

# **Application DeliveryFit Proposition commerciale**

## 1. Contexte

*Décrire ici le contexte global et le contexte lié au besoin métier.*

### **Contexte global**

Alisa’s Closet est un détaillant de vêtements français depuis plus de 10 ans, disposant d'une application en ligne permettant aux clients de commander et se faire livrer. Le système actuel centralise les expéditions depuis un seul centre de livraison, ce qui entraîne des délais et des coûts supplémentaires. L’entreprise souhaite optimiser ce processus en intégrant l’application DeliveryFit, capable de déterminer le point d’expédition optimal en fonction de la proximité et de la disponibilité des articles.

### **Fonctionnalités métier**

DeliveryFit interviendra lors de la validation du panier sur le site e-commerce existant. Elle récupérera l'adresse de livraison, la liste des articles achetés, et la liste des magasins de l’enseigne, pour déterminer le lieu d’expédition optimal (centre de livraison ou magasin) et renvoyer cette information au client avant paiement.

## 2. Réalisation du projet

### **2.1 Définition des tâches techniques**

* **Mise à jour de la base de données** : Mettre à jour la BDD pour permettre le stockage des magasins de l'enseigne, ajouter une table des magasins (id, adresse, nom du gérant, prénom du gérant, sexe du gérant, email du gérant, téléphone du gérant) et une table de jointure entre magasin et produits (id, id produit, id magasin, quantité disponible)
* **Créer l'application Frontend via Angular :** Créer la page de validation du panier avec les informations d'expédition, créer des validateurs de champs dans le but d'éviter l'envoi d'un formulaire incomplet ou dont l'adresse n'existe pas, créer un service permettant le transfert des informations de livraison au backend, créer des modèles de données pour faciliter la structure des DTOs, créer une méthode dans le service pour transmettre les informations de livraison (articles de la commande, contact, adresse de livraison) et création d'une page d'attente de la récupération d'un point d'expédition
* **Réaliser les tests unitaires de la partie frontend avec Jest :** Mise en place d’une batterie de tests en lien avec les formulaires (vérifications des envois de requêtes, des erreurs potentielles, présence de messages personnalisés et autre validation des champs)
* **Intercepter l'URL du site de vente en ligne et rediriger vers DeliveryFit :** Mettreen place un mécanisme permettant la redirection depuis le bouton « Commander »
* **Créer l'application Backend via Spring Boot** : Créer un repository permettant la récupération des établissements et de leurs stocks, créer un service permettant le traitement des établissement et de leurs stocks, créer une méthode permettant de récupérer en BDD les établissements disponibles pour une livraison dans un rayon pré-défini, comparer dans une méthode du service les distances entre les magasins et / ou le centre de livraison et le domicile, créer une méthode permettant le calcul du temps de trajet depuis chaque endroit vers le domicile de la personne et l'obtention de la route la plus rapide, créer un DTO d'un lieu d'expédition, d'une adresse de livraison, de la date potentielle d'expédition et estimation du temps de trajet / date d'arrivée, utiliser le DTO pour envoyer les informations au contrôleur, créer un contrôleur dans l'API pour, à partir d'un DTO d'informations de panier et d'une adresse client, retourner un DTO contenant la réponse appropriée
* **Faire des tests unitaires pour la partie backend** : Réaliser les tests des méthodes du service de traitement des établissements et des stocks, des retours et des potentielles levées d'exceptions, réaliser les tests du contrôleur dans le but de vérifier les codes de retour HTTP et les messages potentiels (code de statut en lien avec des problèmes d’authentification par exemple)
* **Finaliser la partie frontend suite au retour de l’API** : Créer la page de confirmation de la livraison possible (bouton confirmer la livraison / procéder au paiement), créer une page d'erreur informant qu'il n'est pas possible d'effectuer la livraison pour l'adresse demandée (message d'erreur libre), implémenter des tests de la page d'attente / confirmation de la commande potentielle, avec vérification de messages d'erreur en cas de soucis et créer la redirection de DeliveryFit vers le site de vente en ligne une fois la livraison confirmée par l’utilisateur
* **Réaliser des tests de type E2E pour valider le fonctionnement de l’application** : Vérifier que l’application DeliveryFit se lance bien au bon moment de la réalisation d’une commande par l’utilisation, que les requêtes entre l’API et l’application frontend ont bien lieu et retournent les bonnes valeurs. Vérifier également que les messages d’erreurs soient affichés à l’utilisateur en cas de soucis éventuel et enfin que la redirection en cas d'appui sur le bouton de validation de la livraison retourne vers l’application de base et au service de paiement
* **Créer un mécanisme de notification au magasin expéditeur de l'arrivée d'une nouvelle commande à préparer**: Mettre en place un potentiel micro-service dont le déclenchement serait basé sur la réception d’un message provenant d’une queue RabbitMQ ou d’un flux avec Kafka et qui aurait pour but l’envoi d’un email au gérant d’un magasin pour l’informer d’une nouvelle commande

### **2.2 Gestion des points de complexité**

Les points de complexité ont été évalués en fonction de l’expérience personnelle de l’équipe, en partant du principe que les tâches sont réalisées par une personne n’ayant pas de soucis de compréhension théorique des attentes et des objectifs de la tâche. La marge laissée est à la fois dans le sens où la tâche pourrait prendre plus de temps que prévu mais également être réalisée plus rapidement.

La complexité se base sur une échelle allant de 1 à 5, par ordre croissant de difficulté. Les tâches en lien avec la mise à jour de quelques lignes de code dans l’applicatif ou de création de tables supplémentaires en base de données sont jugées les plus aisées. A contrario, la réalisation de la logique métier et d’une interface étant à la fois conforme aux standards de sécurité, d’esthétique et de respect des principes SOLID est évaluée plus complexe. Le summum de complexité étant laissé à la réalisation de l’algorithme permettant la sélection automatique d’un lieu de livraison pour le client, celui-ci étant à la fois un algorithme de type « shortest path » et vérification des stocks des lieux obtenus avant confirmation finale d’un lieu de livraison adapté.

|  |  |  |  |
| --- | --- | --- | --- |
| Tâche technique | Complexité | Estimation temps (en heures) | Marge (en heures) |
| Ajouter une table des magasins en base de données | 1 | 1 | 0,5 |
| Ajouter une table de jointure pour suivre les stocks des magasins | 2 | 1 | 0,5 |
| Réaliser la redirection vers l’application DeliveryFit | 1 | 1 | 0,25 |
| Réaliser la redirection vers l’application de base une fois confirmation de la commande | 1 | 1 | 0,25 |
| Créer une page de formulaire pour l’adresse de livraison | 3 | 5 | 1 |
| Réaliser la vérification des champs du formulaire avant son envoi potentiel | 3 | 2 | 0,5 |
| Réaliser la logique métier permettant l’envoi du formulaire à l’API backend | 4 | 3 | 1 |
| Créer une page d’attente de traitement, de confirmation de livraison possible ou d’erreur potentielle côté serveur | 3 | 5 | 1 |
| Créer le micro-service de type API permettant le traitement des requêtes frontend et leur réponse | 3 | 5 | 1 |
| Réaliser l’algorithme de sélection du magasin ou du centre de livraison le plus proche en fonction des stocks disponibles | 5 | 4 | 2 |
| Réaliser la batterie de tests pour l’application frontend | 3 | 5 | 2 |
| Réaliser la batterie de tests pour l’API | 3 | 5 | 2 |
| Réaliser des tests E2E pour l’ensemble de l’applicatif | 3 | 7 | 4 |
| Mettre en place un environnement de déploiement via Docker pour conteneuriser les applicatifs | 3 | 4 | 2 |

### **2.3 Coûts**

*Servez-vous de la description de l’équipe dans le cahier des charges pour déterminer la charge de travail à prévoir pour chaque tâche, afin de pouvoir en déduire les coûts. Détaillez vos calculs dans cet espace, puis complétez le tableau avec vos conclusions.*

*NB : Le format du tableau est un exemple. Vous pouvez le modifier en fonction de vos trouvailles et/ou de l’organisation de vos idées.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tâche | Temps | Membre de l’équipe | TJM | Total |
| Ajouter une table des magasins en base de données | 1 | *Architecte* |  |  |
| Ajouter une table de jointure pour suivre les stocks des magasins | 1 | *Architecte* |  |  |
| Réaliser la redirection vers l’application DeliveryFit | 1 | *Dev Frontend* |  |  |
| Réaliser la redirection vers l’application de base une fois confirmation de la commande | 1 | *Dev Frontend* |  |  |
| Créer une page de formulaire pour l’adresse de livraison | 5 | *Dev Frontend* |  |  |
| Réaliser la vérification des champs du formulaire avant son envoi potentiel | 2 | *Dev Frontend* |  |  |
| Réaliser la logique métier permettant l’envoi du formulaire à l’API backend | 3 | *Dev Frontend* |  |  |
| Créer une page d’attente de traitement, de confirmation de livraison possible ou d’erreur potentielle côté serveur | 5 | *Dev Frontend* |  |  |
| Créer le micro-service de type API permettant le traitement des requêtes frontend et leur réponse | 5 | *Dev Backend* |  |  |
| Réaliser l’algorithme de sélection du magasin ou du centre de livraison le plus proche en fonction des stocks disponibles | 4 | *Dev Backend* |  |  |
| Réaliser la batterie de tests pour l’application frontend | 5 | *Dev Frontend* |  |  |
| Réaliser la batterie de tests pour l’API | 5 | *Dev Backend* |  |  |
| Réaliser des tests E2E pour l’ensemble de l’applicatif | 7 | *Dev Frontend* |  |  |
| Mettre en place un environnement de déploiement via Docker pour conteneuriser les applicatifs | 4 | *Architecte* |  |  |

### 

### **2.4 Risques identifiés**

*Ajouter ici une matrice des risques.*

### **2.5 Définition des objectifs de performance**

*Ajouter ici des objectifs de performance pour montrer la faisabilité quantifiable de la solution. Il est également possible d’ajouter cette section dans la troisième partie.*

## 3. Synthèse

*Rédiger ici un résumé du document avec la proposition chiffrée, détaillée et comparée avec le budget initial. Dans l’idéal, proposer une deuxième proposition dépassant le budget, avec des améliorations en plus.*