

## Unités de formation « Projet de développement Web » et « Projet de développement SGBD »

### Portofio d'évaluation de la deuxième partie

Document de présentation du cahier des charges, de l'environnement de développement et de l'évaluation pour le travail servant de base à l'évaluation de la deuxième partie du « portfolio »

<h3><u>Cahier des charges</u></h3>
------------------------------------

#### Sujet

Vous devez développer un sous ensemble d'une application de gestion d'écoles de « Promotion Sociale », plus spécifiquement la gestion des personnes (étudiants, enseignants, personnel administratif), des sections, des unités d'enseignement (ue), des capacités, des ue organisées, du planning, de la grille d'évaluation et de la prise de présence.

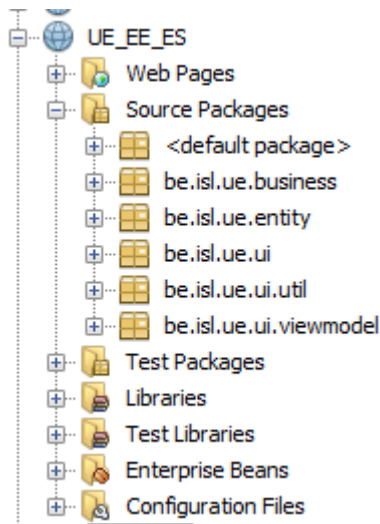
L'application sera développée dans le langage Java EE pour la partie « Dev SGBD » et Java EE, PrimeFaces, HTML, CSS, JavaScript et pour la partie « Dev Web ». L'utilisation de Bootstrap est tolérée, aucune autre librairie tierce ne peut être utilisée.

L'application sera réalisée en utilisant à bon escient les technologies vues aux cours.

Les données seront stockées dans une base de données PostgreSQL se trouvant sur le serveur Titus de l'Institut.

## Structure du projet principal

Le code de l'application sera organisé en « package » (entity, business, ui, services...).



### ***Package Entity (be.isl.ue.entity)***

Contiendra les classes entity au sens JPA.

### ***Package Business (be.isl.ue.business)***

Contiendra les EJB Stateless. Ils implémenteront la logique métier.

### ***Package Business (be.isl.ue.service)***

Contiendra les Web Services.

### ***Package UI (be.isl.prj.orders.ui)***

Contiendra les « Backing Beans » encore appelés Managed Beans ou controller. Ils implémenteront la logique de l'interface utilisateur.

### ***Package util (be.isl.ue.ui.util)***

Contiendra les éventuelles classes utilitaires pouvant aider à la gestion de l'interface utilisateur.

### ***Package viewmodel (be.isl.ue.ui.viewmodel)***

Contiendra les éventuelles classes modèles correspondantes aux vues utilisateurs.

### ***Fichiers « web »***

Une page xhtml par domaine à gérer.

Un fichier CSS pour tout le site.

Soit

- un fichier JavaScript pour tout le site, ou
- un fichier JavaScript commun, et un fichier par page.

## Interface utilisateur

L'interface utilisateur sera divisée en 5 parties comme illustré dans le schéma ci-dessous.

La zone de « Titre » servira à afficher le titre de la page courante.

La zone « Menu » affichera le menu composé de liens permettant d'accéder aux différentes pages de l'application.

La zone « Critères de Recherches » sera constituée d'un formulaire permettant à l'utilisateur d'encoder ses critères de recherches, qui seront utilisés pour construire une requête «JPQL» et ainsi interroger la base de données.

La zone « Liste Résultats » affichera un tableau reprenant en lignes les enregistrements correspondant aux critères de recherches, et en colonnes les informations les plus pertinentes permettant à l'utilisateur d'identifier une ligne. Chaque ligne de ce tableau comportera un hyperlien qui permettra d'afficher le détail de l'élément sélectionné.

La zone « Détail Élément Sélectionné » affichera le détail de l'élément sélectionné. Elle permettra à l'utilisateur de créer, modifier et supprimer cet élément.

## Fonctionnalités à implémenter

### *Sprint 1 - CRUD*

**CRUD** classiques pour la gestion

1. des personnes
2. des sections
3. des UE et des capacités
4. des UE et des UE Organisées

### *Sprint 2*

#### 1. **Gestion de l'organisation des sections.**

- a. Copie de l'organisation d'une section d'une année scolaire à la suivante (non inclus le planning).
  - i. Sélection d'une section et d'une année scolaire.
  - ii. Bouton pour démarrer la copie.
  - iii. Permette la suppression et le re-copie (si pas année en cours et si planning n'existe pas)
- b. Visualiser le résultat dans une liste pour permettre une vue globale.
- c. Depuis la liste permettre le « **CRUD** » des différentes UE organisées.
- d. Depuis la page de gestion des personnes, permettre les inscriptions d'un étudiant aux UE organisées d'un niveau d'une année scolaire d'une section et de visualiser toutes les inscriptions de l'étudiant.

Inscrire

Bac Info

2020-2021

niveau 2

Inscrire

Liste des inscriptions

Section	UE	Date Début	Niveau
Zoo	Moucher un hippo	29-10-2019	niveau 10 su
Zoo	Soigner le torticoli d'une giraffi	07-01-2020	Sensei

### 2. **Génération du planning** sur base de l'organisation de la section.

- a. Permettre la génération pour toute la section, pour un niveau ou pour une UE.
- b. Permettre la suppression sur base des mêmes critères si les présences n'existent pas.
- c. Permettre la re-génération sur base des mêmes critères si les présences n'existent pas.
- d. Visualisation/Modification (CRUD) par le responsable de section du planning semaine par semaine (liste des UE/prof/jour/heure/local).
- e. Visualisation par un professeur de son planning semaine par semaine (liste des UE/prof/jour/heure/local), incluant éventuellement le planning des autres professeurs de la section.

### 3. **Gestion des présences**

- a. Sur base du planning et des inscriptions, générer le « tableau » des présences pour toute la section, pour un niveau ou pour une UE.
- b. Permettre la suppression sur base des mêmes critères.
- c. Permettre la re-génération sur base des mêmes critères.
- d. Visualisation/Modification (CRUD) sous forme de tableau avec en ligne les étudiants et en colonne toutes les dates de l'UE.
- e. Visualisation/Modification (CRUD) sous forme de tableau synthétique qui sera utilisé par la secrétaire lors de la prise des présences. Le tableau synthétique reprendra les 3 séances précédentes et la séance du jour.

### 4. **Gestion de la grille d'évaluation**

- a. Copie par le professeur d'une grille d'évaluation d'une année précédente
- b. Prévoir le choix de l'UE parmi les UE données par le professeur l'année scolaire en cours
- c. Permettre le choix de l'année précédente indépendamment du professeur.
- d. Visualiser et prévoir la modification (CRUD) de la grille d'évaluation

### Sprint 3 - Web Services

1. **Web Service** pour réaliser les opérations CRUD pour les personnes, les sections, les UE avec les Capacités et les UE avec les UE organisées.
2. **Application Java SE** cliente indépendante qui utilisera les Web Services.

## Consignes

### Quelques règles à suivre dans le développement de votre code

L'application sera développée dans les langages Java (SE et EE) / JavaScript / HTML / CSS, en utilisant à bon escient les technologies vues aux cours. Les données seront stockées dans une base de données PostgreSQL se trouvant sur le serveur Titus de l'Institut.

**L'anglais** sera systématiquement utilisé pour définir les noms de tous les identificateurs dans votre code (classes, fonctions, méthodes, variables, ...).

Le nom du projet Netbeans (et de votre application) sera : **UE\_XYZ**, où **XYZ** désigne de votre groupe.

Le suffixe « \_id » devra être supprimé dans les « entity beans », partout où il est associé à une référence à un autre « entity bean ».

Le suffixe « collection » (ou « list, ... ») devra être remplacé par la forme plurielle dans les « entity beans ».

La méthode « toString » des « entity beans » devra retourner une chaîne de caractère permettant à l'utilisateur d'identifier directement une instance de l'entité en question. Par exemple, pour une personne la méthode toString pourrait retourner une concaténation du nom de famille et du prénom (+ éventuellement la date de naissance).

Les listes résultats des recherches et les listes de choix (<select>) seront triées.

**La structure du projet définie dans ce document sera respectée.**

**L'utilisation de la technologie AJAX sera maximisée.**

*Les champs de recherches correspondants aux entités liées seront présentés sous forme de zones de textes et non sous forme de listes de choix.*

Vous veillerez encore à :

- Respecter les conventions de dénomination standard de Java,
- Construire les requêtes JPQL en utilisant les paramètres.

### Groupes

Quatre groupes sont constitués :

- Le groupe **AAE** (Abdul Ahmadi, Andrei Chamourine, Edin Spahic)
- Le groupe **AFS** (Arnaud Paligot, Fabian Jassin, Salvatore Celestri)
- Le groupe **DMPP** (Diego Barudy, Mario Viel, Pascal Douyez, Pierre Hupkens)
- Le groupe **RSSY** (Romain Ledda, Soufian Haddouchi, Sylvestre Salerno, Yassine Fares)

### Github

Tout le code source du projet Netbeans sera placé dans des « repository » correspondants à vos groupes à l'intérieur de l'« organisation » Github <https://github.com/orgs/ISL-BE-BI>.

Minimum 3 branches seront créées tout au long du développement de l'application :

## Évaluation Projets de Développements 2022-2023 - Partie 2

- Sprint 1 : CRUD de base
- Sprint 2 : UE Organisées / Planning / Présences / Grille Evaluation
- Sprint 3 : Web services

### Tests

Un jeu de test « **propre** » par groupe devra être constitué, qui comprendra :

- une section => nom du groupe,
- minimum 6 UE, avec leurs capacités, dont les noms commenceront par le nom du groupe, par exemple dgm-ue-1, lrs-ue-1, réparties sur 2 niveaux,
- minimum 10 personnes « étudiants » qui seront inscrites dans les UE de la section, par exemple aae-etu-1, rssy-etu-1, avec par exemple 6 étudiants inscrits au niveau 1 et 4 au niveau 2.
- minimum 6 personnes « professeurs », par exemple afs-prof-1, dmpp-prof-1 ; les professeurs donneront les UE organisées. Au moins une ue-organisée sera donnée par 2 professeurs,
- une personne coordinateur de la section, par exemple aae-pers-1, rssy-pers-1,
- les UE organisées pour 2022-23,
- le planning pour les UE organisées de 2022-23 avec au moins 3 séances par UE organisée,
- les présences pour les 10 étudiants inscrits.

Le jeux de test sera créé par des requêtes SQL Insert, placées dans un script SQL, lui-même ajouté au repository Github.

### Un rapport doit être réalisé :

- Justifiant pour chaque fonctionnalité à implémenter la méthode de résolution et la technologie utilisée.
- Expliquant la structure de la base de données.
- Expliquant la structure du code source de l'application (découpe en packages, modules, classes)
- Expliquant la procédure d'installation de l'application et les prérequis logiciels permettant de la réaliser.
- Expliquant les procédures de démarrage et d'utilisation de l'application.

## Environnement technologique

### Client

- Navigateur : Firefox, Edge et Chrome dans une version récente.
- Systèmes d'exploitations : Windows, Mac OS X Linux.

### Serveur

- Java SE 17
- Serveur Java EE : Payara 5
- Java EE 8 (JSF 2.2, EJB 3.1, JPA 2.1)
- Primefaces
- PostgreSQL 10.x
- Linux ou Windows

### Environnement de développement

- Netbeans 17 (sous Linux ou Windows)

### Échéances

**L'archive zip du projet Netbeans ainsi que le rapport (pdf) seront placés sur la plate-forme Moodle de l'ISL pour le mardi 20 juin 2023**

## Évaluation

L'évaluation est directement basée sur les capacités terminales des unités de formations « Projet de développement Web » et Projet de développement SGBD ».

Les capacités terminales sont divisées en deux parties : « seuil de réussite » et « degré de maîtrise ».

Pour réussir l'unité de formation, il faut avoir acquis TOUTES les capacités du seuil de réussite, 50% des points sont alors attribués (en d'autres termes vous avez 50 sur 100).

### CAPACITÉS TERMINALES Projet de développement Web

*Pour atteindre le seuil de réussite, l'étudiant sera capable :*

au départ d'un cahier des charges proposé par le chargé de cours, face au matériel et au logiciel adéquat et face à une structure informatique opérationnelle connectée à Internet, disposant des logiciels appropriés et de la documentation nécessaire, en utilisant le vocabulaire technique et l'orthographe adéquate, et en respectant les normes et standards en vigueur,

- de créer et d'exploiter des scripts clients basés sur des classes prédéfinies ;
- de créer et d'exploiter ses propres classes ;
- de créer et d'exploiter des scripts basés sur une bibliothèque tierce ;
- d'envoyer des informations venant du client vers le serveur et de les traiter ;
- de générer un ensemble de pages web contenant un système de navigation et un contenu dynamiques intégrant formulaires et résultats.

*Pour la détermination du degré de maîtrise, il sera tenu compte des critères suivants:*

- les techniques de programmation utilisées,
- le degré de pertinence des commentaires dans le code,
- la lisibilité du code,
- le niveau de fiabilité des tests,
- le degré d'autonomie atteint.

### CAPACITÉS TERMINALES Projet de développement SGBD

*Pour atteindre le seuil de réussite, l'étudiant sera capable :*

dans le respect des consignes données, en disposant d'une station informatique opérationnelle équipée d'un logiciel « Bases de données », d'un outil de développement et sur base d'un cahier des charges fourni par le chargé de cours, en utilisant les dernières techniques d'accès aux bases de données,

- d'élaborer et de défendre un dossier technique reprenant :
  - le schéma de la base de données,
  - l'expression des contraintes en langage usuel,



- la documentation du code et la gestion des erreurs ;
- d'implémenter une base de données et l'intégrité des données;
- de programmer, de tester et de défendre la programmation de l'interface visuelle qui permet la gestion des données.

*Pour la détermination du degré de maîtrise, il sera tenu compte des critères suivants :*

- l'utilisation pertinente des procédures développées,
- le niveau de fiabilité des tests,
- le degré d'autonomie atteint,
- l'utilisation judicieuse du vocabulaire informatique.