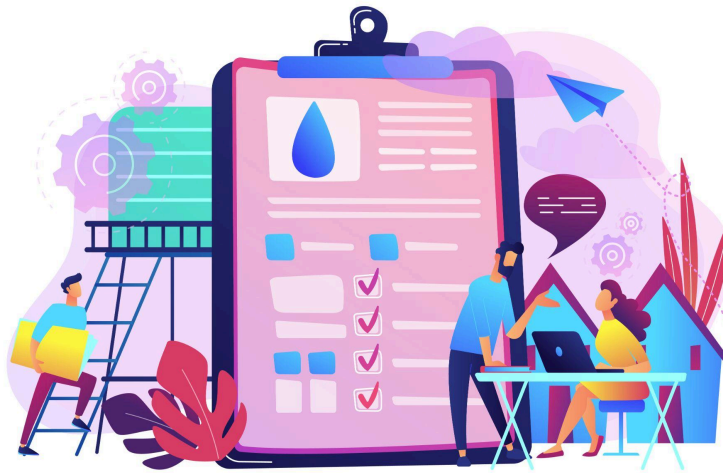


Génie Logiciel



Plan Assurance Qualité

Document regroupant les spécifications et informations clés du projet

Référente : Mme. Valérie GUIMARD

Maîtrise d'oeuvre : M. Jérôme LEVENEZ (VIVERIS)

Maîtrise d'ouvrage : Équipe KAULAB

Responsables : Uzeir JOOMUN & Baptiste TERNISIEN

Consultés : Khalil OUNIS & Antoine BARBAULT

Informés : Lorraine GRANDJEAN, Aurélien DAGNOT

Version 1.0

Table des matières

I- Objectif du Plan Assurance Qualité.....	1
II- Présentation du projet.....	2
a) Présentation de l'application.....	2
b) Contexte du projet.....	2
c) Objectifs du projet.....	3
d) Enjeux du projet.....	4
III- Périmètre du projet.....	5
IV- Livrables.....	7
V- Fonctionnement interne.....	8
a) Organisation de l'équipe.....	8
b) Organisation du projet.....	9
VI- Outils utilisés.....	10
VII- Technologies utilisées.....	12
VIII- Macro planning.....	13
IX- RACI.....	14
X- Matrice des risques.....	17
XI- Processus et Procédures d'Assurance Qualité.....	21
XII- Lexique.....	22

I- Objectif du Plan Assurance Qualité

L'objectif du Plan Assurance Qualité est de définir dès le début du projet, les informations clés ainsi que les attentes du client pour ce projet. L'intérêt est également d'aboutir à un document accessible aux parties prenantes afin de les aligner sur la même compréhension du projet. Le but étant de structurer les différentes parties du projet, les livrables ou encore les attentes du client.

Les processus permettant d'assurer la qualité sont également définis dans ce présent document pour que chacun puisse connaître explicitement son rôle et la marche à suivre pour effectuer correctement sa tâche.

Le Plan Assurance Qualité permet également de communiquer facilement sur l'avancement du projet et les intérêts alignés avec l'équipe de développement.

II- Présentation du projet

a) Présentation de l'application

Notre projet consiste à développer une application mobile visant à faciliter le tri des déchets pour le grand public, c'est-à-dire de 7 à 99 ans, en respectant les consignes du ministère de la Transition écologique et de la Cohésion des territoires. L'application a pour mission d'encourager le tri des déchets pour contribuer à la préservation de l'environnement en réduisant la pollution, les émissions de gaz à effet de serre et la consommation de ressources naturelles.

L'application doit permettre à n'importe qui de scanner un QR code ou un code-barres apposé sur le déchet, afin d'identifier rapidement le type de déchet et d'indiquer la poubelle adéquate pour le jeter, avec une explication du choix. L'utilisation de l'application doit être facile, intuitive et amusante, notamment grâce à l'intégration d'un aspect de gamification, afin de motiver et d'étendre les bonnes pratiques de tri à toutes tranches d'âge. Le joueur est récompensé en fonction du nombre de déchets triés, par un score, des badges et autres cosmétiques visibles par les autres utilisateurs. Il peut adhérer à un clan afin de rejoindre d'autres joueurs. Chaque semaine, les joueurs peuvent participer à une guerre contre un autre clan. L'aspect visuel de l'application doit la rendre attrayante et en adéquation avec la marque de Viveris et son projet écologique.

Ce PAQ définit la façon de travailler et de collaborer au sein de l'équipe, avec l'organisation, les rôles et les responsabilités de chacun dans ce projet.

b) Contexte du projet

Ce projet est réalisé dans le cadre d'un projet de génie logiciel, en partenariat avec l'entreprise Viveris. De nos jours, l'écologie prend une place grandissante dans la société et il est important que tous les citoyens mettent en œuvre des actions afin d'agir dans l'intérêt de tous. Une des actions à entreprendre est la diminution de l'empreinte carbone, et cela se fait par la réduction de nos déchets et la revalorisation de ces derniers. En ce sens, Viveris a souhaité mettre en œuvre une solution permettant d'aider les citoyens à trier correctement leurs déchets.

Du côté de l'équipe KAULAB, l'objectif est de simuler les conditions réelles d'un projet informatique d'entreprise avec des exigences de qualité et des attentes précises de la part de la maîtrise d'œuvre. La composition de l'équipe est décrite dans la partie 5a.

c) Objectifs du projet

L'entreprise Viveris souhaite que l'application soit utilisée de façon pérenne et pour vérifier cela, plusieurs critères sont à prendre en compte :

- Obtenir le nombre d'utilisateurs à la semaine et au mois.
- Obtenir le nombre d'utilisateurs connectés à un instant T.
- Obtenir le nombre de personnes ayant téléchargé l'application (disponible directement sur les magasins d'applications).
- Obtenir le classement des personnes qui ont engrangé des points.

D'autres objectifs sont également à atteindre tels qu'avoir une application qui fonctionne sans bugs majeurs et de manière fluide.

d) Enjeux du projet

Grâce à ce projet, l'entreprise Viveris tend à proposer une application qui vise à encourager les citoyens à trier de façon ludique et instructive. De par la large tranche d'âge visée, l'application pourra instruire les plus jeunes sur les bonnes pratiques à adopter et ainsi construire les citoyens responsables de demain. Les personnes plus âgées pourraient également en apprendre davantage sur le tri et pourraient être sensibilisées indirectement aux causes environnementales, notamment la réduction de gaz à effet de serre et à une consommation plus responsable.

Un autre enjeu important pour Viveris est la fidélisation de l'utilisateur. Ce dernier doit pouvoir revenir régulièrement sur l'application afin de l'utiliser. Cela se fait par l'ajout du volet de gamification afin de rendre le tri plus ludique mais de créer une dynamique vertueuse encourageant à trier de façon collective grâce aux clans et aux guerres entre clans pour se défier. Ainsi, l'application gagnera en popularité et de plus en plus de citoyens pourraient rejoindre l'application pour ensuite se fixer des objectifs personnels de tri et créer un cercle vertueux.

Pour l'équipe KAULAB, les enjeux sont divers. Tout d'abord, ce projet permettra de mettre en application des métiers de l'industrie informatique tels que les rôles de chef de projet, développeurs full-stack/back-end/front-end ou encore Lead Dev, qui sont des rôles cruciaux dans des projets de développements d'applications. Ce projet est donc un premier pas vers le monde professionnel avec cette simulation réaliste avec un réel client.

Ensuite, ce projet permettra la montée en compétence de l'ensemble de l'équipe tant au niveau technique qu'opérationnel. Cela se fait en intégrant l'apprentissage d'un nouveau langage de programmation, des bonnes pratiques de programmation en milieu professionnel mais également en s'identifiant avec le rôle qui nous a été attribué.

Le projet est également un exercice complexe qui demande une bonne organisation afin de gérer efficacement les différents projets que l'équipe a en parallèle de celui-ci. L'organisation intègre également une gestion suffisante des risques et imprévus pouvant corrompre la bonne tenue du planning prévisionnel du rendu des livrables.

Enfin, l'enjeu final est de livrer un produit fonctionnel, répondant aux attentes du client, codé avec de bonnes pratiques et surtout réutilisable par de futures équipes de développement. La documentation a donc tout intérêt à être rédigée avec soin.

III- Périmètre du projet

Le projet porte sur une application développée avec du code multi-plateforme mais elle sera testée uniquement sur le système d'exploitation Android. Le test de la version iOS nécessitant des outils spécifiques, nous ne la prendrons pas en charge.

L'application ne sera adaptée que pour les utilisateurs résidant à Paris, par souci de simplicité pour le développement étant donné que les consignes de tri peuvent être différentes selon les communes de France. Cependant, comme convenu lors de la réunion du 31/10/2024, nous essayerons de rester le plus général possible en ce qui concerne les règles de tri, afin de ne pas rendre complètement inutile l'application dans d'autres villes.

De plus, nous prendrons toutes les dispositions nécessaires en matière de réglementation sur l'utilisation des données personnelles (RGPD), car l'application sera utilisée en France, donc en Europe.

En ce qui concerne les fonctionnalités que nous nous engageons à livrer pour le moment toutes les exigences dont l'id est Mx.

Une liste de déchets à prendre en compte dans l'application a été sélectionnée en accord avec le client afin de simplifier le développement :

- Papier et carton : journaux, magazines, cartons d'emballage
- Plastiques : bouteilles avec code barre, flacons, emballages alimentaires
- Métaux : canettes, boîtes de conserve, aluminium (emballages)
- Verre : bouteilles, bocaux, pots
- Médicaments (incertain)
- Déchets électroniques encombrants (incertain)

Ces déchets doivent posséder un code-barre ou un QR code pour pouvoir être utilisés dans l'application.

Pour l'heure, aucune solution n'a été définie pour les déchets n'ayant pas de code-barre. Une solution pourra éventuellement être choisie, cependant, nous ne nous engageons pas dans la réalisation de cette dernière.

Nous nous engageons à fournir un diagramme des classes complets et mis à jour du back-end lors de la livraison finale. Seulement les interfaces principales des classes de l'IHM seront représentées sur ce dernier.

La date de livraison finale pour le proof of concept (POC) est fixée pour le 17/02/2025, après cette livraison nous ne mettrons plus à jour le code source. Il en sera de la responsabilité du client de le maintenir s'il le souhaite.

De plus, nous excluons de notre périmètre le déploiement de l'application. Cela signifie que le code source fourni travaillera avec des composants locaux, et que nous ne mettrons pas en place de serveur distant pour faire fonctionner l'application.

Les tests d'utilisabilité seront réalisés sur une portion très réduite de personnes sauf si le client nous met à disposition des personnes qui vont tester et nous faire des retours sur l'application.

IV- Livrables

Des rendus de livrables ont déjà été prédéfinis dans la structure du projet Génie logiciel, les voici :

Le premier rendu est prévu pour être livré au plus tard le 05/11/2024. Le contenu de ce livrable est le suivant : la première version de l'expression des besoins, les choix technologiques (non définitifs) que nous envisageons, le Plan Assurance Qualité. Une réunion de lancement devra être prévue la semaine suivant le rendu.

Le second rendu est prévu pour être livré au plus tard le 16/12/2024. Le contenu de ce livrable est le suivant : la version finale de la matrice des exigences, les maquettes des IHM, mini prototype avec une fonctionnalité, un diagramme des classes du back-end, avec les interfaces principales des classes de l'IHM, et 2 diagrammes cas d'utilisations avec userflow associé, ainsi que le planning mis à jour.

Le 15/01/2025, une réunion de pré-soutenance dans laquelle nous ferons une démonstration de l'application avec les fonctionnalités déjà développées et nous présenterons le cahier de recette qui est un document présentant les scénarios de tests que nous devons effectuer.

Le 17/02/2025 correspond à la livraison finale du projet où nous fournirons et présenterons la matrice de respect des exigences du cahier des charges, les dossiers de tests, la documentation utilisateur, ainsi que les fichiers sources et de tests.

En parallèle de ces rendus, nous prévoyons des rendus intermédiaires notamment des comptes-rendus de réunions avec le client et/ou avec notre référente ou encore des versions intermédiaires et stables de l'application. Pour l'heure, nous n'avons pas encore exactement défini ces rendus.

V- Fonctionnement interne

a) Organisation de l'équipe

Afin de bien mener le projet, nous avons décidé de répartir les tâches de travail entre les membres de l'équipe. Pour avoir le meilleur rendement possible, nous avons décidé de procéder avec 3 sous-domaines : un développeur back-end, un développeur full-stack, et un développeur front-end. Ils seront dirigés par un lead-dev qui sera Aurélien Dagnot. Du côté full-stack, on trouvera Lorraine Grandjean. En back-end, on aura Khalil Ounis qui sera responsable de cette partie, Baptiste Ternisien viendra lui apporter de l'aide pour le soutenir dans ses tâches de développement. En ce qui concerne le front-end, c'est Antoine Barbault qui en aura la responsabilité. Le projet sera dirigé par Uzeir Joomun avec l'assistance de Baptiste Ternisien. Pour sa part, Uzeir Joomun viendra ponctuellement apporter son aide sur la partie front-end. Pour une meilleure visualisation de l'organisation de l'équipe, veuillez regarder l'organigramme ci-dessous.

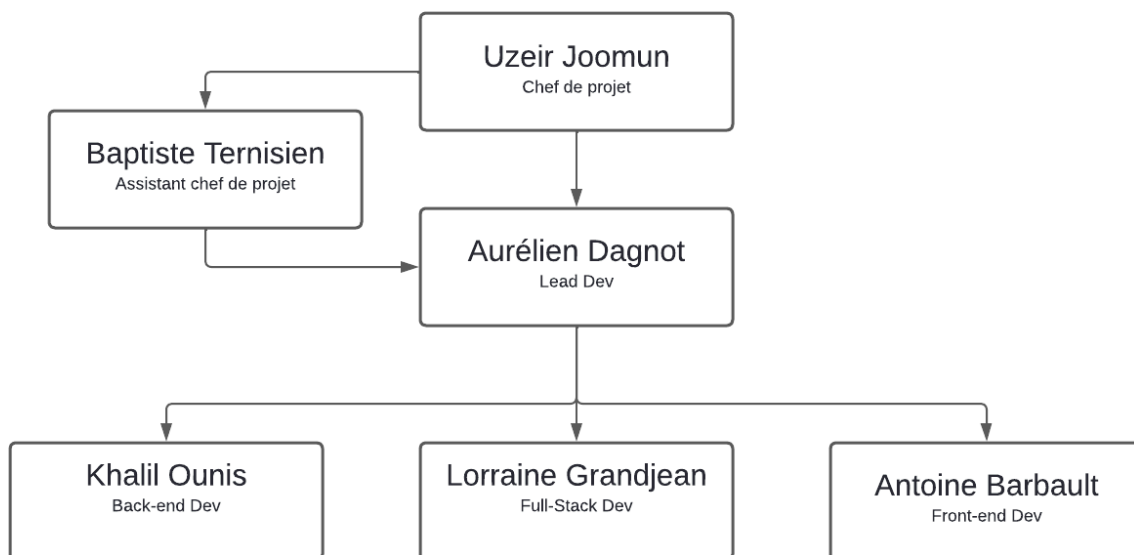


Figure 1 : Organigramme de l'équipe de projet en date du 2/11/2024

b) Organisation du projet

Pour ce projet, l'équipe fonctionnera avec une méthode agile de type Kanban. Ce choix a été fait, car il nous permet de définir des jalons réguliers et une communication récurrente avec le client. Cela nous offre une certaine adaptabilité dans le développement des fonctionnalités.

Le choix de la méthode Kanban a été fait, car la méthode SCRUM nous imposait trop de contraintes temporelles. En effet, les mêlées quotidiennes donnaient un rythme trop important au projet qui ne nous permettrait pas de travailler dans de bonnes conditions. De plus, la méthode Kanban, avec l'utilisation de tableaux, est beaucoup plus visuelle que la méthode SCRUM, ce qui permet de connaître en un coup d'œil l'état de l'avancée du backlog en cours. La tenue d'un tableau Kanban permet à chacun d'être autonome sur ses tâches mais en ne perdant pas de vue l'échéance à respecter, puisque chacun avance à son rythme bien que le résultat final d'une tâche doit respecter l'échéance définie en amont. Cependant, nous reprenons quelques éléments de la méthode SCRUM qui sont l'attribution des rôles et les échéances sur les tâches.

Des réunions régulières entre les membres de l'équipe seront organisées toutes les deux semaines, afin que chacun puisse rendre compte de son avancée, des problèmes rencontrés, et des points bloquants à clarifier avec le client lors de la prochaine réunion.

VI- Outils utilisés

Pour organiser et suivre le travail de l'équipe, nous utilisons Trello, un outil de gestion de projet en ligne basé sur une méthode visuelle et collaborative, avec un grand panel d'actions. Les tâches à réaliser sont représentées par des planches placées dans des répertoires indiquant son état d'avancement : à faire, en cours, vérifications, terminé ou bloqué. On peut assigner une tâche à une ou plusieurs personnes de l'équipe, y attacher des liens vers les documents associés, y ajouter des commentaires et une date d'échéance.

Un autre outil, Jira, sera utilisé afin de définir plus finement les tâches de chacun mais également le diagramme de Gantt et les temps passés par chacun sur leurs tâches respectives. Ce dernier est directement en lien avec le tableau Trello précédemment explicité.

Un espace sur Google Drive rassemble l'ensemble des documents du projet, que ce soit les documents PDF de présentation de l'atelier génie logiciel, les livrables ou les documents à vocation interne à l'équipe comme les notes de réunion. Les Google Docs et Google Sheets sont des outils de traitement de texte et tableur en ligne utilisés de manière collaborative dans le Drive.

Github sera utilisé pour faire de la gestion de version du code source de l'application. Différentes branches seront utilisées pour développer les nouvelles fonctionnalités. La branche principale contiendra une version stable de l'application et sera protégée pour toujours avoir une base saine sur laquelle repartir. Nous utiliserons le système de "*pull request*" pour fusionner les nouvelles fonctionnalités et la branche principale. Cela permettra à l'ensemble de l'équipe d'évaluer le code qui va être introduit dans la version stable.

Les maquettes des IHM de l'application seront faites sur Figma, car ce service est collaboratif et gratuit et permet d'avoir des maquettes avec des tailles réelles de téléphones actuellement sur le marché. Les diagrammes illustrant l'architecture interne de l'application, à savoir les diagrammes de cas d'utilisations, de flux d'utilisateur ou de classes seront faits sur le site web Drawio.com. Ce site a été utilisé de nombreuses fois par les membres de l'équipe pour définir l'architecture de plusieurs projets, c'est donc l'outil le plus pratique pour l'équipe.

Les communications internes entre tous les membres de l'équipe sont faites sur un groupe Discord. Les membres peuvent par exemple poser des questions ou fixer des réunions sur le vocal du groupe. La communication avec le client et notre référente se fait, en revanche, par le biais d'une réunion Google Meets.

Lors des réunions officielles avec le client et notre référente, l'équipe fait sa présentation de son travail avec des slides dont le modèle récupéré sur Slidesgo, et qui sera toujours le même. Le modèle sera modifié sur Google Slides afin que chacun accède à la présentation en cours de préparation.

L'application sera développée sur l'IDE Visual Studio Code. Ce choix a été fait car il s'agit d'un éditeur de code léger et gratuit que l'ensemble de l'équipe connaît et utilise régulièrement. De nombreuses extensions gratuites existent permettant de faciliter le développement.

VII- Technologies utilisées

L'application sera développée en Dart 3.5.4 avec le framework Flutter en version 3.24.4. Les outils de développement Android seront installés par le biais de l'installation de l'interface de développement Android Studio Ladybug en version 2024.2.1.

Nous avons fait le choix de développer l'application pour des systèmes utilisant au minimum Android 11 car 19% des systèmes utilisent ce système. Si l'on additionne les pourcentages des systèmes utilisant des systèmes d'exploitation entre Android 11 à Android 14, cela représente près de 70% des systèmes. L'application serait donc adaptée à la plupart des systèmes ayant un écran de 6.3 pouces.

Le choix de Dart et Flutter a été fait pour permettre à l'équipe de développement de monter en compétence et d'étoffer leurs connaissances. Dart et Flutter permettent de créer des applications multi-plateformes avec le même code, ce qui sera avantageux pour le futur de l'application, si l'on imagine une version iOS de l'application (non incluse dans le périmètre actuel du projet). Les performances de l'application sont hautes de par les différents types d'exécutions possibles du code. Par ailleurs, sa syntaxe relativement simple et rapprochée de langages déjà utilisés par l'équipe le rend plus simple à apprendre. Enfin, les équipes de Viveris travaillant déjà dans ces technologies, la reprise du projet sera donc simplifiée.

Le gestionnaire de version utilisé sera Git en version 2.47.0. Ce dernier sera utilisé par GitHub pour partager le code entre les différents membres de l'équipe.

De plus, afin de simuler une interaction entre le backend de l'application et un serveur, nous utiliserons Docker en version 27.3.1 afin de conteneuriser notre serveur pour reproduire une véritable interaction client-serveur. Par ailleurs, avec l'aide des dockerfile et des dockercompose cela permet le partage d'un environnement commun pour le serveur à tous les membres de l'équipe.

Il est à noter que l'utilisation de docker sous Windows nécessite l'installation de Windows Subsystem for Linux (WSL) en version 2 sur le système. Ce module permet de créer une machine virtuelle Linux de manière native par Windows.

En ce qui concerne l'interface de programmation d'application (API), qui nous permettra d'interagir avec la base de données distante. Nous hésitons encore entre Javascript avec Node.js en version 22.11.0, car nous avons déjà étudié comment faire une API avec ces technologies, et Dart afin d'avoir le même langage de programmation pour l'ensemble de l'environnement.

Il s'agit des dernières versions en date du 31 octobre 2024.

VIII- Macro planning

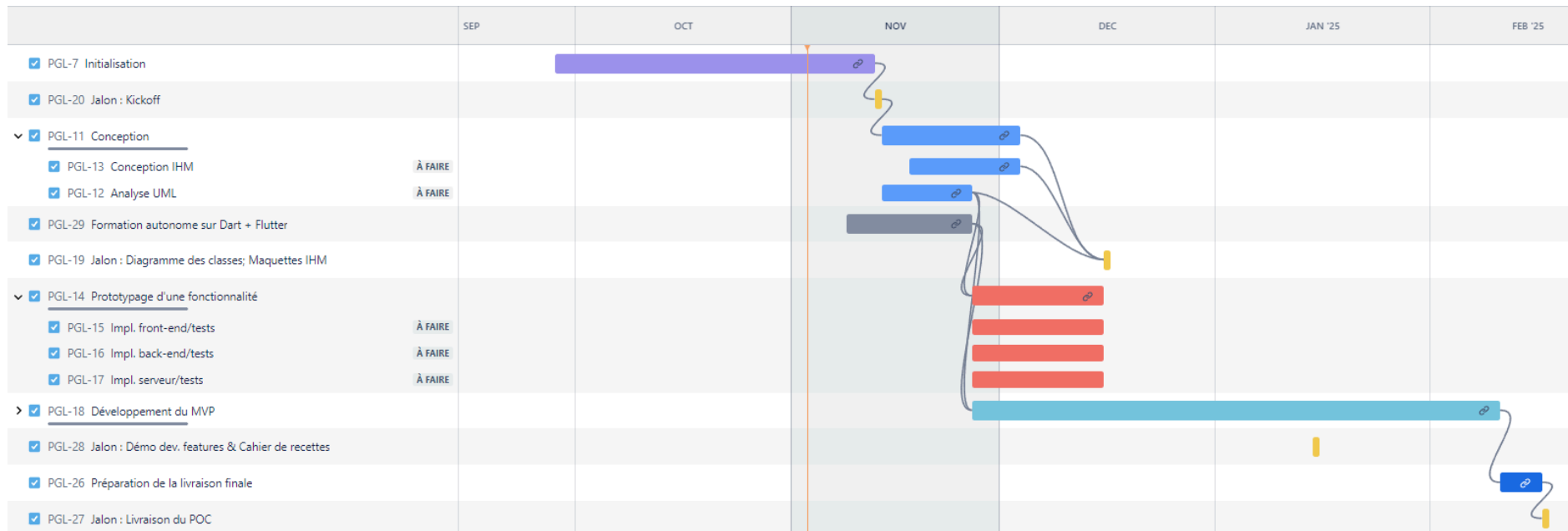


Figure 2. Macro-planning établi en date du 3/11/2024

IX- RACI

N°	Libellé Tâche	Chef de projet	Assistant Chef de projet	Lead Dev	Back-end Dev	Full-stack Dev	Front-end Dev	Responsable Client Viveris
B1	Implémentation du back-end de l'application	A	I	CI	R	C		
B2	Implémentation du back-end au niveau serveur	A	R	CI	C	C		
D1	Rédaction des comptes rendus de réunion	AC	R	I	I	I	I	I
D10	Etablissement du macro planning	R	A	C	I	I	I	I
D11	Conception du diagramme de Gantt	R	A	C	I	I	I	I
D12	Documentation utilisateur	A	C	R	C	C	C	
D2	Conception du diagramme des classes	A	I	R	C	C	C	
D3	Plan Assurance Qualité	R	A	C	I	I	I	CI
D4	Liste des déchets	C	R	I	I	I	I	A

D5	Etablissement de la matrice des exigences	R	C	I	I	I	I	AC
D6	Conception des diagrammes de cas d'utilisations	C	C	R	I	C	I	A
D7	Création des diapositives de réunions	AC	R	C				
D8	Matrice des risques	R	A	C				I
D9	Rédaction du cahier de recettes	AC	R	C	I	I	I	CI
F1	Conception des maquettes IHM	I	I	I		C	R	A
F2	Implémentation des maquettes IHM (front-end)	AC	I	CI		C	R	
T1	Choix technologiques pour l'application	A	I	R	C	C	C	I
T2	Gestion du répertoire GitHub		AC	R	I	I	I	
T3	Livraison du premier prototype de fonctionnalité	A	CI	R				I
T4	Etablissement des fichiers de tests	A	I	R	C	C	C	I

T5	Livraison du prototype final	A	CI	R				I
T6	Validation des tests	A	I	R	I	I	I	I

Figure 3. Matrice RACI établie en date du 2/11/2024

X- Matrice des risques

Risque	Gravité (1 à 4)	Fréquence (1 à 4)	Criticité (G * F)	Prévention	Réparation	Gravité modifiée (1 à 4)	Fréquence modifiée (1 à 4)	Criticité modifiée (G * F)
Non réponse du client	2,5	3	7,5	Relances régulières et faire sa demande suffisamment en avance. Anticiper la prise de contact	Adaptation du planning pour ne pas ralentir l'avancée du projet	1,5	2	3
Arrêt maladie de courte durée	2	3	6	Prévenir à l'avance l'équipe d'éventuels antécédents médicaux qui pourraient ressurgir (prédisposition à certaines maladies...). Mettre l'équipe au courant au plus tôt si un arrêt est déclenché	Si la tâche réalisée par le collaborateur est critique, demander à ce qu'il partage le travail déjà réalisé et affecter un autre membre de l'équipe pour finir cette tâche.	1,5	3	4,5
Arrêt maladie de longue durée	4	1,5	6	Prévenir à l'avance l'équipe d'éventuels antécédents médicaux qui pourraient ressurgir (prédisposition à certaines maladies...). Mettre l'équipe au courant au plus tôt si un arrêt est déclenché	Répartir les tâches que le collaborateur devait effectuer entre les différents membres. Si la charge de travail est trop importante, convenir avec le client soit le report/abandon de fonctionnalités soit un report de la livraison du projet	3	1,5	4,5
Panne d'internet	3	1,5	4,5	Vérifier les branchements. Définir un emplacement de secours où il y aurait une connexion internet	Utiliser un partage de connexion ou rejoindre un lieu connecté à internet (bibliothèque, université...)	2	1	2

Panne d'ordinateur	3	1,5	4,5	Sauvegarde régulière sur un gestionnaire de versionnage de code distant	Réparation de l'ordinateur ou demande de prêt d'ordinateur au service informatique	2	1	2
Mauvaise communication avec le client	3	2	6	Être clair dans les tâches à réaliser et laisser le moins de place à l'interprétation (poser des questions, proposer des idées en adéquation avec la demande client...)	Convenir d'une réunion clarifiant les sujets de discorde	2	1,5	3
Retard dans la livraison d'une fonctionnalité	2,5	2,5	6,25	Informers l'équipe et le client si on sent qu'il y a une difficulté qui mènera à un retard. Suivre régulièrement le projet et les membres, avec par exemple des outils adaptés (Kanban...)	Rajouter d'autres membres sur la tâche pour apporter de l'assistance supplémentaire et décaler la livraison de la fonctionnalité	2	1,5	3
Changement inopiné d'une fonctionnalité de la part du client	2,5	2	5	Définir des échéances pour prendre des décisions et empêcher le client de changer d'idée ou de fonctionnalité après une certaine date	Voir avec le client quelles fonctionnalités peuvent être mis de côté pour pouvoir faire le changement sans trop impacter le planning global	1,5	1,5	2,25
Visions différentes de l'application par les différents membres de l'équipe	3,5	3	10,5	Rappeler régulièrement la vision globale du projet, les objectifs communs et les enjeux définis par le client et l'équipe.	Convenir d'une réunion d'urgence entre les différents membres de l'équipe pour se remettre tous en adéquation vers le même objectif.	2,5	2	5

Conflit d'emploi du temps avec les membres de l'équipe	2,5	3	7,5	Partager les emplois du temps de chacun le plus tôt possible et définir des plages où l'ensemble de l'équipe est disponible. Communiquer les contraintes pouvant affecter l'emploi du temps	Si possible omettre le ou les membres indisponibles pour la réunion ou tâche. Les membres avancent sur leurs tâches personnelles	1,5	2	3
Manque de temps pour réaliser une tâche	2,5	2,5	6,25	Prévenir l'ensemble de l'équipe que la tâche ne pourra pas être réalisée dans le temps imparti.	Affecter un ou plusieurs membres sur la tâche en question	1,5	2	3
Problèmes de compatibilité entre les versions du langage utilisées	3	2	6	Mettre en place un environnement virtuel qui permet d'unifier les versions des composants/langages pour tous	Mettre à jour les versions des composants posant une difficulté	1,5	1	1,5
Technologies utilisées pour le projet, non utilisées par le client	3	3	9	Mettre en place un document regroupant toutes les technologies utilisées conjointement pour pouvoir se mettre d'accord sur les technologies utilisées	Accord avec le client pour apprendre ces technologies à leurs équipes ou changement de technologies dans le projet mais avec un report de livraison de ce dernier.	2,5	1,5	3,75
Mauvaise utilisation des outils de gestion par les membres	2	3	6	Apprendre à chacun à utiliser les outils collaboratifs pour informer tout le monde des tâches en cours	Demander à chacun d'utiliser les outils correctement et regrouper les informations éparpillées sur les outils	1	2,5	2,5

Indisponibilité du chef de projet	3	2,5	7,5	Désigner un suppléant au chef de projet. Prévenir le suppléant dès que possible d'une indisponibilité.	Prise en charge des tâches clés du chef de projet par le suppléant	1,5	2,5	3,75
Sous-estimation de la durée d'une tâche	2,5	2	5	Prendre une durée plus large pour la réalisation d'une tâche	Peut n'avoir aucun effet si ce n'est pas une tâche critique. Sinon, affecter un ou plusieurs membre supplémentaire sur la tâche	1,5	1	1,5

Figure 4. Matrice des risques réalisée en date du 12/10/2024

XI- Processus et Procédures d'Assurance Qualité

Lors du commencement du développement du code pour une fonctionnalité, le développeur en charge de la fonctionnalité doit écrire le programme de test correspondant. Chaque test doit être isolé et ne doit pas dépendre de l'état d'autres tests, facilitant ainsi l'identification des sources d'erreur lors du travail de débogage. Chaque test doit donc restaurer l'état initial après son exécution. Tous les tests doivent pouvoir être exécutés avec une seule commande, pour pouvoir ainsi vérifier rapidement si un programme satisfait bien tous les tests.

Une fois que le développeur a terminé d'écrire le code d'une fonctionnalité et que celui-ci passe tous les tests, il doit se rendre sur Trello pour déplacer la tâche correspondante à l'état de vérification. Le Lead Dev est alors responsable de réaliser des revues de code, ce qui permet de détecter les erreurs à un stade précoce du projet et d'éviter des retards causés par des révisions tardives.

Le code du projet ainsi que celui des tests doivent être bien documentés, de sorte que les attentes de chaque test et fonctionnalité soient claires pour tous les membres de l'équipe et les nouvelles personnes arrivant sur le projet.

Du côté de la gestion du projet, le tableau Kanban modifié permet de visualiser les échéances de chaque tâche et d'ajuster leur durée respective en fonction des événements pouvant survenir. Cela permet aux chefs de projets de prévoir d'éventuelles extensions de temps, d'ajuster le planning, mais également s'il faut rajouter des ressources humaines supplémentaires sur une tâche.

XII- Lexique

Framework : Cadre de développement qui contient des composants logiciels permettant de mettre en place l'architecture d'une application

IDE (Environnement de développement intégré) : Logiciel permettant de coder, compiler et exécuter une application.

Proof of Concept (POC) : Prototype fonctionnel démontrant la viabilité des fonctionnalités en situation réelle.

Interface Hommes-Machines (IHM) : Interface visible par l'utilisateur et faisant le lien avec la logique interne de l'application.

Front-end : Il s'agit de la réalisation, en code, des maquettes IHM d'une application définies par le designer UI/UX. Le développeur doit s'assurer de la bonne utilisabilité de l'application par l'utilisateur et doit pouvoir transmettre les actions et requêtes utilisateurs à la partie interne de l'application.

Back-end : Il s'agit de la réalisation, en code, de la logique interne d'une application. Le développeur back-end ne s'occupe que des technologies qui concernent la partie serveur d'une application et du traitement des requêtes effectuées par l'utilisateur.

Full-Stack : Il s'agit de l'union des parties front-end et back-end. Le développeur full-stack se charge donc de réaliser et de faire le lien entre ces 2 parties.

XIII - Sources

- Image de la page de garde conçue par vectorjuice/Freepik -
https://www.freepik.com/free-vector/people-near-huge-paper-tablet-with-report-water-flow-checkboxes-analyzing-data-illustration_10780374.htm#fromView=search&page=1&position=35&uuid=8ee34c7a-e8ba-4f6a-af25-6fa5190b5458
- Organigramme réalisé sur LucidChart -
https://lucid.app/lucidchart/6bb1b0c8-544c-486f-9378-901c066cb736/edit?viewport_loc=-615%2C-794%2C2846%2C1316%2C0_0&invitationId=inv_0fe3ce0d-8d9c-44bb-bd3c-4ded8efce287