



Réalisation d'une IHM Java pour le projet de la Qualité de Service.
Département informatique – 2014

Diffusion du rapport (y compris en version électronique) :

- ☐ Non autorisée (le rapport ne sera pas diffusé, mais archivé par obligation légale)
- ☐ Autorisée en interne à l'INSA
- ☐ Autorisée en interne et en externe

Auteur : Boschet François
Maître de stage : Lamie Frederic
Correspondant INSA : Babel Marie

Remerciements

Je remercie tout d'abord Mr Frédéric Lamie, mon maître de stage, pour sa disponibilité et son aide quand j'étais en galère et pour avoir bien voulu passer du temps pour m'expliquer comment fonctionne le SI. Je tiens aussi à remercier tous les gens (du backoffice) qui ont été sympathique et ont bien voulu m'expliquer quel était leur travail et les Je remercie aussi Marc Gloanec pour m'avoir donné un sujet de stage intéressant. Remercie aussi Roselyne Bordas qui m'a vraiment aidé à trouver ce stage.

Table des matières

Remerciements	iii
1 FORTUNEO	1
Historique	1
L'organisation	2
2 Présentation du sujet	3
Le projet de la QS	3
Là où j'entre en jeu	4
3 La réalistaion	5
Etape 1 : Le Cahier des Charges et les spécifications	5
Etape 2 : Préparation	8
Etape 3 : Développement	9
4 Conclusion	13
Les apports à l'entreprise	13
Ce que j'en retire	14
5 Résumé	15
Anglais	15
Français	15
A Anciens Rapports	17



FORTUNEO

Historique

Fortuneo est un établissement bancaire français créé en 2000 et qui propose des services de bourse en ligne. A l'origine, Fortuneo ne proposait que de la gestion de portefeuille d'action. L'entreprise devient une filiale du group Crédit Mutuel Arkéa en 2006 et devient une véritable banque en ligne en 2009. A l'heure actuelle, Fortuneo est une banque a part entière, elle propose toute une gamme de produits comme par exemple les comptes courants, les assurances vies et toujours la gestion de portefeuille boursier.

L'organisation

Durant mon stage, j'ai travaillé pour le domaine MOE¹ qui fait parti du service informatique. La MOE est composé des employés qui conçoivent et réalisent les outils informatique et logiciels. Leur rôle est de concevoir développent et maintiennent les outils informatiques et logiciels utilisé par les employés et les particuliers, comme par exemple le [site internet](http://fortuneo.fr): fortuneo.fr.

La maitrise d'oeuvre réalise les projets proposés par la MOA². Le rôle de la maitrise d'ouvrage est d'identifier et de rassembler leurs besoins pour rédiger un cahier des charges. Ils établissent ensuite un calendrier de projet et fixe un budget pour réaliser le projet. Ce projet est ensuite donné à la MOE, qui établie les spécifications techniques et réalise ensuite le développement. La MOA a aussi le rôle de valoriser le produit qu'ils ont commandé. La MOA est le client de la MOE, c'est elle qui décide si un projet est terminé et qu'il peut être livré.

moe -> conception + realisation moa -> valoriser le produit qu'ils demandent

¹ Maitrise d'oeuvre

² Maitrise d'ouvrage

2

Présentation du sujet

Le projet de la QS

Courant 2013, Fortuneo a initié un projet sur la qualité du service de la souscription en ligne. Le projet avait pour but de :

pouvoir mesurer de manière objective le niveau de qualité de service sur les aspects applicatifs, de manière exhaustive, et non plus se mettre en attente de remontés potentielles de nos prospects/clients lors de ces incidents

Le projet a été réalisé de manière à répondre aux besoins suivant:

- Superviser en temps réel la production de manière automatique afin de s'assurer que toute la chaîne de souscription est disponible de bout en bout
- Remonter en temps réel à la production les indisponibilités des services supervisés
- Alerter rapidement (temps réel) de manière automatique, et à partir d'un certain seuil, l'assistance Fortuneo afin de communiquer, au Service Clients et au Service Commercial, l'indisponibilité ou la sous exposition d'une souscription

- Générer des rapports (quotidien, hebdomadaire, mensuel) sur les indisponibilités des services et sur le parcours utilisateur
- Remonter les informations des prospects lors d'une anomalie à destination du Service Commercial.

Pour ce faire, plusieurs outils et applications ont été développées. Notamment une sonde qui teste le parcours client régulièrement, ce qui permet d'avoir des informations régulières sur la disponibilité de la chaîne de souscription. Pour pouvoir référencer les différents événements du parcours d'un utilisateur, il a fallu aussi mettre en place des outils qui répertorient l'intégralité du parcours de l'utilisateur afin de savoir si son parcours s'est bien déroulé ou si une erreur est survenue et à quel moment. Toutes les données sont déversées dans une table d'une base de données Oracle. Pour exploiter les résultats, il a été mis en place un système de rapports (annexe A.1) qui permet d'obtenir le nombre d'abandons entre deux étapes de souscriptions. L'abandon est dû, soit à un arrêt de la souscription par le prospect, soit à une erreur fonctionnelle.

Là où j'entre en jeu

Le système de supervision de la QS fonctionne très bien et a permis de corriger plusieurs problèmes au sein du processus de souscription. Cependant, le principal souci de la solution actuelle est l'accessibilité des résultats. Les rapports qui sont générés étaient trop peu lisibles, assez lourds et ne proposaient que de visualiser le taux d'attrition (ou d'abandon). Pour obtenir d'autres indicateurs, comme la répartition des erreurs au cours du temps en fonction des types d'erreurs, la MOA devait faire une requête SQL puis un traitement du résultat dans un tableur (comme Excel). L'utilisation d'un tableur Excel pour traiter un aussi grand volume de données (~ 400 000 lignes pour le mois de Janvier 2014) n'est pas efficace, et créer des graphiques et des tableaux récapitulatifs peut être très long.

Mon but est alors de réaliser une Interface Homme-Machine sous la forme d'une Application Web qui permettrait d'afficher les différents indicateurs de la qualité de service sous la forme de tableaux et de graphiques dynamiques.

3

La réalistaion

Etape 1 : Le Cahier des Charges et les spécifications

L'expression du besoin par la MOA

La MOA voulait pouvoir avoir acces à trois grands indicateurs de la qualité de service: 1. La volumetrie d'erreurs 1. La taux de rupture de souscription 1. Le percentile 90 du temps de réponse utilisateur

Tous ces indicateurs devant pouvoir être visualisé sur des périodes de temps variables. Par exemple, pouvoir voir l'évolution d'un indicateurs, sur un an avec un pas de un mois, sur un mois avec un pas d'une journée, et sur une journée avec un pas de 15 minutes.

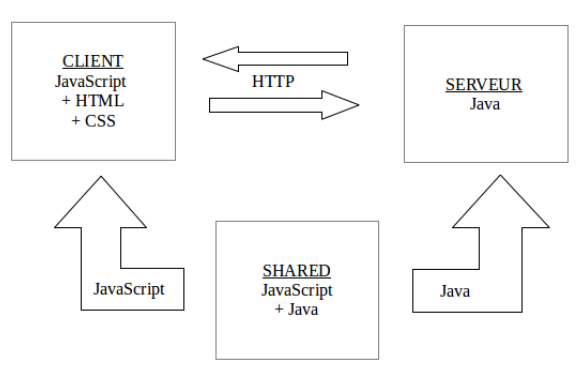
Le premier indicateur, la volumétrie, correspond au volume d'erreurs sur une période de temps donné. Il permet de cibler des pics d'indisponibilité et de cibler par la même occasion quel(s) service(s) était indisponible, cela permet de gagner du temps dans l'exploitation des logs du service en question.

Le second indicateur, le taux de rupture, est le rapport entre le nombre de personne qui ont atteints la fin du tunnel de souscription et le nombre de personne qui ont eu une erreur bloquante (site qui ne reponds pas, un service indisponible, ...). Cet indicateur est indépendant de l'abandon "conscient" de l'utilisateur.

Le troisième indicateur est le percentile 90 du temps de reponse utilisateur. Il s'agit en fait de la valeur pour laquelle 90% des temps de réponse de l'échantillon d'utilisateurs est inferieur a cette valeur. Cet indicateur permet de savoir le moment où le site est surchargé et si cette surcharge est récurrente ou simplement sporadique.

Presentation des technologies

Figure 3.1: Schematisation de GWT



Java

Le langage de programmation utilisé principalement par le groupe Arkea/Fortuneo est principalement le Java. La maniere la plus commune d'utiliser Java pour developper des Applications Web c'est d'utiliser la technologie Apache/Tomcat qui est un serveur http qui gere les conteneurs de servlet (?).

Pour développer une application qui sera déployé sur un serveur Apache/Tomcat, j'ai du utiliser le framework¹ GWT (figure 3.1), développé par Google. Ce framework demande de séparer le projet en trois parties, la partie serveur, la partie client et une troisième partie partagée par le client et le serveur que l'on nommera "Shared". Le plugin va donc traduire la partie client en JavaScript, HTML et CSS pour être interprété par un navigateur web. Il va aussi compiler la partie serveur en java pour être executé par le serveur Tomcat, et il va compiler la partie Shared à la fois en Javascript et en java pour pouvoir être interpreté par les deux entités Client et Serveur.

¹ Framework : Ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes d'un logiciel.

Oracle

La table qui contient les informations qui me permettront de calculer les différents indicateurs demandés par la MOA est stockée dans une base de données relationnelle gérée par un moteur Oracle. Ce gestionnaire de base de données est beaucoup utilisé dans les grandes entreprises notamment parce qu'il est très efficace pour traiter des données volumineuses.

Oracle Database est aussi intéressant puisqu'il possède de nombreux outils pour s'interfacer du code, Java notamment.

Etape 2 : Préparation

Archétypes

La première difficulté de ce stage est venu du framework de développement. Pour cette application, j'ai du utiliser un framework du nom de Archétypes, qui est le framework GWT auquel on a greffé la charte graphique du groupe Crédit Mutuel Arkéa. Puisque la charte graphique est déjà intégré, cela permet au developpeur de se concentrer sur le reste du code et ne pas passer trop de temps a s'occuper du design.

Néanmoins, il a été assez difficile de prendre complètement en main ce framework car il est fourni avec une documentation très réduite, il faut aller chercher les noms des méthodes "à la main" dans le code déjà généré. C'est vraiment une étape assez longue mais qui montre l'intérêt de documenter le code.

Les requêtes SQL

La deuxième partie du travail effectué en amont du développement a été de préparer les requêtes qui allaient être utilisés pour attaquer la base de données. La table était stocké dans une base de données relationnelle et les requetes sont ecritent en langage sql. L'écriture des a m'a toutefois permis d'apprendre a maitriser des fonctions sql/oracle comme la fonction *PERCENTILE_DISC* qui permet de recupere le n-ième centile d'un champ d'une requete. Ces fonctions sont très pratique car le traitement de plusieurs centaines de milliers de lignes est plus efficace sur le serveur au moment de la requête plutôt que dans l'application.

Etape 3 : Développement

Partie Serveur

La partie serveur, schématisé par la figure 3.2 est composé de trois partie :

1. Les servlets, qui sert a formater la reponse à la requete grace à des appels à un ou plusieurs business. J'ai choisi d'écrire une servlet par page pour éviter d'avoir des classes trop grande.
2. Les classes Business, qui servent à faire le lien entre les servlets et la DAO. Dans mon projet il s'agit surtout de mappers qui servent à formater les réponses des appels a la base de données en un format intelligible par le client
3. Les classes DAO² qui sont les classes qui contiennent les accesseurs permettant d'attaquer différents service, comme, dans mon projet, une base de données.

Découper la partie serveur en trois packages permet de bien séparer les différentes opérations, d'avoir un programme structuré et donc d'améliorer les phases de debug et de maintenance. Cela passe aussi par une rigueur dans la gestion des exceptions et des logs.

Le rôle des servlets est de recevoir la requête ainsi que les paramètres de la requête pour appeler la bonne classe Business et de renvoyer la réponse approprié au client. La Servlet reçoit la requête par l'intermédiaire d'une classe nommée ServletMapping qui fait le lien entre les url et les servlets associés.

² Device Access Service ou Object d'accès aux données.

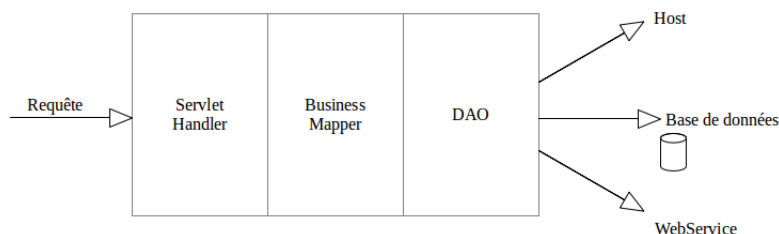


Figure 3.2: Schematisation de la partie serveur

Les classes Business quand à elles, servent à formater les résultat de leurs appels aux classes DAO pour qu'ils soient envoyés au Client par les Servlets. Dans mon cas il s'agissait principalement de transformer des resultat de requêtes SQL en une table de hachage compréhensible par le Client. Les retours des requêtes SQL sont sous la forme d'une liste Java qui contient des Map. Chaque Map contient une ligne du retour de la requete SQL, il faut donc extraire les information de cette map et les trier pour pouvoir les exploiter dans la partie client.

La dernière partie du Serveur sont les classes de DAO. Il s'agit des classes qui font les appels à la base de données. Les informations relatives à la connexion (adresse du serveur, identifiants, pool de connexions, ...) sont stockés dans un fichier de conf. Ce sont ces classes qui attaquent les différentes sources de données. Dans mon cas ces classes servent à effectuer les requêtes SQL et à retourner le résultat de ces requêtes à la classe business qui l'appelle.

Partie Client

Etant donné que j'utilise le Framework Archétype, je dispose déjà d'un système de gestion des pages, d'un menu à remplir avec mes nouvelles pages et de la charte graphique interne. Il me reste donc à développer les pages en elles même et les appels aux Servlet.

La partie client se compose de deux grands blocs et d'un bus d'évt. Il y a les pages, représentés par des '*Workbenches*' qui sont des widgets et qui sont affichés dans la partie centrale du '*mainWidget*' qui est en fait l'implémentation de la charte graphique. Le changement de workbenches se fait au clique dans le menu, il y a seulement la partie centrale (le Workbench) qui est mis à jour.

Pour visualiser les 3 indicateurs demandé, il a été choisi de faire 6 pages.

La première page (Annexe B.1) est celle contenant le graphique de la volumétrie d'erreur. Il s'agit simplement d'un graphique avec en abscisse les dates et en ordonnée le nombre d'erreurs. L'utilisateur a aussi la possibilité de naviguer au sein du graphique. C'est à dire, lorsque l'on visualise la volumetrie par tranche d'un mois, le clique sur le mois permet d'avoir le graphe détaillé sur la journée. Et le clique sur une journée amène à la page n°5 qui affiche tous les messages d'erreurs de la journée sélectionnée. Il est aussi possible de filtrer les erreurs pour avoir des résultats plus précis.

La seconde page (Annexe B.2) est une version détaillé de la première. Les erreurs

sont, cette fois ci, affichés dans un tableau qui les trie par date et par type d'erreurs. La lecture du tableau n'étant pas très facile, il m'été demandé de pouvoir exporter les résultats au format csv via un bouton. Les données au format csv pouvant être ensuite traitées par un tableur.

La troisième page (Annexe B.3) représente le taux de rupture de souscription du site. Il s'agit de la page la plus attendue car elle permet d'avoir une vision d'ensemble de l'évolution de la qualité de service des tunnels de souscriptions. Comme pour la première page, ce graphique permet, au clic, de passer d'une visualisation par mois à une visualisation par jour et à une visualisation par quart d'heures. Le taux de rupture (en vert) est le rapport entre le nombre de souscriptions (en bleu), et le nombre de ruptures (en rouge). La moyenne de ce taux est représenté par une ligne rose.

La quatrième page (Annexe B.4) représente le percentile 90 du temps de réponse utilisateurs. Il s'agit de la valeur pour laquelle 90% des utilisateurs ont un temps de chargement de la page inférieur à cette valeur. Il est représenté par une ligne bleu, la moyenne de ce percentile est représenté par la ligne rose.

La cinquième page (Annexe B.5) affiche sous la forme d'un tableau le récapitulatif des erreurs d'une journée. A la différence de la version détaillé de la volumétrie, cette page affiche aussi le message d'erreur généré au moment de l'erreur. Cette page est pratique pour identifier précisément la classe qui a causé l'erreur et voir si cette erreur est récurrente ou pas.

Il y a aussi une sixième (Annexe B.6) page qui sert de rappel des différents code d'erreurs. Les erreurs sont stockés dans la base de données avec un code unique en fonction du service qui a causé l'erreur. Il est intéressant de pouvoir avoir à disposition la signification de ces "*codes d'erreurs*".

Au fur et a mesure du developement, on m'a proposé quelques idées d'amélioration et d'ajout, comme par exemple l'ajout d'un "moteur de recherche" commun à toute les pages qui permet de factoriser le code et aussi permet à l'utilisateur de ne pas avoir a copier plusieurs fois les memes parametres. Le moteur de recherche se compose de deux champs de dates pour définir les bornes de la recherche, un menu déroulant pour choisir le pas de visualisation (ex: Mensuel, Journalier, ...), et deux autres menu déroulants pour définir sur quels contrats faire la recherche. Pour les pages 1 et 5, j'ai ajouté un autre menu déroulant pour pouvoir choisir précisément quels types d'erreurs afficher.

4

Conclusion

Les apports à l'entreprise

Ce que j'ai fait et à quoi ça sert

Grace a ma réalisation, l'entreprise Fortuneo possède désormais une application qui leur permet d'afficher les données relatives à la souscription des clients d'une manière plus intuitive que les simples rapports qui étaient générés. Ainsi qu'un moteur de recherche qui leur permet de moduler l'affichage des indicateurs.

Ce qui'il reste a faire et ce qui a été commencé

Le projet de la QS a beaucoup évolué durant ma période de stage. Au début il s'agissait de monitorer la souscription aux différents produits de Fortuneo. Par la suite il a été envisagé d'étendre le processus d'analyse des erreurs à d'autres parties du site internet. Il y reste donc du travail d'instrumentalisation de certaines parties du site ainsi que quelques modifications de mon interface graphique pour pouvoir afficher les résultats.

Il reste aussi une partie importante de la phase de développement que sont les tests unitaires. D'un commun accord avec mon maitre de stage, nous avons décidé de

ne pas écrire de tests formels. Ce choix a été fait par manque de temps. Nous avons préféré finir le développement du projet dans les temps plutôt que d'écrire des tests unitaires. L'application néanmoins fournit des résultats cohérents avec les exploitations manuelles des données.

Ce que j'en retire

J'ai pu me perfectionner en java, découvrir la programmation Web en java au travers du framework GWT et découvrir la programmation événementielle. J'ai toujours été intéressé par le monde des réseaux et de la programmation. Rédiger du code java pour faire une application Web a vraiment été une expérience intéressante et enrichissante.

Ce stage m'a permis d'apprendre comment fonctionne la gestion de projet dans une grande entreprise, de la réception du Cahier des Charges, en passant par l'édition des spécifications, jusqu'à la livraison en environnement de recette.

regard critique

5

Résumé

Anglais

Et advicatur namet.

Français

Tout au long de mon stage, j'ai pu apprendre le fonctionnement de GWT, un Framework Java, qui m'a permis de développer un application web permettant de visualiser des indicateurs de la qualité de service du site internet de Fortuneo : <http://www.fortuneo.fr>. Au delà des aspects techniques, j'ai pu observer de le fonctionnement du tandem MOE/MOA dans une grande entreprise.



Anciens Rapports

Attrition Journée

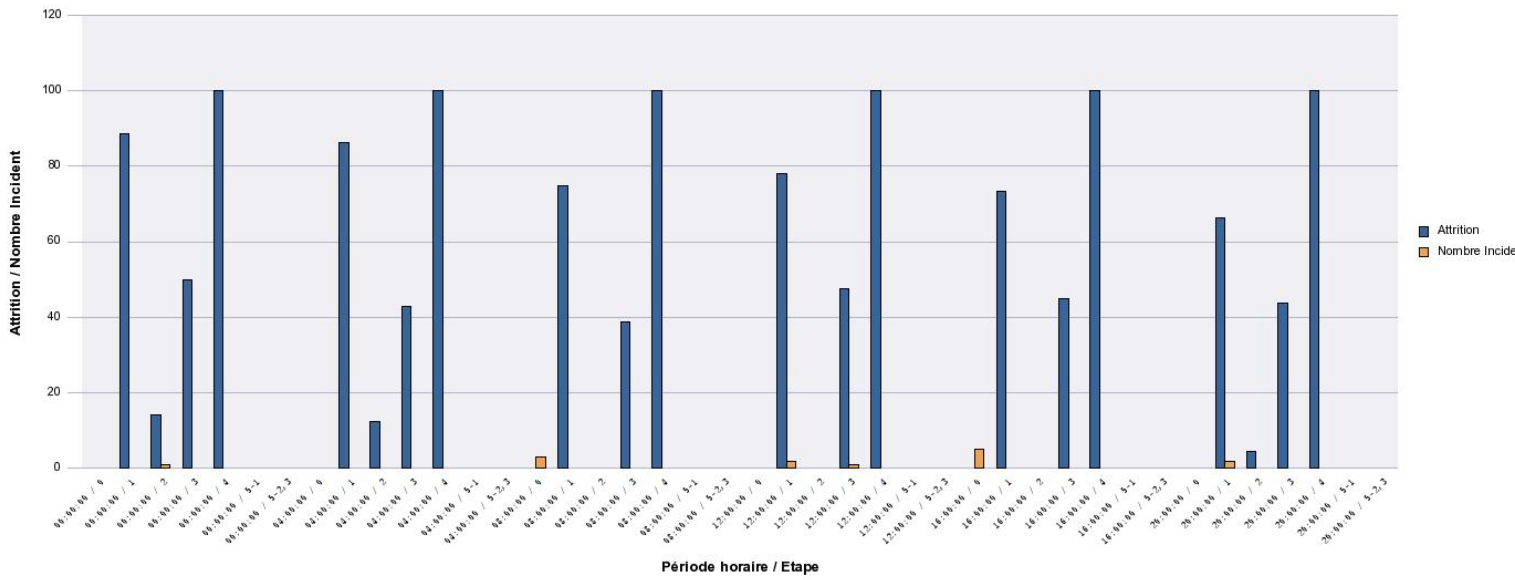


Figure A.1: Les anciens rapports

B

Page du site internet

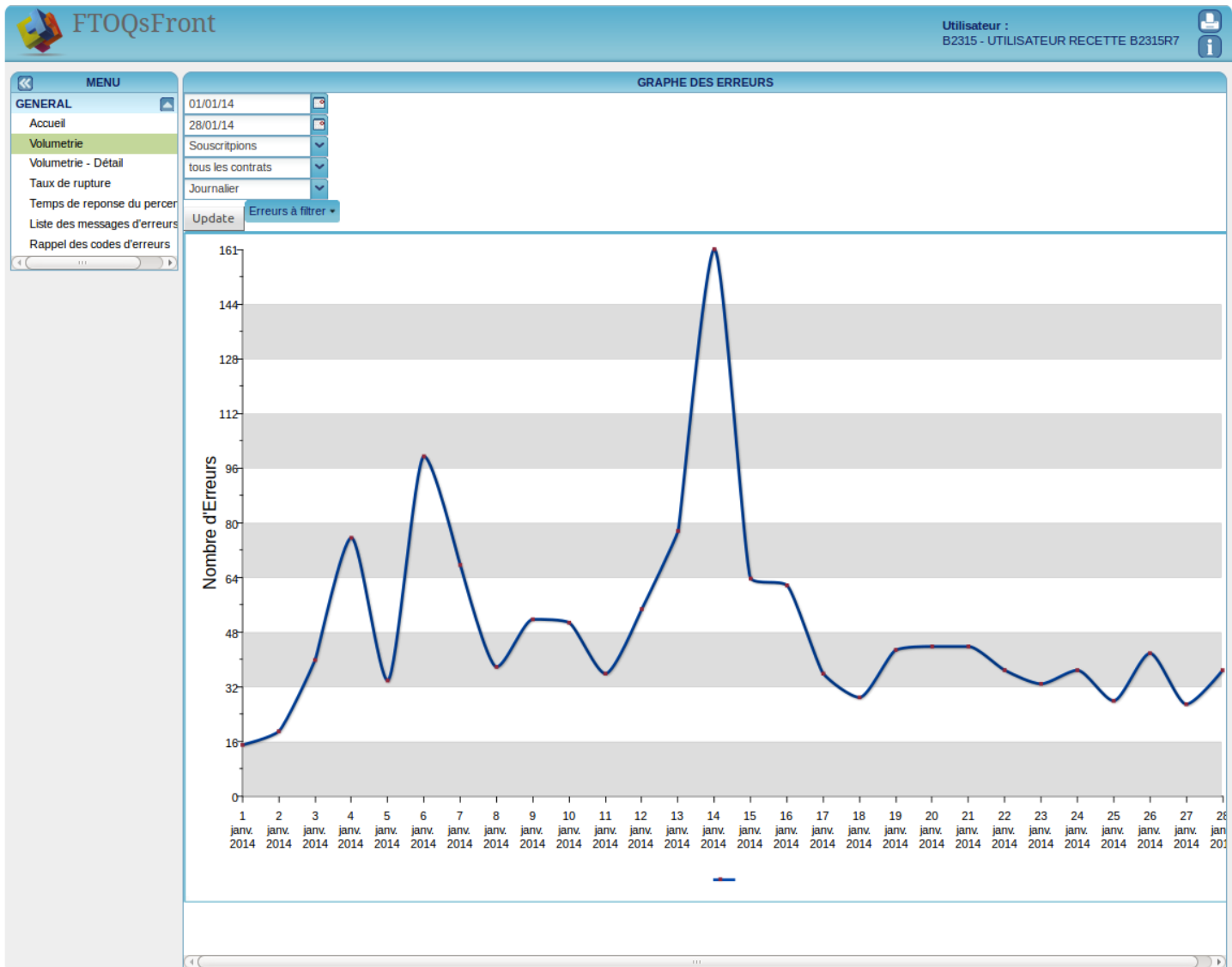


Figure B.1: La page de la volumétrie sous la forme d'un graphique

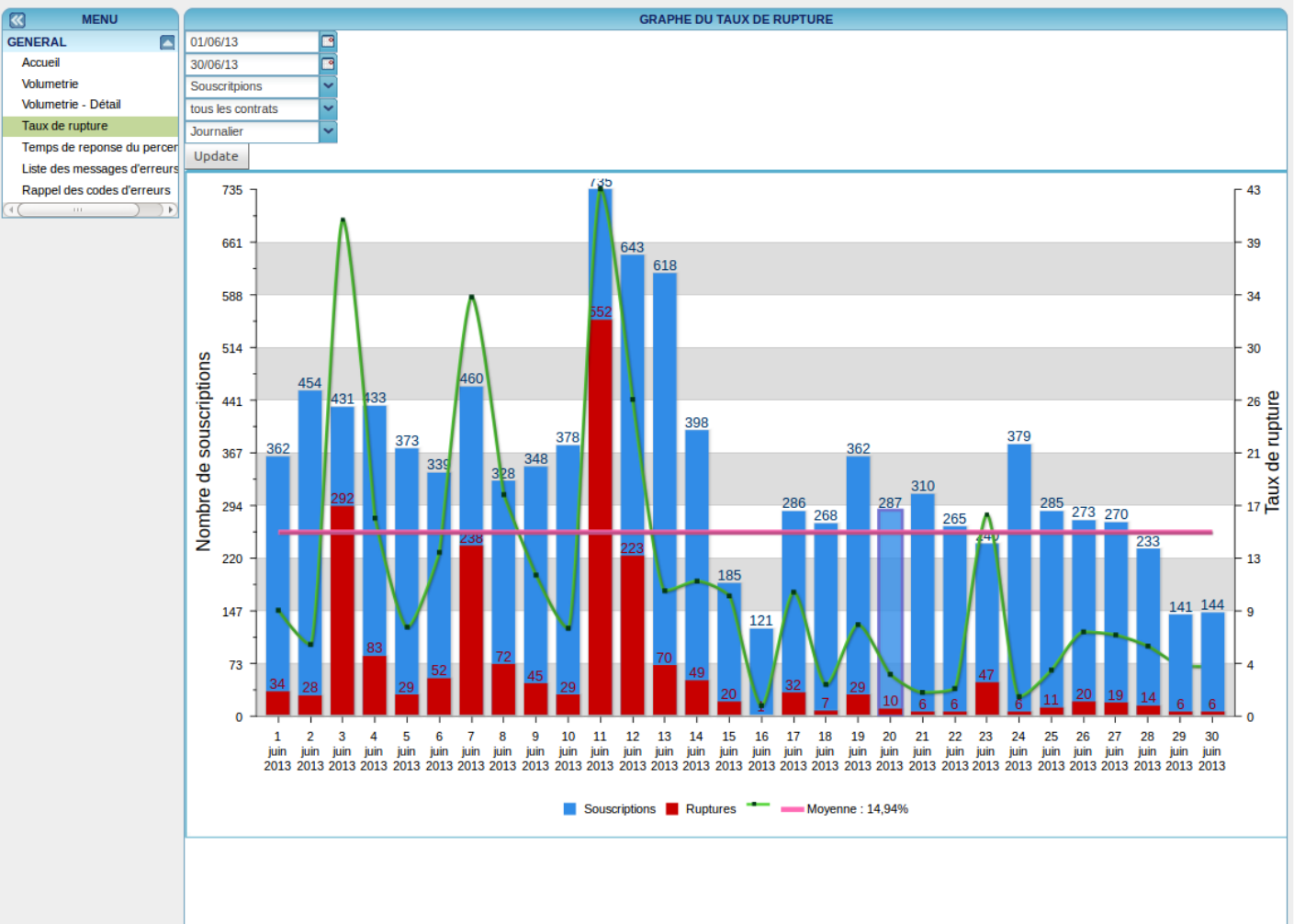




Figure B.3: La page du taux de rupture



Figure B.4: La page du temps de réponse utilisateur


FTOQsFront

Utilisateur :
 B2315 - UTILISATEUR RECETTE B2315R7



MENU

GENERAL

Accueil
 Volumetrie
 Volumetrie - Détail
 Taux de rupture
 Temps de reponse du percer
 Liste des messages d'erreurs
 Rappel des codes d'erreurs

RECAPITULATIF DES CODES D'ERREURS

Numero d'Erreur	Description de l'erreur
0	Pas d'erreur
10	Erreur Technique
100	Problème Host
210	Problème Panier
220	Problème Pacman
230	Problème Syvie - recupererContratAssuranceVie
240	Problème Backend
260	Problème SouscriptionBack (réservé)
270	Problème CUSTKN (réservé)
290	Problème Backend
310	Problème Conse
410	Erreur Livret WS
411	Erreur CAT WS
510	Problème Suravenir - recupereListeContrats
511	Problème Suravenir - recupereUnContratIdtExt
512	Problème Suravenir - EditionNotice
610	Problème Servau - getInfosAccess
710	Problème Savheol - syntheseTousComptes
811	Problème Topaze Adresse - normaliserAdresse
812	Problème Topaze Commune - rechercherListeCommune
814	Problème topaze Personne - getDonneesRecherchePT - enregistrerDonneesPT
813	Problème Topaze Personne - getDonneesRechercheAD - enregistrerDonneesAD
815	Problème Topaze Personne - getDonneesRecherchePP
816	Problème Topaze Personne - enregistrerDonneesPP
900	Probleme Serveur Mail
1010	Problème Odet - Génération PDF
1011	Problème Odet - Signature PDF

Figure B.6: La page du récapitulatif des codes d'erreurs