

## Projet de Développement d'applications Web Technologies PHP et Java

### Thème : Gestion de la Mobilité des Etudiants

Le but du projet est la place d'une application de gestion de la mobilité des étudiants au sein d'une association d'universités. L'application intégrera les données des universités partenaires impliquées dans les programmes d'échanges à la disposition des étudiants.

Vous trouverez en annexe le schéma conceptuel et le modèle relationnel de la base de données sur laquelle s'appuiera l'application à développer.

La description liée à la modélisation est la suivante :

Les universités offrent des diplômes qui sont composés de cours. Les étudiants sont inscrits à un seul diplôme dans leur université d'origine. Les universités proposent des programmes d'échanges d'étudiants. Les programmes concernent deux diplômes de niveaux équivalents provenant de deux universités différentes et pour une durée de plusieurs mois. Les étudiants s'inscrivent à l'application et peuvent, ensuite, effectuer des demandes de mobilités pour suivre des enseignements dans l'université d'accueil du programme concerné. Lors de la demande de mobilité, l'étudiant choisi les cours (compatibles avec les cours de son diplôme initial) qu'il suivra dans l'université d'accueil. Si sa demande est acceptée, un contrat de mobilité sera généré et l'étudiant pourra solliciter auprès de son université d'origine une demande de financement (montant de la bourse fixé par l'université d'origine, financement attribué « au mérite »).

Lorsqu'un étudiant effectue une demande de mobilité, son état est « en cours ». Le conseil de l'université examine la demande et lui confère le statut « refusée » ou « acceptée ». Si la demande est « acceptée », l'étudiant doit la valider (passage à l'état « validée ») pour finaliser la demande. Lors de la validation, un contrat est généré avec le statut « à réaliser ». Ce statut passera à « en cours » lors du déplacement de l'étudiant, puis, lors de son retour, à « réalisé ».

Un étudiant peut demander une bourse de mobilité pour son contrat. Le montant de la bourse est fixe et elle sera attribuée par le conseil de l'université d'origine en fonction des résultats de l'étudiant. Lors de sa création par l'étudiant, la bourse a le statut « cours » puis elle prendra les statuts « refusée » ou « acceptée » en fonction de la décision du conseil.

L'application comportera deux volets :

- un volet administrateur permettant la gestion des données relatives aux universités, diplômes, programmes, etc.
- un volet utilisateur permettant aux étudiants de consulter les programmes d'échanges et de faire les demandes de mobilité ou de financement.

**La partie administrateur sera réalisée en PHP tandis que la partie utilisateur sera réalisée en JSP.**

### Volet administrateur (PHP)

Cette partie permettra la gestion des diplômes, des cours et des programmes, la modification des statuts des demandes de mobilité et de financement en fonction des retours des conseils, et la mise à jour des états des contrats.

#### Gestion des données

- Création, suppression, modification des nouveaux diplômes et cours
- Création, suppression, modification des programmes
- Consultation des diplômes par université
- Consultation des cours par diplôme
- Consultation des programmes

#### Gestion des demandes et contrats

- Consultation et mise à jour des demandes de mobilités
- Consultation et mise à jour des demandes de financement
- Consultation et mise à jour des contrats

### Volet utilisateur (JSP)

Cette partie est l'interface de l'étudiant. On y trouvera l'inscription de l'étudiant au dispositif et sa connexion à l'application. Une fois connecté, il pourra faire des demandes de mobilité ou consulter l'état de ses demandes en cours.

- Inscription/connexion de l'étudiant (identifiant = email de l'étudiant)
- Consultation les programmes d'échanges impliquant l'université de l'étudiant connecté et compatibles avec son niveau de diplôme (avec une liste des cours de l'université d'accueil)
- Création et consultation de ses demandes de mobilité, validation des demandes acceptées :
  - Attention à proposer des cours compatibles avec ceux du diplôme d'origine de l'étudiant
  - Attention, la validation d'une demande génère un contrat
- Consultation des contrats de l'étudiant
- Création et consultation des demandes financières (liées à un contrat)

## Consignes

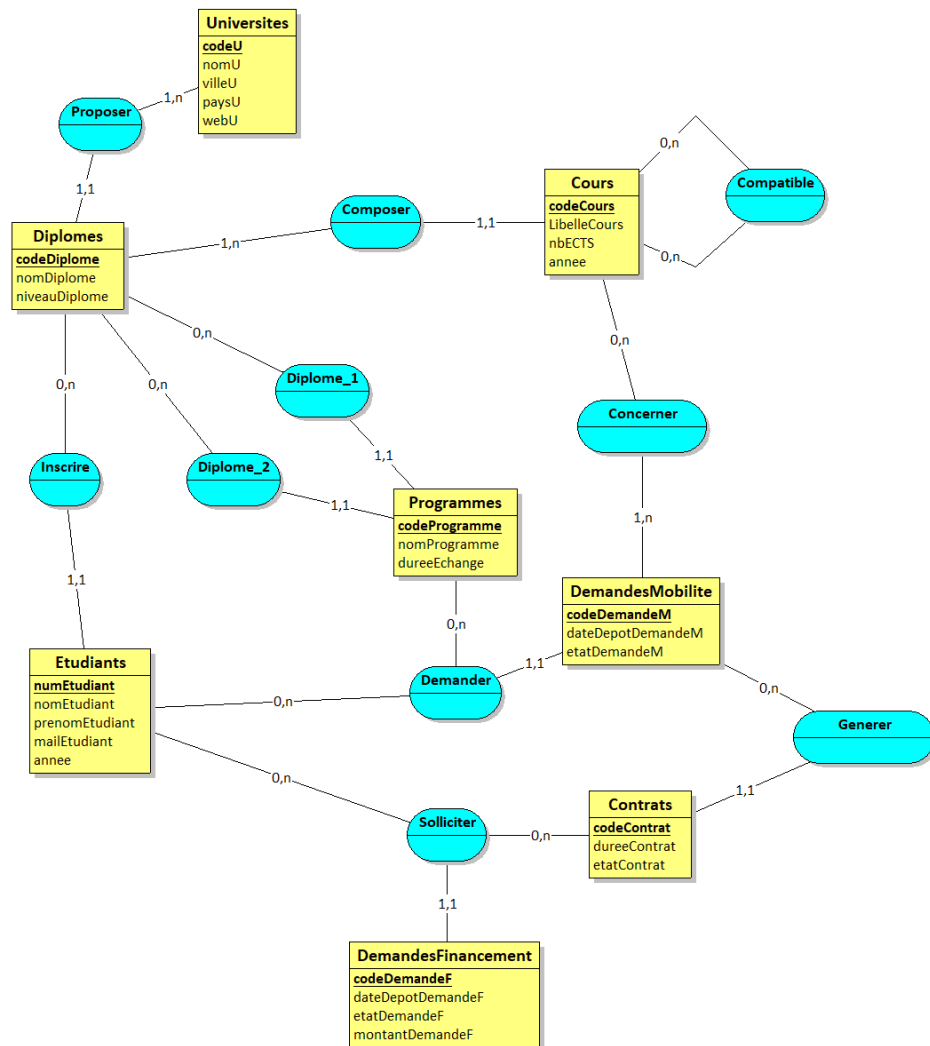
Ce travail est à réaliser par groupe de 3 à 4 étudiants. Il sera déposé sur Moodle au plus tard le **vendredi 26 avril** à minuit sous la forme d'une archive (.zip exclusivement) contenant les deux parties de l'application séparées dans deux dossiers différents.

A ces dossiers sera ajouté un rapport décrivant les architectures MVC des volets administrateur et utilisateur. Pour chacune de ces parties, vous décrierez aussi les différents vues utilisées et leur fonctionnement.

Pour finir, n'oubliez pas d'associer à l'archive le script SQL permettant de recréer la base de données que vous avez utilisée pour votre projet. Les deux parties doivent travailler sur la même base de données.

Une conception MVC pour les deux parties est obligatoire. L'utilisation d'un framework MVC n'est pas obligatoire. Toutefois, seuls les frameworks vus dans le cadre des enseignements (Laravel et Spring) sont autorisés.

## Annexe Modèle conceptuel



### Schéma relationnel

**Universites** = (codeU INT, nomU VARCHAR(50), villeU VARCHAR(50), paysU VARCHAR(50), webU VARCHAR(50));

**Diplomes** = (codeDiplome INT, nomDiplome VARCHAR(50), niveauDiplome VARCHAR(50), #codeU);

**Programmes** = (codeProgramme INT, nomProgramme VARCHAR(50), dureeEchange INT, #codeDiplome, #codeDiplome\_1);

**Cours** = (codeCours INT, LibelleCours VARCHAR(50), nbECTS VARCHAR(50), annee INT, #codeDiplome);

**Etudiants** = (numEtudiant VARCHAR(50), nomEtudiant VARCHAR(50), prenomEtudiant VARCHAR(50), mailEtudiant VARCHAR(50), annee INT, #codeDiplome);

**DemandesMobilité** = (codeDemandeM INT, dateDepotDemandeM DATETIME, etatDemandeM VARCHAR(50), #numEtudiant, #codeProgramme);

**Contrats** = (codeContrat INT, dureeContrat INT, etatContrat VARCHAR(50), #codeDemandeM);

**DemandesFinancement** = (codeDemandeF INT, dateDepotDemandeF DATETIME, etatDemandeF VARCHAR(50), montantDemandeF DECIMAL(15,2), #codeContrat, #numEtudiant);

**Compatible** = (#codeCours, #codeCours 1);

**Concerner** = (#codeCours, #codeDemandeM);