



2022/2023

Projet de fin d'année

BERTRAND Julien

IVARS Grégoire

BEGOT Kévin

ISITECH

Table des matières

Cahier des charges du projet QRCoffee	4
Origine du projet	4
Expression du besoin	4
Présentation du projet.....	4
Charte Graphique	5
Nos couleurs	5
Notre Logo.....	6
Présentation des différentes vues de l'application	7
MCD	13
Contraintes du projet	15
Périmètre du projet	15
Composition de l'équipe de projet.....	16
Distribution des rôles	16
La gouvernance	16
Étude du marché.....	17
Contexte	17
Analyse SWOT.....	18
Analyse Macro-Économique et Micro-Économique	19
Etude des Contraintes	20
Introduction.....	20
Les contraintes.....	21
Contrainte liée à notre projet.....	22
Etude des moyens et ressources disponible.....	23
Les ressources humaines	23
Les ressources matérielles	23
Les ressources financières	23
Moyens nécessaires à la réalisation.....	24
Etude de la transversalité	26
Matrice des risques	26
Evaluation de l'impact : Délai.....	26
Evaluation de l'impact : Qualité	26
Evaluation de l'impact : Coût	26
Mise en place des mesures préventives	27
Stack Technique	28
Front-end - React + TypeScript (TSX).....	28
Back-end - Node.js + TypeScript (TSX)	28
Authentification - Clerks	28
Base de données - MongoDB.....	28
Contrôle de Version - Git	28

Hébergement du Code - Bitbucket	28
Tests unitaires sur le Front-end avec Cypress:	29
Tests unitaires sur le Back-End avec Mocha et Chai.....	29
Enveloppe Budgétaire à prévoir	30
Financement du projet.....	30
Evolution de l'application	31
Cours terme :.....	31
Moyen terme	31
Long terme	32
Rapport du projet.....	33
Présentation des outils de gestion utilisés	33
Project	33
Atlassian.....	38
Méthodologie de gestion de projet	41
La méthode en cascade + Kanban	41
Architecture de l'application.....	42
Architecture FrontOffice	42
Architecture de l'authentification	44
Architecture Back-end.....	46
Base de données	47
Mise en place des tests unitaires Backend et FrontEnd.....	48
Mise en place de l'hébergement	50
Hébergement Web.....	50
Hébergement de l'API.....	51

Cahier des charges du projet QRCoffee

Origine du projet

Le monde se digitalise. De plus en plus d'entreprise et de particuliers cherchent à faire le lien entre des supports de communication physique et digitaux de manière totalement intuitif.

QRCoffee est une jeune start-up composée de 3 futurs experts SI spécialisés en développement WEB.

Implantés à Lyon depuis 2022, nous proposons un service tout en un de génération de QRCode, destinés aux particuliers et aux professionnels.

Notre but est de permettre à nos clients de générer des QRcodes statiques et dynamiques afin qu'ils puissent selon leur besoin d'afficher un contenu rapidement.

Ce projet vise à faciliter le partage des données de nos clients et de proposer un service tout en un.

Expression du besoin

Les professionnels (privé, libéraux) ont tous besoin d'une visibilité sur le web afin de toucher un maximum de prospects.

Les particuliers ont besoin de pouvoir accéder à des données rapidement comme un code Wifi, un numéro de téléphone, un document et bien d'autres.

Le problème est que l'accès à ces données sont souvent long.

Afin de remédier à cela, nous proposons une application Web permettant la génération de QRCode afin d'accéder à la donnée souhaitée en quelques secondes. Il faudra pour cela que l'utilisateur scanne le QRCode.

Présentation du projet

Notre objectif principal est de satisfaire le client. Pour cela, nous lui proposons dans un premier temps, une génération de QRCode statique customisable afin de résoudre la problématique de l'obtention de données rapidement.

Dans un second temps, nous offrons la possibilité via une offre premium, la génération d'un QRCode dynamique. Cela permet au client d'obtenir un ensemble de rapport permettant le suivi de son QRCode.

Ces rapports présentent les données suivantes :

Le nombre de lecture

La position géographique de l'utilisateur

Par quel objet le QRCode a-t-il été lu (téléphone, tablette...)

L'heure à laquelle le QRCode a-t-il été lu

Enfin, notre application Web se doit d'être simple d'utilisation notamment grâce à une interface claire, épurée et lisible sur n'importe quel support.

Charte Graphique

Pour mener à bien notre projet, il est essentiel de concevoir une charte graphique.

Elle permet de structurer les idées quant à l'apparence de notre application web, en établissant des règles précises concernant les couleurs, typographies et formes à utiliser. Ainsi, nous garantissons une expérience utilisateur homogène et intuitive.

La charte graphique contribue également à créer une identité visuelle propre à notre projet.

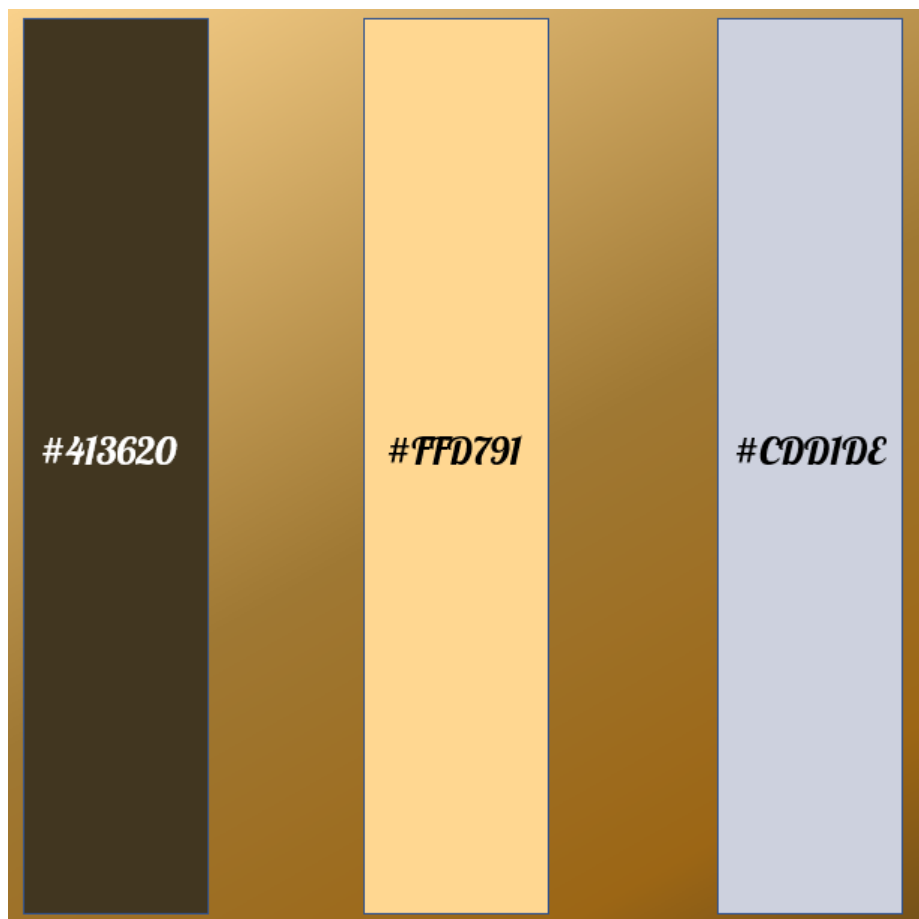
En choisissant des couleurs et des formes spécifiques, nous renforçons notre image de marque et nous démarquons de la concurrence.

Les choix esthétiques que nous faisons reflètent nos objectifs et notre vision, et permettent de communiquer efficacement ces idées à notre public.

Enfin, la charte graphique transmet les valeurs et la philosophie de notre projet à travers le design. Elle est donc un outil essentiel pour communiquer nos idées à notre public.

En somme, la création d'une charte graphique pour notre projet est indispensable pour assurer une expérience utilisateur homogène et intuitive, pour renforcer notre image de marque, et pour communiquer efficacement nos valeurs et notre vision à travers le design.

Nos couleurs



Jaune (#FFD791) :

Cette nuance de jaune est chaleureuse, énergique et optimiste.

En utilisant cette couleur, nous mettons en place une ambiance accueillante et joyeuse.

Le jaune vif peut également attirer l'attention des utilisateurs et mettre en valeur certains éléments importants de votre application web.

Marron (#413620) :

Cette nuance de marron est riche et profonde, ce qui apporte une touche d'élégance et de sophistication à notre design.

Le marron est également associé à la stabilité, à la sécurité et à la nature, ce qui peut contribuer à renforcer l'image de fiabilité et de confiance de notre

start-up.

Gris (#CDD1DE) :

Nous l'avons choisie pour sa douceur et sa sérénité. Elle vient compléter notre palette en apportant une touche de fraîcheur et de tranquillité, soulignant notre volonté de créer un environnement intuitif et apaisant pour nos utilisateurs.

Notre

Logo



Le logo de QRCoffee est une représentation ingénieuse et ludique de l'univers du QRCode, reflétant notre objectif de fournir un service tout-en-un pour la génération de QRCode.

Le design met en évidence un QRCode stylisé qui intègre subtilement une tasse de café, en accord avec notre slogan : "Le QRCode qui fait même le café".

Le nom "QRCoffee" est intégré de manière subtile sous le QRCode/tasse de café, dans une typographie élégante et moderne, renforçant ainsi notre identité

de

marque tout en conservant une lisibilité optimale.

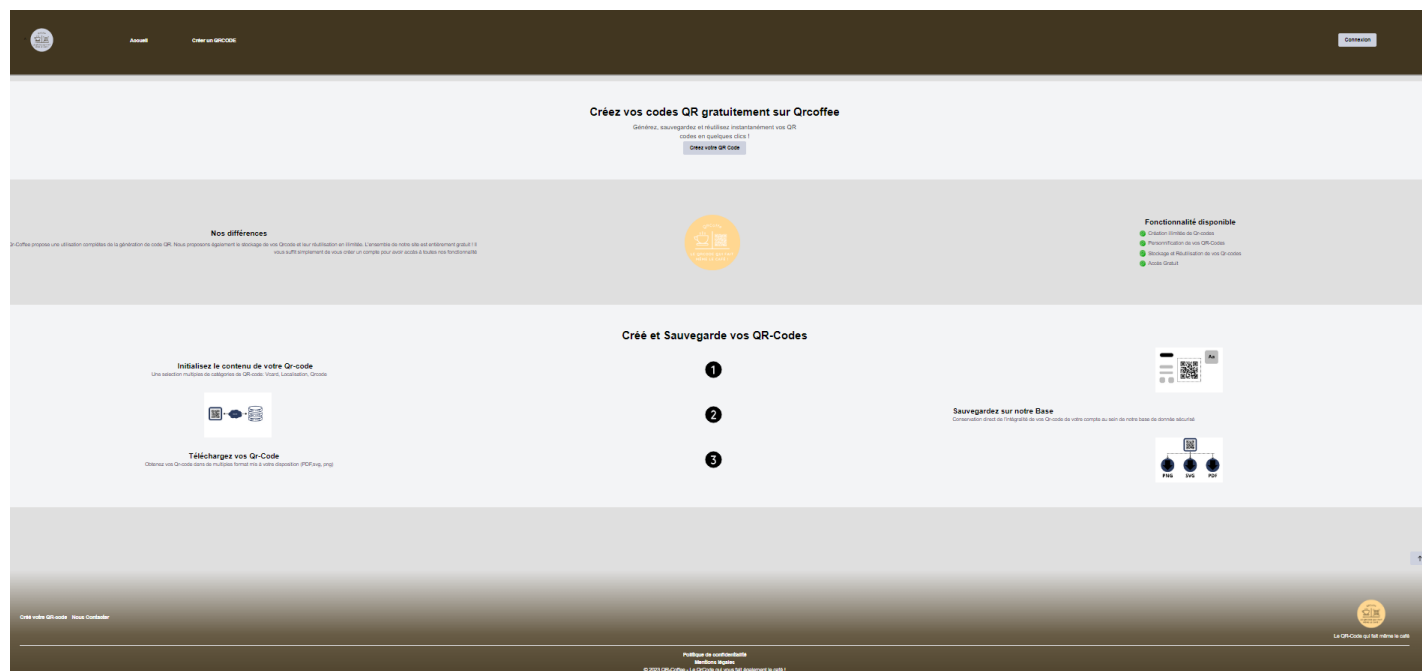
Ce logo est conçu pour être mémorable et unique, représentant parfaitement notre entreprise spécialisée dans la vente de QRCode.

Il communique notre ambition de fournir un service intuitif, professionnel et innovant qui répondra aux besoins de nos clients, tout en ajoutant une touche de convivialité et de plaisir dans leur expérience avec notre plateforme.

Présentation des différentes vues de l'application

Page d'accueil

Vu de la page d'accueil :



La vue de création de QRCode

Qrcode Lien Qrcode Vcard Qrcode Localisation

Nom du Qr-Code

URL

www.google.com

Taille

256

Couleur de l'arrière plan

Couleur du QrCode

Niveau d'erreur

L

Générer le QRCODE

Qrcode Lien **Qrcode Vcard** Qrcode Localisation

Prénom

julien

Nom

bertrand

Couleur de l'arrière plan

Couleur du QrCode

Taille

256

Travail

Ingénieur software

Email

QRCoffee@gmail.fr

Téléphone

0651230375

Adresse

69 route du bonheur

Générer le QRCODE

Qrcode Lien
Qrcode Vcard
Qrcode Localisation

Nom du Qr-Code

Latitude

Longitude

Taille

Couleur de l'arrière plan

Couleur du QrCode

Générer le QRCODE

Pour chaque QRCode, lorsque vous cliquez sur le bouton généré, une prévisualisation du QRCode est générée :

Qrcode Lien
Qrcode Vcard
Qrcode Localisation

Nom du Qr-Code

URL

Taille


Couleur de l'arrière plan

Couleur du QrCode

Niveau d'erreur

Générer le QRCODE

Prévisualisation du QRCode

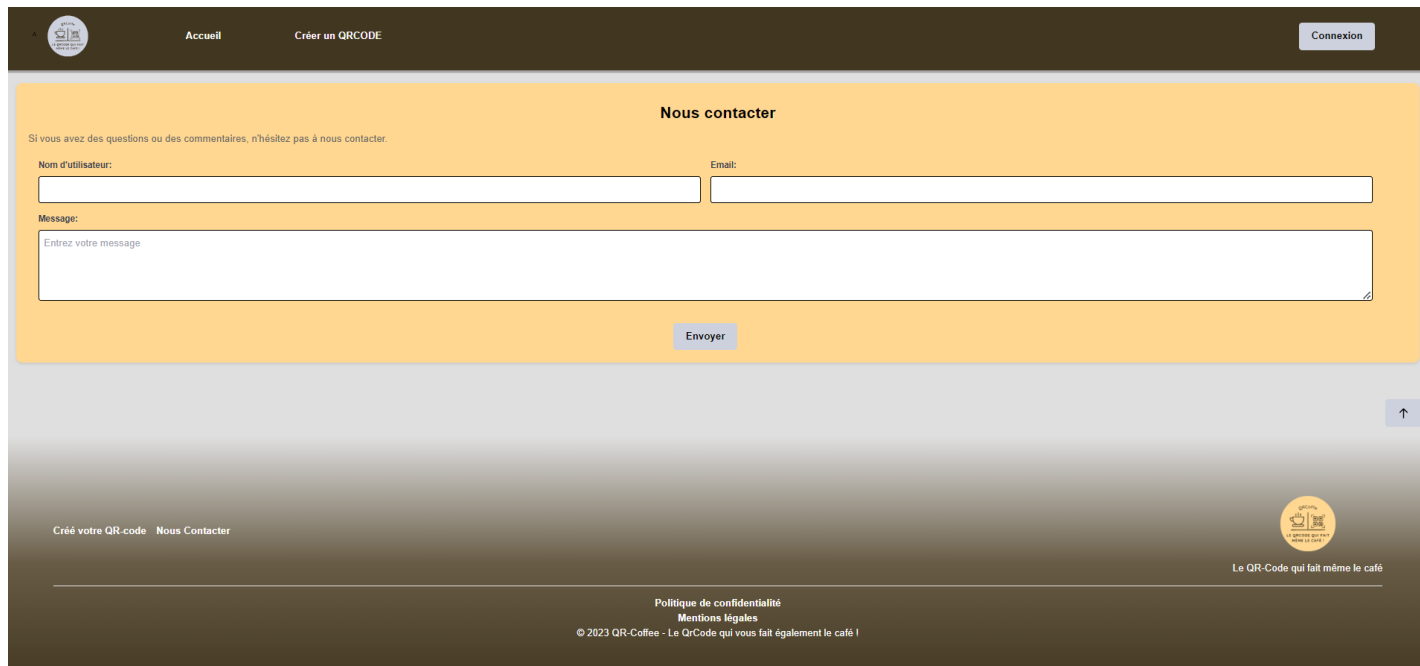


Ajouter la bordure

Télécharger

La vue Contact

Sur cette page, vous pouvez si vous le souhaitez nous envoyer un mail avec un message :



The screenshot shows a web interface for contacting the site. At the top, there is a dark brown header with a logo on the left, navigation links 'Accueil' and 'Créer un QR CODE' in the center, and a 'Connexion' button on the right. Below the header is a large orange section titled 'Nous contacter'. Inside this section, there is a sub-header 'Nous contacter' and a line of text: 'Si vous avez des questions ou des commentaires, n'hésitez pas à nous contacter.' Below this, there are two input fields: 'Nom d'utilisateur:' and 'Email:'. Underneath these is a large text area for 'Message:' with the placeholder text 'Entrez votre message'. At the bottom of the orange section is an 'Envoyer' button. Below the orange section is a grey section with a small blue '↑' button on the right. At the bottom of the page is a dark brown footer. On the left, it says 'Créez votre QR-code' and 'Nous Contacter'. On the right, there is a circular logo with a QR code and the text 'Le QR-Code qui fait même le café'. In the center, there is a line of text: 'Politique de confidentialité', 'Mentions légales', and '© 2023 QR-Coffee - Le QRCode qui vous fait également le café !'.

Accueil Créer un QR CODE Connexion

Nous contacter

Si vous avez des questions ou des commentaires, n'hésitez pas à nous contacter.

Nom d'utilisateur: Email:

Message:

Entrez votre message

Envoyer

↑

Créez votre QR-code Nous Contacter

Le QR-Code qui fait même le café


Politique de confidentialité
Mentions légales
© 2023 QR-Coffee - Le QRCode qui vous fait également le café !

La vue de l'espace QR code

Sur cette page, vous avez accès à une liste complète de tous vos QR codes, avec la possibilité de les filtrer, de les trier, de les supprimer ou de les télécharger.


Espace Qrcode Espace Vcard Espace Localisation

Trier par date Trier par ordre Alphabétique




Créer le: 27/07/2023 16:04
Name: Name inconnue

Supprimer Télécharger



Créer le: 27/07/2023 16:04
Name: zre

Supprimer Télécharger



Créer le: 27/07/2023 16:04
Name: zre

Supprimer Télécharger



Créer le: 27/07/2023 16:04
Name: zre

Supprimer Télécharger




Créer le: 27/07/2023 16:05
Name: zreez

Supprimer Télécharger


Espace Qrcode Espace Vcard Espace Localisation

Trier par date Trier par ordre Alphabétique




Créer le: 27/07/2023 16:07
Name: A

Supprimer Télécharger




Créer le: 27/07/2023 16:08
Name: B

Supprimer Télécharger




Créer le: 27/07/2023 16:08
Name: E

Supprimer Télécharger



Créer le: 27/07/2023 16:08
Name: E

Supprimer Télécharger




Créer le: 27/07/2023 16:08
Name: A

Supprimer Télécharger


Espace Qrcode Espace Vcard Espace Localisation

Trier par date Trier par ordre Alphabétique




Créer le: 27/07/2023 16:05
Name: ezrezre

Supprimer Télécharger



Créer le: 27/07/2023 16:05
Name: ezrezre

Supprimer Télécharger

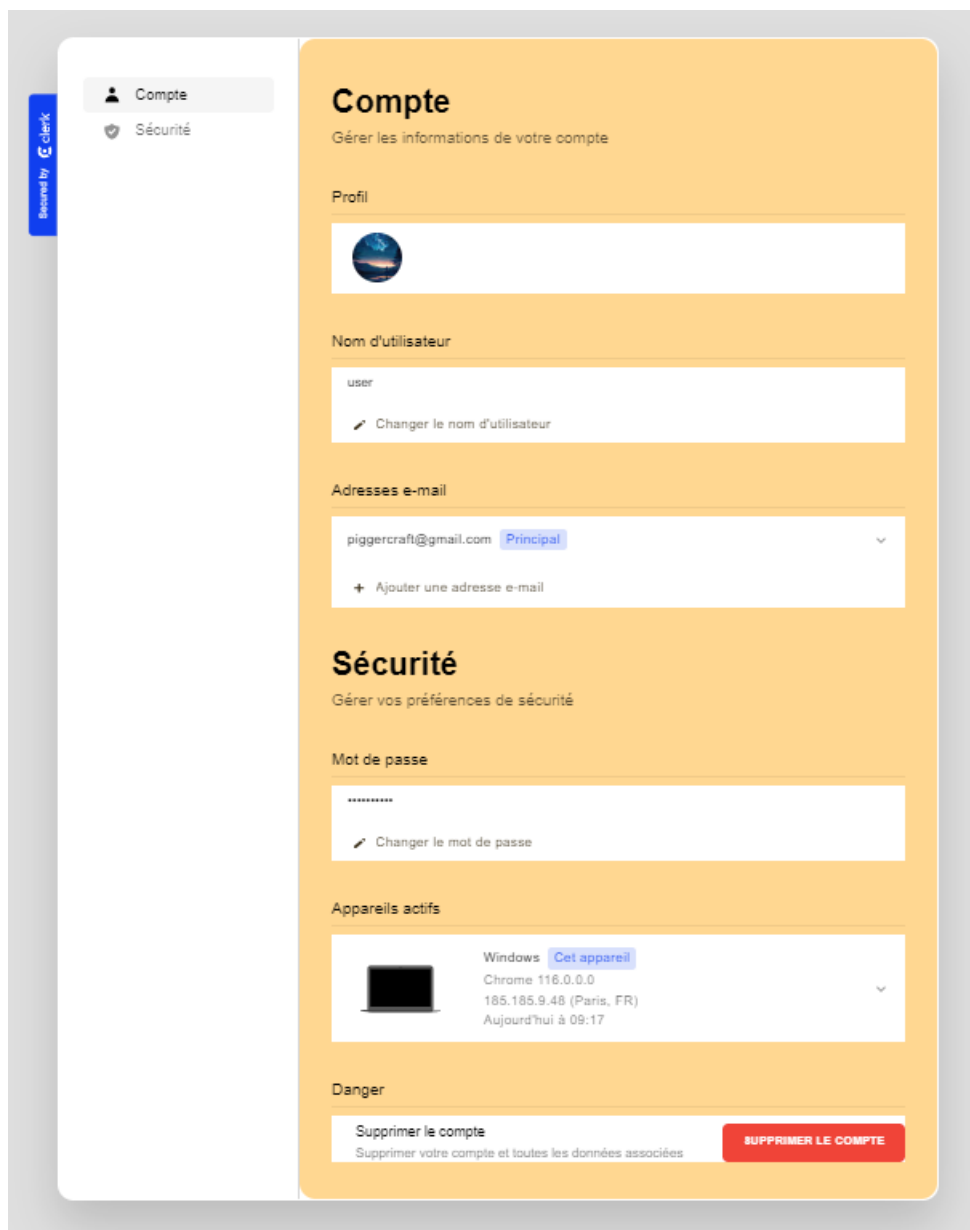


Créer le: 27/07/2023 16:08
Name: AAAA

Supprimer Télécharger

La vue de l'espace mon compte

Sur cette page, vous avez accès à toutes les informations relatives à votre compte, ainsi qu'à la possibilité de modifier votre profil, votre nom d'utilisateur, votre adresse e-mail et votre mot de passe. Vous pouvez également, si vous le désirez, supprimer votre compte de manière irréversible.



Compte
Gérer les informations de votre compte

Profil

Nom d'utilisateur

user

Changer le nom d'utilisateur

Adresses e-mail

piggercraft@gmail.com Principal

Ajouter une adresse e-mail

Sécurité
Gérer vos préférences de sécurité

Mot de passe

Changer le mot de passe

Appareils actifs

Windows Cet appareil

Chrome 116.0.0.0

185.185.9.48 (Paris, FR)

Aujourd'hui à 09:17

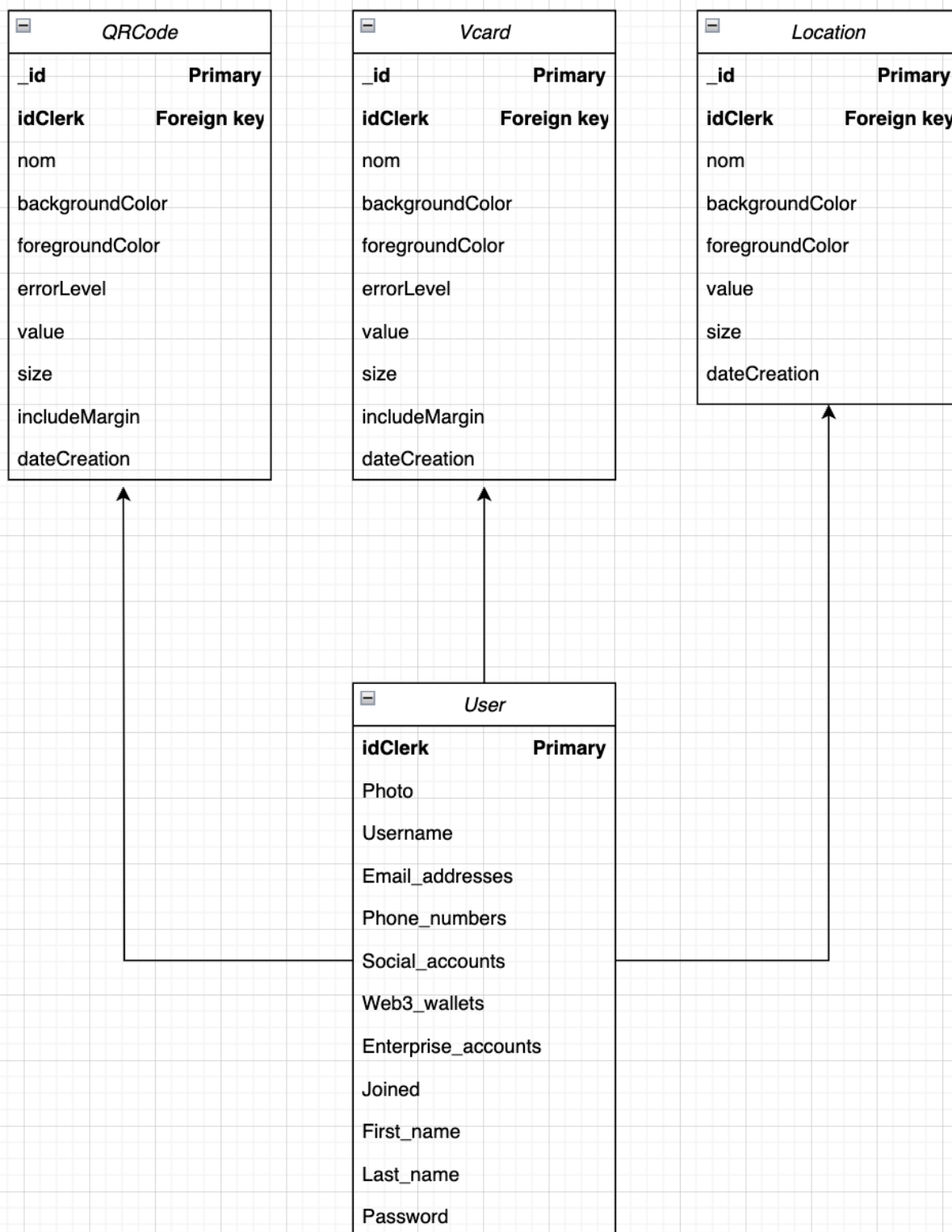
Danger

Supprimer le compte

Supprimer votre compte et toutes les données associées

SUPPRIMER LE COMPTE

MCD



Cette partie vise à décrire le modèle conceptuel de données (MCD) qui sera utilisé pour structurer la base de données et gérer les informations relatives aux QR Codes.

Voici un aperçu détaillé du MCD :

Table "QRCode" :

Cette table sera utilisée pour stocker les QR Codes générés dans notre système.

Elle contient les colonnes suivantes :

- ⇒ "_id" : Un identifiant unique pour chaque QR Code, permettant de les distinguer sans ambiguïté.
- ⇒ "idClerk" : Un champ pour enregistrer l'identifiant de l'utilisateur associé à chaque QR Code.
- ⇒ "nom" : Un champ pour spécifier le nom du QR Code, facilitant son identification.
- ⇒ "backgroundColor" : Ce champ stocke la couleur d'arrière-plan du QR Code.
- ⇒ "foregroundColor" : Ce champ stocke la couleur de premier plan du QR Code.
- ⇒ "errorLevel" : Un champ indiquant le niveau d'erreur autorisé dans le QR Code.
- ⇒ "value" : Ce champ contient le contenu du QR Code, pouvant être un lien, du texte ou d'autres données.
- ⇒ "size" : Un champ pour spécifier la taille du QR Code.
- ⇒ "includeMargin" : Un indicateur booléen pour indiquer si des marges doivent être incluses autour du QR Code.
- ⇒ "dateCreation" : Un champ pour enregistrer la date de création de chaque QR Code.

Table "Vcard" :

Cette table est destinée à stocker des cartes de visite sous forme de QR Codes.

Elle possède les colonnes suivantes :

- ⇒ "_id" : Un identifiant unique pour chaque QR Code, permettant de les distinguer sans ambiguïté.
- ⇒ "idClerk" : Un champ pour enregistrer l'identifiant de l'utilisateur associé à chaque QR Code.
- ⇒ "backgroundColor" : Ce champ stocke la couleur d'arrière-plan du QR Code.
- ⇒ "foregroundColor" : Ce champ stocke la couleur de premier plan du QR Code.
- ⇒ "errorLevel" : Un champ indiquant le niveau d'erreur autorisé dans le QR Code.
- ⇒ "value" : Ce champ contient le contenu de la carte de visite, généralement un lien vers un profil en ligne ou des informations de contact.
- ⇒ "size" : Un champ pour spécifier la taille du QR Code.
- ⇒ "includeMargin" : Un indicateur booléen pour spécifier si des marges doivent être incluses autour de la carte de visite.
- ⇒ "dateCreation" : Un champ pour enregistrer la date de création de chaque QR Code.

Table "Location" :

Cette table est utilisée pour stocker des QR Codes liés à des informations de localisation géographique.

Elle contient les colonnes suivantes :

- ⇒ "_id" : Un identifiant unique pour chaque QR Code, permettant de les distinguer sans ambiguïté.
- ⇒ "idClerk" : Un champ pour enregistrer l'identifiant de l'utilisateur associé à chaque QR Code.
- ⇒ "value" : Ce champ contient les informations de localisation, pouvant être un lien vers des coordonnées géographiques ou d'autres informations spécifiques.
- ⇒ "size" : Un champ pour spécifier la taille du QR Code.
- ⇒ "backgroundColor" : Ce champ stocke la couleur d'arrière-plan du QR Code.
- ⇒ "foregroundColor" : Ce champ stocke la couleur de premier plan du QR Code.
- ⇒ "dateCreation" : Un champ pour enregistrer la date de création de chaque QR Code.

Contraintes du projet

Notre projet QRCoffee a beau être innovant, ce dernier est soumis à diverses contraintes.

Le marché du QR code étant très actif, il existe beaucoup d'applications Web permettant la génération de QRCode.

Les contraintes techniques :

- Pouvoir garantir la cohérence de la base de données.
- Faire en sorte que la web-app soit responsive.
- Faciliter l'accès aux données récolté par le QRCode (interface spécifique nécessaire)

Les contraintes de sécurité :

- Sécuriser l'accès aux données selon les rôles de l'utilisateur.
- Sécuriser les types de données.
- Sécurisation des mots de passe (Hash).
- Sécurisation contre les injections SQL

Les contraintes de performance :

- Pouvoir garantir la fluidité de la web-app avec le flux de données.
- Permettre le stockage de QRCode

Périmètre du projet

Le périmètre de l'application Web se décompose de la manière suivante :

- ⇒ Type de clientèle : Nous visons les particuliers et les professionnels. Notre application Web sera utilisable par tout le monde. Le but est d'aider un maximum de personne souhaitant partager de la donnée rapidement.
- ⇒ Localisons : Le projet étant une application Web, il n'a donc aucune limite. Cependant, il faudra si besoin traduire le site en anglais afin qu'il puisse être compris hors du territoire français.
- ⇒ L'accessibilité : L'application Web devra au maximum respecter les règles RGAA.

Composition de l'équipe de projet

BERTRAND Julien, 26 ans, est diplômé d'un BTS Systèmes numérique option A informatique et réseaux. Après avoir travaillé 5 ans dans les Télécom, il a décidé de reprendre le chemin de l'école dans le but d'obtenir un BAC+5. Pour cela, il a intégré la licence RPI en spécialité développement web en alternance. Il réalise son alternance chez Steamulo Lyon. Il occupe le poste de développeur Web. C'est donc tout naturellement qu'il a pris le poste de chef de projet au sein du projet.

IVARS Grégoire, 21 ans, diplômé d'un BTS Service informatique aux organisations option SLAM. À la suite de ce diplôme, il a décidé de continuer sur une 3e année voir peut-être jusqu'à la 5e année. Actuellement en Licence Responsable de projet Informatique. Alternant dans l'entreprise INETUM en tant que Consultant Junior. On lui a alors attribué le rôle de développeur.

BEGOT Kevin, 20 ans est diplômé d'un BTS Système numérique aux organisations avec une option dans le développement web. Après le BTS, il a décidé de continuer son chemin pour rejoindre une licence RPI spécialisée dans le développement web en alternance chez ADSTRATEGY et occupe le rôle de développeur. Il a pour objectif de valider son BAC+3 est d'obtenir par la suite un BAC+5. Il a donc rejoint le projet QRCoffee au poste de développeur.

Distribution des rôles

BERTRAND Julien : Chef de projet

IVARS Grégoire : Développeur Web

BEGOT Kevin : Développeur Web

La gouvernance

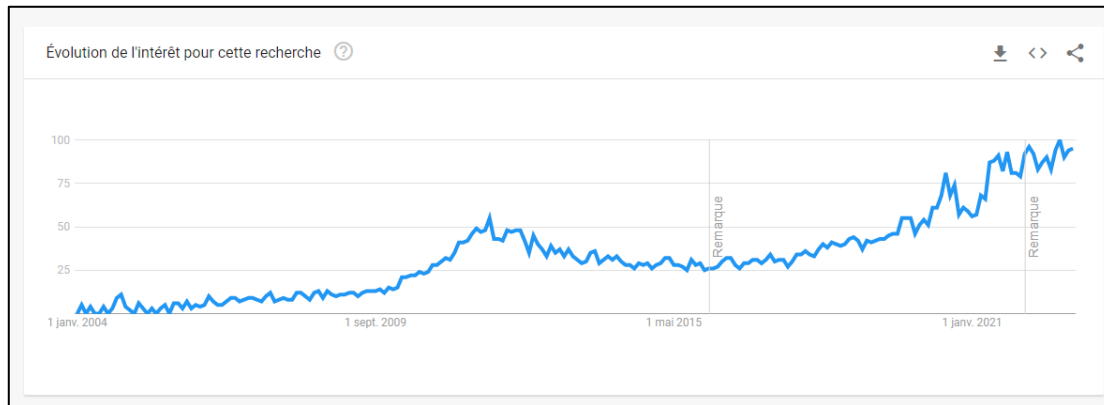
BERTRAND Julien	IVARS Grégoire	BEGOT Kevin
Définition du Front-end et Back-End de l'application	Partie gestion de projet	Définition du Front-end et Back-End de l'application
Analyse du marché	Analyse du marché	Analyse du marché
Encadrement de la partie technique du FRONT-END et BACK-END	Partie écrite du dossier	Encadrement de la partie technique du BACK-END
Partie écrite du dossier	Diaporama	Partie écrite du dossier

Étude du marché

Contexte

Le marché du QRCode est en constante évolution depuis 2004.

Ci-dessous, vous trouverez un tableau qui illustre l'évolution de la demande de QRCode depuis 2004. (Source Google Trend)



Les codes QR dynamiques générés par les utilisateurs ont accumulé un total de 6 825 842 scans d'utilisateurs mondiaux, soit une augmentation de 433 % par rapport aux chiffres de 2021.

Sur la base de la base de données de QRTIGER, voici les 10 premiers pays avec l'activité de numérisation la plus élevée pour le premier trimestre 2022 :

- ⇒ États-Unis – 42,2 %
- ⇒ Inde – 16,1 %
- ⇒ France – 6,4 %
- ⇒ Royaume-Uni – 3,6 %
- ⇒ Canada – 3,6 %
- ⇒ Arabie saoudite – 3,0 %
- ⇒ Colombie – 3,0 %
- ⇒ Malaisie - 2,1 %
- ⇒ Singapour – 1,7 %
- ⇒ Mexique – 1,6 %

Analyse SWOT

Nous avons réalisé une analyse SWOT (Strengths, Weaknesses, Opportunities, Threats) pour nous permettre de faire le point sur les rapports qu'entretient notre projet avec le marché du QRCode. La démarche présente le projet sous un angle stratégique qui nous aide pour prendre du recul et à faire les bons choix tactiques et stratégiques.



Nous en avons déduit plusieurs conclusions :

- ➔ La première est que notre projet a du potentiel.
- ➔ La deuxième est que le marché est en forte demande et cela ne cessent de croître depuis 2004 (source Google Trend)
- ➔ La troisième est que beaucoup de concurrents sont déjà en place :
 - ⇒ QR Code Monkey
 - ⇒ QR Stuff
 - ⇒ QR code generator
 - ⇒ [Tableau de la concurrence](#)

Analyse Macro-Économique et Micro-Économique

Introduction

Un QR code (Quick Response code) est un type de code-barres bidimensionnel qui peut être numérisé pour accéder rapidement à une information sur un support en ligne. Les QR codes sont de plus en plus utilisés dans les transactions commerciales en raison de leur efficacité et de leur rapidité.

L'étude de la macro-économie se concentre sur l'analyse du marché dans son ensemble, tandis que l'étude de la micro-économie se concentre sur les décisions économiques prises par les individus et les entreprises.

Macro-économie

Les QR codes sont devenus très populaires dans les transactions commerciales, car ils permettent aux entreprises de traiter rapidement et efficacement les paiements en ligne.

Ils sont souvent utilisés pour la vente au détail, la restauration et les services financiers, et leur utilisation peut stimuler l'économie en facilitant les transactions commerciales et en permettant aux entreprises de gagner du temps et de l'argent.

Micro-économie

Ils peuvent être utilisés pour améliorer la qualité du service client en permettant aux entreprises de suivre les habitudes d'achat et d'offrir des offres personnalisées. Les consommateurs peuvent également bénéficier de l'utilisation des QR codes, car ils peuvent faciliter la comparaison des prix et des produits, ce qui peut les aider à prendre des décisions d'achat plus informées.

En comparant avec d'autres sites/entreprises proposant des QR codes, nous pouvons voir qu'ils sont de plus en plus populaires et sont utilisés par de nombreuses entreprises, telles que Alipay et WeChat en Chine, qui offrent des services financiers en utilisant des QR codes. Les entreprises peuvent également utiliser des QR codes pour offrir des coupons et des remises à leurs clients, ce qui peut les inciter à acheter davantage.

Ils peuvent également être utilisés pour améliorer la sécurité des transactions en ligne, car les codes QR cryptés peuvent être utilisés pour vérifier l'identité des utilisateurs et pour protéger les informations sensibles. C'est sur cette base que nous est venue l'idée de mettre en place nous-même une plateforme accessible à la génération de QR codes qui viendront également rajouter une fonctionnalité responsive améliorant alors la gestion de données. Nous permettant de devenir de sérieux concurrents.

Conclusion

En conclusion, les QR codes peuvent être considérés comme bénéfiques pour l'économie dans son ensemble, car ils peuvent stimuler les transactions commerciales, améliorer la qualité du service client et renforcer la sécurité des transactions en ligne.

Faisant alors la génération de QR Codes un Marché des plus viables et exploitables.

Etude des Contraintes

Introduction

Définition

Quand on parle de contrainte, le sujet peut-être assez large, mais on peut le définir comme :

- ⇒ Un facteur pouvant nous empêcher ou compromettre les performances globales d'un projet entrepris ou d'un système plus généralement.

L'impact de cette contrainte risque de réduire la capacité de l'équipe à atteindre son objectif fixé.

Les contraintes dans le domaine de l'informatique peuvent être très diverses et variées. Cela peut toucher à la technologie employée, au moyen financier mis en place, à la sécurité nécessaire pour le bon fonctionnement de ce dernier et bien d'autre encore.

Dans notre métier le respect de deadline est primordial, donc tout retard risque d'être très compromettant. Pour ce faire, on peut mettre en place des moyens de gestion de projet pour prévoir et agir sur ces dernières pour éviter qu'elles n'impactent trop le projet.

Théorie des contraintes

La théorie des contraintes peut être perçue comme un moyen d'amélioration continue ou d'administration du changement, et même comme une nouvelle perspective sur la gestion de projet. Néanmoins, son objectif ultime est de fournir aux praticiens l'ensemble complet d'outils dont ils ont besoin pour réussir.

Cette méthode diffère des autres méthodes car le problème ici est d'identifier les contraintes qui empêchent l'atteinte des objectifs énoncés. Développée à l'origine pour le secteur de la production industrielle, la Théorie des Contraintes est maintenant applicable à d'autres domaines.

Les contraintes

En premier temps pour identifier nos éventuelles contraintes, nous allons les lister et définir pour clarifier ce qui peut nous impacter durant la réalisation de notre projet.

Contrainte Sécurité

Lors de l'administration d'un SI, il est crucial de prendre en compte les contraintes de sécurité impliquées, telles que : Les facteurs qui ont un impact sur la technologie moderne et ses utilisateurs comprennent le cloud computing, les réglementations, les options d'utilisation flexibles, l'Internet des objets et la présence de pirates informatiques et d'individus qui utilisent les systèmes d'information. Les processus de collecte et de stockage des données par le SI sont soumis à divers degrés de surveillance réglementaire. Les données peuvent désormais être contrôlées et régulées grâce à la mise en place du Règlement Général sur la Protection des Données (RGPD). La plupart des failles de sécurité proviennent de l'implication humaine, que ce soit par des pirates externes ou des utilisateurs internes à l'entreprise.

Contrainte Contractuelle

Les contrats de licences gratuites sont des accords mutuellement contraignants entre les concédants et les titulaires de licence. Il faut envisager les contraintes qui peuvent être rencontrées aussi bien pour l'usage au sein d'autres licences libres, que lors de tout autre contrat qui peut graviter autour des licences libres.

La contrainte contractuelle peut généralement nuire à un auteur ou créateur sur la publication de ce son projet voire la propriété de ce dernier.

En effet, dans certains cas précis, si ce dernier se retrouve sous un contrat avec une entreprise et crée un produit propulsant considérablement la croissance de l'entreprise.

Ce dernier peut se retrouver privé de toute considération pour ce projet le laissant alors dans l'ombre.

Contrainte Matérielle

Les contraintes Matérielle sont liées à la technologie employée telle qu'un serveur cela peut être défini comme l'indisponibilité de matérielle venant empêcher la réalisation du projet.

Comme cité au-dessus, ça touche les serveurs, les machines mise à disposition les bureaux pour se retrouver et bien d'autres.

Contrainte Technique

Les contraintes techniques sont provoquées principalement par des choses non-physique.

Telle qu'un projet bien trop ambitieux, nécessitant des connaissance et capacité que l'équipe ne possède pas.

Ou trop en avance par rapport à la technologie actuelle ne permettant pas sa réalisation comme convenu.

Contrainte Performance

Les contraintes de performance peuvent être provoquées par des objets tangibles et intangibles voir même humain par moment.

Les contraintes de performance se caractérisent et se provoquent par la manière dont la programmation à été effectuée, le langage employé, le Framework mis à disposition, les capacités de serveurs mis en place et d'autre.

Contrainte Financière

Les contraintes financières, comme sont l'indique provient essentiellement des revenus et des moyen pécuniaires mis à disposition pour le bon déroulement d'un projet.

Ce dernier peut alors faire effet domino et déclencher d'autres contraintes.

Cette dernière force par moment à des sacrifices d'option au sein du projet pour assurer la pérennité et l'aboutissement de ce dernier.

Contrainte liée à notre projet

Comme vu ci-dessus, on peut désormais analyser et prévoir par rapport à notre projet qu'elles sont les éventuelles contraintes qui peuvent nous impacter et influencer.

Cela va nous permettre d'identifier et de répondre plus efficacement aux futures contraintes que nous allons rencontrer durant le développement de notre application.

Dans notre projet, nous avons différents objectifs :

- ⇒ La création d'un moyen d'analyse des données via notre QR-Codes, nous serons fortement impactés par les contraintes Sécuritaire telles que la protection des données de nos clients, et la gestion des stockages de nos données devant être conforme aux normes imposées par la RGPD.
- ⇒ La création de condition d'utilisation sera alors obligatoire et primordiale.

Notre projet va rencontrer aussi des contraintes financières, en effet, en tant qu'étudiant nous ne possédons pas de grands moyens pécuniaires, ce qui risque de nous restreindre dans le choix d'hébergement, d'achat de licences de développement.

Cette dernière peut aussi entraîner des contraintes de performances, car sans réel moyen, nous ne pouvons garantir une performance pouvant réaliser avec des applications web ayant réalisé des levés de fond.

Etude des moyens et ressources disponible

Les ressources humaines

Le chef de projet

Le chef de projet sera le responsable de la planification, de la mise en œuvre et de la surveillance du projet. Il doit également gérer les ressources humaines, les délais et les budgets pour garantir la réalisation du projet. Il doit échanger avec toutes les entités qui gravitent autour du projet afin de s'assurer que tout le monde soit aligné avec les objectifs globaux.

Les développeurs Web

Les développeurs Web auront pour responsabilité le côté technique de l'application Web, c'est-à-dire la création et la maintenance de l'application.

Les ressources matérielles

Ordinateurs et équipements de développement

Les développeurs auront besoin d'ordinateurs et de logiciels de développement pour écrire et tester le code.

Serveurs et infrastructures

Il peut y avoir des besoins en matière de serveurs pour exécuter l'application web, stocker les données et gérer les utilisateurs.

Connexion internet

Une connexion internet stable et rapide sera nécessaire pour accéder aux différents outils et ressources nécessaires au développement et à l'exécution de l'application web.

Disques durs, stockage cloud

Il peut y avoir des besoins pour stocker les données et les versions du projet.

Les ressources financières

Ordinateurs et logiciels de développement

Les ordinateurs ainsi que les logiciels de développement sont essentiels pour l'exécution des tâches des développeurs, mais aussi du chef de projet. En effet, ça leur permettra de programmer, de suivre l'avancement du projet et communiquer au besoin.

Les coûts liés aux serveurs et à l'infrastructure

Il peut y avoir des coûts pour l'hébergement, la maintenance et la mise à niveau des serveurs utilisés pour exécuter l'application web.

Moyens nécessaires à la réalisation

Le projet web QRCoffee requiert divers éléments pour sa réalisation. Il est indispensable d'avoir un environnement de développement pour écrire et tester le code, un navigateur web pour vérifier le rendu et les interactions des composants React, ainsi qu'un serveur Node.js pour exécuter l'application Back-end. De plus, une base de données MongoDB est nécessaire pour stocker les données, et une interface d'authentification est indispensable pour gérer les utilisateurs et les sessions. De plus, il est important de planifier des outils et des processus pour effectuer des tests unitaires, des tests d'intégration et des tests de régression afin de garantir la qualité du code. Ces mesures contribueront à assurer la stabilité et la pérennité du projet à long terme.

Pour héberger le code et gérer les versions, l'utilisation d'un repository Bitbucket est prévue. De même, un tableau Jira et un Gantt serviront de système de gestion de projet pour planifier et suivre l'avancement du projet. Une connexion internet est essentielle pour accéder aux différents outils et ressources nécessaires. Enfin, des outils de développement seront employés afin de faciliter le processus de développement. De plus, pour assurer la sécurité des données, il est prévu de planifier des sauvegardes régulières et d'implémenter des stratégies de reprise après sinistre. Ces mesures préventives nous aideront à éviter toute perte de données en cas d'incident imprévu.

Il est essentiel de prévoir une documentation claire et exhaustive afin de faciliter la maintenance future et de permettre à d'autres personnes de comprendre et de contribuer au projet. Si le projet est destiné à être durable dans le temps, il sera nécessaire de mettre en place une équipe de support et de maintenance pour répondre aux questions des utilisateurs et assurer une maintenance continue.

En fonction des besoins spécifiques du projet, d'autres outils ou ressources pourraient également être requis.

Voici donc les compétences nécessaires à acquérir pour mener à bien le projet :

Langages de programmation : Maîtrise des langages de programmation utilisés, tels que JavaScript et TS(pour React et Node.js), HTML, CSS, et éventuellement d'autres langages selon les besoins spécifiques du projet.

Framework React : Une solide connaissance du framework React et TypeScript pour le développement de l'interface utilisateur du projet.

Base de données : Compétences dans la gestion et l'interaction avec une base de données MongoDB pour stocker et récupérer les données.

Backend : Compréhension du développement backend avec Node.js et TypeScript pour créer les API et les services nécessaires au fonctionnement de l'application. Clerk pour la gestion des utilisateurs, l'authentification et des autorisations.

Gestion de version : Utilisation d'un système de contrôle de version comme Git, ainsi que la maîtrise des pratiques de fusion, de gestion de branches, etc.

Conception web : Connaissances en conception web, y compris la mise en page, les styles, la responsivité, pour créer une interface utilisateur attrayante et fonctionnelle.

Gestion de projet : Une bonne compréhension des méthodologies de gestion de projet, comme Agile ou Scrum, pour planifier, organiser et suivre l'avancement du projet.

Communication : Une capacité à communiquer clairement et efficacement au sein de l'équipe et avec les parties prenantes du projet.

Documentation : Savoir créer une documentation complète et précise pour faciliter la maintenance future et l'intégration d'autres membres de l'équipe.

Résolution de problèmes : Être capable d'identifier et de résoudre les problèmes techniques qui pourraient survenir pendant le développement du projet.

Connaissance de React et TypeScript pour développer des composants et des interactions utilisateur en JavaScript. (Réalisation de l'interface utilisateur)

Connaissance de Node.js et TypeScript pour développer des applications Back-end performantes. (Réalisation de l'API/logique de l'application Web)

Etude de la transversalité

Matrice des risques

Tâche	Livrables	Risques	Incidences
Planification de l'application	Spécifications fonctionnelles	Exigences mal définis	Retard de développement
Développement de l'interface utilisateur	Code source	Erreurs de programmation	Mauvaise performance, Application difficile à comprendre
Développement de la base de données	Scripts SQL	Erreur de conception	Problème de sécurité, problème de performance, problème de stockage des données
Mise en place des tests	Scripts de test	Test manquants	Problème de qualité
Déploiement	Fichier de configuration	Erreurs de configuration	Problème de performance
Maintenance	Documentation technique	Documents manquants	Problème de maintenance

Evaluation de l'impact : Délai

Incidences	Impact sur le délai
Retard de développement	Augmentation du délai total du projet de xx jours
Mauvaise performance	Report du délai de livraison de xx jours
Problèmes de sécurité	Augmentation du délai de correction des bugs de xx jours

Evaluation de l'impact : Qualité

Incidences	Impact sur la qualité
Retard de développement	Augmentation du nombre de bugs de xx%
Mauvaise performance	Augmentation du taux de défaillance de xx%
Problèmes de sécurité	Diminution de la satisfaction des utilisateurs de xx%

Evaluation de l'impact : Coût

Incidences	Impact sur le coût
Retard de développement	Augmentation du coût total du projet de xx%
Mauvaise performance	Augmentation du coût par jour de retard de xx€
Problèmes de sécurité	Augmentation du coût des corrections de bugs de xx%

Mise en place des mesures préventives

Retard de développement

Pour gérer le risque de retard de développement, il est important d'organisation des réunions de clarification des exigences fonctionnelles avec les membres du projet. Cela nous permettra de s'assurer que les exigences sont bien comprises. Cela nous évitera des malentendus et des erreurs de développement.

- ⇒ Réunion toutes les 3 semaines + hebdomadaire si besoin.

De plus, il est important de mettre en place un système de suivi des tâches pour s'assurer que les tâches soient terminées à temps et identifier les tâches qui risquent de prendre plus de temps que prévu.

- ⇒ Mise en place d'un gant en début de projet et mis à jour tout au long du déroulement de celui-ci.

Mauvaise performance

Pour gérer le risque de mauvaise performance, nous devons mettre en place des tests unitaires pour détecter les problèmes potentiels.

De plus, nous devons le plus possible optimiser notre code pour l'amélioration de performance général de l'application.

Cela peut inclure :

- ⇒ Des utilisations de bibliothèques spécifiques.
- ⇒ L'utilisation d'un serveur performant.
- ⇒ La réutilisation de fonction pour éviter la duplication de code.

Problème de sécurité

Pour gérer le risque de problème de sécurité, nous devons mettre en place des politiques de sécurité pour protéger les données.

La mise en place des tests de sécurité pour détecter les vulnérabilités est donc essentielle.

Cela peut inclure :

- ⇒ Tests d'intrusions
- ⇒ Analyse de vulnérabilité
- ⇒ Audits de sécurité

En plus de la mise en place de tests, il est nécessaire de mettre en place des protocoles de sécurité.

Cela peut inclure :

- ⇒ Mise en place d'un système d'authentification.
- ⇒ Mise en place d'un système de rôle permettant l'autorisation ou non à certaines données.
- ⇒ Mise en place d'un protocole de sauvegarde et restauration des données.

Stack Technique

La stack technique qui sera utilisée pour le développement de QRCoffee sera :

Front-end - React + TypeScript (TSX)

Le front-end de l'application sera développé en utilisant React, un framework JavaScript populaire, couplé avec TypeScript (TSX) pour un développement à la fois efficace et sûr. React permettra de créer une interface utilisateur dynamique et interactive, facilitant ainsi une expérience utilisateur fluide et réactive. TypeScript, en tant que langage superset de JavaScript, renforcera la qualité du code en détectant les erreurs potentielles dès la phase de développement.

Back-end - Node.js + TypeScript (TSX)

Le back-end de notre application sera alimenté par Node.js, un environnement d'exécution JavaScript côté serveur reconnu pour sa rapidité et sa facilité de mise en œuvre. Combiné avec TypeScript (TSX), le développement du back-end sera plus structuré, offrant une maintenance facilitée et une meilleure collaboration au sein de l'équipe de développement. Node.js nous permettra également de gérer les requêtes de manière asynchrone, améliorant ainsi les performances globales de l'application.

Authentification - Clerks

La sécurité étant une priorité absolue, nous utiliserons le service d'authentification de Clerks pour gérer l'identification et l'autorisation des utilisateurs. Clerks est réputé pour son approche sécurisée, offrant des fonctionnalités d'authentification robustes telles que l'authentification à deux facteurs (2FA) et la gestion des sessions utilisateur. L'intégration avec notre application sera fluide et permettra de garantir la confidentialité des données de nos utilisateurs.

Base de données - MongoDB

Pour stocker et gérer les données de notre application, nous utiliserons MongoDB, une base de données NoSQL flexible et évolutive. MongoDB est particulièrement adapté aux applications web modernes grâce à sa capacité à gérer des données non structurées ou semi-structurées, offrant ainsi une grande souplesse lors du développement. Son modèle de données basé sur des documents permettra de stocker les informations de manière efficace, tout en facilitant les opérations de lecture et d'écriture. L'utilisation de MongoDB nous permettra de mettre en œuvre des fonctionnalités évolutives et performantes pour notre application, tout en garantissant une réactivité optimale aux besoins de nos utilisateurs.

Contrôle de Version - Git

Pour garantir une gestion efficace du code source et faciliter la collaboration entre les membres de l'équipe de développement, nous utiliserons Git comme système de contrôle de version. Git permettra de suivre l'évolution du code, d'enregistrer les modifications apportées à chaque fichier et de revenir à des versions antérieures si nécessaire. L'utilisation de Git favorisera également la création de branches de développement pour travailler sur des fonctionnalités spécifiques de manière isolée, minimisant ainsi les conflits de code. Grâce à Git, nous pourrons également assurer une meilleure traçabilité des modifications apportées au code source, ce qui contribuera à maintenir un code propre et bien organisé tout au long du cycle de développement du projet.

Hébergement du Code - Bitbucket

Pour héberger notre code source de manière sécurisée et faciliter la collaboration entre les membres de l'équipe, nous utiliserons Bitbucket. Bitbucket est une plateforme d'hébergement de code qui offre des fonctionnalités avancées de gestion de dépôts Git. Nous pourrons créer des dépôts privés pour protéger notre

code tout en permettant un accès contrôlé aux développeurs concernés. De plus, Bitbucket fournit des outils de suivi des problèmes (issue tracking) et de gestion des demandes de fusion (pull requests), ce qui facilitera la communication et la revue de code entre les membres de l'équipe. Grâce à Bitbucket, nous pourrions centraliser notre code source et profiter d'un environnement de développement collaboratif efficace pour assurer le succès de notre projet.

Tests unitaires sur le Front-end avec Cypress:

Afin de garantir la qualité et la fiabilité du Front-end de notre application développée en React et TypeScript (TSX), nous mettrons en place des tests unitaires à l'aide de Cypress, un puissant framework de test End-to-End. Les tests unitaires nous permettront de vérifier le bon fonctionnement des composants React et d'identifier rapidement d'éventuelles erreurs de logique ou de comportement. Grâce à TypeScript, nous bénéficierons d'un typage fort qui détectera les erreurs potentielles dès la phase de développement, renforçant ainsi la robustesse du code. Cypress facilitera la mise en œuvre de ces tests en offrant une syntaxe simple et expressive, permettant de simuler les interactions utilisateur, comme le clic sur des boutons, la saisie dans des champs de formulaire, et la vérification des éléments visuels. Ces tests unitaires automatisés garantiront une maintenance plus aisée du code et une meilleure réactivité aux éventuels changements futurs, assurant ainsi une expérience utilisateur fluide et sans faille.

Commande de lancement : ``npm run test``

Tests unitaires sur le Back-End avec Mocha et Chai

Mocha sera notre choix de framework de test. Il offrira un environnement bien structuré et hautement personnalisable. Cela nous permettra de définir des scénarios de test avec précision, de manière à couvrir toutes les fonctionnalités et les aspects de notre backend. Sa flexibilité nous permettra de créer des suites de tests claires et organisées, simplifiant ainsi la gestion de notre suite de tests au fil du temps.

Chai, de son côté, sera notre allié pour les assertions. Sa syntaxe expressive et intuitive rendra la rédaction et la lecture des tests plus fluides. Avec Chai, nous pourrions formuler des attentes claires sur le comportement de notre code, ce qui facilitera la vérification des résultats attendus.

En combinant Mocha et Chai, nous exploiterons la puissance de ces deux outils pour renforcer notre processus de développement. Les tests unitaires que nous mettrons en place nous permettront d'identifier rapidement les erreurs, d'assurer la stabilité du code et de prévenir les régressions à mesure que nous apporterons des modifications et des améliorations.

Commande de lancement : ``npm run test``

Enveloppe Budgétaire à prévoir

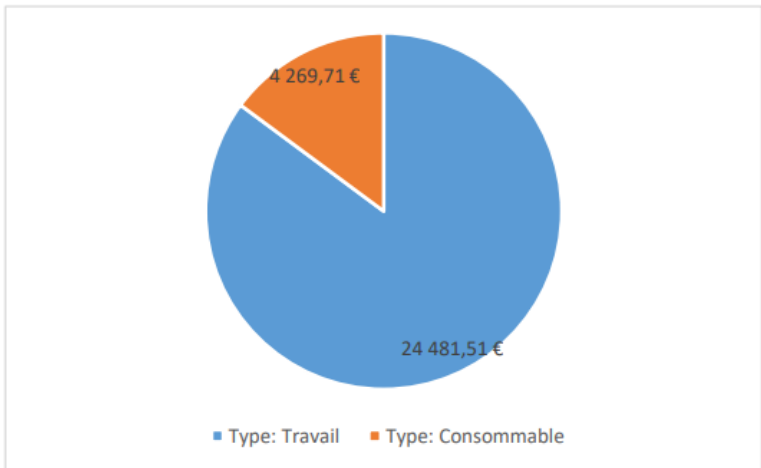
DÉTAILS DES COÛTS

Détails des coûts pour toutes les ressources de travail.

Nom	Travail réel	Coût réel	Taux standard	Coût
Bertrand Julien	548 h	11 880,00 €	22,00 €/h	12 232,00 €
Grégoire Ivars	256,2 h	4 926,73 €	19,23 €/h	5 711,31 €
Kevin BEGOT	328 h	6 153,60 €	19,23 €/h	6 538,20 €
PC 1	0,94	843,83 €	900,00 €	900,00 €
PC 2	0,94	750,07 €	800,00 €	800,00 €
PC 3	0,94	2 250,20 €	2 400,00 €	2 400,00 €
Licences outils	0,94	0,00 €	0,00 €	0,00 €
VPS	8	48,00 €	6,00 €	48,00 €
Hébergement Web : Pack Web qrcoffee.fr	0,94	64,69 €	69,00 €	69,00 €
Service Mail : Pack email qrcoffee.fr	0,94	49,42 €	52,71 €	52,71 €

RÉPARTITION DES COÛTS

Répartition des coûts parmi les différents types de ressources.



Ce tableau présente les principales dépenses mensuelles associées à notre application web. Certaines catégories telles que les licences logicielles et l'hébergement web, sont actuellement gratuites grâce à notre licence étudiante. Nous assumons également le développement supplémentaire, la maintenance et le support, ainsi que les activités de marketing et de promotion en interne, sans coûts supplémentaires.

Cependant, le VPS constitue actuellement notre principal coût mensuel, s'élevant à 6 euros. Il est important de garder à l'esprit que ce montant peut varier en fonction de nos besoins en ressources et de l'évolution de notre application.

Le coût des développeurs est une estimation étant donné que nous sommes tous associés, nous ne nous versons pas de salaire. Nous nous rémunérerons lorsque l'application générera de l'argent grâce au pub et abonnement.

Financement du projet

Afin de pouvoir payer l'infrastructure du projet, nous proposons un service de génération de QR code sur le site ComeUp. Ce service est tout simple : un utilisateur se rend sur notre page dédiée où il peut commander jusqu'à 20 QR codes en une seule fois. Une fois la commande passée, notre équipe dévouée se met immédiatement au travail pour réaliser les QR codes selon les spécifications de l'utilisateur.

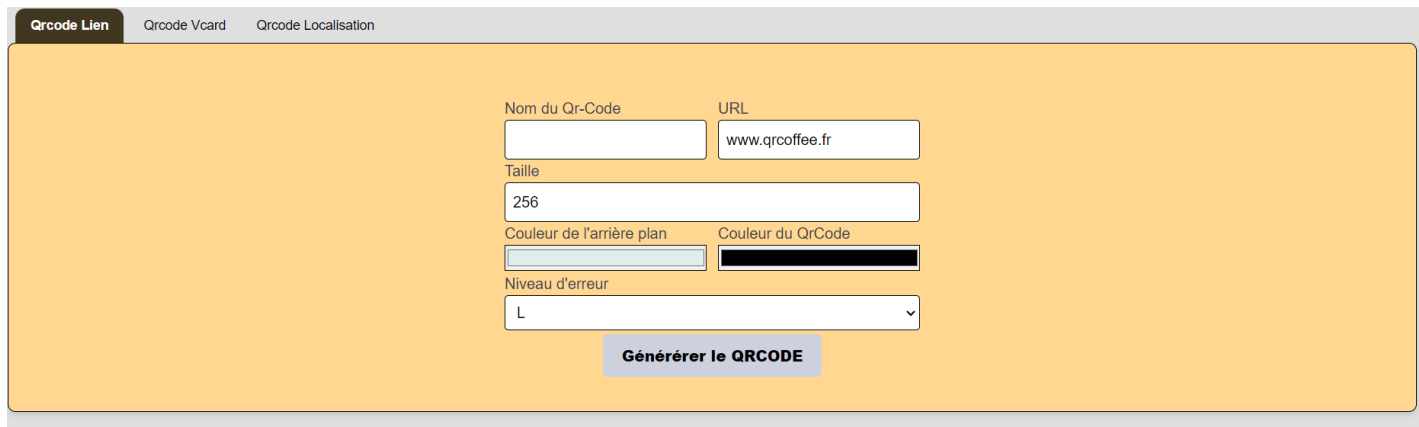
Notre processus de création est rapide et efficace. Nous nous assurons que chaque QR code généré répond aux exigences de nos clients, qu'il s'agisse de promouvoir leur entreprise, de partager des informations importantes ou de faciliter les interactions avec leur public.

Grâce à notre approche orientée client et à notre engagement envers la qualité, nous avons réussi à attirer de nombreux utilisateurs satisfaits. Le bouche-à-oreille et les retours positifs de nos clients nous ont permis de générer un chiffre d'affaires conséquent, nous permettant ainsi de payer l'infrastructure tout au long de l'année 2023.

Evolution de l'application

Cours terme :

Dans un premier temps, il serait bon d'implémenter de nouvelles fonctionnalités tel que l'ajout supplémentaire de QRCode possible.



The screenshot shows a web interface for generating a QR code. At the top, there are three tabs: 'Qrcode Lien' (selected), 'Qrcode Vcard', and 'Qrcode Localisation'. The main area is orange and contains the following fields:

- Nom du Qr-Code**: An empty text input field.
- URL**: A text input field containing 'www.qrcoffee.fr'.
- Taille**: A text input field containing '256'.
- Couleur de l'arrière plan**: A color selection box showing a light blue swatch.
- Couleur du QrCode**: A color selection box showing a black swatch.
- Niveau d'erreur**: A dropdown menu with 'L' selected.

At the bottom center, there is a button labeled 'Générer le QRCODE'.

En effet, nous ne proposons que trois types de QRCode :

- ⇒ Les QRCode possédant un lien brut.
- ⇒ Les QRCode Vcard permettant l'ajout rapide de contact.
- ⇒ Les QRCode Localisation.

Il serait bon d'implémenter des types de QRcodes supplémentaire : (La liste est non-exhaustive)

- ⇒ QRCode de compte PayPal
- ⇒ QRCode de chaine YouTube
- ⇒ QRCode de compte Facebook
- ⇒ QRCode de Twitter
- ⇒ QRCode de WIFI
- ⇒ QRCode d'évènement
- ⇒ QRCode de SMS

Moyen terme

Dans un moyen terme, c'est-à-dire dans les 6 à 12 mois, il serait pertinent d'implémenter :

➔ Une fonction de suivi des QRCode.

Ce « tracking » permettra de suivre l'évolution des QRcode générés. En effet, le but étant que l'utilisateur puissent connaître :

- ⇒ En temps différé ou réelle le nombre de fois où sont QRcode a été scanner
- ⇒ Dans quelle zone géographique.
- ⇒ De connaître les types de profils à trouver et utiliser son QRCode.

Ces rapports se ferai sou forme de liste et/ou de graphique sur des échelles de temps allant de 1 jour à plusieurs mois.

Les clients potentiels seraient en grande partie des professionnels dans le but de collecter des données marketing. Il y aurait aussi des particulier qui souhaitent suivre l'évolution de leur QRcode Facebook par exemple.

→ Une refonte graphique est à prévoir afin d'y incorporer les règles RGAA. En effet, le respect de ces règles est devenu une norme à respecter afin que notre application Web puisse être accessible à tous.

Long terme

Au long terme, il serait bon d'implémenter des fonctionnalités financières comme :

- ⇒ Ajout de publicité :
 - Sur l'application web.
 - Lors de la génération de QRCode.
- ⇒ Revente de données utilisateurs dans le respect de la loi.
- ⇒ Ajout de la possibilité de réaliser des dons pour soutenir l'application web.
- ⇒ Un service d'abonnement premium offrant des possibilités de :
 - Générer des types de QRCode différent.
 - Augmenter le nombre maximal de QRCode pouvant être généré.
 - L'accès au service de suivi des QRCode.
 - Suppression des publicités.

En conclusion, à court terme, l'ajout de nouvelles fonctionnalités telles que la possibilité de générer différents types de QRCode, comme des QRCode PayPal, des QRCode de chaînes YouTube, des QRCode de comptes Facebook, des QRCode de lieux géographiques, des QRCode de comptes Twitter, des QRCode de réseaux Wi-Fi, des QRCode d'événements et des QRCode de SMS, serait bénéfique pour les utilisateurs.

À moyen terme, la mise en place d'une fonction de suivi des QRCode serait pertinente. Cela permettrait aux utilisateurs de suivre en temps différé ou en temps réel le nombre de scans de leurs QRCode, la localisation géographique des scans et les profils des utilisateurs qui ont scanné les QRCode. Ces informations seraient présentées sous forme de listes et/ou de graphiques sur différentes échelles de temps.

À long terme, il serait intéressant d'envisager des fonctionnalités financières telles que l'ajout de publicités sur l'application web et lors de la génération de QRCode, la possibilité de revendre des données utilisateurs dans le respect de la loi, la mise en place d'un système de dons pour soutenir l'application web, et la création d'un service d'abonnement premium offrant des avantages tels que la génération de types de QRCode différents, l'augmentation du nombre maximal de QRCode pouvant être généré, l'accès au service de suivi des QRCode et la suppression des publicités.

En mettant en œuvre ces différentes mesures, l'application de génération de QRCode pourra offrir une plus grande variété de fonctionnalités aux utilisateurs, tout en offrant des opportunités de revenus supplémentaires à travers la publicité, la revente de données et les abonnements premium.

Rapport du projet

Présentation des outils de gestion utilisés

Project

Introduction :

Pour le suivi du projet tout au long de l'année, nous avons décidé d'utiliser Microsoft Project qui est un outil d'aide à la planification de projet.

Avec cet outil, nous avons défini les ressources (humaines et matérielles), les tâches, les coûts du projet.

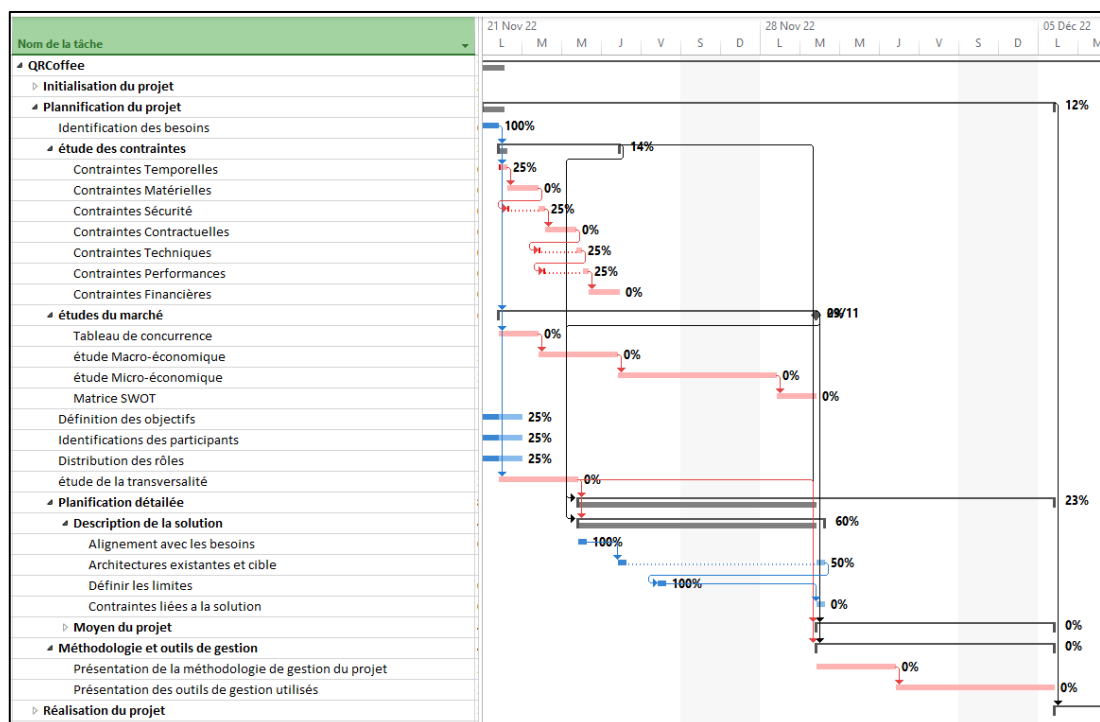
Nom de la ressource	Type	Étiquette Matériau	Initiales	Groupe	Unités max	Tx. standard	Tx. hrs. sup.	Coût/Utili	Allocation	Calendrier de base
Bertrand Julien	Travail		B		100%	22,00 €/h	0,00 €/h	0,00 €	Proportion	Standard
Grégoire Ivars	Travail		G		100%	19,23 €/h	0,00 €/h	0,00 €	Proportion	Standard
Kevin BEGOT	Travail		K		100%	19,23 €/h	0,00 €/h	0,00 €	Proportion	Standard
PC 1	Consommable		P			900,00 €		0,00 €	Proportion	
PC 2	Consommable		P			800,00 €		0,00 €	Proportion	
PC 3	Consommable		P			2 400,00 €		0,00 €	Proportion	
Licences outils	Consommable		L			0,00 €		0,00 €	Proportion	
VPS	Consommable		V			6,00 €		0,00 €	Proportion	
Hébergement Web : Pack Web qrcode.fr	Consommable		H			69,00 €		0,00 €	Proportion	
Service Mail : Pack email qrcode.fr	Consommable		S			52,71 €		0,00 €	Proportion	

Nom de la tâche	Durée	Début	Fin	Prédécesseurs
QRCoffee	70,25 jrs	Lun 17/10/22	Lun 23/01/23	
Initialisation du projet	25 jrs	Lun 17/10/22	Ven 18/11/22	
Proposition du projet	4,8 sem.	Lun 17/10/22	Jeu 17/11/22	
Approbation du projet	1 jr	Ven 18/11/22	Ven 18/11/22	3
Plannification du projet	10,25 jrs	Lun 21/11/22	Lun 05/12/22	2
Identification des besoins	0,25 jr	Lun 21/11/22	Lun 21/11/22	
étude des contraintes	3,13 jrs	Lun 21/11/22	Jeu 24/11/22	6
Contraintes Temporelles	0,5 jr	Lun 21/11/22	Lun 21/11/22	6
Contraintes Matérielles	0,5 jr	Lun 21/11/22	Mar 22/11/22	8
Contraintes Sécurité	0,5 jr	Lun 21/11/22	Mar 22/11/22	9
Contraintes Contractuelles	0,5 jr	Mar 22/11/22	Mer 23/11/22	10
Contraintes Techniques	0,5 jr	Mar 22/11/22	Mer 23/11/22	11
Contraintes Performances	0,5 jr	Mar 22/11/22	Mer 23/11/22	12
Contraintes Financières	0,5 jr	Mer 23/11/22	Jeu 24/11/22	13
études du marché	6 jrs	Lun 21/11/22	Mar 29/11/22	6
Tableau de concurrence	1 jr	Lun 21/11/22	Mar 22/11/22	6
étude Macro-économique	2 jrs	Mar 22/11/22	Jeu 24/11/22	16
étude Micro-économique	2 jrs	Jeu 24/11/22	Lun 28/11/22	17
Matrice SWOT	1 jr	Lun 28/11/22	Mar 29/11/22	18
Définition des objectifs	1 jr	Lun 21/11/22	Lun 21/11/22	4
Identifications des participants	1 jr	Lun 21/11/22	Lun 21/11/22	4
Distribution des rôles	1 jr	Lun 21/11/22	Lun 21/11/22	4
étude de la transversalité	2 jrs	Lun 21/11/22	Mer 23/11/22	6
Planification détaillée	8 jrs	Mer 23/11/22	Lun 05/12/22	7;15;23
Description de la solution	4,5 jrs	Mer 23/11/22	Mar 29/11/22	7;15;23
Alignement avec les besoins	0,5 jr	Mer 23/11/22	Mer 23/11/22	
Architectures existantes et cible	1 jr	Jeu 24/11/22	Mar 29/11/22	26
Définir les limites	0,5 jr	Ven 25/11/22	Ven 25/11/22	27
Contraintes liées a la solution	0,5 jr	Mar 29/11/22	Mar 29/11/22	28
Moyen du projet	4 jrs	Mar 29/11/22	Lun 05/12/22	7;15;23
Méthodologie et outils de gestion	4 jrs	Mar 29/11/22	Lun 05/12/22	7;15;23
Présentation de la méthodologie de gestion du projet	2 jrs	Mar 29/11/22	Jeu 01/12/22	
Présentation des outils de gestion utilisés	2 jrs	Jeu 01/12/22	Lun 05/12/22	36

➤ Réalisation du projet	35 jrs	Lun 05/12/22	Lun 23/01/23	5
➤ Réalisation du front	17 jrs	Lun 05/12/22	Mer 28/12/22	5
Mise en place de la page Accueil	1 jr	Lun 05/12/22	Mar 06/12/22	5
Mise en place de la page Compte utilisateur	1 jr	Mar 06/12/22	Mer 07/12/22	40
Mise en place de la page de création de QRCode	2 jrs	Mer 07/12/22	Ven 09/12/22	41
Mise en place de la page Rapport	3 jrs	Ven 09/12/22	Mer 14/12/22	42
Mise en place de la page service	5 jrs	Mer 14/12/22	Mer 21/12/22	43
Mise en place de la page QRCode Créé	2 jrs	Mer 21/12/22	Ven 23/12/22	44
Réalisation du Header	1 jr	Ven 23/12/22	Lun 26/12/22	45
Réalisation du side Menu	1 jr	Lun 26/12/22	Mar 27/12/22	46
Réalisation du footer	1 jr	Mar 27/12/22	Mer 28/12/22	47
➤ Réalisation du Back	11 jrs	Lun 05/12/22	Mar 20/12/22	5
➤ Mise en place de l'API	11 jrs	Lun 05/12/22	Mar 20/12/22	5
➤ Mise en place de la base de donnée	1 jr	Lun 05/12/22	Mar 06/12/22	5
Dimensionnement de la Base de donnée	1 jr	Lun 05/12/22	Mar 06/12/22	5
Mise en place de l'authentification	5 jrs	Mar 13/12/22	Mar 20/12/22	54
Création des Routes	5 jrs	Mar 06/12/22	Mar 13/12/22	52
Sécurisation des Routes	2 jrs	Mar 13/12/22	Jeu 15/12/22	54
Test / Validation	2 jrs	Mer 28/12/22	Ven 30/12/22	39;49
➤ Rédaction des livrables	10 jrs	Ven 30/12/22	Ven 13/01/23	56
Cahier de recettes	2 jrs	Ven 30/12/22	Mar 03/01/23	56
Cahier des charges	10 jrs	Ven 30/12/22	Ven 13/01/23	56
Documentation technique	5 jrs	Ven 30/12/22	Ven 06/01/23	56
Livraison / Remise	5 jrs	Ven 13/01/23	Ven 20/01/23	57
Cloture	1 jr	Ven 20/01/23	Lun 23/01/23	61

Gant :

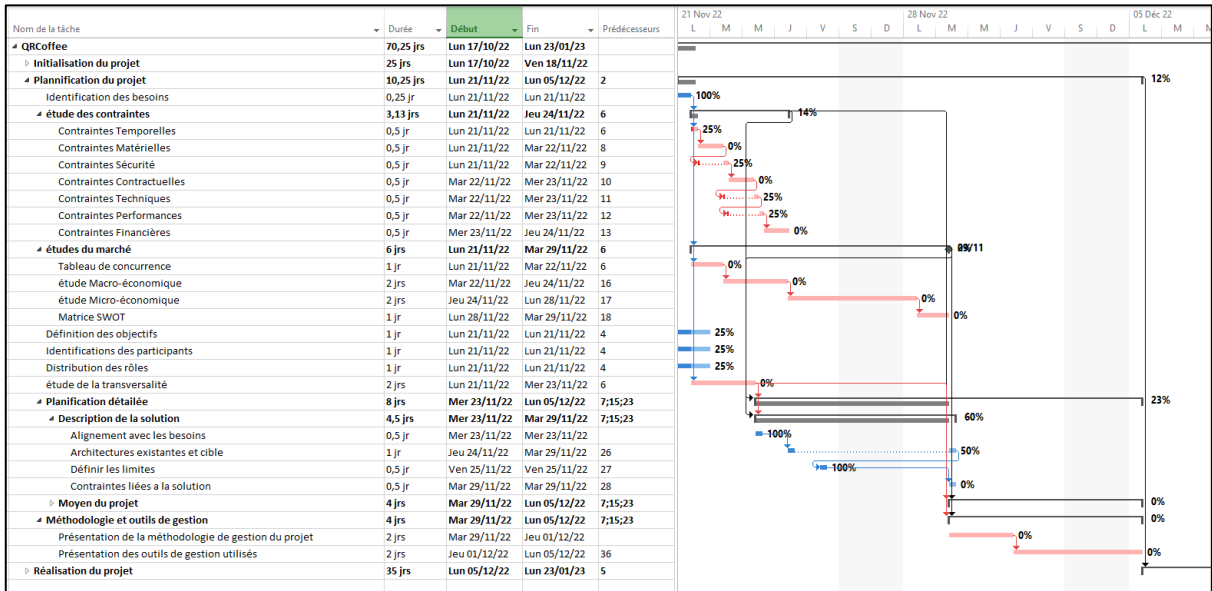
Avec Project, nous avons pu établir un gant qui sera notre fil conducteur tout au long du projet. En effet, c'est avec celui-ci que nous pouvons voir où nous en sommes sur l'avancement des tâches.



Dans l'exemple ci-dessus, nous en sommes à la « Planification du projet ».

Comme nous pouvons le voir, nous avons réalisé 12% (au moment de la capture d'écran) de la tâche « Planification du projet ». Grâce à cette vision, nous pouvons voir où nous en sommes sur l'avancement du projet.

Chaque sous-tâche s'est vu affecter une durée théorique permettant la génération du gant. Afin que le Gant soit des plus cohérent, nous avons ajouté à chaque tâche un ou plusieurs prédécesseurs afin que celle-ci soit débloqué une fois que la/les précédentes soient réalisés.



Les rapports :

En plus de généré un Gant, Project nous permet si nous le souhaitons la génération de rapports personnalisable :

PROJECT OVERVIEW

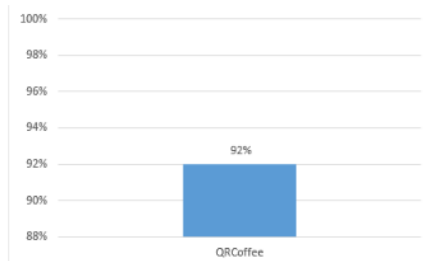
MAR 04/10/22 - MAR 01/08/23

% ACHÉVÉ

92%

Nom	Début	Fin	Durée	% achevé
QRCoffee	Mar 04/10/22	Mar 01/08/23	201 jrs	92%
Initialisation du projet	Ven 21/10/22	Ven 18/11/22	2,5 jrs	100%
Proposition du projet	Ven 21/10/22	Ven 21/10/22	0,5 jr	100%
Approbation du projet	Ven 18/11/22	Ven 18/11/22	0,5 jr	100%
Planification du projet	Sam 19/11/22	Lun 06/03/23	80 jrs	100%
Identification des besoins	Sam 19/11/22	Lun 21/11/22	2 jrs	100%
étude des contraintes	Lun 21/11/22	Jeu 24/11/22	3,38 jrs	100%
Contraintes Temporelles	Lun 21/11/22	Lun 21/11/22	0,5 jr	100%
Contraintes Matérielles	Lun 21/11/22	Mer 22/11/22	0,5 jr	100%
Contraintes Sécurité	Lun 21/11/22	Mer 22/11/22	0,5 jr	100%
Contraintes Contractuelles	Mar 22/11/22	Mer 23/11/22	0,5 jr	100%
Contraintes Techniques	Mar 22/11/22	Mer 23/11/22	0,5 jr	100%
Contraintes Performances	Mar 22/11/22	Mer 23/11/22	0,5 jr	100%
Contraintes Financières	Mer 23/11/22	Jeu 24/11/22	0,5 jr	100%
études du marché	Mer 21/12/22	Lun 06/03/23	57 jrs	100%
Tableau de concurrence	Mer 21/12/22	Mer 21/12/22	1 jr	100%
étude Macro-économique	Sam 04/03/23	Lun 06/03/23	0,5 jr	100%
étude Micro-économique	Dim 05/03/23	Lun 06/03/23	0,5 jr	100%
Matrice SWOT	Jeu 22/12/22	Jeu 22/12/22	1 jr	100%
Définition des objectifs	Ven 23/12/22	Ven 23/12/22	1 jr	100%
Identifications des participants	Sam 24/12/22	Lun 26/12/22	1 jr	100%
Distribution des rôles	Lun 26/12/22	Lun 26/12/22	1 jr	100%
étude de la transversalité	Mar 27/12/22	Mer 28/12/22	2 jrs	100%
Planification détaillée	Mer 23/11/22	Jeu 29/12/22	26,35 jrs	100%

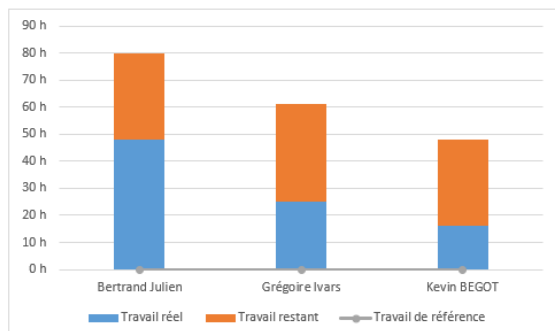
% COMPLETE



RESOURCE OVERVIEW

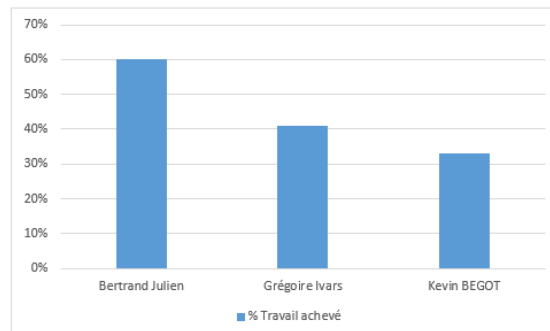
RESOURCE STATS

Work status for all work resources.



WORK STATUS

% work done by all the work resources.



RESOURCE STATUS

Remaining work for all work resources.

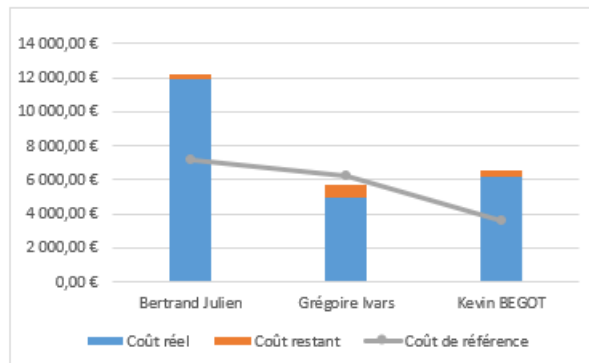
Nom	Début	Fin	Travail restant
Bertrand Julien	Lun 21/11/22	Lun 05/12/22	32 h
Grégoire Ivars	Lun 21/11/22	Mar 29/11/22	36 h
Kevin BEGOT	Lun 21/11/22	Lun 05/12/22	32 h

Ces rapports une fois mis en place reflètent les dernières informations du projet sans aucune mise à jour de notre part. C'est un gain de temps lors des présentations d'avancement du projet car nous n'avons pas besoin de tout remettre à jour.

Les rapports nous ont permis de surveiller les coûts du projet :

ÉTAT DES COÛTS

État des coûts pour les ressources de travail.



DÉTAILS DES COÛTS

Détails des coûts pour toutes les ressources de travail.

Nom	Travail réel	Coût réel	Taux standard	Coût
Bertrand Julien	548 h	11 880,00 €	22,00 €/h	12 232,00 €
Grégoire Ivars	256,2 h	4 926,73 €	19,23 €/h	5 711,31 €
Kevin BEGOT	328 h	6 153,60 €	19,23 €/h	6 538,20 €
PC 1	0,94	843,83 €	900,00 €	900,00 €
PC 2	0,94	750,07 €	800,00 €	800,00 €
PC 3	0,94	2 250,20 €	2 400,00 €	2 400,00 €
Licences outils	0,94	0,00 €	0,00 €	0,00 €
VPS	8	48,00 €	6,00 €	48,00 €
Hébergement Web : Pack Web qrcoffee.fr	0,94	64,69 €	69,00 €	69,00 €
Service Mail : Pack email qrcoffee.fr	0,94	49,42 €	52,71 €	52,71 €

Rapport synthétique du coût des ressources en €



En effet, nous pouvons surveiller le coût par développeur ainsi que par ressource pour mieux gérer notre budget et optimiser nos dépenses. Actuellement, les développeurs ont un coût par heure de 19,23€, ce qui nous permet de suivre leur contribution financière à nos projets. Le chef de projet quant à lui est à 22€ de l'heure car il gère aussi le projet dans sa globalité.

Cependant, il est essentiel de ne pas négliger le coût supplémentaire lié à l'utilisation des VPS. Les VPS sont des serveurs virtuels privés qui jouent un rôle crucial dans le développement et le déploiement de nos applications. Le prix du VPS varie généralement en fonction de ses spécifications, de la puissance de calcul, de la mémoire et du stockage alloués.

Atlassian

Introduction :

Atlassian est un éditeur de logiciels qui développe des produits pour la gestion de projet. La force d'Atlassian réside dans son écosystème. En effet, chacune de ses solutions peuvent interagir avec les autres. Cela est bien utile lorsque nous développons un projet.

Choix des solutions :

Nous avons choisi de travailler avec deux des solutions d'Atlassian :

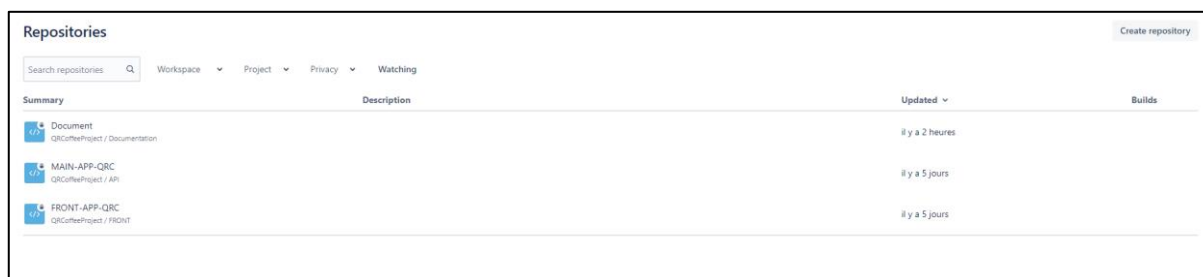
- ⇒ Bitbucket
- ⇒ Jira Software

Bitbucket

Bitbucket est une plateforme collaborative basée sur Git, spécialement conçue pour les équipes de développement. Ses intégrations avec Jira offrent désormais une performance optimale, permettant ainsi de réunir efficacement toute l'équipe autour de projets communs.

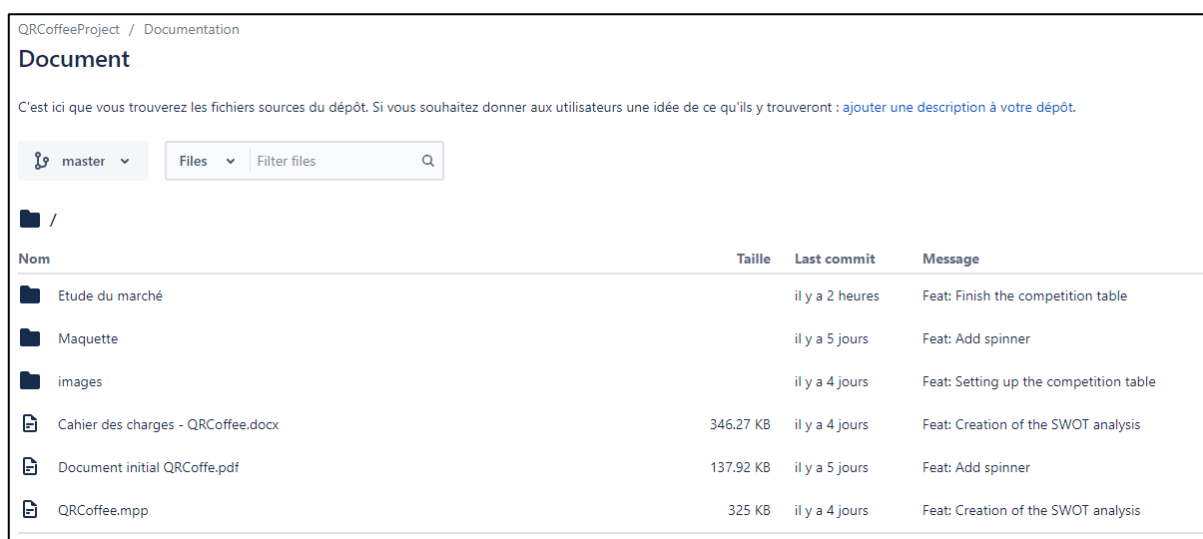
Avec Bitbucket, nous allons stocker notre code source ainsi que nos documents liés au projet.

Nous avons défini l'architecture comme tel :



Document :

Ici, nous allons stocker les documents. (PDF, images, rapports, gant...)



MAIN-APP-QRC :

Ici, nous allons stocker le code source du Back-End de notre application.

FRONT-APP-QRC :

Ici, nous allons stocker le code source du Front-End de notre application.

Intégration continue et gestion de version

Dans le cadre de notre projet de groupe, nous avons pris une décision cruciale pour garantir le bon déroulement de notre développement logiciel : adopter l'intégration continue. Cette approche, fondamentale dans les méthodes de travail modernes, nous permet d'assurer une cohésion et une qualité constantes de notre code tout au long du processus de développement.

Pour gérer notre code, nous utilisons Bitbucket comme gestionnaire de dépôt et Git comme système de contrôle de version. Cette combinaison nous offre une excellente plateforme pour collaborer efficacement et maintenir un historique complet de nos modifications.

Dans notre infrastructure de gestion de code, nous avons créé deux branches distinctes pour notre projet : "develop" et "main". Chacune de ces branches joue un rôle spécifique dans notre flux de travail.

La branche "develop" est au cœur de nos activités quotidiennes. C'est ici que chaque membre du groupe travaille sur les fonctionnalités, les améliorations et les corrections. Chacun développe ses modifications à partir de cette branche et les soumet ensuite en tant que demande de fusion.

Une fois que la demande de fusion a été approuvée, nous sommes prêts à fusionner les modifications dans la branche "main".

La branche "main" représente la version stable et déployée de notre application. Lorsque nous sommes prêts à publier une nouvelle version, nous fusionnons les changements de la branche "develop" dans "main" et lançons une nouvelle version stable.

Cette approche nous permet de travailler de manière itérative, d'assurer la cohésion de notre code et de minimiser les risques liés aux conflits lors des intégrations. En outre, l'utilisation de Bitbucket comme gestionnaire de dépôt et Git comme système de contrôle de version nous offre une collaboration fluide et une gestion efficace des versions de notre application.

Structuration des Commits avec Commitizen

Au sein de notre équipe de développement, nous avons adopté une approche structurée pour nos commits en utilisant Commitizen. Commitizen est un outil qui facilite la rédaction de messages de commit clairs et bien définis, conformes aux conventions établies par notre projet.

L'objectif principal de cette démarche est d'améliorer la lisibilité et la compréhension de l'historique des modifications de notre code source. En structurant nos commits, nous rendons le processus de revue de code plus efficace et facilitons l'identification rapide des modifications effectuées sur une fonctionnalité spécifique ou une résolution de bogue particulière.

Exemple de commits :

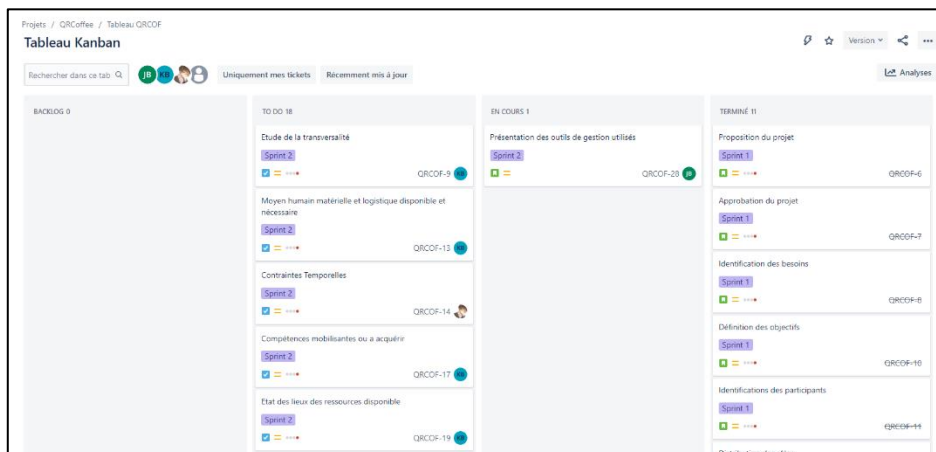
- ⇒ Feat(QRCOF-65) : Implementation QRCode Localisation
- ⇒ Fix(QRCOF-65) : Fix de la génération du QRCode Localisation

Jira Software

Jira Software est un outil de planification, de suivi et de génération de rapports sur le travail. Avec cette solution, nous pouvons créer des « Sprints ». Ces Sprints (de 1 mois) vont nous permettre de nous donner des objectifs à réaliser.

Nous parlerons alors de ticket.

Ces tickets vont être associé à une tâche spécifique du gant. Lors de chaque Sprint, nous allons créer des tickets en lien avec les tâches à réaliser.



Nous avons défini le tableau Kanban en 4 colonnes :

La colonne Backlog :

Cette colonne permet de stockage des tickets qui n'ont pas encore été affectés.

Nous créons les tickets de chaque sprint juste avant que celui-ci ne commence. Cette colonne est vide 90% du temps

La colonne « To Do » :

Cette colonne est celle qui indique les tickets à réaliser durant le sprint.

Chaque membre de l'équipe peut filtrer les tickets sur ceux qui lui sont affectés. Il verra alors l'ensemble des tickets qu'il doit réaliser.

La colonne « En cours » :

Cette colonne indique les tickets qui sont en cours de réalisation par quelqu'un de l'équipe.

Par exemple, nous voyons que le ticket QRCOF-28 (c'est le ticket pour rédiger cette partie du cahier des charges) est en cours de réalisation.

La colonne « Terminé » :

Cette colonne indique les tickets terminés. Dès qu'un membre de l'équipe a terminer son ticket, il le passe dans la colonne.

Par exemple, les tickets du sprint 1 sont tous dans la colonne « terminé »

Jira permet aussi d'associer ces tickets créés à des branches de développement. Pour cela, le développeur va devoir :

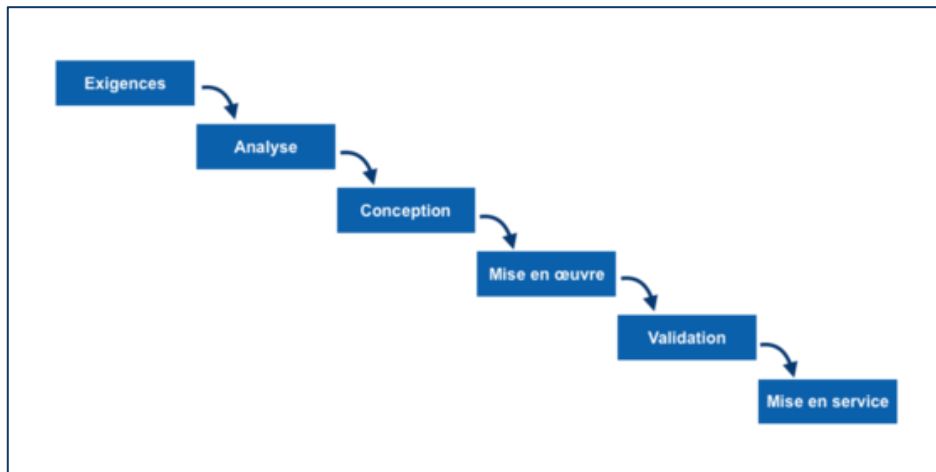
- ⇒ Fork le projet.
- ⇒ Créer une branche.
- ⇒ Réaliser son développement sur cette branche.
- ⇒ Push le commit de cette branche sur le répertoire distant.
- ⇒ Créer un pull Request

Méthodologie de gestion de projet

La méthode en cascade + Kanban

Pour mener à bien notre projet, nous avons choisis d'utiliser la méthode « en cascade ». C'est un enchaînement de tâches. Lorsque qu'une tâche est terminée, la suivante est déclenchée :

- ⇒ Analyse du besoin
- ⇒ Conception
- ⇒ Planification
- ⇒ Vérification
- ⇒ Mise en production
- ⇒ Maintenance



Avantage de la méthode en cascade :

- ⇒ Les délais sont connus de tous
- ⇒ Les tâches se succèdent les unes après les autres
- ⇒ Le suivi du projet est linéaire.
- ⇒ Requiert peu de personnel.
- ⇒ Adapté au petit projet (site Web)

Cependant, cette méthode possède des inconvénients :

Très peu flexible (s'il y a un problème, nous recommençons tout)

Ne convient pas au moyen et gros projets.

Pour essayer de contrebalancer les inconvénients, nous utilisons aussi la méthode Kanban. Introduite chez Toyota dans les années 1950, la méthode Kanban vise à améliorer la production en se basant sur le juste temps et sur un processus évolutif.

Elle fonctionne avec un système de ticket et permet de se situer sur l'avancement des différentes tâches du sprint en cours.

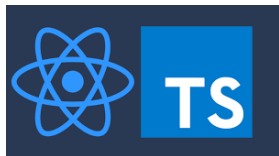
Les tâches (tickets) sont réparties en colonnes. Pour notre projet, nous avons créé 4 colonnes :

- ⇒ Backlog
- ⇒ To Do
- ⇒ En cours
- ⇒ Terminé

Architecture de l'application

Notre application web est conçue avec une architecture moderne pour répondre aux besoins actuels et futurs de nos utilisateurs. Nous avons choisi des technologies réputées pour garantir une expérience utilisateur fluide, des performances élevées et une sécurité renforcée.

Architecture FrontOffice



Pour le Front, nous avons décidé d'utiliser la librairie React avec TypeScript. A cela, nous avons choisi d'utiliser le Framework CSS Tailwind.

Le gros avantage d'utiliser TypeScript par rapport à Javascript est de pouvoir typer fortement nos variables. Cela permet de rendre notre application moins sensible aux bugs.

L'utilisation du Framework CSS Tailwind va nous permettre de faciliter la conception des pages web de notre application.

Organisation du Front

Notre code repose sur une structure en couches, où chaque couche joue un rôle bien défini dans le flux de traitement des requêtes. Voici un aperçu des différentes composantes de notre architecture :

Dossier Composant : Ce dossier regroupe les composants React réutilisables de notre application. Chaque composant est conçu pour être autonome et modulaire, facilitant ainsi la construction de l'interface utilisateur. Cette approche favorise la réutilisation du code et simplifie la maintenance à long terme.

Dossier Composant en commun : Dans ce dossier, nous stockons les composants qui sont partagés entre plusieurs fonctionnalités de notre application. Il s'agit de composants génériques qui peuvent être utilisés dans différentes parties de l'interface utilisateur pour garantir une cohérence visuelle et une réutilisation efficace du code.

Dossier Constantes : Ce dossier abrite les constantes spécifiques à notre front-end. Il peut contenir des valeurs telles que les URLs des API, les clés de configuration ou d'autres éléments qui nécessitent une utilisation fréquente et centralisée dans notre application.

Dossier Helpers : Les helpers sont des fonctions utilitaires qui aident à simplifier certaines tâches ou à effectuer des opérations récurrentes dans notre front-end. Ils peuvent inclure des fonctions de formatage de dates, de manipulation de chaînes, de gestion des erreurs, etc. Le regroupement de ces fonctions dans un dossier dédié facilite leur utilisation et leur maintenabilité.

Dossier Images : Ce dossier contient toutes les images utilisées dans notre interface utilisateur. En les stockant de manière centralisée, nous améliorons l'organisation et la gestion des ressources visuelles de notre application.

Dossier Interface : Ce dossier comprend les interfaces TypeScript définissant la structure des données manipulées dans notre front-end. Les interfaces aident à garantir la cohérence des données et facilitent la collaboration entre les membres de l'équipe de développement.

Dossier StylesCss : Dans ce dossier, nous stockons les fichiers de styles CSS spécifiques à notre application. Nous utilisons des feuilles de styles modulaires pour garantir l'encapsulation des styles et éviter les conflits.

Dossier View : Ce dossier contient les différentes vues de notre application, organisées par fonctionnalité ou page. Chaque vue est composée de composants réutilisables et est responsable de l'affichage et de la gestion des données spécifiques à cette fonctionnalité.

Cette architecture organisée et modulaire nous permet de développer notre front-end de manière efficace et maintenable. Elle favorise la réutilisation du code, la cohérence visuelle et la scalabilité de notre application. En utilisant React TSX, nous bénéficions des avantages de TypeScript, tels que la vérification statique des types et l'intelligence

Architecture de l'authentification



Dans un premier temps nous avons opté pour un serveur d'authentification keycloak. Keycloak est une solution open-source d'Identity and Access Management (IAM) développée par Red Hat car :

Sécurité renforcée : Keycloak fournit des fonctionnalités d'authentification et d'autorisation robustes, ce qui permet de sécuriser facilement vos applications web et API. Il prend en charge les protocoles standard tels que OAuth 2.0, OpenID Connect et SAML 2.0 pour assurer une authentification et une autorisation sécurisées.

Single Sign-On (SSO) : Keycloak offre la possibilité de mettre en œuvre le SSO, ce qui permet aux utilisateurs de se connecter une seule fois et d'accéder à plusieurs applications sans avoir besoin de se reconnecter à chaque fois.

Gestion des utilisateurs et des rôles : Keycloak propose une interface d'administration pour gérer les utilisateurs, les groupes et les rôles. Cela simplifie la gestion des identités et des autorisations dans votre application.

Gestion des tokens et des sessions : Keycloak gère les tokens d'accès et de rafraîchissement, ce qui simplifie la gestion de la sécurité et des sessions pour vos applications.

Cependant, Keycloak présente certaines complexités lors de sa mise en place, et notre stack actuelle n'est pas optimale pour son utilisation. C'est pourquoi, après une évaluation approfondie, nous avons décidé de remplacer Keycloak par Clerk, car cette dernière offre plusieurs avantages qui correspondent mieux à nos besoins :

Facilité d'intégration et de déploiement : Clerk est conçu pour être facile à intégrer dans notre application web, notamment avec des bibliothèques de développement telles que Clerk.js. L'installation et le déploiement de Clerk sont simplifiés, ce qui est un avantage pour la mise en place de l'authentification et l'autorisation.

Expérience utilisateur transparente : Clerk propose une expérience utilisateur fluide et transparente, notamment grâce à des fonctionnalités telles que le formulaire de connexion sécurisé par e-mail et la gestion des sessions en arrière-plan. Cela peut améliorer l'expérience utilisateur et réduire les frictions liées à la connexion et à l'inscription.

Authentification par e-mail : Clerk met l'accent sur l'authentification par e-mail, ce qui signifie que les utilisateurs peuvent se connecter en utilisant uniquement leur adresse e-mail, sans avoir besoin de se souvenir d'un mot de passe. Cela peut simplifier le processus de connexion pour les utilisateurs et réduire les problèmes liés aux mots de passe oubliés.

Protection contre les attaques de force brute : Clerk dispose de mécanismes de protection contre les attaques de force brute, ce qui renforce la sécurité de votre application en limitant le nombre de tentatives de connexion.

De plus, un avantage non négligeable est que Clerk offre une version gratuite jusqu'à 5000 utilisateurs, ce qui nous permet de profiter de ses fonctionnalités sans frais supplémentaires, du moins jusqu'à ce que notre base d'utilisateurs dépasse ce seuil.

En remplaçant Keycloak par Clerk, nous avons anticipé une intégration et un déploiement plus faciles, une expérience utilisateur améliorée, une sécurité renforcée et un support technique dédié, le tout sans coût supplémentaire pour une base d'utilisateurs inférieure à 5000. C'est pourquoi nous sommes convaincus que Clerk est le meilleur choix pour répondre à nos besoins actuels en matière de gestion des identités et des accès.

Architecture Back-end



Notre architecture Back-end pour notre application-web est conçue pour offrir une performance optimale, une scalabilité élevée et une sécurité renforcée.

L'architecture Back-end est basée sur Node JS pour garantir une gestion efficace des données et des requêtes. Nous avons également mis en place une scalabilité verticale pour gérer la croissance de notre base utilisateurs.

En termes de sécurité, nous avons Clerk pour protéger les données de nos utilisateurs et pour nous assurer que notre application web est conforme aux normes de sécurité les plus strictes.

Nous sommes convaincus que cette architecture Back-end offrira une performance exceptionnelle à nos utilisateurs et répondra à leur besoin futur.

Architecture du serveur Node

Notre serveur Node repose sur une structure en couches, où chaque couche joue un rôle bien défini dans le flux de traitement des requêtes. Voici un aperçu des différentes composantes de notre architecture :

Contrôleur (Point d'entrée) : Au cœur de notre serveur, nous avons un fichier de point d'entrée appelé "app.js" qui fait office de contrôleur principal. Il gère la réception des requêtes entrantes, les achemine vers les bonnes ressources et orchestre la logique de l'application.

Middleware : Notre architecture utilise des middlewares pour la sécurisation des routes et la gestion des aspects transversaux tels que l'authentification, la gestion des sessions et la validation des données. Ces middlewares interviennent entre la réception des requêtes et l'exécution des routes spécifiques, permettant ainsi de centraliser les fonctionnalités de sécurité et d'améliorer la maintenabilité de notre code.

Services : Nous avons structuré notre code en utilisant le concept de services. Chaque service se concentre sur une fonctionnalité spécifique de notre application, comme la gestion des utilisateurs, les opérations de base de données, les appels à des API externes, etc. Cette approche favorise la modularité et facilite le développement, la maintenance et la réutilisation du code.

Constantes et Interfaces : Nous avons créé des dossiers dédiés aux constantes et aux interfaces pour centraliser et organiser ces éléments essentiels. Les constantes regroupent les valeurs statiques, telles que les clés d'API ou les paramètres de configuration, tandis que les interfaces définissent la structure des objets et des données manipulés dans notre application, améliorant ainsi la lisibilité et la maintenabilité de notre code.

Grâce à cette architecture bien pensée, notre serveur Node offre une structure claire et cohérente. Elle nous permet de développer de nouvelles fonctionnalités de manière efficace et d'ajuster notre code en fonction des besoins changeants de notre projet. De plus, l'utilisation de middlewares et de services facilite la gestion de la sécurité, la scalabilité et la maintenabilité de notre application.

Base de données



Pour la base de données, nous avons opté pour MongoDB, une base de données NoSQL populaire souvent utilisée dans les projets web en raison de ses fonctionnalités flexibles et performantes. MongoDB stocke les données sous forme de documents JSON, ce qui permet une approche plus souple en termes de schéma et facilite le traitement des données.

L'utilisation de MongoDB s'est avérée particulièrement avantageuse pour notre application, notamment en raison de sa parfaite intégration avec Node.js. Nous avons pu interagir avec la base de données en utilisant le package MongoDB officiel pour Node.js, connu sous le nom de MongoClient. Cette approche directe nous a permis d'exploiter pleinement les fonctionnalités de MongoDB sans l'abstraction supplémentaire d'un ORM comme Mongoose.

Un avantage clé de MongoDB est sa capacité à évoluer facilement pour s'adapter aux changements du schéma de données. Dans notre application, où les exigences évoluaient régulièrement, cette flexibilité s'est révélée extrêmement précieuse. Nous avons pu ajouter ou modifier des champs dans les documents sans perturber les données existantes, ce qui a facilité la maintenance de l'application au fil du temps.

En utilisant directement le MongoClient, nous avons également bénéficié de performances optimales lors des opérations de lecture et d'écriture dans la base de données. MongoDB étant conçu pour gérer efficacement de grandes quantités de données, notre application a pu maintenir des temps de réponse rapides même lors de pics d'utilisation.

Cependant, comme toute technologie, MongoDB présente également quelques inconvénients. L'un des principaux défis a été de gérer les opérations de jointure, qui peuvent être plus complexes dans une base de données NoSQL par rapport aux bases de données relationnelles. Nous avons dû concevoir notre schéma de données avec soin pour optimiser les performances et minimiser les requêtes complexes.

Mise en place des tests unitaires Backend et FrontEnd



Pour les tests unitaires sur le front-end, nous avons opté pour le framework Cypress. Cypress est un framework open-source conçu pour faciliter le test des applications web modernes. Il est spécialement adapté pour tester les applications basées sur des technologies telles que JavaScript, TypeScript, React, Angular, Vue.js, etc. Cypress offre une approche novatrice et puissante pour écrire et exécuter des tests automatisés sur le front-end d'une application.

Pour chaque développement sur le Front-end, nous avons réalisé un test unitaire à l'aide de Cypress. Ces tests unitaires nous ont permis de vérifier le bon fonctionnement et la robustesse de chaque composant React développé. Grâce à la facilité d'utilisation de Cypress et à sa syntaxe expressive, nous avons pu écrire des tests clairs et compréhensibles, couvrant différentes interactions utilisateur et scénarios d'utilisation.

L'un des avantages majeurs de Cypress pour les tests unitaires est sa capacité à simuler des interactions utilisateur de manière réaliste. Nous avons pu tester le rendu visuel des composants, les interactions de clics, les saisies dans les champs de formulaire, les validations de données, et bien plus encore. Cette approche a permis de s'assurer que chaque composant réagisse correctement aux actions de l'utilisateur, garantissant ainsi une expérience utilisateur sans erreurs ni dysfonctionnements.

De plus, Cypress nous a offert un environnement de débogage interactif, nous permettant de mettre en pause l'exécution des tests et d'inspecter l'état de l'application à tout moment. Cela s'est avéré très utile pour diagnostiquer rapidement les problèmes potentiels et pour comprendre le comportement du code pendant l'exécution des tests.

Grâce aux tests unitaires réalisés avec Cypress, nous avons pu maintenir un niveau élevé de qualité tout au long du développement du Front-end. Les tests automatisés nous ont permis de détecter et de corriger rapidement les erreurs, évitant ainsi des problèmes coûteux à mesure que l'application prenait de l'ampleur.



Pour les tests unitaires sur le backend, nous avons fait le choix stratégique d'utiliser le framework "Mocha" en combinaison avec "Chai".

Actuellement, l'intégration de Mocha et Chai à notre processus de développement renforce notre capacité à identifier rapidement les erreurs et les problèmes de logique. Mocha, en tant que cadre de test, nous fournit un environnement structuré et souple pour organiser nos tests. Cela nous permet de créer des scénarios de test exhaustifs qui couvrent toutes les fonctionnalités de notre backend.

En parallèle, Chai améliore nos capacités d'assertion en offrant une syntaxe élégante et intuitive. Cela nous permet de formuler des attentes claires concernant le comportement de notre code, simplifiant ainsi la vérification des résultats attendus. Grâce à cette combinaison d'outils, nous sommes en mesure de développer et d'exécuter des tests unitaires cohérents et précis.

Les avantages de cette approche sont significatifs. Les tests unitaires avec Mocha et Chai contribuent à prévenir les erreurs avant qu'elles ne se propagent dans le code, assurant ainsi une maintenance plus aisée et une évolution en douceur de notre backend. En nous appuyant sur ces tests, nous pouvons effectuer des modifications en toute confiance, sachant que notre code répondra aux attentes.

Mise en place de l'hébergement

Hébergement Web

infomaniak

Notre interface est hébergée sur les serveurs d'Infomaniak, ce qui nous assure une disponibilité élevée et une réactivité optimale pour nos utilisateurs. Leur infrastructure robuste garantit une expérience utilisateur fluide et sécurisée.

En plus de l'hébergement web, nous avons également enregistré un nom de domaine pour notre application : www.qrcoffee.fr. Ce nom de domaine personnalisé renforce notre présence en ligne et facilite l'accès à notre application pour nos utilisateurs.

En outre, grâce à Infomaniak, nous bénéficions également d'un serveur de messagerie pour gérer nos courriels professionnels. Cela nous permet d'avoir une communication fluide et professionnelle avec nos utilisateurs et nos partenaires.

Infomaniak se distingue par son engagement en matière de sécurité et de confidentialité des données. Ils mettent en œuvre des mesures de protection avancées pour assurer la confidentialité de nos données sensibles et protéger notre application contre les menaces potentielles.

Nous sommes pleinement satisfaits de notre choix d'hébergement avec Infomaniak. Leur service fiable, leur support technique compétent et leur engagement envers la performance et la sécurité nous permettent de nous concentrer sur le développement de notre application sans avoir à nous soucier des problèmes d'infrastructure.

- ⇒ Nous avons choisi Infomaniak pour :
- La gratuité de leurs services pour les étudiants.
 - La possibilité de mise en place de certificat SSL automatique.
 - La pré-configuration de l'accès à l'application web en définissant un .htaccess.
 - Une interface simple et facile d'utilisation
 - Une FAQ complète et un support de qualité.

En résumé, notre IHM est hébergée sur les serveurs d'Infomaniak, qui nous offrent une plateforme stable et sécurisée. Grâce à leur service d'hébergement, nous pouvons fournir une expérience utilisateur fluide et professionnelle à nos utilisateurs, tout en bénéficiant d'un nom de domaine personnalisé et d'un serveur de messagerie intégré.

Hébergement de l'API



Notre API Node.js est hébergée sur un VPS de chez pulseHeberg. Afin d'optimiser un maximum notre serveur, nous avons opté pour dockeriser l'ensemble de notre API.

En faisant cela, nous pouvons nous permettre d'utiliser un serveur VPS à moindre coût.

Caractéristique du VPS

- ⇒ 4 vCPU
- ⇒ 4GB RAM
- ⇒ 40GB SSD

Architecture du VPS

Nous avons 2 dossiers principaux :

- ⇒ APIQRCoffee
- ⇒ Nginx-proxy-conf

APIQRCoffee

Ce dossier contient notre API. En effet, si nous avons besoin de réaliser une modification en production de l'API ou une mise à jour à la suite de l'implémentation d'une nouvelle fonctionnalité, nous viendrons faire les actions ici même.

Nginx-proxy-conf

Ce dossier est un peu spécial, c'est un serveur Nginx qui va nous permettre de générer des certificats SSL avec letsencrypt pour nos sous-domaines de notre serveur. Celui-ci va écouter les ports ouvert et actifs et en fonction de comment seront remplis les docker-compose.yml de chaque serveur, celui-ci va délivrer un certificat ssl.

Pour l'API, nous avons défini ces deux lignes dans le docker-compose.yml :

- ⇒ - LETSENCRYPT_HOST= \${LETSENCRYPT_HOST}
- ⇒ - LETSENCRYPT_EMAIL= \${LETSENCRYPT_EMAIL}

Ces deux variables sont essentielles à la délivrance du certificat. Pour la variable LETSENCRYPT_HOST, nous y définissons notre sous domaine (api.qrcoffee.fr) et dans la variable LETSENCRYPT_EMAIL nous y définissons l'adresse mail du propriétaire.

Le but étant de fournir un protocole HTTPS à nos sous-domaines afin qu'ils puissent être exploités.