GUIDE  
Développeur Full Stack

Table des matières

[1. A1 - Participer à la gestion d'un projet d’application ou de site web 10](#_Toc137992233)

[1.1. C1- Participer à une réunion en identifiant la typologie du demandeur (client) pour comprendre sa demande et participer à l’élaboration d’une réponse adaptée à ses besoins 10](#_Toc137992234)

[1.1 C2- Conseiller le Maître d’Ouvrage en mettant à contribution ses connaissances et compétences techniques pour la réalisation du cahier des charges 10](#_Toc137992235)

[1.2 C3 - Participer à l’élaboration d’une planification réaliste en tenant compte de ses propres contraintes et compétences afin de garantir l’atteinte des objectifs fixés pour la réussite du projet 10](#_Toc137992236)

[1.3 C4 - Concevoir une ou plusieurs maquettes « wireframe » en utilisant un outil spécialiser afin de fournir une ébauche au demandeur 10](#_Toc137992237)

[1.4 C5 - Mettre en œuvre un environnement de développement collaboratif adapté au projet d’application afin d’optimiser le temps de développement, le transfert de compétences auprès de ses pairs et la qualité logicielle 10](#_Toc137992238)

[1.4.1 C5.1 - Mettre en œuvre et maîtriser l’utilisation d’un système de gestion de code source distribué (SCM) permettant de conserver l’historique des développements, d’organiser la collaboration des développeurs et d’appliquer un workflow standard pour la revue de code et le suivi des bugs. 10](#_Toc137992239)

[1.4.2 C5.2 - Mettre en œuvre un environnement de développement local ou distribué reproductible reposant sur l’utilisation de la virtualisation système ou de la virtualisation applicative 11](#_Toc137992240)

[1.4.3 C5.3 Utiliser un environnement de développement intégré (IDE) adapté aux technologies sur lesquelles repose l’application et tirer parti de ses fonctionnalités 11](#_Toc137992241)

[1.4.4 C5.4 - Sélectionner et mettre en œuvre les outils et méthodes Agile de développement afin de collaborer efficacement avec les acteurs du projet d’application 11](#_Toc137992242)

[1.5 C6 - Comprendre le cycle de développement et mettre en œuvre les principales méthodes de gestion de projet de développement afin de les appliquer au sein d’une équipe (XP, SCRUM, DSDM, ASD). 11](#_Toc137992243)

[1.6 C7 - Rédiger des compte-rendu d’activité destinés aux membres d’une équipe de projet afin de permettre le suivi de l’avancement du projet et la traçabilité des réalisations techniques 11](#_Toc137992244)

[2 A2 - Concevoir et modéliser une application ou un site web 12](#_Toc137992245)

[2.1 C8 - Rédiger les spécifications techniques de besoin (STB) d’un projet d’application ou de site web à partir d’un cahier des charges afin de décrire de manière exhaustif les exigences à satisfaire en termes d’utilisation 12](#_Toc137992247)

[2.2 C9 - Modéliser l’application logicielle et ses données en utilisant une méthode standard afin d’abstraire la réalité et mieux comprendre le système à réaliser 12](#_Toc137992248)

[2.2.1 C9.1 - Utiliser le formalisme du langage de modélisation unifié UML et de UP afin de traduire les besoins de l’application 12](#_Toc137992249)

[2.2.2 C9.2 - Constituer le dossier de modélisation comprenant les diagrammes, les cas d’utilisation, les classes d’analyse et décrivant l’architecture logicielle n-tiers de l’application 12](#_Toc137992250)

[2.2.3 C9.3 - Utiliser des patrons de conceptions afin d’appliquer des solutions logicielles standards 12](#_Toc137992251)

[2.3 C10 - Concevoir l’architecture des bases de données à l’aide d’un outil de modélisation afin de représenter la structure de la couche de persistance nécessaire au fonctionnement de l’application ou du site web 13](#_Toc137992252)

[2.3.1 C10.1 - Recenser les informations / données du domaine 13](#_Toc137992253)

[2.3.2 C10.2 - Établir le schéma entité-association des données et définir le schéma de la base de données afin de permettre l’élaboration d’une base de données normalisée 13](#_Toc137992254)

[2.3.3 C10.3 - Dans le cas d’une base de données existante, déduire le schéma entité-association des données de la base à faire évoluer 13](#_Toc137992255)

[2.4 C11 - À partir du dossier de spécifications fonctionnelles et techniques, déterminer l’architecture logicielle de l’application ou du site web 13](#_Toc137992256)

[3 A3 - Développer la partie front-end d’une application ou d’un site web en appliquant les bonnes pratiques d’UX, de sécurité informatique et d’éco-conception 14](#_Toc137992257)

[3.1 C12 - Développer la partie front-end d’une application web, hybride, mobile ou desktop en utilisant plusieurs langages de programmation et en appliquant les bonnes pratiques d’UX, de sécurité informatique et d’éco-conception 14](#_Toc137992259)

[3.1.1 C12.1 - À partir d’une maquette et/ou d’une charte graphique, structurer et intégrer une interface utilisateur responsive pour le web 14](#_Toc137992260)

[3.1.2 C12.2 - À partir d’une maquette, structurer et intégrer une interface utilisateur mobile first ou desktop en utilisant une technologie hybride (web app / progressive web app) 14](#_Toc137992261)

[3.1.3 C12.4 - Sélectionner et mettre en œuvre un ou plusieurs langages de programmation front-end adaptés aux exigences et contraintes de l’application ou du site web 15](#_Toc137992262)

[3.1.4 C12.5 - Sélectionner et mettre en oeuvre un framework de développement front-end adapté aux exigences et contraintes de l’application ou du site web 15](#_Toc137992263)

[3.1.5 C12.6 - Développer des composants front-end graphiques et / ou fonctionnels en choisissant des structures de données adaptées et des algorithmes pertinents afin d’assurer la qualité logicielle 15](#_Toc137992264)

[3.2 C12.7 - Appliquer les bonnes pratiques UX (expérience utilisateur) 15](#_Toc137992265)

[3.3 C12.8 - Appliquer les recommandations de sécurité pour le développement front-end afin de s’inscrire dans une démarche de « sécurité en profondeur » 15](#_Toc137992266)

[3.4 C13 - Tester la partie front-end de l’application à plusieurs niveaux en utilisant des méthodes de test standards afin de garantir sa conformité vis-à-vis des spécifications et assurer la non-régression des composants implémentés 16](#_Toc137992267)

[3.4.1 C13.1 - Élaborer un plan de test logiciel exhaustif 16](#_Toc137992268)

[3.4.2 C13.2 - Réaliser des tests de conformité du langage de balisage structurant le contenu des interfaces web, mobile hybride et desktop 17](#_Toc137992269)

[3.4.3 C13.3 - Réaliser des tests unitaires pour les composants fonctionnels 17](#_Toc137992270)

[3.4.4 C13.4 - Réaliser des tests fonctionnels, utilisateurs et de compatibilité avec les différentes plateformes 17](#_Toc137992271)

[3.5 C14 - Industrialiser le développement de la partie front-end de l’application et automatiser les processus d’assurance qualité 18](#_Toc137992272)

[3.5.1 C14.1 - Automatiser l’exécution d’un vérificateur / correcteur de syntaxe pour chaque langage de programmation constituant le code source de la partie front-end, si nécessaire. 18](#_Toc137992273)

[3.5.2 C14.2 - Utiliser un gestionnaire de paquets pour automatiser la gestion des dépendances (composants tiers) 18](#_Toc137992274)

[3.5.3 C14.3 - Utiliser un groupeur de modules afin de configurer et d’exécuter la chaine de build et ainsi optimiser les performances de l’interface 18](#_Toc137992275)

[4 A4 - Développer la partie back-end d’une application ou d’un site web en appliquant les bonnes pratiques de sécurité informatique, d’éco-conception et DevOps 20](#_Toc137992276)

[4.1 C15 - Développer la couche de persistance des données en appliquant les bonnes pratiques de sécurité informatique et de performance 20](#_Toc137992278)

[4.1.1 C15.1 - Mettre en œuvre et administrer des bases de données relationnelles de manière sécurisée 21](#_Toc137992279)

[4.1.2 C15.2 - Exploiter et interroger des bases de données relationnelles depuis le back-end de l’application 21](#_Toc137992280)

[4.1.3 C15.3 - Mettre en œuvre et administrer des bases de données NoSQL de manière sécurisée 21](#_Toc137992281)

[4.1.4 C15.4 - Exploiter et interroger des bases de données NoSQL 21](#_Toc137992282)

[4.1.5 C15.5 - Faire évoluer les bases de données 21](#_Toc137992283)

[4.2 C16 - Développer la partie back-end d’une application en utilisant plusieurs langages de programmation et en appliquant les bonnes pratiques de sécurité informatique et d’éco-conception 21](#_Toc137992284)

[4.2.1 C16.1 - Sélectionner et mettre en œuvre un ou plusieurs langages de programmation back-end adaptés aux exigences et contraintes de l’application ou du site web 21](#_Toc137992285)

[4.2.2 C16.2 - Sélectionner et mettre en œuvre un framework de développement back-end adaptés aux exigences et contraintes de l’application ou du site web 21](#_Toc137992286)

[4.2.3 C16.3 - À partir du référentiel de modélisation, développer des composants back-end fonctionnels en choisissant des structures de données adaptées et des algorithmes pertinents afin d’assurer la qualité logicielle 22](#_Toc137992287)

[4.2.4 C16.4 - Appliquer les recommandations de sécurité pour le développement back-end afin de s’inscrire dans une démarche de « sécurité en profondeur » 22](#_Toc137992288)

[4.2.5 C16.5 - Appliquer les bonnes pratiques d’écoconception afin de minimiser l’impact écologique de la partie back-end de l’application 22](#_Toc137992289)

[4.3 C17 - Consommer une API de manière sécurisée en s’appuyant sur des mécanismes d’identification et d’authentification robustes afin d’intégrer des sources de données et des services tiers 22](#_Toc137992290)

[4.4 C18 - Développer une API permettant à des services tiers de consommer des données produites par l’application au travers de formats ouverts 22](#_Toc137992291)

[4.4.1 C18.2 - Sélectionner et mettre en oeuvre un format d’échange de données adapté aux caractéristiques de l’application et à son environnement (consommateurs d’API) 22](#_Toc137992292)

[4.4.2 C18.3 - Sécuriser l’accès à l’API en implémentant des mécanismes d’identification et d’authentification robustes 22](#_Toc137992293)

[4.4.3 C18.4 - Documenter et publier une API 22](#_Toc137992294)

[4.5 C19 - Tester la partie back-end de l’application à plusieurs niveaux en utilisant des méthodes de test standards afin de garantir sa conformité vis-à-vis des spécifications et assurer la non-régression des composants implémentés 22](#_Toc137992295)

[4.5.1 C19.1 - Élaborer un plan de tests logiciels exhaustif pour le back-end 22](#_Toc137992296)

[4.5.2 C19.2 - Réaliser des tests unitaires sur les composants du back-end 22](#_Toc137992297)

[4.5.3 C19.3 - Réaliser des tests fonctionnels sur les composants du back-end 22](#_Toc137992298)

[4.6 C20 - Industrialiser le développement de la partie back-end de l’application ou du site web et automatiser les processus d’assurance qualité 23](#_Toc137992299)

[4.6.1 C20.1 - Mettre en oeuvre un processus d’intégration continue de manière à minimiser les erreurs humaines et de rationaliser le cycle de développement de la partie back-end de l’application. 23](#_Toc137992300)

[4.6.2 C20.2 - Utiliser un gestionnaire de paquets pour automatiser la gestion des dépendances (composants tiers) de la partie back-end de l’application 23](#_Toc137992301)

[4.6.3 C20.3 - Automatiser la construction de l’application en configurant la chaînes de build, l’exécution des tests, l’exécution des outils d’assurance qualité et l’analyse statique de sécurité. 23](#_Toc137992302)

[4.7 C21 - Faire évoluer ses connaissance et compétences en développement back-end 23](#_Toc137992303)

[4.7.1 C21.1 - Formaliser et mettre en œuvre une démarche de veille technologique 23](#_Toc137992304)

[4.7.2 C21.2 - Utiliser un outil de knowledge management afin de capitaliser la connaissance et faciliter sa transmission 23](#_Toc137992305)

[5 A5 - Déployer et assurer le maintien en production d’une application ou d’un site web en appliquant les bonnes pratiques de sécurité informatique et DevOps (Phase livraison) 24](#_Toc137992306)

[5.1 C22 - Préparer, mettre en œuvre et automatiser la mise en production d’une application dans différents types d’environnements techniques d’exploitation 24](#_Toc137992308)

[5.1.1 C22.1 - Identifier les différents types de plateformes d’hébergement et sélectionner une plateforme d’hébergement pertinente et adaptée aux exigences techniques de l’application(dimensionnement) en prenant en compte l’optimisation des coûts 24](#_Toc137992309)

[5.1.2 C22.2 - Définir l’architecture, mettre en oeuvre et administrer des services d’hébergement dans un environnement dédié, cloud ou conteneurisé en appliquant les bonnes pratiques de sécurité 25](#_Toc137992310)

[5.1.3 C22.3 - Mettre en œuvre et exploiter un système d’orchestration de conteneurs 25](#_Toc137992311)

[5.1.4 C22.4 - Mettre en œuvre un environnement de qualification externe 25](#_Toc137992312)

[5.1.5 C22.5 - Élaborer un diagramme de déploiement et mettre en œuvre un système de déploiement automatisé d’une application pour différents types d’hébergement, avec ou sans processus d’intégration continue, en respectant les bonnes pratiques DevOps 25](#_Toc137992313)

[5.1.6 C22.6 - Automatiser la mise en œuvre des architectures à l’aide d’un gestionnaire de configuration ou d’un système d’ « Infrastructure as Code » 25](#_Toc137992314)

[5.1.7 C22.7 - Gérer des noms de domaines, administrer les serveurs de noms de domaine, et mettre en œuvre des certificats de sécurité 25](#_Toc137992315)

[5.2 C23 - Sécuriser et superviser l’application et l’environnement technique d’exploitation afin d’en assurer le maintien en production 25](#_Toc137992316)

[5.2.1 C23.1 - Sécuriser les services et les réseaux d’une architecture d’hébergement 25](#_Toc137992317)

[5.2.2 C23.2 - Mettre en oeuvre les liaisons sécurisées entre les différents services et les composants de l’application 25](#_Toc137992318)

[5.2.3 C23.3 - Mettre en oeuvre des systèmes de sauvegarde et de redondance de l’infrastructure d’hébergement 26](#_Toc137992319)

[5.2.4 C23.4 - Identifier les métriques et indicateurs de performance de l’application et des services d’hébergement correspondants ; définir un « budget » performance 26](#_Toc137992320)

[5.2.5 C23.5 - Réaliser un audit de performance d’une application et des services d’hébergement correspondants 26](#_Toc137992321)

[5.2.6 C23.6 - Rédiger un rapport d’audit de performance de l’application et des services d’hébergement correspondants; émettre des préconisations 26](#_Toc137992322)

[5.2.7 C23.7 - Préparer et déployer des mesures correctives pour l’amélioration des performances de l’application et de son environnement 26](#_Toc137992323)

[5.2.8 C23.8 - Mettre en œuvre un système de journalisation et implémenter une stratégie d’audit des journaux de l’application 26](#_Toc137992324)

[5.2.9 C23.9 - Mettre en œuvre un système de supervision d’une application et de services d’hébergement, définir des sondes et des alertes (état des services et sécurité) 26](#_Toc137992325)

[5.2.10 C23.10 - Détecter, diagnostiquer et analyser l’origine des bugs et problèmes techniques rencontrés 26](#_Toc137992326)

[5.2.11 C23.11 - Préparer et déployer des mesures correctives pour les erreurs ou les failles de sécurité identifiées dans des délais adaptés 27](#_Toc137992327)

[5.3 C24 - Produire la documentation technique de l’application et alimenter une base de connaissances afin de capitaliser l’information et d’effectuer un transfert de compétences auprès de ses pairs 27](#_Toc137992328)

[5.3.1 C24.1 - Commenter son code source en respectant les standards et bonnes pratiques en la matière afin de permettre la génération automatique de documentation 27](#_Toc137992329)

[5.3.2 C24.2 - Mettre en oeuvre la génération automatique de documentation du code source afin de l’intégrer dans un processus d’intégration continue 27](#_Toc137992330)

[5.3.3 C24.3 - Rédiger la documentation technique d’un projet afin d’effectuer un transfert de compétences auprès de ses paris 27](#_Toc137992331)

[5.3.4 C24.4 - Rédiger la documentation d’une API afin de faciliter sa consommation par des tiers 27](#_Toc137992332)

[6 A6 - Mettre en oeuvre des solutions techniques répondant aux besoins contextuels d’une application ou d’un site web (lois, normes et règlements, accessibilité, outils marketing et e-commerce) et piloter les performances. 28](#_Toc137992333)

[6.1 C25 - Mettre en oeuvre les outils et techniques permettant de respecter les aspects réglementaires au regard des données informatiques traitées et stockées 28](#_Toc137992335)

[6.1.1 C25.1 - Identifier les données et traitements informatiques concernés par les lois, normes et règlements (CNIL, RGPD,…) 28](#_Toc137992336)

[6.1.2 C25.2 - Implémenter les moyens techniques et technologiques permettant d’assurer le respect des lois, normes et règlements (CNIL, RGPD,…) applicables aux données traitées par une application et stockées dans une base de données. 28](#_Toc137992337)

[6.1.3 C25.3 - Connaître et respecter les lois et règlements en matière de propriété intellectuelle au regard des composants logiciels et codes sources informatiques 28](#_Toc137992338)

[6.2 C26 - Mettre en oeuvre les outils et techniques permettant de respecter les aspects réglementaires au regard de l’accessibilité numérique 28](#_Toc137992339)

[6.2.1 C26.1 - Identifier, dans une application, les éléments de présentation (UX), traitements et contenus informationnels concernés par les obligations d’accessibilité numérique. 29](#_Toc137992340)

[6.2.2 C26.2 - Implémenter les moyens techniques et technologiques permettant d’assurer le respect des obligations d’accessibilité numérique (RGAA - Référentiel général d'amélioration de l ’accessibilité). 29](#_Toc137992341)

[6.3 C27 - Mettre en oeuvre les outils et techniques permettant de mesurer et de suivre les performances marketing d’une application 29](#_Toc137992342)

[6.3.1 C27.1 - Identifier les outils et techniques adaptés et pertinents au regard du contexte d’une application afin de mesurer et de suivre ses performances marketing 29](#_Toc137992343)

[6.3.2 C27.2 - Implémenter les outils et techniques permettant de mesurer et de suivre les performances marketing d’une application (analytics) 29](#_Toc137992344)

[6.4 C28 - Mettre en oeuvre les outils et techniques permettant d’améliorer les performances SEO (Search Engine Optimization / référencement naturel) d’une application logicielle 29](#_Toc137992345)

[6.4.1 C28.1 - Identifier les critères techniques d’une application ayant un impact sur le référencement naturel : les facteurs bloquants et les facteurs favorisants 29](#_Toc137992346)

[6.4.2 C28.2 - Identifier et savoir utiliser les outils d’assistance au référencement naturel afin d’analyser les performances SEO d’une application 29](#_Toc137992347)

[6.4.3 C28.3 - Implémenter les outils et techniques favorisant la SEO 29](#_Toc137992348)

[6.4.4 C28.4 - Implémenter les outils et techniques permettant la SMO (Social Media Optimization) 30](#_Toc137992349)

[6.4.5 C28.5 - Prévoir un plan de migration SEO dans le cadre de l’évolution d’une application 30](#_Toc137992350)

[6.5 C29 - Mettre en œuvre les outils et techniques permettant de monétiser une application 30](#_Toc137992351)

[6.5.1 C29.1 - Identifier les fournisseurs et solutions de paiement en ligne adaptés au contexte et à l’environnement d’une application 30](#_Toc137992352)

[6.5.2 C29.2 - Implémenter une solution de paiement en ligne adaptée au contexte et à l’environnement d’une application, en appliquant les principes standards de sécurité informatique 30](#_Toc137992353)

[6.5.3 C29.3 - Identifier les fournisseurs et solutions de monétisation (advertising)adaptés au contexte et à l’environnement d’une application 30](#_Toc137992354)

[6.5.4 C29.4 - Implémenter une solution de monétisation (advertising et marketing d’affiliation) adaptée au contexte et à l’environnement d’une application 30](#_Toc137992355)

# A1 - Participer à la gestion d'un projet d’application ou de site web

## C1- Participer à une réunion en identifiant la typologie du demandeur (client) pour comprendre sa demande et participer à l’élaboration d’une réponse adaptée à ses besoins

* Recueil des besoins
* [C1- Note de synthèse.docx](C1-%20Note%20de%20synthèse.docx)

Une note de synthèse présentant les préconisations techniques formulées pour le projet (écrit)

## C2- Conseiller le Maître d’Ouvrage en mettant à contribution ses connaissances et compétences techniques pour la réalisation du cahier des charges

* Mentionner les propositions faites au client lors du recueil des besoins
* Cahier des charges

## C3 - Participer à l’élaboration d’une planification réaliste en tenant compte de ses propres contraintes et compétences afin de garantir l’atteinte des objectifs fixés pour la réussite du projet

* Planning prévisionnel Poker planning)
* Proposition budgétaire

CE3 - Le planning prévisionnel et la proposition budgétaire formulés sont réalistes.

## C4 - Concevoir une ou plusieurs maquettes « wireframe » en utilisant un outil spécialiser afin de fournir une ébauche au demandeur

* Charte graphique
* Maquette en utilisant Figma

CE4 - La maquette « wireframe » réalisée correspond précisément à une des vues de l’application telle que souhaitée dans la note de cadrage fournie. Elle est conforme aux standards de la profession notamment en matière d’UX

## C5 - Mettre en œuvre un environnement de développement collaboratif adapté au projet d’application afin d’optimiser le temps de développement, le transfert de compétences auprès de ses pairs et la qualité logicielle

E5 - Une procédure de mise en œuvre d’un environnement complet de développement basé sur le SCM GIT et l’utilisation de la virtualisation sur le poste de travail local (écrit et oral)

### C5.1 - Mettre en œuvre et maîtriser l’utilisation d’un système de gestion de code source distribué (SCM) permettant de conserver l’historique des développements, d’organiser la collaboration des développeurs et d’appliquer un workflow standard pour la revue de code et le suivi des bugs.

* Mise en place du repo de l’organisation Github
* Git rules (branche Main, Dev, Fonctions)
* Respect des conventions de nommages des commits

### C5.2 - Mettre en œuvre un environnement de développement local ou distribué reproductible reposant sur l’utilisation de la virtualisation système ou de la virtualisation applicative

* Développer dans un environnement DOCKER
* Développer sur une VM Linux identique pour tous

### C5.3 Utiliser un environnement de développement intégré (IDE) adapté aux technologies sur lesquelles repose l’application et tirer parti de ses fonctionnalités

* IntelliJ

Préciser les plugins installés :

### C5.4 - Sélectionner et mettre en œuvre les outils et méthodes Agile de développement afin de collaborer efficacement avec les acteurs du projet d’application

* KANBAN ( Trello )
* Compte rendu (Sprints, back logs, user story, Daily meeting, planning poker)

E7 - Une trame type de compte rendu d’activité correspondant à la méthode projet retenue lors de l’évaluation E6 (écrit)

## C6 - Comprendre le cycle de développement et mettre en œuvre les principales méthodes de gestion de projet de développement afin de les appliquer au sein d’une équipe (XP, SCRUM, DSDM, ASD).

* Note de synthèse

CE6 - La note de synthèse produite démontre que le/la candidat/e comprend le cycle de développement et connaît les méthodes AGILE.

## C7 - Rédiger des compte-rendu d’activité destinés aux membres d’une équipe de projet afin de permettre le suivi de l’avancement du projet et la traçabilité des réalisations techniques

* ?

E7 - Une trame type de compte rendu d’activité correspondant à la méthode projet retenue lors de l’évaluation E6 (écrit)

# A2 - Concevoir et modéliser une application ou un site web



## C8 - Rédiger les spécifications techniques de besoin (STB) d’un projet d’application ou de site web à partir d’un cahier des charges afin de décrire de manière exhaustif les exigences à satisfaire en termes d’utilisation

* Document de Spécifications Fonctionnelles
* Document de spécifications Techniques

E8 - Constituer le dossier de conception traduisant les besoins du client exprimés dans le cahier des charges en spécifications techniques de besoin (STB) et modéliser les cas d’utilisation, les classes d’analyse et de conception, le schéma entité-association et le schéma des données. Décrire l’architecture logicielle choisie en vue du développement de l’application.

CE8 - Le dossier de conception est complet.

Les cas d’utilisation couvrent l’ensemble des exigences exprimées dans le cahier des charges fourni.

Les classes d’analyse et de conception sont définies et cohérentes avec le cas d’étude proposé.

L’architecture logicielle est conforme aux usages de la profession et adaptés au cas d’étude proposé.

Le dossier de spécifications est structuré et documenté en conformité avec la démarche choisie

## C9 - Modéliser l’application logicielle et ses données en utilisant une méthode standard afin d’abstraire la réalité et mieux comprendre le système à réaliser

### C9.1 - Utiliser le formalisme du langage de modélisation unifié UML et de UP afin de traduire les besoins de l’application



### C9.2 - Constituer le dossier de modélisation comprenant les diagrammes, les cas d’utilisation, les classes d’analyse et décrivant l’architecture logicielle n-tiers de l’application

* Diagramme des cas d'utilisation (Use Case Diagram)
* Diagrammes de séquence (Sequence Diagrams)
* Diagrammes d'activité (Activity Diagrams)
* Diagrammes de classes (Class Diagrams)
* Diagrammes d'états (State Diagrams)
* Diagrammes de déploiement (Deployment Diagrams)
* Description architecture n-tiers :

Voici comment vous pouvez décrire l'architecture n-tiers de votre application:

* **1. Introduction:** Commencez par une introduction générale de ce qu'est une architecture n-tiers et pourquoi elle est utilisée dans votre application.
* **2. Description de chaque couche:**
  + **Couche de Présentation :** Cette couche est celle avec laquelle l'utilisateur interagit directement. Elle comprend l'interface utilisateur et la gestion des entrées et des sorties. Pour décrire cette couche, indiquez quels langages de programmation ou quelles technologies vous utilisez pour le front-end, comme HTML, CSS, JavaScript, React, Angular, etc.
  + **Couche Logique métier (ou Logique d'application) :** Cette couche contient la logique métier de l'application. Elle fait le lien entre la couche de présentation et la couche de données. Pour décrire cette couche, mentionnez le langage de programmation (PHP, Python, Java, etc.), le framework utilisé si nécessaire, et comment la logique métier est gérée et traitée.
  + **Couche de Données :** C'est la couche qui interagit avec la base de données pour créer, lire, mettre à jour et supprimer des données. Pour décrire cette couche, mentionnez le type de base de données utilisée (SQL, NoSQL, etc.), le logiciel de gestion de base de données (MariaDB, MongoDB, etc.) et l'ORM/ODM utilisé (Doctrine, etc).
* **3. Diagramme d'architecture:** Incluez un diagramme qui montre comment les différentes couches interagissent entre elles. Le diagramme doit clairement montrer les couches distinctes et comment les données et les demandes d'information passent d'une couche à l'autre.
* **4. Justification de l'architecture:** Expliquez pourquoi vous avez choisi une architecture n-tiers pour votre application. Parlez des avantages de cette architecture, comment elle facilite la maintenance, le déploiement, et l'évolutivité de l'application.
* **5. Conclusion:** Résumez vos points principaux et expliquez comment l'architecture n-tiers soutient les objectifs globaux de l'application.

### C9.3 - Utiliser des patrons de conceptions afin d’appliquer des solutions logicielles standards

* MVC…

## C10 - Concevoir l’architecture des bases de données à l’aide d’un outil de modélisation afin de représenter la structure de la couche de persistance nécessaire au fonctionnement de l’application ou du site web

### C10.1 - Recenser les informations / données du domaine

* Dictionnaire des données

### C10.2 - Établir le schéma entité-association des données et définir le schéma de la base de données afin de permettre l’élaboration d’une base de données normalisée

* MCD
* MLD
* MPD

### C10.3 - Dans le cas d’une base de données existante, déduire le schéma entité-association des données de la base à faire évoluer

## C11 - À partir du dossier de spécifications fonctionnelles et techniques, déterminer l’architecture logicielle de l’application ou du site web

* Schéma de l’architecture logiciel ?

# A3 - Développer la partie front-end d’une application ou d’un site web en appliquant les bonnes pratiques d’UX, de sécurité informatique et d’éco-conception



## C12 - Développer la partie front-end d’une application web, hybride, mobile ou desktop en utilisant plusieurs langages de programmation et en appliquant les bonnes pratiques d’UX, de sécurité informatique et d’éco-conception

* Réaliser avec Next.js

Respect de la charte graphique

respect des specs fonctionnelles et techniques

eco conception / performance

Compatibilité avec tous les navigateurs

* Sécurité :

Certificat SSL valide

entêtes http de sécurité

CORS

CSP

E9 - Intégrer et développer intégralement la partie front-end en conformité avec le dossier de conception fourni, dans le respect de la charte graphique et des bonnes pratiques (UX, sécurité, écoconception)

CE9 - Le développement est globalement conforme aux exigences décrites dans les spécifications et repose sur un choix de technologies et frameworks front-end adapté Plus précisément :

Le développement implémente et couvre l’ensemble des cas d’utilisation décrit dans le dossier de conception

Le code source est valide et conforme aux référentiels des langages utilisés

Le développement propose une UX conforme aux usages actuels et aux bonnes pratiques

Le développement implémente les mécanismes et pratiques de sécurité standards et notamment : Certificat SSL valide, entêtes HTTP de sécurité, CORS et CSP

Le cas échéant, le développement repose sur des composants tiers à jour et sans vulnérabilité connue

Le développement est réalisé dans une démarche d’écoconception et tient compte des questions de performance

L’application ou le site web est compatible avec les plateformes et navigateurs actuels

### C12.1 - À partir d’une maquette et/ou d’une charte graphique, structurer et intégrer une interface utilisateur responsive pour le web

* Responsive

### C12.2 - À partir d’une maquette, structurer et intégrer une interface utilisateur mobile first ou desktop en utilisant une technologie hybride (web app / progressive web app)

* Interface mobile first, web app

### C12.4 - Sélectionner et mettre en œuvre un ou plusieurs langages de programmation front-end adaptés aux exigences et contraintes de l’application ou du site web

* Javascript

### C12.5 - Sélectionner et mettre en oeuvre un framework de développement front-end adapté aux exigences et contraintes de l’application ou du site web

* Next.js

### C12.6 - Développer des composants front-end graphiques et / ou fonctionnels en choisissant des structures de données adaptées et des algorithmes pertinents afin d’assurer la qualité logicielle

* Développement par composant
* SASS pour styliser

## C12.7 - Appliquer les bonnes pratiques UX (expérience utilisateur)

* **Recherche facile :** Permettez aux utilisateurs de rechercher facilement des gîtes en fonction de divers critères tels que la localisation, le prix, la taille, les dates de disponibilité, etc. Le processus de recherche doit être intuitif et rapide.
* **Informations claires et détaillées :** Chaque gîte doit avoir une page de détails fournissant des informations complètes et précises. Cela peut inclure des photos, une description du gîte, des informations sur la région environnante, les prix, les disponibilités, les règles de la maison, etc.
* **Processus de réservation facile :** Le processus de réservation doit être simple et sans friction. Il doit être clair pour l'utilisateur à quel stade du processus de réservation il se trouve et quels sont les prochaines étapes.
* **Avis et évaluations :** Permettez aux utilisateurs précédents de laisser des avis et des évaluations. Cela donne aux futurs utilisateurs plus de confiance dans la réservation d'un gîte.
* **Assistance utilisateur :** Offrez un support facilement accessible pour aider les utilisateurs qui rencontrent des problèmes ou ont des questions. Cela pourrait prendre la forme d'une FAQ détaillée, d'un chat en direct, d'une assistance par e-mail, etc.
* **Responsive Design :** Assurez-vous que votre site est optimisé pour tous les types d'appareils, y compris les ordinateurs de bureau, les tablettes et les smartphones. Les utilisateurs doivent pouvoir effectuer toutes les tâches, y compris la réservation d'un gîte, facilement sur n'importe quel appareil.
* **Simplicité et clarté :** Votre site doit être propre et sans encombrement. Chaque page doit avoir un objectif clair et tous les éléments inutiles doivent être éliminés.
* **Accessibilité** : Assurez-vous que votre site est accessible à tous les utilisateurs, y compris ceux qui ont des besoins spéciaux en matière d'accessibilité.

## C12.8 - Appliquer les recommandations de sécurité pour le développement front-end afin de s’inscrire dans une démarche de « sécurité en profondeur »

Une gestion des dépendances est mise en œuvre

* **Validation côté client** : Bien qu'il ne soit pas sûr de compter uniquement sur la validation côté client (car elle peut être facilement contournée), elle peut néanmoins servir de première ligne de défense contre des données d'entrée malveillantes. Elle peut également améliorer l'expérience utilisateur en fournissant une rétroaction instantanée.
* **Échappement de sortie :** Lors de l'affichage de données d'entrée contrôlées par l'utilisateur (par exemple, dans un commentaire de blog ou un champ de formulaire), assurez-vous qu'elles sont correctement échappées pour prévenir les attaques de type Cross-Site Scripting (XSS).
* **Content Security Policy (CSP) :** Utilisez CSP pour limiter les sources à partir desquelles certains types de contenu peuvent être chargés, afin de prévenir les attaques XSS.
* **Utiliser HTTPS :** HTTPS crypte les données qui circulent entre le navigateur de l'utilisateur et votre serveur, protégeant contre les attaques de type "homme du milieu". Assurez-vous que votre site est accessible uniquement via HTTPS.
* **Contrôle d'accès côté client :** Ne vous fiez jamais uniquement au contrôle d'accès côté client. Même si vous masquez un bouton ou un lien côté client, un attaquant pourrait toujours accéder à la fonctionnalité correspondante en modifiant directement la requête au serveur.
* **Gestion sécurisée des cookies :** Les cookies qui contiennent des informations sensibles devraient être marqués comme "Secure" (transmis uniquement sur HTTPS) et "HttpOnly" (non accessible via JavaScript), pour protéger contre les attaques de type interception et XSS.
* **Mise à jour des bibliothèques/frameworks :** Les bibliothèques et frameworks JavaScript obsolètes peuvent contenir des failles de sécurité connues. Il est donc important de les maintenir à jour.
* **Utilisation de CORS :** Si votre application doit faire des requêtes à des domaines différents du sien (par exemple, une API), utilisez la politique de partage de ressources cross-origin (CORS) pour contrôler quelles origines sont autorisées.
* **Réduire la surface d'attaque :** Plus votre application est complexe, plus elle est susceptible d'avoir des vulnérabilités. Essayez de garder votre code aussi simple et minimaliste que possible.
* **Sécurité des dépendances :** Les dépendances obsolètes ou malveillantes sont une autre source commune de vulnérabilités. Utilisez des outils pour surveiller la sécurité de vos dépendances.

## C13 - Tester la partie front-end de l’application à plusieurs niveaux en utilisant des méthodes de test standards afin de garantir sa conformité vis-à-vis des spécifications et assurer la non-régression des composants implémentés

E10 - Élaborer et mettre en oeuvre un plan de tests cohérent avec le front-end précédemment développé (E9).

CE10 - Le plan de test est cohérent au regard des exigences décrites dans les spécifications.

Les tests présentent une couverture du code source au moins égale à 50%

E11 - Mettre en oeuvre l’industrialisation du développement front-end en automatisant l’exécution des tests définis (E10) et des outils d’assurance qualité adaptés, la gestion des dépendances et l’exécution d’une chaine de build orientée performance.

### C13.1 - Élaborer un plan de test logiciel exhaustif

### C13.2 - Réaliser des tests de conformité du langage de balisage structurant le contenu des interfaces web, mobile hybride et desktop

* **Validation HTML :** Utilisez des outils de validation HTML, comme le validateur HTML du W3C, pour vérifier la syntaxe de votre code HTML. Cela vous aidera à repérer les erreurs de balisage, comme les balises mal fermées, les attributs manquants, etc.
* **Accessibilité :** Assurez-vous que votre site est accessible en utilisant des outils comme aXe ou Lighthouse de Google. Ces outils peuvent tester si votre site est conforme aux directives d'accessibilité du web (WCAG), par exemple si les images ont des textes alternatifs, si le contraste de couleur est suffisant, etc.
* **Performance :** Utilisez des outils comme Lighthouse ou PageSpeed Insights de Google pour tester la performance de votre site. Ces outils peuvent vérifier si votre site est optimisé pour une charge rapide, par exemple si les images sont optimisées, si le JavaScript et le CSS sont minifiés, etc.
* **Responsive Design :** Assurez-vous que votre site est bien affiché sur tous les types d'appareils (desktop, tablette, mobile) et toutes les tailles d'écran. Vous pouvez utiliser les outils de développement de votre navigateur pour simuler différents appareils et tailles d'écran, ou utiliser un outil en ligne comme BrowserStack.
* **Compatibilité des navigateurs :** Testez votre site sur différents navigateurs (Chrome, Firefox, Safari, Edge, etc.) et différentes versions de navigateurs pour vous assurer qu'il fonctionne correctement sur tous. Encore une fois, des outils comme BrowserStack peuvent être utiles pour cela.
* **Sécurité :** Enfin, utilisez des outils comme Mozilla Observatory pour vérifier si votre site est sécurisé, par exemple si le protocole HTTPS est utilisé, si le Content Security Policy (CSP) est correctement défini, etc.

### C13.3 - Réaliser des tests unitaires pour les composants fonctionnels

* Utiliser Jest, un framework de test javascript

### C13.4 - Réaliser des tests fonctionnels, utilisateurs et de compatibilité avec les différentes plateformes

* **Tests fonctionnels :** Ils ont pour but de vérifier que chaque fonctionnalité de votre site fonctionne comme prévu. Cela peut inclure des choses comme remplir et soumettre des formulaires, naviguer à travers le site, effectuer des recherches, etc. Vous pouvez réaliser ces tests manuellement, ou les automatiser en utilisant des outils comme **Selenium, Jest ou Cypress.**
* **Tests utilisateurs :** Ils impliquent de véritables utilisateurs interagissant avec votre site. L'objectif est de voir si les utilisateurs peuvent accomplir leurs tâches facilement et intuitivement. Cela peut impliquer des tests d'utilisabilité, où vous observez les utilisateurs en train d'utiliser votre site, ou des tests A/B, où vous comparez différentes versions de votre site pour voir laquelle performe le mieux.
* **Tests de compatibilité :** Ils vérifient que votre site fonctionne correctement sur différentes plateformes et navigateurs. Cela comprend différents systèmes d'exploitation (Windows, macOS, Linux, Android, iOS), différents navigateurs (Chrome, Firefox, Safari, Edge), différentes tailles d'écran et différents types de connexion internet. Des outils **comme BrowserStack ou LambdaTest** peuvent être utiles pour cela.

## C14 - Industrialiser le développement de la partie front-end de l’application et automatiser les processus d’assurance qualité

CE11 - Le processus d’industrialisation est fonctionnel

Le choix des outils de qualité mis en oeuvre est cohérent au regard des exigences décrites dans les spécifications

Une gestion des dépendances est mise en oeuvre

La chaine de build permet effectivement d’améliorer les performances du front-end de l’application ou du site web réalisé

### C14.1 - Automatiser l’exécution d’un vérificateur / correcteur de syntaxe pour chaque langage de programmation constituant le code source de la partie front-end, si nécessaire.

* Utilisation d’outil comme ESLint ou TSLint

### C14.2 - Utiliser un gestionnaire de paquets pour automatiser la gestion des dépendances (composants tiers)

* Utilisation de npm ou yarn
* **Scripts :** Le fichier package.json peut aussi définir des scripts, qui sont des commandes personnalisées que vous pouvez exécuter avec npm run [nom\_du\_script] ou yarn run [nom\_du\_script]. Par exemple, vous pouvez définir un script pour démarrer votre serveur de développement, un autre pour créer une build de production, un autre pour exécuter vos tests, etc.
* **Sécurité :** npm et yarn ont tous deux des fonctionnalités pour aider à gérer la sécurité de vos dépendances. Par exemple, npm audit ou yarn audit peut être utilisé pour vérifier les vulnérabilités connues dans vos dépendances, et npm update ou yarn upgrade peut être utilisé pour les mettre à jour et résoudre ces vulnérabilités.

### C14.3 - Utiliser un groupeur de modules afin de configurer et d’exécuter la chaine de build et ainsi optimiser les performances de l’interface

* Next.js utilise webpack par défaut

vous pouvez l'utiliser pour optimiser les performances de votre interface :

* **Transpilation :** Webpack utilise Babel pour transpiler votre code ES6 en code ES5 qui peut être exécuté dans tous les navigateurs. Vous n'avez pas besoin de configurer cela manuellement, car Next.js le fait pour vous.
* **Minification :** Webpack peut minifier votre code JavaScript, CSS et HTML pour réduire la taille des fichiers et améliorer les temps de chargement. Encore une fois, Next.js le fait automatiquement pour vous en mode production.
* **Diviser le code :** Webpack peut diviser votre code en plusieurs "chunks" qui peuvent être chargés en parallèle ou à la demande. Next.js utilise cette fonctionnalité pour implémenter le "splitting de code au niveau des pages", où chaque page de votre application est un chunk séparé qui est chargé uniquement quand l'utilisateur navigue vers cette page.
* **Support pour CSS et autres actifs :** Webpack peut également traiter les fichiers CSS, les images, les polices, et d'autres types de fichiers. Next.js le configure pour vous, vous pouvez donc simplement importer vos fichiers CSS (ou SCSS, ou modules CSS) directement dans vos composants React, et ils seront automatiquement ajoutés au bundle.
* **Hot Module Replacement (HMR) : Cette** fonctionnalité permet de mettre à jour le code en direct sans avoir à rafraîchir la page entière. Next.js l'active automatiquement en mode développement.

En bref, grâce à Next.js et Webpack, vous pouvez bénéficier d'une chaîne de build hautement optimisée sans avoir à faire beaucoup de configuration vous-même. Cependant, si vous avez besoin d'une configuration Webpack plus personnalisée, vous pouvez le faire en créant un fichier next.config.js dans le répertoire racine de votre projet.

# A4 - Développer la partie back-end d’une application ou d’un site web en appliquant les bonnes pratiques de sécurité informatique, d’éco-conception et DevOps

E12 - Développer et mettre en oeuvre la couche de persistance des données

E13 - Développer intégralement la partie back-end en conformité avec le dossier de conception fourni et intégrer la partie front-end, dans le respect des bonnes pratiques (sécurité, écoconception)

CE13 - Le développement est globalement conforme aux exigences décrites dans les spécifications et repose sur un choix de technologies et frameworks back-end adapté Plus précisément :

Le développement implémente et couvre l’ensemble des cas d’utilisation décrit dans le dossier de conception

Le code source est valide et conforme aux référentiels du/des langage/s utilisés

Le développement repose sur une organisation du code conforme aux usages actuels et aux bonnes pratiques

Le développement implémente les mécanismes et pratiques de sécurité standards et notamment le filtrage des données entrantes et sortantes, l’authentification forte, la journalisation et le monitoring, la validation des configurations et le contrôle d’accès

Le cas échéant, le développement repose sur des composants tiers à jour et sans vulnérabilité connue

Le développement est réalisé dans une démarche d’écoconception et tient compte des questions de performance

L’application ou le site web est compatible avec les plateformes et versions des langages actuelles



## C15 - Développer la couche de persistance des données en appliquant les bonnes pratiques de sécurité informatique et de performance

* Sécurité :
* **Prévention des injections SQL** : Doctrine utilise des requêtes paramétrées, ce qui peut aider à prévenir les injections SQL. Assurez-vous de toujours utiliser cette fonctionnalité et d'éviter d'interpoler les variables directement dans vos requêtes SQL.
* **Validation des données :** Symfony fournit des composants de validation que vous pouvez utiliser pour assurer que les données entrantes sont valides avant de les insérer dans la base de données.
* **Gestion des erreurs :** Assurez-vous de gérer correctement les erreurs de base de données et d'éviter d'exposer les détails de l'erreur à l'utilisateur, car cela pourrait révéler des informations sensibles.
* Performance :
* **Optimisation des requêtes :** Utilisez les outils de profilage et d'analyse de Doctrine pour comprendre comment vos requêtes sont exécutées et comment vous pouvez les optimiser.
* **DQL au lieu de SQL brut :** DQL est le langage de requête de Doctrine, qui est plus orienté objet et plus sûr que le SQL brut. Utiliser DQL vous permet également de profiter du cache de requêtes de Doctrine.
* **Cache :** Doctrine offre des options de mise en cache pour les résultats de requête et les métadonnées. Utilisez-les pour améliorer les performances de lecture.
* **Lazy Loading et Eager Loading :** Doctrine utilise le "lazy loading" par défaut, ce qui signifie qu'il ne récupère les données associées que lorsque vous en avez réellement besoin. Cela peut être bon pour la performance, mais si vous savez que vous aurez besoin des données associées, vous pouvez utiliser le "eager loading" pour les récupérer en une seule requête et éviter le problème N+1.
* **Pagination :** Si vous devez récupérer un grand nombre de lignes, utilisez la pagination pour diviser les résultats en plusieurs pages. Symfony et Doctrine fournissent des outils pour faciliter la mise en œuvre de la pagination.Notes

### C15.1 - Mettre en œuvre et administrer des bases de données relationnelles de manière sécurisée

* Gestion des privilèges
* Authentification forte
* Chiffrement
* Sauvegardes (+ tester la restauration de sauvegardes)
* Mises à jour
* Audit et surveillance (logs)
* Séparation des environnements (tests, dev, prod)

### C15.2 - Exploiter et interroger des bases de données relationnelles depuis le back-end de l’application

### C15.3 - Mettre en œuvre et administrer des bases de données NoSQL de manière sécurisée

### C15.4 - Exploiter et interroger des bases de données NoSQL

### C15.5 - Faire évoluer les bases de données

* Mettre en place un environnement de compte rendu (créer des modèles)
* Préparer la prochaine réunion et le partager
* Faire un planning organisationnel (instaurer une routine daily meeting, sprints…)

## C16 - Développer la partie back-end d’une application en utilisant plusieurs langages de programmation et en appliquant les bonnes pratiques de sécurité informatique et d’éco-conception

### C16.1 - Sélectionner et mettre en œuvre un ou plusieurs langages de programmation back-end adaptés aux exigences et contraintes de l’application ou du site web

* PHP

### C16.2 - Sélectionner et mettre en œuvre un framework de développement back-end adaptés aux exigences et contraintes de l’application ou du site web

* Symfony

### C16.3 - À partir du référentiel de modélisation, développer des composants back-end fonctionnels en choisissant des structures de données adaptées et des algorithmes pertinents afin d’assurer la qualité logicielle

### C16.4 - Appliquer les recommandations de sécurité pour le développement back-end afin de s’inscrire dans une démarche de « sécurité en profondeur »

### C16.5 - Appliquer les bonnes pratiques d’écoconception afin de minimiser l’impact écologique de la partie back-end de l’application

## C17 - Consommer une API de manière sécurisée en s’appuyant sur des mécanismes d’identification et d’authentification robustes afin d’intégrer des sources de données et des services tiers

* API booking, Airbnb

## C18 - Développer une API permettant à des services tiers de consommer des données produites par l’application au travers de formats ouverts

* API de réservation

### C18.2 - Sélectionner et mettre en oeuvre un format d’échange de données adapté aux caractéristiques de l’application et à son environnement (consommateurs d’API)

### C18.3 - Sécuriser l’accès à l’API en implémentant des mécanismes d’identification et d’authentification robustes

### C18.4 - Documenter et publier une API

## C19 - Tester la partie back-end de l’application à plusieurs niveaux en utilisant des méthodes de test standards afin de garantir sa conformité vis-à-vis des spécifications et assurer la non-régression des composants implémentés

E14 - Élaborer et mettre en oeuvre un plan de tests complet et cohérent avec le back-end et la couche de persistance des données précédemment développés (E12 et E13).

CE14 - Le plan de test est cohérent au regard des exigences décrites dans les spécifications.

Les tests présentent une couverture du code source au moins égale à 50%

### C19.1 - Élaborer un plan de tests logiciels exhaustif pour le back-end

### C19.2 - Réaliser des tests unitaires sur les composants du back-end

### C19.3 - Réaliser des tests fonctionnels sur les composants du back-end

## C20 - Industrialiser le développement de la partie back-end de l’application ou du site web et automatiser les processus d’assurance qualité

E15 - Mettre en oeuvre l’industrialisation du développement back-end en automatisant l’exécution des tests définis (E14) et des outils d’assurance qualité adaptés, la gestion des dépendances et l’exécution d’une chaine de build orientée performance et sécurité.

CE15 - Le processus d’industrialisation est fonctionnel

Le choix des outils de qualité mis en oeuvre est cohérent au regard des exigences décrites dans les spécifications

Une gestion des dépendances est mise en oeuvre

La chaine de build permet effectivement d’améliorer les performances et la sécurité du backend de l’application ou du site web réalisé

### C20.1 - Mettre en oeuvre un processus d’intégration continue de manière à minimiser les erreurs humaines et de rationaliser le cycle de développement de la partie back-end de l’application.

### C20.2 - Utiliser un gestionnaire de paquets pour automatiser la gestion des dépendances (composants tiers) de la partie back-end de l’application

### C20.3 - Automatiser la construction de l’application en configurant la chaînes de build, l’exécution des tests, l’exécution des outils d’assurance qualité et l’analyse statique de sécurité.

## C21 - Faire évoluer ses connaissance et compétences en développement back-end

E16 - Formaliser par écrit une méthodologie de veille technologique relative au composants et technologies constitutifs de l’application (back-end et front-end)

CE16 - La méthodologie de veille technologique proposée est cohérente et adaptée aux exigences et au contexte de l’application ou du site web

### C21.1 - Formaliser et mettre en œuvre une démarche de veille technologique

### C21.2 - Utiliser un outil de knowledge management afin de capitaliser la connaissance et faciliter sa transmission

# A5 - Déployer et assurer le maintien en production d’une application ou d’un site web en appliquant les bonnes pratiques de sécurité informatique et DevOps (Phase livraison)

Détails de la mise en situation professionnelle simulée en équipe constituée de 2 à 3 personnes : Une application web existante est fournie. Elle se compose principalement :

D’un front-end

D’un back-end REST

D’une base de données relationnelle

D’une base de données NoSQL

D’au moins un micro-service serverless

D’appels API externes et de webhooks qui interviennent en back-end de la version mobile de l’application.

Cette application comporte volontairement des bugs techniques, des failles de sécurité et repose sur certains composants obsolètes. À partir ce cet application, le/la candidat/e doit réaliser :



## C22 - Préparer, mettre en œuvre et automatiser la mise en production d’une application dans différents types d’environnements techniques d’exploitation

E17 - La définition de l’architecture technique et le dimensionnement d’un environnement d’hébergement de type cloud adaptés à l’application, formalisés dans un livrable rédigé et argumenté.

CE17- L’architecture technique d’hébergement proposée est adaptée à l’application.

Le principe d'élasticité a été prise en compte dans l’architecture proposée

Le dimensionnement et les coûts sont en cohérence avec les besoins de l’application.

E18 - La mise en oeuvre sécurisée de l’environnement de production décrit lors de la 1ère itération et l’administration des services adaptés à l’aide d’un gestionnaire de configuration. La mise en oeuvre de l’environnement de qualification (pré-production) correspondant. L’utilisation d’un système de conteneurs et d’orchestration pour l’hébergement de la partie front-end.

CE18 - Les services sont installés, configurés et fonctionnels.

Un gestionnaire de configuration a été utilisé.

L’environnement cloud standard est fonctionnel.

L’environnement cloud conteneurisé est fonctionnel.

Les bonnes pratiques de sécurité en termes d’architecture et de configuration des services ont été appliquées (notamment mise en œuvre d’un hôte bastion).

### C22.1 - Identifier les différents types de plateformes d’hébergement et sélectionner une plateforme d’hébergement pertinente et adaptée aux exigences techniques de l’application(dimensionnement) en prenant en compte l’optimisation des coûts

### C22.2 - Définir l’architecture, mettre en oeuvre et administrer des services d’hébergement dans un environnement dédié, cloud ou conteneurisé en appliquant les bonnes pratiques de sécurité

### C22.3 - Mettre en œuvre et exploiter un système d’orchestration de conteneurs

### C22.4 - Mettre en œuvre un environnement de qualification externe

### C22.5 - Élaborer un diagramme de déploiement et mettre en œuvre un système de déploiement automatisé d’une application pour différents types d’hébergement, avec ou sans processus d’intégration continue, en respectant les bonnes pratiques DevOps

E20 - La mise en oeuvre d’un système de déploiement automatisé pour les environnements précédemment créés.

CE20 - Un diagramme de déploiement a été formalisé.

Le choix du système de déploiement est adapté à l’application.

Le déploiement sur l’environnement de qualification a été prévu et effectué.

Le système de déploiement est fonctionnel sur les environnements de qualification et de production et respecte les recommandations DevOps

### C22.6 - Automatiser la mise en œuvre des architectures à l’aide d’un gestionnaire de configuration ou d’un système d’ « Infrastructure as Code »

### C22.7 - Gérer des noms de domaines, administrer les serveurs de noms de domaine, et mettre en œuvre des certificats de sécurité

E19 - La réservation d’un nom de domaine, la configuration des serveurs de noms et des services correspondants, l’installation et la vérification des certificats de sécurité pour les deux environnements précédemment créés.

CE19 - Un nom de domaine a été réservé. Les déclarations administratives ont été effectuées.

Les serveurs de noms correspondants sont configurés et fonctionnels. Les bonnes pratiques de sécurité ont été appliquées.

Les certificats de sécurités ont été créés. Ils sont installés, configurés et fonctionnels pour les différents services.

## C23 - Sécuriser et superviser l’application et l’environnement technique d’exploitation afin d’en assurer le maintien en production

### C23.1 - Sécuriser les services et les réseaux d’une architecture d’hébergement

### C23.2 - Mettre en oeuvre les liaisons sécurisées entre les différents services et les composants de l’application

### C23.3 - Mettre en oeuvre des systèmes de sauvegarde et de redondance de l’infrastructure d’hébergement

E22 -La mise en oeuvre des outils de sauvegarde et d’une stratégie de tolérance de panne de l’infrastructure d’hébergement (redondance) pour les environnements précédemment créés.

CE22 - Les outils de sauvegarde sont fonctionnels et efficaces.

La stratégie de tolérance de panne est pertinente et efficace.

### C23.4 - Identifier les métriques et indicateurs de performance de l’application et des services d’hébergement correspondants ; définir un « budget » performance

### C23.5 - Réaliser un audit de performance d’une application et des services d’hébergement correspondants

### C23.6 - Rédiger un rapport d’audit de performance de l’application et des services d’hébergement correspondants; émettre des préconisations

### C23.7 - Préparer et déployer des mesures correctives pour l’amélioration des performances de l’application et de son environnement

### C23.8 - Mettre en œuvre un système de journalisation et implémenter une stratégie d’audit des journaux de l’application

E21 - La mise en oeuvre de la journalisation des services et des outils d’audit correspondants pour les deux environnements précédemment créés.

CE21 - La journalisation des services a été implémentée.

Les outils d’audit sont fonctionnels et permettent une analyse efficace des erreurs.

### C23.9 - Mettre en œuvre un système de supervision d’une application et de services d’hébergement, définir des sondes et des alertes (état des services et sécurité)

E23 - La mise en oeuvre des outils de supervision et la configuration des alertes adaptées au contexte de l’application pour les deux environnements précédemment créés.

CE23 - Les outils de supervision sont fonctionnels.

Les sondes et alertes définies sont pertinentes.

### C23.10 - Détecter, diagnostiquer et analyser l’origine des bugs et problèmes techniques rencontrés

E24 - La détection des bugs de l’application et proposer des mesures correctives.

CE24 - Les bugs ont été identifiés

Les mesures correctives proposées sont efficaces

E25 - La détection des failles de sécurité, et proposer des mesures correctives.

### C23.11 - Préparer et déployer des mesures correctives pour les erreurs ou les failles de sécurité identifiées dans des délais adaptés

CE25 - Les failles de sécurité ont été identifiées

Les mesures correctives proposées sont efficaces

## C24 - Produire la documentation technique de l’application et alimenter une base de connaissances afin de capitaliser l’information et d’effectuer un transfert de compétences auprès de ses pairs

E26 - La génération de la documentation à partir code source et mettre à jour le journal des évolutions (CHANGELOG) avec les informations relatives aux itérations 8 et 9

CE26 - La documentation du code source a été générée

Les mises à jour relatives aux évaluations 8 et 9 ont été portées dans le journal des évolutions.

### C24.1 - Commenter son code source en respectant les standards et bonnes pratiques en la matière afin de permettre la génération automatique de documentation

### C24.2 - Mettre en oeuvre la génération automatique de documentation du code source afin de l’intégrer dans un processus d’intégration continue

### C24.3 - Rédiger la documentation technique d’un projet afin d’effectuer un transfert de compétences auprès de ses paris

### C24.4 - Rédiger la documentation d’une API afin de faciliter sa consommation par des tiers

# A6 - Mettre en oeuvre des solutions techniques répondant aux besoins contextuels d’une application ou d’un site web (lois, normes et règlements, accessibilité, outils marketing et e-commerce) et piloter les performances.



## C25 - Mettre en oeuvre les outils et techniques permettant de respecter les aspects réglementaires au regard des données informatiques traitées et stockées

E27 - La mise en conformité règlementaire RGPD de l’application pour le traitement des données à caractère personnelles (sur le plan technique exclusivement).

CE27 :

Un outil de recueil du consentement de l’utilisateur vis-à-vis des cookies a été implémenté.

Une vue « politique de confidentialité et traitement des données à caractère personnel » a été prévue.

Un formulaire de demande d’accès aux données personnelles et le traitement correspondant ont été implémentés. À défaut, les informations nécessaires et une adresse de contact DPO ont été publiés.

Les traitements collectant des données personnelles ont été modifiés pour mettre en place un processus de double opt-in.

### C25.1 - Identifier les données et traitements informatiques concernés par les lois, normes et règlements (CNIL, RGPD,…)

### C25.2 - Implémenter les moyens techniques et technologiques permettant d’assurer le respect des lois, normes et règlements (CNIL, RGPD,…) applicables aux données traitées par une application et stockées dans une base de données.

### C25.3 - Connaître et respecter les lois et règlements en matière de propriété intellectuelle au regard des composants logiciels et codes sources informatiques

## C26 - Mettre en oeuvre les outils et techniques permettant de respecter les aspects réglementaires au regard de l’accessibilité numérique

E28 - La mise en conformité règlementaire RGAA version 4 de l’application pour l’accessibilité numérique (sur le plan technique exclusivement)..

CE28 :

L’application est conforme à au moins 50% des critères de la méthodologie de test concernant les obligations d’accessibilité numérique en vigueur actuellement.

### C26.1 - Identifier, dans une application, les éléments de présentation (UX), traitements et contenus informationnels concernés par les obligations d’accessibilité numérique.

### C26.2 - Implémenter les moyens techniques et technologiques permettant d’assurer le respect des obligations d’accessibilité numérique (RGAA - Référentiel général d'amélioration de l ’accessibilité).

## C27 - Mettre en oeuvre les outils et techniques permettant de mesurer et de suivre les performances marketing d’une application

E29 - Le choix pertinent et l’intégration d’outils de mesure d’audience et de performance marketing adaptés au contexte de l’application fournie

CE29 :

Le choix des outils de mesure d’audience et de performance marketing est pertinent.

L’intégration des outils de mesure de performance marketing est fonctionnelle.

### C27.1 - Identifier les outils et techniques adaptés et pertinents au regard du contexte d’une application afin de mesurer et de suivre ses performances marketing

### C27.2 - Implémenter les outils et techniques permettant de mesurer et de suivre les performances marketing d’une application (analytics)

## C28 - Mettre en oeuvre les outils et techniques permettant d’améliorer les performances SEO (Search Engine Optimization / référencement naturel) d’une application logicielle

E30 - L’optimisation technique du code source du front-end de l’application fournie dans une démarche orientée SEO et SMO et l’intégration d’un outil de mesure des performances techniques.

CE30 :

L’application est conforme à au moins 70% des critères d’optimisation technique SEO en vigueur actuellement.

### C28.1 - Identifier les critères techniques d’une application ayant un impact sur le référencement naturel : les facteurs bloquants et les facteurs favorisants

### C28.2 - Identifier et savoir utiliser les outils d’assistance au référencement naturel afin d’analyser les performances SEO d’une application

### C28.3 - Implémenter les outils et techniques favorisant la SEO

### C28.4 - Implémenter les outils et techniques permettant la SMO (Social Media Optimization)

### C28.5 - Prévoir un plan de migration SEO dans le cadre de l’évolution d’une application

## C29 - Mettre en œuvre les outils et techniques permettant de monétiser une application

E31 - L’intégration d’un système de paiement en ligne (fournisseur imposé, en mode test) basé sur une API

CE31 :

L’intégration du système de paiement est fonctionnelle.

L’intégration du système de paiement respecte les recommandations en matière de sécurité.

E31 - Un proposition de un système de monétisation adapté au contexte de l’application. Formaliser cette proposition par un livrable écrit argumenté.

CE32 :

Le système de monétisation proposé est adapté au contexte de l’application et pertinent au regard de la cible marketing.

### C29.1 - Identifier les fournisseurs et solutions de paiement en ligne adaptés au contexte et à l’environnement d’une application

### C29.2 - Implémenter une solution de paiement en ligne adaptée au contexte et à l’environnement d’une application, en appliquant les principes standards de sécurité informatique

### C29.3 - Identifier les fournisseurs et solutions de monétisation (advertising)adaptés au contexte et à l’environnement d’une application

### C29.4 - Implémenter une solution de monétisation (advertising et marketing d’affiliation) adaptée au contexte et à l’environnement d’une application