4. Vocabulaire familier</li> <li>5. Paragraphes courts</li> <li>6. Paragraphes séparés par ligne blanche</li> <li>7. Lignes de plus de 26 caractères</li> <li>8. Pas de coupure de mot</li> <li>9. Ne pas justifier à droite</li> <li>10. Lignes espacées de la hauteur d'un caractère : double interligne</li> </ul>	
	<b>Note</b>	<b>5</b>
	<b>GRAND TOTAL (/120)</b>	<b>100.5</b>

## ANOMALIES ET RÉOLUTIONS

Décrire dans les tableaux suivants au minimum 4 anomalies identifiées dans les interfaces graphiques de l'application, à partir de la grille d'évaluation ci-dessus.

<b>Description du problème</b>	<b>L'initialisation est dans le même menu que lister ou statistique, ce qui fait qu'un utilisateur peu attentif pourrait écraser ses informations par accident.</b>
<b>Fenêtre de l'anomalie</b>	<b>menu gestionnaire</b>
<b>Règle ou critère enfreint</b>	MODES DE DIALOGUE (INTERACTIVITÉ) par bouton: Isoler les commandes à risque
<b>Résolution de l'anomalie</b>	<b>Séparer le menu Gestionnaire en deux menus : Groupe et Données, le premier comportant lister, modifier et statistique alors que le deuxième contient tout ce qui a trait au traitement de fichiers.</b>

<b>Description du problème</b>	<b>Le manque d'icônes peut faire perdre l'utilisateur, dû à un manque de repères.</b>
<b>Fenêtre de l'anomalie</b>	<b>Toutes les fenêtres.</b>
<b>Règle ou critère enfreint</b>	GRAPHISME Icônes : <ol style="list-style-type: none"> <li>1. Utiliser icônes pour objets et commandes fréquemment employés</li> <li>2. Lien entre icône et référent doit être le plus direct possible</li> <li>3. Toujours accompagner icône par nom</li> <li>4. Limiter nombre icône : min 12 - max 20</li> </ol> S'assurer que l'utilisateur distingue bien les différentes icônes
<b>Résolution de l'anomalie</b>	<b>Ajouter un icône unique pour toutes les fenêtres des menus Eleve, Gestionnaire et Aide. Ainsi, l'utilisateur pourra savoir vers où il se dirige grâce aux icônes.</b>

## AUTO-ÉVALUATION - ERGONOMIE INTERFACES GRAPHIQUES

<b>Description du problème</b>	<b>Le manque de couleurs variées peut non seulement enlever des repères à l'utilisateur, mais il peut également nuire à l'expérience visuelle de l'utilisateur.</b>
<b>Fenêtre de l'anomalie</b>	<b>Toutes les fenêtres.</b>
<b>Règle ou critère enfreint</b>	<p style="text-align: center;">GRAPHISME</p> <p>Choix des couleurs :</p> <ol style="list-style-type: none"> <li>1. Bleu recommandé pour les encadrements</li> <li>2. Choisir couleur faiblement saturée (pâle) pour afficher l'information</li> <li>3. Choisir couleur très peu saturée pour information de moindre importance (Gris)</li> <li>4. Utiliser mélange de jaune-vert pour plus d'effet sensibilité</li> <li>5. Utiliser couleurs peu contrastées pour similarité</li> </ol> <p>Utiliser couleurs très contrastées pour différences</p>
<b>Résolution de l'anomalie</b>	<b>Associer des couleurs à chaque menu (Eleve, Gestionnaire, Aide), ainsi qu'ajouter une couleur de fond, pour améliorer le guidage de l'utilisateur, et pour améliorer l'aspect visuel du programme en général</b>

<b>Description du problème</b>	<b>Manque de guidage</b>
<b>Fenêtre de l'anomalie</b>	<b>Fenêtre principale</b>
<b>Règle ou critère enfreint</b>	<p><b>Guidage :</b> ORGANISATION VISUELLE Dialogue</p> <p>1. Présenter les commandes et les données dans l'ordre de leur utilisation</p>
<b>Résolution de l'anomalie</b>	<p><b>Penser d'abord à l'utilisateur :</b>  <b>-À ce qu'il veut accomplir avec l'application</b>  <b>-À ce qu'il s'attend comme déroulement avant de faire toute l'interface.</b>  <b>Ce serait sûrement plus facile à corriger avec JavaFx (pour tester différentes organisations rapidement).</b></p>

## AUTO-ÉVALUATION - ERGONOMIE INTERFACES GRAPHIQUES

*Ce document complété est à placer en annexe à la fin de votre rapport*

### Grille d'évaluation

Noms :	Très satisfaisant	Satisfaisant	Assez satisfaisant	Peu satisfaisant	Insatisfaisant
- Correction du TP3 - Rapport	10	8	5	2	0
- Correction du TP3 - Codage	10	8	5	2	0
- Packages	10	8	5	2	0
- Javadoc	10	8	5	2	0
- Fichier Jar	10	8	5	2	0
- Rédaction aide en ligne	10	8	5	2	0
- Autoévaluation ergonomie	10	8	5	2	0
- Améliorations ergonomiques	15	12	8	4	0
- Autocritique	10	8	5	2	0
- Normes et qualité langue	5	4	3	2	0
Total	...../ 100				