



RAPPORT DU PROJET DE PROGRAMMATION ORIENTÉE OBJET

Ticket Me

Réalisé par:
Groupe F

BOUMALI Liza
THOMAS Antoine
ZUHAIR Huma
MARTIN Thomas
YANG Zilu
ZHANG Zimeng
MILLORD Clémence

Encadré par:
LABROCHE Nicolas

Nom du logiciel : **TicketMe**

I. Problème posé

Pour assurer le bon fonctionnement d'une entreprise ou d'une administration, il est nécessaire pour elle d'avoir un logiciel de support de dernière génération afin de résoudre ses problèmes au plus vite, et ainsi assurer une productivité plus efficace.

Le but d'un service de support est d'organiser toutes les demandes de problèmes multiples, émanant soit d'une demande d'un utilisateur du service (client, usager du service public ou employé interne à l'entreprise qui utiliserait un de ses logiciels).

Parce que les entreprises peuvent avoir un volume élevé de requêtes, elles ont besoin d'un outil informatique puissant permettant de gérer efficacement le cycle de vie de chaque requête (ticket).

Chaque ticket correspond à un problème et il appartiendra au technicien concerné de résoudre ce problème dans les plus brefs délais via l'interface fournie par le logiciel de support ou logiciel de "ticketing".

II. Solution

A. Les fonctionnalités du logiciel

Nous proposons un logiciel de support efficace, fiable et intuitif qui sera utilisé par trois types d'utilisateurs : Un "user" qui soumettra une question, un technicien qui sera chargé de résoudre le problème, enfin un "admin" qui sera chargé de manager les équipes de techniciens et pourra configurer le logiciel pour l'adapter au service en question. Le logiciel dispose de plusieurs fonctionnalités :

- Chaque ticket créé par l'utilisateur donne un aperçu du problème grâce à un titre et un niveau d'urgence. L'utilisateur peut par ailleurs lier à son ticket une image telle qu'une capture d'écran pour permettre au technicien d'identifier son problème plus précisément. Il renseigne pour chaque ticket un descriptif détaillé du problème et mentionne le type de problème.

- Le technicien reçoit les tickets sous une forme de liste et peut résoudre chaque ticket selon le type de problème auquel son équipe a été affectée. Lorsqu'il résout le problème il indique la solution de manière détaillée et remplit le type de solution. Ce type de solution (code de completion) est important pour le service : Par exemple, si le service s'aperçoit que beaucoup de tickets avaient comme mode de résolution "training", cela signifie non pas que le service était défaillant, mais qu'il n'était pas intuitif et qu'il faut donc renforcer la formation au logiciel ou améliorer son ergonomie.

- Chaque utilisateur est averti que son ticket est résolu par mail. Il peut alors noter le technicien qui s'est occupé de son ticket.

- Une partie statistique permet au technicien d'accéder à sa moyenne, de visualiser les statistiques relatives aux tickets, à leur mode de résolution, etc.

- Une FAQ est un système de chat entre le technicien et l'utilisateur devant être implémenter.

-Le “super ticket” est un ticket créé par le technicien qui permet de résoudre plusieurs tickets qui ont la même cause simultanément, permettant pour le technicien de gagner du temps.

-L’admin crée les groupes de techniciens et modifie les catégories de problème en fonction de l’évolution du service.

B. La technologie sélectionnée

Pour pouvoir proposer ce logiciel, nous avons décidé de le coder en JAVA, en programmation objet. L’héritage est nécessaire en terme d’efficacité, en ce que user, technicien et admin héritent d’une même classe (personne) ou encore que super ticket hérite de la classe ticket.

L’interface utilise la librairie SWING. C’est une librairie que nous connaissons de part notre première année en informatique.

Javamail est l’API utilisée pour envoyer des mails.

III. Les verrous

Au niveau de la modélisation des classes, on s’aperçoit que rien ne différencie l’user, le technicien et l’admin en terme d’attributs et de méthodes. Ils dérivent tous d’une même classe “personne”.

De plus ils ne possèdent pas de méthodes propres car elles sont contenues dans d’autres classes, comme la classe ticket. Or on veut par exemple que seul un technicien puisse clôturer un ticket ou que seul un user puisse noter un technicien. L’encapsulation a priori ne permet pas de résoudre ce problème. En réalité c’est grâce au typage que nous sommes parvenu à le surmonter : En indiquant en paramètre pour chaque méthode un type précis. Ansin la méthode pour noter un technicien devra avoir en argument un objet de type user, ce qui l’a rend donc inutilisable par un autre type de personne. La modélisation des classes reste donc pertinente.

Javamail nécessite un mot de passe et un identifiant pour que le logiciel envoie des mails. Le projet étant publié sur github, il fallait trouver une méthode pour masquer le mot de passe. On a donc créé un fichier config.properties contenant les identifiants et les mots de passe. Gitignore permet de masquer les fichiers properties. Une personne qui importe le projet depuis github devra donc créer son propre fichier config contenant ses propres identifiants.

Pour les statistiques, JAVA FX a été choisi comme librairie. Mais problème de compatibilité avec Eclipse et le JDK 12 (javafx a été supprimé après le JDK 8.0), il fallait chercher une version récente. Soucis d’incompatibilité. Donc retour vers SWING : Absence de librairie complète comme sur JAVA FX en matière de graphisme de 2D.

En matière de front SWING est assez archaïque en matière de layout et a nécessité un temps d’adaptation.

Pour importer une image et la lier à un ticket l'importation d'une image en BLOB pose des problèmes : L'image ne s'affiche pas ou message d'erreur si la taille de l'image est trop importante. Donc obligation de les importer en LONGBLOB, ce qui implique une taille plus grande donc une BD plus grosse. De ce fait l'exportation de base de données concernant les images prend du temps.

Pour l'email, le système de sélection des tickets de la fenêtre Résolution_Ticket maintenant c'est de cliquer sur le ticket et afficher les détails du ticket, il est plus pratique de récupérer le nom d'utilisateur pour sélectionner son email. Cependant, le système précédent en saisissant le numéro de ticket rendait plus compliqué l'obtention des noms d'utilisateur et des noms de ticket.

Enfin par manque de temps ou de moyens, les parties tchat, super ticket et admin ont été abandonnées.

IV. La répartition des tâches

Base de donnée:	ZUHAIR Huma et YANG Zilu
Back:	ZHANG Zimeng et YANG Zilu
Front:	THOMAS Antoine et MARTIN Thomas
Statistiques:	BOUMALI Liza
Importation/Exportation d'images:	ZUHAIR Huma