

POO/COO Rapport Projet PDLA

Fait par Hazem MEJRI C1

I. Introduction:

Le projet en cours vise à créer une plateforme de gestion de missions bénévoles, offrant une solution conviviale pour la collaboration entre divers utilisateurs, tels que les bénévoles, les responsables de missions, et les Valideurs . L'objectif principal consiste à concevoir une application Java dotée d'une interface graphique Swing, accompagnée d'une base de données MySQL pour stocker les informations relatives aux utilisateurs, aux missions, et à leur statut.

Le système prend en charge trois types d'utilisateurs, chacun disposant de fonctionnalités spécifiques. Les utilisateurs généraux, les bénévoles et les validateurs peuvent accéder à des actions spécifiques une fois authentifiés dans l'application. Les fonctionnalités principales englobent l'authentification des utilisateurs, la gestion des missions, l'ajout, la modification, et la suppression de missions, ainsi que le suivi de l'état des missions.

La structure du projet repose sur des classes distinctes, chacune étant responsable d'une partie spécifique de la logique métier. Avec des fonctionnalités telles que l'ajout, la modification et la suppression de missions, le système répond aux exigences de coordination et de suivi des missions bénévoles. L'interface intuitive et l'accès sécurisé garantissent une expérience utilisateur optimale. Les sections suivantes fourniront des détails approfondis sur les fonctionnalités mises en œuvre dans les différentes parties du projet.

II. US1: En tant qu'utilisateur, je veux pouvoir me connecter pour accéder à mon espace et ajouter une mission

Tâches:

1. Interface de Connexion

- Classe `cnx(n)` : Mise en place de l'interface utilisateur pour la connexion. Ici `n=1`
- Utilisation de composants Swing pour la mise en page.
- Ajout de champs pour le mail et le mot de passe.
- Intégration d'un bouton "soumettre" pour valider la connexion.
- Gestion des erreurs d'authentification et affichage de messages appropriés.

2. Validation d'Utilisateur

- Méthode `validerUtilisateur` dans la classe `cnx` : Validation des informations d'identification de l'utilisateur dans la base de données.
- Requête SQL pour vérifier mail et le mot de passe.
- Retour d'un identifiant utilisateur valide ou -1 en cas d'échec.

3. Redirection après Connexion Réussie

- Utilisation de `JOptionPane` pour afficher un message de réussite.
- Création d'instances de pages spécifiques (`PageMission`) en fonction du type d'utilisateur.
- Masquage de la fenêtre de connexion après une connexion réussie.

4. Interface d'Ajout de Mission pour Bénévole

- Classe `PageMission` : Ajout d'un bouton "**Ajouter**" pour permettre au bénévole d'ajouter une nouvelle mission.
- Utilisation d'une boîte de dialogue pour saisir les informations de la mission (description), en mode **WAITING**.

5. Suppression de la Dernière Mission

- Méthode `deletemission` dans la classe `PageMission` : Suppression de la dernière mission ajoutée avec le bouton "**Supprimer**".
- Exécution de la requête SQL DELETE correspondante.
- Mise à jour du modèle de tableau et affichage.

id	nom	prenom	mail	mot_de_passe
1	Mejri	Hazem	mejri.insa	insa2324
2	Sarra	Kouki	kouki.insa@toulouse.fr	sarrakouki
3	Imad	Bk	bk@gmail.com	bkbk
4	ibk	ibk	ibk@gmail.com	ibk
5	test1	test2	test@mail.com	password123

Table 1: Table Client

III. US2: En tant que Valideur, je veux pouvoir me connecter pour accéder à mon espace et valider les missions en attente

Tâches:

1. Interface de Connexion

- Classe `cnx(n)` : Mise en place de l'interface utilisateur pour la connexion. Ici $n=0$
- Utilisation de composants Swing pour la mise en page.
- Ajout de champs pour le mail et le mot de passe.
- Intégration d'un bouton "soumettre" pour valider la connexion.
- Gestion des erreurs d'authentification et affichage de messages appropriés.

2. Validation d'Utilisateur

- Méthode `validerValideur` dans la classe `cnx` : Validation des informations d'identification de l'utilisateur dans la base de données.
- Requête SQL pour vérifier mail et le mot de passe.
- Retour d'un identifiant utilisateur valide ou -1 en cas d'échec.

3. Redirection après Connexion Réussie

- Utilisation de `JOptionPane` pour afficher un message de réussite.
- Création d'instances de pages spécifiques (`PageMissionValid`)
- Masquage de la fenêtre de connexion après une connexion réussie.

4. Interface de Liste des Missions en Attente pour Valideur

- Création de l'interface utilisateur pour afficher les missions en attente de validation.

id	mission_description	date	States	Commentaire
1	Nouvelle Mission	1970-01-01	FINISHED	Pas de commentaire
2	Nouvelle Mission1	1970-01-02	FINISHED	Pas de commentaire
3	Mission Import	1970-01-02	FINISHED	Mission finie
4	jkazje	1970-01-01	FINISHED	Mission finie
5	Finir Pdlia avant dimanche	1970-01-01	FINISHED	Mission finie
6	PDLA	1970-01-01	REFUSED
7	nvl mission	1970-01-01	FINISHED	Pas de Commentaire
8	bkbk	2023-12-09	REFUSED	bkbk impossible
9	je suis la	2023-12-09	REFUSED	Mission Impossible
10	Finir PDLA	2023-12-09	FINISHED	Mission finie
11	je veux ma mission	2023-12-10	ACCEPTED	Mission Acceptée
12	derniere mission	2023-12-10	Waiting	Pas de Commentaire
13	hghg	2023-12-10	IN PROGRESS	Pas de Commentaire

Choix MissionRefuser Mission

Table 2: Interface Mission Valideur

5. Choix et Validation/Refus de Mission pour Valideur

- Ajout des fonctionnalités permettant au valideur de choisir une mission en attente et de l'accepter ou de la refuser. Le choix se fait par le **Numéro (Id)** de la mission dans le tableau.
 - Les missions validées doivent être marquées comme "**IN PROGRESS**" et les missions refusées comme "**REFUSED**" dans la base de données.
6. **Ajout de Commentaire en Cas de Refus pour Valideur**
- Intégration d'une boîte de dialogue pour ajouter un commentaire en cas de refus de mission.
7. **Mise à Jour de la Base de Données après Validation/Refus**
- Implémentation de la logique pour mettre à jour l'état de la mission dans la base de données.

id	nom	prenom	mail	mot_de_passe
1	Hazem	mejri	MejriV	bkbk
2	Imad	bkbk	imad.bk	bkbk
3	ibk2	ibk2	ibk2	ibk2
4	test1V	test2V	valideur@mail.com	password789

Table 3: Table Valideur

IV. US3: En tant que bénévole, je veux pouvoir me connecter pour accéder à mon espace, accepter des missions en attente et les marquer comme terminées après leur réalisation.

1. Interface de Connexion

- Classe `cnx(n)` : Mise en place de l'interface utilisateur pour la connexion. Ici `n=2`
- Utilisation de composants Swing pour la mise en page.
- Ajout de champs pour le mail et le mot de passe.
- Intégration d'un bouton "soumettre" pour valider la connexion.
- Gestion des erreurs d'authentification et affichage de messages appropriés.

2. Validation d'Utilisateur

- Méthode `validerBenevole` dans la classe `cnx` : Validation des informations d'identification de l'utilisateur dans la base de données.
- Requête SQL pour vérifier mail et le mot de passe.
- Retour d'un identifiant utilisateur valide ou -1 en cas d'échec.

3. Redirection après Connexion Réussie

- Utilisation de `JOptionPane` pour afficher un message de réussite.
- Création d'instances de pages spécifiques (`PageMissionBenev`)
- Masquage de la fenêtre de connexion après une connexion réussie.

4. Interface de Liste des Missions en Attente pour Bénévole

- Création de l'interface utilisateur pour afficher les missions en attente d'acceptation.

id	mission_description	date	States	Commentaire
1	Nouvelle Mission	1970-01-01	FINISHED	Pas de commentaire
2	Nouvelle Mission1	1970-01-02	FINISHED	Pas de commentaire
3	Mission import	1970-01-02	FINISHED	Mission finie
4	jkazje	1970-01-01	FINISHED	Mission finie
5	Finir Pdlia avant dimanche	1970-01-01	FINISHED	Mission finie
6	nvl mission	1970-01-01	FINISHED	Pas de Commentaire
7	Finir PDLA	2023-12-09	FINISHED	Mission finie
8	je veux ma mission	2023-12-10	ACCEPTED	Mission Acceptée
9	hghg	2023-12-10	IN PROGRESS	Pas de Commentaire
Choix Mission				

Table 4: Interface Mission Benevole

5. Choix et Acceptation de Mission pour Bénévole

- Ajout des fonctionnalités permettant au bénévole de choisir une mission en attente et de l'accepter. Avec le Bouton **Choix Mission** et Choisir le **Numéro(Id)** de la mission
- Implémentation de la logique pour mettre à jour l'état de la mission dans la base de données après acceptation, l'état de la mission doit changer vers **ACCEPTED**

6. Marquage de Mission comme Terminée par le Bénévole

- Ajout des fonctionnalités permettant au bénévole de marquer une mission comme terminée. Avec le Bouton **Choix Mission** et Choisir le **Numéro(Id)** de la mission
- Implémentation de la logique pour mettre à jour l'état de la mission dans la base de données après achèvement, l'état de la mission doit changer vers **FINISHED** et on doit passer en commentaire le nom prénom du bénévole.

id	nom	prenom	mail	mot_de_passe
1	mejribenev	hazem	lsdkldk.mle	mlml
2	Mejril	Hazem	mejri.insa1	insa2324
3	test1benev	test2benev	benevole@mail.com	password456

Table 5: Table Benevole

id	id_user	mission_description	States	Commentaire	date
1	1	Nouvelle Mission	FINISHED	Pas de commentaire	1970-01-01
2	2	Nouvelle Mission1	FINISHED	Pas de commentaire	1970-01-02
11	2	Mission Import	FINISHED	Mission finie	1970-01-02
12	1	jkazje	FINISHED	Mission finie	1970-01-01
13	3	Finir Pdla avant dimanche	FINISHED	Mission finie	1970-01-01
14	1	PDLA	REFUSED	-----	1970-01-01
15	1	nvl mission	FINISHED	Pas de Commentaire	1970-01-01
17	1	bkbk	REFUSED	bkbk impossible	2023-12-09
18	1	je suis la	REFUSED	Mission Impossible	2023-12-09
19	4	Finir PDLA	FINISHED	Mission finie	2023-12-09
20	1	je veux ma mission	ACCEPTED	Mission Acceptée	2023-12-10
21	1	derniere mission	Waiting	Pas de Commentaire	2023-12-10

Table 6: Table Mission

V. Project Management:

J'ai dû prolonger le premier sprint. Ensuite, j'ai trouvé le bon rythme et le processus est devenu plus fluide lorsque j'ai commencé à travailler en parallèle. Vous pouvez trouver les légendes des différents sprints que j'ai mis en œuvre au cours du projet. Cependant, j'ai rencontré des problèmes avec l'outil Jira en ligne chaque fois que le sprint que je définis est supprimé de manière aléatoire, c'est pourquoi je devais les réécrire constamment. Cela pourrait être le cas lorsque vous visitez mon compte.