



Spécification des Conditions requises pour l'Architecture

Projet : Concevez une nouvelle architecture afin de soutenir le développement de votre entreprise

Client : FOOSUS

Préparé par : Henrick AGNAME

—
N° de Version du Document : 0.1

Titre : Spécification des Conditions requises pour l'Architecture

Date de Version du Document : 16/06/2023

Revu par : Henrick AGNAME

Date de Révision : 16/06/2023

Table des Matières

1. Objet de ce document

2. Mesures du succès
3. Conditions requises pour l'architecture
4. Contrats de service business
5. Contrats de service application
6. Lignes directrices pour l'implémentation
7. Spécifications pour l'implémentation
8. Standards pour l'implémentation
9. Conditions requises pour l'interopérabilité
10. Conditions requises pour le management du service IT
11. Contraintes
12. Hypothèses

Objet de ce document

La Spécification des Conditions requises pour l'Architecture fournit un ensemble de déclarations quantitatives qui dessinent ce que doit faire un projet d'implémentation afin d'être conforme à l'architecture.

Une Spécification des Conditions requises pour l'Architecture constitue généralement un composant majeur du contrat d'implémentation, ou du contrat pour une Définition de l'Architecture plus détaillée.

Comme mentionné ci-dessus, la Spécification des Conditions requises pour l'Architecture accompagne le Document de Définition de l'Architecture, avec un objectif complémentaire : le Document de Définition de l'Architecture fournit une vision qualitative de la solution et tâche de communiquer l'intention de l'architecte.

La Spécification des Conditions requises pour l'Architecture fournit une vision quantitative de la solution, énumérant des critères mesurables qui doivent être remplis durant l'implémentation de l'architecture.

Mesures du succès

Métrique	Technique de mesure	Valeur cible	Justification	Notes supplémentaires
Nombre de bugs par release	Suivi des bugs dans un outil de gestion de bugs/JIRA	< 5	Un nombre faible de bugs indique une bonne qualité du code	Les bugs critiques doivent être traités en priorité
Temps de réponse	Surveillance des temps de	< 200 ms	Une réponse rapide améliore	Ceci est une moyenne,

moyen des services	réponse à l'aide d'outils de surveillance du réseau		l'expérience utilisateur	certains services peuvent avoir des temps de réponse plus longs
Disponibilité du système	Surveillance de la disponibilité du système avec des outils de surveillance du réseau	99.9%	Une haute disponibilité est essentielle pour l'expérience utilisateur et la satisfaction du client	Cela équivaut à environ 9 heures de temps d'arrêt non planifié par an
Taux de succès des transactions	Surveillance des transactions réussies et échouées	> 98%	Un taux de succès élevé indique que le système est fiable et performant	Les transactions échouées doivent être analysées pour identifier et résoudre les problèmes
Satisfaction des utilisateurs	Enquêtes de satisfaction des utilisateurs, feedback des utilisateurs	Score moyen > 4 sur 5	La satisfaction des utilisateurs est un indicateur clé de la qualité du système	Les retours négatifs doivent être analysés et abordés
Nombre d'adhésions d'utilisateurs par jour	Suivi des inscriptions d'utilisateurs dans la base de données du système	Augmentation de 10%	L'augmentation des adhésions d'utilisateurs indique une croissance de l'audience et une adoption accrue du système	Cela pourrait nécessiter des campagnes de marketing ou des améliorations de l'UX pour stimuler l'adhésion
Adhésion de producteurs alimentaires	Suivi des inscriptions des producteurs dans la base de données du système	Passer de 1/mois à 4/mois	L'augmentation des adhésions de producteurs améliore la variété des produits disponibles et la valeur du système pour les utilisateurs	Cela pourrait nécessiter des efforts de recrutement et de partenariat

Délai moyen de parution*	Suivi du temps entre la soumission et la publication des produits	Réduit de 3,5 semaines à moins d'une semaine	Un délai de parution plus court permet une rotation plus rapide des produits et une meilleure réactivité aux préférences des utilisateurs	Cela pourrait nécessiter des améliorations du processus ou de l'automatisation
Taux d'incidents de production P1	Suivi des incidents P1 dans un outil de gestion des incidents	Pour commencer : réduit de >25/mois à moins de 1/mois	Un faible taux d'incidents P1 indique une stabilité et une fiabilité élevées du système	Les incidents P1 doivent être traités en priorité et analysés pour éviter leur récurrence

Conditions requises pour l'architecture

Condition Requise	Description
Scalabilité	Le système doit pouvoir supporter la croissance future de la base d'utilisateurs sans compromettre les performances.
Réactivité	Le système doit offrir une expérience utilisateur réactive, avec des temps de réponse rapides pour assurer une utilisation fluide et agréable.
Disponibilité	Le système doit être hautement disponible, garantissant un accès continu aux fonctionnalités et services, avec une faible tolérance aux interruptions.
Sécurité	Le système doit mettre en place des mesures de sécurité solides pour protéger les données des utilisateurs et assurer la confidentialité des informations.
Intégrité des données	Le système doit garantir l'intégrité des données, en veillant à ce qu'elles soient exactes, cohérentes et non altérées tout au long du processus.

Interopérabilité	Le système doit être capable de s'intégrer avec d'autres systèmes et services externes de manière fluide et efficace.
Évolutivité	Le système doit être évolutif, capable de s'adapter à de nouvelles fonctionnalités, technologies et exigences sans remettre en question sa structure de base.

Contrats de service business

Contrat de Service	Description
Disponibilité du Service	Le service doit être disponible 24 heures sur 24, 7 jours sur 7, avec un objectif de temps d'arrêt minimal pour assurer l'accessibilité constante de la plateforme.
Temps de réponse	Le service doit répondre aux requêtes des utilisateurs dans un délai maximum de 2 secondes, garantissant ainsi une expérience réactive et fluide.
Support Utilisateur	Une équipe de support dédiée doit être disponible pour aider les utilisateurs, répondre à leurs questions et résoudre leurs problèmes dans un délai raisonnable.
Sécurité des Données	Les données des utilisateurs doivent être protégées de manière rigoureuse, conformément aux meilleures pratiques de sécurité et aux réglementations en vigueur.
Mises à jour régulières	Le service doit bénéficier de mises à jour régulières pour améliorer les fonctionnalités, corriger les bugs et assurer la stabilité et la performance de la plateforme.
Confidentialité des Données	Les informations personnelles des utilisateurs doivent être traitées avec la plus grande confidentialité, conformément aux politiques de confidentialité et aux lois applicables.
Support plateforme Multi-	Le service doit être accessible sur différentes plateformes (mobile, web) pour permettre aux utilisateurs de se connecter de manière pratique à partir de différents appareils.

Intégration avec des Tiers	Le service doit offrir des possibilités d'intégration avec des partenaires et des services tiers, permettant ainsi une collaboration et une fonctionnalité étendue pour les utilisateurs.
Mesure de Satisfaction	La satisfaction des utilisateurs sera mesurée régulièrement pour évaluer la qualité du service, identifier les points d'amélioration et garantir une expérience utilisateur optimale.

Accords de niveau de service

Service : Plateforme d'application mobile Foosus

Objectif : Fournir une expérience utilisateur fluide et fiable pour les utilisateurs de l'application mobile Foosus.

1. Disponibilité du service :
 - Le service sera disponible 24 heures sur 24, 7 jours sur 7, avec une disponibilité cible de 99,9%.
 - Les temps d'arrêt planifiés pour la maintenance seront annoncés au moins 48 heures à l'avance. (À éviter)
 - Lors de la maintenance la plateforme devra rester disponible à l'utilisateur avec une réduction d'accès services.
2. Temps de réponse :
 - Le temps de réponse moyen pour les requêtes utilisateur ne dépassera pas 1 secondes.
 - Les requêtes critiques nécessitant un traitement plus long seront gérées de manière transparente avec des indicateurs de progression clairs pour l'utilisateur.
3. Performances :
 - Les pages de l'application mobile doivent se charger rapidement, avec un temps de chargement moyen inférieur à 2 secondes.
 - Les fonctionnalités clés, telles que la recherche de producteurs locaux et la passation de commandes, doivent être hautement réactives et ne présenter aucun délai perceptible.
4. Sécurité et confidentialité des données :
 - Les données personnelles des utilisateurs seront protégées conformément aux réglementations en vigueur, telles que le Règlement général sur la protection des données (RGPD).
 - Les mesures de sécurité, telles que le chiffrement des données en transit et au repos, seront mises en place pour prévenir tout accès non autorisé ou toute violation de données.
5. Support utilisateur :
 - Un support utilisateur sera disponible par le biais d'une assistance en ligne et d'un système de tickets pour répondre aux demandes et aux problèmes des

utilisateurs.

- Les demandes de support seront traitées dans les délais suivants : les problèmes critiques seront résolus dans les 24 heures, les problèmes majeurs dans les 48 heures et les demandes générales dans les 72 heures.

Service : Web Foosus

Objectif : Fournir une expérience utilisateur conviviale et informative aux visiteurs du site internet de Foosus.

Accord de niveau de service :

1. Disponibilité du service :

- Le site sera disponible 24 heures sur 24, 7 jours sur 7, avec une disponibilité cible de 99,9%.
- Les temps d'arrêt planifiés pour la maintenance seront annoncés au moins 48 heures à l'avance.

2. Temps de chargement :

- Le site se chargera rapidement, avec un temps de chargement moyen inférieur à 1 seconde pour les pages principales.
- Les pages contenant des éléments interactifs ou des médias plus lourds seront optimisées pour assurer des temps de chargement raisonnables.

3. Accessibilité :

- Le site sera conçu et développé en respectant les normes d'accessibilité Web afin de garantir une expérience utilisable pour tous les visiteurs, y compris les personnes handicapées.

4. Contenu à jour :

- Le contenu du site, y compris les informations sur les producteurs et les artisans locaux, sera régulièrement mis à jour pour refléter les dernières informations disponibles.
- Les promotions, les événements et les actualités pertinents seront publiés en temps opportun pour tenir les visiteurs informés.

5. Sécurité du site :

- Des mesures de sécurité seront mises en place pour protéger le site contre les attaques et les tentatives de piratage.
- Les données des visiteurs, telles que les informations de contact fournies via les formulaires, seront traitées de manière confidentielle et conformément aux réglementations en vigueur.

6. Support utilisateur :

- Un support utilisateur sera disponible par le biais d'une assistance en ligne et d'un système de contact pour répondre aux questions et aux demandes des visiteurs.
- Les demandes de support seront traitées dans les délais suivants : les demandes critiques seront résolues dans les 24 heures, les demandes majeures dans les 48 heures et les demandes générales dans les 72 heures.

Lignes directrices pour l'implémentation

Lignes directrices pour l'implémentation	Description
Développement Agile	Adopter une approche de développement agile pour favoriser la collaboration, l'itération rapide et l'adaptation aux changements. Utiliser la méthodologie Agile Scrum pour organiser les travaux et favoriser la flexibilité.
Architecture microservices	La structure du système sera basée sur une architecture microservices, où chaque service aura une responsabilité unique et communiquera avec les autres via des interfaces REST. Cela permettra une évolutivité, une maintenabilité et une réutilisation efficaces.
Intégration continue et déploiement continu	Mettre en place des processus d'intégration continue et de déploiement continu pour automatiser les tests, la construction, la validation et le déploiement des nouvelles fonctionnalités. Cela permettra des livraisons fréquentes et fiables de version. (jenkins)
Sécurité et confidentialité des données	Mettre en place des mesures de sécurité robustes pour protéger les données sensibles des utilisateurs, telles que le chiffrement des données en transit et au repos, la gestion des identités et des accès, et la conformité aux réglementations en matière de protection des données.
Évolutivité	Concevoir l'architecture de manière à pouvoir faire évoluer le système en ajoutant de nouvelles fonctionnalités, en gérant une augmentation du nombre d'utilisateurs et en s'adaptant à des charges de travail plus importantes. Mise à l'échelle horizontale pour répondre à ces besoins.
Surveillance et gestion des performances	Mettre en place des outils de surveillance et de gestion des performances pour surveiller l'état du système, détecter les problèmes de performance et y remédier rapidement.
Documentation complète	Fournir une documentation complète de l'architecture, des composants, des interfaces et des processus, afin de faciliter la compréhension, la maintenance et l'évolution du système. Cette documentation doit être accessible à toutes les parties prenantes concernées.
Collaboration interfonctionnelle	Encourager la collaboration étroite entre les équipes de développement, d'architecture, de conception, et de test pour favoriser une compréhension partagée des objectifs, des contraintes et des décisions

	architecturales, ainsi qu'une résolution efficace des problèmes.
Tests et validation approfondis	Mettre en place des processus de test rigoureux pour valider les fonctionnalités et garantir leur qualité. Cela inclut les tests unitaires, les tests d'intégration, les tests de performance, les tests de sécurité. L'automatisation des tests est encouragée, avec un taux de couverture de tests à 85%.
Maintenance et support continus	Assurer une maintenance continue du système en identifiant et en corrigeant les problèmes, en appliquant des correctifs de sécurité, en améliorant les fonctionnalités existantes.

Spécifications pour l'implémentation

Spécifications pour l'implémentation	Description
Langage de programmation et framework	Java, Spring, Angular.
Architecture logicielle	Mettre en place une architecture microservices.
Base de données	Mettre en place Oracle.
Sécurité et authentification	API Gateway protocoles HTTPS, la gestion de rôles , authentification à double facteur, génération de token, OAuth.
Interfaces utilisateur	Interface web, une application mobile.
Intégrations externes	Services de paiement : Protocole HTTPS, format des données JSON, authentification OAuth, clé API, Token. Services de géolocalisation : protocole HTTPS, format des données JSON, authentification API.

8. Gestion des erreurs et des exceptions	<p>1. Gestion des exceptions :</p> <ul style="list-style-type: none"> • Toutes les exceptions doivent être attrapées et traitées de manière appropriée pour éviter que l'application ne plante. • Le système doit utiliser un gestionnaire d'exceptions global qui attrape toutes les exceptions non gérées. • Pour les exceptions prévisibles liées à des cas d'utilisation spécifiques (par exemple, échec d'une opération de base de données), des exceptions personnalisées doivent être définies et utilisées. <p>2. Journalisation des erreurs :</p> <ul style="list-style-type: none"> • Toutes les exceptions et erreurs doivent être consignées dans un journal avec suffisamment d'informations pour permettre le débogage, y compris l'heure de l'erreur, l'exception ou le message d'erreur, la pile d'appels et le contexte de l'erreur. • Les logs doivent être conservés dans un endroit sécurisé et accessible, comme un système de gestion de logs centralisé Log4j. <p>3. Messages pour les utilisateurs :</p> <ul style="list-style-type: none"> • Les utilisateurs doivent recevoir des messages d'erreur clairs et compréhensibles lorsqu'une opération échoue. • Les messages d'erreur ne doivent pas exposer des détails techniques qui pourraient être exploités pour des attaques de sécurité. <p>4. Notifications d'erreur :</p> <ul style="list-style-type: none"> • Les erreurs critiques doivent déclencher une alerte à l'équipe de développement ou au service d'opération, par exemple via email ou SMS.

	<p>5. Reprise après erreur :</p> <ul style="list-style-type: none"> L'application doit avoir une stratégie pour se rétablir après une erreur. Par exemple, pour une transaction de base de données qui échoue, l'application peut réessayer l'opération un certain nombre de fois avant d'échouer définitivement.
9. Documentation et commentaires	Assurer une documentation claire et complète pour faciliter la compréhension et la maintenance du code.
10. Tests et validation	Spécifier les types de tests à effectuer, tels que les tests unitaires, les tests d'intégration, les tests de régression. Définir les scénarios de test et les critères de succès pour assurer la qualité de l'application.

Standards pour l'implémentation

Standard pour l'implémentation	Description
Conventions de codage	Utiliser la convention java définie par Oracle pour assurer une cohérence dans la syntaxe, le nommage des variables.
Bonnes pratiques de développement	Adopter l'approche Agile et pratiquer la méthodologie Scrum pour favoriser une collaboration efficace, des itérations rapides et une flexibilité dans le développement de l'application.
Sécurité des données	API Gateway protocoles HTTPS, la gestion de rôles, authentification à double facteur, génération de token, OAuth.
Gestion des versions	Utiliser Git comme système de contrôle de version pour gérer le code source. Suivre la convention de nommage des branches (feature branches, release branches, etc.) et effectuer des fusions régulières pour maintenir une base de code propre et à jour.
Documentation du code	Fournir des commentaires clairs et concis dans le code pour expliquer les fonctionnalités, les algorithmes complexes et les dépendances

	importantes.
Tests unitaires	Mettre en place des tests unitaires pour chaque fonctionnalité ou module clé de l'application.
Intégration continue	Configurer un processus d'intégration continue à l'aide d'outils tels que Jenkins.
Documentation utilisateur	Créer une documentation utilisateur détaillée expliquant les fonctionnalités de l'application, les étapes d'inscription, les processus de recherche et de commande.
Maintenance évolutive et	Concevoir l'architecture de l'application de manière modulaire pour permettre des mises à jour et des ajouts de fonctionnalités sans perturber l'ensemble du système. Adopter des modèles de conception tels que MVC (Modèle-Vue-Contrôleur) pour séparer les responsabilités et faciliter la maintenance future.
Suivi des performances	Intégrer des outils de surveillance des performances tels que New Relic et Kibana pour collecter des métriques en temps réel sur les performances de l'application. Analyser ces données pour détecter les goulots d'étranglement et optimiser les parties critiques de l'application.

Conditions requises pour l'interopérabilité

Le système doit être capable de s'interfacer et de communiquer efficacement avec des systèmes tiers, tels que des services de paiement en ligne, des fournisseurs de données externes.

Critères d'acceptation :

1. Le système doit prendre en charge des protocoles standard tels que HTTP, REST, JSON, XML, etc., pour permettre l'intégration avec d'autres systèmes.
2. Le système doit fournir une API bien documentée et facilement accessible pour permettre aux systèmes tiers de s'y connecter.
3. Les données échangées entre le système et les systèmes tiers doivent être correctement formatées et structurées pour garantir une capacité à opérer de manière interactive.
4. Le système doit prendre en charge des mécanismes d'authentification et de sécurité

appropriés pour protéger les échanges de données avec les systèmes tiers.

5. Le système doit être capable de gérer et de traiter les réponses et les erreurs renvoyées par les systèmes tiers de manière appropriée, en fournissant des messages d'erreur clairs et des mécanismes de reprise en cas d'échec de la communication.
6. Le système doit être régulièrement testé pour garantir la compatibilité et l'interopérabilité avec les systèmes tiers, en utilisant des outils et des protocoles de test appropriés.
7. Le système doit prendre en compte les mises à jour et les évolutions des protocoles et des interfaces utilisés par les systèmes tiers, en assurant une compatibilité continue et en effectuant les ajustements nécessaires en cas de changements.
8. Les intégrations avec les systèmes tiers doivent être documentées de manière précise et à jour, y compris les spécifications des interfaces, les paramètres d'authentification, les limites de taux...
9. Le système doit être évolutif pour permettre l'ajout et la gestion de nouvelles intégrations avec des systèmes tiers au fur et à mesure des besoins de l'entreprise.
10. Des mécanismes de surveillance et de journalisation appropriés doivent être mis en place pour suivre et analyser les performances et la disponibilité des intégrations avec les systèmes tiers.

Conditions requises pour le management du service IT

Le système doit être géré de manière efficace et efficiente pour assurer une disponibilité continue, une performance optimale et une gestion proactive des incidents et des problèmes.

Critères d'acceptation :

1. Le système doit être surveillé en permanence pour détecter les incidents, les erreurs de performance ou les comportements anormaux. Des outils de surveillance appropriés doivent être en place pour collecter les métriques et les journaux pertinents.
2. Un processus de gestion des incidents doit être établi pour enregistrer, prioriser, diagnostiquer et résoudre rapidement les incidents. Les tickets d'incident doivent être suivis, attribués aux équipes responsables et résolus dans les délais convenus.

3. Un processus de gestion des problèmes doit être en place pour analyser les causes profondes des incidents récurrents, les problèmes de performance ou les pannes. Des actions correctives doivent être prises pour prévenir leur récurrence.
4. Des indicateurs clés de performance (KPI) doivent être définis pour mesurer la disponibilité du système, le temps de réponse, le taux de résolution des incidents, etc. Ces KPI doivent être régulièrement surveillés et rapportés aux parties prenantes concernées.
5. Un processus de gestion des changements doit être suivi pour gérer les mises à jour logicielles, les mises à niveau matérielles et toute modification de l'infrastructure. Les changements doivent être planifiés, testés, autorisés et documentés de manière appropriée.
6. Une documentation complète et à jour doit être disponible pour décrire l'architecture du système, les composants, les configurations, les dépendances et les procédures opérationnelles standard. Cette documentation doit être accessible à l'équipe IT et aux parties prenantes concernées.
7. Un plan de continuité des activités et un plan de reprise après sinistre doivent être établis pour minimiser les perturbations en cas de défaillance majeure du système. Des sauvegardes régulières des données critiques doivent être effectuées et testées.
8. Des processus de gestion des fournisseurs et des contrats de service doivent être en place pour assurer la disponibilité des ressources externes nécessaires, tels que l'hébergement cloud, les services de support technique, etc. Les niveaux de service convenus doivent être respectés.
9. Des politiques de sécurité informatique doivent être mises en œuvre pour protéger les données sensibles, prévenir les intrusions et assurer la confidentialité, l'intégrité et la disponibilité du système. Des mécanismes d'authentification et de chiffrement appropriés doivent être utilisés.
10. Les responsabilités et les rôles doivent être clairement définis au sein de l'équipe IT, et les membres de l'équipe doivent être formés sur les procédures opérationnelles et les bonnes pratiques.