

Cahier des charges

Startup P8

Conception et réalisation plateforme collaborative



plateforme de développement collaborative

Méthode Agile

Équipe du projet :

- Kenza BESSACI
- Lila ACHOUR
- Youcef ARAB

Table des matières

1	La présentation de l'équipe de travail	4
2	Présentation du projet	4
2.1	Intitulé du projet	4
2.2	Présentation générale	4
2.3	Cadre du projet / Etude de l'existant	4
2.4	Problématique	5
2.5	Solution proposée	5
2.6	Langage et méthodologie de conception	5
2.6.1	La méthode Agile	5
2.6.2	La méthodologie SCRUM	6
2.6.3	Le langage de modélisation UML	6
3	Analyse & Spécification détaillée du projet	7
3.1	Capture des besoins	7
3.2	Identification des acteurs	7
3.3	Les besoins fonctionnels	8
3.4	Les besoins non fonctionnels	9
3.5	Planning du traitement des cas d'utilisation	10
3.5.1	Étude des cas d'utilisation du sprint 1	10
3.5.2	Description du cas	11
3.5.3	Diagramme de cas d'utilisation système	11
3.5.4	Description du cas	11
3.5.5	Diagramme de cas d'utilisation système	12
3.5.6	Étude des cas d'utilisation du sprint 2	12
3.5.7	Description du cas	12
3.5.8	Diagramme de cas d'utilisation système	12
3.5.9	Étude des cas d'utilisation du sprint 3	13
3.5.10	Description du cas	13
3.5.11	Diagramme de cas d'utilisation système	14
3.5.12	Description du cas	14
3.5.13	Diagramme de cas d'utilisation système	14
3.5.14	Priorité	15
3.5.15	Technique d'estimation agile : Big / Uncertain / Small	15
3.6	Diagramme de Gantt	15
3.7	Prototypage des interfaces	16
3.8	Pilotage du projet avec SCRUM	17
3.9	Les outils SCRUM	18
3.10	Équipe et rôles	19

3.11	Déroulement de scrum	20
3.12	Le backlog du produit	21
3.13	Planning des sprints	23
4	Étude et réalisation du sprint -1-	24
4.1	Spécification fonctionnelle & non fonctionnelle du sprint . . .	24
4.2	Étude des cas d'utilisation du sprint 1	24
4.3	Conception du Sprint 1	24
4.3.1	Diagramme de classe	24
4.3.2	Diagramme de séquence	25
4.4	Planification des Tests et validation	25
5	Étude et réalisation du sprint -2-	26
5.1	Spécification fonctionnelle & non fonctionnelle du sprint . . .	26
5.2	Conception du Sprint 2	27
5.2.1	Diagramme de classe	27
5.2.2	Diagramme de séquence	28
5.2.3	Diagramme de séquence	29
5.2.4	Diagramme de séquence	29
5.3	Planification des Tests et validation	30
6	Étude et réalisation du sprint -3-	30
6.1	Spécification fonctionnelle & non fonctionnelle du sprint . . .	30
6.2	Conception du Sprint 3	32
6.2.1	Diagramme de séquence	32
6.3	Planification des Tests et validation	35
7	Étude et réalisation du sprint -4-	36
7.1	Spécification fonctionnelle & non fonctionnelle du sprint . . .	36
7.2	Étude du sprint 4	36
7.3	Planification des Tests et validation	37
8	Burndown chart	37
8.1	Burndown chart sprint 1	38
8.2	Burndown chart sprint 2	38
9	Rétrospective	39
10	Conclusion	39
11	Perspective	40

1 La présentation de l'équipe de travail

Pour la réalisation de ce projet intitulé « PLateforme développement collaborative », Une startup nommée "P8", spécialisée dans le développement de solution infomatique, est mise en place en collaboration avec trois personnes :

- Kneza BESSACI : Master 1 Informatique
- Lila ACHOUR : Master 1 Informatique
- Youcef ARAB : Master 1 Big data

2 Présentation du projet

2.1 Intitulé du projet

Conception et réalisation d'une plateforme de développement collaboratif des applications métiers d'entreprise en se basant sur les méthodes agiles et plus précisément sur le framwork Scrum.

2.2 Présentation générale

Le logiciel collaboratif est devenu un facteur primordiale pour toutes les entreprises, afin de gérer au mieux leur projet de développement et cela en intégrant un environnement sain est idéal pour ses développeurs et leur limiter tout interruption inutile pendant leurs travail pour gagner en temps et en performance.

2.3 Cadre du projet / Etude de l'existant

Differentes plateformes de planification ou de gestion de taches existent déjà tél que :

- * monday.com
- * Jira
- * Trello
- * etc...

néanmoins il n'existe aucune plateforme spécialisée, dédié a la gestion de projet d'une manière complète en se basant sur la méthode agile.

2.4 Problématique

Les entreprises sont confrontées à des problèmes de collaboration et de coordination entre les acteurs du projet (maître d'ouvrage, maître d'œuvre) dans son cycle de vie (de définition des besoins jusqu'au déploiement et la maintenance).

2.5 Solution proposée

Afin de pouvoir mettre en place un environnement sain qui encourage et facilite le travail au sein d'une entreprise, une solution est proposée qui consiste à un logiciel / plateforme collaborative interne qui permettra de créer les différents comptes spécialisés pour le product owner, scrum master ainsi qu'aux développeurs et leur fournir tous les outils nécessaires pour la bonne gestion de chaque tâche propre au métier.

2.6 Langage et méthodologie de conception

2.6.1 La méthode Agile

L'agilité est la capacité de s'adapter au changement et d'y répondre, les organisations agiles voient le changement comme une opportunité non comme une menace. Devenir agile c'est d'être en mesure de créer des produits qui procurent régulièrement de la valeur pour le client tout en répondant favorablement aux demandes de changement dans un environnement incertain. La méthodologie Agile se base sur ce principe simple : planifier la totalité de votre projet dans les moindres détails avant de le développer est contre-productif.

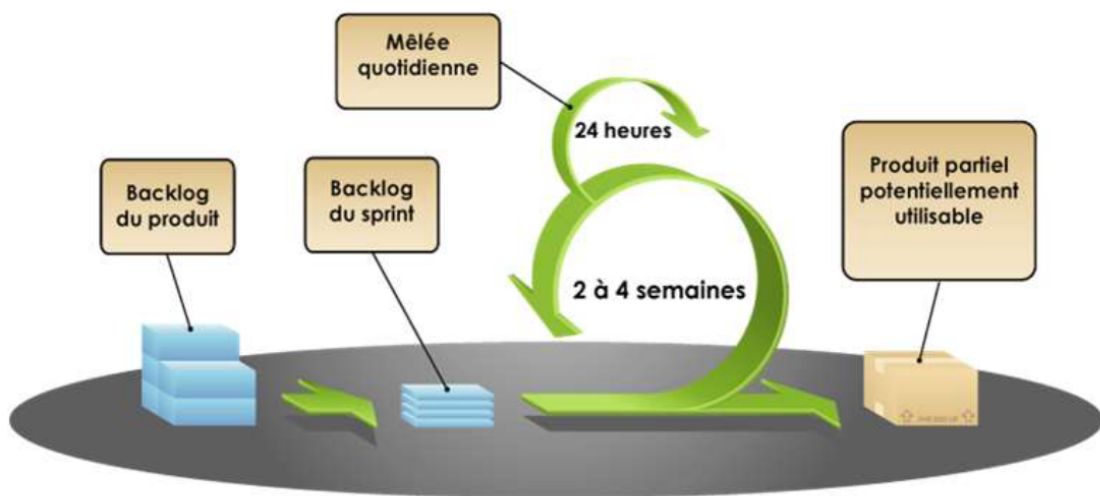
Les principes de la méthode agile sont :

- 1 - La satisfaction du client est primordiale.
- 2 - Livrer rapidement
- 3 - Accueille positivement les changements de besoins même tardif dans le développement.
- 4 - Livrez fréquemment un logiciel opérationnel
- 5 - Les utilisateurs et les développeurs doivent travailler ensemble quotidiennement tout au long du projet.

2.6.2 La méthodologie SCRUM

Scrum est une manière d'organiser et de gérer le travail dans lequel les acteurs peuvent aborder des problèmes complexes et adaptées, en livrant de manière efficace et créative des produits de la plus grande valeur possible.

ci-dessous le schéma du framwork Scrum :



2.6.3 Le langage de modélisation UML

UML est un langage de modélisation graphique, il est apparu dans le cadre de la conception orientée objet, couramment utilisé dans les projets logiciels, UML est utilisé pour spécifier, visualiser et construire les documents

nécessaires au bon développement d'un logiciel orienté objet. Il offre un standard de modélisation, pour représenter l'architecture logicielle.

bien évidemment, différents plateforme et logiciel s'offrent à nous, cependant après une étude approfondie, on a opté pour le logiciel :

*** Visual Paradigme for UM**

qui est un outil de conception et de modélisation, conçu pour faciliter le développement de logiciels

3 Analyse & Spécification détaillée du projet

3.1 Capture des besoins

Comme on l'a déjà évoqué, les entreprises sont confrontées aux différents problèmes de gestion de projet au sein de groupe, ces besoins peuvent se présenter sous différents aspects tel que :

- Collaboration et coordination entre acteurs qui participent sur le cycle de vie de réalisation logiciel (analyse, concepteur, développeur, testeur, intégrateur, validateur, police de qualité...),
- Enrichissement des modèles (version) et le retour en arrière (itération)
- Validation et approbation des itérations
- Planification et suivi des tâches
- Maîtrise des méthodes agile de développement collaboratif des logiciels

3.2 Identification des acteurs

Le logiciel sera dédié pour un public ouvert qui souhaiteraient la réalisation des projets logiciel selon la méthode agile Scrum.

et plus précisément aux :

- Entreprises ayant une méthodologie de travail agile et plus spécifiquement, celles basées sur le framework scrum.

ainsi les principaux acteurs utilisant l'application seront : les product owner, scrum masters et développeurs au sein de l'entreprise.

3.3 Les besoins fonctionnels

Il s'agit des fonctionnalités du système. Ce sont les besoins spécifiant un comportement d'entrée / sortie du Système.

- Authentification et accès
 - 1 - Pouvoir s'identifier pour accéder à la zone réservée aux membres.
 - 2 - Les utilisateurs du groupe XYZ ont accès aux fonctionnalités A, B et C, ainsi les développeurs auront accès à l'espace réservé qu'a leur partie et non d'autre parties du chef de projet ou autre..
 - 3 - Les administrateurs ont accès à toutes les fonctionnalités.
- Actions de saisie, d'ajustements, et d'annulation
 - 1 - Pouvoir planifier une réunion ainsi que la modification de cette dernière.
 - 2- Demande de participation à une réunion
 - 3 - Pouvoir Ajouter différents membres au groupe ainsi que le reject ou la suppression de participation d'un membre.
 - 4- Pouvoir ajouté/soumettre/modifier/supprimer des documents en toute sécurité et définir le mode / groupe de partage
 - 5 - Pouvoir se renseigner sur une réunion / backlog
- Processus
 - 1 - La réception de mail de notification automatique de planification d'une réunion ou demande de la part d'un membre
 - 2 - Validation d'une étape / document par la personne concernée.
- Collecte et utilisation des données
 - 1- Chaque membre aura un formulaire, ou une liste de tâches bien précise, évidente à fin de bien s'organiser et s'orienter sans difficulté
 - 2 - Accessibilité aux documents partagés, dédié pour un poste précis par chaque membre du métier adéquat.
- Interactions avec des outils et interfaces externes
 - 1 - Une bonne visualisation de l'interface graphique
 - 2 - Sauvegarde des données nécessaire en local / google drive afin de faciliter l'accessibilité directe.
 - 3 - Accessibilité aux différents outils de gestion.
- Respect des obligations légales
 - 1 - Le responsable d'un document partagé doit être informé de l'accès

- à ce document par une autre personne
- 2 - Le responsable d'une réunion planifiée doit être informé de l'accès à cette réunion par une autre personne
- 3 - L'utilisateur est informé sur la finalité de la collecte de données.
- 4 - tous les membres doivent se fier aux règlements légaux instaurés au sein de l'entreprise.
- 5 - L'utilisateur doit être tenu au courant en cas de brèche de sécurité.

3.4 Les besoins non fonctionnels

Outre que les besoins fonctionnels cités précédemment, pour bien répondre aux exigences des utilisateurs, le système devra pouvoir assurer les besoins non fonctionnels suivants :

- L'ergonomie et la convivialité : L'application fournira une interface conviviale et simple à utiliser et qui ne requiert aucun pré requis, donc elle pourra être exploitable par tout type d'utilisateurs (même les non informaticiens).
- La sécurité : L'accès aux informations n'est possible qu'après vérification des privilèges et des droits d'accès. Ainsi tout utilisateur passera par une phase d'authentification pour pouvoir consulter les services offerts par l'application logicielle.
- L'extensibilité : L'architecture de l'application permettra l'évolution et la maintenance (ajout ou suppression ou mise à jour) au niveau de ses différents modules d'une manière flexible.
- Contraintes humaines : Notre solution doit tenir compte des besoins de chaque acteur participant dans la gestion des changements. Elle doit leur faciliter la tâche et assurer leur activité dans les bonnes conditions. Donc, notre logiciel doit être facile à utiliser en présentant des interfaces ergonomiques.
- La fiabilité
 - 1 - Le système peut être restauré à une date ultérieure en cas de problème
 - 2 - Les données de l'historique des membres peuvent être récupérées en cas de panne serveur
- La facilité d'utilisation (utilisabilité)
 - 1 - Facilité de compréhension.
 - 2 - Facilité d'accès.
 - 3 - Facilité d'exploitation/d'utilisation.

4 - Attractivité de l'interface.

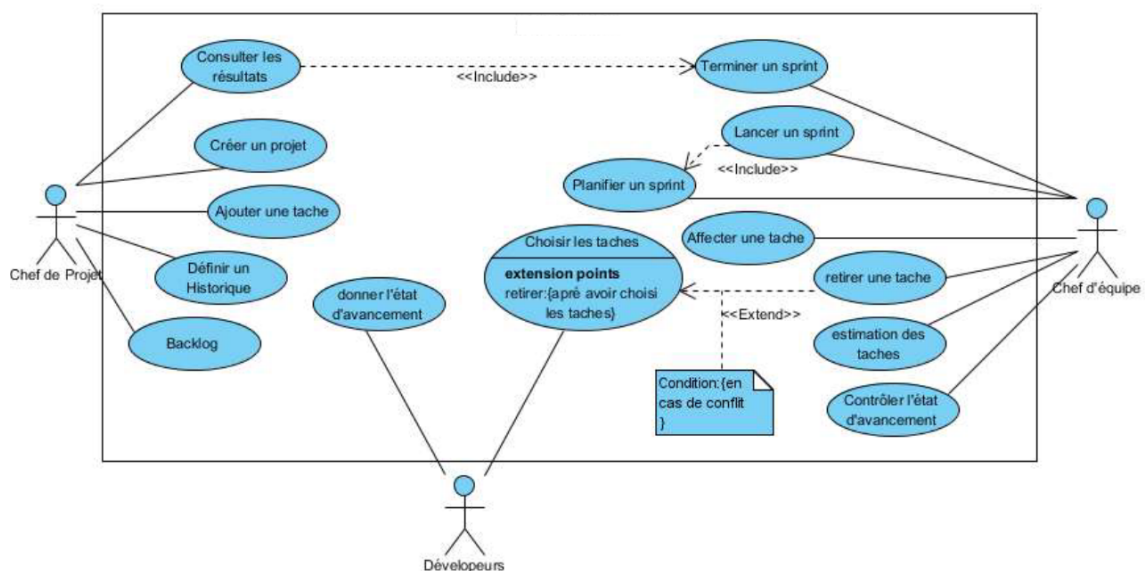
- Le rendement et l'efficacité
 - 1 - Comportement vis-à-vis du temps.
 - 2 - Comportement vis-à-vis des ressources.
- La maintenabilité
 - 1 - Facilité de modification.
 - 2 - Stabilité
 - 3 - Testabilité

3.5 Planning du traitement des cas d'utilisation

3.5.1 Étude des cas d'utilisation du sprint 1

-Diagramme des cas d'utilisation global

Un diagramme de cas d'utilisation global est présenté afin de définir l'architecture globale du logiciel en reliant tous les acteurs qui définissent un projet, product owner, scrum master et les développeurs le diagramme de séquence est comme suit :

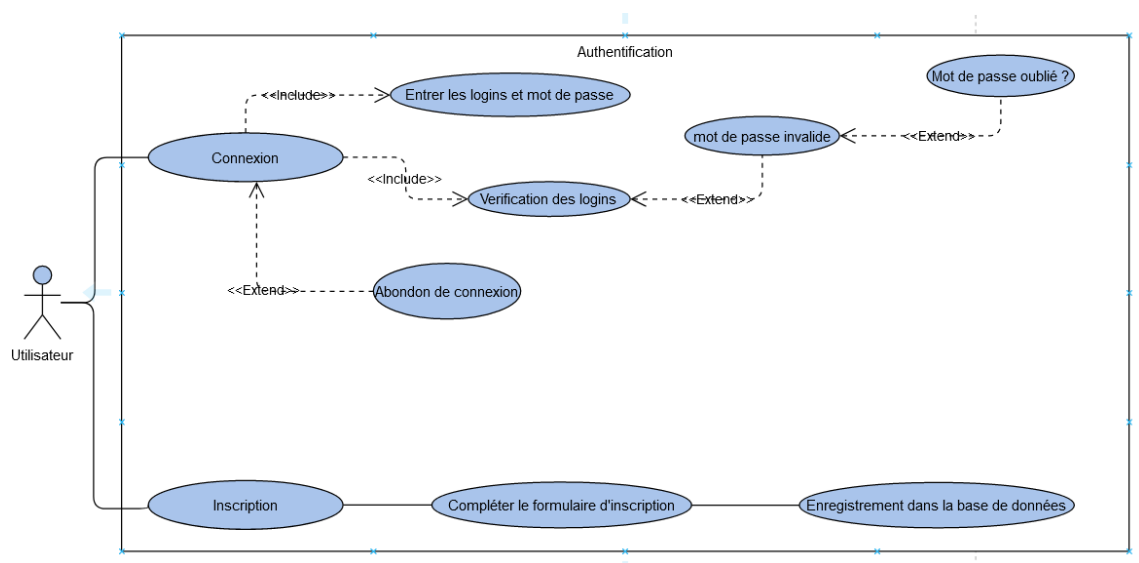


Cas 1 :

3.5.2 Description du cas

le premier module consiste à développer une interface d'authentification pour les nouveaux utilisateurs ainsi que de pouvoir s'identifier en tant que product owner, scrum master ou développeur

3.5.3 Diagramme de cas d'utilisation système

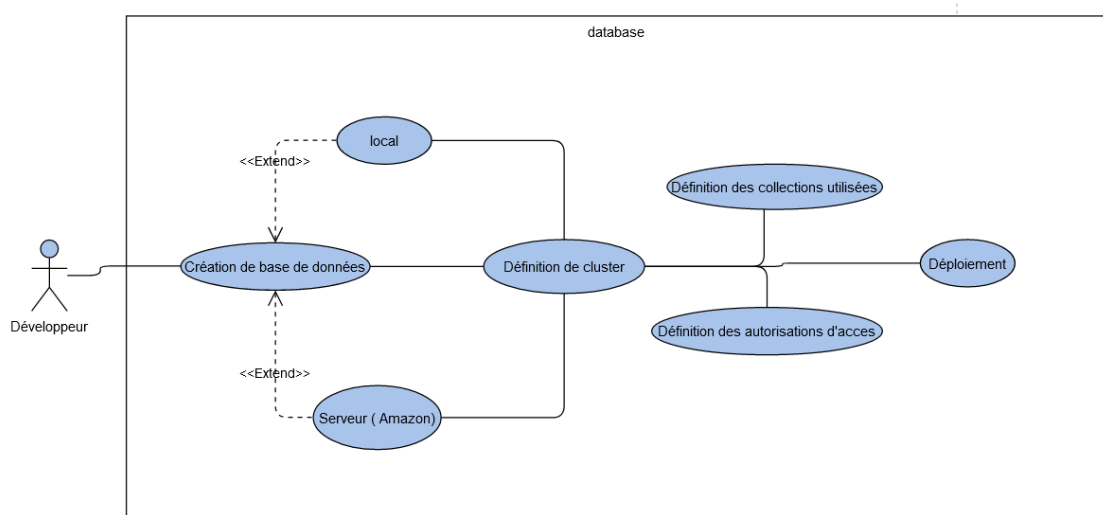


Cas 2 :

3.5.4 Description du cas

Etablir une base de données basé sur le cloud Amazon afin de pouvoir stocker les informations des clients / utilisateurs ainsi que les documents soumis, Ajouté à cela, une connexion entre l'interface du logiciel et la base de données

3.5.5 Diagramme de cas d'utilisation système



3.5.6 Étude des cas d'utilisation du sprint 2

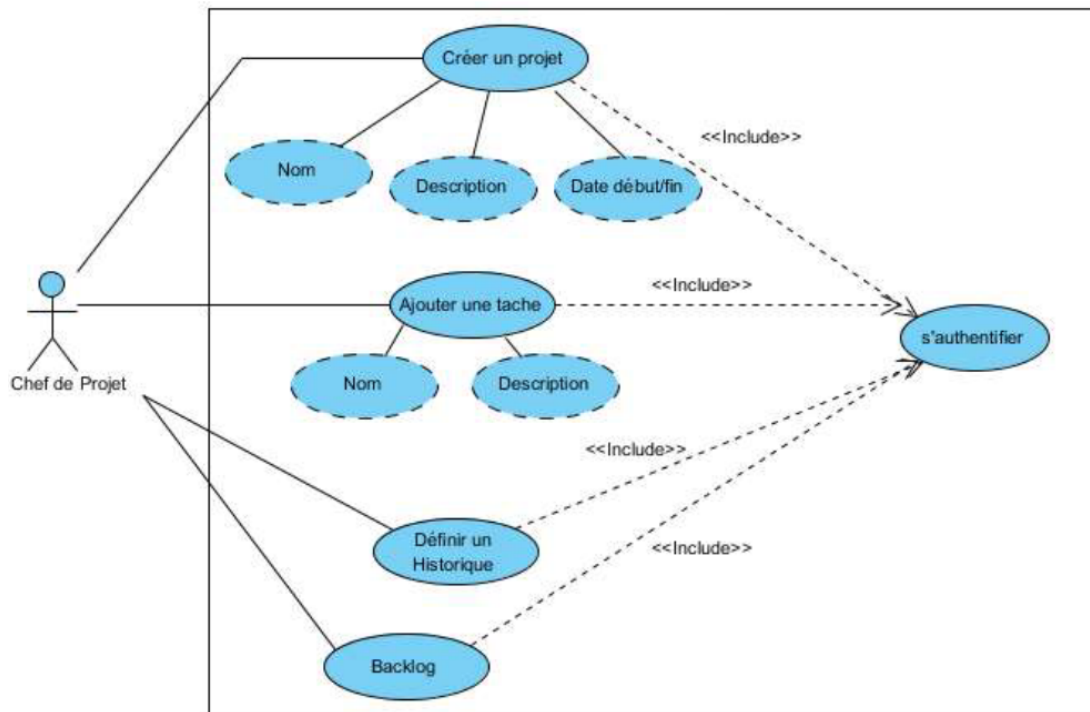
Cas 1 :

3.5.7 Description du cas

le deuxième module consiste à développer une interface pour le chef de projet

3.5.8 Diagramme de cas d'utilisation système

Ci-dessous le diagramme de cas d'utilisation pour le chef de projet



3.5.9 Étude des cas d'utilisation du sprint 3

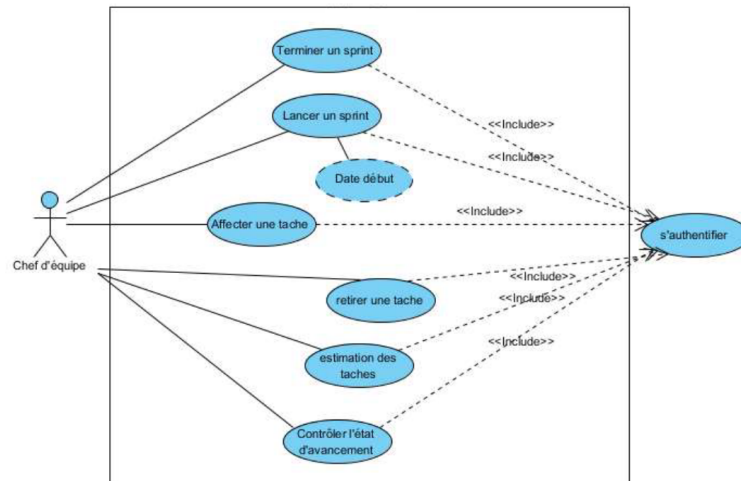
Cas 1 :

3.5.10 Description du cas

Une fonctionnalité qui permettra au chef de projet d'assigner une tâche, la retirer, ainsi que lancer et terminer un sprint

3.5.11 Diagramme de cas d'utilisation système

Ci-dessous le diagramme de cas d'utilisation :

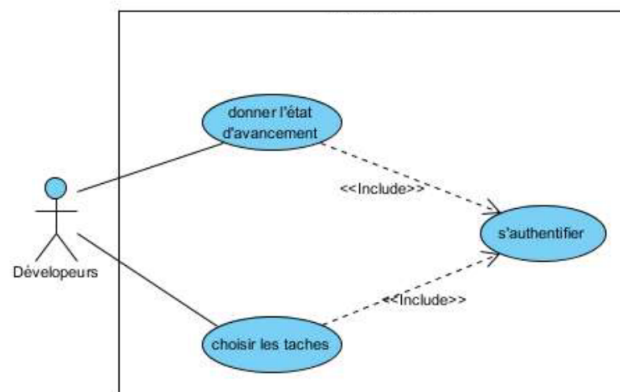


Cas 2 :

3.5.12 Description du cas

Ce module consiste à l'interface développeur

3.5.13 Diagramme de cas d'utilisation système



3.5.14 Priorité

Nous commençons par définir les tâches de chaque utilisateur du logiciel. Ces tâches, exprimées sous forme de besoins, sont priorisées par le Product Owner ce qui permet d'établir un ordre à respecter lors de la réalisation de l'application.

Taches	Priorité
Backlog	1
Créer un projet	2
Ajouter une tâche	3
Lancer un sprint	4
Affecter une tâche	5
Choisir une tâche	6
Donner l'état d'avancement	7
Consulter l'état d'avancement	8
Estimation des tâches	9
Définir un historique	10
Consulter des résultats	11

3.5.15 Technique d'estimation agile : Big / Uncertain / Small

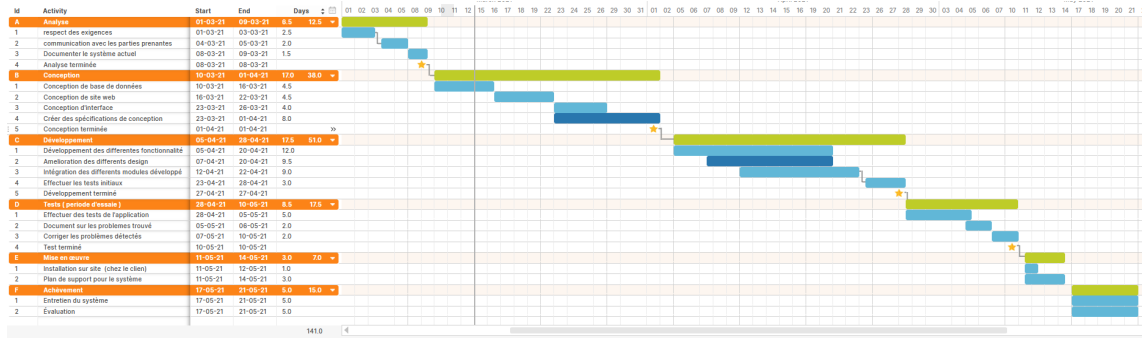
Big / Uncertain / Small est une méthode d'estimation agile où les éléments à estimer sont placés par les participants qui classent simplement les fonctionnalités en trois catégories : big (importante), uncertain (incertaine) et small (petite).

Big	Uncertain	Small
Backlog	Définir un historique	Consulter une tâche
Créer un projet	Estimation d'une tâche	Ajouter une tâche
Lancer un sprint		Retirer une tâche
Terminer un sprint		Donner l'état d'avancement d'une tâche
Affecter une tâche		
Choisir une tâche		
Consulter l'état d'avancement		

3.6 Diagramme de Gantt

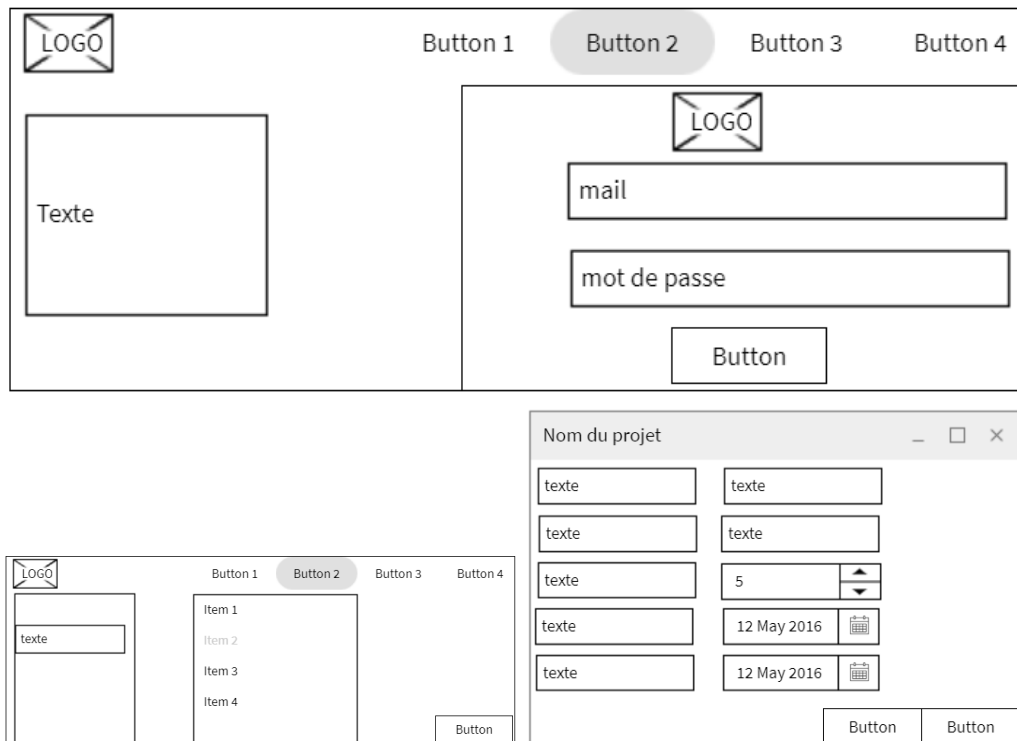
Concernant la réalisation du projet, une estimation des différentes étapes à été faite depuis la conceptions des différents documents jusqu'au dévelop-

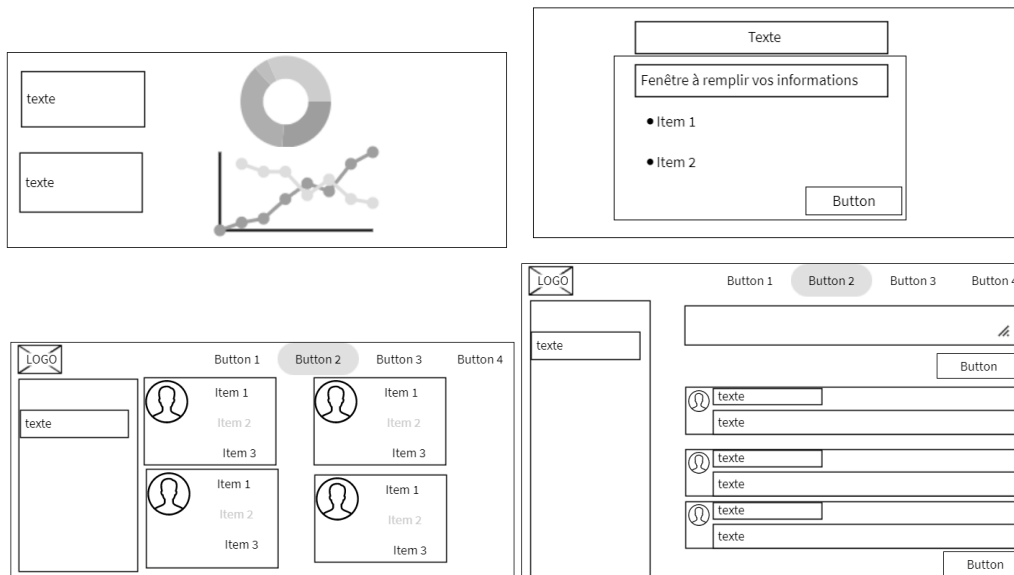
pement du logiciel.
ci-dessous, le diagramme de gantt :



3.7 Prototypage des interfaces

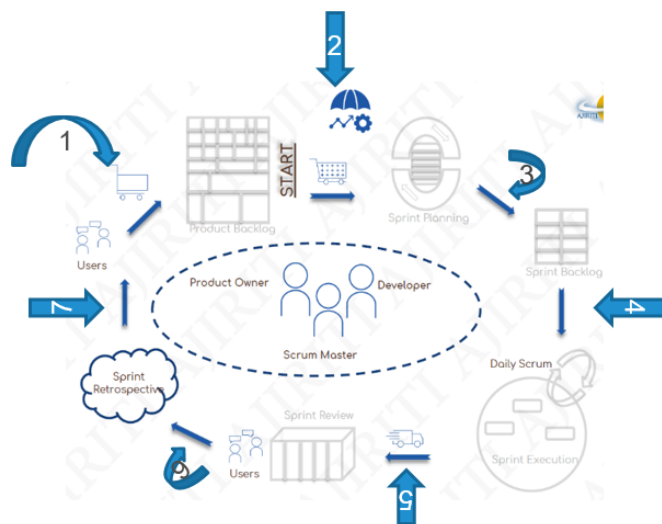
Les différentes maquettes de la plateforme collaborative sont comme suit :
Les différentes maquettes de l'application web « simulation du coronavirus » sont comme suit :





3.8 Pilotage du projet avec SCRUM

Pour piloter un projet, différents étapes sont nécessaires comme le montre le schéma ci-dessous :



Ainsi, pour un premier temps, les utilisateurs (client) définissent leur besoin puis :

- 1 - Stocker les besoins de l'utilisateur dans le product backlog qui vont être rangé par ordre de priorité.
- 2 - Les besoins vont être émis lors de la cérémonie du sprint planning où les développeurs et le product owner vont se mettre d'accord sur ce que l'équipe peut développer dans un temps défini à l'avance.
- 3 - L'engagement de l'équipe est matérialisé dans le sprint backlog, l'exécution du sprint peut alors commencer.
- 4 - Chaque jour un daily scrum de 15 min à lieu au début de journée où chaque développeur va répondre aux 3 questions (qu'est-ce que j'ai fait hier, qu'est-ce que je vais faire, qu'est ce qui me bloque)
- 5 - Une fois les fonctionnalités terminées un sprint review est organisé avec les utilisateurs à fin de vérifier si ce qui est développé correspond à la demande du client.
- 6 - Lors de la dernière cérémonie sprint rétrospective notre équipe analyse ce qui fonctionne pendant le sprint et ce qui n'a pas fonctionné et ce qui est améliorable.
- 7 - Puis l'équipe décide ensemble des mesures à mettre en place puis un nouveau cycle commence.

3.9 Les outils SCRUM

Pour le développement de notre plateforme en méthode agile scrum, on s'est lancé sur différentes plateformes afin d'avoir une vision globale du domaine et pouvoir instaurer une technologie intéressante et une visibilité adéquate :

et parmi les outils utilisés sont :

- Un planificateur de projet
- Un tableau de bord pour les réunions en visioconférence
- Une visualisation facilitée de l'avancement des projets
- Des fonctions pour gérer le temps des équipes
- Consulter les différentes versions
- Définir des sprints et les retours sur les réunions
- Créer des bogues pour matérialiser les erreurs à corriger.
- Différentes intégrations possibles avec Zoom, Slack, Gmail, Zapier...

3.10 Équipe et rôles

Au sein du cadre de travail le framework Scrum nous trouvons différents processus technique, rôles différents et des événements.

Les 3 rôles de l'équipe de Scrum sont :

1 - Product owner, ces principaux rôles sont :

- Représenter le client à fin d'avoir une bonne vision du produit.
- Maximise la valeur du produit.
- Participer au sprint planning à fin d'apporter les précisions sur les fonctionnalités pour les développeurs.
- Vérifier avec l'utilisateur les fonctionnalités livrées en sprint review.
- Recevoir un feedback du l'utilisateur sur ce qui est livré ainsi de lui remonter ce qui constitue le prochain incrément.
- Ajuste les fonctionnalités à chaque sprint si nécessaire
- Accepte ou rejette les résultats
- Prioriser et affiner les sprints dans le product backlog

2 - Scrum master qui est le garant du framework scrum ces principaux rôles sont :

- Anime le daily scrum et le retrospective.
- Elimine tous les obstacles qui bloque l'équipe.
- Motive l'équipe pour qu'elle soit auto-organiser.
- Encourage l'équipe pour quelle soit multi-compétente.
- Protège l'équipe des interférences extérieures.

3 - Equipe Scrum (Development Team) qui est composé de 5 a 10 personnes (Architecte, concepteur, développeur, spécialiste IHM, testeur, etc.) :

- Effectue l'exécution du sprint (conception, codage, test).
- Participe au daily scrum.
- Affine le carnet du produit.
- La composition de l'équipe ne doit pas changer pendant un sprint

3.11 Déroulement de scrum

* Planification du sprint :

- L'équipe choisit, à partir de backlog de produit les éléments qu'elle s'engage à finir.
- La liste des tâches est créée :
 - Les tâches sont identifiées et estimées (en jours-homme)
 - Collectivement, pas seulement le chef d'équipe

* Le scrum quotidien :

- Paramètres
 - Tous les jours
 - 15 minutes
- Participants :
 - Tous le monde est invité
 - Seuls les membres d'équipe peuvent parler
- Chacun répond aux différentes questions mais principalement :
 - Qu'est-ce que t'as fait hier ?
 - Que vas-tu faire aujourd'hui ?
 - Y'a-t'il des obstacles qui te freinent ?

* Revue de sprint (La démo) :

À la fin de chaque sprint on fait une démo, elle pousse l'équipe à terminer leur travail et le livrer

- L'équipe présente ce qu'elle a fait pendant le sprint
- Toute l'équipe participe
- On invite les autres à découvrir ce que l'équipe fait

* Rétrospective du sprint :

C'est une réunion qui vise à améliorer les sprints et ne plus faire les mêmes erreurs

- Se fait à la fin de chaque sprint
- Tout l'équipe participe même le chef de projet éventuellement client et autres intervenants
- Chacun donne son avis
 - ce qu'il pense être bien
 - ce qu'il pense qui peut être mieux

- ou ce qu'il devrait être différent pour le prochain sprint

Et à la fin on garde les suggestions qui peuvent améliorer le prochain sprint.

3.12 Le backlog du produit

Nous commençons par définir le backlog qui consiste en une liste de fonctionnalités à réaliser. Ces fonctionnalités, exprimées sous forme de besoins, sont priorisées par le Product Owner ce qui permet d'établir un ordre à respecter lors de la réalisation de l'application :

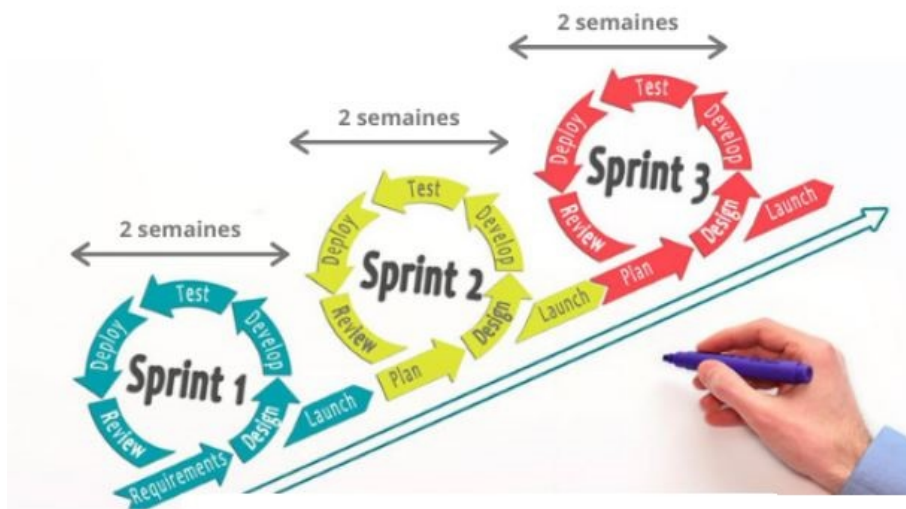
Backlog produit	User stories	Priorité	Estimation (jour)
Création de la BDD	En tant qu'administrateur j'aurai la possibilité de stocker les informations de chaque membre de mon groupe, de pouvoir sauvegarder les différents fichiers des différentes versions	1	4
Création d'interface web	En tant qu'utilisateur j'aurai la possibilité de créer un compte et s'authentifier à mon espace personnel	2	3
Création d'un espace projet	En tant qu'utilisateur j'aurai la possibilité d'ajouter un projet, la mise à jour, la modification, la suppression, définir un deadline, visualiser les membres participants au projet ainsi que la visualisation de l'ensemble des projets	3	7
Un outil de visualisation de l'avancement du projet	En tant qu'utilisateur j'aurai la possibilité de visualiser l'avancement de mon projet, l'avancement de chaque membre de la tâche qui lui a été assigné, ainsi que le temps restant avant la date de livraison	4	7

Espace membre	En tant qu'utilisateur j'aurai la possibilité d'accéder a mon espace membre où je pourrai visualiser mes projets en cours, l'avancement de chaque projet ainsi que l'avancement de mes propres taches	6	7
Espace de dépôt et de partage	En tant qu'utilisateur j'aurai la possibilité d'accéder à un espace de déport de documents partagé au cours de la réalisation du projet	5	7
Espace d'échange	En tant qu'utilisateur j'aurai la possibilité d'échanger avec les membres du projets ainsi que le clients sur un forum dédié ou de lancer une vidéo-conférences en utilisant différents outils bien définis	7	7

3.13 Planning des sprints

Pour la réalisation de l'application "Plateforme de développement collaborative", nous avons mis en place différents sprints qui s'étendent sur l'ensemble du temps alloué depuis la conception jusqu'à la livraison. pour bien travailler, il est essentiel de bien définir les sprints :

- De 2 à 4h de sprint planning pour un sprint de 2 semaines
- De 4 à 8h de sprint planning pour un sprint de 4 semaines



Ayant fixé la date du début du projet le : 01-06-2021 :
ci-dessous l'ensemble des sprints mis en place :

Sprint	Date	User stories	Durée
1	14-06-2021	Définition des premiers objectifs	10 jours
2	28-06-2021	Intégration des premiers modules et preparation pour le test	14 jours
3	20-07-2021	Développement & intégration des nouveaux modules	14 jours
4	15-08-2021	Hébergement & préparation pour le déploiement de l'application	7 jours

4 Étude et réalisation du sprint -1-

4.1 Spécification fonctionnelle & non fonctionnelle du sprint

* Revoir les notes de révision rétrospective
durant ce sprint on va discuter sur :

- Ce qui a été fait ?
- Les problématiques rencontrées ?
- Les changements qu'on doit mettre en place ?

* Revoir les modules développés :

- Interface d'authentifications
- Intégration de la base de données
- Développement de l'architecture globale des modules (une première version)

* Affectation de répartition des tâches :

- Revoir les documents attribués aux tâches, aux propriétaires et à la date limite pour plus de visibilité
- Intégration de la base de données
- Développement de l'architecture globale des modules (une première version)

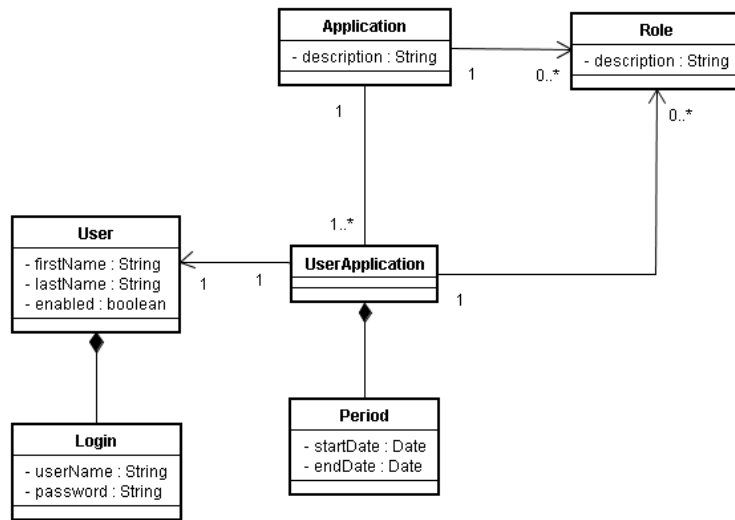
* Définition des objectifs à venir :

- Ce qu'on doit modifier, commiter, valider et retirer
- Amélioration des modules développés
- Commencer la phase de tests des modules finis
- Développement de l'espace des membres (équipe)

4.2 Étude des cas d'utilisation du sprint 1

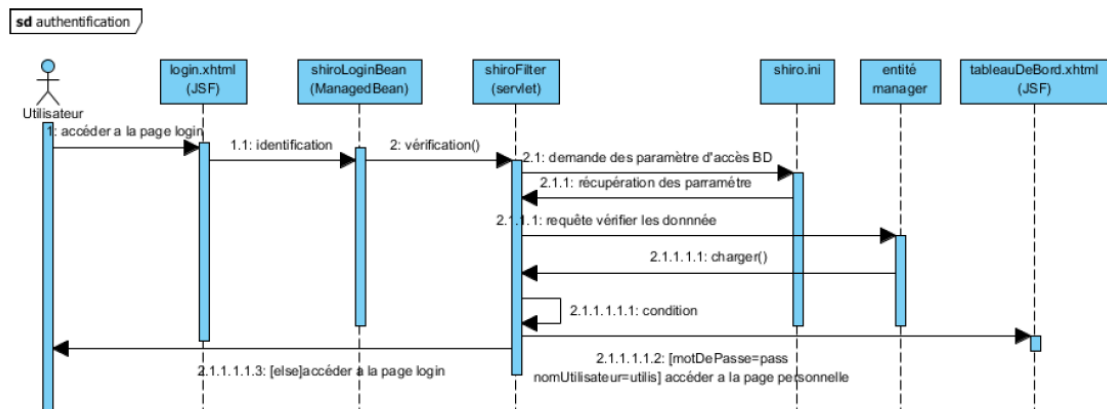
4.3 Conception du Sprint 1

4.3.1 Diagramme de classe



4.3.2 Diagramme de séquence

ainsi le diagramme de cas d'utilisation qui illustre l'authentification est comme suit :



4.4 Planification des Tests et validation

Afin de tester la bonne conception des modules développés, différents tests seront appliqués afin de trouver les composants qui seront modifier ou

validé.

pour la réalisation des tests, un tableau de planification des tests est mis en place qui va définir un exemple de test ainsi que la sortie attendu.

ci-dessous le tableau :

Fonctionnalité	test	Résultat attendu	Commentaire
Login	l'utilisateur entre des identifiants qui n'existe pas	Erreur d'authentification, utilisateur n'existe pas	lors de la connexion, si l'utilisateur ne dispose pas d'un compte déjà créer, il aura un message d'erreur qui va lui demander de créer un compte ou de saisir les bons identifiants
Login	l'utilisateur saisie des identifiants correctes	Connexion établie	Redirection à son espace personnelle
Login	l'utilisateur saisie le nom d'utilisateur correcte & le mot de passe incorrecte	Connexion non-établie	Le système affiche un message "mot de passe incorrecte" et il affiche une option pour récupérer le mot de passe
Register	l'utilisateur saisie un identifiant ou un email déjà existant	Impossible d'enregistrer	Le système affiche un message "identifiant ou mail existe déjà !, veuillez vous connecter"
Register	l'utilisateur saisie un identifiant ou un email déjà existant	le compte a été crée avec succès	Redirection à son espace personnelle

5 Étude et réalisation du sprint -2-

5.1 Spécification fonctionnelle & non fonctionnelle du sprint

* Revoir les notes de révision rétrospective
durant ce sprint on va discuter sur :

- Ce qui a été fait ?
- Les problématiques rencontrées ?

- Les changement qu'on doit mettre en place ?

* Revoir les modules développés :

- Interface de chef de projet
- Création d'une tâche
- Définir l'historique des tâches
- Visualisation du backlog

* Affectation de répartition des tâches :

- Revoir les documents attribués aux tâches, aux propriétaires et à la date limite pour plus de visibilité
- Intégration de la base de données
- Développement de l'architecture globale des modules (une premiere version)

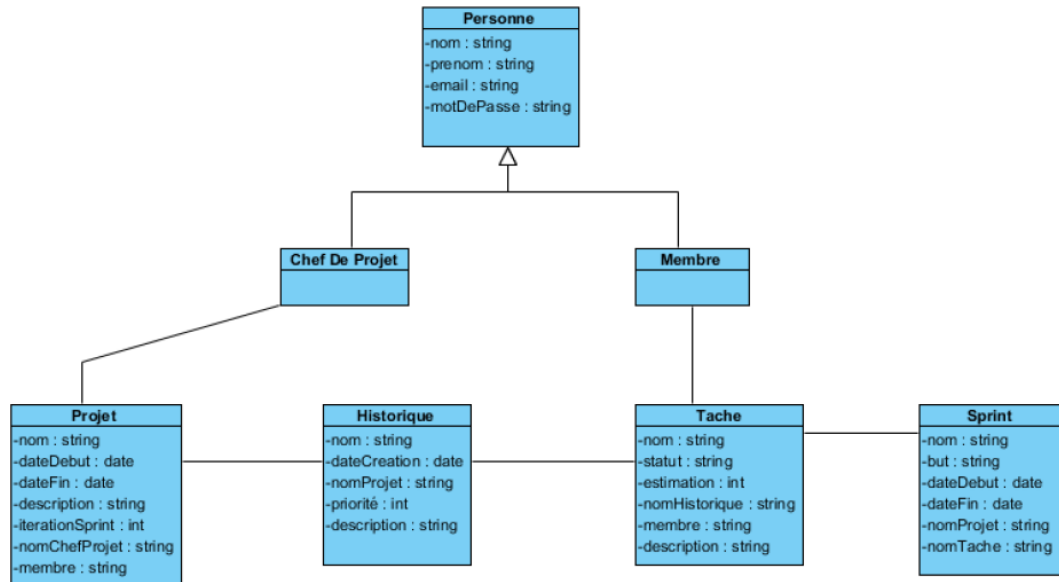
* Definition des objectifs à venir :

- Ce qu'on doit modifier, commiter, valider et retirer
- Amélioration des modules développés
- Commencer la phase de tests des modules finis
- Développement de l'espace des membres (équipe)

5.2 Conception du Sprint 2

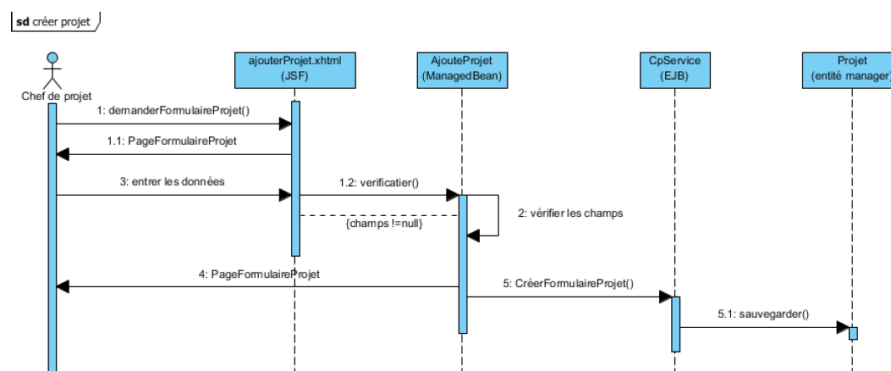
5.2.1 Diagramme de classe

Ci-dessous le diagramme de classe concernant la création d'un projet



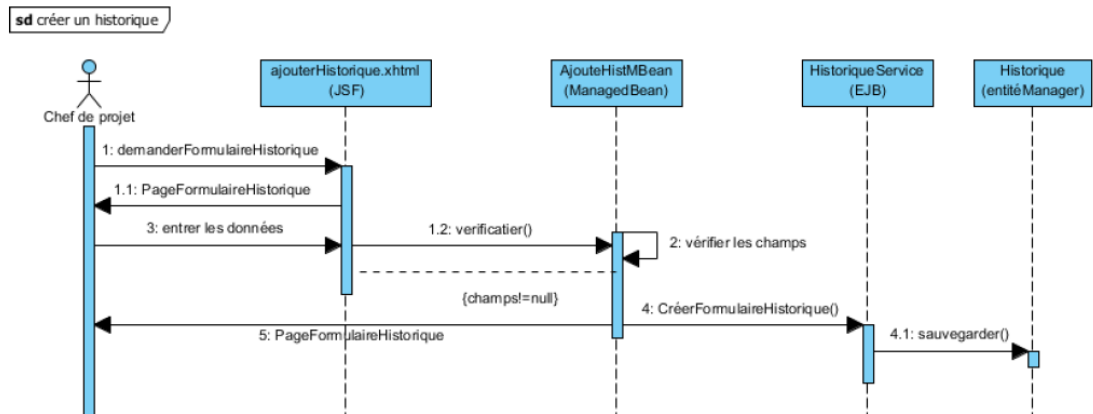
5.2.2 Diagramme de séquence

Ci-dessous le diagramme de séquence concernant la création d'un projet.



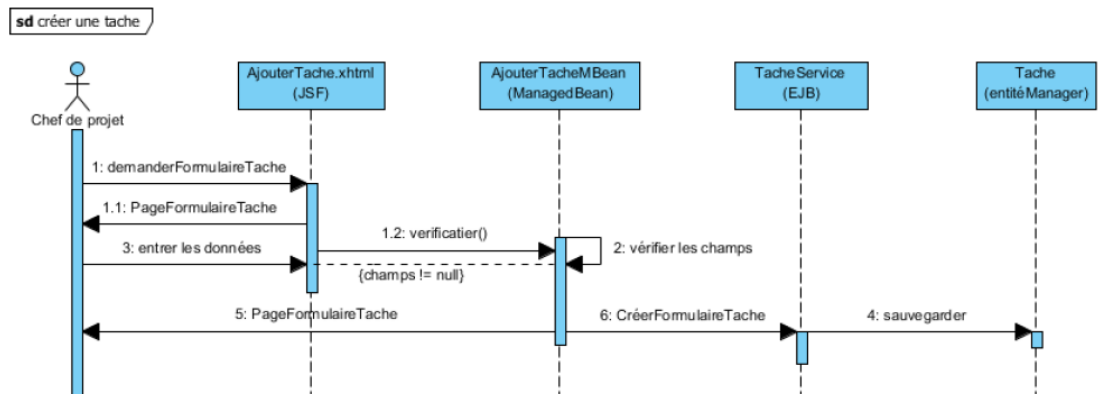
5.2.3 Diagramme de séquence

Ci-dessous le diagramme de séquence concernant l'historique des tâches



5.2.4 Diagramme de séquence

Ci-dessous le diagramme de séquence concernant la création d'une tâche



5.3 Planification des Tests et validation

Ci-dessous le tableau :

Fonctionnalité	test	Résultat attendu	Commentaire
Création d'un projet	l'utilisateur clique sur le bouton d'ajout d'un projet et saisie les informations nécessaires	Création du projet	Le système crée le nouveau projet et l'ajoute à la liste des projets
Création d'un projet	l'utilisateur crée un projet avec un nom déjà existant	Erreur projet existe déjà !	Le système affiche un message d'erreur comme quoi un projet pourtant le même nom existe déjà !
Création d'une tâche	l'utilisateur crée une tâche	Création de la tâche avec succès	Le système affiche créé la tâche dans le projet ciblé
Définir un historique	l'utilisateur définit un historique d'une réunion	Création de l'historique	Le système sauvegarde l'historique
Visualisation du backlog	l'utilisateur clique sur le bouton du backlog	Le backlog s'affiche	Le système affiche le backlog définit par le product owner

6 Étude et réalisation du sprint -3-

6.1 Spécification fonctionnelle & non fonctionnelle du sprint

* Revoir les notes de révision rétrospective
durant ce sprint on va discuter sur :

- Ce qui a été fait ?
- Les problématiques rencontrées ?
- Les changement qu'on doit mettre en place ?

* Revoir les modules développés :

- Amélioration de l'interface du chef de projet.
- Interface développeur.
- Choisir une tâche à réaliser
- Donner l'état d'avancement d'une tâche

* Affectation de répartition des tâches :

- Revoir les documents attribués aux tâches, aux propriétaires et à la date limite pour plus de visibilité
- Intégration de la base de données
- Développement de l'architecture globale des modules (une premiere version)

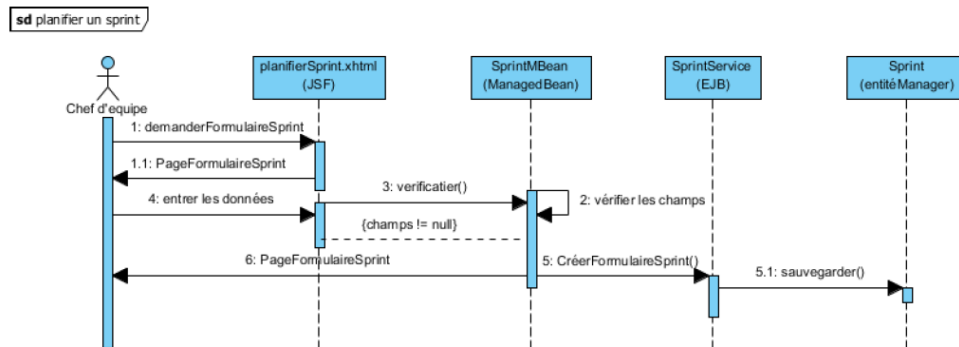
* Definition des objectifs à venir :

- Ce qu'on doit modifier, commiter, valider et retirer
- Amélioration des modules développés
- Commencer la phase de tests des modules finis
- Développement de l'espace des membres (équipe)

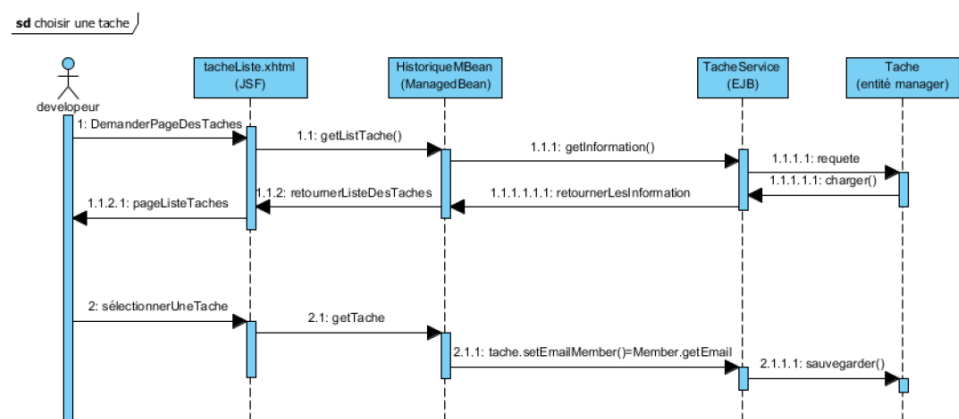
6.2 Conception du Sprint 3

6.2.1 Diagramme de séquence

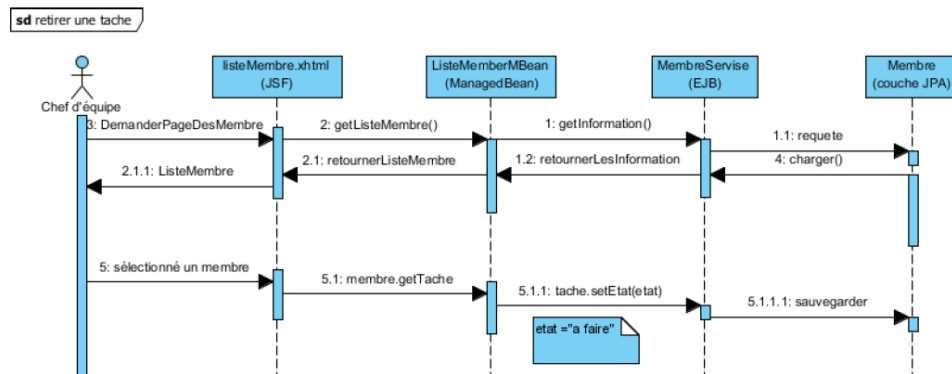
Ci-dessous le diagramme de séquence concernant la planification d'un sprint



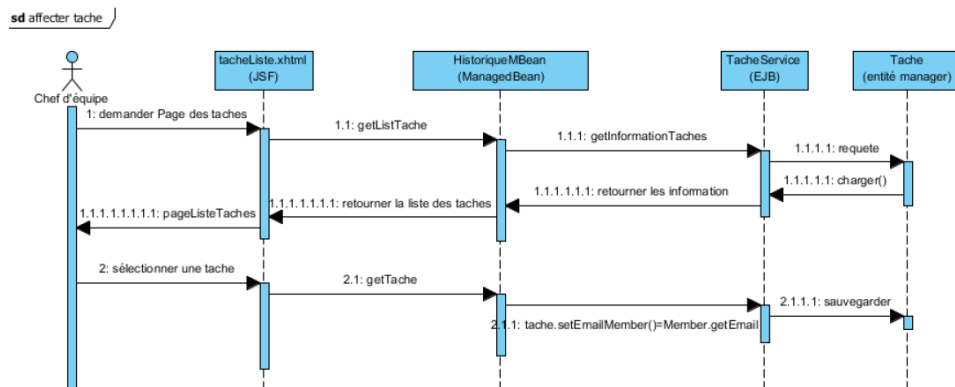
Ci-dessous le diagramme de séquence concernant le choix d'une tâche pour le développeur.



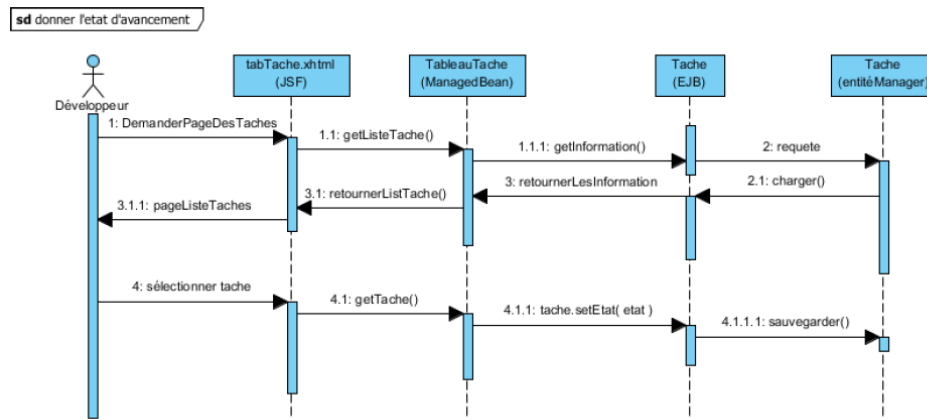
Ci-dessous le diagramme de séquence concernant le retrait d'une tâche par le chef de projet.



Ci-dessous le diagramme de séquence concernant l'affectation d'une tâche



Ci-dessous le diagramme de séquence pour donner l'état d'avancement d'une tâche.



6.3 Planification des Tests et validation

Ci-dessous le tableau :

Fonctionnalité	test	Résultat attendu	Commentaire
Planifier un sprint	Le chef de projet clique sur le bouton pour planifier un projet	une fenetre pour planifier un projet s'ouvre	le système va afficher un pop up avec un calendrier qui permettra la planification d'un sprint, et l'affiche sur la liste des sprint à venir
Planifier un sprint	Le chef de projet planifie un sprint avec une date antérieure au jours actuel	Un message d'erreur s'affiche	le système va afficher un message d'erreur disant que la date n'est pas adéquat, veuillez selection une date supérieure au jours actuel
Retrait d'une tache	Le chef de projet sélectionne un membre et retire une tache	La tache retiré ne sera plus disponible pour le membre	Le système doit pouvoir retiré une tache d'un membre X et affiche comme quoi la tache n'est plus disponible pour cette derniere.
Affectation d'une tache	Le chef de projet sélectionne un membre et affecte une tache	La tache affecté sera plus disponible pour le membre	Le système doit pouvoir affecté une tache pour un membre X et affiche comme quoi la tache est actuellement attribué pour cette derniere.
Affectation d'une tache	Le chef de projet sélectionne affecte une tache à un membre qui n'existe pas	Un message d'erreur sort	Le système doit pouvoir verifier si un membre au quel on affecte une tache existe ou pas, et si il n'existe pas un message d'erreur sera afficher disant que le membre n'existe pas
Choisir une tache	le développeur choisit une tache	Tache sélectionner	Le système supprime la tache sélectionner puis il la déplace dans l'espace du développeur
Mise à jour de la tache	le développeur met en place l'avancement de la tache	Tache ajouter après modification	Le système affiche l'historique des nouvelles taches pour que chaque membre de l'équipe peuvent les visualisées

7 Étude et réalisation du sprint -4-

7.1 Spécification fonctionnelle & non fonctionnelle du sprint

* Revoir les notes de révision rétrospective
durant ce sprint on va discuter sur :

- Ce qui a été fait ?
- Les problématiques rencontrées ?
- Les changements qu'on doit mettre en place ?

* Revoir les modules développés :

- Intégration des différents modules développés
- Déploiement du logiciel

* Affectation de répartition des tâches :

- Revoir les documents attribués aux tâches, aux propriétaires et à la date limite pour plus de visibilité
- Intégration de la base de données
- Développement de l'architecture globale des modules (une première version)

* Définition des objectifs à venir :

- Ce qu'on doit modifier, commiter, valider et retirer
- Amélioration des modules développés
- Commencer la phase de tests des modules finis
- Développement de l'espace des membres (équipe)

7.2 Étude du sprint 4

Au cours de ces derniers sprints, on s'assure que tous les modules marchent bien, on intègre les derniers modules mis à jour et on prépare le déploiement de l'application

7.3 Planification des Tests et validation

Ci-dessous le tableau :

Fonctionnalité	test	Résultat attendu	Commentaire
Intégration	Le chef de projet vérifie que tous les modules sont intégrés	vus sur tout les modules	Le système dispose de l'ensemble des fonctionnalités développées, intégrées et qui marchent sans problème
Déploiement	Les développeur déploie l'application en ligne	Disponibilité d'accès à l'application	l'application sera développer sur le réseau et sera accessible avec un lien précis
Test de fonctionnement	Le chef de projet test l'application sur différents navigateurs	L'application marche sans problème	le chef de projet va tester l'application sur différents navigateurs afin de s'assurer le bon fonctionnement du logiciel
Test final	Le client test l'application	L'application marche sans problème	le client va tester la version finale de l'application sur différents navigateurs afin de s'assurer le bon fonctionnement du logiciel

8 Burndown chart

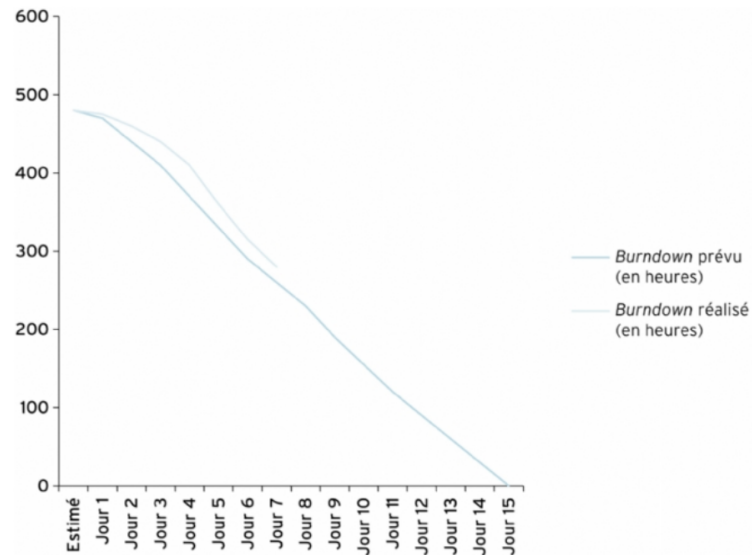
Le burndown chart de sprint est un graphique actualisé tous les jours dans le cadre de la mêlée, montrant la tendance de l'avancement dans la réalisation des tâches du backlog de sprint. Il montre aux équipiers l'utilité de demander tous les jours l'état de l'avancement des travaux, car ces informations données sont immédiatement affichées, et accessibles à toutes les parties prenantes du projet, dont le propriétaire du produit.

Ces objectifs :

- Disposer d'une vision rapide de l'avancement d'un sprint.
- Matérialiser la performance collective de l'équipe, et remettre du poids au collectif.

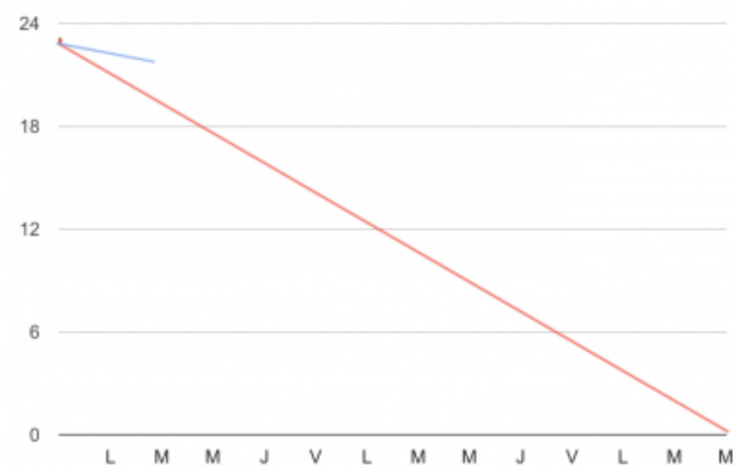
8.1 Burndown chart sprint 1

Ci-dessous le burndown estimer pour le sprint 1



8.2 Burndown chart sprint 2

Ci-dessous le burndown estimer pour le sprint 1



9 Rétrospective

Organisée à la fin de chaque sprint suite à la démo, la rétrospective est une cérémonie Scrum qui permet aux membres d'une équipe d'échanger sur le vécu du dernier sprint afin de pouvoir s'améliorer sur les suivants.

Son objectif est d'identifier les dysfonctionnements dans l'organisation de l'équipe afin de pouvoir définir des actions concrètes à mettre en place dans les sprints suivants et les résoudre. Le but n'est donc pas seulement de revenir en arrière, mais également de se projeter pour préparer au mieux le futur de l'équipe. C'est un moment particulier, où l'équipe ne va plus produire mais réfléchir et échanger sur ses expériences ainsi qu'un plan d'action à mettre en place. Cet exercice est donc un rendez-vous majeur, qui permet l'amélioration continue du processus de travail de l'équipe.

10 Conclusion

Pour conclure, Ce module nous a permis d'aborder non seulement l'aspect théorique de la methode agile scrum, mais aussi l'aspect pratique auquel on a mis l'ensemble des connaissances ainsi que competences acquise pour la réalisation de ce projet, on a pu réaliser une approche l'environnement de l'entreprise ainsi que le monde proffessionel et plus précisément le monde de l'informatique d'une manière concrete,

Car avoir des problèmes de gestions de projets, Avoir des problemes de communication au sein d'un groupe.. oui c'est courant, mais trouver une solution pour les resoudres est encore plus adapté est notre devoir,

de ce fait, nous nous somme imérger dans le monde du scrum et du développement en mode agile, et avons mis en pratique l'ensemble des comptences et pratiques nécessaires afin de pouvoir développer un logiciel permetant aux entreprises de réaliser un projet en agilité, de résoudre les problèmes de communication au seins du groupe et optimiser chaque étape du processus de la réalisation d'une applicatio.

et pour finir, ce projet nous permis de nous surpasser du coté conaissances ainsi que du coté techniques et de faire un pas de plus vers une gestion parfaite de nos projets.

11 Perspective

Bien évidemment, cette application est une version BETA pour le développement d'un logiciel parfait, et parmi les fonctionnalités qu'on pourrait implémenter, l'intelligence artificielle, qui nous permettra de gérer nos projets, de proposer un schéma de test adéquat, d'aider à la planification de tout le cycle de développement et on pense que ça sera un atout parfait pour les entreprises de pouvoir bénéficier d'une technologie pareille.

Références

- [1] [juin 2012] Smile Gestion du cycle de vie des applications (ALM) le meilleur des solutions de l'open source
- [2] [2007] Henrik Kniberg : Scrum et XP depuis les Tranchées –comment nous appliquons Scrum-
- [3] [CollabNet 2013] Scrum Methodology [http ://scrummethodology.com/](http://scrummethodology.com/)
- [4] [Scrum.org 2013] Scrum [http ://www.scrum.org/Scrum-Guides](http://www.scrum.org/Scrum-Guides)
- [5] [gestion-de-projet.org] [https ://blog-gestion-de-projet.com/comment-organiser-un-sprint-planning/](https://blog-gestion-de-projet.com/comment-organiser-un-sprint-planning/)