**Spécifications fonctionnelles :**

**Description :**

Ce cas d'utilisation permet à l'utilisateur de consulter le détail d'une fiche client.

**Acteurs :**

Membre de l'équipe commerciale

**Pré conditions :**

Le client doit exister dans la base de données.

**Résultat :**

La fiche client détaillée est affichée.

**Scénario nominal :**

1. A partir de la liste de clients, le commercial sélectionne un client et choisit l'action "afficher fiche client".

2. La fiche client est affichée avec les informations qui ont rentrées lors de sa création et/ou de sa modification.

**Description :**

Cette page permet aux membres de l’équipe commerciale de consulter l’ensemble des données décrivant un client.

|  |  |  |  |
| --- | --- | --- | --- |
| Désignation | Nature | Type de Champ | Règles de gestion |
| Nom du client | Lecture |  |  |
| Prénom du client | Lecture |  |  |
| Nom de l’entreprise | Lecture |  |  |
| Adresse e-mail | Lecture |  |  |
| Téléphone fixe | Lecture |  |  |
| Téléphone mobile | Lecture |  |  |
| Etat du client | Lecture | Lead/client |  |
| Garanties | Lecture | Oui/non |  |
| Commentaires | Lecture |  |  |

**Actions possibles :**

|  |  |  |
| --- | --- | --- |
| Désignation | Action | Règles de gestion |
| Consulter les devis | Ouvre la liste des devis liée au client |  |
| Consulter les commandes | Ouvre la liste des commandes liée au client |  |
| Gérer fiche client | Permet de modifier/créer/supprimer fiche client |  |

**Exigences non fonctionnelles**

La Performance pour garantir un accès rapide à l’application aux utilisateurs ;

- La Fiabilité pour assurer un fonctionnement maîtrisé sans plantage du CRM ;

- La Maintenabilité, l’application sera reprise par la DSI de POEC une fois finalisée, le code source devra être accompagné d’une documentation adaptée et d’une bonne couverture de code par des tests unitaires ;

- La Scalabilité est le dernier enjeu de taille qui doit permettre de soutenir le développement commercial de l’entreprise.

L’application devra être conteneurisée afin de facilité la scalabilité de l’application. La solution Docker est préconisée pour mettre en place cette conteneurisation.

`

Une usine logicielle (CI/CD) doit être mise en place pour automatiser un certain nombre de tâches ainsi que le déploiement.

La qualité de code doit être garantie par une couverture du code source d’à minima 50%.

Le développement du CRM s’appuiera sur des technologies récentes et standards : Java, Spring, HTML, CSS. La base de données relationnelle doit être gérée par PostgreSQL.

L’application devra être développée en Java avec l’usage de Spring, JUnit devra être utilisé pour décrire les tests unitaires.

La base de données sera basée sur PostgreSQL.

Les technologies pour la partie front est laissée à la discrétion de l’équipe de développement. Un point d’attention concerne cependant l’utilisabilité, la simplicité et l’esthétique de l’application.

L’équipe de développement devra utiliser GIT et les bonnes pratiques de gestion des branches.

L’équipe de développement devra intégrer Docker à sa solution. La granularité des images Docker est laissée à la discrétion de l’équipe de développement afin de garantir le meilleur rapport possible.

Une usine logicielle doit être mise en place afin d’automatiser un certain nombre de tâches récurrentes :

- Build de l’application

- Passage des tests unitaires

- Mesure de la qualité du code

- Packaging