

Présentation prestataire

1.1 - Rôle du prestataire dans le projet Orientation+

Le prestataire est l'entité ou l'équipe de développement chargée de la conception, de l'implémentation et de la maintenance technique de la plateforme Orientation+. Il a pour mission de garantir que la solution numérique livrée réponde aux besoins définis dans le cahier des charges et respecte les contraintes techniques et fonctionnelles.

Responsabilités principales :

1. Analyse des besoins clients et rédaction du cahier des charges.
2. Conception technique (choix de l'architecture et des technologies).
3. Développement des fonctionnalités principales et des interfaces utilisateur.
4. Tests de performance, sécurité et compatibilité multiplateforme.
5. Maintenance et mises à jour régulières après la livraison.

1.2 - Équipe prestataire

L'équipe prestataire est composée de plusieurs profils spécialisés pour assurer une exécution optimale du projet :

1. Chef de projet :

- Coordonne l'ensemble des étapes et communique avec le client.
- Valide les livrables à chaque étape (analyse, développement, tests).

2. Développeurs (Front-end et Back-end) :

- **Back-end** : Gestion des bases de données, développement des fonctionnalités Java (JDBC) et sécurisation des données.
- **Front-end** : Développement des interfaces utilisateurs (JavaFX, HTML, CSS, JavaScript si nécessaire).
- Implémentation des frameworks comme Bootstrap pour le design responsive (JavaFX et des styles CSS).

3. Spécialiste UX/UI :

- Conçoit une interface intuitive et ergonomique.
- Assure l'accessibilité et l'adaptabilité sur différents supports (ordinateurs, tablettes, smartphones).

4. Technicien QA (Quality Assurance) :

- Teste les fonctionnalités pour identifier et corriger les bugs avant la mise en production.
- Vérifie la conformité avec les exigences définies dans le cahier des charges.

1.3 - Technologies utilisées par le prestataire

1. Back-end :

- **Java avec JDBC (Java Database Connectivity)** : Permet une gestion efficace de la base de données et une interaction fluide avec celle-ci dans le cadre de l'architecture client lourd.
- Exemple d'implémentation pour la connexion à la base de données :

```
public class UserModel {  
    public User getUserByEmail(String email) throws  
SQLException {  
        String query = "SELECT * FROM users WHERE email = ?";  
        try (PreparedStatement stmt =  
connection.prepareStatement(query)) {  
            stmt.setString(1, email);  
            ResultSet resultSet = stmt.executeQuery();  
            if (resultSet.next()) {  
                return new User(resultSet.getInt("id"),  
resultSet.getString("email"),  
resultSet.getString("password"));  
            }  
        }  
    }  
}
```

```

        return null;
    }
}

```

2. Base de données :

- **MySQL** : Gestion des données utilisateurs, fiches métiers et fichiers uploadés.
- Création de tables pour structurer les informations.
- Exemple de table `users` :

3. Front-end :

- **JavaFX** pour une interface graphique moderne et interactive dans le cadre d'un projet client lourd en Java.
- Utilisation de **CSS** pour personnaliser les éléments d'interface et rendre le design responsive.
- Exemple d'interface utilisateur JavaFX :

```

VBox loginForm = new VBox();
loginForm.getChildren().addAll(
    new Label("Email"),
    new TextField(),
    new Label("Mot de passe"),
    new PasswordField(),
    new Button("Se connecter")
);

```

4. Hébergement (si nécessaire pour la partie back-end) :

- Serveur Apache ou NGINX pour l'hébergement des fichiers et la gestion des requêtes.
- Utilisation de protocoles HTTPS pour sécuriser les échanges de données (dans le cas où des interactions avec une base de données en ligne ou un serveur web sont nécessaires).

5. Sécurité :

- **Hashage des mots de passe avec bcrypt** pour garantir la sécurité des informations sensibles.
- Contrôle d'accès basé sur les rôles (élève ou professeur).
- Validation des fichiers uploadés pour prévenir les vulnérabilités, en particulier lors des échanges de documents ou autres données sensibles.

1.4 - Étapes réalisées par le prestataire

Le prestataire suit une méthodologie rigoureuse pour garantir la qualité du projet, en respectant les étapes suivantes :

1. **Rédaction du cahier des charges** : Définition des besoins du client et des fonctionnalités à implémenter.
2. **Conception et développement** : Choix des technologies et mise en place de l'architecture du projet en Java et MySQL, avec la structuration des classes et des modèles MVC.
3. **Tests et validation** : Réalisation des tests de sécurité, performance et compatibilité pour garantir le bon fonctionnement du logiciel.