



Compte-rendu des Tests



Table des matières

1 Présentation.....	3
2 phase 1.....	3
2.1 Réunion technique : MOE – Chef de projet.....	3
3 Test unitaire de la phase 2.....	3
3.1 présentation :.....	3
3.1.1 vérification du fonctionnement de la connexion.....	3
3.1.2 Vérification de l'affichage du menu.....	4
4 Test unitaire de la phase 3.....	4
4.1 Pour un utilisateur cela correspond au choix :.....	4
4.2 Pour un modérateur cela correspond au choix :.....	4
5 Test unitaire de la phase 4.....	4
5.1 Vérification sur le rendu d'une recherche.....	4
5.2 réunion qualité (MOA, MOE et le chef de projet).....	5
5.3 reprise des tests unitaires de la phase 4 en reprenant.....	5
6 Réunion technique.....	5
6.1 Reprise des tests unitaires de la phase 4.....	6
6.2 Réunion technique.....	6
7 Tests unitaires de la phase 5.....	6
7.1 Se Connecter avec le compte de Lenaïc pour faire :.....	6
7.2 Se connecter avec le nouveau compte créer faire :.....	6
7.3 Se connecter avec le compte de Lenaïc pour :.....	6
7.4 Réunion qualité :.....	6
7.5 Réunion technique.....	7
7.5.1 1st cas traiter.....	7
7.5.2 2nd cas traiter.....	7
7.5.3 3th cas traiter.....	7
8 Conclusion.....	7



1 Présentation

Pour la gestion de ce projet, on a utilisé un cycle de développement sous une méthode Agile. La mise en place des tests se sont faits durant tout le développement, du programme. Ainsi, lorsque des erreurs sont apparues lors des séries de test, des méthodes ont été mis en place pour les corriger dans délais très court, en mettant en avant le point de vu du client. Parfois, les tests ont mis en évidence des problèmes importants du côté client, ce qui a déclenché des réunions du comité qualité ou du comité technologique.

Vous trouverez dans le fichier : [incrémentation Publio.sql](#), les explications techniques des tests ainsi qu'une série de lignes de code à insérer dans la base de donnée.

Ce compte rendu sert pour confirmer, le respect intégral de cahier des charges, des décisions prises par le comité technique et par le comité qualité durant chaque phase de développement.

2 phase 1

2.1 Réunion technique : MOE – Chef de projet

Suite à la mise en place de contraintes sur les tables, le développeur procède à la création d'un Numéro d'Identification (ID), ce numéro sera attribué, par la Base de donnée (SERIAL), pour chaque table avec mise en place de clé primaire.

Ainsi, on obtient une facilité de programmation pour les visualisations (recherche plus simple sur BD).

La mise en place de clés étrangères dans les tables est nécessaire pour : autoriser, demander, lier, pret_ret, avec comme clé primaire : utilisateur, livre.

Pas d'incidence financier, et un peu moins de latence dans l'application.

3 Test unitaire de la phase 2

3.1 présentation :

3.1.1 vérification du fonctionnement de la connexion.

- L'utilisateur se connecte avec un nom valide, un prénom valide et un mot de passe valide
- L'utilisateur avec un nom valide, un prénom valide et un mot de passe invalide mais avec une phrase valide
- Simulation l'utilisateur a été créé par un modérateur, donc c'est ça 1er connexion, le mot de passe et la phrase ont une valeur NULL

La conclusion du test nous apporte que, le processus est conforme au cahier des charges avec un mot de passe que l'on peu utiliser que 2 fois et ensuite une question qui permet de réinitialiser le mot de passe.



Mise en place d'une série de tests d'erreur pour vérifier le fonctionnement de la connexion :

- Sur invalidité du mot de passe et de la phrase => retour sur la page de connexion.
- Sur la confirmation du nouveau mot de passe => accès au menu.
- Sur la confirmation du lieu de naissance => bouton modifier mot de passe, création d'un nouveau mot de passe. Passage par une phase d'enregistrement avant d'accéder au menu.

Les corrections sont :

La création d'une page tampon, c'est la page JSP, « verif_donnees », elle permet de séparer la phase de connexion et la phase de choix dans le menu, bien pratique pour éviter les retours en arrière, lorsque l'on navigue sur d'autre page. Cette page sert de point de départ ou de repère pour l'utilisateur. On y présente la carte d'identité de l'utilisateur.

3.1.2 Vérification de l'affichage du menu

- Si Lenaïc se connecte affichage du « menuplus » (module utilisateur et modérateur) = OK
- Si Cyril se connecte affichage du menu (seulement module utilisateur) = OK

Fin test unitaire de la phase 2.

4 Test unitaire de la phase 3

Vérification de l'apparence pour les sorties. Ce sont toutes les visualisations qui apparaissent au menu.

4.1 Pour un utilisateur cela correspond au choix :

Choix 3 : visualiser l'ensemble des livres = OK

Choix4 : visualiser les livres en cours d'emprunt = OK

Choix8 : visualiser les demande de prêt = OK

Choix10 : visualiser les demandes d'autorisation = OK

4.2 Pour un modérateur cela correspond au choix :

Choix15 : visualiser les livres de sa bibliothèque = OK

Choix18 : visualiser les demandes de prêt à traiter = OK

Choix20 : visualiser les demandes d'autorisation à traiter = OK

Observation du MOA : Les tableau ne sont pas très jolie ! Manque de couleur.

Fin test unitaire de la phase 3

5 Test unitaire de la phase 4

5.1 Vérification sur le rendu d'une recherche

Pour un utilisateur, cela correspond au choix :



Choix5 : rechercher les livres empruntés = OK

Choix6 : rechercher les informations d'un livre = si il n'y a pas de livre le programme continue et soulève une erreur.

Observation du MOA :

Choix5 = le terme 'rechercher' est remplacé par le terme visualiser dans le menu.

Il n'est pas possible de choisir un livre, si il n'y en a pas !

Arrêt test unitaire phase 4 : suite aux 2 remarques du MOA pour la mise en place d'une réunion.

5.2 réunion qualité (MOA, MOE et le chef de projet)

La satisfaction du besoin est la priorité. Une modification dans le fichier CSS, afin d'apporter un peu de couleur dans les tableau de visualisation.

Le chef de projet en profite pour appliquer une Exception avec un message en rouge pour prévenir si les bibliothèques sont vides ou si l'utilisateur a fait aucun choix.

5.3 reprise des tests unitaires de la phase 4 en reprenant

Une partie des tests unitaires de la phase 3, sera effectuée pour contrôler la représentation graphique des tableaux et l'oublie de sélectionner un élément.

Module utilisateur :

Choix 5 : rechercher les livres empruntés = OK

Choix 6 : rechercher les informations d'un livre = OK

Choix 7 : demande le prêt d'un livre = OK

Choix 8 : visualiser les demandes de prêt = OK

Choix 9 : Demander une autorisation accès à une bibliothèque = OK

Choix10 : visualiser les demandes d'autorisation = OK

Module modérateur :

Choix 13 : modifier dans un compte utilisateur = OK

Choix 14 : modifier dans un livre de sa bibliothèque, ainsi que la date de prêt = OK

Choix 15 : visualiser les livres de sa bibliothèque = OK

6 Réunion technique

Préférence pour la mise en place d'une fonction dans le SGBD, afin d'assurer la modification de la date du Choix14. Car impossibilité par le développeur de manipuler les dates sous J2E, donc la décision a été prise d'éviter la difficulté, et d'utiliser le SGBD pour tous les traitements de dates.

Le temps passé à comprendre comment on utilise et incrémente une fonction en langage JPQL, n'est pas passé inaperçu. Mais n'a pas entravé le bon déroulement du projet.



6.1 Reprise des tests unitaires de la phase 4

Choix 16 : Rechercher la liste des livres lus par l'un des ses utilisateurs =OK

Choix 17 : Rechercher les informations d'un membre = OK

Choix 20 : visualiser les demandes d'autorisation à traiter = OK

6.2 Réunion technique

La création des fonctions PL/PgSQL pour les avertissements a été effectuée avec une relative rapidité... Relative car ; le temps perdu à comprendre comment déclencher et incrémenter la première fonction en langage JPQL, à permis une meilleur réactivité pour les fonctions d'avertissement.

7 Tests unitaires de la phase 5

Suite à la réunion qualité de la phase 4(§5.2), une modification majeure a été apportée sur les pages Choix 19 et Choix 21

7.1 Se Connecter avec le compte de Lenaïc pour faire :

- modifier son compte (adresse ou/et téléphone),
- créer un compte utilisateur,
- créer un livre,
- modifier dans un compte utilisateur,
- modifier dans un livre de sa bibliothèque, ainsi que la date de prêt.

7.2 Se connecter avec le nouveau compte créer faire :

- demander une autorisation accès à une bibliothèque (celle d'Yvain),
- demander le prêt d'un livre (auprès de Lenaïc),
- se connecter avec le compte de Yvain pour : répondre à une demande d'autorisation d'accès à la bibliothèque.

7.3 Se connecter avec le compte de Lenaïc pour :

- répondre à un utilisateur sur le prêt d'un livre = OK
- valider le retour d'un livre :
- mise à jour date_retour_prévu = OK
- mise à jour Nombre d'emprunt_ en cours
- mise à jour nombre d'avertissement retard = OK
- mise à jour nombre d'avertissement état = Ok

7.4 Réunion qualité :

- Soit on autorise un modérateur à être lié à lui-même = au risque de le voir se demander un livre.
- Soit on modifie le programme afin de se contraindre à retirer les livres des modérateurs lorsqu'ils se connectent pour rechercher un livre (et lorsqu'ils formulent une demande de prêt.



Décision de la MOA : modification du programme et des requêtes JPQL.

7.5 Réunion technique

7.5.1 1st cas traiter

Un risque majeur dans le programme : un modérateur peut prêter le même livre à 2 personnes différentes en même temps, mise en place d'une fonction de contrôle pour vérifier avant si le livre est en cours de prêt.

La correction suivante a été apportée :

Dans la servlet « ReponseEmprunt », on a rajouté des conditions sur les boutons de type « radio ». Si le livre contient déjà une date de retour prévu, cela veut dire qu'il est en cours de prêt, dans ce cas les 2 boutons « radio » forcent la valeur « activée ». La condition « true » pour les 2 booléens, crée une phrase d'erreur pour prévenir activement l'utilisateur, ainsi qu'une phrase de conseil.

7.5.2 2nd cas traiter

Si il existe plusieurs demandes en cours, on image supérieur à 2 demandes à des modérateurs différents, pour un même utilisateur. Il n'y a pas de contrôle permettant de vérifier, lors que les modérateurs acceptent sa demande, que cet utilisateur a déjà atteint son quota de livres. Car l'incréméntation du nombre de livre se fait seulement lors de la demande de prêt de l'utilisateur.

La correction suivante a été apportée :

Dans la servlet « ReponseEmprunt », on a rajouté une évaluation, par la condition « si le nombre de livre chez un utilisateur est strictement supérieur à 1 » alors création d'un message d'erreur, prévenant le modérateur, ainsi qu'un message de conseil.

7.5.3 3th cas traiter

Suite à la mise en place de mesure corrective sur « l'oublie de sélectionner un élément », de la phase 4(§5.2), une faille de sécurité a été levée. L'utilisateur peut modifier le numéro d'identification présent dans l'en-tête « Information de l'utilisateur ». Ce qui peut produire de graves erreurs dans le SI Publio, surtout si la personne est mal intentionnée.

La correction suivante a été apportée :

Pour palier à ce défaut, il a été décidé de rajouter le terme 'readonly' sur chaque champs texte, du fieldset : « Information de l'utilisateur » de toutes les pages jsp nécessaires.

8 Conclusion

Le programme Publio est validé par une réunion comité qualité, avec une présentation finale au futur utilisateurs, auxquels la société Cyril et Cie leur transmet de chaleureux remerciements pour leurs participations durant les phases de test, et pour leurs remarques.