

Collège Rosemont
Sciences Informatiques et Mathématiques

Développement de programmes dans un environnement graphique
TP3

GESTINOTES

Professeure : Mme Niar Wafaa

Cours : 420-203-RE

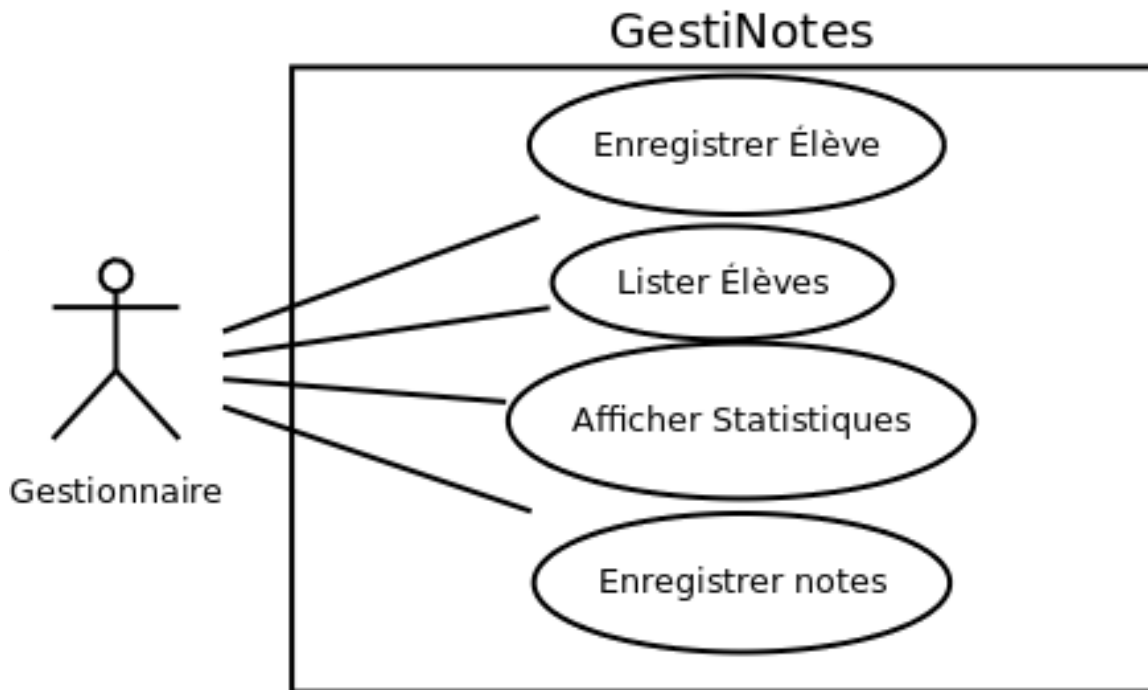
Fait par : *Dominguès Patrick et Duchaîne Raphaël*

Session : Automne 2016

1- Consignes :
(Ci-jointes)

2- ANALYSE :

2.1. Diagrammes des cas d'utilisation :



Actions Acteur	Réponses système
1-Le 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, prénom, date, note finale, code permanent) et demande si l'utilisateur 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 :

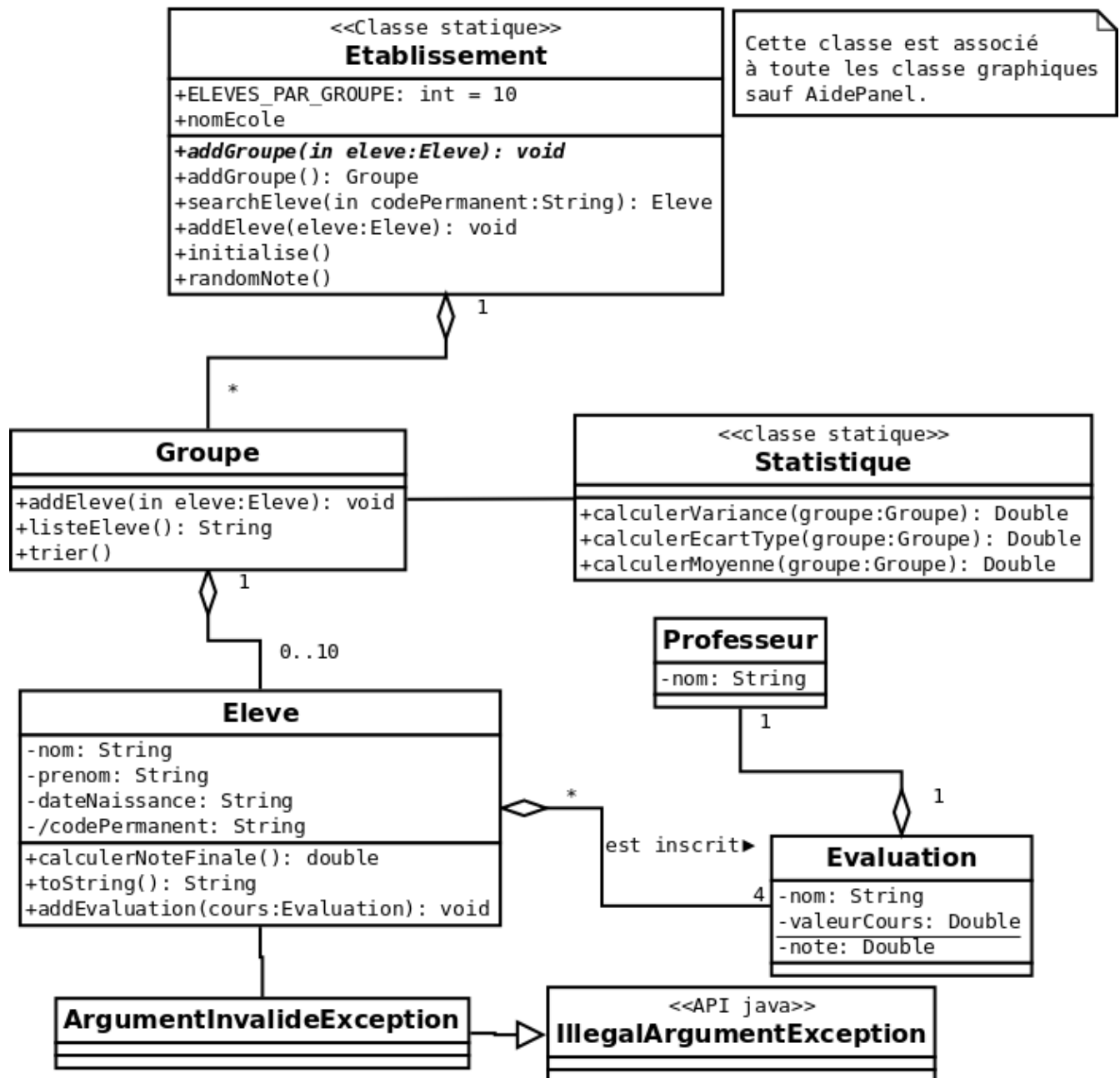
Ligne 2 :

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

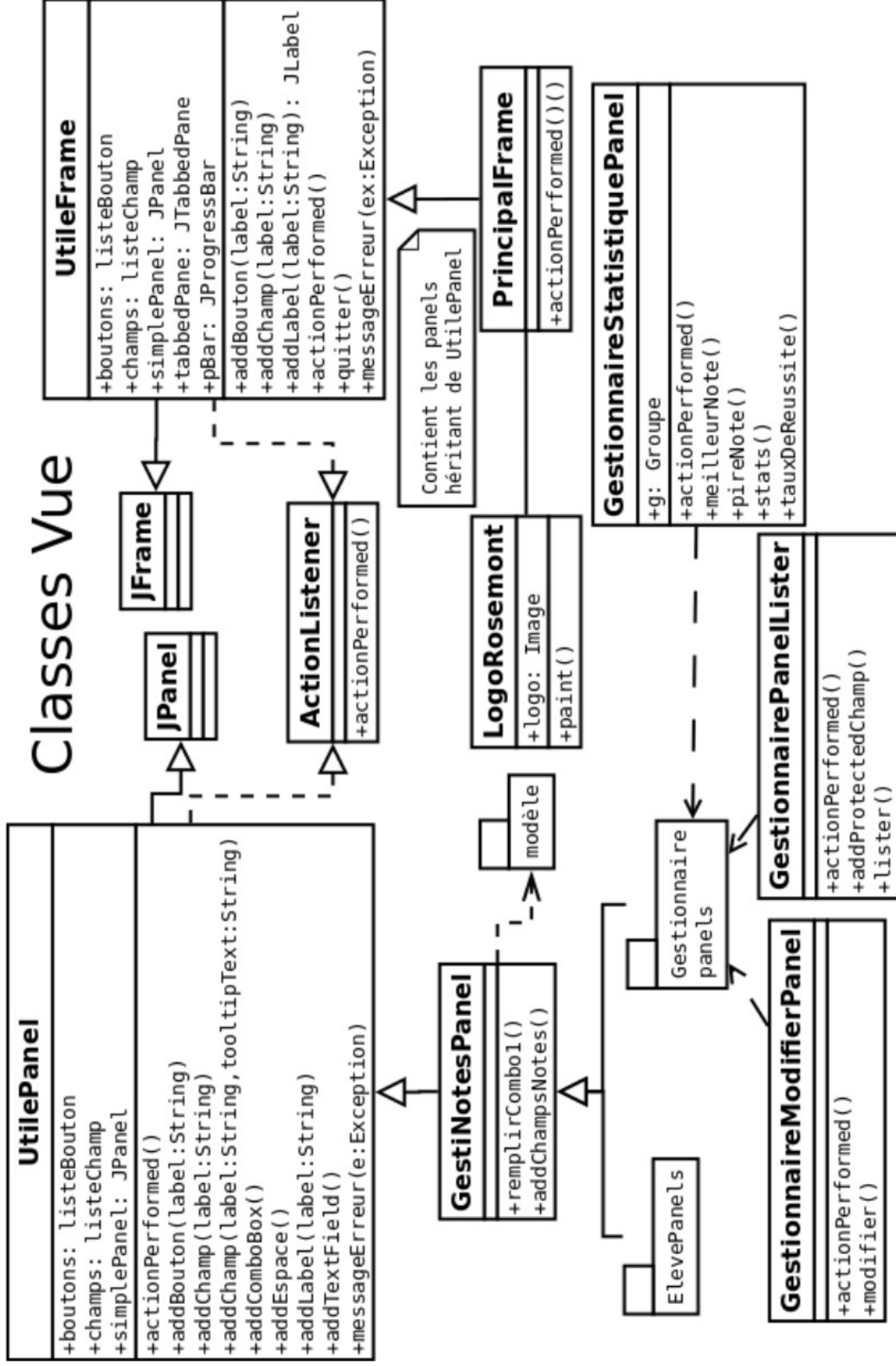
Ligne 4:

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

Classe Modèles



Classes Vue



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 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	
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	

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.