LOG1000
–
Ingénierie logicielle TP #2 :

Objectifs de la modélisation UML

- À partir des requis, concevoir le design d'un logiciel en utilisant la modélisation UML.
- Réaliser des diagrammes de cas d'utilisation, de séquence et de classe.
- Comprendre les concepts d'association, d'agrégation, de composition et d'héritage.

Déroulement du travail pratique (TP)

Le TP va couvrir une partie de la conception logicielle essentielle afin de transformer la vision d'un client en un produit logiciel fiable, maintenable et flexible. Vous devrez d'abord bien analyser la vision du client et en extraire des cas d'utilisation montrant les différents acteurs,opérations et relations. Ces cas d'utilisation reflètent chacun un scénario particulier.
Par la suite, suite à l'obtention des cas, vous élaborez des diagrammes de séquence décrivant de manière chronologique le déroulement de certains cas. Ayant bien identifié les différents objets constituant le système, il sera par la suite possible de bâtir un diagramme de classe montrant les liens entre les classes essentielles et leurs méthodes.

Rédaction du rapport

Votre rapport sera un document PDF contenant les divers diagrammes et les réponses aux questions demandées. Le rapport sera remis dans
un dossier nommé TP2 dans votre répertoire Git.
N'oubliez pas de vérifier après la remise si votre rapport est beau et bien visible sur le serveur et pas seulement sur votre copie locale.

UMLet

Pour effectuer vos diagrammes, vous devez utiliser l'outil UMLet que vous devez l'installer dans votre dossier personnel. Pour vous renseigner aux directives d'installation et d'utilisation de cet outil, veuillez visiter le lien suivant (important : téléchargez la version stand--alone) :http://www.umlet.com/

Voici un tutorial qui explique comment faire un diagramme de cas
d'utilisation et un diagramme de séquence en utilisant l'outil UMLet :
https://www.youtube.com/watch?v=HVSxE296JLc.

Trucs et astuces:

- Les acteurs ne sont pas des objets.
- Utilisez des verbes infinitifs au début des noms de cas d'utilisation et de fonctions.
- Les objets et les classes sont indépendants de hardware/plateforme.
- Voir aussi les astuces sur Moodle.

PARTIE 1: Conception des diagrammes UML

Mise en Contexte :

Moodle est une plateforme d'apprentissage en ligne écrite en PHP. Développée à partir de plusieurs principes pédagogiques, permet de créer des communautés s'instruisant autour de contenus et d'activités. Le mot « Moodle » est
l'abréviation de *Modular Object-Oriented Dynamic Learning Environment* : « Environnement orienté objet d'apprentissage dynamique modulaire ».
Outre la création de cours à l'aide d'outils intégrés (ressources et activités) à l'usage des formateurs, Moodle offre des possibilités d'organisation des cours sous forme de filières (catégories et sous-catégories, cohortes…) qui lui donnent également des caractéristiques propres à la mise en place de dispositifs complets d'enseignement.
À un système de gestion de contenu (SGC) (en anglais : *Content Management System* ou CMS) déjà cité, Moodle ajoute aussi de nombreux outils d'interactions pédagogiques et communicatives créant un environnement d'apprentissage en ligne : cette application permet de créer, via le réseau, des interactions entre pédagogues, apprenants et ressources pédagogiques.
Grâce à son architecture modulaire, Moodle profite de plugins développés par sa communauté pour permettre l'extension de ses fonctionnalités et de répondre ainsi à des besoins spécifiques[1].

Pour ce laboratoire, on vous demande de concevoir un système logiciel pareil que Moodle.

Votre travail dans le cadre de ce TP est de réaliser en partie cette modélisation qui se divise en trois aspects (tel que décrit ci-dessus)

- Aspect 1 : Le diagramme des cas d'utilisation.
- Aspect 2 : Le diagramme de séquence.
- Aspect 3 : Le diagramme de classes.

Afin que vous puissiez vous familiariser avec le fonctionnement de ce genre d'application, vous avez ci-dessous quelques informations supplémentaires. Ainsi, les objectifs de cette application sont :

La gestion des utilisateurs par un administrateur de système:

- Gérer des utilisateurs.
- Attribuer des rôles.
- Mettre à jour du système.

Les utilisateurs autre que les enseignants:

- Se connecter à moodle à l'aide d'un mot de passe.
- Éditer leur profil.
- Écrire un message à un autre utilisateur de moodle(même établissement scolaire).
- Consulter leurs cours.
- Faire la remise des travaux.
- Passer un examen.
- Avoir accès à leurs notes.
- Participer ou créer un post sur le forum dédié à cet effet.

Personnes en charge d'enseignement:

- Publier un cours.
- Modifier la visibilité d'une publication.
- Programmer une action sur un fichier(Ex: le rendre visible à une date précise).
- Créer un quiz.
- Consulter les données d'un participant au cours.

Tout ce qui est décrit au-dessus et bien autres sont les objectifs basiques et essentiels qui doivent se trouver dans une plate forme scolaire comme Moodle.

Aspect 1:Le diagramme des cas d'utilisation (30 pts)

Afin de réaliser le diagramme des cas d'utilisation du système, basez-vous sur la mise en situation et les exigences fonctionnelles que vous en extrayez.

Q1: Faites un diagramme de cas d'utilisation basé sur les exigences discutées au-dessus.
Votre diagramme devrait contenir l'ensemble des acteurs identifiés et l'ensemble des cas d'utilisation qui vous semblent pertinents. N'oubliez pas les relations entre les cas!

Q2: Décrivez textuellement les 3 cas d'utilisation qui vous semblent les plus essentiels en suivant le modèle ci-dessous (cf. exemple du cours), et expliquez aussi pourquoi vous avez choisi ces 3 cas

Titre	
Acteurs impliqués	
Préconditions	
Post conditions	
Scénario principale	Liste numérotée
Scénario d'exception	Liste numérotée

Aspect 2: Le diagramme de séquence (30 pts)

Construire 3 diagrammes de séquence, selon les scénarios ci-dessous :

- Scénario 1: Ecrire un message à un autre utilisateur de moodle
- Scénario 2: Faire la remise des travaux
- Scénario 3: Passer un examen en ligne

Le diagramme de séquence permet de visualiser le déroulement chronologique de chacun des cas d'utilisation.Il vous faut considérer:

- Tous les objets impliqués et les messages qu'ils s'envoient. Choisissez un nom clair pour chaque classe, méthode et argument dans vos diagrammes.
- L'ordre chronologique des événements doit se dérouler du haut vers le bas.

Aspect 3 : Diagramme de Classes (20 pts)

Maintenant que vous avez identifié les objets qui constituent le système, vous êtes en mesure de réaliser le diagramme de classe contenant toutes les classes pour implémentervotre part du système. Même pour les cas qu'ils n'ont pas été élaborés, il faut ajouter les classes nécessaires.

Q1 :Identifiez chacune des classes qui constituent votre système informatique.

Q2 : **Dessinez le diagramme de classe représentant votre logiciel. Il vous faut considérer :**

- Tous les attributs et méthodesessentiels de chacune des classes. • Les relations entreles classes : agrégation, composition, association ou héritage.

- Les quantités pour chaque relation, s'il y a lieu (par exemple, un-à-plusieurs, plusieurs-à plusieurs, etc.)

PARTIE 2 :Rétro-ingénierie de diagramme de classe à partir du code (20 pts)

Dans le dossier **src** fourni avec les fichiers source du tp, vous allez trouver un ensemble de fichier .c .cpp et .h. Votre mission pour cette partie du tp est de faire un diagramme de classe pour ces fichiers.

il faut considérer que:

- Les attributs et méthodes essentiels qui montrent les liens entre les classes.
- Les relations entre les classes: agrégation, composition, association ou héritage.

Considérations importantes pour la fin du TP Toujours faire un « git add » des nouveaux fichiers, un « git commit » et un « git push » de vos dernières modifications pour que l'on puisse voir la dernière version de votre travail lors de la correction. Si vous ne faites pas de commit, il se peut que l'on évalue une version différente de votre TP local sur le serveur Git. Vérification que vos travaux sont présents dans le répertoire Git du serveur :
Vous pouvez utiliser la commande « git ls-files » afin de voir les fichiers dans le dernier snapshot de l'entrepôt Git lui-même.

Tout ce qui est « commit » après cette date ne sera pas considéré dans la correction. !!
Pénalités pour retard : 10% par jour !!

Date de remise:

B2 : 28/09/2018
B1 : 05/10/2018

références :

1- https://fr.wikipedia.org/wiki/Moodle
2- http://www.uml-diagrams.org/