

TITRE EXPERTISE INFORMATIQUE ET SYSTEME D'INFORMATION

BLOC E7.4 – Concevoir et développer des solutions applicatives métier et spécifiques (mobiles, embarquées et ERP)

Cahier des Charges de la MSPR « Conception d'une solution applicative en adéquation avec l'environnement technique étudié »

COMPÉTENCES ÉVALUÉES :

- Collecter les besoins métiers des utilisateurs en menant des interviews auprès d'eux pour comprendre leurs activités et leurs contraintes métier afin d'étudier les opportunités et la faisabilité technologique d'une solution applicative spécifique ou métier.
- Concevoir une architecture applicative selon la complexité du système d'information existant de type architecture distribuée, ou micro-service évolutive et tolérante aux pannes.
- Développer une application adéquate selon la stratégie applicative de l'environnement en utilisant un langage de programmation approprié dans le respect du cahier des charges établi afin de répondre aux besoins utilisateurs/directions métiers.
- Développer une solution applicative intégrée en utilisant le paramétrage et le langage de programmation spécifique de l'éditeur dans le respect du cahier des charges établi afin de répondre aux besoins utilisateurs/directions métiers.
- Effectuer les tests de la solution applicative paramétrée ou développée pour identifier les erreurs et dysfonctionnements et établir les plans de correction/d'amélioration avant sa mise en production.
- Appliquer l'intégration continue dans le cadre du développement d'une application en utilisant un outil d'intégration continue afin de vérifier la conformité de la solution et les besoins utilisateurs.
- Vérifier la conformité entre la solution développée ou paramétrée et les fonctionnalités attendues à partir des retours des directions métiers afin de rédiger la documentation et les référentiels orientés utilisateurs.
- Conduire le changement auprès des métiers lors du déploiement d'une solution applicative ou intégrée en mettant en place une démarche de participation, de communication et de formation pour accompagner les utilisateurs à l'intégration du nouvel outil dans leurs habitudes de travail.

PHASE 1 : PRÉPARATION DE CETTE MISE EN SITUATION PROFESSIONNELLE RECONSTITUÉE

Durée de préparation : 21 heures

Mise en œuvre : Travail d'équipe constituée de 4 apprenants-candidats (5 maximum si groupe impair)

Résultat attendu – Dossier à produire par équipe :

- Réalisation d'une solution applicative fonctionnelle sous la forme d'API REST.
- Travailler selon les méthodes Agiles et pouvoir l'expliquer.

PHASE 2 : PRÉSENTATION ORALE COLLECTIVE + ENTRETIEN COLLECTIF

Durée totale par groupe : 30 mn se décomposant comme suit :

- 10 mn de soutenance orale par l'équipe.
- 20 mn d'entretien collectif avec le jury (questionnement complémentaire), en français.
- Objectif : mettre en avant et démontrer que les compétences visées par ce bloc sont bien acquises

Jury d'évaluation : 2 personnes (binôme d'évaluateurs) par jury – Ces évaluateurs ne sont pas intervenus durant la période de formation et ne connaissent pas les apprenants à évaluer.

I - PRÉSENTATION DE L'ENTREPRISE ET DE SON ACTIVITÉE

- Préambule : L'entreprise choisie pour cette MSPR est fictive, les prénoms sont fictifs, toute ressemblance à un cas réel serait purement fortuite.



La société PayeTonKawa est spécialisée dans l'import de café en France.

Depuis 2018, la société vend du café provenant du monde entier aux particuliers via son site internet public payetonkawa.fr.

Cette année elle souhaite se lancer dans la vente de café aux professionnels de la restauration via un réseau de distributeurs dédiés.

II– DESCRIPTION DU SYSTEME D'INFORMATION CLIENT

La société dispose actuellement d'un ERP contenant les informations sur les commandes, les produits et stocks. Elle dispose aussi d'un outil de CRM contenant la liste de ses clients.



La société dispose actuellement d'une équipe de 2 développeurs et 1 informaticien SysOps qui déploient et maintiennent la solution existante sans usine logicielle.

III– CONTEXTE DU BESOIN – CAHIER DES CHARGES

3.1 Développements :

Evolution de l'infrastructure

Le but de cette demande est de faire évoluer l'infrastructure pour permettre d'ouvrir l'accès aux données aux distributeurs tout en garantissant la sécurité des accès données et rendre le service plus résilient aux pannes potentielles.

Cela devra donc passer par la mise en place d'une plateforme qui permettra de mettre à disposition des APIs REST exposant les données aux consommateurs (webshop / revendeurs) en limitant leurs accès aux seuls données utiles à leur besoin.

Application mobile

La mise à disposition d'une application mobile permettra aux revendeurs de pouvoir présenter facilement les propriétés des produits et, via de la réalité augmentée, pouvoir présenter le produit aux futurs clients.

Cette application pourra récupérer les informations des produits depuis les APIs mises à dispositions précédemment. Affichage d'une liste de produits et fiches produits affichant les détails des produits.

La réalité augmentée devra permettre de visualiser un produit (ex : une machine à café) dans son environnement final. Cela permettra aux clients du revendeur de se projeter et voir ainsi comment la machine à café se positionnera dans son espace.

Réorganisation des équipes

La mise en place d'une plateforme d'intégration continue et de déploiement continue permettra d'automatiser les tests et les déploiements de la solution afin que permette la mise en place d'une organisation d'équipe séparant les rôles de développement et de déploiement (2 équipes distinctes).

Un process Agile permettra de gérer, via un backlog, l'avancement des développements, bugs, et suivi projet.

Tests et qualité

Les APIs étant critiques et une indisponibilité des APIs impactant le chiffre d'affaires, la société PayeTonKawa souhaite s'assurer de la qualité des livrables pour s'assurer que ses revendeurs ainsi que le Webshop puissent commander à tout instant.

Il faudra donc mettre en place des outils afin de pouvoir s'assurer de la qualité des livrables, ainsi qu'un outil permettant de mesurer la qualité du code source et la dette technique des applications ainsi que les alertes de sécurité potentielles (OWASP TOP 10 sera le référentiel que la société souhaite utiliser).

3.2 Objectifs :

Le travail demandé sur ce projet consiste à :

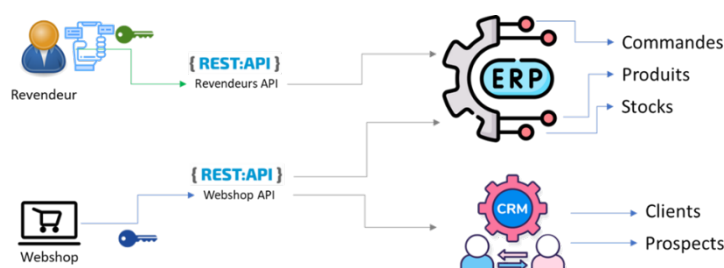
- Créer 2 APIs REST permettant l'exposition des données de l'ERP et du CRM telles que décrites en annexes (voir swagger des APIs).
- Sécuriser l'accès aux APIs pour permettre de n'exposer une API qu'à un utilisateur donné.
- Écrire 1 programme (langage ouvert) pour chaque API.

Les sources seront stockées sur un dépôt git.

- Créer une application mobile disponible sur Android et iOS consommant des APIs REST et pouvant afficher des modèles en réalité augmentée.

- Configurer une usine logicielle capable d'exécuter des pipelines de CI (intégration continue) et des pipelines de CD (Déploiement continue)

L'architecture doit évoluer pour être équivalente à cette cible :



3.3 Les cibles :

Le webshop aura accès à l'API « Webshop » uniquement via une clé d'autorisation dédiée.

Il n'y a pas de clé par client du webshop.

Les revendeurs auront accès à l'API « Revendeurs » uniquement via l'application mobile.

Chaque revendeur doit avoir une clé d'autorisation qui lui est propre. Le but est de pouvoir modifier une clé ou supprimer un accès sans impacter tous les revendeurs.

3.4 Livrables :

Pour chacune des APIs livrées, la société PayeTonKawa souhaite que soient livrés et accessibles à tout instant :

- Le code source, disponible sur un repository git
- Une collection POSTMAN et un fichier environnement permettant de tester manuellement l'API
- La documentation technique décrivant les points suivants (argumentés) :
 - Schéma d'architecture (infrastructure et dépendances)
 - Sécurité de l'API
 - Langage de programmation
 - Règles d'hébergement (scaling de la solution)
 - Gestion du code source
 - Gestion du déploiement (CI / CD)

Pour l'application mobile, la société PayeTonKawa souhaite que soient livrés et accessibles à tout instant :

- Le code source, disponible sur un repository git
- La documentation technique décrivant les points suivants (argumentés) :
 - Langage de programmation
 - Gestion du code source
- Gestion du déploiement (CI / CD)

3.5 Contraintes techniques :

API Webshop

Le langage de programmation est libre. Python, C# Java. Il devra être justifié.

Vous êtes libres de construire le swagger de l'API en fonction des contraintes technique des clients et des besoins des applications. Vous ne devez pas exposer d'informations inutiles.

Application mobile

Le langage de programmation est libre. De même que pour le développement des interfaces utilisateurs le client PayeTonKawa attend des propositions qui soient argumentées.

Pour s'authentifier, l'utilisateur devra scanner un QRCode depuis l'application mobile. Ce QRCode contiendra la clé d'authentification qui sera utilisée par l'application pour consommer les APIs REST. Un mail contenant le QRCode sera envoyé via l'API REST lorsqu'un nouvel utilisateur sera créé :

Appel de l'API REST pour créer un utilisateur (email obligatoire) > Génération d'une clé d'authentification > Envoi du QRCode via email > Scan du QRCode par l'application mobile par le revendeur.

Intégration continue

Le choix de l'outil de CI/CD est libre et devra être justifié (ex : Jenkins, Github Actions, ...)

La société utilise actuellement GitFlow pour sa gestion de branches et souhaite garder ce principe pour tous ses repositories. Les commits devront être organisés pour être lisible et refléter l'évolution du produit sur la branche de développement et la branche principale. L'objectif étant de pouvoir facilement retracer les évolutions en cas d'erreur en production pour retrouver le code et l'évolution ayant générée un bug.

Hébergement des APIs

Le choix de l'hébergement devra être justifié. Dans le cadre de la démonstration un déploiement local est accepté mais une solution hébergée chez un hébergeur devra être étudiée et argumentée.

Tests

Des tests unitaires devront être mis en place et une couverture du code d'au moins 95% est attendue.

Des tests d'intégration automatisés devront être mis en place afin de pouvoir tester de manière automatisée les APIs déployées.

La plateforme d'intégration continue devra être utilisée pour automatiser ces exécutions.

IV– EXPRESSION DE LA DEMANDE – RÉALISATION ATTENDUE DE L'APPRENANT.E

La réalisation attendue de l'apprenant est de se mettre à la place du prestataire réalisant les applications.

Il devra être décisionnaire tout au long du projet et proposer une solution technique en cohérence avec le cahier des charges du client. Cela nécessite qu'il montre son savoir-faire sur les sujets suivants :

- Choix de l'outil de développement et du langage de programmation à adopter
- Appel d'API Rest
- Sécurisation d'une API Rest
- Développement et tests unitaires
- Réalisation de documentation technique
- Mise en place de pipelines de CI / CD sur un repository git

V– SPÉCIFICATION(S) TECHNIQUE(S), DOCUMENTATION, ANNEXES, SCHEMAS, BASES

...et tout élément nécessaire à la bonne réalisation du projet

5.1 : Sources de données

Des données du CRM et de l'ERP sont mockées et mises à votre disposition via des APIs REST. Vous pouvez donc effectuer vos tests d'intégration sur la base de ces APIs. Les données renvoyées étant mockées, le contenu renvoyé peut ne pas avoir de lien avec le contexte (cafés).

Le mock des APIs ERP et CRM est accessible via l'url :

<https://615f5fb4f7254d0017068109.mockapi.io/api/v1>.

CRM

La liste des clients issue du CRM est accessible via l'API REST : /customers.

La liste des commandes d'un client est accessible via l'API REST /customers/<customer id>/orders.

La liste de produits d'une commande est accessible via l'API REST /customers/<customer id>/orders/<order id>/products.

Ex d'url pour accéder aux produits de la commande 25 du client 8 :

<https://615f5fb4f7254d0017068109.mockapi.io/api/v1/customers/8/orders/25/products>

Cette url peut ne pas renvoyer de données. 25 et 8 ayant été choisis au hasard.

ERP

La liste des produits issue de l'ERP est accessible via l'API REST : /products.

Le détail d'un produit peut être accessible via l'API REST : /products/<product id>.

La liste des produits peut varier des informations produits liées aux commandes (cf CRM).

Cette différence vient de la plateforme de Mock. Cela ne doit pas vous bloquer pour vos développements et tests.

5.2 : Réalité Augmentée

Vous êtes libres de pouvoir récupérer des modèles 3D préfaits ou d'en construire. L'objectif n'est pas de valider que vous savez construire des modèles 3D, mais de pouvoir d'en utiliser dans un contexte de réalité augmentée:

Positionner un modèle virtuel sur une table pour simuler le rendu final.

- Les compétences évaluées durant cette MSPR :
- *Collecter les besoins métiers des utilisateurs en menant des interviews auprès d'eux pour comprendre leurs activités et leurs contraintes métier afin d'étudier les opportunités et la faisabilité technologique d'une solution applicative spécifique ou métier.*
- *Concevoir une architecture applicative selon la complexité du système d'information existant de type architecture distribuée, ou micro-service évolutive et tolérante aux pannes.*
- *Développer une application adéquate selon la stratégie applicative de l'environnement en utilisant un langage de programmation approprié dans le respect du cahier des charges établi afin de répondre aux besoins utilisateurs/directions métiers.*
- *Développer une solution applicative intégrée en utilisant le paramétrage et le langage de programmation spécifique de l'éditeur dans le respect du cahier des charges établi afin de répondre aux besoins utilisateurs/directions métiers.*
- *Effectuer les tests de la solution applicative paramétrée ou développée pour identifier les erreurs et dysfonctionnements et établir les plans de correction/d'amélioration avant sa mise en production.*
- *Appliquer l'intégration continue dans le cadre du développement d'une application en utilisant un outil d'intégration continue afin de vérifier la conformité de la solution et les besoins utilisateurs.*
- *Vérifier la conformité entre la solution développée ou paramétrée et les fonctionnalités attendues à partir des retours des directions métiers afin de rédiger la documentation et les référentiels orientés utilisateurs.*
- *Conduire le changement auprès des métiers lors du déploiement d'une solution applicative ou intégrée en mettant en place une démarche de participation, de communication et de formation pour accompagner les utilisateurs à l'intégration du nouvel outil dans leurs habitudes de travail.*