

PROJET DE FIN DE FORMATION EN JAVA

DOSSIER D'EXPRESSION DU BESOIN

CONCEPTION ET DEVELOPPEMENT D'UNE APPLICATION JAVA POUR LE
SUIVI DES COURS ET L'ETABLISSEMENT DES ETATS DE PAIEMENT.

Formateur :
Ghislain AKINOCHO

Page
1/6

Table des matières

1.	Introduction	2
1.1.	Objectif du projet	2
1.2.	Thème proposé et Enjeux.....	2
1.3.	La notation.....	4
1.4.	La documentation.....	4
1.5.	Délai et présentation de l'application	4
2.	Spécifications techniques	5
2.1.	Exigences graphiques	5
3.	La procédure de réalisation	5

<p>Ecole Supérieure Multinationale des Télécommunications</p>  <p>ESMT Ecole Supérieure Multinationale des Télécommunications</p>	<p>PROJET DE FIN DE FORMATION EN JAVA</p> <p>DOSSIER D'EXPRESSION DU BESOIN</p> <p>CONCEPTION ET DEVELOPPEMENT D'UNE APPLICATION JAVA POUR LE SUIVI DES COURS ET L'ETABLISSEMENT DES ETATS DE PAIEMENT.</p>	<p>Formateur : Ghislain AKINOCHO</p>	<p>Page 2/6</p>
--	---	---	---------------------

1. Introduction

1.1. Objectif du projet

Les objectifs de ce projet sont les suivants :

1. Approfondir les compétences acquises tout le long du module « **Programmation Orientée Objet en Java** » à l'aide de la programmation d'une application Java légèrement complexe.
 - a. Mettre en œuvre les concepts de programmation Java largement présentés et discutés en cours et en travaux pratiques dans un contexte réel.
 - b. Exploiter et mettre en évidence la puissance de la POO (*Héritage, Polymorphisme, Encapsulation, ...*).
 - c. Accéder aux bases de données relationnelles.
 - d. Développer des interfaces graphiques ergonomiques.
2. Favoriser enfin le travail d'équipe. A ce sujet, ce projet devra donc être en groupe de 3 ou 4 au maximum.

1.2. Thème proposé et Enjeux

Le thème proposé est le « **développement d'une application de suivi des cours** ».

Il s'agira de développer une application capable d'effectuer le suivi des cours dispensés par les enseignants et de sortir différentes statistiques qui seront utiles pour l'établissement des états de paiement entre autres.

Ci-dessous les opérations possibles par acteur sur le système :

- **Administrateur** :
 - Gestion des comptes utilisateur du système (création de comptes utilisateur, suppression, mise à jour, lecture)
- **Assistant de programme** :
 - Il est chargé de saisir les emplois du temps des enseignants dans le système. La saisie des emplois du temps consiste à créer un cours à une certaine période de la journée.
 - Il peut visualiser les emplois du temps de tous les enseignants et les modifier.
 - Il peut clôturer un module, établir l'état de l'enseignant pour un module dans une classe et demander le déclenchement du processus de paiement. L'état présente tous les cours dispensés (jour, contenu, heure, durée, ...)
 - Il peut visualiser les états de tous les enseignants.
 - Il peut modifier ses informations personnelles
 - Il peut créer un module et y associer un enseignant
 - Il peut créer une classe et y ajouter un enseignant

<p>Ecole Supérieure Multinationale des Télécommunications</p>  <p>ESMT Ecole Supérieure Multinationale des Télécommunications</p>	<p>PROJET DE FIN DE FORMATION EN JAVA</p> <p>DOSSIER D'EXPRESSION DU BESOIN</p> <p>CONCEPTION ET DEVELOPPEMENT D'UNE APPLICATION JAVA POUR LE SUIVI DES COURS ET L'ETABLISSEMENT DES ETATS DE PAIEMENT.</p>	<p><u>Formateur :</u> Ghislain AKINOCHO</p>	<p>Page 3/6</p>
--	---	--	---------------------

- **Chef de classe :**

- Le chef de classe peut modifier un cours à la fin de la session en définissant l'heure effective de début, celle de fin et en définissant le contenu dispensé.
- Le chef de classe ne peut visualiser et modifier que les cours de sa classe

- **Enseignant :**

- L'enseignant peut valider ou rejeter les modifications effectuées sur le cours par le chef de classe. Dans ce dernier cas, il ajoute un commentaire pour que le chef de classe les prenne en compte. Une fois le consensus trouvé, le cours dispensé est validé dans le système sinon il reste à l'état non dispensé.
- Il peut visualiser la situation d'un cours (les séances réalisées, le nombre d'heures restant, etc ...)
- Il peut visualiser les emplois du temps passés, l'emploi du temps de la période en cours et s'il est disponible celui de la période suivante ...
- Il peut consulter la situation les états de paiement pour les modules terminés (dont tous les cours ont été dispensés et validés). L'état de paiement contient pour chaque module le montant total à payer à l'enseignant (nombre d'heures dispensés effectivement multiplié par le coût horaire (**configurable dans le système**)).
- L'enseignant peut demander une annulation ou un report d'un cours programmé dans la période en cours ou dans celle à venir.

- **Responsable pédagogique :**

- Le responsable pédagogique peut visualiser les emplois du temps et les états de cours et de paiements de tous les enseignants.
- Il valide les états de paiements pour qu'ils soient transmis au comptable qui procédera au paiement.
- Il peut demander des informations supplémentaires au chargé des programmes avant la validation. Les échanges se feront dans le système et seront visualisables.

- **Comptable :**

- Le comptable ne peut consulter que les états de paiements validés par le responsable pédagogique.
- Il procède au paiement des états en établissement le chèque puis en le mentionnant dans le système.

<p>Ecole Supérieure Multinationale des Télécommunications</p> 	<p>PROJET DE FIN DE FORMATION EN JAVA</p> <p>DOSSIER D'EXPRESSION DU BESOIN</p> <p>CONCEPTION ET DEVELOPPEMENT D'UNE APPLICATION JAVA POUR LE SUIVI DES COURS ET L'ETABLISSEMENT DES ETATS DE PAIEMENT.</p>	<p>Formateur : Ghislain AKINCHO</p>	<p>Page 4/6</p>
---	---	--	---------------------

1.3. La notation

Le projet sera noté (sur /20) et constituera l'examen final du module. Les éléments qui entreront en ligne de compte dans l'évaluation du projet sont les suivants :

1. La documentation (Cf. §1.4) (3)
2. La fonctionnalité du programme (toutes les fonctionnalités techniques et métier devront être correctement implémentés conformément au besoin exprimé.) (5)
3. Le design et l'ergonomie de l'application (simplicité et convivialité dans l'exploitation de l'application sont attendues) (4)
4. La qualité du code (respect des conventions, paquetages, application des fondements de la POO, etc.) (5)
5. La modélisation (qualité des diagrammes UML reflétant le besoin exprimé) (3)

1.4. La documentation

La documentation doit contenir le cahier de charge, la gestion du projet (analyse, intervenants et répartition des tâches, etc ...), la structuration du programme (*diagrammes UML via un outil de modélisation standard*), les problèmes rencontrés et les solutions, les technologies utilisées etc ... Le journal de travail, les fiches de tests de validation basé sur les cas d'utilisations identifiés devront être fournis en annexe.

Mettre à disposition le code source du programme et tous les Pré-requis à son exploitation et à son exécution dans un environnement de développement. Le rapport sans les annexes ne devrait pas contenir plus de **10 pages**.

1.5. Délai et présentation de l'application

Le délai pour la remise du projet est de **1 mois** à compter de la date de réception de ce présent document soit le **Vendredi 25 Février 2022 à minuit**.

La présentation de l'application se fera en présence de toute la classe et sous format PowerPoint (**10 slides au maximum**) pour une **durée maximale de 10 min** durant laquelle le travail et sa conception (tous les diagrammes UML demandés) seront exposés. Prévoir une brève démo de l'application présentant les fonctionnalités principales. Les débordements auront un impact négatif sur la note finale.

<p>Ecole Supérieure Multinationale des Télécommunications</p>  <p>ESMT Ecole Supérieure Multinationale des Télécommunications</p>	<p>PROJET DE FIN DE FORMATION EN JAVA</p> <p>DOSSIER D'EXPRESSION DU BESOIN</p> <p>CONCEPTION ET DEVELOPPEMENT D'UNE APPLICATION JAVA POUR LE SUIVI DES COURS ET L'ETABLISSEMENT DES ETATS DE PAIEMENT.</p>	<p><u>Formateur :</u> Ghislain AKINOCHO</p>	<p>Page 5/6</p>
--	---	--	---------------------

2. Spécifications techniques

2.1. Exigences graphiques

Les contraintes sur les interfaces graphiques sont claires :

- 1- Il est interdit de lancer plusieurs fonctionnalités et/ou plusieurs fenêtres en même temps (une à la fois). Les boîtes de dialogues n'étant pas concernées.
- 2- Les fenêtres doivent être dimensionnées de façon à être adaptées au contenu et non redimensionnables.
- 3- La fermeture des fenêtres se fait uniquement via un bouton « Fermer » et/ou « Annuler ».
- 4- Toutes les fenêtres doivent partager une bannière commune qui prouve leur appartenance à l'application.
- 5- Toutes les exceptions doivent faire l'objet d'un message dans une boîte de dialogue appropriée.
- 6- Garantir une interaction simplifiée et ergonomique avec les acteurs principaux du système (opération réussie/Echec de l'opération : *raison*/etc ...). Lorsqu'une opération est terminée, revenir à la fenêtre principale.

3. La procédure de réalisation

- a. Spécifier la partie gestion de projet (organisation du travail au sein de l'équipe, ...)
- b. Réaliser le design du programme (diagrammes UML)
 - Faire le diagramme de classes
 - Faire le diagramme des cas d'utilisation
 - Faire le diagramme de séquences des fonctionnalités principales
 - Faire le diagramme d'état-transitions pour les objets **cours** et **paiements**.
 - Faire le diagramme de composants
- c. Définir le code du programme
- d. Tester le code.
- e. Constituer un journal de travail, dans lequel vous évoquez les connaissances acquises ou consolidées, les problèmes rencontrés ainsi que les solutions pour ces problèmes.
- f. La documentation ne devrait pas être écrite à la fin du projet, mais en parallèle.
- g. Les extensions (fonctionnalités ou traitements non couverts par le besoin) sont souhaitées mais pas obligatoires.

- BON COURAGE -