Qualification :

* + 1. Ticket 2

Problème relevé :

Nous avons découvert un bug de la gestion des prolongations de prêt. En effet, un usager peut prolonger un prêt après la date butoir. Il ne doit pas être possible pour l’usager de prolonger un prêt si la date de fin de prêt est dépassée.

La correction de cette anomalie est urgente car elle nous pose de gros problèmes de gestion. Actuellement beaucoup trop de personnes attendent le mail de rappel pour prolonger leur prêt. Cela entraîne la plainte de certains usagers à qui nous avions communiqué une date de retour et qui se déplacent en bibliothèque pour rien car le prêt a été prolongé a posteriori.

Je vous demande de livrer une nouvelle version de l’application au plus vite afin de corriger ce bug mais sans inclure nos autres demandes en cours. Nous souhaiterions pouvoir mettre en production rapidement sans avoir à faire de recette sur les autres évolutions demandées.

Etat des lieux : En effet, il n’y avait aucune vérification effectuée sur le code existant pour empêcher une telle action. Il faut donc vérifier cette date, puis enfin effectuer une comparaison comme expliqué ci-dessous.

Solution apportée :

Explication générale :

La date limite d’emprunt est comparée à celle où l’utilisateur se connecte à son profil (date du jour). Si cette date est postérieure à la date limite, alors il est impossible de prolonger l’emprunt.

Explication technique :

Un attribut dateToday est ajouté au contrôleur (GestionBookAction). Cet attribut prend la date du jour au format XMLGregorianCalendar, et le compare à l’attribut dateEnd (limite d’emprunt).   
Par un simple jeu de comparaison des données, on affiche ou pas la possibilité de prolonger l’emprunt.

* 1. Ticket 1

Fonctionnalité demandée :

L’usager peut vouloir emprunter un livre dont aucun exemplaire n’est actuellement disponible. Il a la possibilité de poser une option de réservation.

Etat des lieux : Aucune fonctionnalité de ce type aujourd’hui. Seul l’emprunt est possible, ainsi que la prolongation de cet emprunt, mais pas la réservation. Il faudra donc modifier le MPD pour faire une table correspondant à ces réservations.

Solution apportée :

Explication générale :

Ajout de la possibilité de réserver un ouvrage, et donc de s’inscrire dans une liste d’attente.

Explication technique :

Création d’une nouvelle table (reservation), dans laquelle chaque réservation s’inscrit grâce à la méthode doReservation, présente dans le fichier GestionBookAction.

Chaque réservation possède l’id de l’user, l’id du book, la date de réservation ainsi que :

* La colonne « is\_active » qui permet de savoir si la réservation est toujours active
* Email\_send pour savoir si l’utilisateur a reçu un email pour venir récupérer le livre réservé
* Datetime\_email\_send : pour savoir quand a été envoyé l’email
* Is\_ready\_for\_borrow pour savoir si l’user peut venir récupérer le livre

Fonctionnalité demandée :

Le premier à avoir réservé aura la priorité de la réservation dès le retour d’un exemplaire

Solution apportée :

Explication générale :

La date de chaque réservation est enregistrée, et lorsqu’un ouvrage est de nouveau dispo, on sélectionne la réservation la plus ancienne grâce à cette date

Explication technique :

La requête fait un « order by datetime » qui permet de prendre en compte quelle est la réservation la plus ancienne

Fonctionnalité demandée :

Dès qu’un exemplaire est de retour à la bibliothèque, un mail est envoyé au premier usager de la liste…

Solution apportée :

Explication générale :

Un batch tourne toutes les 24h et sélectionne la réservation la plus ancienne, pour envoyer à son user un email l’informant que l’ouvrage est à nouveau dispo

Fonctionnalité demandée :

L’utilisateur dispose de 48h pour venir le chercher.

Solution apportée :

Explication générale :

On compare la date et l’heure d’envoi de mail avec la date et l’heure actuelle, si ça dépasse 48h, la réservation est annulée

Explication technique :

Le batch va constamment vérifier le datetime de la colonne Datetime\_email\_send, et le comparé aux valeurs, ie la date et l’heure du batch. Si ceux-ci sont supérieures à 48h, la réservation est automatiquement annulée, et le boolean « is\_active » passe à false.

Fonctionnalité demandée :

La liste de réservation ne peut comporter qu’un maximum de personnes correspondant à 2x le nombre d’exemplaires de l’ouvrage

Solution apportée :

Explication générale :

On compare seulement le nombre d’ouvrage total et le nombre de réservation. Si inférieur à 2x, alors on ne peut pas réserver

Explication technique :

On fait cette vérification grâce aux méthodes déjà existantes qui permettent à la fois de connaitre le nombre d’exemplaire total, mais aussi le nombre de réservation. S’il y a un facteur 2 entre les deux, une condition en struts empêche la réservation via l’interface utilisateur.

Fonctionnalité demandée :

Il n’est pas possible pour un usager de réserver un ouvrage qu’il a déjà en cours d’emprunt.

Solution apportée :

Explication générale :

On vérifie les emprunts en cours de l’usager, et on les compare avec ceux dispo. Ainsi on bloque en cas de concordance.

Explication technique :

Une méthode permet de récupérer la liste des ouvrages déjà empruntés par l’utilisateur. On a plus qu’à comparer sur le fichier list.jsp cette liste, avec les id des ouvrages afficher. S’il y a concordance, alors on bloque l’affichage de la fonctionnalité de réservation.

Fonctionnalité demandée :

L’usager doit pouvoir annuler une réservation.

Solution apportée :

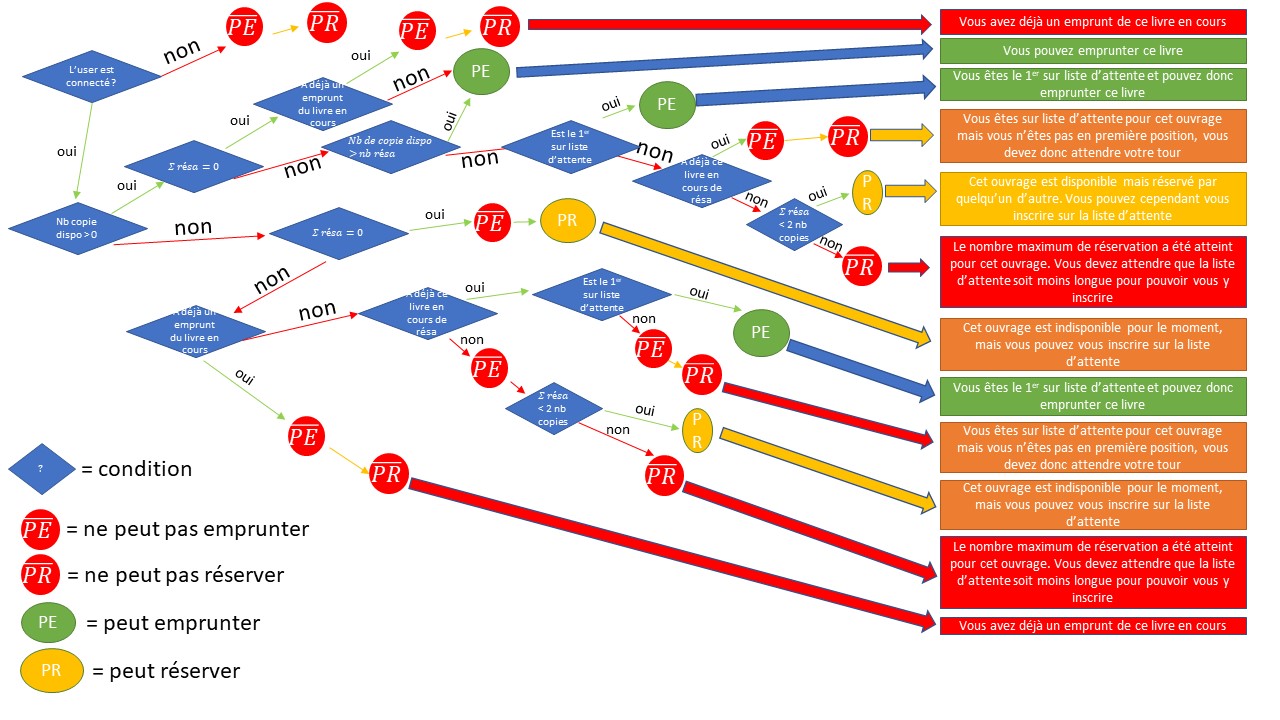
Explication générale :

Création d’une nouvelle fonctionnalité et modification de l’espace personnel pour ajouter cette possibilité.

Explication technique :

Une méthode « cancelReservation » permet de passer la réservation de l’état « is\_active » à false, ce qui permet de ne plus considérer cette réservation dans le reste du code.

Logigramme :



* 1. Ticket 3

Fonctionnalité demandée :

Possibilité d’activer une option (dans leur profil sur l’application web) qui permet le rappel 5 jours avant, ou moins, la fin d’un emprunt par email

Etat des lieux : Actuellement, rien ne permet dans l’espace personnel d’activer ou non cette option.

Solution apportée :

Explication générale :

Ajout d’un attribut dans la table user, qui active ou non cette fonctionnalité

Explication technique :

Simple passage d’un attribut true/false pour modifier le booléen présent en base de données dans la table user colonne « reminder\_active »

Fonctionnalité demandée :

Envoi d’un mail de rappel 5 jours calendaires avant la fin de leur prêt.

Etat des lieux : Il n’y a rien a apporter niveau BDD. Cependant il faut créer une méthode pour récupérer la liste des emprunts arrivant à terme, et un batch automatique.

Solution apportée :

Explication générale :

Une requête SQL permet de sélectionner les emprunts dont la fin se termine sous 5 jours. Ensuite un batch récupère la liste de réservation obtenue par cette requête, et envoi aux utilisateurs concernés un email de rappel.

Explication technique :

Un batch est créé, et sélectionne les emprunts toujours en cours dont la fin se termine dans 5 jours ou moins. Ce batch va ensuite chercher dans la table user si l’option est activée par l’utilisateur, c’est-à-dire si la colonne « reminder\_active » est à true. Dans ce cas, le batch envoi l’email, dans le cas inverse il ne se passe rien.