

BUG_Bloquant_Nr_1 : Suppression d'utilisateur sans vérification des comptes

Explication du bug

Dans la méthode `deleteUser()` du contrôleur `ListeCompteManager.java`, il n'y a pas de vérification préalable des soldes des comptes avant la suppression d'un utilisateur. Cela peut entraîner la perte de données financières si un utilisateur est supprimé alors qu'il a encore de l'argent sur ses comptes.

Modification du code:

```
public String deleteUser() {  
    // BUG BLOQUANT : Si on supprime un client, sans vérifier le solde de ses comptes,  
    // on peut supprimer un client qui a un solde positif.  
    try {  
        //solution : vérifier le solde de ses comptes avant de supprimer le client  
  
        if (client instanceof Client) {  
            Map<String, Compte> comptesNonVides = ((Client)client).getComptesAvecSoldeNonNul();  
            if (!comptesNonVides.isEmpty()) {  
                return "NONEMPTYACCOUNT";  
            }  
        }  
  
        setUserInfo(client.getIdentity());  
        banque.deleteUser(client);  
        return "SUCCESS";  
    } catch (TechnicalException e) {  
        e.printStackTrace();  
        return "ERROR";  
    } catch (IllegalOperationException ioe) {  
        ioe.printStackTrace();  
        return "NONEMPTYACCOUNT";  
    }  
}
```

BUG_Bloquant_Nr_2 : Absence de validation du format du découvert autorisé

Explication du bug

Dans la méthode `changementDecouvert()` du contrôleur `DetailCompteEdit.java`, il n'y a pas de validation que le découvert autorisé est un nombre positif avant de l'appliquer.

Modification du code:

```
public String changementDecouvert() {
    if (!(getCompte() instanceof CompteAvecDecouvert)) {
        return "ERROR";
    }
    try {
        Double decouvert = Double.parseDouble(decouvertAutorise);
        if (decouvert < 0) {
            return "NEGATIVEOVERDRAFT";
        }
        banque.changeDecouvert((CompteAvecDecouvert) getCompte(), decouvert);
        return "SUCCESS";
    } catch (NumberFormatException nfe) {
        nfe.printStackTrace();
        return "ERROR";
    } catch (IllegalFormatException e) {
        return "NEGATIVEOVERDRAFT";
    } catch (IllegalOperationException e) {
        return "INCOMPATIBLEOVERDRAFT";
    }
}
```

BUG_Non_Bloquant_Nr_1: Absence de logo

Explication du bug

Dans la page index.jsp soit la page d'accueil du site, il y figure la mention logo sans l'afficher. Ce problème est dû à un potentiel changement de lieu de stockage. Il suffit donc pour le résoudre de changer le lien. Par exemple ici j'ai mis un autre lien de logo de l'IUT.

Modification du code:

```
<p>  
  ←/— BUG NON-BLOQUANT : Le logo de l'IUT Metz n'est pas affiché —>  
  ←/— Solution : Changer le chemin de l'image —>  
    
</p>
```

BUG_Non_Bloquant_Nr_2 : Absence de logging des opérations critiques

Explication du bug

Dans DetailCompte.java, les opérations de débit et crédit ne sont pas loggées, ce qui rend difficile le suivi des opérations.

Modification du code:

```
6 + import org.slf4j.Logger;
7 + import org.slf4j.LoggerFactory;
8
9 import com.iut.banque.exceptions.IllegalFormatException;
10 import com.iut.banque.exceptions.InsufficientFundsException;
11
12 @@ -12,13 +14,15 @@
13
14 import com.iut.banque.modele.Gestionnaire;
15 import com.opensymphony.xwork2.ActionSupport;
16
17 +
18 public class DetailCompte extends ActionSupport {
19
20     private static final long serialVersionUID = 1L;
21     protected BanqueFacade banque;
22     private String montant;
23     private String error;
24     protected Compte compte;
25 + private static final Logger logger = LoggerFactory.getLogger(DetailCompte.class);
26
27     /**
28      * Constructeur du controleur DetailCompte
29
30 @@ -127,15 +131,21 @@
31
32     public String debit() {
33         Compte compte = getCompte();
34         try {
35             banque.debit(compte, Double.parseDouble(montant.trim()));
36             double montant = Double.parseDouble(this.montant);
37             logger.info("Tentative de débit de {} sur le compte {}", montant, compte.getNumeroCompte());
38             banque.debit(compte, montant);
39             logger.info("Débit réussi");
40             return "SUCCESS";
41         } catch (NumberFormatException e) {
42             e.printStackTrace();
43             logger.error("Erreur de format de montant");
44             return "ERROR";
45         } catch (InsufficientFundsException ife) {
46             logger.error("Fonds insuffisants");
47             ife.printStackTrace();
48             return "NOTENOUGHFUNDS";
49         } catch (IllegalFormatException e) {
50             logger.error("Montant négatif");
51             e.printStackTrace();
52             return "NEGATIVEAMOUNT";
53         }
54     }
55 }
```

Qualité finale Sonar



