

OUISTIGO

Rapport de Conception



Kenza Belkacem-Boussaïd Sarunyaa Satkunarajah Hakan Gunes

Année 2017 / 2018



Historique du Document

Date	Auteur des modifications	Description des modifications	Version
15/11/17	GH	Démarrage du document	1.0
19/11/17	GH	Contexte	1.1
22/11/17	GH	Nouvelle Mise en Forme des Titres	1.2
23/11/17	GH	Besoins Fonctionnels Mise en Forme	1.3
30/11/17	GH – SS – KBB	Refonte des besoins fonctionnels	1.4
06/12/17	GH – SS – KBB	Evolution du cahier des charges	1.5
21/12/17	GH – SS	Partie Conception Ajout des Diagrammes Use Cases	2.0
24/01/18	GH – SS – KBB	Partie Conception Ajout des diagrammes de Classes Mind Map et Road Map Refonte des titres/sous-titres	3.0
25/01/18	GH – SS – KBB	Actualisation Diagramme de Classes Mise en forme Changement nom du fichier en « Rapport de Projet e-Miage »	3.1
08/01/2018	GH – SS – KBB	Diagrammes de Séquences Actualisation du Diagramme de Classes Mise à jour roadmap	3.2
22/02/2018	SS – KBB	Diagramme de Packages Diagramme d'Etats	3.3
13/03/2018	GH – SS – KBB	Remise en forme et actualisation de l'expression des besoins Remise en forme du diagramme de packages Tableau des noms anglais-français dans le diagramme de Classes	3.4
15/03/2018	GH – SS – KBB	Finalisation du rapport de Conception Description des diagrammes de Séquences et d'Etats	4.0

KBB: Kenza Belkacem-Boussaïd SS : Sarunyaa Satkunarajah **GH**: Gunes Hakan

Avant-Propos

Comme chaque année lors du M1 MIAGE à l'Université Evry Val d'Essonne, les étudiants sont confrontés à la réalisation d'un projet en lien avec leurs études.

Enjeu majeur de leur année, ce travail est l'occasion pour eux de mettre en œuvre toutes les compétences en informatique acquises durant leurs études supérieures ou bien en entreprise (stages ou alternance). A travers des thèmes concrets et variés, cet exercice offre la possibilité aux miagistes d'emmagasiner de l'expérience sur différents points : apprentissage de nouveaux langages de programmation, travail en équipe, gestion du temps ou bien respecter une exigence de qualité

Table des Matières

Hist	oriq	ue du Document	2
Ava	nt-Pr	opos	3
Tabl	le de	s Matières	4
l.	Cah	ier des Charges	6
1.	. C	ontexte	7
	A.	Généralités	7
	В.	Présentation de l'existant	7
	C.	Problématique & Objectifs	8
	D.	Motivations Personnelles	8
2.	. Е	xpression des Besoins	. 11
	A.	Utilisateur quelconque	. 11
	В.	Responsable Pédagogique	. 12
	C.	Tuteur	13
	D.	Apprenant	13
	E.	Secrétariat	14
	F.	Service Formation en Continue	. 15
II.	Con	ception	16
1.	. D	viagrammes de Cas d'Utilisations	. 18
	A.	Responsable Pédagogique	. 18
	В.	Apprenant	21
	C.	Secrétariat	24
	D.	Tuteur	26
	E.	Service Formation en Continue	28
2.	. D	viagramme de Classes	30
	A.	Diagramme de Classes Général	31
	B.	Acteurs	. 33
	C.	Objets	40
3.	. D	iagramme de Packages	43
	A.	Diagramme de Packages	43

4.	Diagrammes de Séquences	44
A.	Première Inscription et Paiement	44
В.	. Préinscription à un module	45
C.	. Affectation d'un module à un tuteur	46
D.	. Validation d'un module	47
E.	. Présentiel et Paiement	48
5.	Diagrammes d'Etats	49
A.	Module	49
В.	. Live Meeting	51
III.	Annexes	53
A.	. Tableau de Bord Individuel	54
В.	. Répartition des tâches	55
C.	. Outils UtilisésError! Bookmar	k not defined.

I. Cahier des Charges



1. Contexte

A. Généralités

Grâce aux technologies de l'information, les méthodes d'enseignement sont aujourd'hui diverses et variées. Dans ce sens, et depuis plusieurs années, les formations MIAGE de France se sont organisées afin de proposer des cursus en ligne aux personnes déscolarisées, salariées, etc afin de parfaire leurs formations : ainsi est le but de l'E-MIAGE.

Cette formation à distance a pour objectif de proposer une grande souplesse aux « e-étudiant », en leurs offrants la possibilité de choisir le nombre de modules à passer durant chaque semestre, un suivi régulier avec les professeurs, etc.

Concept basé sur la digitalisation, elle recourt bien entendu à une plateforme de communication en ligne qui sert comme point de liaison aux professeurs et aux apprenants.

Pour plus de détails, voir la figure 1

B. Présentation de l'existant

Les professeurs disposent d'une plateforme leur offrant plusieurs possibilités telles que l'extraction de statistiques diverses via Excel (voir figure ci-dessous).

Ce fichier Excel reprend les données suivantes :

- Les modules suivis par l'apprenant par semestre
- Les notes aux contrôles continus et à l'examen
- Vérifier si l'apprenant a été présent aux présentiels
- Les modules déjà acquis
- Pour plus de détails, voir la figure 1

C. Problématique & Objectifs

Le but de notre projet est de proposer une plateforme dématérialisée de gestion administrative des e-étudiant (essentiellement pour faciliter le rôle du responsable pédagogique)

Notre travail concernera exclusivement la **gestion** administrative, c'est-à-dire des fonctionnalités comme, la gestion des examens, l'obtention du diplôme, etc ...

Ainsi, nous ne traiterons pas de l'aspect « universitaire » comme la mise en ligne d'un cours sur la plateforme.

La plateforme respectera des directives de conception modernes tout en offrant une productivité maximale à ses utilisateurs.

Les principaux axes de travail seront :

- Une interface graphique claire et ergonomique
- Une conception orienté objet offrant une flexibilité maximale
- Un code facilement implémentable

D. Motivations Personnelles

Tant par sa complexité fonctionnelle que technique, ce travail représente un véritable challenge à nos yeux. Nous aurons l'opportunité d'appliquer diverses connaissances mais aussi d'en acquérir de nouvelles. Nos capacités d'autogestion seront déterminantes afin de trouver les outils nécessaires pour mener à bien nos objectifs.



Figure 1: fichier Excel



PROJET PLATEFORME DE COMMUNICATION MASTER E-MIAGE

E-MIAGE

CONFORME A LA NORME "OCTAVE"

Ouvert - Cohérent - Tutoré - Accessible- Viable- Evolutif

CONTENUS EN LIGNE

Cours détaillées (web ou hors ligne), Exercices, Travaux Pratiques proposés par les professeurs





SERVICES PEDAGOGIQUES

Tutorat par mail/chat, correction de travaux, calendrier des formations, organisation de session de révisions

SUIVI ADMINISTRATIF

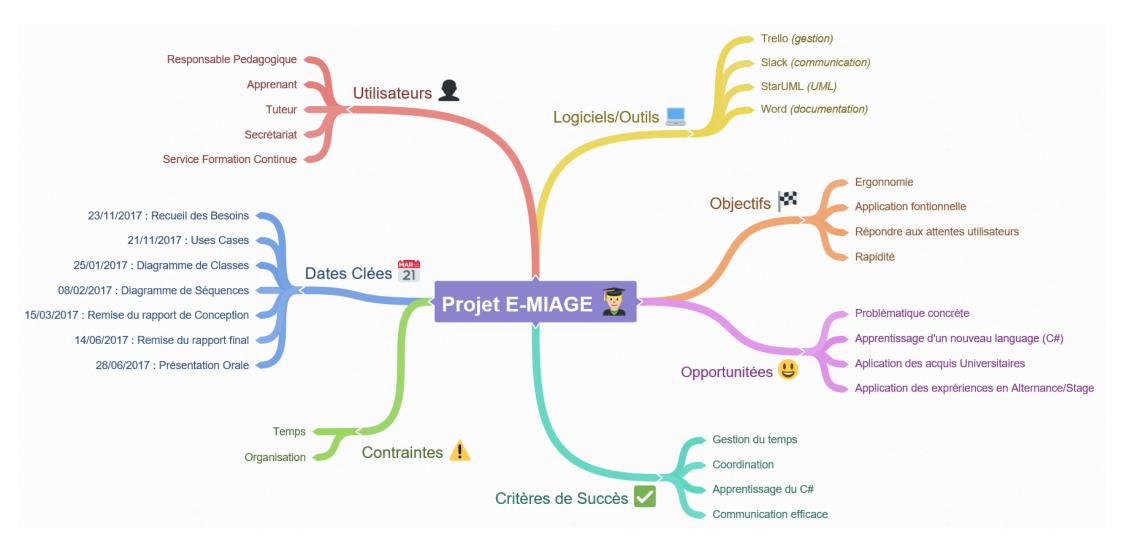
Site web permettant la gestion administrative des e-étudiants (inscriptions, affectation aux cours, désignation des professeurs, paiement des droits, etc)







<u>Figure 2</u>: Brochure d'information sur l'E-MIAGE (réalisé par Hakan Gunes)



<u>Figure 3</u> : Mind Map (réalisé par Hakan Gunes)



2. Expression des Besoins

Dans cette partie, nous étudierons tous les besoins fonctionnels selon les points de vue utilisateurs. Nous avons dégagé plusieurs profils, avec des besoins divers et variées :

- Le Responsable Pédagogique
- L'Apprenant
- Le Secrétariat
- Le Tuteur
- Le Service de Formation Continue

Pour définir les besoins des différents profils, nous avons longuement échangé avec l'actuelle Responsable pédagogique Mme Benzakki, ce qui nous a permis de mieux interpréter ses attentes vis-à-vis de l'application (notamment les évolutions de l'existant)

N.B: les modules sont amenés à évoluer

A. Utilisateur quelconque



Tous les utilisateurs inscrits sur la plateforme

Permettre à l'utilisateur de se connecter/déconnecter

Réinitialisation du mot de passe

Modifier son profil utilisateur

Les utilisateurs pourront modifier toutes les informations administratives les concernant : adresse, numéro de téléphone, adresse mail, photo de profil, etc.

B. Responsable Pédagogique



Il s'agit de l'utilisateur central de l'application Il a une vue globale des différents acteurs et pilote le service pédagogique de l'E-MIAGE

Valide les modules choisis par les apprenants

Valider les inscriptions à chaque module que l'apprenant souhaite suivre durant le semestre

Q Consulter les notes

Le responsable pédagogique doit pouvoir consulter les notes de chaque apprenant, en cas de nécessité il peut être amené à procéder à des modifications

Affecter un tuteur enseignant à un module

Une fois les modules validés, le responsable pédagogique va affecter chaque module à un tuteurenseignant, ainsi l'apprenant prendra connaissance des modalités du déroulement des modules qu'il a choisi

- ▼ Fournir des identifiants aux différents apprenants et tuteur-enseignant lors de leur première connexion
- Supprimer/Créer des nouveaux utilisateurs
- Exporter statistiques sous format Excel

C. Tuteur



Il est affecté à un module Il se charge d'accompagner les apprenants qui sont tutorés

- Consulter les notes des apprenants affectés à son module
- Récapitulatif des échanges avec les apprenants

Le tuteur accède à la plateforme pour fournir un suivi des actions menées avec son ou ses apprenants. Notamment en notifiant les interventions et/ou interactions par mail ou autre qu'il aurait avec ces derniers, et ceux dans un but de justification auprès du service de la formation continue.

Attribuer une note de contrôle continue à un apprenant

Permet de donner une note de contrôle continue aux apprenants qui lui sont afféctés

D. Apprenant



C'est 'un « e-étudiant » La plateforme lui servira de portail pour connaître toutes les informations administratives

Pré-Inscription aux examens

L'apprenant doit choisir les modules auxquelles il souhaite suivre Contraintes : Maximum 6 modules choisis par semestre

Consulter ses notes

L'apprenant aura la possibilité de consulter ses notes de partiels mais aussi de contrôles continus pour tous les modules qu'il aura passé

• Consulter les dates d'examens

Chaque apprenant pourra avoir accès à une ChekList "personnalisée" en fonction des modules auxquels il est inscrit : il aura des informations concernant les dates et lieux des examens.

A noter, qu'à la date d'approche de l'examen, l'apprenant recevra sa convocation via l'application

• Accepter les dates de présentiels

Accepte ou non les séances en présentiel en fonction de ses disponibilités.

E. Secrétariat



Acteur qui gère les notes de l'apprenant

Entrer les dates d'examens

L'apprenant doit être averti de la date de l'examen (convocation à envoyer)

• Entrer les notes des apprenants

Une fois les notes reçues (cette fonctionnalité n'est pas gérée par l'application), le secrétariat entre les notes de chaque apprenant dans l'application

Organise les séances de présentiel

F. Service Formation en Continue



Acteur qui gère le l'aspect administratif

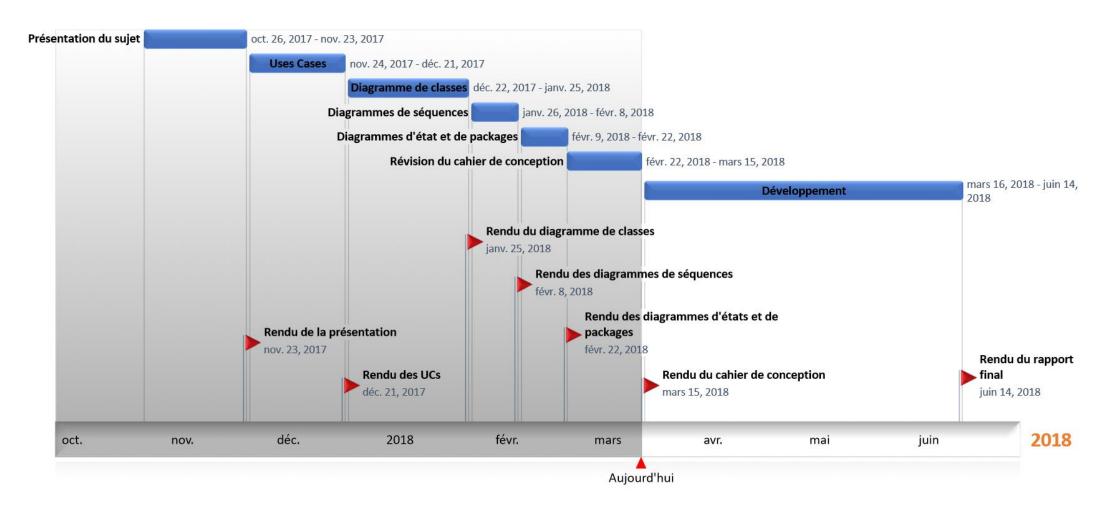
Validation des paiements et inscription définitive

Pour s'inscrire aux modules qu'il souhaite suivre, l'apprenant doit payer des droits d'inscriptions. Le paiement doit s'effectuer par module

Paiement des tuteurs

Les tuteurs sont rémunérés en fonction de la présence des apprenants aux séances de présentiel

II. Conception



<u>Figure 4</u> : Road Map au 25/01/18 (réalisé par Sarunyaa Satkunarajah)



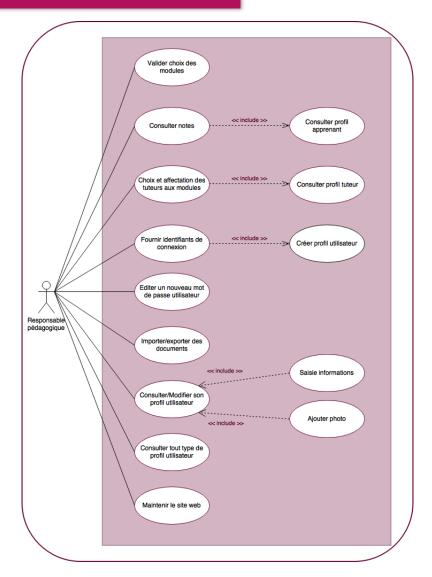
1. DIAGRAMMES DE CAS D'UTILISATIONS

Le Recueil des Besoins nous a permis de cibler les différentes attentes utilisateurs autour de la plateforme de communication eMiage.

Toutes ses attentes seront retranscrites en Cas d'Utilisations afin de cibler les modules fonctionnels de notre application.

Plus tard, nous feront un tri pour chaque utilisateur sur les modules à implémenter en priorité

A. Responsable Pédagogique

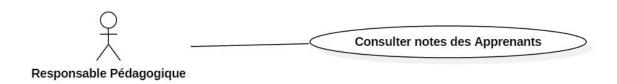


♀ Valider le module



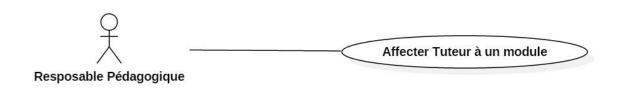
Informations	Description textuelle
Acteurs Concernés	<u>Primaire</u> : Responsable Pédagogique <u>Secondaire</u> : Apprenant
Conditions	Connexion
Post Conditions	néant
Besoins en Interface Graphique	 Avoir une liste tribale de tous les apprenants avec leurs notes Avoir des onglets avec les modules (liste moins fournie mais plus exploitable)
Scénario	 Le RP est sur sa page de gestion Il clique sur le bouton de consultation des notes

Consulter notes des Apprenants



Informations	Description textuelle
Acteurs Concernés	<u>Primaire</u> : Responsable Pédagogique <u>Secondaire</u> : Apprenant
Conditions	Connexion
Post Conditions	néant
Besoins en Interface Graphique	 Avoir une liste de tous les apprenants avec leurs notes Avoir des onglets avec les modules (liste moins fournie mais plus exploitable)
Scénario	 Le RP est sur sa page de gestion Il clique sur le bouton de consultation des notes

• Affecter Tuteur à un module

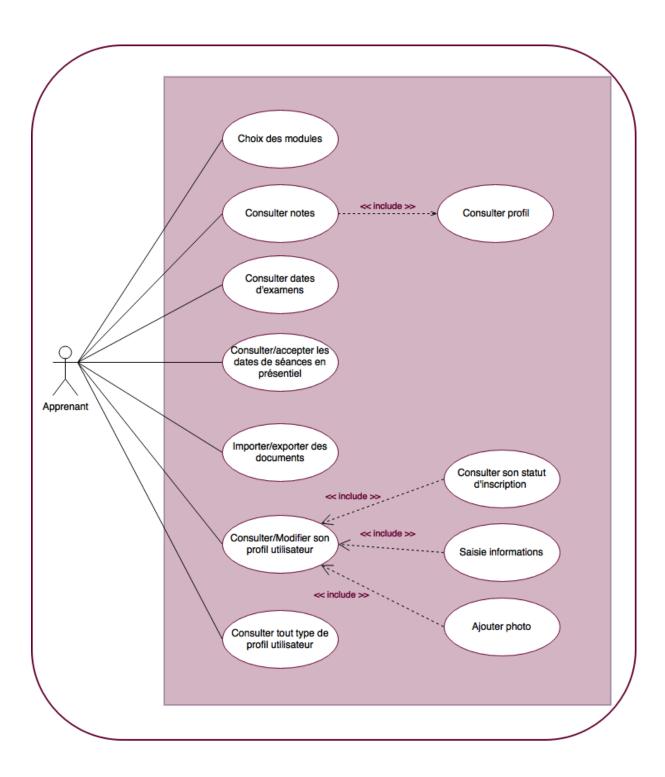


Informations	Description textuelle
Acteurs Concernés	<u>Primaire</u> : Responsable Pédagogique <u>Secondaire</u> : Tuteur
Conditions	Connexion
Post Conditions	 Un tuteur sera affecté à un module Le Responsable Pédagogique ne pourra plus affecter d'autres Tuteur au module
Besoins en Interface Graphique	 Avoir une liste avec tous les Tuteurs disponibles Avoir une liste avec tous les modules n'ayant pas encore de Tuteurs affectés
Scénario	 Le RP¹ est sur sa page de gestion Il clique sur le bouton d'affectation des modules Il lie un tuteur à un apprenant

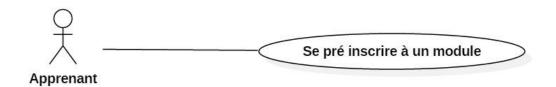
_

¹ Responsable Pédagogique

B. Apprenant

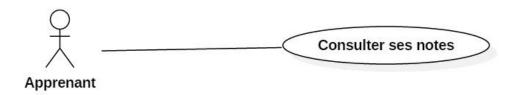


Choisir un module



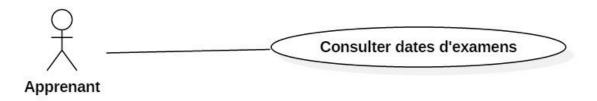
Informations	Description textuelle
Acteurs Concernés	Primaire : Apprenant Secondaire : Responsable Pédagogique
Conditions	
Post Conditions	L'Apprenant sera en attente de réponse
Besoins en Interface Graphique	Liste de tous les modules disponibles Le choix se fera via des checkbox
Scénario	 L'apprenant est sur sa page de gestion Il clique sur un bouton d'inscription aux modules Il choisit les modules Il clique sur un bouton de validation

Consulter ses notes



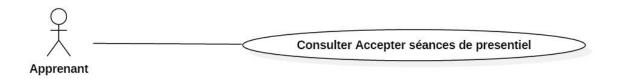
Informations	Description textuelle
Acteurs Concernés	<u>Primaire</u> : Apprenant
Conditions	Connexion
Post Conditions	Néant
Besoins en Interface Graphique	Avoir une liste avec tous les modules
Scénario	 L'apprenant est sur sa page de gestion Il clique sur le bouton de consultation des notes

Consulter dates d'examen



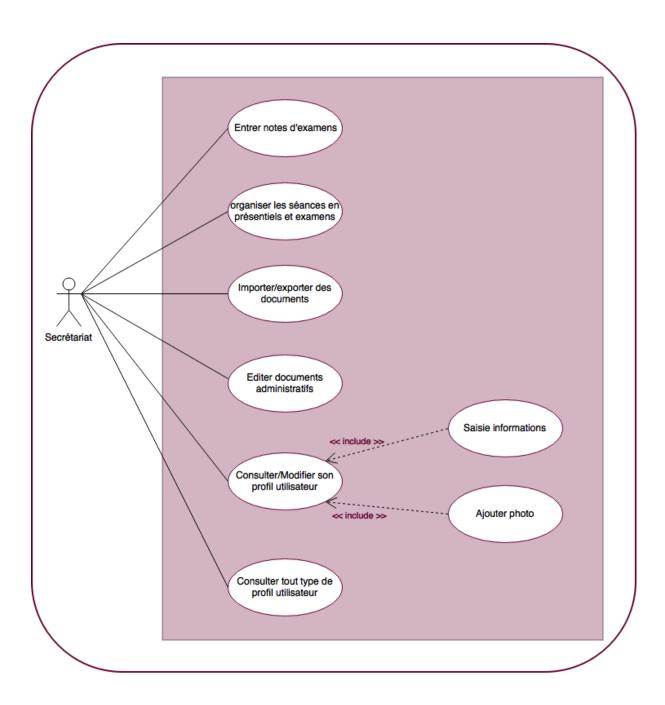
Informations	Description textuelle
Acteurs Concernés	Primaire : Apprenant
Conditions	Connexion Le Secrétariat a rentré les notes
Post Conditions	Néant
Besoins en Interface Graphique	BeToHave : Liste avec les notesNiceToHave : Calendrier
Scénario	 L'apprenant est sur sa page de gestion Il clique sur le bouton de consultation des dates d'examens

• Consulter et accepter dates de présentiel

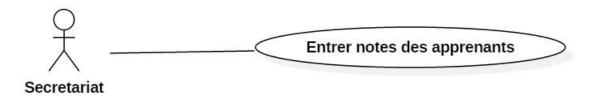


Informations	Description textuelle
Acteurs Concernés	<u>Primaire</u> : Apprenant <u>Secondaire</u> : Tuteur, Responsable Pédagogique
Conditions	Pour qu'une date apparaisse, le tuteur aura entré une date
Post Conditions	Le tuteur sera notifié de l'acceptation/refus de l'alternant
Besoins en Interface Graphique	 Une liste avec tous les présentiels proposés par modules
	1) L'apprenant est sur sa page de gestion
Scénario	2) Il clique sur le bouton de gestion des présentiels
	3) Via la liste, il peut soit accepter, soit refuser

C. Secrétariat



Q Entrer notes d'examens



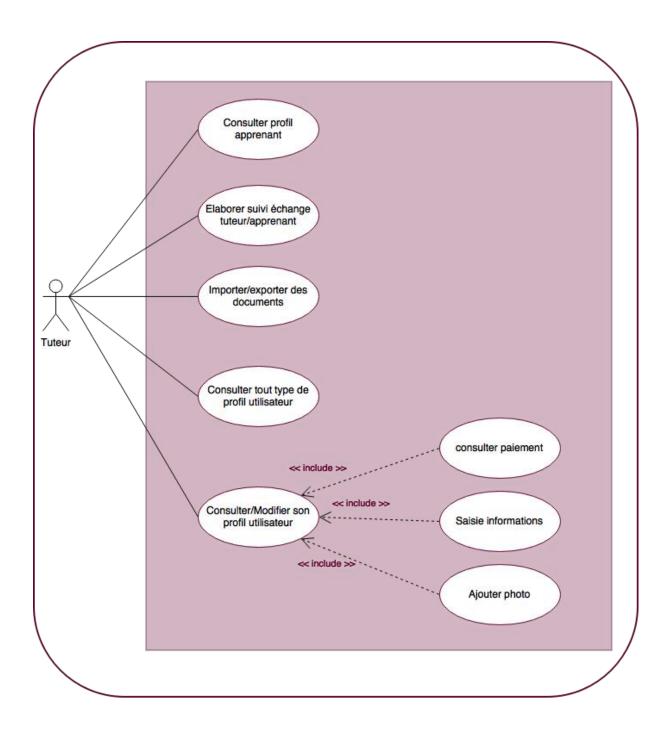
Informations	Description textuelle
Acteurs Concernés	Primaire: Secrétariat
, icicars correctives	Secondaire: Tuteur, apprenant, service formation en continue
Conditions	Le module doit être ouvert pour le semestre en cours
Post Conditions	Une date d'examen est fixée
	 Un bouton " Organiser Examens/Présentiels" sur la page de
Besoins en Interface Graphique	gestion du secrétariat
besoins en interface Grapfilique	 Une liste des modules ouverts pour le semestre
	 Un champ date 'Date d'examen' à renseigner
	1- Le secrétariat accède à sa page de gestion
Scénario	2- Il accède à la liste des modules ouvert en cliquant sur le bouton
SCENATIO	"Organiser Examens/Présentiels"
	3- Il accède au module et rempli le champ date 'Date d'examen'

Organiser séances en présentiel/examens

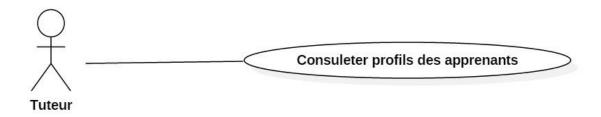


Informations	Description textuelle
Acteurs Concernés	<u>Primaire</u> : Secrétariat <u>Secondaire</u> : Tuteur, apprenant, service formation en continue
Conditions	L'apprenant et le tuteur doivent être liés par un module
Post Conditions	Une date est fixée pour une séance en présentielle
Besoins en Interface Graphique	 Un bouton " Organiser Examens/Présentiels" sur la page de gestion du secrétariat Une liste des modules ouverts pour le semestre Un champ date 'Séance en présentiel' à renseigner
Scénario	1- Le secrétariat accède à sa page de gestion 2- Il accède à la liste des modules ouvert en cliquant sur le bouton "Organiser Examens/Présentiels" 3- Il accède au module et rempli le champ date 'Séance en présentiel'

D. Tuteur



Q Consulter profil des apprenants



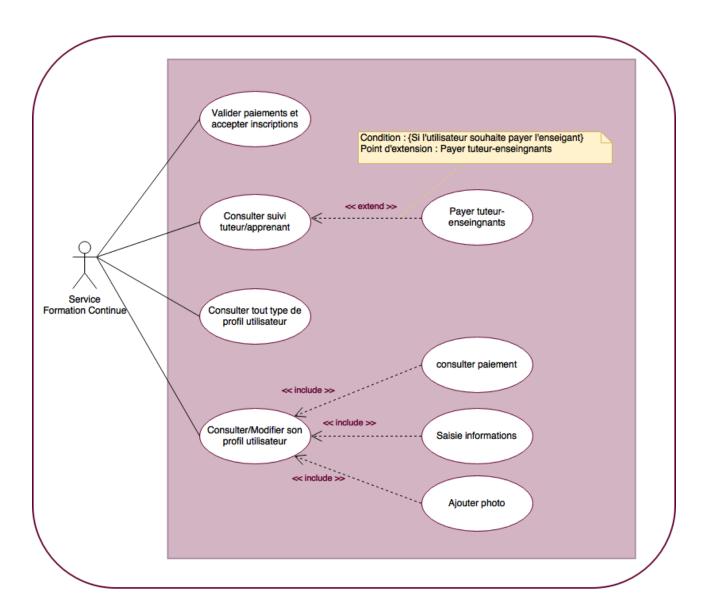
Informations	Description textuelle
Acteurs Concernés	Primaire: Tuteur
Acteurs Concernes	Secondaire: Apprenants
Conditions	Etre le tuteur attribué à l'apprenant
Post Conditions	Néant
Besoins en Interface Graphique	Avoir une liste des apprenants dont il est le tuteur
Caémaria	1- Le tuteur est sur sa page de gestion
Scénario	2- Il se rend sur le suivi des apprenants

Récapituler les échanges avec ses apprenants



Informations	Description textuelle
Acteurs Concernés	Primaire: Tuteur
Conditions	Secondaire: Apprenants Etre le tuteur attribué à l'apprenant
Post Conditions	Néant
Besoins en Interface Graphique	Avoir une liste des apprenants dont il est le tuteur.
Scénario	1- Le tuteur est sur sa page de gestion 2- Il se rend sur le suivi des échanges
Section	3- Il renseigne l'ensemble des échanges ayant eu lieu avec l'apprenant

E. Service Formation en Continue



Valider paiement



Informations	Description textuelle
Acteurs Concernés	<u>Primaire</u> : Service de formation en continue
Acteurs Concernes	Secondaire: Apprenant et Responsable pédagogique
Conditions	Néant
Post Conditions	L'apprenant est inscrit
	Un bouton " Valider Paiement"
Besoins en Interface Graphique	Une liste des apprenants dont l'inscription n'est pas encore validée.
	Un pop-up de confirmation avec un champ date
	1- Le service de formation en continue accède à sa page de
	gestion
Scénario	2- Il clique sur le bouton "Valider paiement"
	3- Il clique sur le nom de l'apprenant et confirme le paiement sur la
	fenêtre pop-up et renseigne également la date de validation.

Q Consulter le suivi tuteur/apprenant



Informations	Description textuelle	
Acteurs Concernés	<u>Primaire</u> : Service de formation en continue <u>Secondaire</u> : Apprenant et Responsable pédagogique	
Conditions	Les tuteurs ont renseigné les apprenants présents en présentiels	
Post Conditions	Le tuteur sera rémunéré	
Besoins en Interface Graphique	Liste avec tous les présentiels effectués	
Scénario	 1- Le service de formation en continue accède à sa page de gestion 2- Il clique sur le bouton « Consulter suivi des présentiels » 3- Il pourra payer un tuteur si la séance de présentiel a bien eu lieu 	



2. DIAGRAMME DE CLASSES

Le diagramme de classes nous permet d'avoir une vue d'ensemble des <u>acteurs</u> mais aussi des <u>objets</u> de notre application.

Par la suite, nous nous baserons sur ce diagramme afin de réaliser notamment notre Modèle Relationnel de Données.

Nous avons <u>9</u> classes différentes, réparties dans <u>2</u> catégories différentes



Catégorie **ACTEURS**

- « User »
- « HeadTeacher »
- « Teacher »
- « Learner »
- « Secretariat »
- « FormationUnit »



Catégorie OBJETS

- « Mark »
- « Module »
- « LiveMeeting »

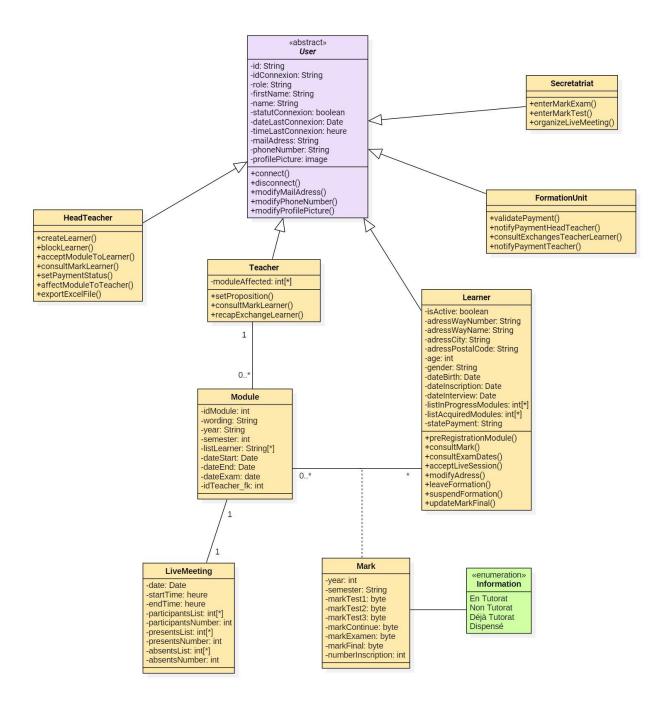
Toutes les classes/attributs/méthodes sont en anglais afin de respecter au mieux les normes de programmation actuelles.

Dans cette partie, chaque classe a une description de ses attributs et ses méthodes

Nous avons volontairement omis de mettre les getters/setter afin de ne pas surcharger le diagramme de classes ainsi que les descriptions associées. Ils seront bel et bien présents dans l'implémentation du code.

Enfin, nous avons mis le caractère « – » lorsque la description ne laisse place à aucune ambiguïté.

A. Diagramme de Classes Général



Nom Anglais	Nom Français
User	Utilisateur
HeadTeacher	Responsable Pédagogique
Teacher	Tuteur
Learner	Apprenant
Secretariat	Secrétariat
FormationUnit	Service Formation Continue
Module	Module
Mark	Note
LiveMeeting	Présentiel

B. Acteurs

i. User (classe abstraite)

«abstract» User -id: String -idConnexion: String -role: String -firstName: String -name: String -statutConnexion: boolean -dateLastConnexion: Date -timeLastConnexion: heure -mailAdress: String -phoneNumber: String -profilePicture: image +connect() +disconnect() +modifyMailAdress() +modifyPhoneNumber() +modifyProfilePicture()

Il s'agit d'une classe abstraite. Son rôle est d'assurer une meilleure architecture du code, tout en factorisant le maximum d'attributs et de méthodes entre les différents types d'utilisateurs.

	АТТ	RIBUTS	
Nom	Type	Visibilité	Description
id	String		Identifiant de l'utilisateur Il est unique
idConnexion	String		Identifiant de connexion de l'utilisateur
role	String		-
firstName	String		Prénom
name	String	private	Nom de famille
statutConnexion	boolean		Permet de savoir si l'utilisateur est connecté Si oui = true Sinon = false
dateLastConnexion	Date		Permet la traçabilité
timeLastConnexion	heure		Permet la traçabilité
mailAdress	String		Adresse mail
phoneNumber	String		Numéro de téléphone

profilePicture	image	Photo de profil

	М	ETHODES	
Nom	Type de Retour	Visibilité	Description
connect()	void	public	Connexion statutConnexion passe à true
disconnect()	void		Déconnexion statutConnexion passe à false
modifyMailNumber()	void		Modifier Adresse Mail
modifyPhoneNumber()	void		Modifier numéro de téléphone
modifyProfilePicture()	void		Modifier photo de profil

ii. HeadTeacher

+createLearner() +blockLearner() +acceptModuleToLearner() +consultMarkLearner() +setPaymentStatus() +affectModuleToTeacher() +exportExcelFile()

	АТ	TRIBUTS	
Nom Type Visibilité Description			
Néant			

	М	ETHODES	
Nom	Type de Retour	Visibilité	Description
createLearner()	void	public	Crée un apprenant au sein de l'application Un apprenant est créé avec son nom, prénom, date d'entretien, un identifiant et son mot de passe
blockLearner()	void		Bloque un apprenant de l'appli s'il

		n'a pas payé ses droits
		Permet aussi de débloquer une fois que l'apprenant aura payé ses droits
acceptModuleToLearner()	void	Valide/Refuse les modules présélectionnés par les Apprenants
consult Mark Learner ()	void	Consulter les notes de tous les apprenants présents dans l'application
setPaymentStatus()	void	Passe le statut du paiement de l'apprenant à true/false
affectModuleToTeacher()	void	Affecter un module à un Tuteur
exportExcelFile()	void	Permet d'exporter sous format Excel des données depuis l'application: Notes des apprenants Echanges Apprenants-Tuteur Dernière connexions des acteurs de l'application



Teacher
-moduleAffected: int[*]
+setProposition() +consultMarkLearner() +recapExchangeLearner()

ATTRIBUTS					
Nom	Туре	Visibilité	Description		
moduleAffected	int	private	Identifiant du ModuleTutore rattaché		

METHODES						
Nom	Type de Retour	Visibilité	Description			
setProprosition()	void	public	Accepte ou refuse la proposition du RP pour l'affectation du module			
consult Mark Learner ()	void		Affiche toutes les notes des Apprenants <u>lié à son module</u>			
recapExchangeLearner()	void		Récapitule tous ses échanges avec			

Projet Plateforme de commu	Projet Plateforme de communication Master e-Miage						
			les Apprenants				

iv. Learner

Learner

-isActive: boolean

-adressWayNumber: String

-adressWayName: String

-adressCity: String

-adressPostalCode: String

-age: int

-gender: String

-dateBirth: Date

-dateInscription: Date

-dateInterview: Date

-listInProgressModules: int[*]
-listAcquiredModules: int[*]

-statePayment: String

+preRegistrationModule()

+consultMark()

+consultExamDates()

+acceptLiveSession()

+modifyAdress()

+leaveFormation()

+suspendFormation()

+updateMarkFinal()

ATTRIBUTS				
Nom	Туре	Visibilité	Description	
isActive	boolean		Savoir si l'apprenant est inscrit à des modules <u>pour le</u> <u>semestre en cours</u> Si oui = true Sinon = false	
dateInterview	Date		Date de l'entretien avec le RP Il est entré par le RP	
adressWayNumber	String		Numéro de voie de l'adresse de l'apprenant	
adressNameWay	String		Nom de voie de l'adresse de l'apprenant	
adressCity	String	private	Ville de l'adresse de l'apprenant	
adressPostalCode	int	,	Code postal de l'apprenant	
birthDate	Date		Date de naissance	
age	int		Age de l'apprenant. Il est calculé automatiquement via l'attribut dateNaissance	
gender	String		Sexe de l'apprenant 2 valeurs possibles : « Masculin » ou « Féminin »	
listInProgressModules	<int></int>		Liste des modules en cours Pour l'affichage, la méthode fera le lien avec le libellé du	

		module via une table dédiée
listAcquiredModules	<int></int>	Liste des modules acquis
		Etat du paiement pour le semestre en cours
statePayment	string	3 valeurs possibles :
dateInscription	Date	Date à laquelle l'apprenant s'est inscrit sur la plateforme

METHODES					
Nom	Type de Retour	Visibilité	Description		
preRegistrationModule()	void		Se préinscrire à des modules		
consultMark()	void		Consulter ses notes		
consultExamDates()	void		Consulter les dates d'examens		
acceptLiveSession()	void	public	Accepter un présentiel		
modifyAdress()	void		Modifier adresse		
leaveFormation()	void		Quitter la formation		
suspendFormation()	void		Suspendre la formation		
updateMarkFinal()	void		Actualise la note finale Calculée selon le barème suivant : (noteExamen*2 + noteContinueMoyenne) / 3		



Secretatriat +enterMarkExam() +enterMarkTest() +organizeLiveMeeting()

ATTRIBUTS					
Nom Type Visibilité Description					
Néant					

METHODES					
Nom	Type de Retour	Visibilité	Description		
enterMarkExam()	void		Entrer les notes d'examens		
enterMarkTest()	void	public	Entrer les notes de contrôles continus		
organizeLiveMeeting()	void				

FormationUnit

FormationUnit			
+validatePayment() +notifyPaymentHeadTeacher()			
+consultExchangesTeacherLearner() +notifyPaymentTeacher()			

ATTRIBUTS					
Nom Type Visibilité Description					
Néant					

METHODES				
Nom	Type de Retour	Visibilité	Description	
validatePayment()	void		Valider administrativement le paiement des droits pour un apprenant	
notifyPaymentHeadTeacher()	void	public	Notifier que l'apprenant a payé ses droits	
consultExchangesTeacherLearner()	void	public	Consulter les échanges effectués entre les Apprenants et les Tuteurs	
notifyPaymentTeacher()	void		Notifie au tuteur qu'il va être rémunéré	

C. Objets

i.

Module

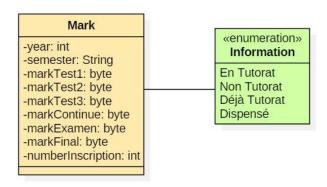
Module

-idModule: int
-wording: String
-year: String
-semester: int
-listLearner: String[*]
-dateStart: Date
-dateEnd: Date
-dateExam: date
-idTeacher_fk: int

ATTRIBUTS					
Nom	Туре	Visibilité	Description		
idModule	Int		Identifiant du module		
wording	String		Libellé		
year	Date		Année		
semester	String		Semestre d'activité du module. Il peut prendre 2 valeurs : « \$1 » ou « \$2 »		
description	String	private	Donne une description brève du module Utile quand l'apprenant se préinscrit à des modules pour la première fois		
listLearner	String		Liste des apprenants affectés à ce module		
dateStart	Date		Date de début du module		
dateEnd	Date		Date de fin du module		
dateExam	Date		Date d'examen du module		

METHODES						
Nom	Nom Type de Retour Visibilité Description					
Néant						

ii. Mark (classe association)



Les notes étant sur 20, nous avons utilisons des types byte (qui ont une plage de valeurs réduite) afin d'économiser de l'espace mémoire.

ATTRIBUTS					
Nom	Туре	Visibilité	Description		
annee	int		Année de passage du module par l'apprenant		
semester	String		Semestre de passage du module. Il peut prendre 2 valeurs : « S1 » ou « S2 »		
markDevoir1	byte				
markDevoir2	byte		Notes entrées par le Tuteur Elles sont facultatives		
markDevoir3	byte	n ni sata			
markContinue	byte	private	Note de contrôle continue continue		
markExam	byte		Note de l'examen Elle est entrée par le Secrétariat		
markFinal	byte		La note finale Calculée automatiquement via la méthode updateMarkFinal() de Learner		
numberInscription	int		Nombre d'inscription de l'apprenant au module lié		

METHODES					
Nom Type de Retour Visibilité Description					
Néant					

iii. LiveMeeting

LiveMeeting

-date: Date
-startTime: heure
-endTime: heure
-participantsList: int[*]
-participantsNumber: int
-presentsList: int[*]
-presentsNumber: int
-absentsList: int[*]
-absentsNumber: int

ATTRIBUTS						
Nom	Туре	Visibilité	Description			
date	Date		Date à laquelle a eu lieu le présentiel Valeur entrée par le Secrétariat			
startTime	heure		Heure de début du présentiel			
endTime	heure		Heure de fin du présentiel			
participantsList	<int></int>		Liste des identifiants des apprenants qui participent au présentiel			
participants Number	int	private	Actualisé automatiquement via la taille de listeParticipants			
presentsList	<int></int>	,	Liste des identifiants des apprenants qui étaient présents au présentiel Cette liste est actualisée via le tuteur			
presentsNumber	int		Actualisé automatiquement via la taille de listeParticipants			
absentsList	int		-			
absentsNumber	<int></int>		Actualisé automatiquement via la taille de listeAbsents			

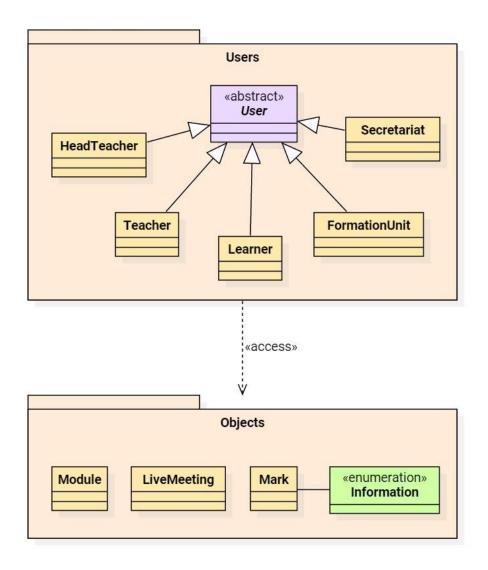
METHODES						
Nom	Type de Retour	Visibilité	Description			
Néant						



3. DIAGRAMME DE PACKAGES

Comme vue dans le Diagramme de Classes, 2 catégories se sont clairement détachées. D'un côté, nous avons les Utilisateurs de notre plateforme, et d'un autre côté les Objets utilisés par ceux-ci. On peut ainsi établir un Diagramme de Packages.

A. Diagramme de Packages





4. DIAGRAMMES DE SEQUENCES

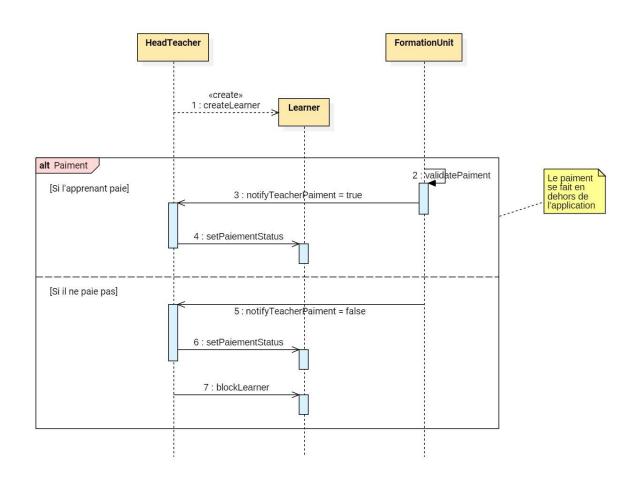
Afin de visualiser au mieux les interactions entre les différents acteurs de l'application, le Diagramme de Séquences s'imposait naturellement.

Nous pourrons de plus cibler au mieux les impacts des différentes méthodes sur les différents acteurs.

Dans cette partie, tous les diagrammes de séquences sont accompagnés d'une description textuelle.

NB: les numéros des descriptions ne correspondent pas forcement avec les numéros sur les diagrammes.

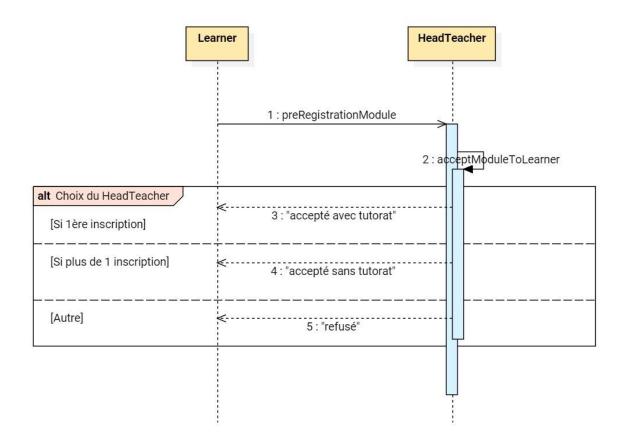
A. Première Inscription et Paiement



Projet Plateforme de communication Master e-Miage

- 1. Le Responsable Pédagogique va créer un Apprenant
- 2. La Formation Continue va valider ou non le paiement
- 3. La Formation Continue va notifier le Responsable Pédagogique du paiement
- 4. Le Responsable Pédagogique va paramétrer le statut de paiement de l'Apprenant S'il n'a pas payé, il sera bloqué

B. Préinscription à un module

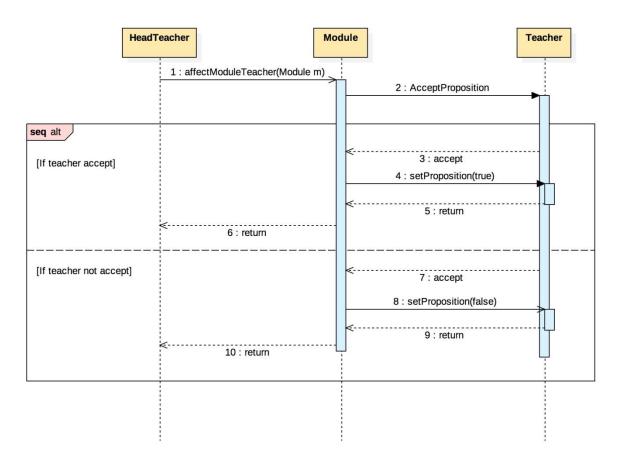


- 1. L'Apprenant se pré inscrit à un module
- 2. Le Responsable Pédagogique fait un choix pour cette préinscription

Plusieurs cas possibles:

- 3. Il <u>accepte avec tutorat</u> si c'est la 1^{ère} inscription à un module
- 4. Il <u>accepte mais sans tutorat</u> si l'Apprenant était déjà inscrit à ce module auparavant
- 5. Il refuse

C. Affectation d'un module à un tuteur

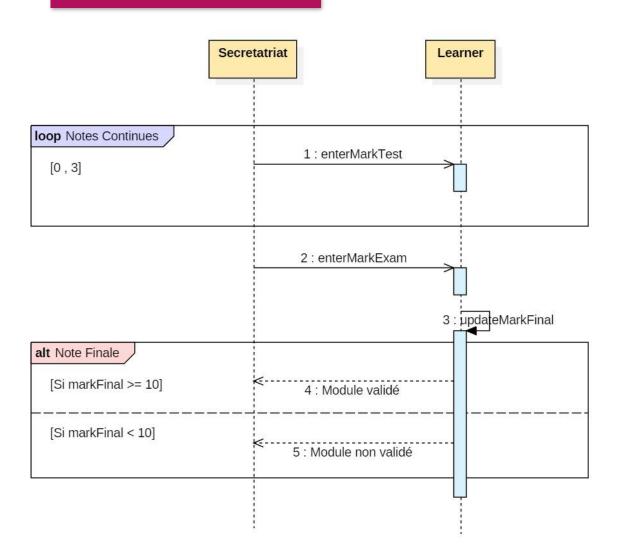


1. Le Responsable Pédagogique affecte un module à un Tuteur

Deux cas possibles:

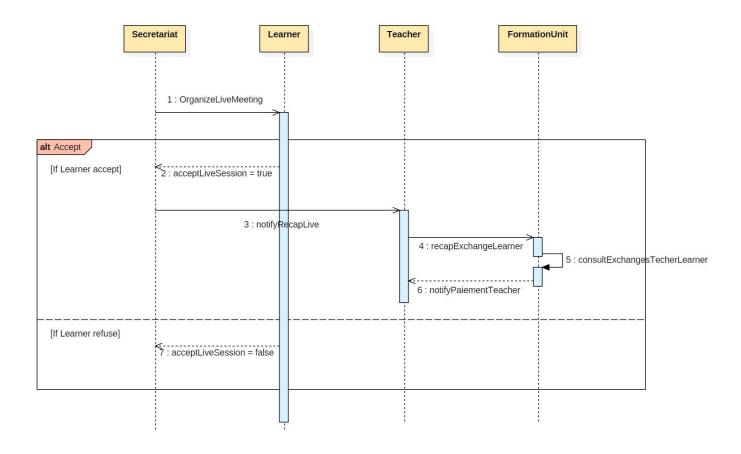
- 2. Le tuteur accepte
- 3. Le tuteur refuse

D. Validation d'un module



- 1. Le Secrétariat entre les notes de contrôles continues (jusqu'à 3)
- 2. Le Secrétariat entre la note d'examen
- 3. La note de l'Apprenant est actualisée
- 4. S'il a 10 ou plus, le module sera validé
- 5. S'il a moins de 10, le module ne sera pas validé

E. Présentiel et Paiement



1. Le Secrétariat va organiser une séance de Présentiel

Deux cas possibles:

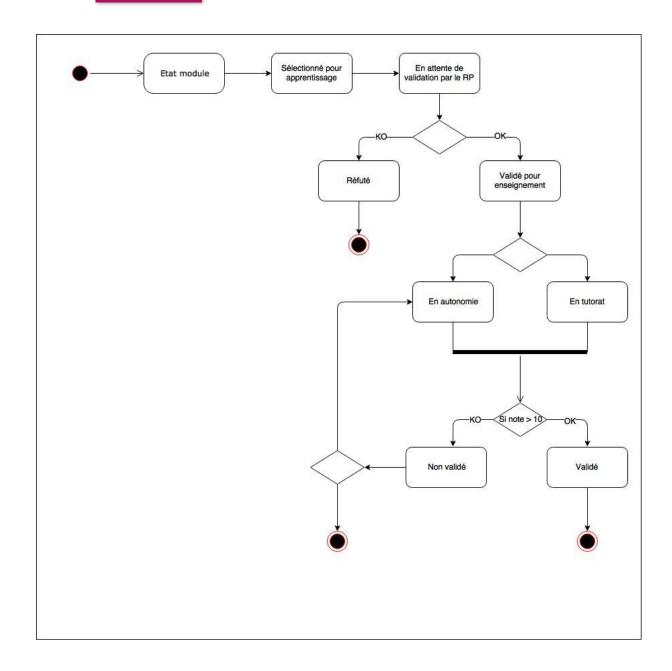
- 2. L'Apprenant va accepter. Dans ce cas le tuteur va recevoir une notification Le Tuteur va pouvoir échanger par la suite avec la Formation Continue pour son paiement
- 3. L'Apprenant refuse cette date de Présentiel



5. DIAGRAMMES D'ETATS

Un Diagramme d'Etat est un type de diagramme comportemental, qui consiste à montrer des transitions et/ou changement d'état d'un objet en particulier.

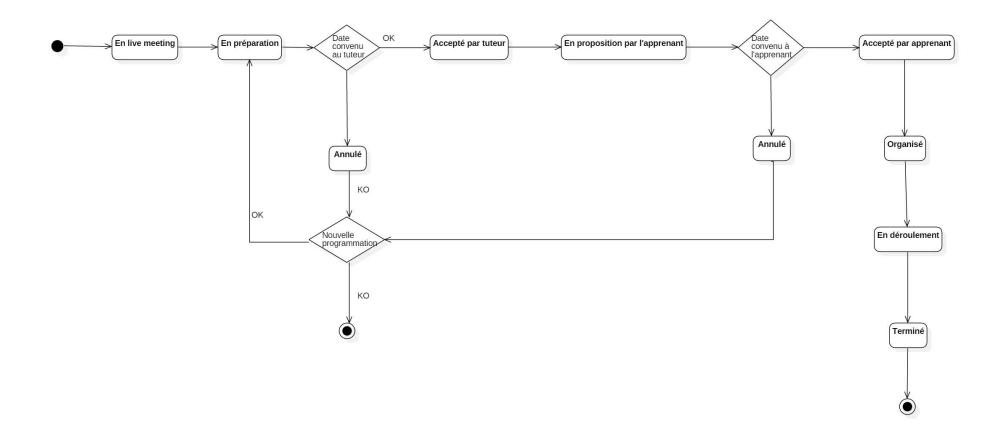
A. Module



Le diagramme suivant représente les différents états de l'objet Module. Initialement le module va être sélectionné par l'apprenant et par la suite validé ou non par le responsable pédagogique.

Ensuite, en fonction de son nouvel état, l'objet à la possibilité de transiter soit en tant que module « en autonomie » soit en tant que « en tutorat », le but étant de savoir par quel moyen le module est étudié. Par la suite, une fois le module étudié, après examen, une condition est mise en place pour effectuer une vérification de la validation de celui-ci. Si le module est validé l'état de l'objet transite en tant que « validé » sinon il transite comme « non validé ».

B. Live Meeting



Rappelons qu'un Apprenant a le droit d'obtenir une séance en Présentiel avec le Tuteur enseignant, afin de résoudre ses interrogations sur certains points qu'il n'aurait pas assimilé. Ces séances, appelées Live Meeting, sont organisées par le Secrétariat (En préparation) et sont ensuite proposé au Tuteur-enseignant et Apprenant pour obtenir leur accord ou non. Après ces conditions validées l'objet va transiter vers trois états différents indiquant que Live Meeting a bien été organisé (horaire et salle confirmé), eu lieu et terminé.

III. Annexes

A. Tableau de Bord Individuel

N.B : Seules les séances de cours sont répertoriées

N° Séance	Date	Elément	Tâche(s) effectuée(s)	Difficulté(s)	
1	1 26/10/2017	Kenza	Composition des équipes et choix du projet	-	
		Hakan	p. 3,01		
			Prise de connaissance le projet et les besoins utilisateurs	Expressions des Besoins	
2	2 09/11/2017	Kenza	Questions/Réponses avec Mme Benzakki	Rôle des différents utilisateurs Besoins Fonctionnels à	
		Hakan	Réparation des tâches	déterminer	
3	23/11/2017	Sarunyaa Kenza Hakan	Questions/Réponses avec Mme Benzakki	-	
4	30/11/2017	Sarunyaa Kenza Hakan	Questions/Réponses avec Mme Benzakki	-	
5	07/12/2017	Sarunyaa Kenza Hakan	Initialisation de Uses Cases Responsable Pédagogique, Apprenant et Secretariat	-	
6	14/12/2017	Sarunyaa Kenza Hakan	Initialisation des Uses Cases Tuteur et Service Formation Continue	-	
7	21/12/2017	Sarunyaa Kenza Hakan	Finalisation des Diagrammes Uses Cases	-	
		Sarunyaa		<u>Diagrammes de Classes</u> Unification des classes	
8 11/01/2017	11/01/2017	Kenza	Premier bilan sur le Diagramme de Classes	ModuleTutoré et ModuleNonTutoré Passage de tous les classes/attributs en anglais	
		Hakan			
		Sarunyaa	Finalisation du Diagramme de Classes	<u>Diagramme de Classes</u> Enumération « Infos » à placer	
9	25/01/2017	Kenza Hakan	Répartition des tâches pour les Diagrammes de Séquences et d'Etats		

B. Répartition des tâches

	Tâches	Sarunyaa	Kenza	Hakan
CONTEXTE				0
EXPRESSION DES BESOINS	Recueil des Besoins	0	0	0
CONCEPTION	Diagrammes de Cas d'Utilisations	0	0	0
	Diagrammes de Classes			0
	Diagrammes de Séquences	0	0	0
	Diagrammes d'Etats	0	0	
	Diagramme de Packages	0	0	
CLÔTURE	Mise en page du rapport			0