Cas YAK A PARTIR Elisee

Dossier A

Question A.1.1

public boolean ancienMdp(String m) {

boolean existe = false;

int i = 0;

while (i < this.lesAnciensMdp.size() && !existe) {

if (this.lesAnciensMdp.get(i).getValMdp().equals(m)) {

existe = true;

} else {

i++;

}

}

return existe;

}

Question A.1.2

/\*\*

\* Vérifie si le mot de passe passé en paramètre est un ancien mot de passe de l'utilisateur.

\*

\* @param m le mot de passe à vérifier

\* @return vrai si le mot de passe est un ancien mot de passe, sinon faux

\*/

public boolean ancienMdp(String m) {

...

}

Question A.2.1

La complexité des mots de passes attendue lors de l’authentification des utilisateurs est un mot de passe de minimum 12 caractères, comportant au moins 1 majuscule, au moins 3 minuscules, au moins 4 chiffres et 1 caractère spécial.

Question A.2.2

public boolean modifierMdp(String valMdp) {

if (verifierMdp(valMdp) && !ancienMdp(valMdp)) {

lesAnciensMdp.add(new MotDePasse(this.motDePasse, LocalDate.now()));

this.motDePasse = valMdp;

return true;

}

return false;

}

Question A.3.1

void verifModifierMdp() {

Utilisateur utilisateur = new Utilisateur(...);

assertTrue(utilisateur.modifierMdp("NouveauMdp456!"));

assertFalse(utilisateur.modifierMdp("AncienMdp123!")); // Le mot de passe est déjà utilisé

assertFalse(utilisateur.modifierMdp("tropcourt1")); // Mot de passe trop court

assertFalse(utilisateur.modifierMdp("MdpSansChiffres !")); // Pas de chiffres dans le mot de passe

assertFalse(utilisateur.modifierMdp("MdpSansMajuscules1")); // Pas de majuscules dans le mot de passe

}

Question A.3.2

Un utilisateur qui n’est pas autorisé peut accéder aux informations sensibles de l’application s’il n’y a pas de restriction d’accès. Cela pourrait entraîner des dommages financiers, l’image de l’entreprise pourrait subir un coup dur, les informations des clients pourraient également se trouver entre les mains de n’importe qui.

Question A.3.3

a.

public int getNiveauHabilitation() {

return sonHabilitation.getNiveau();

}

b.

public AppliHoly(Utilisateur unUtil) throws HeadlessException {

...

for (ElementMenu elementMenu : lesElementsMenu) {

if (elementMenu.getNiveauHabilitation() <= unUtil.getNiveauHabilitation()) {

elementMenu.rendreAccessible();

}

}

}