Projet « GSB – Visiteurs » en Java

Contenu

Cahier des charges

Définition du besoin

Les rapports de visite sont actuellement gérés à l'aide d'une application Access (Cf. dossier joint) que les visiteurs utilisent dans les locaux de l'entreprise sur différents postes dédiés à cet effet.

Un projet d'application Web est en cours pour rendre accessible cette gestion à partir de n'importe quel ordinateur. Cependant, l'accès à cette gestion doit être maintenu à l'aide d'une application de bureau, mais l'application Access doit évoluer.

Il a été décidé de développer une application de bureau en java (swing) et de porter la base de données sur SGBD Oracle.

Fonctionnalités

L'application doit remplir les mêmes fonctionnalités que l'application existante.

Déploiement

Base de données

In fine, le déploiement de la BDD sera effectué sur un SGBD Oracle tournant sur une machine virtuelle VMWare.

Un script SQL de création de cette base de données vous est fourni. Vous aurez peut-être à retravailler ce script.

Application

Le déploiement de l'application devra être prévu sur la machine de test de l'équipe sous la forme d'un jar exécutable avec un raccourci sur le bureau de l'utilisateur "2SLAM_2016".

Contraintes

Contraintes d'accessibilité et de sécurité

L'application doit être accessible facilement (raccourci sur le bureau) à partir de l'ensemble des ordinateurs de l'entreprise que ce soit des plateformes Windows ou Ubuntu.

Contraintes d'architecture du code

L'application respectera le modèle MVC et donc son organisation.

Contraintes d'ergonomie

L'application doit s'inspirer fortement de l'application Access existante et est donc constituée d'un certain nombre d'écrans auxquels on accède à travers un écran de menu principal.

On essaiera de soigner l'ergonomie comme par exemple :

- le contrôle des saisies : ne pas être obligé de cliquer sur un bouton après avoir fait un choix sur une liste déroulante,
- valider ou invalider des composants (boutons de commande en particulier) suivant les circonstances.
- afficher précisément les erreurs et replacer le curseur dans le champ de saisie.

Des améliorations ou variations peuvent être proposées par rapport à l'application Access existante.

Contraintes de codage

Les noms de fichiers, de classes, d'objets, de méthodes doivent respecter les normes de codage habituelles.

Vous utiliserez des composants graphiques Swing.

Vous utiliserez l'environnement de développement NetBeans.

Vous commenterez votre code avec les « tags » (annotions) Javadoc.

Environnement de développement

Vous utiliserez Netbeans sous Windows associé à un SGBD Oracle.

Vous utiliserez un serveur de versioning (GitHub).

Documentation

La documentation de l'application comprendra :

- l'arborescence des fichiers commentée (un texte de présentation de chaque paquetage),
- la « javadoc » produite sur les classes créées,
- la liste des bibliothèques supplémentaires utilisées.

Méthodes de travail

Développement « agile »

Les projets doivent être conduits en équipes, suivant une méthode agile. Nous nous inspirerons de la méthode « Scrum ».

C'est une méthode itérative (itérations courtes, les « sprints »), où l'équipe de développement s'auto-organise. Un sprint doit toujours conduire à un produit livrable et documenté, destiné au maître d'ouvrage. Un sprint peut durer plusieurs séances. A chaque séance, l'équipe doit planifier son travail lors d'une courte réunion (« Daily scrum »). ***La trace de ces « daily - scrum » doit être conservée***. Le daily-scrum doit notamment contenir le bilan de la séance précédente, les décisions de l'équipe comme la répartition des tâches pour la séance en cours.

L'équipe est animée par un chef d'équipe (le « scrum master »).

Le rôle du maître d'œuvre est joué par les enseignants, le maître d'ouvrage étant GSB.

Les tests

Tests unitaires

Les tests unitaires des classes métier et modèle seront systématiques (une classe de test par classe codée).

Le framework JUnit pourra être mis en œuvre à cette occasion. Il faut concevoir les classes de test au fur et à mesure et conserver la trace des diagnostics par date.

Tests fonctionnels

Les tests fonctionnels seront réalisés en utilisant les scénarios et les jeux d'essai prédéfinis. Les tests donneront lieu à des rapports de tests.

Rappels :

- Les **scénarios** et les **jeux d'essai** sont conçus **avant** la réalisation de l'application (codage et tests)
- 1 cas d'utilisation est décrit par plusieurs scénarios : 1 scénario nominal, des scénarios alternatifs, des scénarios d'exception
- 1 jeu d'essai est destiné à valider l'ensemble des scénarios d'un cas d'utilisation

- 1 cas de test est un ensemble de valeurs destiné à tester un scénario
- 1 scénario peut être validé par plusieurs cas de test
- L'ordre des cas de test est important, car un cas de test peut affecter l'état initial de la base de données
- 1 jeu d'essai fait l'objet d'un seul test par rapport de tests : ce test est soit conforme, soit non conforme (dans ce cas, un commentaire est utile)
- Le rapport de tests comporte une ligne de tableau par test

Se reporter aux modèles de documents fournis.

Définition d'un « livrable »

Cf. grilles de livrables annexées aux sprints.

Outils

Gestion de projet

Elle sera basée sur l'outil « Redmine » ou un équivalent open source. Les tâches, leur affectation, l'état de leur avancement, la saisie des compte-rendus des « daily scrums ». Redmine génère les diagrammes de Gantt. Veillez à bien prévoir toutes les tâches en début d'itération et à enregistrer le diagramme de Gantt prévisionnel.

Dépôt de code

Logiciel de gestion de versions : « Git »

Dépôt externalisé : « git-hub.com »

Un IDE comme NetBeans permet de gérer un dépôt de code local et de le synchroniser avec un dépôt de code distant partagé.

Itérations du projet « GSB – Visiteurs »

Sprint n°1

- Titre : « Organisation et spécifications »
- Description : organiser le travail en équipe et préparer le premier « livrable »
- Date indicative de remise du livrable : 22/03/2017
- Tâches principales :
    - Organisation de l'équipe
    - Mise en place des outils : Redmine, gestion des versions
    - Tests fonctionnels de l'application existante
    - Recensement des besoins en formation et des moyens de formation MVC en Java, JDBC, Swing, Javadoc, ...
    - Maquettage en Java
    - Migration de la base de données sous Oracle
    - Modélisation de la base de données (reverse engineering)

Sprint n°2

- Titre : « Affichage des visiteurs médicaux »
- Date indicative de remise du livrable : 05/04/2017
- Tâches principales :
    - Organisation de l'application en M.V.C.
    - Spécification, Codage et Tests de la fonctionnalité

    - Documentation et réalisation du livrable

Sprint n°3

- Titre : « Suivi des rapports de visite par les visiteurs »
- Date indicative de remise du livrable : 19/04/2017
- Tâches principales :
    - Spécification, Codage et Tests d'une fonctionnalité
    - Documentation et réalisation du livrable