Compte-rendu GSB (java)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Sommaire :

1. Introduction
2. Présentation du contexte
3. Analyse de l’existant
4. Description des besoins, des contraintes, des outils, des langages
5. Besoins
6. Contraintes
7. MCD
8. Arborescence

5. Travail réalisé

6. Difficultés rencontrées

7. Bilan

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Introduction

Création d’une application java permettant aux comptables de valider et de mettre en paiement les fiches de frais des visiteurs.

Présentation du contexte

Suite au précédent projet de gestion des fiches de frais via internet, le groupe s’est rendu compte que les justificatifs de frais pouvaient être égarés par des comptables. Ils ont donc décidé de créer une application locale afin que les comptables n’aient accès à l’application que sur leur lieu de travail.

L’application va donc permettre aux comptables de valider et de mettre en paiement les fiches de frais des visiteurs.

Analyse de l’existant

Il n’y a pas besoin de créer une partie visiteur pour cette application, seuls les comptables peuvent s’y connecter. Nous gardons également la même base de données.

Description des besoins, des contraintes, des outils, des langages

**Besoins**

**Définition de l’objet**

L’application doit permettre de présenter un suivi daté des opérations menées par le service comptable (réception des pièces, validation de la demande de remboursement, mise en paiement, remboursement effectué).

**Forme de l’objet**

L’application java sera accessible uniquement en local (sur le lieu de travail) sous forme d’interface.

**Accessibilité/Sécurité**

L’environnement doit être accessible uniquement aux comptables de l’entreprise.

Une authentification préalable sera nécessaire pour l'accès au contenu.

**Contraintes**

**Architecture**

L’application respectera l’architecture des scripts fournis concernant la gestion de l’enregistrement des frais engagés par les visiteurs.

**Ergonomie**

Les pages fournies ont été définies suite à une consultation. Elles constituent une référence ergonomique.

Des améliorations ou variations peuvent être proposées.

**Codage**

L’application sera développé dans le langage Java.

Les éléments à fournir devront respecter le nommage des fichiers, variables et paramètres, ainsi que les codes couleurs et la disposition des éléments déjà fournis.

**Environnement**

Développement en langage Java et création d’une application exécutable sans l’environnement de développement (IDE Eclipse).

**Modules**

L’application ne comporte qu’un seul module : enregistrement des opérations par les comptables.

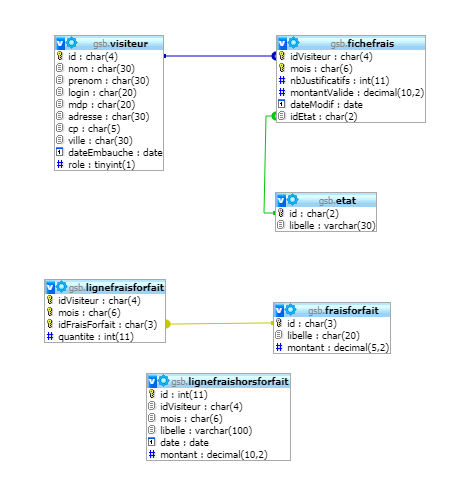
### **Documentation**

La documentation devra présenter l'arborescence des pages pour chaque module, le descriptif des éléments, classes et bibliothèques utilisées, la liste des frameworks ou bibliothèques externes utilisés, et les rapports de tests.

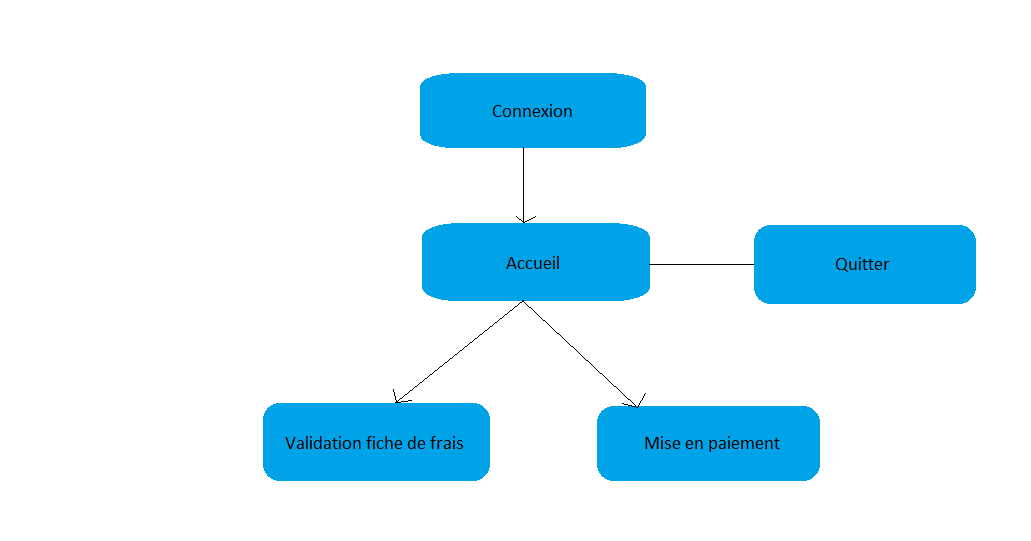
### **Responsabilités**

En plus des documents déjà attendus par GSB dans le cahier des charges (un système opérationnel, une documentation technique permettant un transfert de compétence et un mode opératoire propre à chaque module), mise en place, avant la phase de mise en production, des tests unitaires tout au long du travail (avec JUnit), ainsi qu’une phase de recette avec tous les tests nécessaires à la validation de la solution. Des rapports des tests seront présentés.

MCD :



Arborescence de l’application :



Travail réalisé

* Création d’une interface de connexion pour l’administrateur.
* Sécurisation du formulaire de connexion (cryptage des mots de passe)
* Création de l’interface d’accueil avec barre de menu (ActionListener pour chaque bouton de la barre de menu)
* Lors du clic sur la validation des fiches de frais, on reprend les fiches de frais du mois en cours. S’il y en a aucune, afficher “pas de fiches de frais durant ce mois”
* Choix pour l’administrateur de choisir le visiteur + la date de la fiche qu’il veut sélectionner, s’il n’y a aucune fiche à rembourser, afficher “aucune fiche de frais à valider”
* Affichage de l’état de la fiche, des éléments forfaitisés dans un tableau, et des éléments hors forfaits
* Insérer un nombre de justificatifs fournis et lors du clic sur le “valider”, passe la fiche en état “Validée et mise en paiement”
* Lors du clic sur le suivi de paiement, on affiche les fiches de frais à rembourser. S’il n’y a aucune fiche à rembourser, afficher “aucune fiche de frais à rembourser”
* L’administrateur sélectionne la fiche de frais à rembourser avec la possibilité de la télécharger en PDF (non fonctionnel)
* Lorsque l’administrateur valide les éléments forfaitisés ainsi que les éléments hors forfaits, il enregistre la fiche de frais qui passe dans l’état “Remboursée”
* Lorsque l’administrateur clique sur le bouton “Liste des fiches de frais”, cela affiche toutes les fiches de frais avec un bref résumé (nom du visiteur, état de la fiche, …)
* A la fin du mois, les fiches sont passées dans l’état “Saisie clôturée”
* Toutes les requêtes sont des requêtes préparées pour plus de sécurité et de fluidité
* Lorsque l’administrateur choisit “Déconnexion”, cela arrête la session et le retourne sur la page de connexion

Difficultés rencontrées

Nous avons rencontré principalement trois difficultés.

La première était d’utiliser seulement deux frames puis que des panels afin de ne pas ouvrir une nouvelle fenêtre à chaque fois. Nous avons dû faire de l’héritage sur la classe JFrame.

La deuxième difficulté était de faire apparaître les montants validés dans le suivi de paiement des fiches de frais. Nous avons dû créer une nouvelles méthodes et des nouveaux attributs afin de pouvoir récupérer ces montants. Pour les frais kilométriques, nous avons simplement récupéré la quantité et nous avons multiplié par le montant (0.62).

La troisième difficulté était de créer le PDF. En effet, c’était la première fois que nous devions créer un PDF en Java. Télécharger la fiche en PDF n’est pas encore fonctionnel.

Bilan

Ce projet fut plus facile à réaliser que le précédent projet en GSB. En effet, Rafaël et moi-même maîtrisons mieux le Java que le PHP. De plus, nous pouvions nous inspirer de ce que nous avions fait dans le projet précédent afin de réaliser celui-ci. Malgré les difficultés rencontrées, nous avons réussi à les résoudre et l’adaptation au format MVC ne nous a pas dérangé.

Les évolutions possibles seraient le design, ainsi que de rendre fonctionnel le bouton de génération de PDF.