

Volbi-Tech



Alisa's Closet

Application DeliveryFit

Proposition commerciale

1. Contexte

- a. Contexte global
- b. Fonctionnalités métier

2. Réalisation du projet

- a. Définition des tâches techniques
- b. Gestion des points de complexité
- c. Coûts
- d. Risques identifiés
- e. Définition des objectifs de performance

3. Synthèse

1. Contexte

1.1 Contexte global :

Alisa's Closet est un détaillant de vêtements en ligne cherchant à optimiser ses délais de livraison en développant **DeliveryFit**. Cette application web a pour objectif de réduire le temps de livraison en expédiant les produits depuis le point de vente le plus proche du client, plutôt que depuis un centre de livraison centralisé. Cela vise à améliorer l'expérience client, réduire l'empreinte carbone, et optimiser les coûts logistiques.

Contraintes :

- **Budget** : 10.000 €
 - **Deadline** : 5 mois
 - **Méthodologie** : Agile (Scrum)
-

1.2 Fonctionnalités métier :

- **Algorithme de Comparaison des Distances** : Trouver le point de vente le plus proche.
 - **Interface Utilisateur** : Validation du panier et saisie des informations d'expédition.
 - **Notifications Automatiques** : Informer automatiquement le magasin expéditeur une fois la livraison confirmée.
-

1. Réalisation du projet

2.1 Définition des tâches techniques :

Relations avec les bases de données :

1. Créer la base de données relationnelle
2. Configurer des entités et les relations avec Spring Data JPA
3. Intégrer et connecter la base de données MySQL dans Spring Boot

Backend :

1. Créer l'application Backend en utilisant Spring Boot :

- a. Initialisation du projet Spring Boot avec Spring Initializr
- b. Configuration du projet avec Maven et ajout des dépendances
- c. Mise en place de la structure du projet Spring Boot
- d. Configuration des fichiers de propriétés de l'application

2. Comparer les distances entre les magasins et le domicile :

- a. Création de l'API REST et des endpoints pour récupérer et transmettre les données
- b. Développer et implémenter un algorithme de comparaison des distances pour l'optimisation des livraisons
- c. Création des services, de leur implémentation et des repositories pour la gestion des livraisons

3. Trouver le point de livraison le plus proche :

- a. Récupérer le meilleur résultat de l'algorithme de comparaison des distances et le retourner

4. Envoyer une notification au magasin expéditeur :

- a. Implémenter une solution pour notifier automatiquement le magasin une fois qu'une livraison est confirmée

Frontend :

1. Créer l'application Frontend en utilisant Angular :

- a. Initialisation du projet Angular
- b. Installation et configuration des dépendances (avec RxJS, Angular Material)
- c. Configuration des environnements de build et de déploiement
- d. Création de la structure des composants (modèles, services, routes)
- e. Configuration des tests unitaires et de la couverture de code (à l'aide d'outils tel que Jest ou Cypress)

2. Transmettre les informations de livraison :

- a. Création des modèles de livraison pour la gestion des données
- b. Création des services pour transmettre les données au backend

3. Intercepter l'URL du site :

- a. Implémentation d'un intercepteur pour capturer et rediriger les URLs.

4. Créer la redirection vers le site de vente :

- a. Rediriger l'utilisateur vers le site de vente en utilisant la logique implémentée dans l'intercepteur.

5. Création des formulaires pour la validation du panier et les informations d'expédition :

- a. Développer les formulaires d'expédition avec validation des données

Authentification :

- 1. Mise en place de l'authentification côté backend via un système de token JWT et OAuth2**
- 2. Mise en place de l'authentification côté frontend avec JWT et des guards Angular**

Gestion de projet : L'architecte logiciel va jouer le rôle de SCRUM master pour la gestion du projet.

- 1. Supervision des tâches et de l'équipe**
 - 2. Démonstration au client**
-

2.2 Gestion des points de complexité :

Les tâches sont estimées selon la [Suite de Fibonacci](#) et l'[équation PERT](#).

Estimer la complexité des tâches avec la Suite de Fibonacci :

- **1 :** Pour les tâches mineures ou corrections de bugs rapides.
- **2 :** Pour les tâches de petite envergure ou fonctionnalités basiques.
- **3 :** Pour les tâches avec un peu plus de logique ou d'intégration.
- **5 :** Pour les fonctionnalités plus larges nécessitant une réflexion sur la conception.
- **8 :** Pour les fonctionnalités majeures, des intégrations multiples, ou des tâches nécessitant des analyses supplémentaires.

Calculer le temps pour chaque tâche avec l'équation PERT :

Complexité de la tâche	Temps optimiste (en jour)	Temps plus probable (en jour)	Temps pessimiste (en jour)	Équation PERT	Résultat (en jour)	Arrondir le résultat (en jour)
1	0,25	0,50	1	$(0,25 + (4 * 0,5) + 1) / 6$	0,54	0,5
2	0,5	1,00	2	$(0,5 + (4 * 1) + 2) / 6$	1,08	1
3	1	1,50	3	$(1 + (4 * 1,5) + 3) / 6$	1,67	1,75
5	1,5	2,50	5	$(1,5 + (4 * 2,5) + 5) / 6$	2,75	2,75
8	2,5	4	8	$(2,5 + (4 * 4) + 8) / 6$	4,42	4,5

Calculer la marge de sécurité :

Une marge de sécurité est ajoutée à chaque tâche pour gérer les imprévus et s'assurer de respecter les délais.

Complexité de la tâche	Durée estimée	Marge de sécurité	Durée finale (arrondi)
1	0,5	10 % (0,05 jour)	0 jour
2	1	20 % (0,20 jour)	0,25 jour
3	1,75	25 % (0,44 jours)	0,5 jour
5	2,75	30 % (0,83 jour)	0,75 jour
8	4,5	30 % (1,35 jour)	1,5 jour

Estimation du temps pour chaque tâche :

Tâche technique	Complexité	Estimation temps (en jour)	Marge (en jour)
Sans marge			
Créer l'application Backend en utilisant Spring Boot	1	0,5	0

Créer l'application Frontend en utilisant Angular	1	0,5	0
Créer la redirection de DeliveryFit vers le site de vente en ligne	1	0,5	0
Marge de 20 %			
Intercepter l'URL du site de vente en ligne et rediriger vers l'URL de DeliveryFit	2	1	0,25
Transmettre les informations de livraison	2	1	0,25
Trouver le point de vente le plus proche	2	1	0,25
IHM - Créer la page de validation du panier avec les informations d'expédition	2	1	0,25
Marge de 25 %			
Envoyer une notification au magasin expéditeur	3	1,75	0,5
Marge de 30 %			

Comparer les distances entre les magasins et/ou le centre de livraison et le domicile	5	2,75	0,75
---	---	------	------

2.3 Coûts :

Estimation des coûts basée sur le temps de travail prévu pour chaque tâche et le taux journalier moyen (TJM) des membres de l'équipe.

Tâche	Temps (en jour)	Membre de l'équipe	TJM	Total
Sans marge				
Gestion de projet	0,5	Architecte Logiciel	800 €	400 €
Créer l'application Backend en utilisant Spring Boot	0,5	Architecte Logiciel	800 €	400 €
Créer l'application Frontend en utilisant Angular	0,5	Architecte Logiciel	800 €	400 €
Correction Backend	0,5	Développeur Backend	700 €	350 €

Correction Frontend	0,5	Développeur Frontend	700 €	350 €
Créer la redirection de DeliveryFit vers le site de vente en ligne	0,5	Développeur Frontend	700 €	350 €
Marge de 20%				
Intercepter l'URL du site de vente en ligne et rediriger vers l'URL de DeliveryFit	1,25	Développeur Frontend	700 €	875 €
Transmettre les informations de livraison	1,25	Développeur Backend	700 €	875 €
Trouver le point de vente le plus proche	1,25	Développeur Backend	700 €	875 €
IHM - Créer la page de validation du panier avec les informations d'expédition	1,25	Développeur Frontend	700 €	875 €
Marge de 25 %				

Envoyer une notification au magasin expéditeur	2,25	Développeur Backend	700 €	1 575 €
Marge de 30 %				
Comparer les distances entre les magasins et/ou le centre de livraison et le domicile	3,5	Développeur Backend	700 €	2 450 €
Total	13,75	/	/	9 775 €

Total estimé : 9 775 €

Marge disponible : 225 €

2.4 Risques identifiés :

La gravité d'un risque est mesurée selon l'échelle suivante :

- **1** : Négligeable
- **2** : Mineure
- **3** : Modérée
- **4** : Majeure
- **5** : Catastrophique

La probabilité qu'un risque se produise est évaluée comme suit :

- **1** : Très probable
- **2** : Probable
- **3** : Possible
- **4** : Peu probable
- **5** : Très peu probable

L'impact d'un risque est calculé avec la formule :

Impact = Gravité × Probabilité

L'impact est classé selon les catégories suivantes :

- **Faible** : entre 1 et 6
- **Moyen** : entre 7 et 12
- **Élevé** : entre 13 et 25

Risque	Gravité	Probabilité	Impact	Stratégie de mitigation	Tâche liée
--------	---------	-------------	--------	-------------------------	------------

Dépassement du budget	4	3	12	Suivi hebdomadaire des coûts et ajustement des tâches si nécessaire.	Général
Retard dans le respect des délais	3	4	12	Revue des priorités et adaptation des sprints.	Général
Problème de performances	3	3	9	Optimisation régulière des algorithmes et tests de performance réguliers..	Général

Bugs et erreurs	4	2	8	Utilisation de TDD et d'outils de détection automatisés	<ul style="list-style-type: none">• Backend :<ul style="list-style-type: none">◦ Comparer les distances entre les magasins et le domicile◦ Envoyer une notification au magasin expéditeur◦ ...• Frontend :<ul style="list-style-type: none">◦ Transmettre les informations de livraison◦ Intercepter l'URL du site◦ ...
Failles de la sécurité	3	3	9	Audits réguliers et mise à jour des dépendances.	<ul style="list-style-type: none">• Authentification backend & frontend

2.5 Définition des objectifs de performance

Objectif de performance	Description	Mesure
-------------------------	-------------	--------

Temps de chargement des pages	Les pages doivent se charger en moins de 2s sur une connexion standard.	Google PageSpeed ou Lighthouse.
Temps de réponse des API	L'API doit renvoyer les résultats en moins de 500ms pour assurer une expérience fluide.	JMeter ou Postman.
Disponibilité du système	L'application doit être disponible 99,9% du temps.	Outils de monitoring.
Taux de réussite des tests	Atteindre un taux de réussite d'au moins 95% lors des tests unitaires et d'intégration.	JUnit, Jest ou Cypress..
Couverture des Tests Unitaires	Au moins 80% du code doit être couvert par des tests unitaires.	Jacoco ou SonarQube.

2.6 Définition des sprints

Organisation des Sprints :

Sprint	Durée (en jours)	Tâches principales
Sprint 1	1	Initialisation du projet et des environnements, gestion de projet

Sprint 2	8,25	Développement des fonctionnalités backend, algorithmes
Sprint 3	3	Développement frontend, intégration des API, formulaires.
Sprint 4	0,5	Démonstration client
Sprint 5	0,5	Corrections suite à la démonstration (backend & frontend simultanément)

2. Synthèse

Proposition 1 : Respect strict du budget de 10 000 €

La durée totale du projet est estimée à **13,75 jours** avec un coût de **9 775 €**, respectant ainsi le budget initial avec une petite marge de **225 €**. Cette version garantit la réalisation dans les délais et les coûts.

Répartition du Travail des Membres de l'Équipe :

- **Architecte Logiciel : 1,5 jours**
- **Développeur Backend : 8,75 jours**
- **Développeur Frontend : 3,5 jours**

Comparaison avec le budget Initial :

- **Budget initial alloué : 10 000 €**
- **Coût total de la proposition améliorée : 9 775 €**
- **Reste du budget : 225 €**

Cette proposition convient pour livrer une application fonctionnelle et stable, tout en respectant les contraintes budgétaires et de temps.

Proposition 2 : Version améliorée avec optimisation technique (+15 % du budget)

Cette version propose des améliorations techniques supplémentaires, tout en limitant l'augmentation du budget à **15 %**, soit un total de **11.500 €**. Les optimisations apportent de meilleures performances et une sécurité renforcée sans dépasser excessivement le budget initial.

Améliorations supplémentaires proposées :

- **Optimisation de la performance :**

Optimisation des temps de réponse API avec des algorithmes plus rapides et l'utilisation de la mise en cache.

- **Sécurité renforcée :**

En plus des protocoles JWT et OAuth2, un audit de sécurité sera réalisé pour minimiser les failles. Des mesures supplémentaires de protection des données seront également mises en place.

- **Tests de performance avancés :**

Utilisation d'outils tels que JMeter ou Postman pour garantir un temps de réponse inférieur à 300 ms sous diverses conditions.

Durée totale de réalisation :

La durée estimée pour cette version améliorée est de 14,5 jours de travail.

Répartition du Travail des Membres de l'Équipe :

- **Architecte Logiciel : 2 jours**

- Similaire à la proposition 1
- + Supervision des optimisations techniques

- **Développeur Backend : 10 jours**

- Similaire à la proposition 1
- + Optimisation des algorithmes
- + Sécurisation des API avec des pratiques avancées et audit de sécurité.

- **Développeur Frontend : 3,5 jours**

- Similaire à la proposition 1

- + Amélioration de l'expérience utilisateur

Comparaison avec le budget Initial :

- **Budget initial alloué :** 10 000 €
- **Coût total de la proposition améliorée :** 11 500 €
- **Dépassement du budget :** 1 500 € (+15 %)

Bien que cette version dépasse le budget initial de **1.500 €**, elle apporte des améliorations importantes en termes de sécurité et de performance, tout en optimisant l'expérience utilisateur. Cet investissement supplémentaire reste justifié par l'amélioration de la robustesse et de la rapidité de l'application.