

INF1163 : Modélisation et conception orientée objet

Projet

Système de gestion d'un club vidéo de location de films

Travail en équipe (4 membres par équipe)

Le but de ce projet est d'appliquer les principes de l'analyse et de la conception orientées objet vus en classe, notamment l'approche unifiée et UML. L'implémentation se fera en Java. Le projet vous donne aussi l'occasion d'expérimenter avec quelques outils logiciels performants utilisés dans un environnement de développement professionnel (Git, GitHub, SQLite).

La collaboration entre les membres de chaque équipe est indispensable afin de pouvoir respecter les délais impartis. Les étudiants doivent donc apprendre à travailler en équipe. Ils doivent notamment apprendre à se répartir les tâches équitablement, à développer leur sens de la communication et à synchroniser leur travail afin d'être productifs dans leurs efforts. Il faudra donc nommer un chef d'équipe pour coordonner le travail et veiller à la bonne progression du projet. Le rôle de chef d'équipe pourra être joué, à tour de rôle et pour une période déterminée à l'avance, par chacun des membres de l'équipe. Afin de gérer efficacement le développement de votre code source et de pouvoir rapidement revenir à une version qui fonctionne en cas de problème, il est fortement conseillé d'utiliser l'outil de gestion de version Git (<http://git-scm.com/>). Afin de permettre à plusieurs membres de l'équipe de développer en parallèle et d'intégrer automatiquement leurs différentes portions de code, il est fortement suggéré d'utiliser l'outil de gestion de contrôle GitHub (<https://github.com/>). GitHub est en fait la partie serveur pour l'outil Git. Git est gratuit et peut être installé sur votre PC. GitHub est un serveur sur le nuage qui requiert que vous ouvriez un compte (gratuit).

L'objectif final du projet consiste à développer un système logiciel de gestion pour un club vidéo de location de films. En raison des contraintes temporelles, on se limitera au développement d'une version simplifiée qui implémente principalement la logique d'affaires du club vidéo et qui offre des interfaces utilisateur simples. La composante base de données ne faisant pas partie des objectifs de ce cours, on pourra se contenter d'utiliser des fichiers Java ou Excel pour stocker les données relatives à la gestion au club vidéo. Toutefois, pour les équipes qui le voudront, il est conseillé d'utiliser le système de gestion de bases de données simple et pratique suivant : SQLite (<http://www.sqlite.org/>). Il s'agit d'un SGDB léger et gratuit qui s'installe très facilement sur n'importe quel PC.

Brève description du système

Le club vidéo de location de films, appelé par commodité dans la suite du texte vidéoclub, dispose de films sur support DVD et Blu-ray. Les nouveautés sont louées à la journée, les autres films peuvent être loués à la journée ou à la semaine. C'est le gérant du vidéoclub qui est responsable de l'acquisition de nouveautés et qui décide quand un film n'est plus considéré comme une nouveauté. Le gérant décide aussi du nombre de copies de chaque film que le vidéoclub veut acquérir. Les employés du vidéoclub doivent alors libérer certains présentoirs pour faire de la place aux nouveautés. Les employés sont aussi en charge du rangement des films sur les présentoirs suivant une organisation thématique (films d'horreur, comédies, etc.).

Le vidéoclub propose aussi à la vente, du popcorn et des bonbons, ainsi que des films déjà visionnés dont on veut se débarrasser pour libérer de l'espace.

Ce vidéoclub ne loue pas de jeux vidéo ni de consoles de jeux ou de visionnement.

N'importe qui peut acheter les produits proposés à la vente, mais seuls les membres du vidéoclub peuvent louer des films. Il n'y a pas de frais pour devenir membre, mais il faut fournir une empreinte d'une carte de crédit valide ainsi que son adresse et son numéro de téléphone. Chaque client dispose alors d'un code secret qui, combiné à son numéro de téléphone, permet son identification et son authentification.

Lorsqu'un film n'est pas retourné dans les délais, l'emprunteur se voit imposer une amende égale au montant de la location et ce par période de location entière (jour ou semaine). Par exemple, un client qui retournerait, avec 3 jours de retard, un film loué à la semaine, se verrait facturé pour toute une semaine additionnelle. Lorsque le retard excède 2 jours pour une nouveauté, 5 jours pour un film régulier loué à la journée et 2 semaines pour un film régulier loué à la semaine, un employé du vidéoclub téléphone au client pour faire un premier rappel. Un deuxième et dernier rappel est fait avec la même périodicité. Un film non retourné après 10 jours de retard pour une nouveauté et 60 jours pour un film régulier, est considéré comme perdu et le prix du film, plus taxes, plus 20\$ (pour compenser le manque à gagner du vidéoclub) est chargé sur la carte de crédit de l'emprunteur.

Le système logiciel projeté doit permettre le fonctionnement habituel attendu d'un tel commerce. Pour toute autre information non fournie mais pertinente (coût de location, etc.), vous êtes libres de faire un certain nombre d'hypothèses raisonnables que vous devez clairement mentionner dans votre livrable final.

Plan de travail

Ce plan de travail a pour principal objectif de vous guider en s'assurant que le travail progresse, dans les délais impartis, vers le but final.

À noter que deux livrables intermédiaires sont obligatoires et doivent être remis à échéance sous peine de pénalités (5% de la note du projet pour chaque livrable intermédiaire en retard ou non remis).

Itération 1 (2 semaines : du 24 octobre au 7 novembre) :

Identifier tous les cas d'utilisation appropriés et les représenter dans un diagramme de cas d'utilisation. Classer les cas utilisation identifiés par priorité et développer les deux cas jugés les plus prioritaires.

Proposer un modèle du domaine. Coder en Java et tester les classes conceptuelles retenues. Proposer un DSS pour chaque cas d'utilisation développé, et le cas échéant quelques contrats.

Faire une première esquisse à la main de l'interface utilisateur principale. Dans cette première itération, aucun code relatif à l'interface utilisateur ne devrait être développé.

Construire une première « base de données » des données importantes du restaurant pour des fins de tests.

Attention : livrable intermédiaire obligatoire à rendre le lundi 31 octobre (minuit) : diagramme des cas d'utilisation, un des deux cas d'utilisation documenté au format détaillé, et l'esquisse de la principale interface utilisateur. Déposez dans Moodle votre livrable sous forme d'un **fichier unique**, compressé au besoin, et portant le nom : Equipe_X_Livrable1 (le X devra être substitué par le numéro assigné à votre équipe). **Tous vos textes et diagrammes doivent être incorporés dans un fichier unique Word ou PDF. Aucun diagramme au format mdl ou sms ne sera accepté.** Le but de me rendre ce livrable après juste une semaine c'est de me permettre de vous donner du feedback lors de notre séance de cours du 7 novembre afin que vous puissiez vous assurer que vous êtes sur la bonne voie à la fin de cette première itération.

Itération 2 (2 semaines : du 7 au 21 novembre) :

Développer deux ou trois autres cas d'utilisation (selon leur complexité). Ajuster au besoin le modèle du domaine. Proposer des DSS et le cas échéant des contrats, pour ces cas d'utilisation.

Pour les 2 cas d'utilisation développés lors de la première itération, proposer des diagrammes d'interaction. En déduire un premier diagramme de classes. Coder ces classes et les tester.

Faire des esquisses des autres interfaces utilisateur jugées nécessaires selon les cas d'utilisation développés. Commencer à les coder en utilisant, par exemple, la bibliothèque Swing de Java.

Attention : Livrable obligatoire (lundi 21 novembre au plus tard minuit): modèle conceptuel et 2 diagrammes DSS. Tous vos textes et diagrammes doivent être incorporés dans un fichier unique Word ou PDF. **Aucun diagramme au format mdl ou sms ne sera accepté.** Déposez dans Moodle votre livrable sous forme d'un **fichier unique** (compressé au besoin) et portant le nom : Equipe_X_Livrable2.

Itération 3 (2 semaines : du 21 novembre au 5 décembre) :

Pour les deux cas d'utilisation développés lors de la seconde itération, proposer des diagrammes d'interaction. En déduire le diagramme de classes final.

Finaliser le code de toutes les classes retenues.

Finaliser toutes les interfaces utilisateur.

Tester toutes les fonctionnalités implémentées.

Il n'y a rien à rendre ici, mais lisez la suite.

Présentation orale des projets : lundi 12 décembre en classe (état d'avancement et démo d'un prototype).

Livrable final dû le vendredi 16 décembre à minuit au plus tard

Le livrable final du projet comprenant :

- Tous les artefacts UML adéquats et finaux qui justifient la solution proposée (au minimum : cas d'utilisation développés, DSS, contrats le cas échéant, diagrammes d'interaction, diagramme de classes).
- Un bref texte explicatif résumant vos choix de conception et liant ensemble tous les plans et documents de conception fournis.
- Le code source en Java, adéquatement documenté.
- Toutes les explications nécessaires pour permettre à une tierce personne de compiler, exécuter et utiliser votre prototype. Si vous avez utilisé des bibliothèques non standard, il faut le préciser et donner leur URL et toute autre explication nécessaire à leur intégration dans votre code.
- Une annexe qui synthétise la contribution de chaque membre de l'équipe au projet. Par exemple : « Jean a collaboré à l'écriture des cas d'utilisation 2 et 4, a développé entièrement le diagramme de séquence 9 et a codé 20% de l'application (ou a codé 2 des 3 interfaces utilisateur) », etc.

À l'exception du code source, tous les autres éléments du livrable final doivent être incorporés dans **un fichier unique au format Word ou PDF**. Votre texte (livrable final

sauf code source) doit être tapé en utilisant un traitement de texte (ou autre logiciel) et vérifié, le cas échéant, en utilisant un correcteur orthographique.

Ce livrable final constitué (i) d'un fichier unique contenant texte et diagrammes et (ii) d'un répertoire contenant tous les fichiers de code source, doit être compressé dans un fichier qui portera obligatoirement le nom : **INF1163_Projet_Equipe_X.zip** et devra être déposé via Moodle.

Toute soumission qui ne se conformera pas à ces directives sera refusée.