

# COURSES

## 'Teach by Internet'

Année 2019-2020

encadrant : Jean-Michel RICHER

Prérequis: UML, Java, Swing, MySQL

Etudiants : 4 à 8

## 1 Contexte

L'épidémie de coronavirus qui nous a conduit au confinement nous a également contraint à enseigner et être enseigné par internet. Les différents logiciels comme Moodle, Teams, Discord, OpenClassroom ont vite montré leurs faiblesses : certains sont faits pour échanger de l'information, d'autres pour gérer des cours, mais leurs fonctionnalités, leurs ergonomies ne sont pas adaptées à la réalisation de cours, de TD, de TP de contrôles continus.

## 2 Présentation des éléments

Nous proposons donc la réalisation d'un logiciel qui permette à tout enseignant de mettre en place des cours de façon simple et conviviale. Le modèle que nous allons développer peut également être étendu à tout autre type d'organisation.

Le premier élément à prendre en compte est l'utilisateur, celui-ci doit pouvoir disposer d'un compte sécurisé par mot de passe.

On distinguera trois types d'utilisateurs:

- l'**administrateur** du système qui possède tous les droits
- l'**enseignant** qui doit pouvoir gérer l'ensemble de ses cours
- l'**étudiant** qui doit consulter les cours et consulter le planning des tâches qu'il doit réaliser (lire un cours, faire un TD, rendre le code associé au TD, répondre à un QCM)

### 2.1 L'enseignant

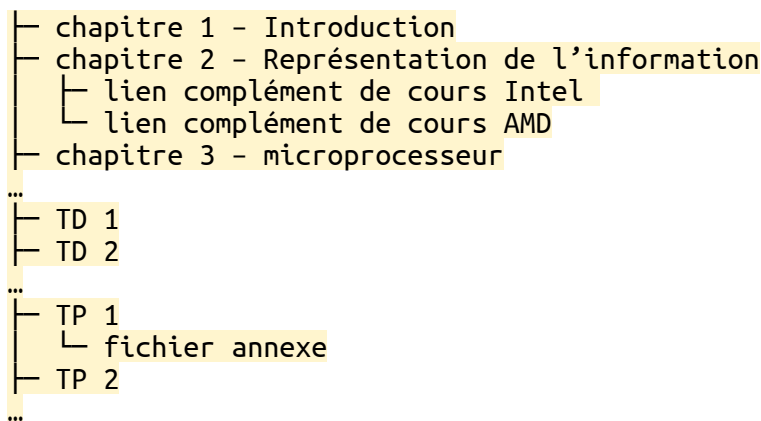
L'enseignant peut sélectionner un cours et l'organiser comme il le souhaite.

Par exemple, je peux vouloir disposer des cours suivants :

- Architecture des ordinateurs
- Programmation parallèle
- Algorithmique et programmation

Dans le cours "Architecture des ordinateurs", je vais créer une structure arborescente avec autant de niveaux que nécessaire :

- Architecture des ordinateurs



Chaque élément de l'arborescence est défini par :

- un titre (texte 80 caractères maxi)
- un descriptif (texte, vide éventuellement)
- un contenu qui est
  - du texte (texte simple, HTML, ...)
  - ou une URL
  - ou un fichier qui sera téléchargeable par les étudiants (.doc, .pdf, .ppt)
  -

Par exemple, "chapitre 1 – Introduction" est une URL vers une des pages de mon site web et "TD1" peut être une URL vers un fichier .pdf ou alors le fichier .pdf lui même qu'il faudra importer.

Un cours peut être rattaché à une **structure** que l'enseignant pourra demander à un administrateur de créer. Une structure aura plusieurs niveaux :

Structure, Unité, Département, Filière, groupe

Ainsi pour le département informatique de l'Université d'Angers :

- Structure : Université d'Angers
- Unité : UFR Sciences
- Département : Informatique
- Filière : L3 Informatique, M1 Informatique, ....
- Groupe : groupe 1, groupe 2, groupe 3

On pourra par la suite créer les autres UFR et départements associés en faisant une demande à l'administrateur. Chacun des éléments liés à une structure est défini par :

- un titre (80 car)
- un descriptif
- une URL (éventuellement vide)
- éventuellement une liste d'utilisateurs (LDAP, ActiveDirectory)

On pourra également ne pas aller jusqu'au niveau du groupe et s'arrêter au niveau de la filière, du département ou de l'unité.

L'enseignant aura également la possibilité de **créer des tâches** qui sont attribuées à des éléments d'une structure à une date fixée. Par exemple, on veut pouvoir créer les tâches :

task1: Lecture du chapitre 1 du cours Architecture des Ordinateurs page 1 à 15, associé à la filière L3 Informatique le lundi 30 Mars 2020 de 10h à 12h

task2: Lecture du chapitre 1 du cours Architecture des Ordinateurs page 15 à 30, associé à la filière L3 Informatique le lundi 6 Avril 2020 de 9h à 11h

task3: TD 1 du cours Architecture des Ordinateurs exercices 1 à 3, associé au groupe 1 de la filière L3 Informatique avec rendu du travail sous forme d'une archive le Jeudi 9 Mai de 14h à 16h

task4: TD 1 du cours Architecture des Ordinateurs exercices 1 à 3, associé au groupe 2 de la filière L3 Informatique avec rendu du travail sous forme d'une archive le Vendredi 10 Mai de 8h à 10h

L'enseignant pourra ajouter, supprimer, modifier, dupliquer des tâches. La suppression et la modification ne seront pas possibles si la date du jour est supérieure à la date de la tâche.

## **2.2 L'étudiant**

L'étudiant doit pouvoir consulter les tâches qui lui sont assignées, consulter les documents qui sont associés à cette tâche, puis les valider (bouton à cocher) ou rendre une archive de son travail pour une tâche donnée.

L'enseignant pourra télécharger l'ensemble des travaux rendus par exemple.

## **2.3 Contacts, email, chats**

On offrira la possibilité pour un enseignant d'envoyer un email à un groupe d'étudiant ou un étudiant seul. Les étudiants peuvent envoyer un message à l'enseignant concernant un élément du cours pour lui poser une question. On a donc deux possibilités : envoi d'email par messagerie électronique ou de message par l'intermédiaire du logiciel, il faudra donc stocker ces messages dans la base de données de l'application.

## **3 Travail demandé**

Tout ne se fera pas en une fois, on vous demande donc de mettre en place les bases de ce travail et de réaliser un POC (Proof Of Concept) afin de montrer ce qui peut être fait.

Le premier travail consiste à construire la base de données qui stockera les informations liées aux structures, aux cours, aux utilisateurs.

Le second travail consiste à mettre en place une application en Java qui permette de gérer structures et cours sous forme arborescente grâce à la classe JTree de Swing par exemple et de sauvegarder ou charger l'information et l'afficher.

Je vous propose la création de 4 TaskForces :

- BD (MYSQL) qui met en place la base de données et la gestion avec Java par exemple avec Hibernate
- STRUCT (pour Structure) qui se charge de la gestion des structures (Structure, Unité, Département, ...)
- COURSE (Pour Cours) qui se charge de l'arborescence des cours
- USER (pour Utilisateurs) qui se charge des utilisateurs : création de compte, connexion, association à une ou plusieurs structures, création de groupes (par exemple sélection des utilisateurs d'une filière pour créer les groupes de TD et TP)

Pour STRUCT et COURSE on utilisera un Jtree avec possibilité d'ajouter un élément à une branche de l'arborescence, d'en supprimer tout ou partie, de déplacer d'une branche vers une autre.