



Journal du projet [EquiWatch]

RESUME :

La société PONYCORNE propose une nouvelle façon de surveiller l'état des équidés des centres équestres et particuliers par le biais d'une application Android

EQUIPE :

L3 – 2018-2019 – GEQUIWATCH
EMERY Yann, GOUGEON Nolan, NEXON Sébastien

DATE :

19/07/2019

Table des matières

I. INITIALISATION du projet.....	3
I.1 Organisation équipe	3
I.2 L'idée de départ	3
I.3 Motifs (justifications de l'idée)	3
I.4 Validation via veille concurrentielle	3
I.5 Estimation de la complexité technique	3
II. PREPARATION du projet	4
II.1 Description du projet	4
II.2 Intégration	4
II.3 Conditions de réussite	4
II.4 Echancier (macro-planning)	6
II.5 Gestion des risques	6
II.6 Organisation	6
II.7 Représentation des contraintes techniques	7
II.8 Description des environnements	8
II.9 Les livrables du projet	9
III. PLANNING initial du projet.....	9
III.1 Votre planning de référence	9
III.2 Vos plans de tests (tests unitaires & tests de charge)	9
III.3 Vos outils.....	9
Votre outil de traçabilité	9
Votre mécanisme de build	9
Votre outil d'analyse et suivi qualité	9
Votre outil de versionning	9
IV. PILOTAGE du projet	10
IV.1.1 Période concernée.....	10
IV.1.2 Planning de référence vs. Planning de suivi, pour la période concernée	10
IV.1.3 Analyse et explication des écarts.....	10
IV.1.4 Trace des risques avérés et/ou des changements opérés	10
IV.2.1 Période concernée.....	10
IV.2.2 Planning de référence vs. Planning de suivi, pour la période concernée	10
IV.2.3 Analyse et explication des écarts.....	10

IV.2.4 Trace des risques avérés et/ou des changements opérés	11
IV.3.1 Période concernée.....	11
IV.3.2 Planning de référence vs. Planning de suivi, pour la période concernée	11
IV.3.3 Analyse et explication des écarts.....	11
IV.3.4 Trace des risques avérés et/ou des changements opérés	11
IV.4.1 Période concernée.....	11
IV.4.2 Planning de référence vs. Planning de suivi, pour la période concernée	12
IV.4.3 Analyse et explication des écarts.....	12
IV.4.4 Trace des risques avérés et/ou des changements opérés	12
IV.5 Métriques.....	12
Sur l'utilisation de l'outil de versionning.....	12
Sur l'utilisation de l'outil de d'analyse qualité	12
Sur les résultats de vos tests	12
Sur l'outil de traçabilité	12
V. BILAN du projet.....	13



I. INITIALISATION du projet

I.1 Organisation équipe

ITESCIA M2I L3 DEV 1
EMERY Yann : développeur.
GOUGEON Nolan : développeur.
NEXON Sébastien : développeur et chef de projet.
Société Ponycorne Stéphanie HOUART Directrice

I.2 L'idée de départ

Développement d'activité pour la gestion équine des particuliers et des centres équestres.
Le projet se situe sur le marché de la géolocalisation des chevaux et gestion des enclos.

I.3 Motifs (justifications de l'idée)

Certain cavaliers, propriétaires et centres équestres place leurs équidés dans des prés loin de chez eux et souhaitent savoir l'état de l'enclos et si les équidés ne se sont pas enfuis.
Produit en direction des particuliers voulant savoir l'état de la clôture, la quantité d'eau, la température et si leurs équidés sont encore dans leurs enclos.
Le système devra être accessible à tous en termes de prix, le coût réel n'a pas encore été fixé.

I.4 Validation via veille concurrentielle

Concurrence :

- Digitanimal :
Points forts :
 - Localisation GPS 24/24.
 - Une application mobile et pc.
 - Alerte lorsque le cheval sort de la zone qui lui est attribuée.Points faibles :
 - Coûts importants : 150€ + abonnement au bout d'un an.
 - Collier pouvant être facilement abimé
- Secutec TR500COW-SOLAR :
Points forts :
 - Sans abonnement.
 - Batterie autonome grâce à des panneaux solaires.
 - WaterproofPoint faible :
 - Coût initial très élevé (300€)

Valeur ajoutée de notre projet :

EQUIWATCH a pour but d'être plus accessible aux particuliers et moins cher.
Le projet permettra également de vérifier, à l'aide de capteurs, si le courant passe toujours dans la clôture, la température dans l'enclos et s'il y a une quantité suffisante d'eau dans l'abreuvoir.

I.5 Estimation de la complexité technique

Le projet est relativement complexe il demande l'utilisation d'api et de nombreux langages.
Nous utiliserons un langage pour androidOs, Python et Bash pour le capteur Raspberry et un langage de haut niveau.

Les technologies suivantes seront utiles pour le projet :

- Capteur hydraulique, capteur électrique, capteur géolocalisation, capteur thermique
- Technologie Android

- Langages : Python, Java, Bash
- Raspberry PI 2 Model B V1.1

Nous aurons besoin de nous former sur les outils liés à Android, Raspberry.

II. PREPARATION du projet

II.1 Description du projet

Acteurs du projet :

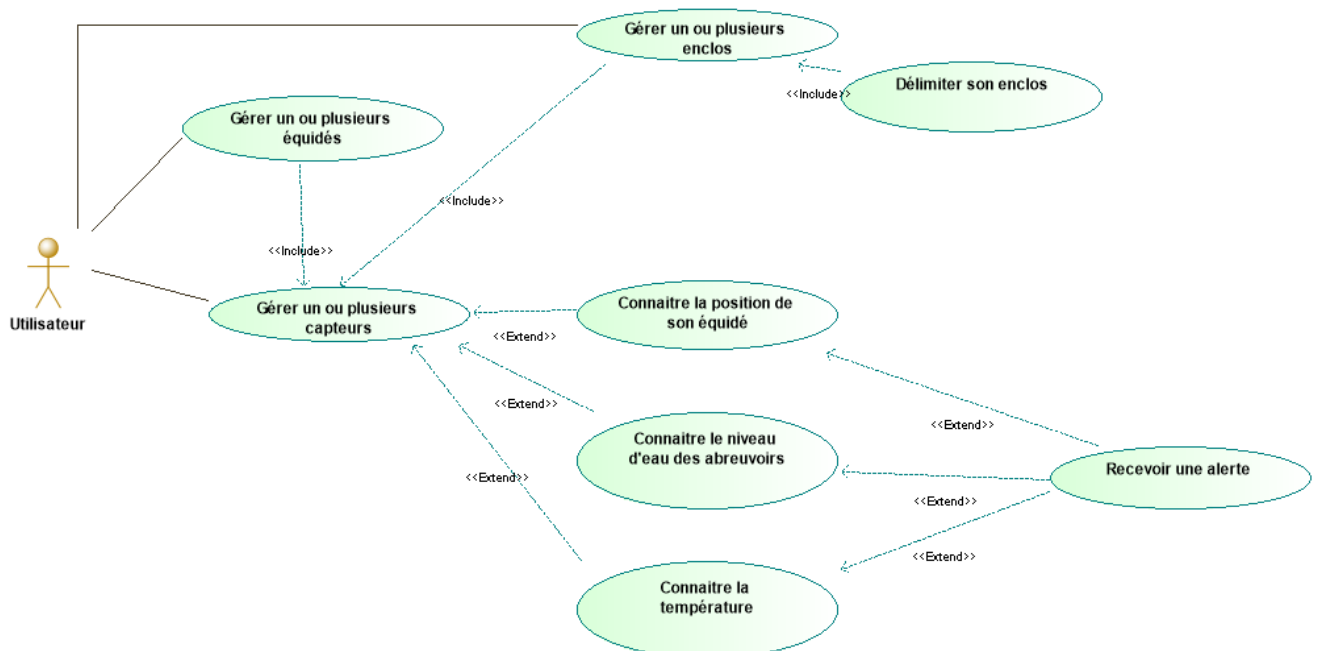
- Utilisateur : Il pourra connaître la position de ses équidés ainsi que le niveau d'eau de leurs abreuvoirs, l'état du courant électrique de sa clôture et la température.

Fonctionnalités attendues :

- L'utilisateur peut voir sa position ainsi que la position de ses équidés sur une carte.
- L'utilisateur saura lorsque que son équidé quitte son enclos.
- L'utilisateur pourra connaître le niveau d'eau de ses abreuvoirs.
- L'utilisateur pourra connaître la température dans son enclos.

Contraintes éventuelles :

- L'application sera disponible uniquement sur Android.



II.2 Intégration

Nous utilisons une interface utilisateur simple avec une page de connexion la première fois qu'on lance l'application (utilisateur pré connecté par la suite) et un menu permettant de gérer les différents CRUD (Create Read Update Delete (Création, Vue, Modification Suppression)). La gestion des utilisateurs notamment des mots de passe sont cryptés.

II.3 Conditions de réussite

Afin que le projet soit considéré comme réussi il faut que :

- L'utilisateur puisse délimiter son enclos à partir de sa position GPS.

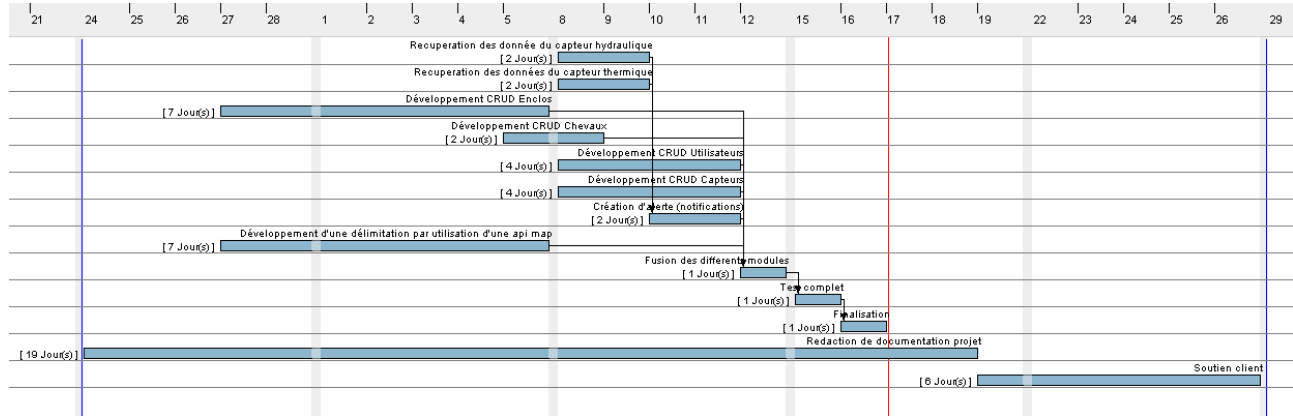


Document : JOURNAL DE PROJET

Réf. :
JOURNAL_PROJET_EQUIWATCH_V4.0
Version : 4
Date : 19/07/2019

- La position GPS des équidés soit connue.
- Une alerte doit être envoyée si :
 - L'équidé sort de son enclos.
 - Le niveau d'eau de l'abreuvoir est trop bas.
 - S'il n'y a plus de courant électrique dans la clôture
 - S'il fait trop chaud dans l'enclos.

II.4 Echancier (macro-planning)



II.5 Gestion des risques

Nature de risque	Description	Gravité /4	Actions préventives	Actions correctrices
Humain	Un membre de l'équipe rencontre des problèmes de santé	3	Commenter le code développé et utiliser un logiciel de gestion de version pour récupérer le code de la personne en question	Prévoir plus de temps pour la conception de l'application en raison de la réduction de l'effectif
Technique	Les machines nécessaires au projet sont dysfonctionnelles	4	Effectuer des sauvegardes récurrentes ou utiliser un outil de gestion de versions décentralisé	Changer de machine et récupérer les données si possible

Le projet comporte des risques potentiels liés au matériel ou aux erreurs humaines.

Les risques techniques / matériels sont les suivants :

- Dysfonctionnement du matériel. Une vérification constante de l'état du matériel devrait être effectuée pour garantir le bon déroulement du projet.
- Pour prévenir les risques liés à la perte des données plusieurs sauvegardes présentes sur plusieurs supports seront effectuées ou alors la mise en commun récurrente des données par le biais d'un outil extérieur comme GIT.

II.6 Organisation

L'équipe est composée de trois développeurs dont un chef de projet. Les tâches de développement ont été réparties entre tous, le chef de projet a des tâches de difficulté moindre pour lui permettre d'organiser le projet et de faciliter le travail du reste de l'équipe.

La communication en interne est gérée à l'aide de l'outil de communication Discord qui permet de partager de manière textuelle des documents et des informations à toute l'équipe et aussi

de faire des conférences vocales.

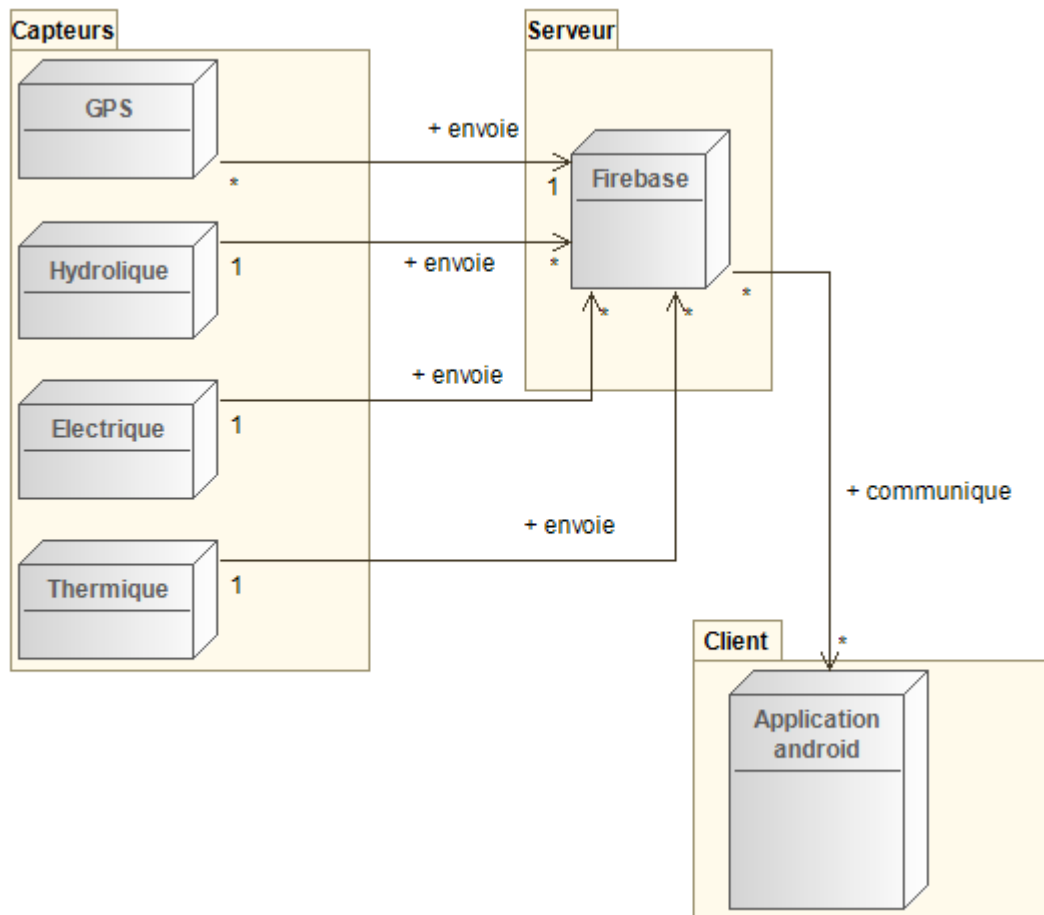
La gestion documentaire est gérée par l'application OneDrive qui permet l'utilisation en « cloud » des documents. Cela permet à toute l'équipe de pouvoir accéder à la dernière version des documents et si besoin de les modifier en gardant toute l'équipe à jour.

La répartition des tâches et l'avancement du projet sont gérés par l'application Trello qui nous permet de créer des cartes qui correspondent à une tâche et d'affecter un développeur dessus. Nous pouvons donc suivre l'avancement du projet en sachant ce qui est fait et ce qu'il reste à faire.

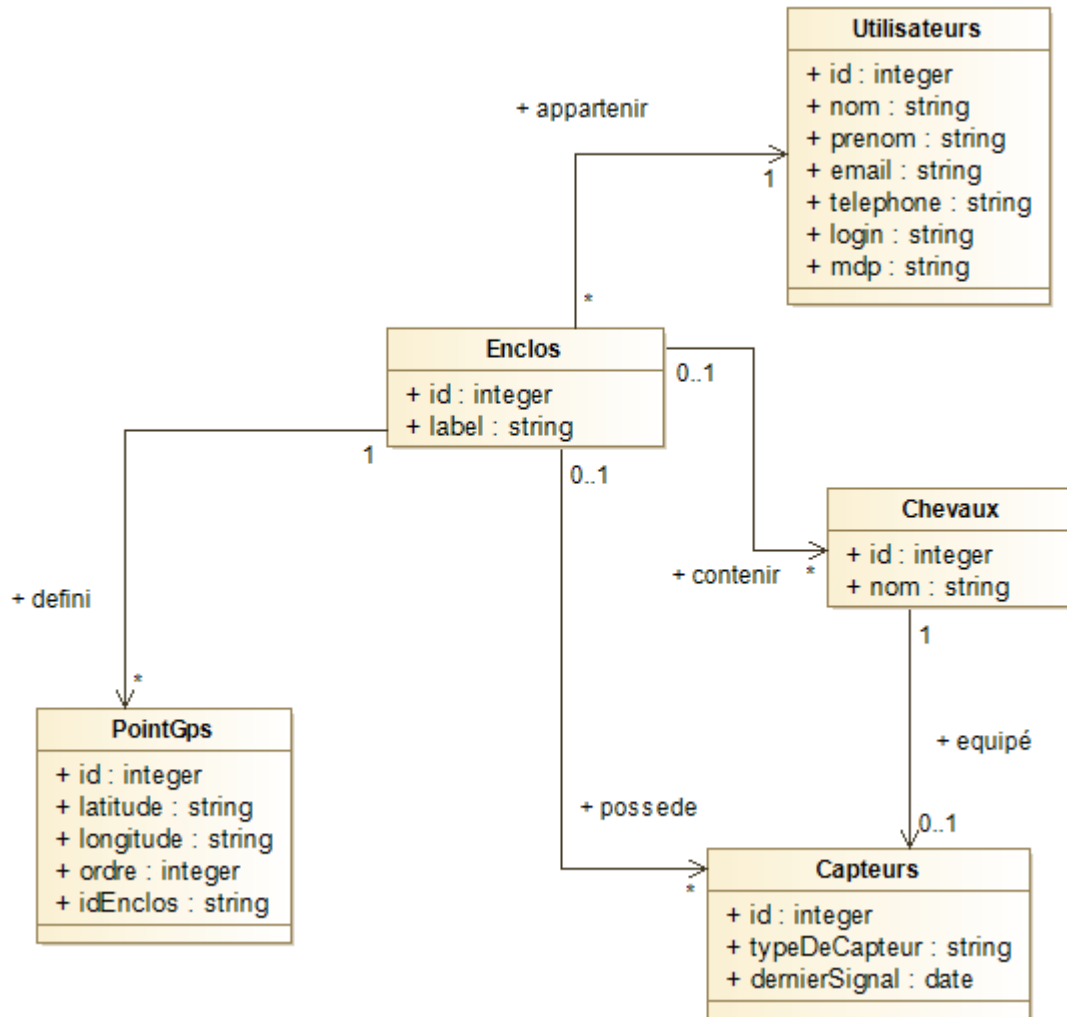
Nous avons assisté à 4 revues de projet.

II.7 Représentation des contraintes techniques

Le projet est en deux 2 modules (front + back).



Il y a plusieurs couches dans notre projet, il a une architecture MVC et contiens les classes suivantes.



II.8 Description des environnements

Environnement de développement : Android Studio 3.4.1

Utilisation d'émulateur Android pour les tests.

Cible production : Android.

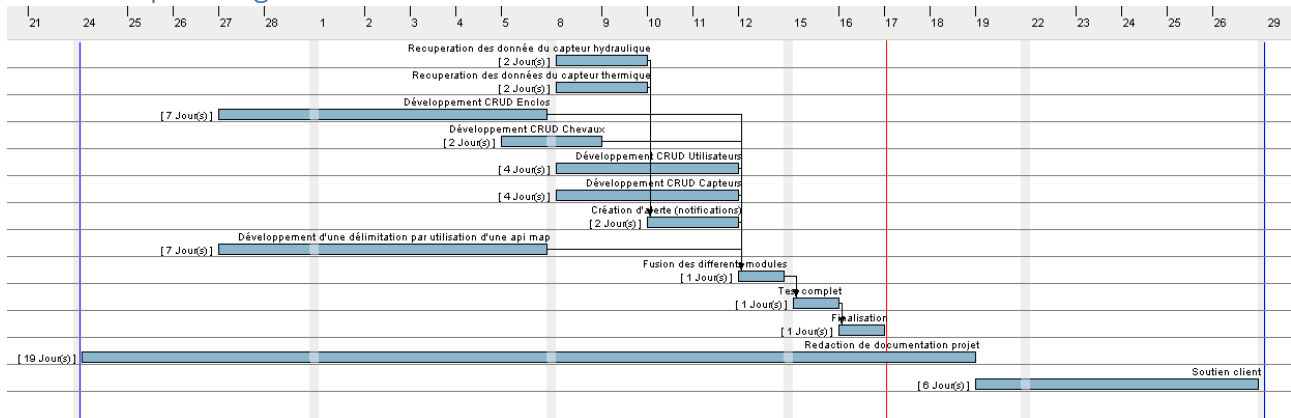
II.9 Les livrables du projet

Repository git : <https://github.com/NexonSebastien/equiwatch/tree/master>

Les livrables se trouvent dans le dossier equiwatch_documentation

III. PLANNING initial du projet

III.1 Votre planning de référence



III.2 Vos plans de tests (tests unitaires & tests de charge)

Pour effectuer nos tests unitaires nous utilisons JUnit.

Pour les tests d'interface utilisateur, nous utilisons Espresso.

III.3 Vos outils

Votre outil de traçabilité

Pour la gestion des anomalies, nous avons choisi Trello.

Tous les développeurs en ont la charge.

Lorsque qu'un bug doit être remonté, on crée une carte dans la colonne Bug à laquelle on ajoute une description et la façon dont on peut reproduire ce bug si possible.

Le statut du bug est défini par la colonne dans laquelle la carte qui lui est liée est placée, soit "à faire" si il est dans la colonne bug soit "en cours" soit "terminé".

Votre mécanisme de build

Notre livrable est build par Gradle.

AndroidStudio utilise Gradle nativement.

Votre outil d'analyse et suivi qualité

Nous avons installé SonarLint.

SonarLint scan notre code afin de nous faire remonter certaines mauvaises pratiques de codage ou des erreurs pouvant provoquer des fuites de mémoires.

Votre outil de versionning

Nous avons choisi d'utiliser un repository GIT hébergé par GitHub.

La branche master contient la dernière version du code stable de l'application en production.

La branche devMaster contient la dernière version stable du code en développement, on y ajoute les dernières fonctionnalités.

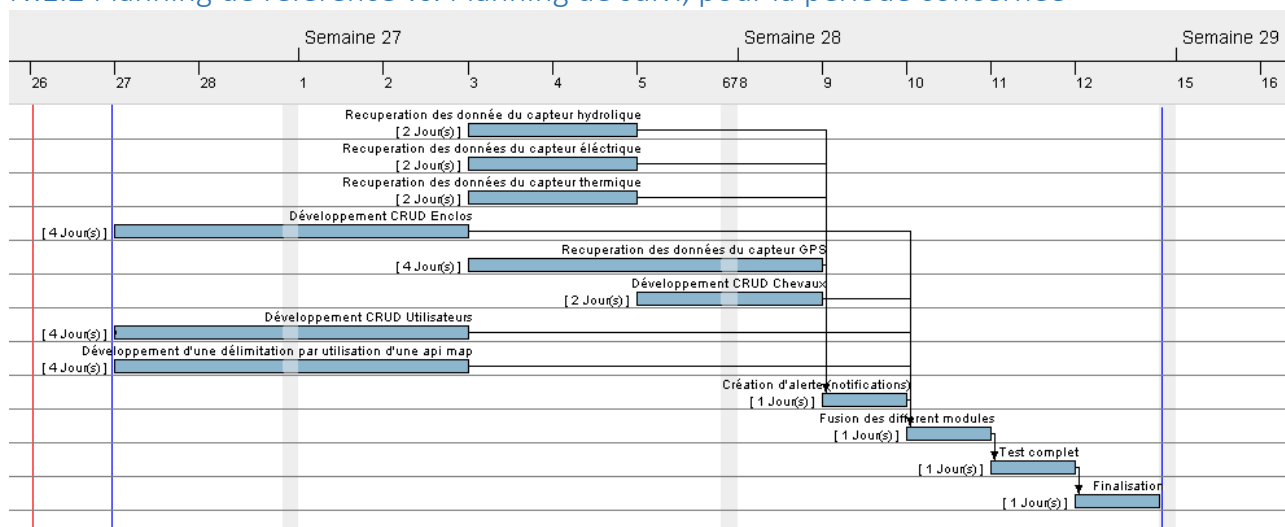
Lorsque qu'une nouvelle fonctionnalité est créée, on crée une nouvelle branche à partir de devMaster que l'on mergera ensuite sur devMaster lorsque la fonctionnalité sera créée.

IV. PILOTAGE du projet

IV.1.1 Période concernée

Date de début : 24/06/2019 Date de fin 27/06/2019

IV.1.2 Planning de référence vs. Planning de suivi, pour la période concernée



IV.1.3 Analyse et explication des écarts

Il n'y a pas d'écart constaté

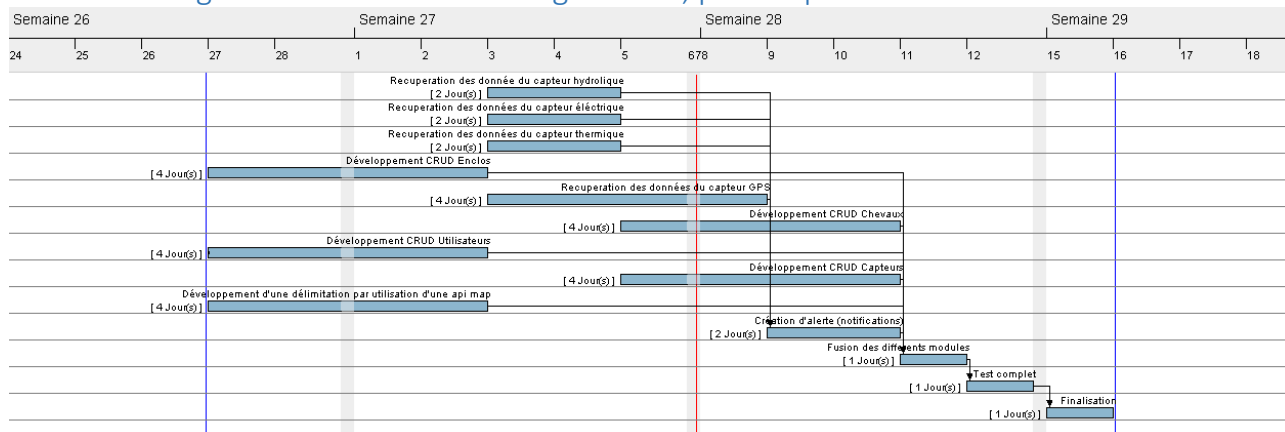
IV.1.4 Trace des risques avérés et/ou des changements opérés

Problème sur le choix du matériel chef de projet et un développeur mobilisé.

IV.2.1 Période concernée

Date de début : 27/06/2019 Date de fin 4/07/2019

IV.2.2 Planning de référence vs. Planning de suivi, pour la période concernée



IV.2.3 Analyse et explication des écarts

Retard sur le développement à la suite de l'initialisation et la mise en place du MVC et de toute

l'architecture de l'application pour gagner du temps par la suite

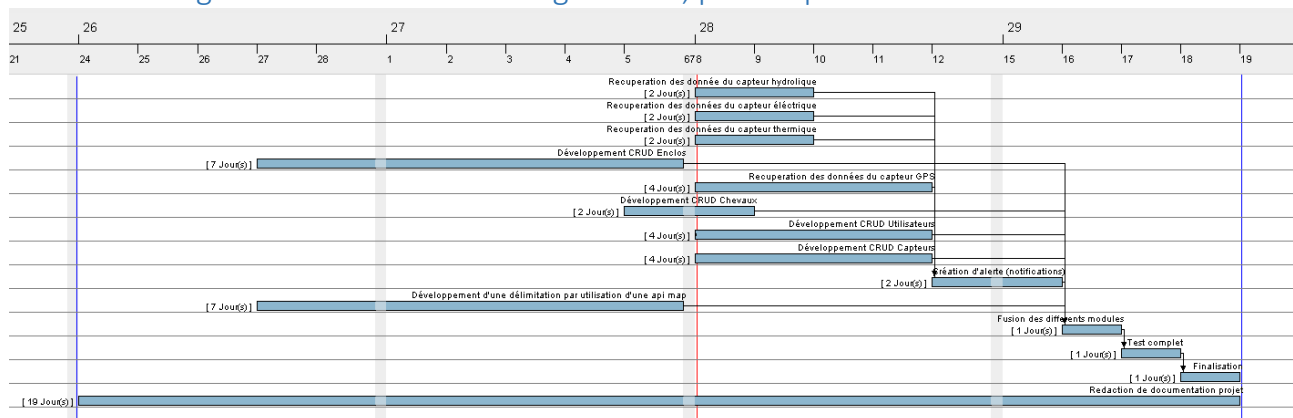
IV.2.4 Trace des risques avérés et/ou des changements opérés

Nous avons donc effectué une modification dans l'ordre des tâches et une réaffectation de certaines. Risque d'une augmentation et d'un retard potentiel si ce temps pris en début de projet n'est pas rentable par la suite

IV.3.1 Période concernée

Date de début : 4/07/2019 Date de fin 11/07/2019

IV.3.2 Planning de référence vs. Planning de suivi, pour la période concernée



IV.3.3 Analyse et explication des écarts

Problème matériel qui retarde le début de la récupération des capteurs.

Problème rencontré dans le code qui a entraîné des changements majeurs dans l'architecture du projet.

IV.3.4 Trace des risques avérés et/ou des changements opérés

Risque matériel à gros impact, toute l'équipe est mobilisée pour trouver une solution.

Changement d'architecture, toute l'équipe mobilisée.

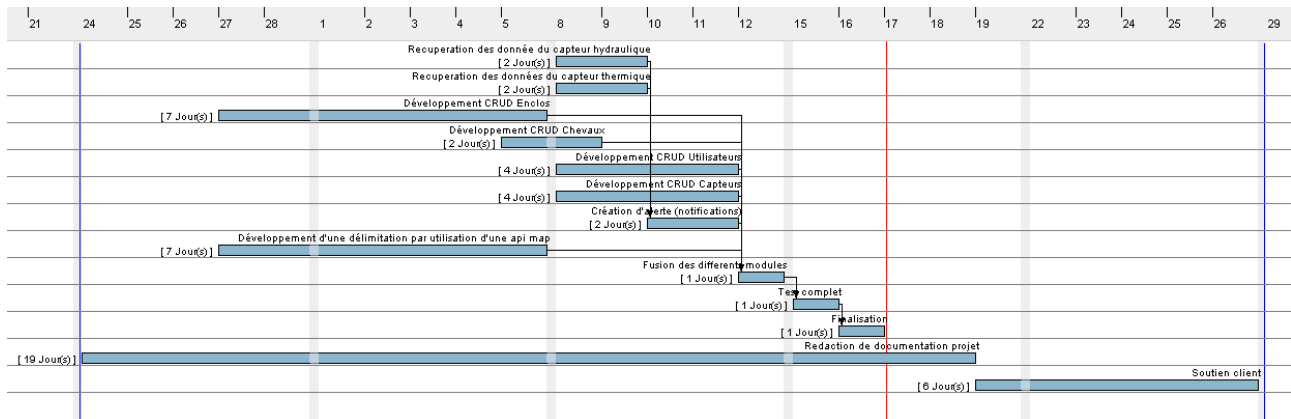
Des risques de retard sont à prévoir car la récupération du capteur GPS se trouve désormais sur le chemin critique du projet.

L'équipe a donc anticipé et commencé une formation préventive pour anticiper l'arrivée de celui-ci.

IV.4.1 Période concernée

Date de début : 11/07/2019 Date de fin 19/07/2019

IV.4.2 Planning de référence vs. Planning de suivi, pour la période concernée



IV.4.3 Analyse et explication des écarts

De nombreux retard notamment avec les capteurs et différent Bug applicatif retarde la fin du projet

IV.4.4 Trace des risques avérés et/ou des changements opérés

Incapacité d'acquérir en temps voulu les compétences en électrotechnique

Incapacité d'utiliser le capteur GPS

Risque de retard et de fonctionnalité absente

L'équipe s'est donc concentrée sur le reste de l'application

IV.5 Métriques

Sur l'utilisation de l'outil de versionning

Tout au long du projet, 16 branches ont été créées en comptant Master et devMaster.

Il y a eu 135 commit au total :

- 74 commit ont été push par Nolan Gougeon
- 36 commit ont été push par Sébastien Nexon
- 7 commit ont été push par Yann Emery

Sur l'utilisation de l'outil de d'analyse qualité

SonarLint a été mis en place lors de la 3-ème semaine de développement.

La qualité du code s'est grandement améliorée depuis.

Sur les résultats de vos tests

100% des test créés réussissent actuellement si tous les prérequis sont respectés.

Actuellement les tests couvrent 3% du code de l'application.

Sur l'outil de traçabilité

Il y avait 9 anomalies.

3 évolutions ont été réalisés pour corriger 3 anomalies :

- Le passage de toute nos vues en Activity au lieu de fragment.
- Abandon d'un serveur local WAMP pour une solution cloud, Firestore.
- Changement de version d'Android.

Le membre de l'équipe ayant fermé le plus d'anomalies est Nolan Gougeon, vient ensuite Sébastien Nexon et Yann Emery.

Il reste une anomalie connue encore ouverte.

V. BILAN du projet

Le projet nous a permis d'acquérir de nouvelles connaissances et une meilleure aisance en termes de gestion de projet.

Nous avons réussi à assimiler les bases du développement en Android rapidement.

Les problèmes de matériel ont freiné et retardé grandement le projet, à l'avenir nous devons revoir ce point pour nous y prendre d'une meilleure façon.

Des problèmes de répartition du temps ont aussi fait surface notamment sur la partie documentation que nous avons sous-estimée.

Lors de ce projet notre début de formation sur les différents outils demandés pour le projet a été commencé trop tard, nous aurions dû les débiter plus en amont.

Si nous pensons que quelque chose n'est pas logique à faire de telle façon nous devons être force de proposition et expliquer pourquoi il ne faudrait pas le faire comme cela mais plus de telle façon.