

TP Génie Logiciel, 3IF, 2017/2018

Rappel : l'objectif principal de ce TP est de vous habituer à utiliser une méthode de travail permettant l'obtention d'un logiciel de qualité.

Sujet

Description

Un entreprise X développe un nouveau capteur à usage médical, permettant de construire une sorte d'empreinte décrivant la santé d'une personne¹. Cette empreinte est définie par un certain nombre de mesures correspondant à plusieurs attributs. L'empreinte est produite sous forme de fichier. Bien évidemment, cette information doit rester confidentielle.

Un organisme gouvernemental en charge du dépistage de plusieurs maladies souhaite mettre en place un système informatique permettant de détecter rapidement les différents risques de maladie pour une analyse médicale (empreinte). La mise en évidence d'un risque permettra d'effectuer des analyses complémentaires (plus coûteuses) pour lever le doute. Un ensemble d'empreintes est déjà disponible, avec les éventuelles maladies correspondantes.

Ce système doit permettre :

- l'analyse d'une nouvelle empreinte ou d'un ensemble d'empreintes
- l'affichage des maladies prises en compte
- la description des caractéristiques permettant d'identifier une maladie

Les capteurs équipent des laboratoires d'analyses médicales. Le capteur est piloté par une tablette numérique (mémoire + processeur + écran).

L'objectif principal de l'entreprise est de limiter le temps d'attente entre l'analyse et le diagnostic.

Vous êtes en charge de la réalisation de ce système, en particulier :

- la spécification détaillée du système,
- la conception et l'implémentation
- les tests qui permettront de garantir la qualité finale du système

Livrables attendus

TP1

- *Spécification des besoins*² : Dossier d'analyse contenant les spécifications fonctionnelles et non-fonctionnelles du système (reprise du CDC initial, éclaircissement des points flous, ...)

¹ Pour des raisons de confidentialité, le contenu "scientifique" du sujet est imaginaire, il a juste une valeur pédagogique.

² Le format de chaque document est défini par l'équipe projet, incluant sur la première page les informations minimales pour assurer la traçabilité (titre, auteurs, date, version, historique des modifications ...)

- *Plan de tests de validation / recette* (négociation avec le client) ; ce document doit contenir en annexe les formats de différents fichiers permettant
- *Description des interfaces éventuelles* (formats des données ...)
- *Manuel utilisateur* (ébauche)
- *Planning* (ébauche)

TP2

- Dossier de conception (décomposition en modules, diagrammes de classes, de séquence, d'états-transitions et autres, description des algorithmes importants, structuration des données ...)
- Plan de tests d'intégration
- Première version des tests unitaires
- Planning (ébauche)

TP3&4

- à la fin de chaque séance : une nouvelle version de l'application
- code source de l'application + code source des tests + éventuels données

TP4

- Manuel utilisateur
- Procédure d'installation
- Code source + guide de construction de l'exécutable
- Tests

Validation

- la date sera précisée en séance

Plan pédagogique (pour 2017-2018 !)

TD1

Travail en quadrinôme - CDP et RQ

Périmètre : expression des besoins (comprendre!) et spécifications détaillées

Lecture et approfondissement du sujet

Livrables attendus :

- Spécification des besoins : définition des besoins fonctionnelles et non-fonctionnelles (interaction avec le client!) ; il est important de classifier ces besoins (exigences) et les donner un identifiant, ceci nous permettra d'associer des tests (tests fonctionnels) etc.
- étude de faisabilité (?) : trop court pour le réaliser - donc ne pas le prendre en compte - sauf pour les quadrinômes trop en avance
- les élèves peuvent produire aussi d'autres documents (recommandé !) - comme par exemple : "Description des entrées sorties côté client" (qui peut être vu comme une annexe)
- planning pour les prochaines séances (à la fin du TD) -> le CDP
- il est fort probable que les livrables attendus seront finalisés dans la première heure du TD2

Questions attendus :

- expression des besoins -> c'est le document que nous avons donné comme sujet
- questions sur l'efficacité du système ? (temps de réponse etc.) - ça fait partie du test de validation final (recette)
- format de données en entrée : chaque groupe va définir ce format, plus tard nous allons en donner un pour tous (TD3 ou TP1 ?)
- le contenu de serveurs est cohérent : une même maladie sera décrite de la même manière, mais il est possible qu'une maladie ne soit pas décrite sur tous les serveurs
- IHM : mode texte (ligne de commande)

Questions :

- ne pas oublier que chaque groupe doit faire le client et le serveur
- l'ordre de mots n'est pas importante
- délai pour une requête : best effort (real time) -> tests de perfs en fct de la taille du dictionnaire et la taille de la requête -> #req/sec
- pas de authentification et chiffage
- traçabilité : log coté client et serveur
- primitives protocole coté serveur : description, req pour une maladie et une analyse et requête pour toutes les maladies et une analyse
-

Contenu minimal d'un dossier des spécifications (très schématique) :

- système composé de trois modules principales : client, serveur et protocole de communication client-serveur

- client : permet de gérer une analyse médicale :
 - C1 : lecture d'une analyse à partir d'un fichier texte
 - C2 : lecture de la liste de serveurs disponibles (fichier texte)
 - C3 : envoyé de la requête concernant l'analyse à tous les serveurs et réception du résultat
 - C4 : affichage du résultat (console + fichier texte)
 - NF1 : traçabilité (fichier texte décrivant les opérations C1-C4 effectuées)
- serveur : permet de répondre à des interrogations :
 - S1 : lecture du dictionnaire (maladie - mots descriptifs) - format fichier texte à définir
 - S2 : lecture de paramètres (port d'écoute, nom fichier de log)
 - S3 : réception d'une requête de type analyse médicale et envoi de la réponse
 - S4 : vérifier si une analyse est susceptible de correspondre à l'une de maladies répertoriées
 - S5 : réception d'une requête demandant les capacités du serveur (quels sont les maladies qu'il sait vérifier)
 - S6 : réception d'une requête demandant si le serveur sait répondre pour une maladie particulière
-
- Description du protocole client-serveur : nécessaire pour permettre à des acteurs extérieurs d'intégrer des nouveaux serveurs (ou clients)
 - primitives minimales : query, description,

TD2

- les élèves ont pris du retard avec le document de spécification, l'objectif est d'avoir une version de ce document complète à la fin de la séance
- une description des tests de validation doit être finalisée
- avant le TD3 l'enseignant doit valider les spécifications - les groupes ont envoyé un document à l'enseignant ; plus facile pour ceux qui ont envoyé un lien google doc, commentaires en ligne ; pour les autres : commentaires par retour de mail
- Specs :
 - il faut pouvoir lire le document sans avoir lu le CDC initial !
 - il faut une description du système et quelques cas d'usage
 - ne pas oublier les formats I/O : fichier text, avec exemples, et le protocole réseau !
-

TD3

- nous sommes dans la situation où nous avons en entrée : dossier de specs, tests de validation, description des fichiers des données et protocole de communication client-serveur ;

- il faut réaliser la conception du système (de deux côtés, client et serveur) et décrire les tests d'intégration
- finaliser le dossier de spécifications (en prenant en compte les commentaires de l'enseignant)
- préparer les tests unitaires

TP1

- 2h : finaliser la conception + prise en compte du format de données et protocole normalisé (document moodle)
- 2h : initiation à Visual Studio (vidéoprojecteur + doc moodle)

TP2

- finaliser la conception + envoi documents
- codage
- implémentation de tests

TP3

- codage
- implémentation des tests unitaires

TP4

Annexes

Introduction Visual Studio

Utilisation du git (avec Visual Studio)

Communication réseau en utilisant les classes MFC Visual Studio

Description filtre de Bloom -> le projet ne l'utilise plus ...