AFL MUSIC

SAE 1.01

Développement d'une application

Dossier: ***Planification du projet***

2022-2023

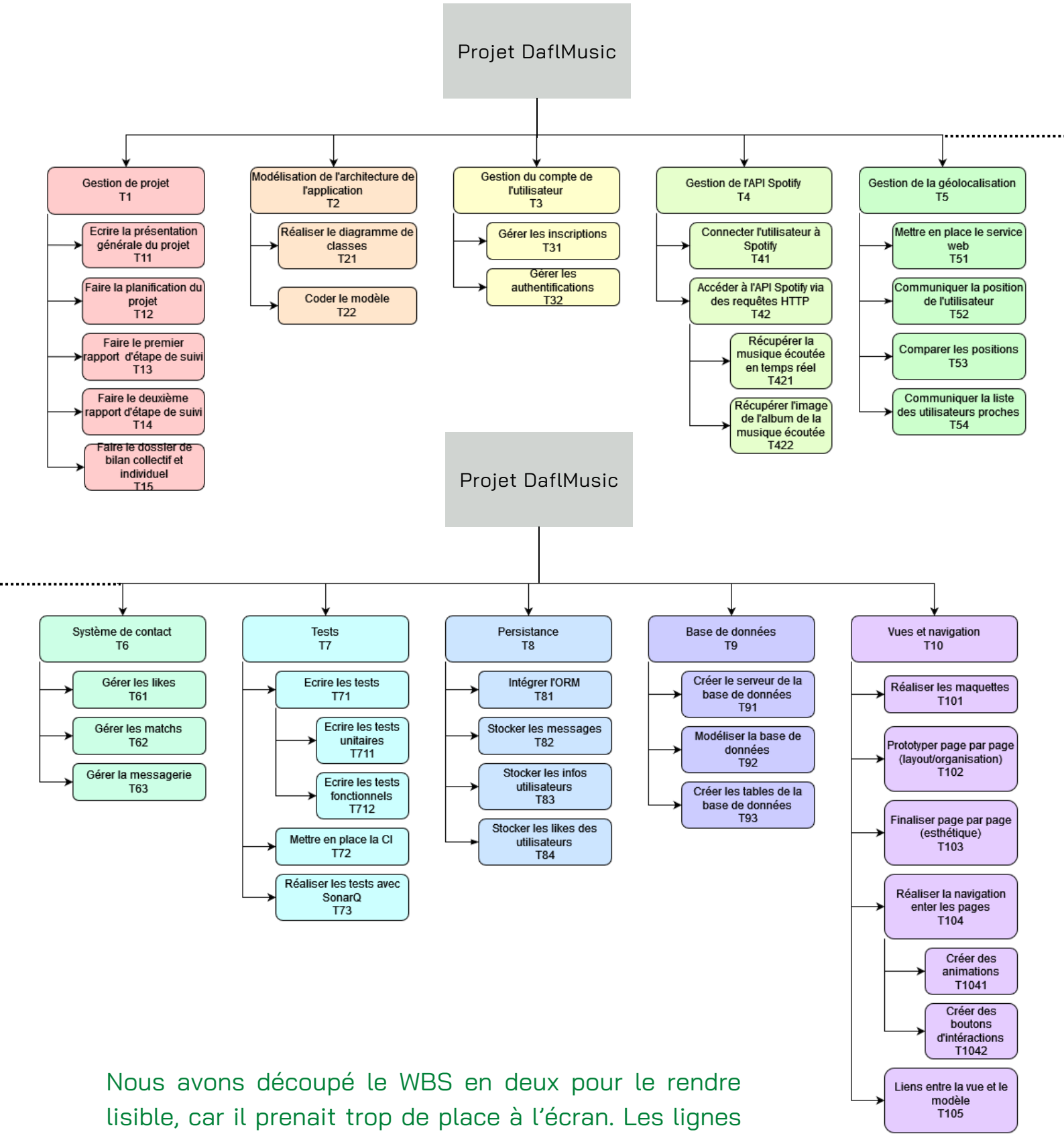
Presenté par

Audric SABATIER,
Félix MIELCAREK,
Dorian HODIN,
Lucas DELANIER

Introduction

Sommaire

1. La décomposition du projet en tâches élémentaires



Nous avons découpé le WBS en deux pour le rendre lisible, car il prenait trop de place à l'écran. Les lignes en pointillés représente la continuité d'une partie du WBS à l'autre.

Les tâches sont regroupées en différentes catégories.

1. Gestion de projet (T1) : cette catégorie de tâches ne concerne pas directement l'application mais fait partie des choses que nous avons à faire dans les 120h de projet. Chaque tâche est expliquée mais les sous-tâches ne sont pas toutes inscrites par soucis de lisibilité.

Ecrire la présentation générale du projet (T11) : c'est la conception du premier rendu avec pour date limite le 1er octobre. Elle est composée de la rédaction de la description succincte de l'idée de projet, de la genèse cette idée, de la présentation de l'équipe et de son organisation. Il y a aussi la conception des différents outils de transcription des besoins, la description des livrables ainsi que l'identification des contraintes.

Faire la planification du projet (T12) : c'est la conception du présent document, avec pour date limite le 7 octobre. Elle est composée de la conception du WBS, de l'estimation des tâches, de la durée globale et des dates jalons. De cette estimation est produit le diagramme PERT, puis la détermination du chemin critique et le GANTT prévisionnel. Ensuite une estimation des coûts prévisionnels et la détermination des indicateurs de suivi de projet et de qualité.

Faire le premier rapport d'étape de suivi (T13) : c'est la conception du troisième rendu avec pour date limite le 26 octobre. Elle est composée du tableau et son analyse des indicateurs sur les 4 semaines précédant le rendu, mais aussi la rédaction des correctifs à apporter.

Faire le deuxième rapport d'étape de suivi (T14) : c'est la conception du quatrième rendu avec pour date limite le 2 décembre. Elle est composée des mêmes étapes que la tâche T13 mais pour les 4 semaines suivantes.

Faire le dossier de bilan collectif et individuel (T15) : conception du dernier rendu et marque de la fin du projet avec pour date limite le 12 janvier. Elle est donc composée de la rédaction des bilans (individuels et collectif).

2. Modélisation de l'architecture de l'application (T2) : travaillant en méthode SCRUM, nous avons voulu respecter au mieux les 4 valeurs du manifeste agile. Pour cela, nous n'avons pas prévu la conception de tous les diagrammes que nous connaissons pouvant représenter le fonctionnement de notre application (diagramme de séquences, ...), pour favoriser la production de logiciels opérationnels à une documentation complète.

Réaliser le diagramme de classes (T21)

Coder le modèle (T22) : le modèle est composé de toutes nos classes, des fichiers de configurations, de fichiers de builds qui sont nécessaires au fonctionnement de l'application.

3. Gestion du compte utilisateur (T3)

Gérer les inscriptions (T31) : cette étape représente la création locale à l'application d'un utilisateur ainsi que l'entrée de ses données obligatoires pour le bon fonctionnement (nom d'utilisateur, ...). Elle prendra en compte la vérification de la validité des informations entrées (mot de passe conformes, ...).

Gérer les authentifications (T32) : c'est l'entrée des données d'identification, la vérification de la correspondance avec la base de données, et ensuite le chargement de ce qui sera nécessaire.

4. Gestion de l'API Spotify : c'est une API web qui fonctionne avec des requêtes HTTP/HTTPS, une documentation est fournie par Spotify sur leur site.

Connecter l'utilisateur à Spotify (T41) : associer chaque compte DAFLMusic à un compte Spotify.

Accéder à l'API Spotify via des requêtes HTTP (T42) : intégrer l'envoie des requêtes grâce à l'utilisation d'une librairie propre à notre langage.
Récupérer la musique écoutée en temps réel (T421)

Récupérer l'image de l'album de la musique écoutée (T422)

5. Gestion de la localisation (T5)

Mettre en place le service web (T51) : cette tâche consiste à déployer sur Docker un petit programme que nous contacterons avec des requêtes HTTP.

Communiquer la position de l'utilisateur (T52) : envoyer au service web la position de l'utilisateur, elle sera récupérée grâce à une librairie à intégrer.
Comparer les positions (T53) : le service web devra trier et regrouper tous les utilisateurs en fonction de toutes les localisations qu'il va recevoir.

Communiquer la liste des utilisateurs proches (T54) : notre serveur (web) renverra au client (l'application mobile) une liste des utilisateurs qu'il peut afficher, donc ceux présent dans le périmètre que nous fixerons. Chaque utilisateur déjà vu par l'utilisateur sera conservé en cache et sauvegardé, chaque profil déjà présent dans cette liste sera donc enlevé de la liste reçue par le client pour éviter qu'un profil soit visionné plusieurs fois.

6. Système de contact (T6) : représente les interactions entre les utilisateurs

Gérer les likes (T61)

Gérer les matchs (T62) : déclencher les fonctionnalités lorsque deux personnes se likent mutuellement (match).

Gérer la messagerie (T63) : créer le système de messagerie quand il y a un match mais aussi quand c'est un message spontané (la conversation sera dans la liste « En attente » du destinataire).

7. Tests (T7)

Ecrire les tests (T71) : ils prendront en compte tous les issus possibles, chercher à faire dysfonctionner le test (pour voir si certains cas provoquerons des bugs) et non pas seulement regarder s'il fonctionne dans le cas attendu.

Ecrire les tests unitaires (T711) : tests du fonctionnement des éléments interne de l'application.

Ecrire les tests fonctionnels (T712) : tests du fonctionnement des différentes fonctions de l'applications.

Mettre en place la CI (T72) : c'est l'exécution automatique des tests à chaque commit sur le repository, nous utiliserons Drone sur Code#0.

Réaliser les tests avec SonarQ (T73) : consiste à vérifier que notre code est bien optimisé (pas de duplications de code, bonne documentation, bonne couverture des tests déployées).

8. Persistance (T8) : la persistance consiste à entrer nos données dans la base de données.

Intégrer l'ORM (T81) : système permettant d'effectuer toutes nos conversations avec notre base de données directement depuis le code du modèle.

Stocker les messages (T82)

Stocker les infos utilisateurs (T83)

Stocker les likes des utilisateurs (T84) : stocker la liste des utilisateurs likés pour chaque utilisateur.

9. Base de données (T9) : ce sera une base de données PostgreSQL

Créer le serveur de la base de données (T91) : configurer les accès et sécuriser la base de données, elle sera hébergée sur Docker.

Modéliser la base de données (T92) : créer le diagramme de classe des tables, penser aux contraintes et au types de données.

Créer les tables de la base de données (T93) : où on insérera nos données.

10. Vues et navigation (T10)

Réaliser les maquettes (T101)

Prototyper page par page (T102) : prévoir les emplacements, l'organisation et les conteneurs de chaque page.

Réaliser la navigation entre les pages (104)

Créer des animations (1041)

Créer des boutons d'interactions (T1042)

Liens entre la vue et le modèle (T105) : lier les fonctions derrière chaque bouton, intégrer le binding des données.

2. Décomposition temporelle

Tâches	Durées (en H)	Antériorités	Dates au plus court		Dates au plus long		Marges
			Début	Fin	Début	Fin	
T11	16	-	0	16	30	46	30
T12	16	T11	16	32	46	62	30
T13	16	T12	32	48	62	78	30
T14	16	T13	48	64	78	94	30
T15	16	T14 - T422 - T54 - T62 - T63 - T72 - T73 - T82 - T83 - T84 - T103 - T1041 - T105	94	110	94	110	0
T21	14	-	0	14	0	14	0
T22	24	T21	14	38	14	38	0
T31	12	T81	58	70	60	72	2
T32	12	T31	70	82	72	84	2
T41	12	T22	38	50	64	76	26
T421	8	T41	50	58	76	84	26
T422	10	T421	58	68	84	94	26
T51	20	T22	38	58	38	58	0
T52	6	T51	58	64	58	64	0
T53	20	T52	64	84	64	84	0
T54	10	T53	84	94	84	94	0
T61	12	T51	58	70	62	74	4
T62	6	T61	70	76	88	94	18
T63	20	T61	70	90	74	94	4
T711	14	T22	38	52	76	90	38
T712	16	T22	38	54	74	90	36
T72	4	T711 - T712	54	58	90	94	36
T73	16	T22	38	54	78	94	40
T81	20	T22 - T93	38	58	40	60	2
T82	10	T32	82	92	84	94	2
T83	6	T32	82	88	88	94	6
T84	10	T32	82	92	84	94	2
T91	10	-	0	10	6	16	6
T92	20	T91	10	30	16	36	6
T93	4	T92	30	34	36	40	6
T101	8	-	0	8	40	48	40
T102	16	T101	8	24	48	64	40
T103	20	T102	24	44	74	94	50
T1041	10	T102	24	34	84	94	60
T1042	10	T102	24	34	64	74	40
T105	20	T22 - T1042	38	58	74	94	36

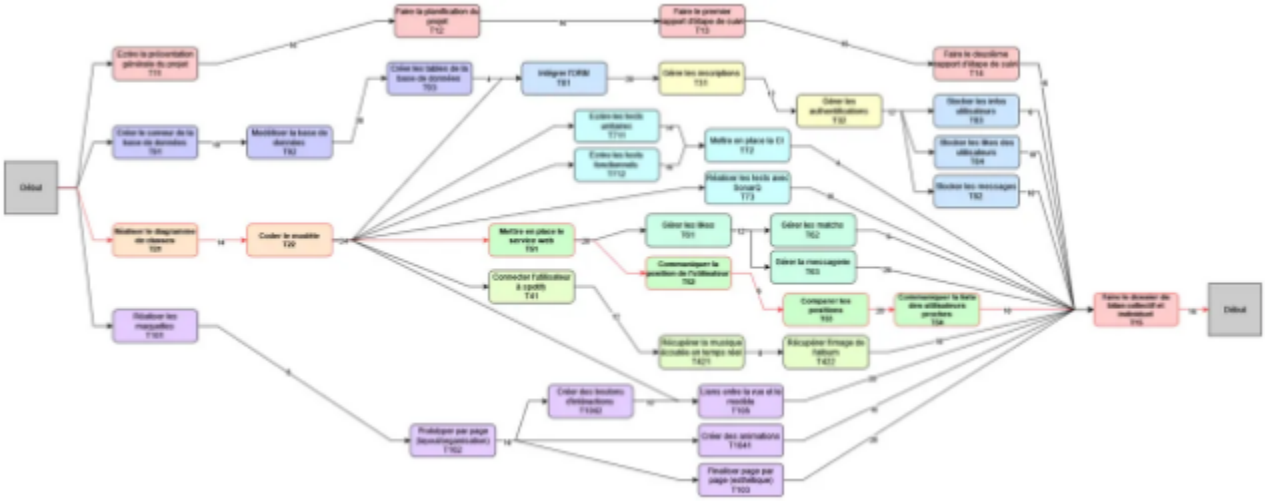
Voici le tableau regroupant l'estimation et les durées en heure de chaque tâche du WBS. La somme des durées de toutes les tâches est égale à 480 heures, soit la durée que nous possédons pour faire notre projet (120h multiplié par les 4 personnes de notre groupe).
De ces estimations et en prenant compte les antériorités, nous obtenons les dates au plus tôt et les dates au plus tard de chaque tâche, nous en déduisons donc les marges en comparant ces données. Notre projet touche à sa fin au bout de 110 heures, nous avons donc 10 heures disponibles si nos estimations ne sont pas complètement justes.

- **Dates jalons**

Pour ce projet, nous travaillons en suivant des méthodes agiles. Un des grands principes de cette méthode est l'adaptabilité aux besoins clients qui peuvent être évolutifs. Nous n'avons donc pas de dates jalon précises car les jalons sont nos releases. Ces releases sont définies avec notre tuteur en fonction des sprints que nous établissons.

3. Outils visuels d'ordonnancement

3.a. PERT

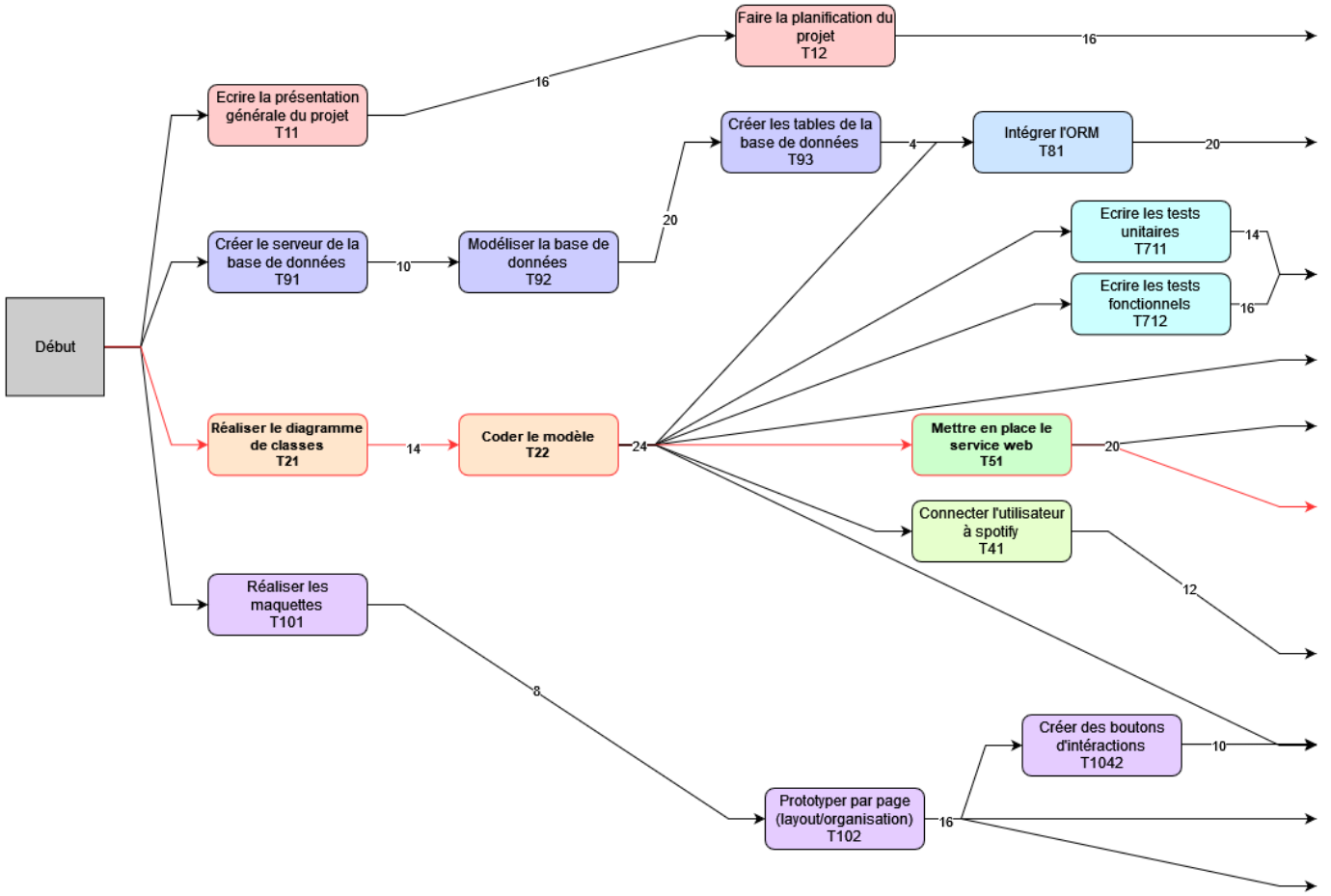


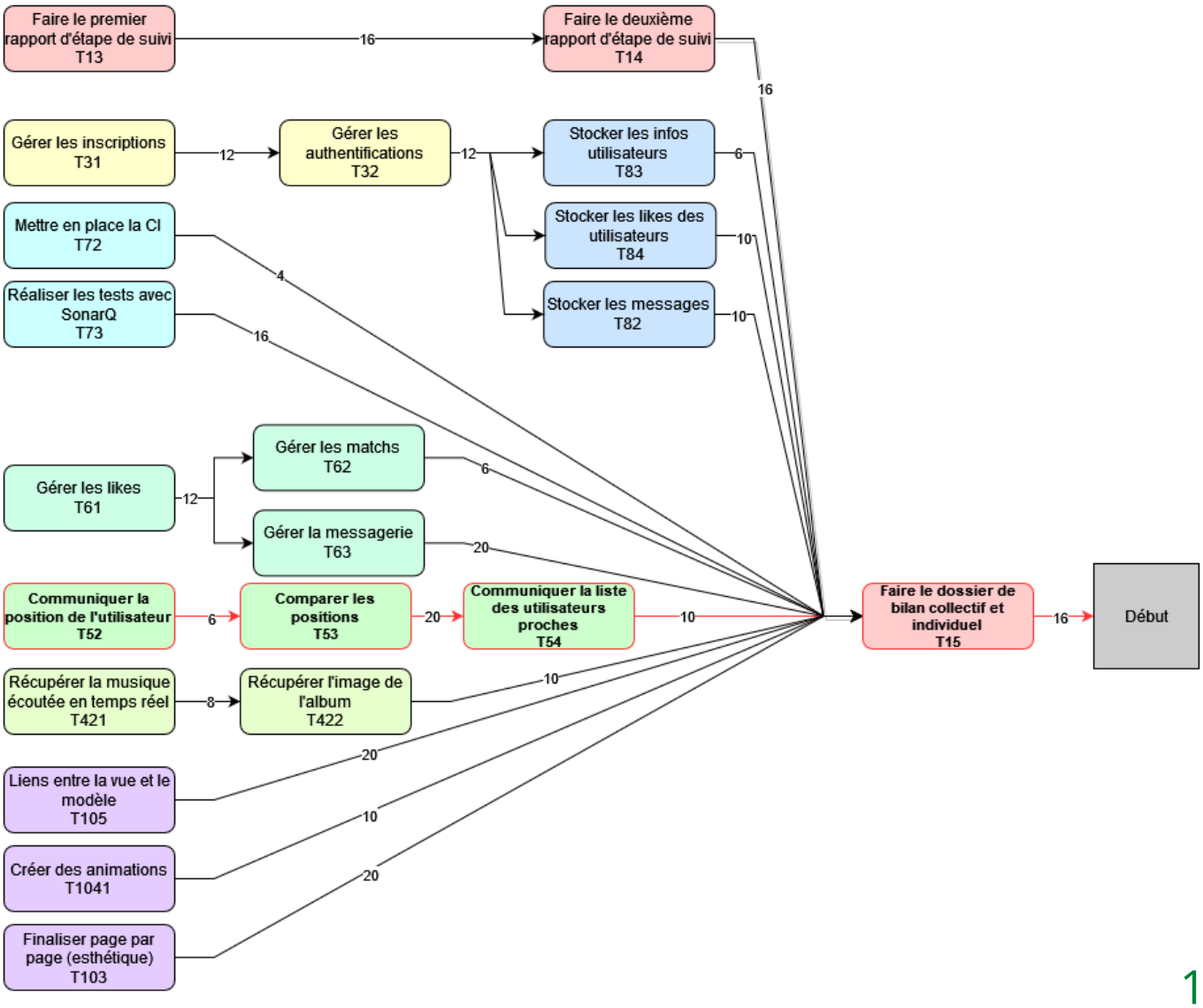
3.b. Chemin critique

Description du PERT et chemin critique

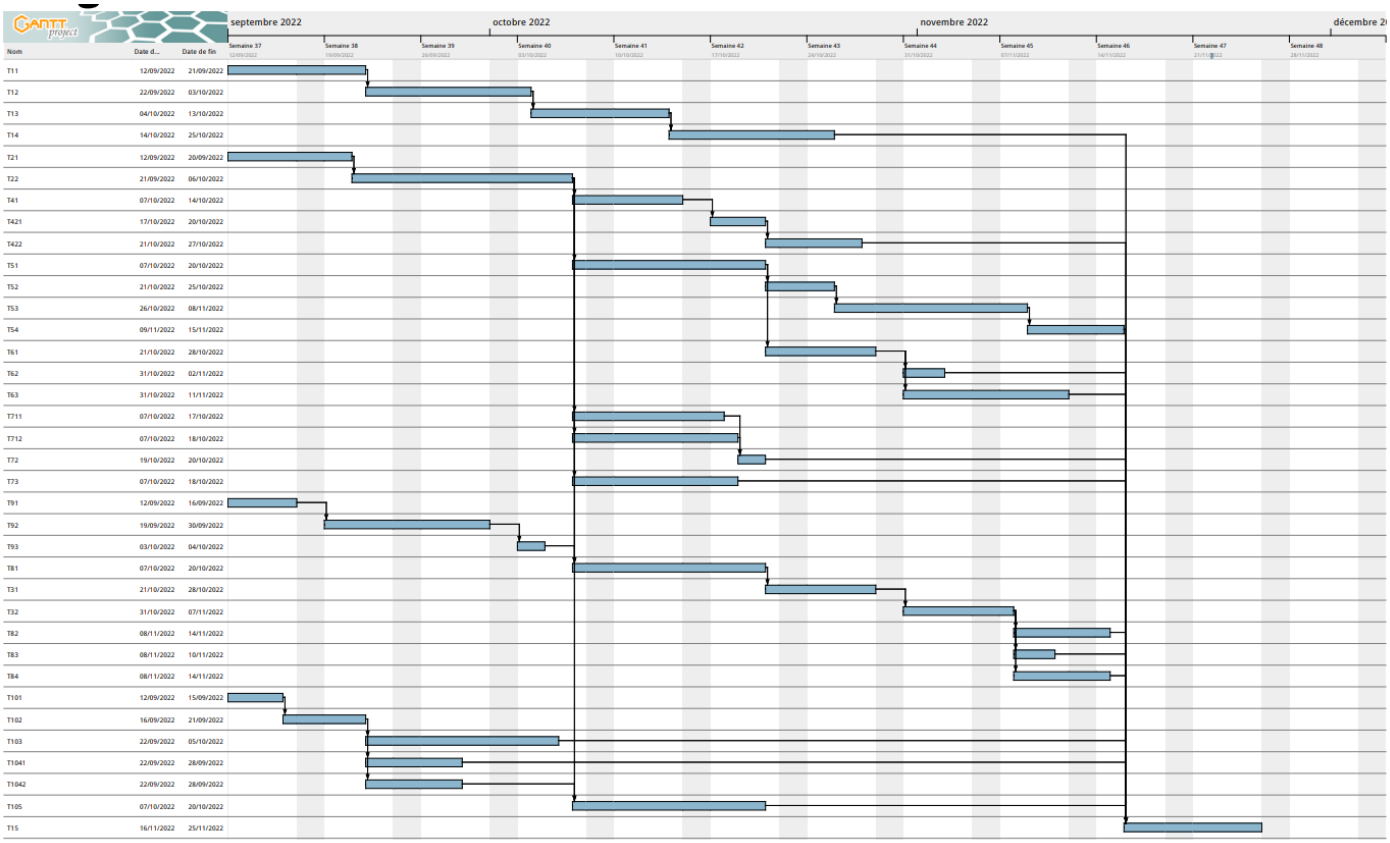
Toutes les tâches qui n'ont pas de marge disponible dans notre estimation des durées, constituent nos tâches critiques (en rouge dans le tableau). Nous pouvons donc déduire notre chemin critique : T21 - T22 - T51 - T52 - T53 - T54 - T15.

La durée de chaque tâche est écrite sur les flèches.


Début
Ecrire la présentation générale du projet T11
Faire la planification du projet T12
Créer le serveur de la base de données T91
Modéliser la base de données T92
Créer les tables de la base de données T93
Intégrer l'ORM T81
Ecrire les tests unitaires T711
Ecrire les tests fonctionnels T712
Réaliser le diagramme de classes T21
Coder le modèle T22
Mettre en place le service web T51
Connecter l'utilisateur à spotify T41
Réaliser les maquettes T101
Prototyper par page (layout/organisation) T102
Créer des boutons d'intéractions T1042
16
16
10
20
20
14
16
14
24
20
12
8
16
10


Faire le premier rapport d'étape de suivi T13
Faire le deuxième rapport d'étape de suivi T14
Gérer les inscriptions T31
Gérer les authentifications T32
Stocker les infos utilisateurs T83
Stocker les likes des utilisateurs T84
Stocker les messages T82
Mettre en place la CI T72
Réaliser les tests avec SonarQ T73
Gérer les likes T61
Gérer les matchs T62
Gérer la messagerie T63
Communiquer la position de l'utilisateur T52
Comparer les positions T53
Communiquer la liste des utilisateurs proches T54
Faire le dossier de bilan collectif et individuel T15
Début
Récupérer la musique écoutée en temps réel T421
Récupérer l'image de l'album T422
Liens entre la vue et le modèle T105
Créer des animations T1041
Finaliser page par page (esthétique) T103
16
16
12
12
6
10
10
4
16
12
6
20
6
20
10
8
10
20
10
20

3.c. GANTT prévisionnel



Voici le Gantt Prévisionnel de notre projet. Sur ce schéma nous pouvons voir les différentes tâches que nous allons réaliser ainsi que leur durées prévisionnelles.

4 .Estimation des coûts prévisionnel

- **Coût du personnel**

Pour ce projet, nous avons en personnel, une équipe de développeur d'application composé de 4 membres. Ces 4 membres sont des développeurs juniors ayant un salaire d'environ 3200€ brut par mois soit 16,46€ net par heure. Ce projet dure 120h en tout donc 480h pour l'équipe complète. Cela revient à 7900,8€ de salaire pour le projet.

- **Coût du materiel**

Afin de réaliser ce projet, nous avons besoin de 4 ordinateurs permettant de faire tourner des éditeurs de code et des pages Internet, donc des ordinateurs basiques. Nous allons compter 500€ par ordinateur, donc 2000€ de materiel pour réaliser ce projet. Nous comptons 500€ par ordinateur pour avoir un ordinateur tout de même avec un processeur puissant car nous allons faire tourner des émulateurs de téléphone pour tester l'application, il faut donc de quoi faire tourner ça sans que le PC devienne lent pour optimiser notre temps de travail.

5 .Indicateurs de suivi de projet:

Indicateurs de suivi de projet:

- **Avancement**

Taux de tâches faites sur le nombre total de tâches du projet. Cela peu se faire notamment grâce au Kanban en comptant le nombre de taches réalisées. *(tâches réalisées /tâches total x 100)*

- **Temps de travail**

Comparaison entre le temps de travail déjà réalisé et le temps de travail total du projet afin de pouvoir nous positionner par rapport à avancement prévisionnel que nous avons fait.

- **Diagramme GANTT**

Utilisation d'un diagramme pour visualiser le début et la fin des taches pour organiser les taches et vérifier l'avancement du projet.

- **Ecart entre temp planifié et réel**

Calcul du temps de retard des échéances en nous basant sur les durées du PERT-TEMPS

Indicateurs de qualité:

- **Le nombre d'erreurs**

Proportion de bug/ erreurs par rapport aux nombres de tâches réalisé. Permet de savoir si notre code est satisfaisant en terme de qualité.

- **Satisfaction des tuteurs**

Prise en compte des retours que nous font nos tuteurs en comptant le nombre de taches insatisfaisante. Particulièrement ceux fait lors de nos fin de sprint par Mr. Provot afin de savoir si nous fournissions un travail de qualité.