



Application DeliveryFit

Proposition commerciale

1. Contexte

1.1 Contexte global

Le client, Alisa's Closet est un détaillant de vêtements français depuis plus de 10 ans. Son activité repose sur la vente de vêtements grâce à une application en ligne, les acheteurs peuvent se faire livrer leurs articles achetés.

1.2 Fonctionnalités métier

Actuellement, tous les produits sont expédiés depuis un seul centre de livraison, ce qui entraîne plusieurs incohérences dans le système de livraison des vêtements. L'objectif de cette mission est d'optimiser la gestion des expéditions achetées par les clients en ligne en développant une nouvelle application.

2. Réalisation du projet

2.1 Définition des tâches techniques

Liste des tâches techniques

- Créer l'application Back End en utilisant Spring Boot
- Créer l'application Front End en utilisant le framework Angular
- Intercepter l'URL du site de vente en ligne et rediriger vers l'URL de DeliveryFit

- Transmettre les informations de livraison (articles de la commande, contact, adresse de livraison)
- Comparer les distances entre les magasins et/ou le centre de livraison et le domicile
- Trouver le point de livraison le plus proche (centre de livraison ou Magasin)
- Créer la page de livraison de validation du panier avec les informations d'expédition
- Créer la redirection de DeliveryFit vers le site de vente en ligne
- Envoyer une notification au magasin expéditeur

2.2 Gestion des points de complexité

Pour les tâches techniques dont la complexité est facile, une demi-journée sera envisagée avec une marge de quart de journée. Même chose pour les tâches techniques dont la complexité est moyenne. Pour les tâches techniques difficiles, on va privilégier une journée entière pour sa réalisation, une marge de sécurité d'une demi-journée sera envisagée contrairement aux tâches faciles et moyennes.

Tâche technique	Complexité (facile, moyenne, difficile)	Estimation temps	Marge
Créer l'application Back End en utilisant Spring Boot	facile	demi-journée	quart de journée
Créer l'application Front End en utilisant le framework Angular	facile	demi-journée	quart de journée
Intercepter l'URL du site de vente en ligne et rediriger vers l'URL de DeliveryFit	moyenne	demi-journée	quart de journée
Transmettre les informations de livraison (articles de la commande, contact, adresse de livraison)	moyenne	demi-journée	quart de journée
Comparer les distances entre les magasins et/ou le centre de livraison et le domicile	difficile	journée	demi-journée

Trouver le point de livraison le plus proche (centre de livraison ou Magasin)	difficile	journée	demi-journée
Créer la page de livraison de validation du panier avec les informations d'expédition	moyenne	demi-journée	quart de journée
Créer la redirection de DeliveryFit vers le site de vente en ligne	moyenne	demi-journée	quart de journée
Envoyer une notification au magasin expéditeur	facile	demi-journée	quart de journée

2.3 Coûts

Budget max : 10 000€

Dév front : 700€ TJM, 350€ demi-journée, 175€ quart de journée

Dév back : 700€ TJM, 350€ demi-journée, 175€ quart de journée

Architecte : 800€ TJM, 400€ demi-journée, 200€ quart de journée

Tâche	Temps	Membre de l'équipe	TJM + marge de sécurité	Total
Mise en place base de données	Demi-journée	Architecte	400,00 € + 200,00 €	600,00 €
Créer l'application Back end avec Spring	Demi-journée	Développeur back	350,00 € + 175,00 €	525,00 €
Créer l'application Front end avec Angular	Demi-journée	Développeur front	350,00 € + 175,00 €	525,00 €
Récupérer	Demi-journée	Développeur	350,00 € +	525,00 €

<i>informations lors de la validation du panier</i>		<i>back</i>	<i>175,00 €</i>	
<i>Calculer si magasin + proche du domicile que centre de livraison et envoyer résultat</i>	<i>Journée</i>	<i>Développeur back</i>	<i>700,00 € + 350,00 €</i>	<i>1 050,00 €</i>
<i>Afficher confirmation du panier avec informations et rediriger utilisateur vers l'application</i>	<i>Demi-journée</i>	<i>Développeur front</i>	<i>350,00 € + 175,00 €</i>	<i>525,00 €</i>
<i>Envoyer notification au magasin expéditeur</i>	<i>Demi-journée</i>	<i>Développeur back</i>	<i>350,00 € + 175,00 €</i>	<i>525,00 €</i>
<i>Déployer l'application</i>	<i>Demi-journée</i>	<i>Architecte</i>	<i>400,00 € + 200,00 €</i>	<i>600,00 €</i>

Budget consacré au développement des fonctionnalités : 3250€ + 1625€ de marge de sécurité, total 4875€

Reste 5125€

Corrections à prévoir : 1375€

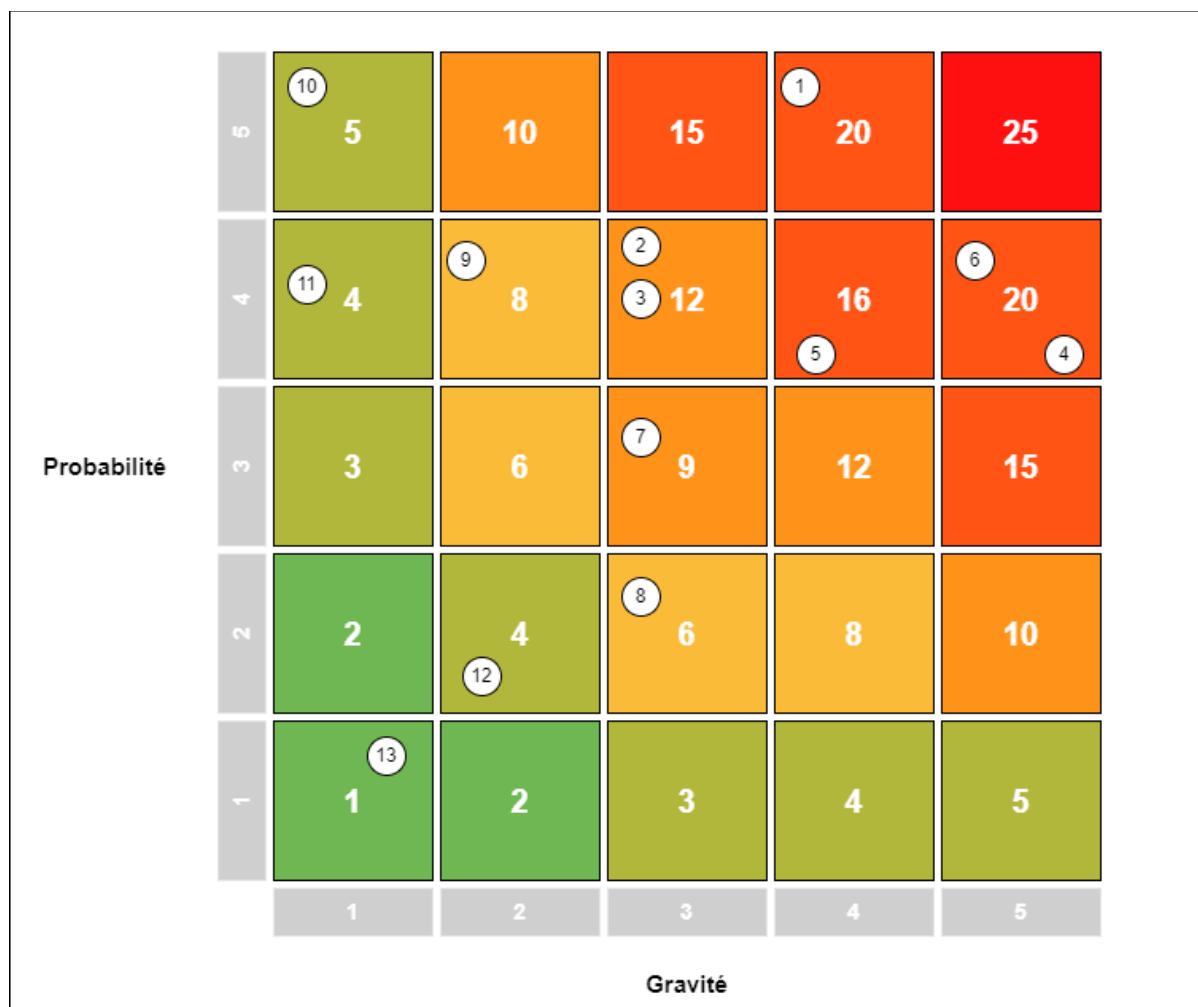
Estimation/découpage des tâches: 1375€

Gestion de projet: 1375€

Reste 1000€

Usage de l'API Distance Matrix de Google pour calculer les temps de trajets en fonction des adresses entrées: <https://developers.google.com/maps/documentation/distance-matrix?hl=fr> , selon la documentation tarifaire, pour 1000€, si on considère que 1€ = 1\$, nous pouvons faire appel à 225000 requêtes pour le service Basic, voir tarifs.

2.4 Risques identifiés



Risques identifiés :

1. Réalisation d'un produit incohérent avec les attentes et les besoins
2. Dépassement de budget
3. Impossibilité de faire évoluer le budget en cours de projet
4. Réalisation d'un produit non fonctionnel
5. Incompréhensions sur des spécifications techniques
6. Anomalies techniques dans les transferts d'informations
7. Dépassement des délais de livraison
8. Allongement des délais de livraison
9. Climat conflictuel dans l'équipe projet
10. Ralentissement dans l'avancement du projet
11. Corrections de bugs
12. Indisponibilité du personnel
13. Panne technique

Solutions potentielles

Risques	Solutions
1. Réalisation d'un produit incohérent avec les attentes et les besoins	Développer et tester chaque fonctionnalité à intervalle régulier afin de répondre aux besoins
2. Dépassement de budget	Imposer une limite inférieure au budget

	maximum. Exemple : budget de 10000€, imposer une limite de 8000€ d'utilisation pour laisser 2000€ de marge. Utiliser les 2000€ restants pour éventuellement poursuivre le développement si aucun problème n'est rencontré.
3.Impossibilité de faire évoluer le budget en cours de projet	Même solution que le risque 2, imposer une limite de budget.
4.Réalisation d'un produit non fonctionnel	Même solution que le risque 1, développer et tester très régulièrement les fonctionnalités.
5.Incompréhensions sur des spécifications techniques	Prévoir avant le développement des réunions entre membres de l'équipe pour s'assurer de la bonne compréhension et du déroulement du développement.
6.Anomalies techniques dans les transferts d'informations	Vérifier l'état du serveur, puis des fonctionnalités implémentées.
7.Dépassement des délais de livraison	Imposer un « sous-délai » antérieur au délai initialement prévu pour tout développer. Peaufiner, vérifier, tester ensuite jusqu'aux délais de livraison.
8.Allongement des délais de livraison	Même solution que le risque 7
9.Climat conflictuel dans l'équipe projet	Être clair sur la stratégie prévue et renforcer l'esprit d'équipe.
10.Ralentissement dans l'avancement du projet	Même solution que le risque 7.
11.Corrections de bugs	Tester régulièrement les fonctionnalités et avoir un rapport de couverture proche de 100%
12.Indisponibilité du personnel	S'arranger en amont avec les autres membres restants sur les tâches sur lesquelles ils pourraient être en mesure de les remplacer.
13.Panne technique	Avoir au moins deux serveurs à des endroits différents. Mettre en place un plan de reprise d'activité et le suivre en cas de panne d'un des deux serveurs.

2.5 Définition des objectifs de performance

Réalisation du projet faisable :

Environnement technique moderne (Angular, Spring)

Découpage des tâches techniques réparties entre membres de l'équipe

Durée de développement rapide

Utilisation d'une API externe de Google (API Distance Matrix) efficace

Objectifs de performance :

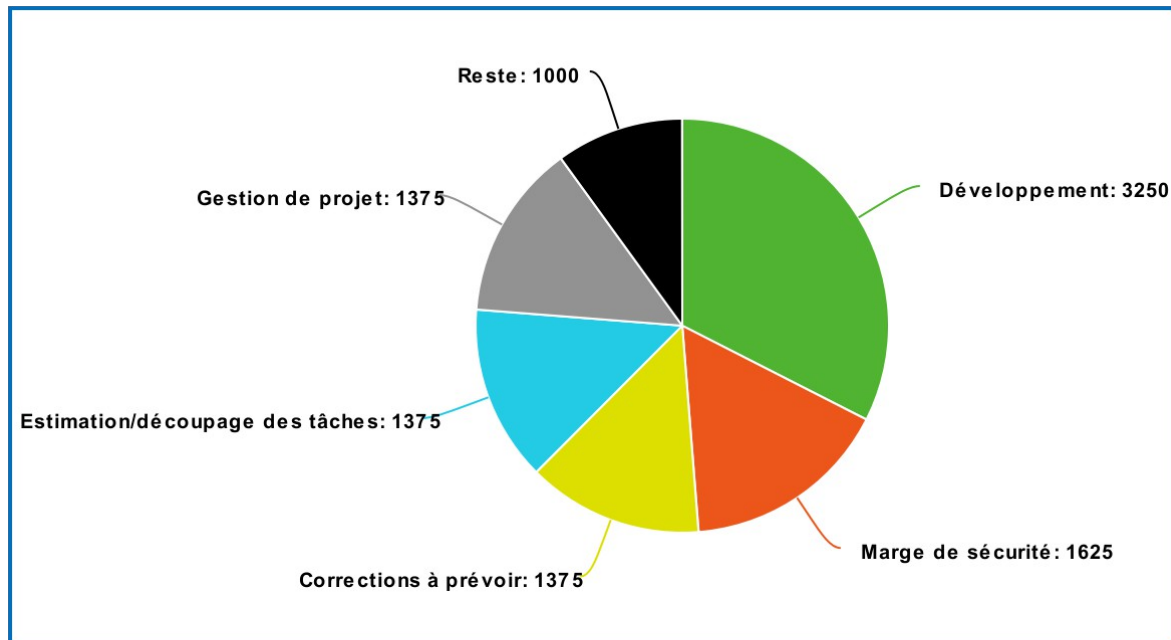
Les outils Angular et Spring simplifient et accélèrent grandement le processus de développement d'applications web quelque soient leurs complexités. Angular repose sur le principe de SPA (single page application), tandis que Spring fournit les outils nécessaires pour créer des systèmes back-end hautement sécurisés. Ces outils sont connus pour leurs performances, vitesse d'exécution et de réponses, offrant ainsi une expérience utilisateur plus que satisfaisante. Ce modèle pourra s'adapter aux demandes de livraisons de plus en plus fortes.

Performance projet	Performance logiciel
<ul style="list-style-type: none">• Respect des délais.• Découpage des tâches entre membres de l'équipe.• Gain d'expérience des membres de l'équipe.	<ul style="list-style-type: none">• Développement rapide, simple, efficace.• Développement avec Angular : principe de SPA (single page application).• Développement avec Spring : création de systèmes back-end hautement sécurisés.• Expérience utilisateur satisfaisante• Temps de réponse du serveur : moins de 100ms, entre 100 et 500ms correct.

3. Synthèse

Budget imposé : 10000€

Proposition retenue : Sachant que le budget imposé est de 10000€, nous allons consacrer environ la moitié pour développer les fonctionnalités (3250€) avec une marge de sécurité (1625€). Nous allons partager ensuite le budget de manière équitable pour la gestion de projet, l'estimation/découpage des tâches et corrections à prévoir (1375€ chacun). Il reste ensuite 1000€ que l'on pourra consacrer au service de Google pour les calculs de distances d'adresses.



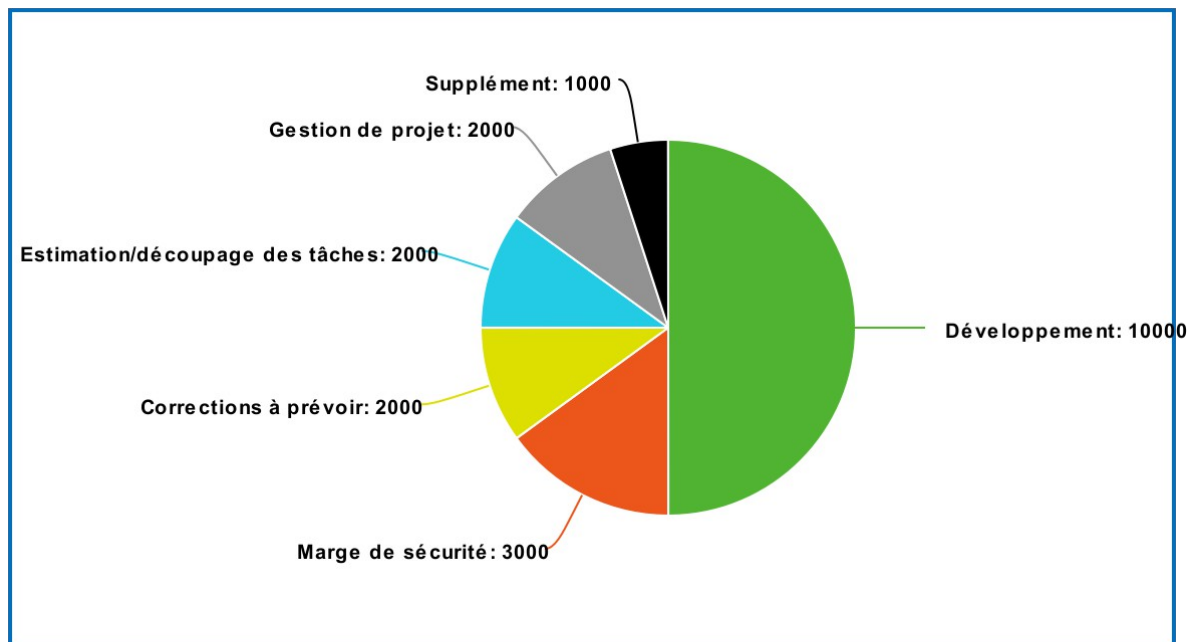
■ Développement ■ Marge de sécurité ■ Corrections à prévoir
■ Estimation/découpage des tâches ■ Gestion de projet ■ Reste

meta-chart.com

Deuxième proposition :

- Développement : 10000€
- Marge de sécurité : 3000€
- Corrections à prévoir : 2000€
- Estimation/découpage des tâches:2000€
- Gestion de projet:2000€
- Supplément (API Distance Matrix Google) : 1000€

Budget total : 20000€



■ Développement ■ Marge de sécurité ■ Corrections à prévoir
■ Estimation/découpage des tâches ■ Gestion de projet ■ Supplément