

# M2107 : PROJET TUTEUR

## OBJECTIF DU MODULE : MISE EN ŒUVRE DES METHODES DE GESTION DE PROJET

### COMPETENCES VISEES AU S2 :

- Mise en pratique de la méthodologie de conduite de projets
- Développement des compétences d'autonomie et d'initiative de l'étudiant
- Développement des aptitudes au travail en équipe

### VOLUME HORAIRE : 80 H / ETUDIANT (HORS PLANNING)

### MODALITES DE REALISATION :

Le projet s'étend sur tout le semestre et porte sur la réalisation d'une application de gestion. Le but du projet est de découvrir, par la pratique, comment un projet de développement informatique est géré et planifié, et comment travailler efficacement en équipe sous la conduite d'un chef de projet. Une équipe de développement est un **tiers de groupe de TP** et contient donc jusqu'à 5 développeurs. Le rôle du chef de projet est joué par un enseignant qui sera affecté à chaque équipe. Le rôle du client est joué également par un enseignant affecté à chaque équipe. Le développement du projet est découpé en **3 phases**, ponctuées par les évaluations du chef de projet et du client.

L'objectif majeur du module étant la **mise en situation professionnelle**, tous les outils utilisés pour la réalisation du projet et la communication dans l'équipe doivent être des outils professionnels, validés par les enseignants. C'est à l'étudiant d'amener la preuve de son implication dans le projet (liste de réalisations, comptes-rendus de réunion, planning effectif précis).

### SUJET :

Tous les groupes sont mis en concurrence et travaillent donc sur le **même sujet**. Le cahier des charges complet est fourni dans le document **Progiciel MSP.docx** fourni.

Il n'est pas demandé de réaliser l'ensemble de l'application, mais uniquement les fonctionnalités de l'application vue par les utilisateurs de type « chef de projet ».

L'analyse et la conception complète de l'application pour les chefs de projets ont été réalisées, et sont disponibles de deux manières :

- Dans le fichier Visual Paradigm **ProgicielMSP.vpp** fourni
- Dans le dossier **ProgicielMSP-docHTML**, visible par un navigateur web, le cahier de conception généré par Visual Paradigm

IL EST IMPERATIF DE BIEN SUIVRE CETTE CONCEPTION (Y COMPRIS AU NIVEAU DU NOM DES CLASSES, DES FONCTIONS, ETC.).

## CHOIX DE TECHNOLOGIES

L'application nécessite l'utilisation de plusieurs technologies :

- On souhaite une **interface graphique**, il faut donc savoir réaliser une IHM standard à partir de boutons, zones de saisies, etc. Les utilisateurs de l'application étant de plusieurs nationalités, tous les textes de l'interface doivent être rédigés en anglais.
- Les données sont présentes dans une **base de données MySQL**, il faut donc savoir se relier à une BDD depuis une application.
- L'application doit pouvoir s'exécuter sur un poste standard de l'entreprise (PC / Windows 10).

Le langage, le framework, l'EDI à utiliser sont au choix de l'équipe. Il est néanmoins fortement conseillé d'utiliser un des outils décrits ci-dessous.

Afin de vous aider à utiliser ces technologies, une mini-application simple a été réalisée, dans plusieurs langages. La conception de cette application (fichier **Tutorial.vpp**) est fournie. L'application a été réalisée dans plusieurs technologies, le code source est disponible dans le dossier **Tutorials**. Les technologies utilisées sont :

- Langage Pascal, EDI Lazarus, IHM par LCL.
- Langage Java, EDI Netbeans
  - IHM par Swing
  - IHM par JavaFX
- Langage C++, EDI Qt, IHM par Qt *widgets*.
- Langage C#, EDI Visual Studio, IHM par WPF.
- Langage Python, EDI spyder (mais d'autres possibles), IHM par Tkinter

La BDD du tutorial est fournie(**tutorial.sql**). C'est un serveur MySQL (accessible uniquement sur le réseau IUT). Ses paramètres sont :

- Host : SRV-IQ-ETU
- DatabaseName : Directory
- User : iq1
- Password : iq1

Afin de pouvoir tester hors de l'IUT, un script `tutorial.sql` permet de créer dans un serveur local (type XAMPP) la même base. Attention : il faudra modifier les sources du tutoriel pour modifier les paramètres de la base.

Il est demandé aux équipes de commencer par étudier cette application test afin de découvrir comment réaliser une IHM et comment utiliser une BDD, dans le langage choisi.

La **base de connaissances** du département (<https://iutdijon.u-bourgogne.fr/intra/iq/bc>) contient des articles pouvant aider sur les points techniques.

## BASE DE DONNEES

La base de données sur laquelle l'application doit s'appuyer est fournie sous forme de deux scripts SQL :

- `CréationBase.sql` permettant de créer une base vierge
- `DonnéesTest.sql` permettant de remplir la base avec des données de test

Le modèle de cette base est disponible dans le **document de conception** ainsi que dans le cahier des charges. La base contient plus de tables que nécessaires à l'application car elle peut être utilisée dans d'autres applications.

---

## REPARTITION DES TACHES ET PLANIFICATION

Le document **Liste des tâches.xlsx** contient la description des différentes tâches devant être réalisées. Chaque tâche possède un certain nombre d'indicateurs :

- Le type de compétence utilisée (IHM, code, base de données...)
- Un niveau de difficulté relative (entre 1 et 3)
- Une durée estimée (en unités relatives)
- Les éventuelles tâches pré-requises

A vous de faire la **répartition des tâches**, en fonction des compétences de chaque membre de l'équipe : il est important que toute l'équipe avance et que l'application soit la plus fonctionnelle possible. L'application est découpée en plusieurs fonctionnalités. Certaines sont dépendantes d'une autre fonctionnalité et devront donc être réalisées dans le bon ordre !

Chaque tâche (à part les 2 premières) n'est normalement effectuée que par une seule personne : à vous d'indiquer dans votre propre version du fichier le nom de la personne qui s'occupe de la tâche.

Cette liste de tâches ne comprend pas la première phase (découverte et auto-formation sur les technologies à utiliser) sur laquelle chacun doit participer. **Tous les membres du groupe** doivent se former sur les technologies choisies.

## SUIVI ET EVALUATION DU PROJET

Le suivi sera confié à deux enseignants pour chaque groupe :

- Un enseignant d'informatique jouant le rôle du chef de projet
- Un enseignant non informatique jouant le rôle du client

Chaque groupe aura un minimum de **3 rendez-vous** avec le chef de projet et avec le client :

- En début de projet (S9)
- En milieu de projet (S18)
- En fin de projet (S24)

Pour chaque rendez-vous, on attend un support de présentation assistée par ordinateur clair, précis et pertinent. Le fichier liste de tâches doit être complété et mis à jour. **La planification doit être la plus précise possible et le planning réel mis à jour en indiquant les heures de travail effectif de chacun pour chaque tâche.**

---

## MAQUETTE MINIMALE :

Avant le premier rendez-vous, chaque groupe doit réaliser un travail sommaire "maquette minimale" dans deux technologies différentes : l'application doit indiquer dans une "boîte de liste" la **liste des utilisateurs de la base**. Ce travail préparatoire est essentiel pour choisir la technologie. Il en s'agit que d'une acquisition de technologies et en aucun cas de l'amorce du projet en lui-même.

### CALENDRIER INDICATIF

Phase 1	<b>S4 → S9</b>	
	<p>Choix de technologies à tester. Compréhension des tutoriels et du sujet : cahier des charges et dossier de conception.</p> <p>Réalisation d'une maquette minimale dans deux technologies différentes. Choix de la technologie de développement. Répartition et planification des tâches du projet.</p>	
	<i>Evaluation Chef de Projet</i>	<i>Evaluation Client</i>
	<p>Présentation des maquettes minimales réalisées et justification du choix de la technologie de développement.</p> <p>Présentation de la répartition et de la planification des tâches.</p>	<p>Présentation de la technologie choisie et de deux maquettes différentes parmi lesquelles le client choisit sa préférée.</p> <p>Présentation des fonctionnalités prévues pour le second rendez-vous.</p>
Phase 2	<b>S10 → S18</b>	
	Réalisation des premières fonctionnalités.	
	<i>Evaluation Chef de Projet</i>	<i>Evaluation Client</i>
	<p>Présentation de l'application en cours.</p> <p>Comparaison de la planification originale et de la planification corrigée (prévue/réelle). Correction éventuelle de la liste des tâches initiales.</p> <p>Estimation du degré d'avancement. Indication des tâches terminées.</p>	<p>Présentation de l'application en cours.</p> <p>Démonstration des fonctionnalités déjà implémentées au client pour validation.</p> <p>Présentation des fonctionnalités prévues pour le dernier rendez-vous.</p>
Phase 3	<b>S19 → S24</b>	
	Développement des dernières fonctionnalités. Corrections et améliorations du produit.	
	<i>Evaluation Chef de Projet</i>	<i>Evaluation Client</i>
	<p>Présentation du produit fini.</p> <p>Retour sur les difficultés du projet</p> <p>Présentation de la répartition / planification définitive, comparaisons avec le prévisionnel.</p> <p>Synthèse des compétences acquises ou perfectionnées grâce à ce projet, des compétences à travailler.</p>	<p>Présentation du produit fini.</p> <p>Explications des raisons des fonctionnalités non implémentées ou réalisées différemment de ce qui était prévu au départ.</p> <p>Mise en avant des atouts du produit.</p>