

INF1163 : Modélisation et conception orientée objet

Projet

Système automatisé de gestion de restaurant

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.

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.

L'objectif final du projet consiste à développer un système logiciel de gestion pour un restaurant où les clients sont servis à table. En raison des contraintes temporelles, on se limitera au développement d'une version simplifiée qui implémente principalement la logique d'affaires du restaurant et qui offre des interfaces utilisateur simples. La composante base de données ne faisant pas partie des objectifs de ce cours, on se contentera d'utiliser des fichiers Java ou Excel pour stocker les données sur les transactions et autres données du restaurant.

Brève description du système

Un propriétaire de restaurant, bien côté et achalandé, vous engage avec le mandat de développer un logiciel pour automatiser la gestion des opérations quotidiennes de son commerce. Le système informatique doit aider les personnels du restaurant (serveurs, cuisiniers, etc.) à coordonner leurs activités et à améliorer leur service, et doit permettre au(x) gérant(s) de suivre l'évolution du commerce et planifier sa croissance.

Le système informatique doit avoir des interfaces utilisateur intuitives et être facilement portable sur différentes plateformes d'exécution. Il doit permettre le fonctionnement habituel attendu d'un tel commerce. Pour toute autre information non fournie mais pertinente, vous êtes libres de faire un certain nombre d'**hypothèses raisonnables** que **vous devez clairement mentionner dans votre livrable final**.

La figure 1 ci-dessous schématise le mode de fonctionnement courant des restaurants non automatisés.

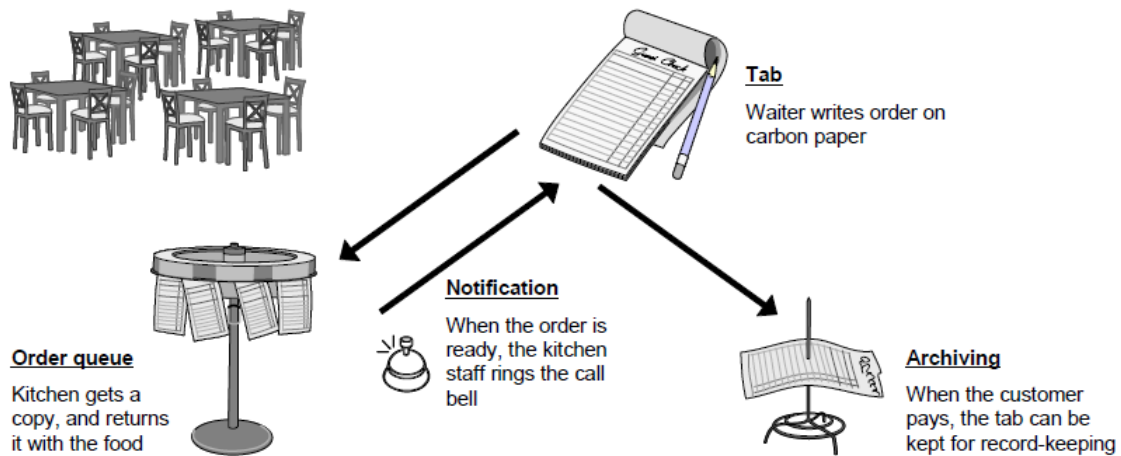


Figure 1: Old-fashioned restaurant operation.

La figure 2 donne une idée de ce à quoi pourrait ressembler un mode de fonctionnement automatisé.

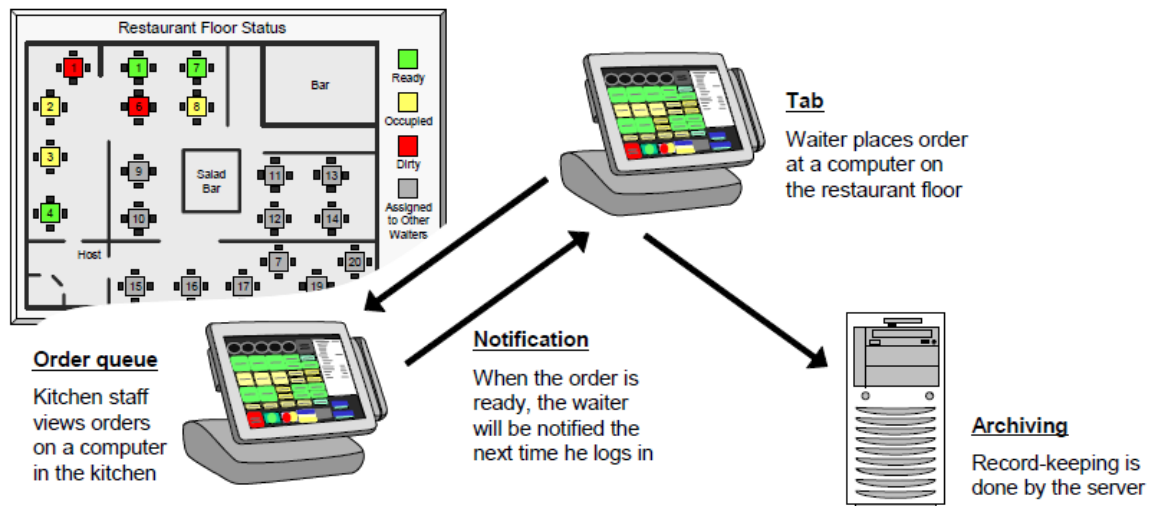


Figure 2: Restaurant automation facilitates staff coordination and record keeping.

Bien sûr, cette figure n'est qu'indicative et incomplète, et vous êtes libres de proposer des variations et/ou améliorations par rapport à ce modèle. Par exemple, pour éviter que le serveur oublie quelque chose entre le moment où vous lui dictez votre commande et le moment où il arrive à la console de saisie (appelée *Tab* sur la figure), il serait plus approprié qu'il soit muni d'une tablette tactile qui lui permet de rentrer directement votre commande à partir de votre table¹. Si vous optez pour une telle solution, vous n'avez pas

¹ À noter que cette amélioration n'a rien d'extraordinaire, elle est déjà en usage dans plusieurs restaurants de la région.

à développer du code spécifique pour tablette tactile; vous vous contenterez de proposer une interface utilisateur accessible à partir de votre application principale mais qui simule ce à quoi pourrait ressembler la vraie interface sur écran tactile si vous aviez le temps et les moyens de développer une **vraie application client-serveur**.

Plan de travail

Ce plan de travail, bien qu'indicatif, est très fortement recommandé. Il 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 certains 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).

Il est très fortement recommandé (voire obligatoire) d'utiliser un système de gestion de version afin de permettre à chaque membre de l'équipe d'être productif et de travailler en parallèle sans corrompre le travail des autres membres.

Itération 1 (2 semaines : du 26 octobre au 9 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 la fin de la première semaine (vendredi 2 novembre au plus tard 8h30) : 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 et portant le nom : Equipe_X_Livrable1.zip (le X devra être substitué par le numéro assigné à votre équipe).

Itération 2 (2 semaines : du 9 au 23 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 (vendredi 23 novembre au plus tard 8h30): modèle conceptuel et 2 diagrammes de séquences du système (DSS). Tous les diagrammes doivent être incorporés dans un document Word ou PDF. Aucun diagramme au format mdl ne sera accepté. Déposez dans Moodle votre livrable sous forme d'un fichier unique (compressé au besoin).

Itération 3 (2 semaines : du 23 novembre au 7 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.

Livrable final dû le samedi 15 décembre à minuit au plus tard

À noter : la période entre la fin de la 3^{ème} itération (7 décembre) et la date limite pour rendre le projet (15 décembre), est destinée à faire face à d'éventuels imprévus et à finaliser le tout. Oubliez que vous disposez de cette petite sécurité et n'intégrez surtout pas cette semaine supplémentaire dans votre planification pour éviter les mauvaises surprises.

Livrable final à rendre:

Samedi 15 décembre 2012 à minuit: 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.
- 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.

Votre livrable final 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 doit être compressé dans un fichier qui portera obligatoirement le nom : **INF1163_Projet_Equipe_X.zip** et devra être déposé via Moodle. **X** est le numéro de l'équipe ; il vous sera assigné en classe.

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