

PLANIFICATION DU PROJET

“Science Quest” - 31 octobre 2023

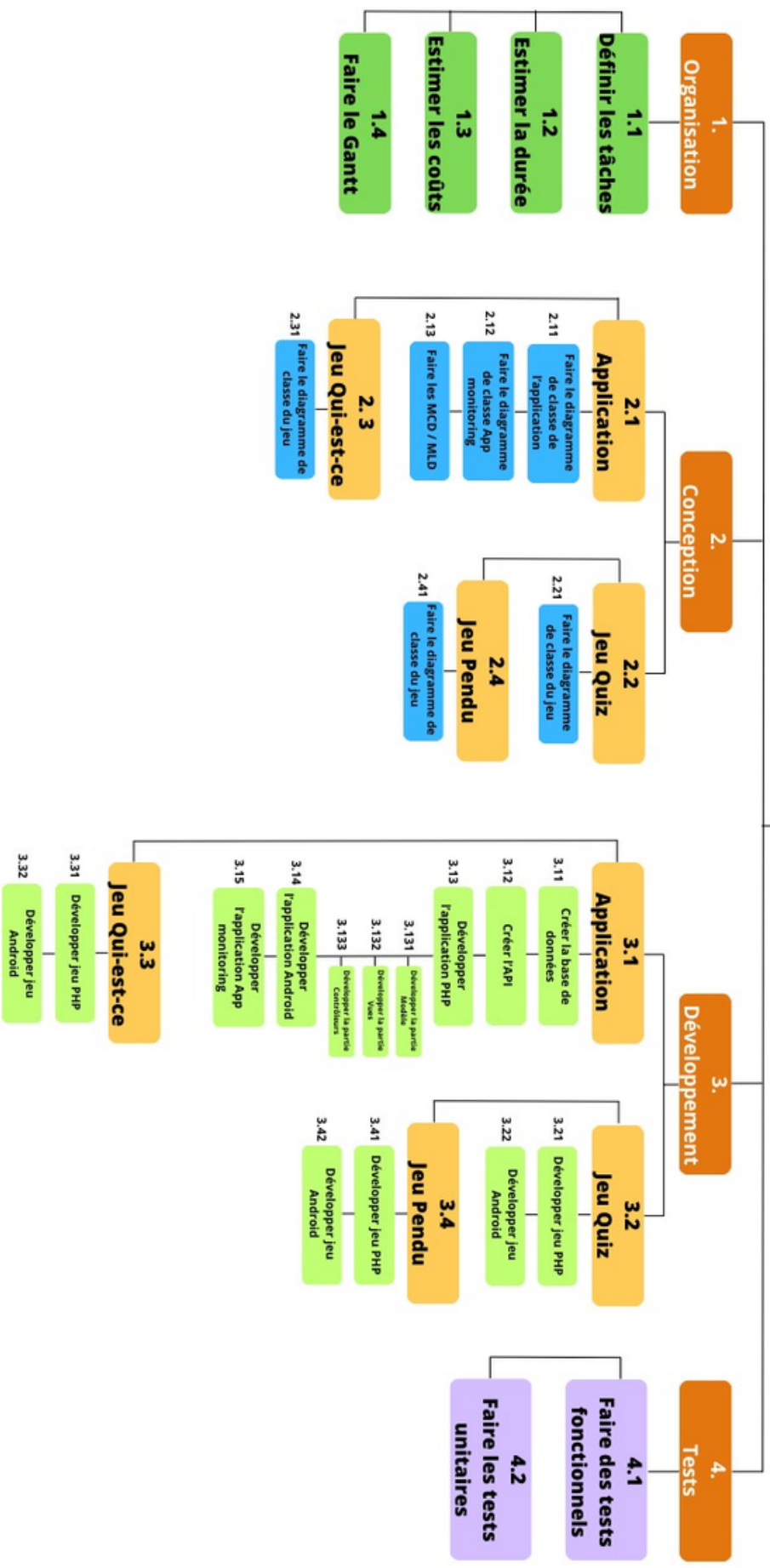
Rédigé par:

SOULIER Victor, BRETON LILIAN, BEURET
Renaud, BIARD TOM, JEUDI--LEMOINE Alix.

SOMMAIRE

SOMMAIRE.....	2
WBS.....	3
EXPLICATION DU WBS.....	4
ESTIMATION DE LA DURÉE DES TÂCHES.....	5
EXPLICATION DE L'ESTIMATION.....	6
GANTT.....	7
EXPLICATION DU GANTT.....	8
PERT.....	9
EXPLICATION DU PERT.....	10
COÛTS PREVISIONNELS.....	11
INDICATEURS DE SUIVI.....	12
OUTILS DE COMMUNICATION.....	13
ORGANISATION DU TRAVAIL.....	14
CONCLUSION.....	15

Science Quest



WBS

EXPLICATION DU WBS

Comme vous pouvez le voir sur le schéma précédent, nous avons défini **quatre grandes étapes** à notre projet.

La première de ces étapes correspond à **l'organisation de notre travail**, étape qui servira de base pour tout le reste du projet. En effet, c'est durant cette dernière que nous allons définir les tâches, et surtout estimer la durée de ces dernières. Cette estimation nous permettra de dégager un **chemin critique**, et donc de savoir quelles tâches peuvent nous poser problème pendant le projet.

Ensuite, nous trouvons une étape de **conception**, étape cruciale pour préparer le terrain avant de commencer à développer. Elle nous permettra d'avoir une **base solide** avant de coder, et donc d'avoir une **idée bien plus claire** de ce que nous devons faire. Notre application regroupant plusieurs mini jeux, nous avons besoin de concevoir l'application générale, mais aussi chacun des jeux que nous implémenterons. De plus, notre projet requiert une application annexe, qui servira à faire de l'app monitoring. Nous aurons donc encore une fois besoin de concevoir cette application, car son fonctionnement différera de notre application principale.

Après cela, nous pourrons commencer sereinement le **développement** de nos différentes applications. Encore une fois, nous devons faire plusieurs versions de notre application : une application web en php et une application mobile en android, en plus de l'app monitoring, soit trois applications à développer. Nous avons ensuite besoin de gérer des données, nous devons donc créer une base de données pour les contenir. Pour accéder à ces données depuis nos différentes applications, nous aurons besoin de coder notre propre API (Application Programming Interface). Elle nous permettra de récupérer, modifier, ajouter ou encore supprimer les données de notre base de données.

La dernière étape est une étape de **test**. Il est très important de bien tester son application afin de rendre un produit le plus abouti possible à notre client.

Nous pensons donc réaliser au moins deux types de tests pour cela :

- Nous voulons réaliser des **tests unitaires**, qui nous permettront de savoir si chaque élément de l'application fonctionne comme il le devrait.
- Nous voulons aussi faire des **tests fonctionnels**, pour nous assurer que notre application fonctionne correctement dans sa globalité, et surtout qu'elle est facile et agréable d'utilisation.

ESTIMATION DE LA DURÉE DES TÂCHES

N°	Task Name	Durée
1	Définir les tâches	1 h
2	Estimer la durée des tâches	1 h
3	Estimer les coûts	0,4 h
4	Faire le GANTT	1 h
5	Application : Faire le diagramme de classe	2 h
6	Jeu : Faire le diagramme de classe	1 h
7	App Monitoring : Faire le diagramme de classe	1 h
8	Faire le MCD / MLD	0,5 h
9	Créer la base de données	1,5 h
10	Créer l'API	12 h
11	PHP : développer la partie modèle	5 h
12	PHP : développer la partie vues	5 h
13	PHP : développer la partie controleur	5 h
14	PHP : développer les classes annexes	5 h
15	Développer l'application Android	20 h
16	Développer l'application App Monitoring	6 h
17	Qui-est ce: implémenter la logique du jeu en PHP	3,33 h
18	Qui-est ce: implémenter la logique du jeu sur Android	3,33 h
19	Quizz: implémenter la logique du jeu en PHP	3,33 h
20	Quizz: implémenter la logique du jeu sur Android	3,33 h
21	Pendu: implémenter la logique du jeu en PHP	3,33 h
22	Pendu: implémenter la logique du jeu sur Android	3,33 h
23	Tests fonctionnels	4 h
24	Tests unitaires	12 h

EXPLICATIONS DURÉE DES TÂCHES

Après avoir effectué un planning poker afin d'évaluer le temps qu'allait nous prendre chaque tâche ainsi que de l'effort requis pour ces dernières, sur le schéma ci-dessus. Chaque tâche a été évaluée en prenant en compte les compétences de l'équipe, les outils disponibles et les dépendances entre les différentes étapes du projet. Le planning a été conçu de manière à assurer une progression logique du projet tout en tenant compte des possibles imprévus.

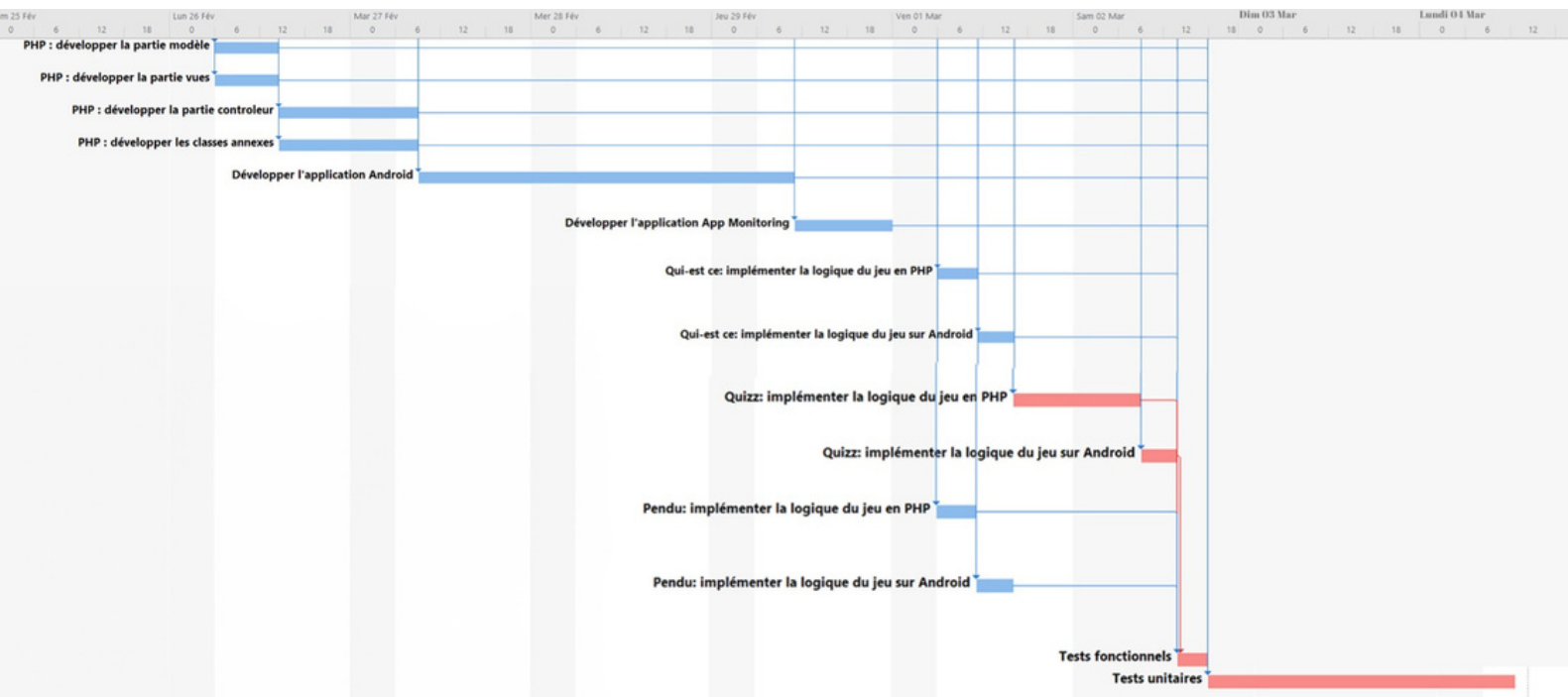
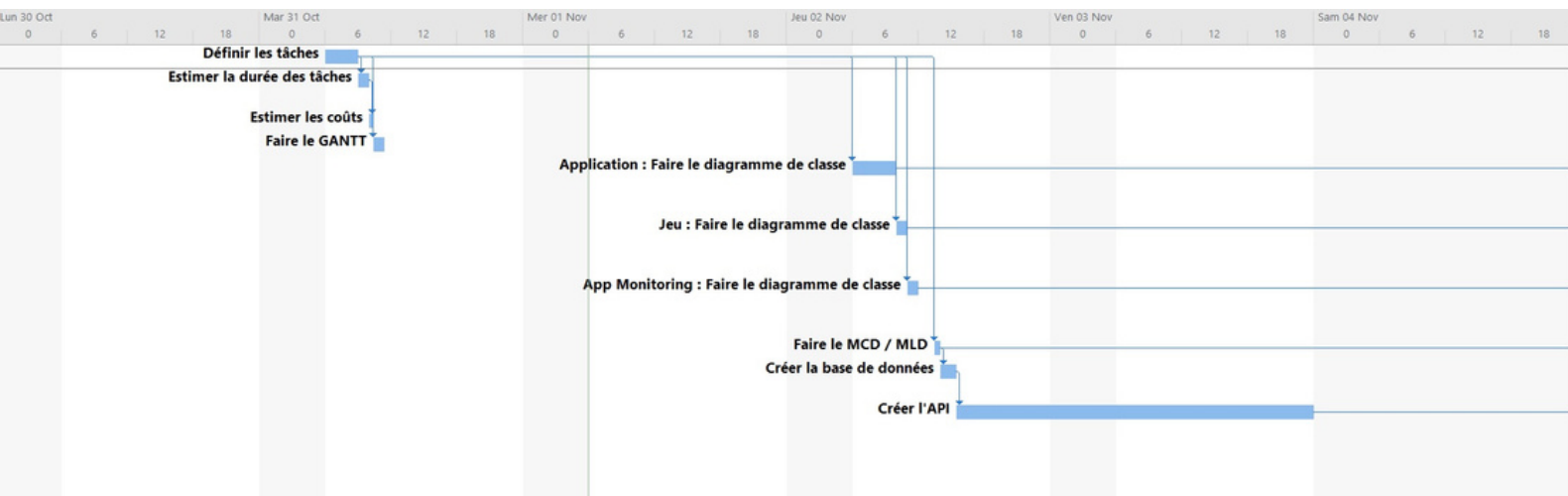
Le Planning Poker étant une technique de estimation utilisée dans la gestion de projets agiles, notamment dans le cadre de la méthode SCRUM. Cette méthode nous a permis d'estimer de manière collaborative et relativement rapide la complexité des tâches à réaliser.

Pour toutes les tâches qui relèvent de la gestion de projet, l'équipe est réunie ce qui diminue drastiquement le temps passé sur chacun de ces tâches de manière individuelle. Il s'agit d'un gain de temps non négligeable en plus d'une assurance que tous les membres de l'équipe seront satisfaits au mieux que possible.

Nous avons conscience que certaines durées seront probablement incohérentes, notamment en ce qui concerne les tâches dont nous ne connaissons pas bien le fonctionnement pour le moment (par exemple la tâche "Créer l'API"). Certains modules auront donc des durées réelles plus longues (ou plus courtes) faute d'expériences et de pratique. Concernant les autres tâches que nous avons déjà travaillées auparavant, bien que n'ayant jamais collaboré sur un projet de cette ampleur, nous sommes assez confiant sur le temps que nous leur avons attribuées.

La tâche "Développer l'application Android" est une bien trop grosse tâche, nous en avons tous conscience, cependant nous n'avons pas de connaissances suffisantes en la matière pour pouvoir la diviser en plusieurs tâches. Il y aura très probablement un assez grand écart entre ce que nous avons estimés et la réalité des choses sur cette tâche.

GANTT



EXPLICATION DU GANTT

Notre projet se divise en 3 périodes, une première du 30 octobre au 3 novembre 2023, une deuxième du lundi 26 février au 1 mars 2024 et enfin une troisième période contenant les 36 heures restantes réparties sur les 5 semaines suivantes durant les cours.

C'est pour cette raison que notre GANTT est divisé en plusieurs parties. Sur la première image nous pouvons retrouver les prévisions pour la première période, et sur la deuxième image celles de la deuxième et troisième période.

Malheureusement, les dates de la troisième période de travail étant encore abstraites, nous n'avons pas terminé de saisir leurs contraintes de temps dans MS Project. Cependant ceci sera corrigé lorsque nous auront à disposition le planning de la troisième période.

En ce qui concerne la suite logique des tâches, nous commençons par la gestion de projet car nécessaire au bon déroulement du projet. Ensuite, la conception, elle sert de plan pour le reste du projet, elle permet de poser les fondations du projet. Nous commençons par le site web en PHP, la première matière que nous voyons cette année, avant le Kotlin et le React Native qui seront utilisés pour l'application mobile. Et enfin, les tests car ils permettent d'être sûrs de livrer une application de qualité.

Le GANTT est basé sur l'estimation de la durée des tâches, celle-ci n'étant pas parfaite, il sera par conséquent modifié en fonction du temps réel que nous prendra chacune des tâches. C'est pourquoi il n'est là qu'à titre indicatif et n'est pas impérativement à respecter.

PERT TEMPS

N°	Task Name	Durée	Début	Fin	Début au plus tôt	Début au plus tard
1	Définir les tâches	1 h	Mar 31/10/23	Mar 31/10/23	Mar 31/10/23	Mer 28/02/24
2	Estimer la durée des tâches	1 h	Mar 31/10/23	Mar 31/10/23	Mar 31/10/23	Lun 04/03/24
3	Estimer les coûts	0,4 h	Mar 31/10/23	Mar 31/10/23	Mar 31/10/23	Lun 04/03/24
4	Faire le GANTT	1 h	Mar 31/10/23	Mar 31/10/23	Mar 31/10/23	Lun 04/03/24
5	Application : Faire le diagramme de classe	2 h	Jeu 02/11/23	Jeu 02/11/23	Jeu 02/11/23	Jeu 29/02/24
6	Jeu : Faire le diagramme de classe	1 h	Jeu 02/11/23	Jeu 02/11/23	Jeu 02/11/23	Jeu 29/02/24
7	App Monitoring : Faire le diagramme de classe	1 h	Jeu 02/11/23	Jeu 02/11/23	Jeu 02/11/23	Sam 02/03/24
8	Faire le MCD / MLD	0,5 h	Jeu 02/11/23	Jeu 02/11/23	Jeu 02/11/23	Mer 28/02/24
9	Créer la base de données	1,5 h	Jeu 02/11/23	Jeu 02/11/23	Jeu 02/11/23	Mer 28/02/24
10	Créer l'API	12 h	Jeu 02/11/23	Ven 03/11/23	Jeu 02/11/23	Mer 28/02/24
11	PHP : développer la partie modèle	5 h	Lun 26/02/24	Lun 26/02/24	Lun 26/02/24	Sam 02/03/24
12	PHP : développer la partie vues	5 h	Lun 26/02/24	Lun 26/02/24	Lun 26/02/24	Sam 02/03/24
13	PHP : développer la partie controleur	5 h	Lun 26/02/24	Mar 27/02/24	Lun 26/02/24	Sam 02/03/24
14	PHP : développer les classes annexes	5 h	Lun 26/02/24	Mar 27/02/24	Lun 26/02/24	Sam 02/03/24
15	Développer l'application Android	20 h	Mar 27/02/24	Jeu 29/02/24	Mar 27/02/24	Jeu 29/02/24
16	Développer l'application App Monitoring	6 h	Jeu 29/02/24	Jeu 29/02/24	Jeu 29/02/24	Sam 02/03/24
17	Qui-est ce: implémenter la logique du jeu en PHP	3,33 h	Ven 01/03/24	Ven 01/03/24	Ven 01/03/24	Dim 03/03/24
18	Qui-est ce: implémenter la logique du jeu sur Android	3,33 h	Ven 01/03/24	Ven 01/03/24	Ven 01/03/24	Dim 03/03/24
19	Quizz: implémenter la logique du jeu en PHP	3,33 h	Ven 01/03/24	Sam 02/03/24	Ven 01/03/24	Dim 03/03/24
20	Quizz: implémenter la logique du jeu sur Android	3,33 h	Sam 02/03/24	Sam 02/03/24	Sam 02/03/24	Dim 03/03/24
21	Pendu: implémenter la logique du jeu en PHP	3,33 h	Ven 01/03/24	Ven 01/03/24	Ven 01/03/24	Dim 03/03/24
22	Pendu: implémenter la logique du jeu sur Android	3,33 h	Ven 01/03/24	Ven 01/03/24	Ven 01/03/24	Dim 03/03/24
23	Tests fonctionnels	4 h	Sam 02/03/24	Sam 02/03/24	Sam 02/03/24	Dim 03/03/24
24	Tests unitaires	12 h	Sam 02/03/24	Lun 04/03/24	Sam 02/03/24	Sam 02/03/24

EXPLICATIONS DU PERT TEMPS

En comparaison des dates au plus tôt et des dates au plus tard, on pourrait penser que nous sommes largement dans les temps pour la gestion de projet. Cependant la date jalon se trouvant aussi très proche de la majorité des dates au plus tard, nous nous devons de garder une allure moyenne tout au long du projet pour ne pas faire face à une situation de surexploitation.

On peut remarquer le même phénomène pour certaines tâches durant les modules de programmation et de développement mobile. Le Product Owner nous ayant imposé d'avoir une première version fonctionnelle pour les portes ouvertes du département à des fins de démonstration, nous devons encore une fois garder une allure de travail correcte pour ne pas avoir à faire tout le projet avant Mars. En considérant le module PHP exigeant un site fonctionnel à sa fin, nous ne pouvons pas nous permettre de prendre trop de retard si nous voulons pouvoir répondre à ces exigences.

La date jalon de la finalité du projet se situe en avril et la soutenance du projet entre le 28 et 30 avril 2024.

Selon nos estimations, nous devrions être dans les temps, actuellement nous avons tous individuellement une dizaine d'heures de marge, qui nous permettront de combler l'incertitude que nous avons vis à vis de certaines tâches comme évoqué dans l'explication du GANTT.

COÛTS PRÉVISIONNELS

Nous allons essayer d'estimer les coûts qu'impliquent notre projet durant tout son développement.

En premier lieu, ce projet devrait durer environ 100 heures par personne. Notre équipe est constituée de cinq développeurs, pour lesquels le salaire horaire médian est de 17.86€.

Ainsi, le coût de la masse salariale sera de $17.86 * 5 * 100 = 8930\text{€}$.

Il est aussi important de prendre en compte les charges salariales. Ces dernières s'élèvent à 22% du salaire brut, soit dans notre cas $0.22 * 8930 = 1964.6\text{€}$.

De plus, nous avons besoin de payer un local pour pouvoir travailler sur ce projet. Nous sommes une équipe de cinq, nous estimons donc avoir besoin d'au moins 45m² pour travailler dans de bonnes conditions. Le prix locatif moyen au m² à Clermont-Ferrand est de 11€ par mois, et le projet durera six mois au total (d'octobre à mars environ). Ainsi, le coût du local est de $11 * 45 * 6 = \underline{2970\text{€}}$.

Ensuite, nous devons considérer les coûts des équipements. Nous prendrons comme base, un ordinateur au prix de 500€. En moyenne, la durée de vie d'un ordinateur de cette gamme est de 4 ans. Ainsi, comme notre projet dure une demi année, cela représente 1/8 de sa durée de vie. Nous pouvons donc estimer le coût en équipement à 62.5€ par ordinateur (1/8 de son prix) * 5 = 312.5€.

Puis, nous devons prendre en compte les différentes charges, tel que l'électricité, le chauffage... Tout d'abord, un ordinateur d'une puissance de 100 Watts coûte environ 3cts par heure. Nous avons cinq ordinateurs, qui seront chacun utilisés 100 heures, soit $0.03 * 5 * 100 = \underline{15\text{€}}$.

Nous devons aussi en compte les autres charges, comme le chauffage, la lumière... Ces coûts sont variables et donc compliqués à estimer, nous allons donc prendre une valeur arbitraire que nous estimons à 15€.

Enfin, nous utiliserons un logiciel payant, MS Project, dont le prix d'une licence s'élève à 28.10€ hors taxe et par mois pour cinq appareils. Comme nous travaillons 6 mois, cela représente $28.10 * 6 = \underline{168.6\text{€}}$.

Au total, ce projet coûtera donc :

$8930 + 1964.6 + 2970 + 312.5 + 15 + 15 + 168.6 = \underline{14365.7\text{€}}$

INDICATEURS DE SUIVI

Pour le suivi de notre projet, nous allons nous baser sur deux catégories.

La première rassemblent les indicateurs de temps :

Pour les indicateurs de temps, nous pensons faire deux choses. Premièrement nous réaliserons des Kanban à chaque début de sprint, avec les tâches que nous devons réaliser durant ce dernier. Cela nous permettra de constater l'avancement de ces tâches en direct, afin de nous rendre compte de l'avance ou du retard que nous avons par rapport à ce que nous avons prévu de faire. Notre objectif étant de prendre le moins de retard possible sur les dates jalons.

Ensuite, nous pensons réaliser un GANTT réel, avec l'avancement des tâches en direct durant l'entièreté du projet afin de le comparer à notre GANTT prévisionnel. Comme le précédent point, cela nous permettra de constater notre avance ou notre retard en temps réel par rapport à ce que nous avons prévu.

Nous allons aussi utiliser des indicateurs de qualité :

- Le premier étant de maintenir une couverture de tests à minimum 33%, ce qui nous aidera à assurer un projet de bonne qualité, est l'implémentation de tests.

- Ensuite, via SonarQube, nous pouvons vérifier la qualité de notre code (répétition de code...). Nous essayerons aussi de maintenir cette valeur au plus bas possible car cela assure un code plus propre et plus facilement maintenable dans le futur.

- Nous avons aussi, via Drone, accès à la dette technique de notre application. De la même manière que les autres, cette valeur devra rester la plus basse possible afin de ne pas perdre trop de temps à la fin de notre projet à récupérer le retard accumulé via cette dette.

OUTILS DE COMMUNICATION

Pour les outils de communications, nous utilisons principalement **Discord** pour communiquer entre membres de l'équipe. Nous avons choisi cette solution car c'est un logiciel que nous connaissons tous bien et que nous utilisons tous régulièrement.

Pour communiquer avec nos professeurs et avec le PO, nous privilégions avant tout les réunions en **face à face**, que nous préférons à de la visioconférence et que nous trouvons surtout plus efficaces. Cependant, nous nous ouvrons aussi la possibilité de réunions en **visioconférence** si un face à face n'est pas possible, notamment lors des deux semaines de travail que nous avons.

Ensuite, nous utilisons **CodeFirst** comme plateforme pour héberger notre code et le partager entre nous. Cette solution nous permet de travailler tous sur le même code en simultané sans rencontrer de soucis. De plus, c'est aussi une plateforme que nous connaissons bien et que nous avons tous déjà eu l'occasion d'utiliser par le passé.

ORGANISATION DU TRAVAIL

Pour ce projet, notre équipe travaillera via la méthode **SCRUM**.

Dans la pratique, notre équipe s'organise en **Sprints**, c'est à dire des périodes d'une semaine durant lesquelles toute l'équipe travaille sur le projet sans interruptions.

Pour chacun de ces sprints, nous définirons un **Sprint Planning**, c'est à dire l'ensemble des tâches que nous allons réaliser au court de ce sprint. Nous mettrons ces tâches dans un **Kan-Ban**, afin de visualiser l'avancement de ces tâches par rapport à nos prévisions.

Ensuite, durant chaque journée du sprint, notre équipe fera un **Daily Meeting**, c'est à dire une petite réunion rapide, dans laquelle chaque membre décrit ce qu'il a fait la veille, ce qu'il va faire aujourd'hui, et ce sur quoi il a bloqué. Elles se feront sur discord, à 9h chaque jour.

Puis, chaque membre de l'équipe réalise ses tâches durant la journée, et demande évidemment de l'aide s'il en a besoin.

A la fin de chaque sprint, nous organiserons aussi deux choses. Tout d'abord, un **Sprint Review**, afin de présenter à notre Product Owner les avancées de l'application au court de ce sprint.

Nous organiserons aussi une **Sprint Retrospective**, qui servira à définir ce qui a fonctionné ou non tout au long de ce sprint, afin de s'améliorer pour le prochain.

Toutes ces étapes se répètent évidemment pour chacun de nos sprints, et nous nous laissons aussi la possibilité de contacter le product owner ou des professeurs en dehors de nos sprints.

CONCLUSION

Notre document détaille la planification, les estimations, les coûts prévisionnels, les outils de communication, et l'organisation du travail pour mener à bien notre projet. Il met en lumière les défis que notre équipe devra relever, notamment en ce qui concerne le développement de l'application Android et la création de l'API, pour lesquels nos estimations sont moins précises en raison de notre manque d'expérience.

Pour assurer un suivi efficace du projet, nous avons mis en place des indicateurs de suivi précis. Tels que le Kanban pour surveiller l'avancement des tâches au cours de chaque sprint, permettant ainsi de visualiser de façon claire les tâches à venir et en cours. De plus, un GANTT actualisé nous servira à comparer l'évolution du projet avec notre planification initiale, nous aidant ainsi à anticiper d'éventuels retards.

Au-delà de la gestion du temps, nous avons également mis en place des indicateurs de qualité. Nous nous engageons à maintenir une couverture de tests d'au moins 33% pour garantir la robustesse de notre application. L'utilisation de SonarQube nous permettra de surveiller la qualité du code en vérifiant la répétition de code et d'autres facteurs de qualité. De plus, grâce à l'outil Drone, nous serons en mesure de contrôler la dette technique, veillant ainsi à ce qu'elle reste à un minimum absolu pour éviter de futurs problèmes.

Tous ces éléments nous fournissent une base solide pour suivre de près l'évolution du projet, identifier les problèmes potentiels, et les résoudre en temps opportun. Ainsi, nous sommes résolus à affronter les défis qui se présenteront et à livrer un produit de qualité à notre client tout en respectant notre budget et nos délais.