



CRÉATION D'UN OUTIL SUPPORT À LA GESTION PAR PARCOURS ÉTUDIANTS DANS UN CURSUS UNIVERSITAIRE À BASE DE MAJEURES/MINEURES

Le projet est commun aux modules de *Bases de données* et *Technologies Web*. Il consiste à développer une application support au suivi du parcours d'étudiants intégrés à une école d'ingénieur avec un fonctionnement du cursus décliné sous forme d'UE majeures-mineures. Les futurs utilisateurs de cette application sont les membres du services de scolarité de l'école.

Durant le projet, vous jouez à la fois le rôle de la maîtrise d'ouvrage (MOA) et celui de la maîtrise d'œuvre (MOE). Votre encadrant de projet joue le rôle du commanditaire, et des éventuels utilisateurs métier (votre encadrant ne pourra en revanche pas jouer le rôle de support technique). Votre travail comportera deux grandes parties. Dans un premier temps, vous concevrez et développerez la base de données sous-jacente à l'application (périmètre du module de *Bases de données*). Dans un second temps, vous développerez quelques codes applicatifs, avec interface graphique, de l'application en langage PHP (périmètre du module de *Technologies web*).

Dans cet énoncé vous trouvez, en section 1, une première description de l'application visée, telle que vous auriez pu l'élaborer après une prise de contact avec vos commanditaires (les membres du service de scolarité de l'école pour laquelle vous développez l'application). Vous trouvez également une liste de fonctionnalités minimales que l'application finale devra pouvoir gérer. Dans la section 2 de ce document, vous trouvez des informations d'ordre organisationnel, la liste des livrables attendus, et une description de la forme de ces livrables.

1 Description générale de l'application

Le contexte est le suivant. Votre entreprise a été chargée de développer un système permettant le suivi du parcours d'étudiants d'une école d'ingénieurs du type de celle de l'ENSSAT, intégrant les étudiants à Bac+2, pour les former en trois années au métier d'ingénieur, avec délivrance d'un diplôme d'ingénieur à l'issue de la formation (si toutes les conditions nécessaires sont

remplies) bien entendu. Votre interlocuteur (et unique utilisateur) est le service de scolarité de l'école (vous n'aurez pas à gérer de multiples utilisateurs de l'outil, cette fonctionnalité étant attendue pour une étape ultérieure du projet, à laquelle vous ne participerez pas).

Le service de scolarité vous a décrit le fonctionnement général de son métier, en déroulant les processus métier sous-jacents (comme le font généralement les utilisateurs), c'est-à-dire en partant d'un scénario de base, et en ajoutant quelques informations complémentaires autour de ce scénario, au moment où l'information vient à l'esprit.

Le texte *en italique* ci-dessous est extrait des propos de votre futur utilisateur.

Scolarité: "À la rentrée, nous inscrivons chaque étudiant dans le système, nous associons à son numéro d'étudiant les données administratives nécessaires au suivi de la scolarité¹. Chaque étudiant choisit un déroulé initial idéal de sa formation en définissant son parcours idéal (dans la vision qu'il a au moment de son inscription, qui pourra évoluer en cours de son cursus). L'étudiant, à l'inscription, choisit un parcours type parmi les parcours types que nous proposons (ces parcours types devront être renseignés dans le système afin que nous puissions nous y référer).

Un parcours type est composé de 5 semestres académiques et un semestre professionnel (le semestre professionnel est un stage de projet de fin d'étude). Un semestre peut être impair ou pair. Les semestres impairs ont lieu sur la première moitié de l'année scolaire (septembre-janvier). Les semestres pairs ont lieu sur la seconde moitié de l'année (février-juin). Normalement (hors cas particuliers dont on parlera après), à chaque semestre impair (resp. pair), un parcours d'étudiant doit comporter une unité d'enseignement (UE) majeure dispensée au semestre impair (resp. pair) et une UE mineure dispensée au semestre impair (resp. pair). Valider une majeure rapporte 21 ECTS. Valider une mineure rapporte 9 ECTS. Chaque unité d'enseignement est validée indépendamment des autres. On veut pouvoir renseigner et stocker l'information de validation des unités d'enseignement dans les parcours des étudiants, pour savoir où en sont les étudiants. À ces 5 semestres académiques s'ajoute une unité professionnelle, qui correspond à un stage qui se déroule sur un semestre. Le semestre professionnel peut avoir lieu en dernière année, en semestre impair ou pair. On ne stocke aucune information concernant le semestre professionnel hors mis le fait que l'étudiant désire le positionner (tout le reste est géré dans une autre entité de l'école), et on veut pouvoir stocker l'information de la validation ce semestre professionnel, qui rapporte 30 ECTS. Dans ces parcours, l'année indiquée est celle de la rentrée de septembre (2022 correspond à l'année scolaire qui va de septembre 2022 à juin 2023).

Les figures 1 et 2 ci-dessous sont deux exemples de parcours types.

Parcours Informatique Cyber			
Année	Semestre	Majeure choisie	Mineure choisie
2022	impair	SoInfoMaj	HardMin
	pair	SoMaths	FondProgMaj
2023	impair	GLMaj	InfoAvMin
	pair	CyberMaj	SysResMin
2024	pair	WebMaj	CloudMin
	impair	UP (unité professionnelle)	

Figure 1: Exemple de parcours parcours Cyber

¹Dans le cadre de votre premier développement, vous pouvez vous limiter à quelques informations incluant le nom, le prénom et la date de naissance de l'étudiant (jour-mois-année).

Parcours Informatique IA			
Année	Semestre	Majeure choisie	Mineure choisie
2022	impair	SoInfoMaj	HardMin
	pair	SoMaths	FondProgMaj
2023	impair	GLMaj	InfoAvMin
	pair	IANumMaj	IASymMin
2024	pair	WebMaj	ProjIMin
	impair	UP (unité professionnelle)	

Figure 2: Exemple de parcours IA

Nous gérons un catalogue des unités d'enseignement qui peuvent être mises dans un parcours. Il prend la forme de la table 1 ci-dessous.²

code character varying (15)	modalite character varying (3)	libelle character varying (50)	impair boolean	pair boolean	capacite integer
CloudMin	Min	Cloud (m)	true	false	35
CyberMaj	Maj	Cyber sécurité (M)	false	true	35
FondProgMaj	Maj	Fondements de la program...	false	true	35
FondProgMin	Min	Fondements de la program...	false	true	35
GLMaj	Maj	Génie Logiciel (M)	true	false	35
HardMin	Min	Matériel (m)	true	false	22
IANumMaj	Maj	Apprentissage et IA numériq...	false	true	35
IASymMin	Min	Gestion des données et IA s...	false	true	35
InfoAvMin	Min	Informatique avancée (m)	true	false	35
ManMathsMin	Min	Mise à niveau Maths (m)	false	true	30
ProjIMin	Min	Projet intégré (m)	true	true	40
SoInfoMaj	Maj	Socle de Base en Info (M)	true	false	35
SoMathsMaj	Maj	Socle de Base en Maths (M)	false	true	30
SysResMin	Min	Système et réseaux (m)	false	true	25
WebMaj	Maj	Web (M)	true	false	35

Table 1: Bout de catalogue de unités d'enseignement

En plus, pour chaque UE, on liste les modules que contient l'UE dans une autre partie du catalogue. Ma collègue vous placera sur l'ENT une partie de ce catalogue.³

Les figures 1 et 2 ne sont que deux parcours parmi d'autres. Si la plupart des étudiants reste sur l'un des parcours type que nous proposons, il leur est quand même possible de s'en écarter un peu. S'il désire faire évoluer son parcours, un étudiant peut passer à la scolarité pour ajuster sa formation, en concertation avec nous. Il arrive par exemple que des étudiants demandent à passer leur UP (leur stage) en semestre impair de troisième année, et revenir à l'école après le stage pour effectuer leur dernier semestre académique en semestre impair. Une telle manipulation sur le parcours de la figure 2 revient à avoir le parcours de la figure 3. Certains étudiants demandent à échanger des UE, par exemple certains étudiants Cyber échangent parfois

²Dans l'objectif de restreindre le périmètre fonctionnel, nous allons considérer que tout module ajouté au catalogue reste dans le catalogue, et que sa capacité est constante également. En revanche, le nom du responsable de l'UE peut changer dans le temps (indiqué dans la description fonctionnelle).

³Voir le dump de départ -sans aucune contrainte-, se trouvant sur l'ENT. À vous de l'exploiter pour l'intégrer dans votre base de données.

le module CloudMin par IASym pour apporter une composante différente IA à leur profil. Une telle manipulation sur le parcours de la figure 1 revient à avoir le parcours de la figure 4. Dans l'idée, les étudiants peuvent demander à échanger d'autres unités d'enseignement, même des majeurs, ou les faire à un autre moment.

Parcours Informatique IA avec UP anticipée			
Année	Semestre	Majeure choisie	Mineure choisie
2022	impair	SoInfoMaj	HardMin
	pair	SoMaths	FondProgMaj
2023	impair	GLMaj	InfoAvMin
	pair	IANumMaj	IASymMin
2024	pair	UP (unité professionnelle)	
	impair	WebMaj	ProjIMin

Figure 3: Exemple de parcours IA avec UP anticipée

Parcours Informatique Cyber avec coloration IA			
Année	Semestre	Majeure choisie	Mineure choisie
2022	impair	SoInfoMaj	HardMin
	pair	SoMaths	FondProgMaj
2023	impair	GLMaj	InfoAvMin
	pair	CyberMaj	SysResMin
2024	pair	WebMaj	IASymMin
	impair	UP (unité professionnelle)	

Figure 4: Exemple de parcours Cyber avec coloration IA plus prononcée

Il n'y a pas tellement de contrainte pour l'échange d'une UE par une autre dans un parcours, les étudiants sont assez libre. Il faut juste que l'unité d'enseignement à placer dans un semestre soit dispensée au semestre dans lequel l'étudiant veut l'insérer (pair ou impair), et que chaque semestre étudiant comporte au plus une majeure et une mineure (mais pas deux majeures, car cela ne rentrerait pas dans l'emploi du temps de l'étudiant). C'est nous qui vérifions si les majeures-mineures d'un même semestre dans un parcours sont compatibles entre elles deux (cela ne peut pas être géré par l'application pour le moment car cela dépend du positionnement des pavés de cours dans l'emploi du temps), et que l'étudiant ne repasse pas des thématiques déjà acquises (par exemple prendre une mineure qu'il a déjà obtenu en majeure).

Le parcours d'un étudiant peut également comporter jusqu'à deux semestres de césure (un semestre de césure ne rapporte aucun ECTS).

Lorsque l'étudiant valide une majeure ou une mineure, nous mettons à jour le système pour l'indiquer. D'un point de vue fonctionnel, il faut qu'on puisse voir l'état d'un parcours pour chaque étudiant (l'afficher, et savoir quelles unités d'enseignement ont été validées), savoir où l'étudiant en est en termes de validation et d'ECTS accumulés, et pouvoir mettre à jour le parcours de l'étudiant.

Pour permettre la diplomation, un parcours doit comporter au moins 5 majeures et une unité professionnelle validées, et 180 ECTS validés au final.

Les choses se compliquent un peu lorsque l'étudiant ne parvient pas à valider certaines unités d'enseignement, car son parcours comporte plus de trois années. L'étudiant peut choisir de

suivre de nouveau l'unité d'enseignement ratée pour tenter de la valider, ou peut choisir de suivre une autre unité d'enseignement. Il faut pouvoir prendre cela en compte également. On peut même avoir des cas un peu plus compliqués pour lesquels l'étudiant fait son stage (unité professionnelle) mais n'a pas complètement validé sa partie académique, et il redouble pour finir sa partie académique. Dans un cas comme celui-là, l'unité professionnelle n'a pas lieu en dernière année puisque l'étudiant doit remplir pour un semestre académique, éventuellement sur une année incomplète. On gère cela en ajoutant le(s) semestre(s) supplémentaire(s) au parcours après le stage.

On veut aussi pouvoir rentrer le parcours d'étudiants qui ne passent que pour une année ou un semestre dans notre école. Dans ce cas, le contrat de l'étudiant (le nombre d'ECTS à décrocher pour ressortir avec une validation de l'enssat) n'est pas de 180 ECTS, mais de 30 ECTS -si passage pour 1 semestre seulement- ou de 60 ECTS -si passage pour une année complète-.

On stocke aussi des enseignants responsables d'unité d'enseignement. Le responsable d'une unité d'enseignement peut changer d'une année sur l'autre. Il faut que nous gardions l'historique des responsables des unités d'enseignement d'année en année, afin de savoir qui a été le responsable de l'unité d'enseignement d'un étudiant dans son parcours."

Votre objectif est de développer une application web servant de support aux différentes étapes de suivi du parcours des étudiants de l'école. Comme dans la "vie réelle", le sujet ne peut pas contenir toutes les informations permettant de développer l'application. À vous de vous tourner vers votre commanditaire (rôle joué par votre encadrant de projet) quand cela vous semble nécessaire, comme vous l'auriez fait en situation réelle.

Fonctionnalités. On précise maintenant les fonctionnalités minimales attendues (leur évaluation tiendra compte de la qualité de l'implantation en elle-même et la qualité de leur présentation et justification dans le rapport).

Fonctionnalités relevant de la couche *bases de données* Les fonctionnalités minimales de l'application finale, d'un point de vue bases de données, sont listées ci-dessous.

- (MOA) le modèle conceptuel UML de l'application, expliqué par les choix conceptuels que vous aurez effectués (justification, limitation, ...), avec une partie de son dictionnaire de données⁴.
- (MOE) le modèle logique de votre application (attention à la normalisation de votre schéma, permettant d'éviter de stocker d'information redondante),
- (MOE) l'implantation des vues, répondant aux besoins fonctionnels suivants (dans une application réelle, nous n'implémenterions pas tous les besoins sous forme de vues mais l'implantation sous forme de vue permet à votre chargé de projet d'accéder facilement à l'implantation de vos requêtes, afin d'étudier le travail que vous aurez fourni) :
 1. création de la vue `VueUEImpair` permettant de récupérer les unités d'enseignement impaires (dispensées en semestre impair);
 2. création de la vue `VueUEPair` permettant de récupérer les unités d'enseignement paires (dispensées en semestre pair);
 3. création de la vue `VueUEImpairPair` permettant de récupérer les unités d'enseignement qui sont à la fois impaires et paires (dispensées en doublons aux deux semestres);
 4. création de la vue `VueModuleNbUE` permettant de récupérer le nombre d'unités d'enseignement auxquelles appartient chaque module;
 5. création de la vue `VueModuleImpair` permettant de récupérer les modules impairs (ceux qui appartiennent à des UE impaires);
 6. création de la vue `VueModulePair` permettant de récupérer les modules pairs (ceux qui appartiennent à des UE paires);
 7. création d'une vue `VueAJourneesDetail` permettant de récupérer la liste des ajournements aux unités d'enseignement (liste des étudiants ayant échoué à une unité d'enseignement (un même étudiant peut avoir échoué à plusieurs unités), avec année et unité concernée, triés par année puis nom de famille de l'étudiant);
 8. création d'une vue `VueAJourneesAgg` permettant de récupérer le nombre d'ajournements, le nombre d'inscrits et le pourcentage d'ajournements, par année, de chaque unité d'enseignement;
 9. création d'une vue `VueCesure` permettant de récupérer les étudiants ayant une césure dans leur parcours, avec l'année et le(s) semestre(s) de césure des étudiants concernés;
 10. création d'une vue `Vue2Cesures` permettant de récupérer les étudiants ayant deux semestres de césure dans leur parcours, avec une seule ligne par étudiant contenant le nom de l'étudiant et les deux années de césure (schéma de la forme `{nom, prenom, anneeCesure1, anneeCesure2}`, ou approchant de cette forme);

⁴On sait que la création d'un dictionnaire est fastidieuse, vous pouvez donc vous limiter, pour le projet, à une partie du dictionnaire.

11. création de la vue **VueECTS** permettant de récupérer la liste des étudiants (avec leur numéro d'étudiant, leur nom et leur prénom), avec le nombre d'ECTS cumulés acquis pour chacun des étudiants, triés par ordre descendant d'année, et par nom ascendant d'étudiant;
12. création de la vue **VueDiplomes** permettant de récupérer la liste des étudiants ayant acquis au moins une unité professionnelle, 5 majeures, et les 180 ECTS académiques nécessaires à la validation de leur diplôme, avec le nom, le prénom et l'année de naissance de l'étudiant, et le rappel du nombre d'ECTS acquis (il peut y en avoir plus de 180, par exemple si un étudiant qui redouble pour une mineure, décide de passer une majeure à la place pour mieux occuper son semestre, voir même décide d'occuper son semestre supplémentaire avec une majeure + une mineure...);
13. création de la vue **VueInscritsUE** permettant de récupérer la liste des inscrits, par année, à une unité d'enseignement, trié par année, par nom d'unité d'enseignement puis nom d'étudiant;
14. création de la vue **VueInscritsModule** permettant de récupérer la liste des inscrits, par année, à un module (un étudiant est inscrit à un module s'il est inscrit à une UE qui contient ce module), trié par année, par nom de module puis nom d'étudiant;
15. création d'une vue **VueEtudiantX**, où X est le numéro de l'étudiant (à instancier avec un numéro d'étudiant de votre instance), paramétrée par un numéro d'étudiant, permettant de récupérer le parcours complet (vous créerez une vue pour un étudiant de votre instance);
16. création d'une vue **VueAcquisX**, où X est le numéro de l'étudiant, paramétrée par un numéro d'étudiant, permettant de récupérer les unités d'enseignement validées ou à valider d'un étudiant (n'affichant pas les unités d'enseignement "ratées");
17. création d'une vue **VueAcquisDetailX**, paramétrable par un numéro d'étudiant, permettant de récupérer le parcours complet de l'étudiant, avec le nombre d'ECTS acquis pour chaque unité d'enseignement, et le nom de l'enseignant responsable de chaque unité au moment où l'étudiant a suivi l'unité;
18. création d'une vue **VueInscritsUnite** affichant par année et par unité d'enseignement, le nombre d'inscrits à l'unité;
19. création d'une vue **VueImpasseSurX**, paramétrée par une chaîne de caractère (par exemples "architecture"), affichant les noms et prénoms des étudiants n'ayant dans leur parcours aucune unité d'enseignement dont le nom ou l'un de modules de UE contient le tag X;
20. Et enfin, les contrôles de qualité des données : vous indiquerez les requêtes qui vous permettraient de cibler les problèmes qualité de la base sur les aspects non contrôlés par le schéma -par exemple détecter les unités pour lesquelles le nombre d'inscrits dépasse la capacité de l'unité, donner les parcours comportant plus de deux semestres de césure, etc- (dans une application, les contrôles qualité pourraient être implantés ou bien dans le code fonctionnel, ou bien dans la base de données dans des fonctions PL-SQL, mais là vous le ferez dans des requêtes de contrôle, sauvegardées dans des vues). Ces vues sont à définir par vous, le nom de chacune de ces vues doit commencer par **QtDefault** (par exemple **QtDefaultOverflow** listant les unités d'enseignement en sur-capacité). Votre rapport devra expliquer les différentes requêtes définies (la raison d'être

de chaque requête, mais sans fournir le code de la requête). Votre rapport devra également indiquer les mises en œuvre de contrôle qualité qui n'ont pas pu être implantées et qui devront être implantées dans la partie fonctionnelle de l'application.

Les vues doivent être **indépendantes** les unes de autres.

Fonctionnalités relevant de la couche fonctionnelle (en PHP) Votre application est une version simplifiée de l'application finale réelle. Elle doit fournir une page d'accueil à partir de laquelle l'utilisateur peut choisir d'afficher le contenu de chacune des vues (chaque vue est affichée dans une nouvelle page via l'interrogation de la base de données).

Pour passer d'une vue à une autre, chaque page disposera d'un lien vers une page principale proposant la liste des fonctions.

Des points supplémentaires seront accordés pour le développement de fonctionnalités supplémentaires relevant de la partie PHP.

2 Organisation du projet

Le projet est à effectuer en **trinôme**. Votre première tâche sera triple 1) constituer votre trinôme, 2) trouver un nom à votre entreprise spécialisée dans l'édition de logiciels support pour la gestion d'enseignement, et 2) trouver un nom à l'application que vous concevrez et développerez. Vous travaillez ensuite en collaboration au sein de ce trinôme pour développer cette application.

2.1 Les deux grandes étapes du projet

Vous pouvez distinguer deux grandes parties dans le développement de l'application. Une partie de représentation des données (modèle de données s'appuyant sur les fondements présentés dans le module de *Bases de données*), et une partie relevant des codes applicatifs et de l'interface (vues et contrôleur s'appuyant sur les fondements présentés dans le module de *Technologies web*).

(Première étape) Vous commencerez par concevoir la base de données sous-jacente à l'application. Pour concevoir la base de données, vous devrez identifier des cas d'utilisation de l'application et quelques scénarii associés, concevoir un modèle conceptuel de la base de données, et en déduire un schéma conceptuel, puis un schéma logique issu de la traduction du schéma conceptuel (on rappelle l'importance de définir un dictionnaire de données associé au schéma logique proposé). Ces éléments devront être présentés dans votre livrable rapport (voir section 3). Vous alimenterez ensuite la base avec des données de votre choix avec des données vous permettant d'exhiber le bon fonctionnement de votre application. Les vues feront partie de la base de données.

(Deuxième étape) Vous pourrez ensuite concevoir le code applicatif et les interfaces. Pour cela, vous devrez a minima définir une architecture de votre application, vous appuyer sur les cas d'utilisation pour développer un code applicatif permettant de les mettre en œuvre. Vous devrez concevoir des interfaces graphiques **conviviales** (et l'enchaînement de celles-ci) pour les différents types d'utilisateurs. D'un point de vue organisation de votre travail, il vous sera nécessaire d'identifier des itérations de développement et de répartition des tâches entre

les membres du projet. Ces éléments devront être présentés dans votre livrable rapport (voir section 3).

3 Forme des livrables attendus du projet, et dates butoir

Trois livrables sont attendus à l'issue du projet, décrits ci-dessous.

(DL1) Un rapport expliquant l'organisation mise en œuvre et vos choix d'implantation. Il s'agit essentiellement de présenter les choix de conception de votre application, et comment vous vous êtes organisés pour le développement en lui-même (itérations, diagramme de GANTT, éventuelle répartition des tâches).

Le rapport comportera un page de garde, un sommaire, et au maximum quinze pages de contenu hors annexes (si vous désirez inclure des annexes, celles-ci ne devront comporter aucune ligne de code SQL). Le rapport devra être structuré, comporter a minima une page de garde comportant tous les éléments permettant d'identifier le sujet traité -sujet, date- et les auteurs, un sommaire, une introduction, un contenu développé dans lequel seront dissociées la partie relevant de la base de données et la partie relevant de la technologie Web, ainsi qu'une conclusion. Le rapport prendra la forme d'un fichier PDF nommé *DL1_NomApplication_M1_M2_M3.pdf* où *NomApplication* est le nom de votre application, et *M1*, *M2* et *M3* sont à remplacer par les noms de famille des membres du groupe triés par ordre alphabétique.

(DL2) Vos programmes de mise en œuvre du système. Votre code devra être fonctionnel et commenté. Le code source sera fourni à votre enseignant sous forme d'une archive intitulée *DL2_NomApplication_M1_M2_M3.tgz* (ou *.zip*) où *NomApplication* est le nom de votre application, et *M1*, *M2* et *M3* sont à remplacer par les noms de famille des membres du groupe triés par ordre alphabétique. Votre code devra comporter, à sa racine, un fichier README.md fournissant la procédure d'installation -pré-requis et étapes d'installation- de l'outil que vous aurez livré. En particulier, le fichier README.md devra donner la marche à suivre pour installer et lancer le script php, les informations de connexion.

Votre archive devra contenir un fichier README indiquant au minimum comment utiliser les fichiers fournis dans l'archive (pré-requis système, procédure d'installation). N'oubliez pas de joindre, en plus des programmes, un fichier *dump* de votre base de données.

Les livrables DL1 (rapport) et DL2 (sources du projet) seront déposés, au plus tard le **mardi 26 avril à 23h55**, sur le dépôt ENT prévu à cet effet (le dépôt se fera sur le compte du membre M1 du projet, c'est-à-dire le premier dans l'ordre alphabétique du nom de famille⁵).

(DL3) Une démonstration scénarisée La dernière séance des TP (3 mai) sera dédiée à des démonstrations. La démonstration de chaque groupe devra durer environ 7 minutes et suivre un scénario que vous aurez judicieusement élaboré pour (i) montrer que votre application respecte le cahier des charges attendu et (ii) mettre l'accent sur les points forts de votre application. Tous les membres du trinôme devront participer à la démonstration (vous découperez la démonstration de façon à prendre la parole à tour de rôle, avec une répartition équitable du temps de parole). Suivront ensuite quelques minutes de questions (et éventuellement de tests en temps réel de votre application).

⁵Ne déposez pas plusieurs fois le même projet sur l'espace de dépôt.

Quelques précisions : (i) Ne négligez surtout pas le rapport qui constituera une partie importante de la note. (ii) Des points seront accordés au respect de l'énoncé (en particulier le respect des modalités de remise des livrables tels que les formats des fichiers, les noms des fichiers archives, et évidemment le respect des dates buttoir).