Rapport de Génie Logiciel 2

MONGODIN Maxime; HIRACLIDES Nicolas;
LESIEUR Christophe; KOWALEWSKI Matthieu

31 mai 2014

Table des matières

Bibliographie

[1] Exemple de diagrammes de cas d'utilisation. http://uml.free.fr/cours/p10.html

[2] Exemple de diagrammes de classes http://uml.free.fr/cours/i-p14.html

[3] Exemple de diagrammes d'activités, http://uml.free.fr/cours/i-p21.html.

Introduction

Dans le cadre de l'école virtuelle de L'EISTI nommée AREL, l'école a besoin de gérer un certain nombre de questionnaires. L'objectif du premier projet génie logiciel était d'établir l'ensemble de la base de données permettant de gérer les données générées par ces questionnaires.

Dans ce deuxième projet génie logiciel, nous devons concevoir un logiciel permettant de gérer et de traiter des QCM. En d'autres termes, nous devons développer une interface capable de créer des QCM, d'y répondre et de stocker et exploiter ces données.s

Première partie

Analyse et conception du projet

Chapitre 1

Présentation du projet

1.1 Expression des besoins

Notre logiciel doit pouvoir gérer trois types d'utilisateurs : enseignants, élèves et administrateurs.

Les administrateurs ont comme fonction de :

- définir d'autres utilisateurs quel que soit leur rôle. Pour un professeur ils préciseront le domaine d'enseignement.
- définir une promotion (liste d'élèves).
- définir et modifier les modules enseignés à l'école en prenant en compte le nom du module, les modules prérequis et le syllabus du module.

Les professeurs doivent pouvoir :

- définir un QCM qui comprend des questions, des réponses prédéfinies (au moins deux par question) vraies ou fausses.
- chaque QCM peut être privé ou public. C'est-à-dire qu'un QCM privé n'est utilisable que par le professeur qui l'a créé. Le rendre public, le rend utilisable par tous les professeurs.
- créer une session de QCM avec dates de début et de fin, associée à un module et une promotion. On peut répondre à un QCM plusieurs fois seulement si cela est précisé par le professeur.
- consulter les résultats complets ou partiels des QCM à tout moment soit par élève, soit sous forme de statistiques (écart type, moyenne, ...)

Les élèves quant à eux peuvent :

- répondre à un QCM dans lequel il est inscrit entre les dates de début et fin du QCM et voir un message d'avertissement en dehors de ces dates.
- consulter les résultats d'un QCM auquel il a participé une fois que la session est fermée.

1.2 Contrainte Plateforme

Pour ce projet, nous avons comme contrainte d'implémenter le code sous Java version 1.7. Nous devons aussi utiliser comme méthode d'analyse et de conception l'UML avec la persistance des données. Nous utiliserons StarUML afin de réaliser les différents diagrammes. Les objets doivent communiquer entre

eux à travers des interfaces. la sauvegarde des données à la fin du programme ou sur commande pour les données modifiées uniquement est obligatoire.

1.3 Ressources Humaines

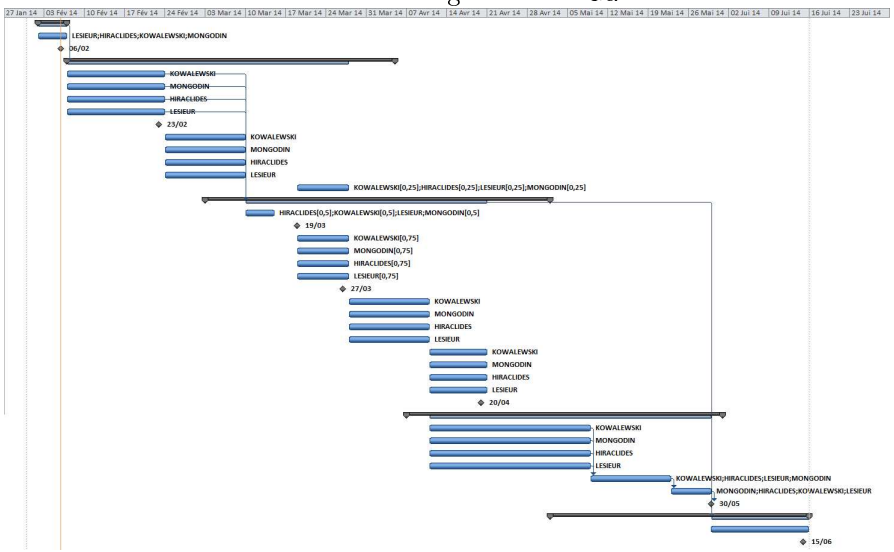
Notre équipe se compose de quatre membres qui sont les suivants :

- MONGODIN Maxime
  - Etudiant à l'EISTI de Cergy
  - connaissance informatique : C, C++, langage web, Ocaml, Pascal, Java.
- KOWALEWSKI Matthieu
  - Etudiant à l'EISTI de Cergy
  - connaissance informatique : C, langage web, Ocaml, Pascal, Java.
- HIRACLIDES Nicolas
  - Etudiant à l'EISTI de Cergy
  - connaissance informatique : C, langage web, Ocaml, Pascal, Java.
- LESIEUR Christophe
  - Etudiant à l'EISTI de Cergy
  - connaissance informatique : C, langage web, Ocaml, Pascal, Java.

1.4 Planning

Nous avons décidé d'attribuer à chaque membre de l'équipe une partie du projet qu'il devrait analyser, conceptualiser et implémenter du début à la fin du projet. Les découpages actuelles des tâches vont sans doute évoluer en fonction du temps et de l'avancement du projet, ainsi que pour respecter l'équité de la quantité de travail à fournir par chaque membre. Voici donc notre diagramme de Gantt :

FIGURE 1.1 – Diagramme de Gantt



Les différentes parties du diagramme sont détaillées ici :

FIGURE 1.2 – Cahier Des charges

Cahier des charges	**5 jours**	**Dim 02/02/14**	**Jeu 06/02/14**
Cahier des charges	5 jours	Dim 02/02/14	Jeu 06/02/14
jalon 1	0 jour	Jeu 06/02/14	Jeu 06/02/14

FIGURE 1.3 – Analyse

Analyse	**57 jours**	**Ven 07/02/14**	**Ven 04/04/14**
Diagramme de cas d'utilisation et de classe général	17 jours	Ven 07/02/14	Dim 23/02/14
Diagramme de cas d'utilisation et de classe professeur	17 jours	Ven 07/02/14	Dim 23/02/14
Diagramme de cas d'utilisation et de classe élève	17 jours	Ven 07/02/14	Dim 23/02/14
Diagramme de cas d'utilisation et de classe administrateur	17 jours	Ven 07/02/14	Dim 23/02/14
jalon 2	0 jour	Dim 23/02/14	Dim 23/02/14
Diagramme de classe avancé général	14 jours	Lun 24/02/14	Dim 09/03/14
Diagramme de classe avancé professeur	14 jours	Lun 24/02/14	Dim 09/03/14
Diagramme de classe avancé élève	14 jours	Lun 24/02/14	Dim 09/03/14
Diagramme de classe avancé administrateur	14 jours	Lun 24/02/14	Dim 09/03/14
vérification de l'analyse	9 jours	Mer 19/03/14	Jeu 27/03/14

FIGURE 1.4 – Conception

conception	**60 jours**	**Lun 03/03/14**	**Jeu 01/05/14**
Préparation de soutenance	5 jours	Lun 10/03/14	Ven 14/03/14
Soutenance	0 jour	Mer 19/03/14	Mer 19/03/14
Diagramme d'état d'activité général	9 jours	Mer 19/03/14	Jeu 27/03/14
Diagramme d'état d'activité professeur	9 jours	Mer 19/03/14	Jeu 27/03/14
Diagramme d'état d'activité élève	9 jours	Mer 19/03/14	Jeu 27/03/14
Diagramme d'état d'activité administrateur	9 jours	Mer 19/03/14	Jeu 27/03/14
jalon 3	0 jour	Jeu 27/03/14	Jeu 27/03/14
Diagramme d'état de séquence général	14 jours	Ven 28/03/14	Jeu 10/04/14
Diagramme d'état de séquence professeur	14 jours	Ven 28/03/14	Jeu 10/04/14
Diagramme d'état de séquence élève	14 jours	Ven 28/03/14	Jeu 10/04/14
Diagramme d'état de séquence administrateur	14 jours	Ven 28/03/14	Jeu 10/04/14
Diagramme d'état de transition général	10 jours	Ven 11/04/14	Dim 20/04/14
Diagramme d'état de transition professeur	10 jours	Ven 11/04/14	Dim 20/04/14
Diagramme d'état de transition élève	10 jours	Ven 11/04/14	Dim 20/04/14
Diagramme d'état de transition administrateur	10 jours	Ven 11/04/14	Dim 20/04/14
jalon 4	0 jour	Dim 20/04/14	Dim 20/04/14

FIGURE 1.5 – Implémentation

Implémentation	**55 jours**	**Lun 07/04/14**	**Sam 31/05/14**
Implémentation générale	28 jours	Ven 11/04/14	Jeu 08/05/14
Implémentation professeur	28 jours	Ven 11/04/14	Jeu 08/05/14
Implémentation élève	28 jours	Ven 11/04/14	Jeu 08/05/14
Implémentation administrateur	28 jours	Ven 11/04/14	Jeu 08/05/14
Mise en commun des implémentations	14 jours	Ven 09/05/14	Jeu 22/05/14
Batterie de test	7 jours	Ven 23/05/14	Jeu 29/05/14
jalon 5	0 jour	Ven 30/05/14	Ven 30/05/14

FIGURE 1.6 – Synthèse

synthèse	45 jours	Ven 02/05/14	Dim 15/06/14
Version livrable du programme	17 jours	Ven 30/05/14	Dim 15/06/14
jalon 6	0 jour	Dim 15/06/14	Dim 15/06/14

Chapitre 2

Analyse

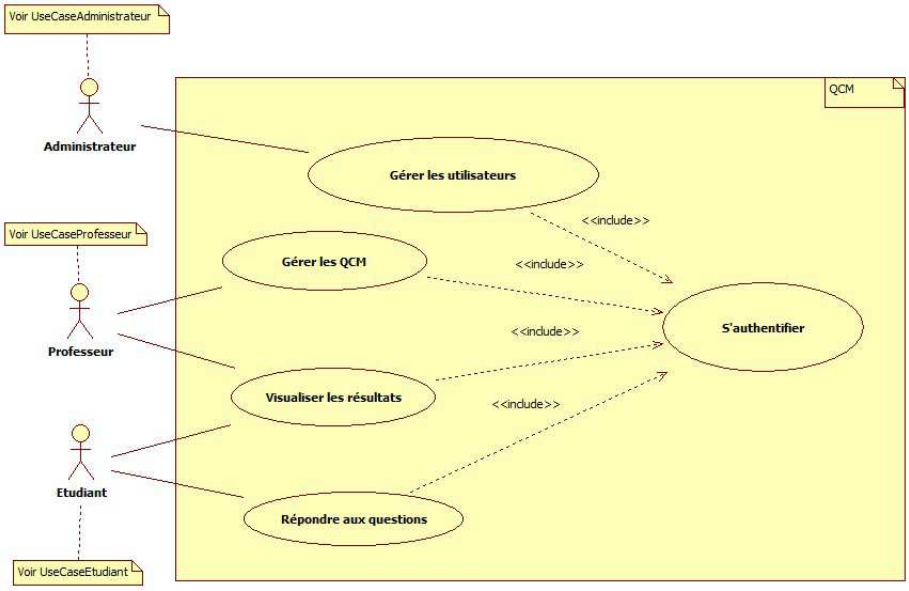
2.1 Cas d'utilisation

Nous avons modélisé le projet sous forme de différents diagrammes de cas d'utilisation afin de pouvoir définir l'ensemble des différents besoins que nous devons réaliser pour ce projet. [1]

2.1.1 Diagramme Général

Le premier diagramme est un diagramme plutôt général, il est peu détailler mais reprend globalement les différentes fonctionnalités demandées.

FIGURE 2.1 – Diagramme Général



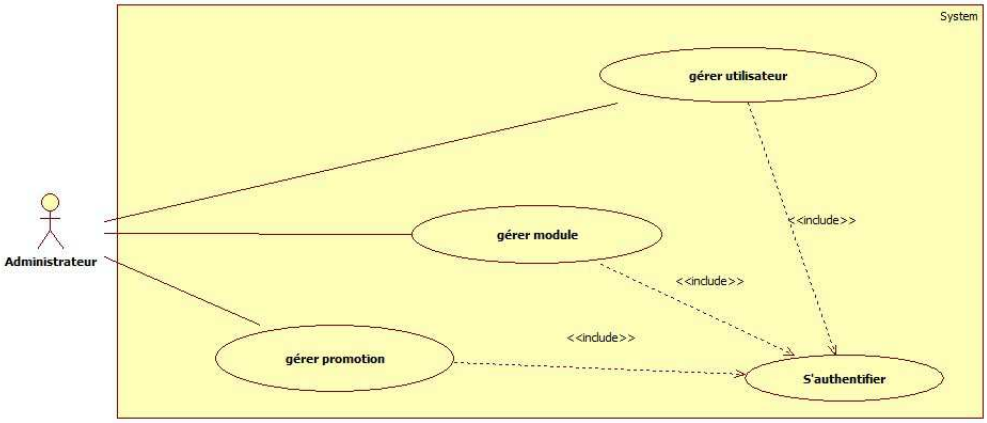
Fiche de descriptive : s'authentifier	
Description	Tous les utilisateurs doivent se connecter avant de réaliser les différentes actions qui leur sont attribuées
Règle d'initiation	Les utilisateurs doivent déjà avoir un compte d'authentification créé par un administrateur
Règle de terminaison	L'authentification se déconnecte à la fin de l'utilisation du logiciel ou à la déconnexion de ce dernier.
Règle d'exception	Aucune
Relation	Il s'agit d'un prérequis pour accéder à n'importe quelles autres fonctionnalités.

Les différents cas d'utilisations présentés sur le diagramme sont expliqués plus en détails par la suite.

2.1.2 Diagramme de cas d'utilisation pour les administrateurs

Ce diagramme est en rapport avec les administrateurs.

FIGURE 2.2 – Diagramme Administrateur



Fiche de descriptive : Gérer utilisateurs	
Description	L'administrateur doit pouvoir, via une autentification, créer un utilisateur
Règle d'initiation	L'administrateur doit avoir son nom d'utilisateur et son mot de passe valide pour s'authentifier. L'utilisateur créé ne doit pas déjà exister.
Règle de terminaison	Création d'un utilisateur dans la base de données.
Règle d'exception	Aucune
Relation	Ce cas inclus le cas s'autentifier Ce cas peut entrainer la précision du domaine d'enseignement dans le cas où un professeur est défini

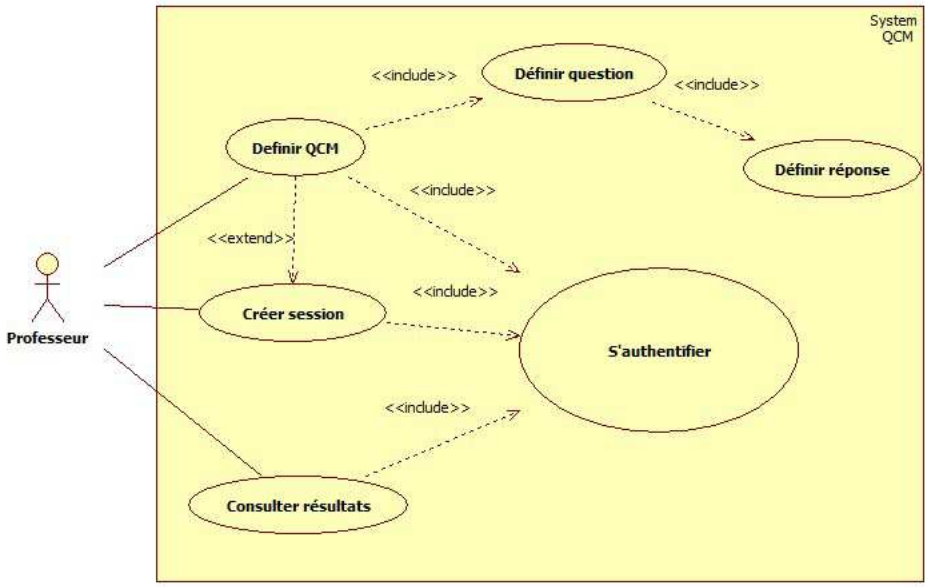
Fiche de descriptive : Gérer module	
Description	L'administrateur doit pouvoir, via une autentification, créer un module
Règle d'initiation	Le module créé ne doit pas déjà exister. Le module désigné comme prérequis doit exister. Pour modifier un module, ce dernier doit exister.
Règle de terminaison	Création d'un module dans la base de données.
Règle d'exception	Aucune
Relation	Ce cas inclus le cas s'autentifier Ce cas peut entrainer la précision des modules prérequis

Fiche de descriptive : Gérer promotion	
Description	L'administrateur doit pouvoir, via une autentification, créer une promotion
Règle d'initiation	La promotion créée ne doit pas déjà exister. Les élèves d'une promotion doivent exister.
Règle de terminaison	Création d'une promotion dans la base de données.
Règle d'exception	Aucune
Relation	Ce cas inclus le cas s'autentifier

2.1.3 Diagramme de cas d'utilisation pour les professeurs

Ce diagramme est en rapport avec les professeurs.

FIGURE 2.3 – Diagramme Professeur



Fiche de descriptive : Définir QCM	
Description	Création d'un ensemble de questions, elles-mêmes munies de plusieurs réponses fermées (qualitative nominale), et qui sont définies commes vraies ou fausses. Lors de la création du QCM, le professeur précise s'il est privé (seul le créateur peut l'utiliser) ou public (d'autres professeurs peuvent l'utiliser). Un libellé doit aussi être donné pour identifier ledit QCM.
Règle d'initiation	Le professeur est authentifié et choisit de créer un QCM. Une QCM possédant le même libellé ne doit pas déjà exister. Une QCM doit être composé d'au moins 1 question. Une question doit avoir au moins 2 réponses. Une réponse est soit vraie, soit fausse.
Règle de terminaison	Un objet QCM est créé. Le libellé est disponible dans la liste des QCMs disponible lors d'une session. Invite de création de session pour le QCM créé.
Règle d'exception	Aucune
Relation	Le cas d'usage 'S'authentifier' est inclus dans le cas d'usage 'Définir QCM' car il permet au professeur de s'authentifier. Les cas d'usage 'Définir question' et, par extension, 'Définir réponse', sont inclus dans 'Définir QCM' car ils sont des étapes nécessaires à la création d'un QCM.

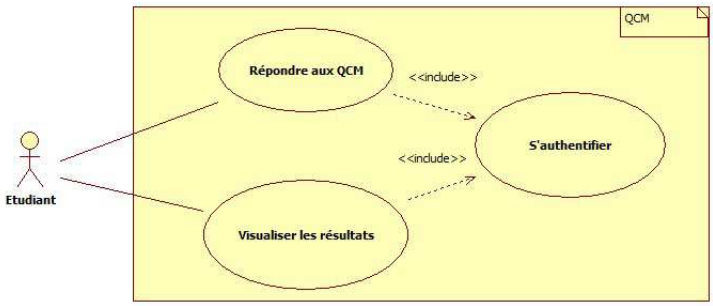
Fiche de descriptive : Créer Session	
Description	Création d'une session de QCM donné, limitée dans le temps par une date de début et une date de fin, toute deux précisées par le professeur. Cette session est associée à une promotion, pour savoir quels élèves doivent passer le QCM ; un module, pour savoir dans le cadre de quel enseignement ces élèves sont interrogés. Un nombre de répétitions (par défaut, 1) est aussi indiqué, pour donner le nombre maximum de fois où un élève donné a le droit de répondre au QCM.
Règle d'initiation	Le professeur doit être authentifié. Le QCM indiqué doit exister. La promotion doit exister et être non vide. Le module doit exister. La date de début doit être avant la date de fin. Un nombre de répétitions est attendu.
Règle de terminaison	Création d'un objet session. Création d'un objet résultats, associé à la session créée, qui stockera les réponses données par chaque élève lors de cette session.
Règle d'exception	Si le QCM indiqué n'existe pas, alors le professeur peut choisir d'en créer un. La date de début peut être la même que celle de fin : la session dure alors une unique journée. Si le aucune nombre de répétitions n'est précisé, celui-ci est fixé par défaut à 1.
Relation	Le cas d'usage 'S'authentifier' est inclus dans le cas d'usage 'Séfinir session' car il permet au professeur de s'authentifier. Le cas d'usage 'Définir QCM' est une extension de 'Définir session', car il peut être appelé au cours de celui-ci.

Fiche de descriptive : Consulter les résultats	
Description	Un professeur doit pouvoir consulter les résultats des sessions qu'il a créées. Ceux-ci peuvent être définitifs (session terminée), ou bien partiels (session toujours en cours). Deux types de résultats peuvent alors être visualisés : pour chaque élève, ses réponses ainsi que son score final, ou bien les statistiques de l'ensemble des élèves, à savoir la moyenne et l'écart-type de tous les scores, et la fréquence de bonnes réponses par question.
Règle d'initiation	Le professeur doit être authentifié. Le professeur voulant consulter les résultats d'une session doit être le créateur de cette session. La session doit avoir commencé (des résultats doivent être enregistrés pour être visualisés).
Règle de terminaison	Aucune
Règle d'exception	Si la session n'a pas encore commencé, un message affiche cet état de fait et invite le professeur à retenter le jour de début de session.
Relation	Le cas d'usage 'S'authentifier' est inclus dans le cas d'usage 'Consulter résultats' car il permet au professeur de s'authentifier.

2.1.4 Diagramme de cas d'utilisation pour les étudiants

Ce diagramme est en rapport avec les étudiants.

FIGURE 2.4 – Diagramme Etudiant



Fiche de descriptive : Répondre aux QCM	
Description	L'étudiant peut, via une authentification, répondre à des QCM.
Règle d'initiation	L'étudiant doit être dans la promotion qui répond à ce QCM et il doit se connecter durant la session.
Règle de terminaison	Une fois que l'étudiant à répondu à un QCM, on doit noter l'étudiant comme ayant validé ce QCM, récupérer les réponses données et les stocker.
Règle d'exception	Affichage d'un message d'erreur si l'étudiant essaye de répondre aux QCM en dehors des dates de début et de fin du QCM
Relation	Répondre aux QCM nécessite de s'authentifier

Fiche de descriptive : Visualiser les résultats	
Description	L'étudiant peut, via une authentification, visualiser ses résultats.
Règle d'initiation	L'étudiant doit avoir répondu aux QCM, doit se connecter une fois la session terminée, et il ne doit pas pouvoir visualiser les QCM auquel il n'a pas répondu.
Règle de terminaison	Aucune
Règle d'exception	Aucune
Relation	Visualiser les résultats nécessite de s'authentifier

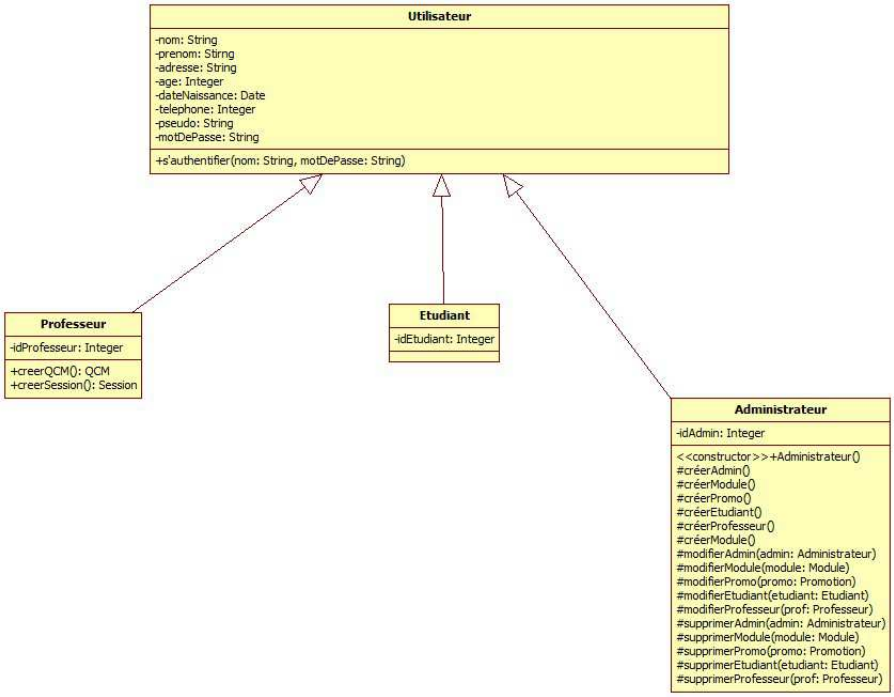
2.2 Diagrammes de classe

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Ce diagramme fait partie de la partie statique d'UML car il fait abstraction des aspects temporels et dynamiques. [2]

2.2.1 Diagramme de classe général

Dans un premier temps nous nous intéressons à la super classe Utilisateur dont un certain nombre de propriétés va engendrer des attributs pour les différents utilisateurs du logiciel. Chacun des utilisateurs est détaillé dans la suite du rapport.

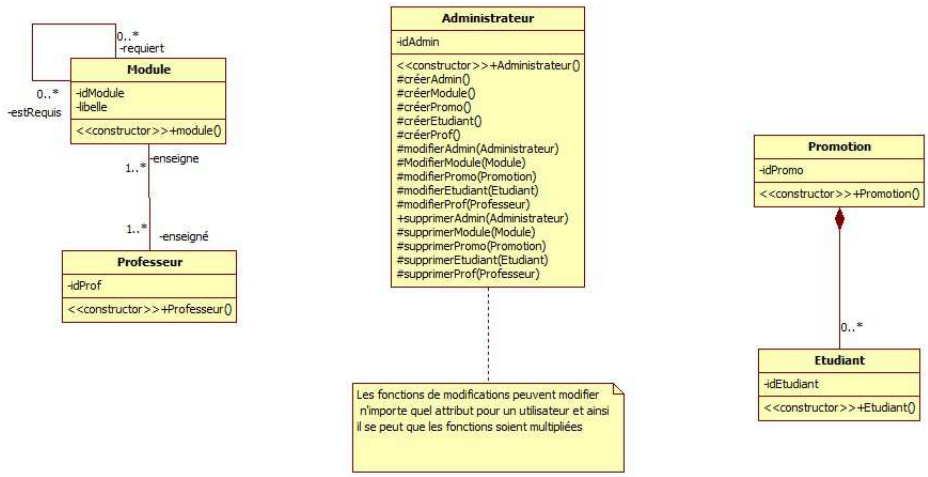
FIGURE 2.5 – Diagramme Administrateur



2.2.2 Diagramme de classe relatif aux administrateurs

Les administrateurs peuvent créer, modifier ou même supprimer un certain nombre d'éléments dans le logiciel. Un premier administrateur devra être ajouter manuellement afin de pouvoir générer un ensemble d'utilisateurs et de QCM.

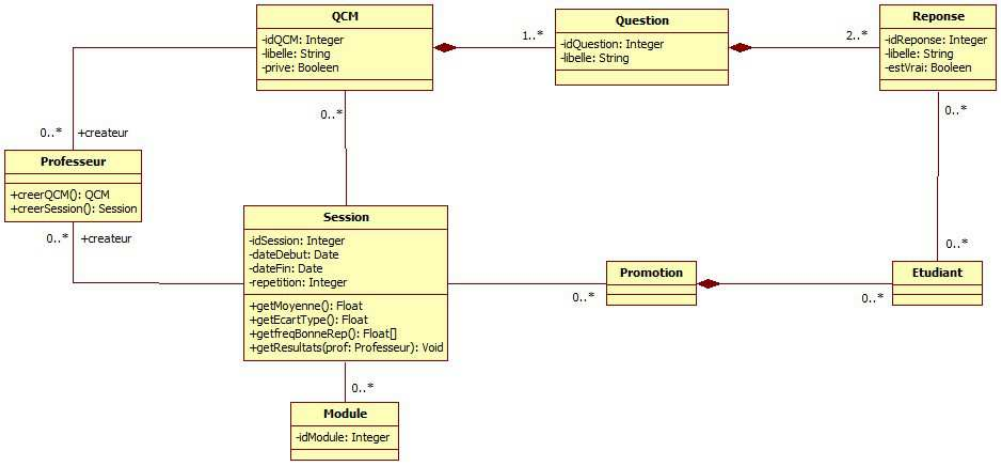
FIGURE 2.6 – Diagramme Administrateur



2.2.3 Diagramme de classe relatif aux professeurs

Le professeur peut créer un QCM ainsi que la session associée à ce QCM. On y retrouve ici un lien avec l'étudiant qui est détaillé dans le prochain diagramme.

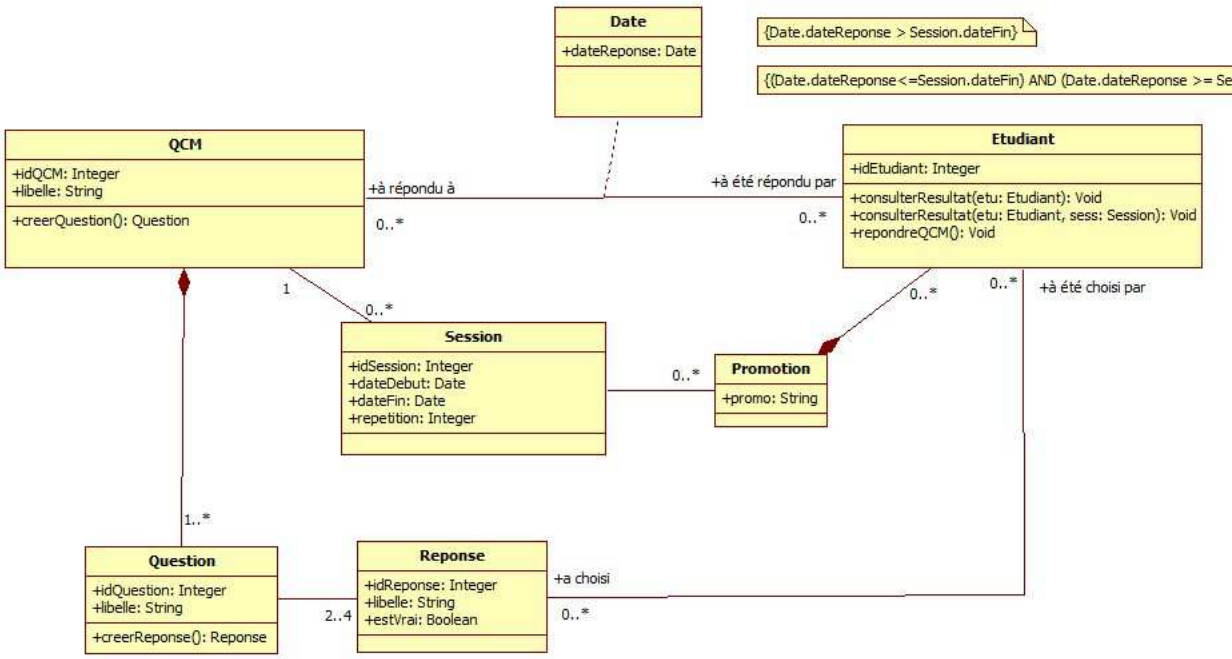
FIGURE 2.7 – Diagramme Professeur



2.2.4 Diagramme de classe relatif aux étudiants

L'étudiant va pouvoir répondre aux questions d'un ou plusieurs QCM dans la limite ou la session est encore active. Si la session est terminée l'étudiant peut alors visualiser ses résultats.

FIGURE 2.8 – Diagramme Etudiant



Chapitre 3

Conception

Par la suite, on s'est focalisé sur la conception du projet avec de nouveaux diagrammes.

3.1 Diagrammes d'activités

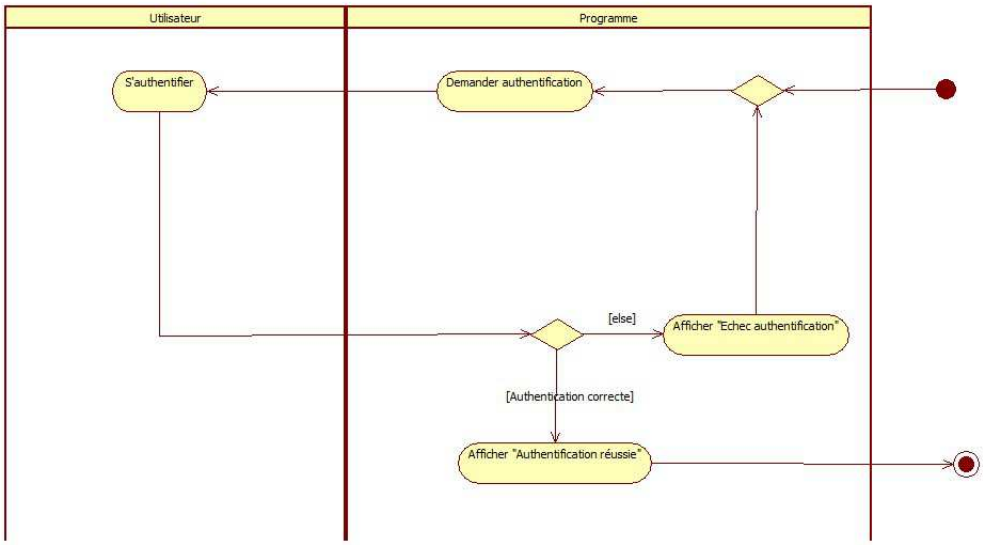
UML permet de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation, à l'aide de diagrammes d'activités.

- Une activité représente une exécution d'un mécanisme, un déroulement d'étapes séquentielles.
- Le passage d'une activité vers une autre est matérialisé par une transition.
- Les transitions sont déclenchées par la fin d'une activité et provoquent le début immédiat d'une autre (elles sont automatiques).
- En théorie, tous les mécanismes dynamiques pourraient être décrits par un diagramme d'activité, mais seuls les mécanismes complexes ou intéressants méritent d'être représentés. [3]

3.1.1 Diagramme général

Ce diagramme correspond à la méthode d'authentification de chaque utilisateur. En effet chaque utilisateur doit se connecter afin de réaliser différentes actions sur le logiciels Voici le diagramme d'activité correspondant :

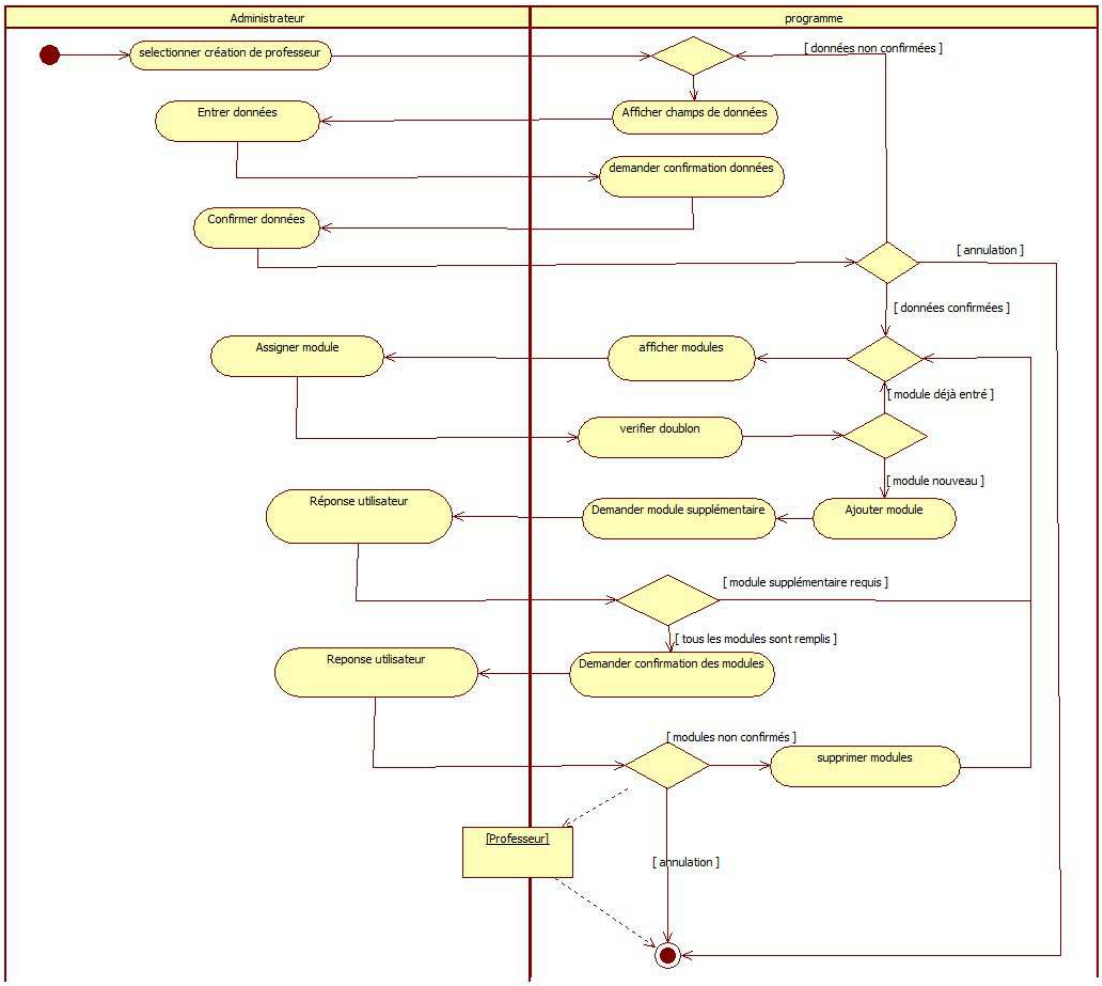
FIGURE 3.1 – Diagramme de création de Professeur



3.1.2 Diagramme de création de Professeur

Ce diagramme correspond à une méthode de l'administrateur. C'est un diagramme générique : il peut aussi être appliqué dans les grandes lignes à la création d'étudiants ou d'administrateurs. L'administrateur doit tout d'abord entrer les données générales d'un utilisateur, i.e. nom, prénom et adresse. Si ces données ne sont pas confirmées ensuite, l'utilisateur devra recommencer ; sinon il pourra accéder à l'affectation de modules à un professeur. Pour chaque module assigné, le programme vérifie s'il n'existe pas déjà dans la liste des modules du professeur. S'il l'est déjà, l'utilisateur devra choisir un autre module. Sinon, il sera ajouté, et l'utilisateur pourra choisir s'il veut ajouter un autre module. Le cas échéant, après demande de confirmation de la liste, une ultime demande de validation de création sera proposée et 3 choix sont possibles : la suppression de la liste de modules, qui ramène alors à l'assignation de modules, l'annulation de création de professeur, qui sort de de la méthode sans créer d'objet, et enfin la validation, qui crée l'objet et le sauvegarde, avant de sortir de la méthode.

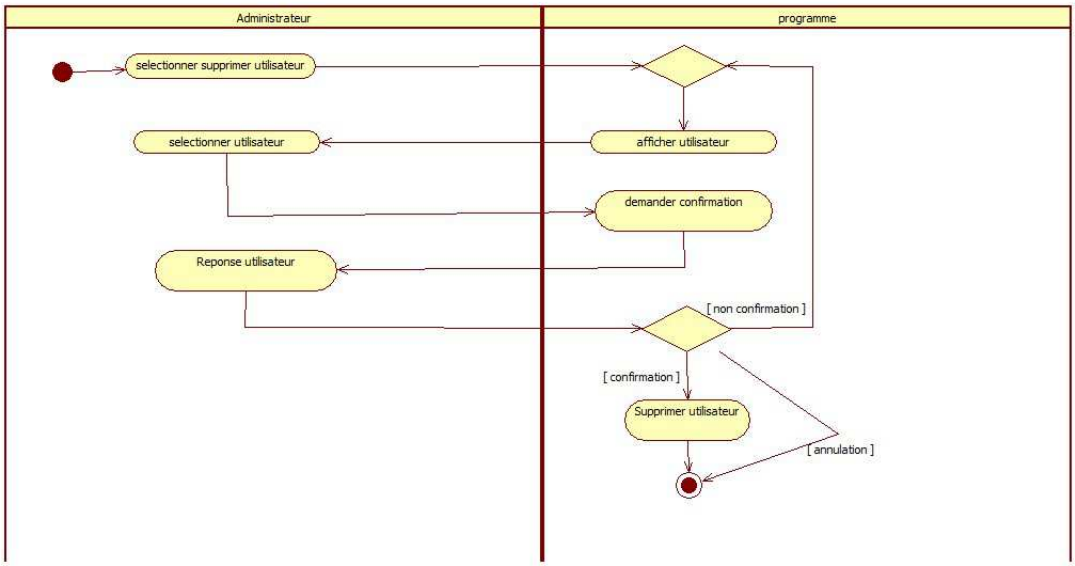
FIGURE 3.2 – Diagramme d'authentification



3.1.3 Diagramme de suppression d'utilisateur

Ce diagramme correspond à une méthode de l'administrateur. C'est un diagramme générique : il peut s'appliquer à toutes les classes héritant d'utilisateur, à savoir Etudiant, Professeur et Administrateur. L'utilisateur est amené à sélectionner un utilisateur parmi une liste précédemment affichée à l'écran. Une invite de confirmation lui est alors présentée pour demander s'il veut bel et bien supprimer l'utilisateur choisi. trois choix sont possibles : l'annulation, qui sort de la méthode sans rien supprimer, la non-confirmation, qui ramène au choix d'utilisateur, et enfin la confirmation, qui supprime l'utilisateur avant de sortir de la méthode.

FIGURE 3.3 – Diagramme de suppression d'utilisateur



3.1.4 Diagramme de création de QCM

Ce diagramme correspond à la méthode creerQCM de la classe Professeur. Le professeur connecté est alors invité à rentrer le nom du QCM à créer, puis à la valider. S'il est validé, la création de question est lancée, sinon l'utilisateur devra rentrer un nom à nouveau. La création de question se déroule comme suit : tout d'abord la saisie d'un libellé pour la question suivie d'une demande de confirmation pour ce libellé, puis la création de l'objet question en lui-même. une fois cet objet sauvegardé, la création des réponses de cette question est lancée, à savoir : l'entrée d'un libellé pour la réponse, suivie d'une invite de validation, tant qu'il n'existe pas au moins 2 réponses et que l'utilisateur choisit de continuer des réponses. Une fois que le professeur a choisi d'arrêter d'ajour des réponses, il peut enfin choisir la / les bonnes réponses parmi celles entrées précédemment, puis éditer la sélection de bonnes réponses s'il le choisit. Le cas échéant, la création de question est terminée et le programme demande si l'utilisateur veut créer une autre question. Si oui, la création de question est relancée, sinon la création de QCM se poursuit en demandant à l'utilisateur si le QCM est privé ou non. A l'issue de ce choix, l'objet QCM est créé et sauvegardé, et la méthode se termine.

3.1.5 Diagramme de création de session

Ce diagramme correspond à une méthode de la classe Professeur. Pour commencer, le professeur souhaitant créer une session est invité à entrer une date de début de session, et ce tant que la date entrée n'est pas à un format correct, puis de même avec une date de fin. Après ces deux entrées, le programme vérifie que la date de fin est bien après la date de début, auquel cas la création de session continue. Le cas échéant, il sera demandé d'entrer une autre date de fin. Après cela, il sera demandé d'entrer le nombre de fois qu'un même élève pourra participer à la session. Ce nombre devra être au moins égal à 1. Enfin, le professeur pourra choisir parmi les QCM disponibles (les siens et les QCM publics) lequel sera le sujet de la session, à quelle promotion la session sera proposée et enfin auquel de ses module la session sera rattachée.

3.1.6 Diagramme de réponse à un QCM

Ce diagramme correspond à une méthode de la classe Etudiant. Après avoir choisi l'option correspondante, l'étudiant se voit proposé les différents QCM disponibles. Il peut alors choisir l'un des QCM, ou quitter la méthode. Si un QCM est choisi, le programme vérifie si la session est en cours ou non. Si elle ne l'est pas, le programme revient à la sélection deQCM. Si elle l'est, le programme affiche alors la première question et ises réponses, l'étudiant est invité à choisir une ou plusieurs réponses. Ensuite, le compteur de questions est incrémenté, les réponses données enregistrées, et si le compteur n'a pas atteint le nombre de question du QCM, la question suivante est affichée. Sinon, un message de validation est affiché, le nombre de participation de l'élève à la session est incrémenté, et la méthode se termine.

3.1.7 Diagramme de visualisation de résultats

Ce diagramme correspond à une méthode de la classe Session. Elle concerne le cas de l'étudiant souhaiter ses résultats : si celle-ci est encore en cours, un message d'erreur s'affiche et renvoie l'utilisateur au menu précédent. En revanche, si la session est terminée, les résultats demandés s'affichent, et un retour au menu s'opère.

FIGURE 3.4 – Diagramme de création de QCM

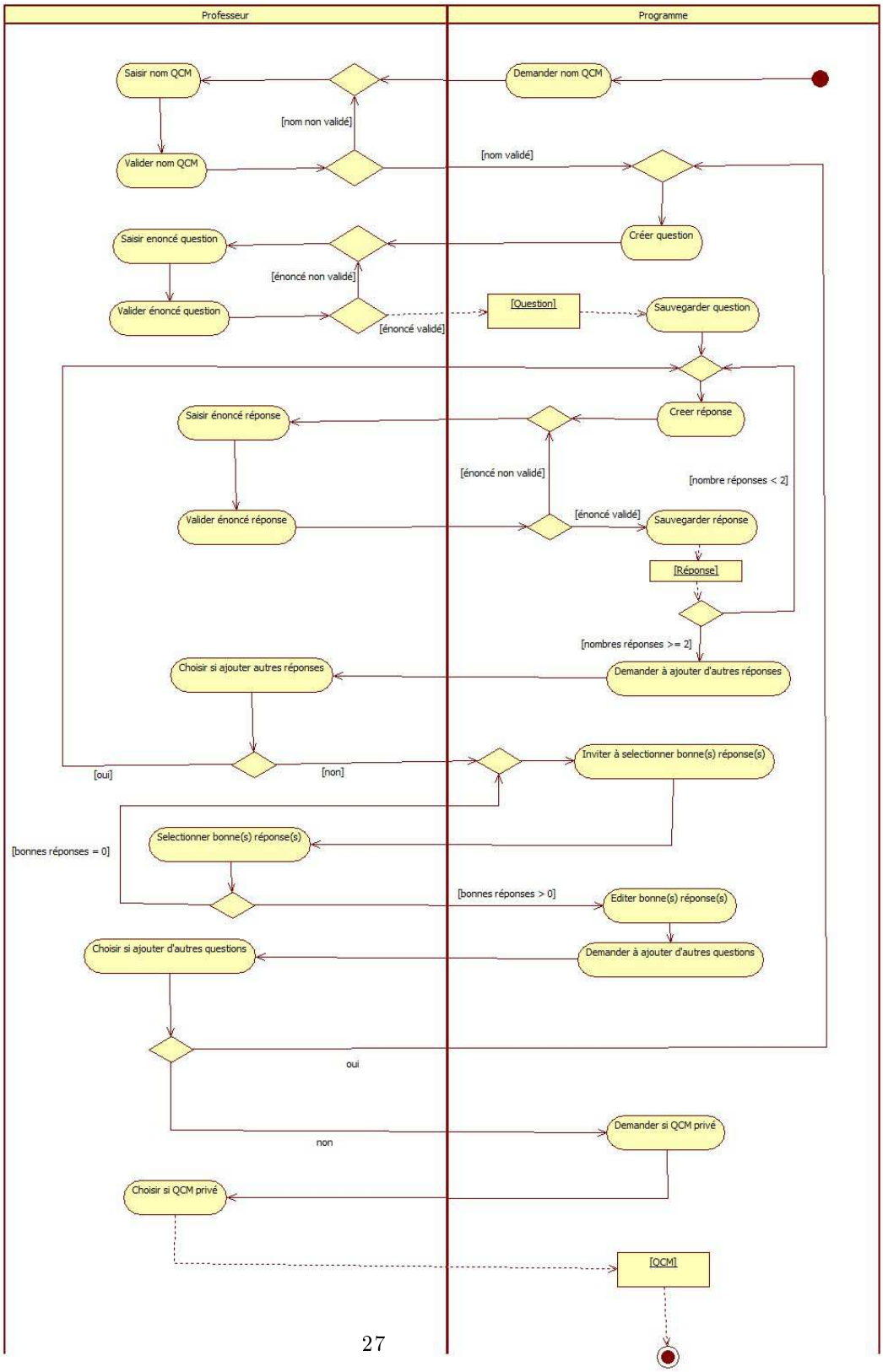


FIGURE 3.5 – Diagramme de création de session

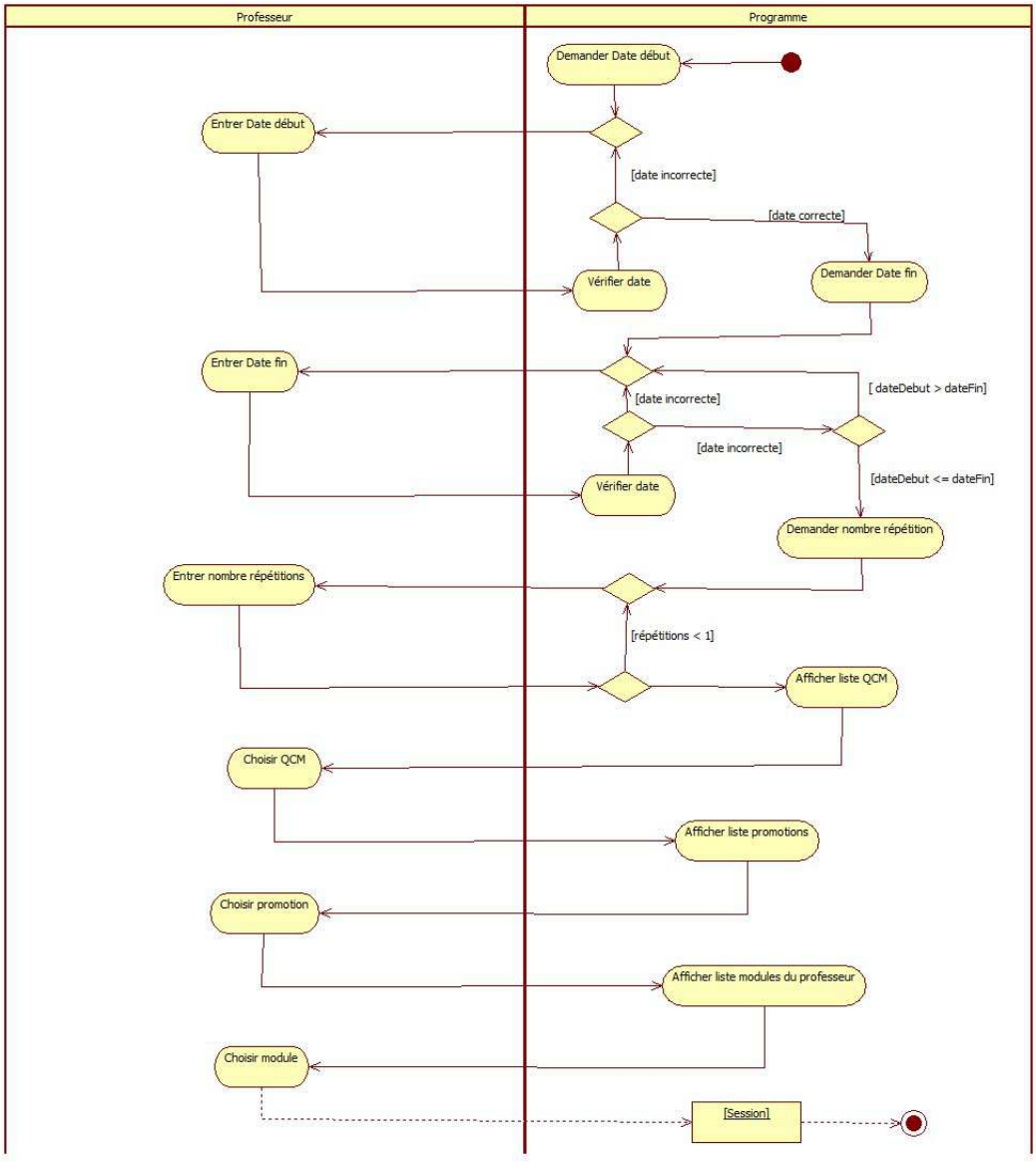


FIGURE 3.6 – Diagramme de réponse à un QCM

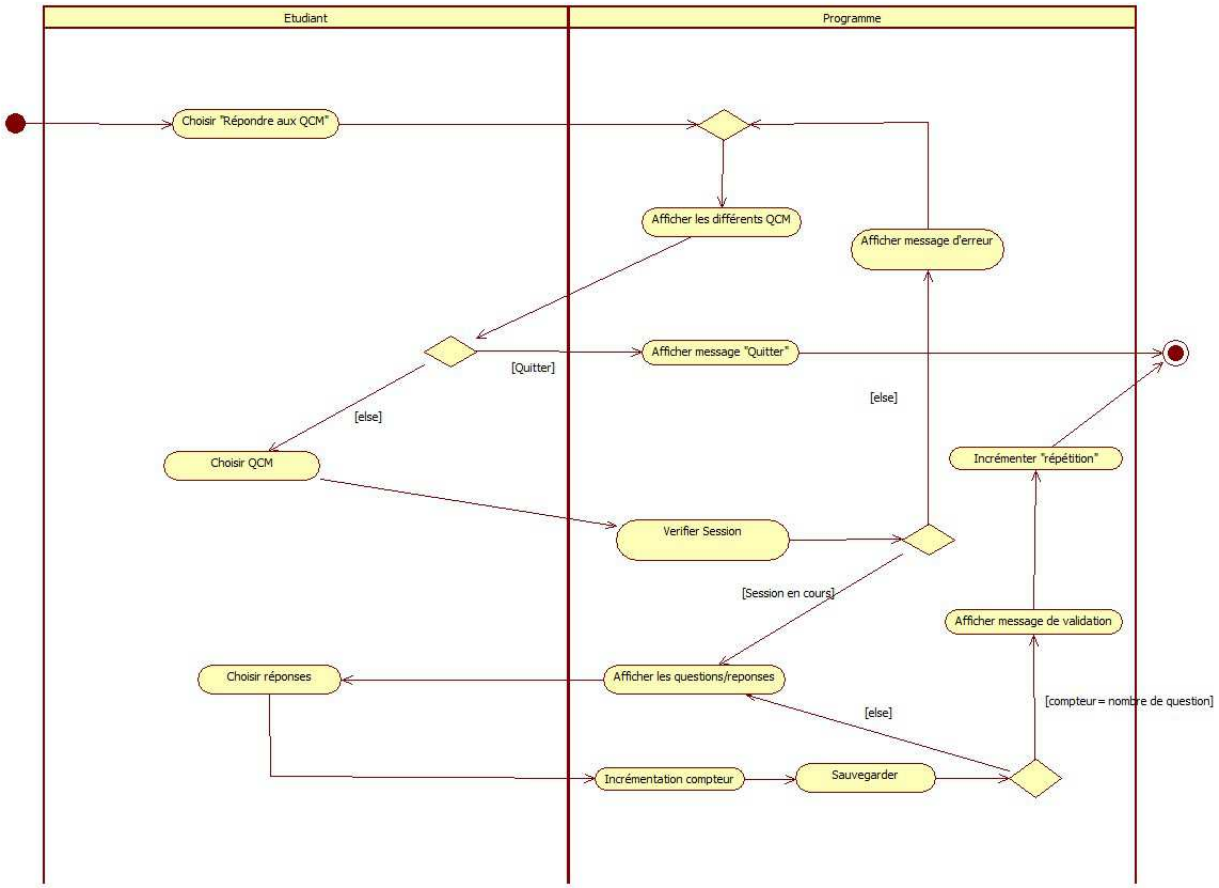
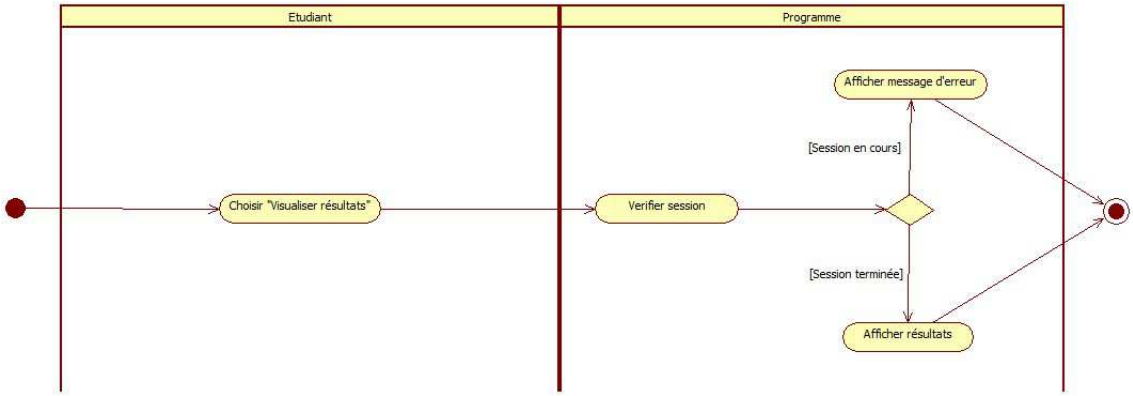


FIGURE 3.7 – Diagramme de visualisation de résultats



Deuxième partie

Présentation de notre logiciel final

3.2 Présentation

Notre logiciel final propose toutes les fonctionnalités demandées par le client. Il dispose aussi d'un système d'authentification via login et mot de passe, créés de manière procédurale lors de l'ajout de nouveaux utilisateurs par un administrateur. De cette manière, chaque utilisateur ne peut accéder exclusivement qu'aux fonctionnalités qui lui sont destinées.
Toutes nos données sont sérialisées, et enregistrées dans le dossier 'datasave' selon l'arborescence décrite dans la figure 3.1. Grâce à cela, nos données sont persistantes, il est donc possible de récupérer à tout moment toutes les données sauvegardées lors de sessions de logiciel précédentes, comme par exemple les résultats des élèves. D'autre part, notre logiciel suit le pattern MVC (Model - View - Controller). Notre dossier src est donc organisé comme indiqué dans la figure 3.2.

- main contient le fichier Main.java , qui contient la classe Main et est donc le point d'entrée de notre logiciel.
- model contient toutes les classes correspondant aux données elle-mêmes et leur traitement (comme Utilisateur et ses classes filles, ou les QCM). Ces classes sont elles organisées en trois sous-dossiers QCM, serializable et utilisateur. Enfin, une une classe dans le dossier model lui-même, RechercheDonnées, qui fournit des méthodes permettant de relire un certain type de données.
- view contient seulement la classe View, placée dans le sous dossier menu. Cette classe, à elle seule, contrôle tout l'affichage et l'entrée du logiciel.
- controller, qui contient les deux classes MenuController et AuthentificationController, qui gèrent respectivement les menus et le système d'authentification.

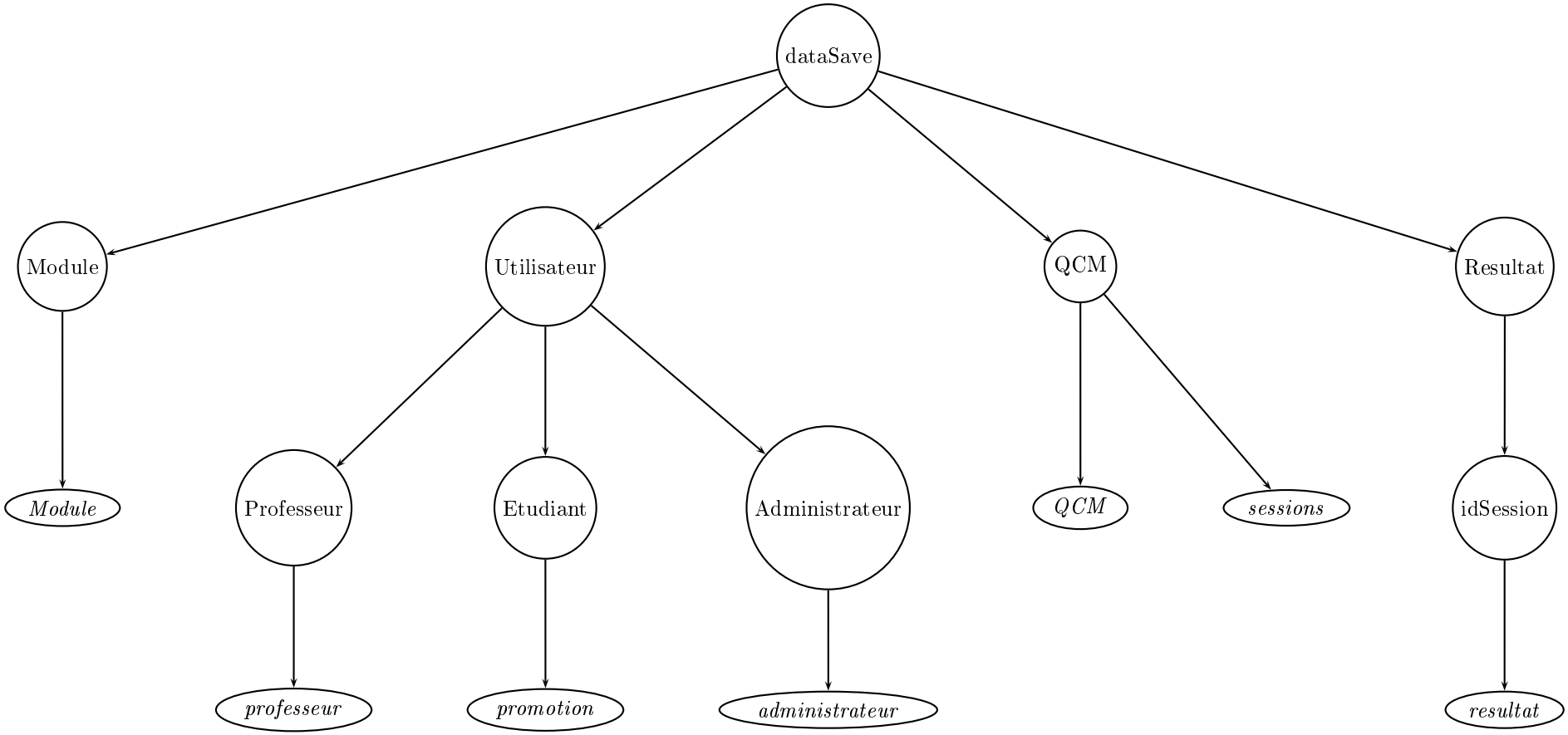


FIGURE 3.8 – Arborescence des données

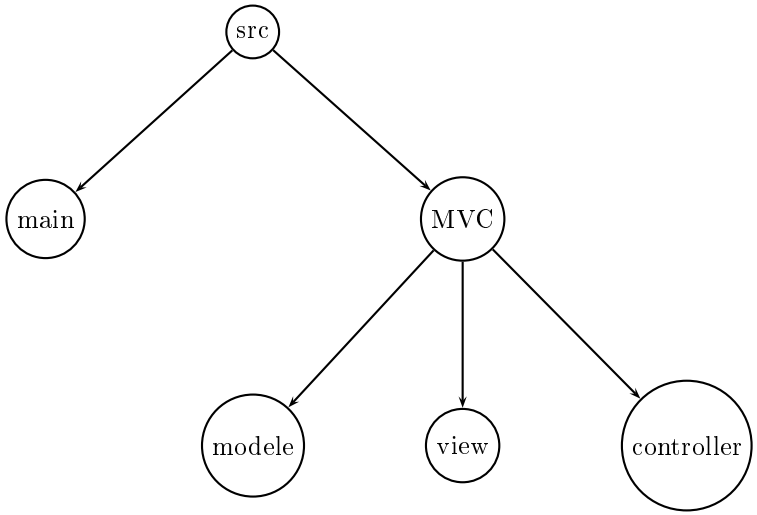


FIGURE 3.9 – Arborescence des fichiers sources

3.3 Modification de la version analytique et conceptuelle

Nous avons créé un manuel d'utilisation ainsi qu'une javadoc pour notre logiciel permettant de comprendre à la fois comment se servir de notre logiciel mais aussi comment il fonctionne en interne. Vous trouverez ces deux rapports dans les dossier "Manuel" et "Javadoc" du programme.

Pour les besoins du projet, nous avons ajouté une classe générique All<T>, qui est un Set générique sérialisable. C'est grâce à cette classe que l'intégralité de nos données est sauvegardée. D'autre part, une classe Resultat, ne figurant pas sur les diagrammes de classes, a elle aussi été ajoutée pour stocker les résultats d'un élève, à savoir : l'élève en lui-même, sa note, la liste des réponses choisies et le nombre de fois qu'il a participé à la session. Ces résultats sont sauvegardés grâce à un objet All<Resultat>, stocké dans un sous-dossier de datasave/Resultat, portant comme nom un identifiant unique correspondant à la session à laquelle correspondent les résultats.

Conclusion

Ce projet a donc été mené à terme. Nous avons réalisé les différentes étapes de création d'un logiciel : de l'analyse à la réalisation du projet en passant par l'étape de conception. Ce rapport final n'est malheureusement pas à la hauteur de nos attentes en raison d'une mauvaise gestion du temps sur la fin du projet. Toutefois, nous avons réalisé un logiciel complet, avec des documentations précises : javadoc et manuel d'utilisation. Ce projet est donc terminé. Nous avons pris beaucoup de plaisir à le réaliser, et même si certains détails restent à peaufiner, nous sommes fier du travail que notre équipe au complet a réalisé.