# Tableau des compétences

|  |  |
| --- | --- |
|  | Pages correspondantes |
| BC01 : Concevoir des applications numériques en intégrant les recommandations de sécurité | |
| Formaliser les procédures des services utilisateurs pour recenser les résultats attendus  Diagramme Use case / Diagramme activité / de séquence | Aquanote  FUTE ? |
| Prendre en compte les impératifs utilisateurs en respectant les contraintes des recommandations qualité de la norme en vigueur pour l’architecture des logiciels  Faire en sorte qu’un aveugle puisse aller sur mon site(norme d’accessibilité) | Aquanote  FUTE |
| Concevoir l’architecture d’une solution fiable en identifiant les spécificités d’une activité pour produire du logiciel générique réutilisable  concevoir des services et composant réutilisable | FUTE  Aquanote ? |
| Concevoir des services d’accès aux données indépendants du mode de stockage en garantissant la sécurité des données pour produire du logiciel partageable  Créer des services qui permet d’accéder a la bdd | Aquanote  Cat? |
| Envisager toutes les possibilités, même les plus improbables pour livrer un logiciel déterminé en recherchant systématiquement l’erreur ou le dysfonctionnement  Chercher les erreurs de code ou d affichage en conditions extreme | Fute  Aquanote ? |
| Estimer la charge de traitement et la puissance de calculs nécessaire proportionnellement aux nombre d’utilisateurs simultanés en vue d’anticiper les évolutions  Prendre en compte que l’app doit avoir plusieurs utilisateurs en simultanés | Aquanote(choix serveur lors du deploiement)  Fute ? |
| Respecter une norme de présentation des écrans et documents de sortie en utilisant les outils de maquettage appropriés en vue de permettre l’adaptabilité des sorties garantissant leur l’accessibilité à différents niveaux de handicap  Faire des maquettes sur figma | Aquanote Fute ? |
| Identifier les risques et leur niveau de criticité pour permettre leur prévention  Faire de la cybersécurité, rendre son app sécurisé | Aquanote (.env, mdp hashé) |
| Orienter son style de programmation en vue de produire du code lisible, maintenable, robuste, fiable, efficace par une approche méthodologique objet  Faire du code propre | Fute  Aquanote |
| Garantir un accès sécurisé aux données en évitant toute corruption de la base de données, par l’usage de contraintes d’intégrité et de déclencheurs  Cacher l’accès à la bdd avec un fichier .env ou  Vérif input dans le controller avant de push bdd | Aquanote |
| BC02 - Piloter un projet DevOps de développement d’application numérique. | |
| Formaliser les procédures des services utilisateurs en contrôlant le respect du management des processus de l’entreprise  Faire des diagrammes / de la doc comme l’entreprise la fait | A REVOIR |
| Réaliser une réingénierie d’un processus de l’entreprise en tenant compte des règles de management de l’entreprise dans un but d’amélioration des résultats et/ou des conditions de travail  Changer une organisation dans l’entreprise (ex maintenant daily) | Explication equipe cat |
| Formaliser la circulation des documents générés en identifiant les acteurs concernés et leur rôle ainsi que les rubriques utilisées et leur provenance  Définir les acteurs d’un document officiel, leurs roles, provenance des sources | Cahier des charges FUTE |
| Modéliser une base de données adaptée aux attentes en formalisant les règles de gestion et d’organisation de l’entreprise des processus concernés et en tenant compte d’un existant possible  Faire un document qui explique les nomenclatures d’une bdd | Aquanote |
| S’insérer dans l’urbanisation présente et future du système d’informations en concevant des éléments logiciels réutilisables et structurés en couches  Avoir une architecture logicielle en mvc ou en service composant. Faire des composants réutilisables. | FUTE Aquanote |
| Choisir le degré de réutilisation à utiliser selon le cas de figure en décidant collectivement en équipe d’une réutilisation totale, partielle ou une écriture neuve et des briques à réutiliser en tentant de se rapprocher du niveau maximum de satisfaction CMMI  Evaluer si c’est mieux de réutiliser un composant ou non, suivant si cela rend le code plus complexe ou plus simple | FUTE |
| Utiliser l’expérience vécue pour anticiper l’avenir en vue d’estimer des délais de réalisation compte tenu du taux réel de disponibilité du réalisateur et des contraintes date départ/date livraison  Anticiper le planning | FUTE |
| Coordonner un projet de développement en utilisant les outils et méthodologies de gestion de projet Agile afin de respecter les contraintes définies (coûts, délais, qualité), tout en minimisant les risques  Gerer un projet en méthode agile | FUTE(je vais voir mon chef à chaque implémentation pour avoir son retour…) LABO ? |
| Clôturer une mission de développement en faisant valider le livrable par les parties concernées et en respectant les préconisations CFTL  Se faire valider ses livrables par le client | FUTE |
| Adapter son discours à l’auditoire en appuyant ses manipulations et explications fonctionnelles à partir des contraintes exprimées tout au long du projet pour obtenir une bonne adhésion des décideurs ou de leurs représentants  Adapter des explications orales de dev suivant l’auditoire | FUTE  LABO ? |
| Réaliser la procédure d’intégrabilité d’un logiciel ou d’un correctif dans l’environnement de tests selon les bonnes pratiques ITIL en vérifiant que l’intégralité des points de contrôles sont positifs  Faire des tests unitaires + vérifier que l’ensemble fonctionne correctement à la main | Aquanote |
| Interagir efficacement dans un environnement de travail collaboratif en reformulant la demande et en adaptant son discours à l’auditoire pour obtenir un niveau de compréhension des demandes optimum en tenant compte des collaborateurs en situation de handicap  Adapter son discours de dev a son interlocuteur | Fute (explication erreur composant log) |
| User d’une communication professionnelle tant en français qu’en anglais en structurant des informations sur une thématique donnée afin de les partager au sein de la structure ou à l’externe  Faire une présentation de son travail a son équipe interne ou externe | Fute(reunion client et methode scrum) |
| BC03 - Développer des applications numériques. | |
| Utiliser les ressources à sa disposition ou faire appel si besoin à un expert externe pour contrôler l’identification et la teneur du résultat attendu ou approfondir un point technique  Faire des réunions clients pour changer les objectifs | Fute |
| Décomposer un problème complexe en sous- problèmes en faisant des analogies et des différenciations tout en changeant d’approche, de point de vue, face à un obstacle en vue de résoudre un problème algorithmique  Faire de l’algorithmie en expliquant  (expliquer son code) | Fute  Aquanote ? |
| Traduire une solution algorithmique dans un langage de codage informatique avec l’utilitaire approprié  Passer d’un document à du code | Aquanote(diagramme a code) Fute ?(cahier des charges a code) |
| Modifier un algorithme sans générer de dysfonctionnements en comprenant et s’adaptant si besoin au mode de pensée de son auteur  Modifier du code | Fute(composant log) |
| Remédier aux erreurs de codage ou de logique en comprenant ou interprétant les messages d’erreur du compilateur ou du système d’exploitation pour mettre au point un élément logiciel opérationnel  Dire qu’on a des erreur de code et qu’on les répare | Fute (composant log) aquanote ? |
| Intégrer des éléments logiciels hétérogènes spécifiques et en réutilisant des services logiciels externes, en local ou à distance pour produire des exécutables livrables en conformité avec la politique RSE  Dire qu’on utilise des qui respecte le RSE, les politiques d’entreprise en terme de vie privé, de sécurisation des données, … | Fute (librairie bootstrap et librairie angular material, respecte la politique RSE (voir avec chatgpt))  Aquanote ? |
| Préparer des jeux d’essai en envisageant toutes les possibilités dans le but de livrer un logiciel déverminé exempt d’anomalies logiques et fonctionnelles  Faire des tests unitaires | Aquanote |
| Estimer son taux de disponibilité réel et rendre compte de son travail en renseignant l’outil de suivi permettant de constater l’avance de la tâche en cours et sa répercussion sur l’ensemble du projet  Faire une estimation de temps et utiliser jira | Fute(jira + planning)  Cat ? |
| BC04 - Réaliser une interface d’échange de données informatisées. | |
| Procéder à une analyse organique d’un logiciel existant par l’étude du code des programmes et des données qui sont accédées dans le but de disposer d’une documentation technique du logiciel jusque-là inexistante ou indisponible  Faire une doc qui n’exister pas encore sur un logiciel/composant | Fute (readme de fin de projet) |
| Faire des analogies et des différenciations entre les données à échanger entre logiciels à l’aide des dictionnaires de données disponibles ou reconstitués en vue de permettre l’échange de données entre les logiciels identifiés  Faire de l’échange de donnée par dictionnaire entre 2 logiciels | Fute |
| Produire des données indisponibles en agrégeant, consolidant ou calculant automatiquement ces nouvelles données à partir de celles existantes dans le but de favoriser les échanges de données entre logiciels dans le respect de la RGPD  Cacher des données en bdd, ex : Crypter et Décrypter un mot de passe en bdd | Aquanote (hash mdp) |
| Permettre l’exportation et l’importation de données entre logiciels en utilisant des formats compatibles entre les systèmes émetteurs et récepteurs grâce à des flux synchrones ou asynchrones  Mise en place d’api et de websocket | Fute  Aquanote ?(routeur api) |
| Écrire des scripts système en langage de commande ou en shell système pour automatiser l’installation, la configuration de systèmes d’exploitation et de middleware permettant la création, la configuration de machines virtuelles, de serveurs d’applications, Web et bases de données dans le but d’adapter et simuler en réel l’environnement d’exécution du logiciel à tester  Ecrire un dockerfile | Fute |

Table des matières

[Tableau des compétences 1](#_Toc141043010)

[Présentation personnelle 6](#_Toc141043011)

[Présentation de l’entreprise 6](#_Toc141043012)

[Historique 6](#_Toc141043013)

[Activités et chiffre clés 6](#_Toc141043014)

[Mon équipe et mes activités 6](#_Toc141043015)

[Présentation du client 6](#_Toc141043016)

[Équipe et activités 6](#_Toc141043017)

[Besoins exprimés 6](#_Toc141043018)

[Définition du projet 6](#_Toc141043019)

[Cahier des charges 6](#_Toc141043020)

[Architecture des interfaces 7](#_Toc141043021)

[Architecture logicielle 7](#_Toc141043022)

[Méthodologie du projet 7](#_Toc141043023)

[Planning et Livraisons 7](#_Toc141043024)

[Logiciels de travail collaboratif 7](#_Toc141043025)

[Méthode Scrum 7](#_Toc141043026)

[Réunions client 7](#_Toc141043027)

[Implémentation du composant log et de son service 7](#_Toc141043028)

[Contexte et objectif 7](#_Toc141043029)

[Interface du composant 7](#_Toc141043030)

[Mise en œuvre et réalisation 7](#_Toc141043031)

[Récupération de la donnée 7](#_Toc141043032)

[Architecture du service 7](#_Toc141043033)

[Architecture du composant 7](#_Toc141043034)

[Problématique rencontrée et solution trouvée 7](#_Toc141043035)

[Identification de la problématique 7](#_Toc141043036)

[Réunion client 8](#_Toc141043037)

[Nouvelle solution implémentée 8](#_Toc141043038)

[Transformation du projet en livrable 8](#_Toc141043039)

[Création d’un Dockerfile 8](#_Toc141043040)

[Création d’une documentation technique 8](#_Toc141043041)

[Validation du livrable final par le client 8](#_Toc141043042)

[Projet personnel : Outil de sauvegarde et de suivis des analyses d’eau 8](#_Toc141043043)

[Présentation du projet 8](#_Toc141043044)

[Cahier des charges 8](#_Toc141043045)

[Conception UML 8](#_Toc141043046)

[Conception MCD et MLD 8](#_Toc141043047)

[Maquettage 8](#_Toc141043048)

[Architecture logicielle 8](#_Toc141043049)

[Tests Unitaires 9](#_Toc141043050)

[Déploiement sur serveur 9](#_Toc141043051)

[Conclusion 9](#_Toc141043052)

[Bibliographie 9](#_Toc141043053)

[Glossaire 9](#_Toc141043054)

[Table des illustrations 9](#_Toc141043055)

[Annexes 9](#_Toc141043056)

# Présentation personnelle

# Présentation de l’entreprise

## Historique

Activités et chiffre clés

## Mon équipe et mes activités

2. Changer une organisation dans l’entreprise (ex maintenant Daily)

# Présentation du client

## Équipe et activités

## Besoins exprimés

# Définition du projet

## Cahier des charges

1.Aveugle peut aller sur le site(double)

2.Définir les acteurs d’un document officiel, leurs rôles, provenance des sources

3.utilise des librairie qui respectent le RSE, politique entreprise sur vie privé, sécu des donnée

## Architecture des interfaces

## Architecture logicielle

2. avoir architecture mvc ou service composant pour être au max réutilisable(double)

4.faire de l’échange de donnée par dictionnaire entre deux logiciels

4.mise en place d’api et de websocket(double)

# Méthodologie du projet

## Planning et Livraisons

2.Anticiper le planning

3.faire une estimation de temps et utiliser jira(double)

## Logiciels de travail collaboratif

3.faire une estimation de temps et utiliser jira(double)

## Méthode Scrum

2.Gerer un projet en méthode agile

2.Faire une présentation de son travail a son équipe interne ou externe

## Réunions client

2.Adapter des explications orales de dev suivant l’auditoire

2.Faire une présentation de son travail a son équipe interne ou externe

# Implémentation du composant log et de son service

## Contexte et objectif

1.Concevoir service et composant réutilisable(double)

(Partir sur un principe de segmentation par test dès le début)

## Interface du composant

## Mise en œuvre et réalisation

### Récupération de la donnée

### Architecture du service

4.mise en place d’api et de websocket(double)

### Architecture du composant

1.Faire du code propre(double)

2.Evaluer en équipe de la réutilisation ou non d’un composant

3.faire de l’algorithmie en expliquant son code

## Problématique rencontrée et solution trouvée

### Identification de la problématique

1.Chercher les erreurs de code ou d’affichage en condition extrême

### Réunion client

2.Adapter son discours de dev a son interlocuteur

3.faire des réunions clients pour changer les objectifs

(je parle de la redéfinition des objectifs)

### Nouvelle solution implémentée

3.dire qu’on a des erreurs de code et qu’on les répare

2.Faire tests unitaires et tests manuelles(double)

3.Modifier du code

# Transformation du projet en livrable

## Création d’un Dockerfile

4.Ecrire un Dockerfile

## Création d’une documentation technique

4.faire une doc qui n’exister pas encore sur un logiciel/composant

## Validation du livrable final par le client

2.Se faire valider ses livrable par le client

# Projet personnel : Outil de sauvegarde et de suivis des analyses d’eau

## Présentation du projet

1.Aveugle peut aller sur le site(double)

## Cahier des charges

1.Faire de la cybersécurité, rendre son app sécurisé (.env, mdp hashé)

1.Faire du code propre(double)

1.Cacher l’accès à la bdd avec un fichier .env, Verif input dans Controller et push bdd(double)

## Conception UML

1.Diagramme usecase/activité/sequence

1.Cacher l’accès à la bdd avec un fichier .env, Verif input dans Controller et push bdd(double)

## Conception MCD et MLD

1.Utiliser une bdd qui peut se connecter a n’importe quel app

2.faire un document qui explique les nomenclatures d’une bdd

4.Cacher des données en bdd, hasher password

## Maquettage

1.Faire des maquettes sur figma

## Architecture logicielle

1.Concevoir service et composant réutilisable(double)

2. avoir architecture mvc ou service composant pour être au max réutilisable(double)

3.passer d’un document diagramme à du code

## Tests Unitaires

2.Faire tests unitaires et tests manuelles(double)

3. faire des tests unitaires

## Déploiement sur serveur

1.Prendre en compte que l’app soit avoir plusieurs utilisateurs en simultanés

# Conclusion

# Bibliographie

# Glossaire

# Table des illustrations

# Annexes

Point a valider :

Clôturer une mission de développement en faisant valider le livrable par les parties concernées et en respectant les préconisations CFTL.

Se faire valider ses livrables par le client

Procéder à une analyse organique d’un logiciel existant par l’étude du code des programmes et des données qui sont accédées dans le but de disposer d’une documentation technique du logiciel jusque-là inexistante ou indisponible.

Faire une doc qui n’exister pas encore sur un logiciel/composant

Écrire des scripts système en langage de commande ou en shell système pour automatiser l’installation, la configuration de systèmes d’exploitation et de middleware permettant la création, la configuration de machines virtuelles, de serveurs d’applications, Web et bases de données dans le but d’adapter et simuler en réel l’environnement d’exécution du logiciel à tester.

Ecrire un dockerfile

A voir si je peut rajouter (mais déjà fait) :

1.Créer des services qui permet d’accéder à la bdd