Contents

[I. Backlog du sprint 1 « Java » : 3](#_Toc506802213)

[II. Diagramme de Cas d’utilisation : 9](#_Toc506802214)

[III. Conception de vue dynamique 10](#_Toc506802215)

[3.1. Diagramme de séquence objet : 10](#_Toc506802216)

[3.2. Diagramme de Classe  : 11](#_Toc506802217)

[IV. Burndown chart 12](#_Toc506802218)

# Backlog du sprint 1 « Java » :

Avant chaque sprint, nous organisons une réunion de planification de sprint. Ce planning sélectionne depuis le backlog du produit les exigences les plus prioritaires pour le client. Elles sont développées, testées et livrées au client à la fin du sprint. Elles constituent le backlog du sprint représenté ci-dessous qui n’est autre qu’un sous ensemble du backlog du produit.

* Date début du sprint : Jeudi 29 Janvier 2018
* Date fin du sprint : Jeudi 26 février 2018
* Temps estimé en heures : 180h
* Echelle de mesure : Une journée égale 6h de travail (de 18h à 00h) .

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID user story | user story | ID tache | tache | estimation en heure | |
| 1.1 | en tant que administrateur je veux m'authentifier | 1.1.1 | Création de l’interface | 2h | Mahdi |
| 1.1.2 | Développer la méthode authentification ainsi déconnexion. | 2h | Mahdi |
| 1.1.3 | Intégration de l'API mail et cryptage | 2h | Mahdi |
| 1.2 | en tant que administrateur je veux consulter la liste des utilisateurs | 1.2.1 | Création de l’interface qui contient la liste des utilisateurs avec le table view | 2h | Mahdi |
| 1.2.2 | développer la méthode d'affichage | 2h | Mahdi |
| 1.3 | en tant que administrateur je peux supprimer les utilisateurs | 1.3.1 | Création de l’interface de suppression | 2h | Mahdi |
| 1.3.2 | développer la méthode de suppression | 2h | Mahdi |
| 1.4 | en tant que client je veux m'authentifier | 1.4.1 | Création de l’interface | 2h | Mahdi |
| 1.4.2 | Développer la méthode authentification ainsi déconnexion. | 4h | Mahdi |
| 1.5 | en tant qu'internaute je peux s'inscrire | 1.5.1 | création de l'interface d'inscription | 3h | Mahdi |
| 1.5.2 | développer la méthode d'ajout | 3h | Mahdi |
| 1.6 | en tant que client je veux modifier mon profile | 1.6.1 | Création de l’interface de modification | 2h | Mahdi |
| 1.6.2 | Développer la méthode de modification. | 1h | Mahdi |
| 2.1 | en tant que administrateur je veux ajouter un plan | 2.1.1 | Création de l’interface d'ajout | 1h | chaima |
| 2.1.2 | Développer la méthode ajouter plan () | 3h | chaima |
| 2.2 | en tant que administrateur je veux modifier un plan | 2.2.1 | Création de l’interface de modification | 1h | chaima |
| 2.2.2 | Développer la méthode modifier plan () | 3h | chaima |
| 2.3 | en tant que administrateur je veux supprimer un plan | 2.3.1 | Création de l’interface de suppression | 1h | chaima |
| 2.3.2 | Développer la méthode supprimer plan () | 2h | chaima |
| 2.4 | en tant que administrateur je veux consulter la liste des plans | 2.4.1 | Création de l’interface de consultation | 1h | chaima |
| 2.4.2 | Développer la méthode Affichage | 2h | chaima |
| 2.4.3 | Développer la méthode de recherche () | 3h | chaima |
| 2.5 | en tant que administrateur je peux valider un plan | 2.5.1 | Création de l’interface de validation | 1h | chaima |
| 2.5.2 | Développer la méthode validation | 2h | Mahdi |
| 3.1 | En Tant que Client je peux ajouter un bon plan | 3.1.1 | Création de l'interface d'ajout d'un plan avec javaFx (utilisation des labels, textfields et des boutons) | 1h | Chaima |
| 3.1.2 | Développer les méthodes d'ajout d'un bon plan | 3h | Chaima |
| 3.1.3 | Tester de la méthode d'ajout d'un plan dans la base (Les requêtes d'insertions qui correspondent à chaque catégorie) | 2h | Chaima |
| 3.2 | En Tant que Client je peux Consulter la liste de mes plans | 3.2.1 | Création de l'interface de consultation javaFx (utilisation d'un tableView) | 1h | Chaima |
| 3.2.2 | Développer la méthode d'affichage de la liste des plans (requête select) | 4h | Chaima |
| 3.2.3 | Tester la méthode d'affichage des bons plans | 2h | Chaima |
| 3.3 | En Tant que Client je peux Modifier mes plans | 3.3.1 | Création de l'interface de modification javaFx (utilisation d'un tableView) | 1h | Chaima |
| 3.3.2 | Développer la méthode de modification d'un plan (requête Update) | 3h | Chaima |
| 3.3.3 | Tester la méthode de Modification d'un plan | 2h | Chaima |
| 3.4 | En tant que Client je peux Supprimer un plan de la liste de mes plans | 3.4.1 | Création de l'interface de Suppression javaFx (utilisation d'un tableView) | 1h | Chaima |
| 3.4.2 | Développer la méthode de suppression d'un plan (requête Delete) | 3h | Chaima |
| 3.4.3 | Tester la méthode de Suppression d'un plan | 2h | Chaima |
| 3.5 | En Tant que Client je peux ajouter un Evènement | 3.5.1 | Création de l'interface d'ajout d'évènement avec javaFx (utilisation des labels, textfields et des boutons) | 2h | Lina |
| 3.5.2 | Développer les méthodes d'ajout d'un Evènement | 2h | Lina |
| 3.5.3 | Tester la méthode d'Ajout d'un Evènement | 2h | Lina |
| 3.6 | En Tant que Client je souhaite Consulter la liste de mes évènements | 3.6.1 | Création de l'interface de l'affichage d'un Evènement avec javaFx (utilisation des labels, ViewTable et des boutons) | 2h | Lina |
| 3.6.2 | Développer la méthode d'affichage de la liste des Evènements (requête select) | 2h | Lina |
| 3.6.3 | Tester la méthode d'affichage des Evènements ("Mes Evènements") | 2h | Lina |
| 3.7 | En Tant que Client je peux Modifier un Evènement qui existe dans ma liste | 3.7.1 | Création de l'interface de modification javaFx (utilisation d'un tableView) | 2h | Lina |
| 3.7.2 | Développer la méthode de modification d'un plan (requête Update) | 2h | Lina |
| 3.7.3 | Tester la méthode de Modification d'un plans | 2h | Lina |
| 3.8 | En Tant que Client je peux Supprimer un Evènement qui existe dans ma liste | 3.8.1 | Création de l'interface de Suppression javaFx (utilisation d'un tableView, bouton Suppression) | 2h | Lina |
| 3.8.2 | Développer la méthode de suppression d'un Evènement (requête Delete) | 2h | Lina |
| 3.8.3 | Tester la méthode de Suppression d'un Evènement | 2h | Lina |
| 4.1 | En tant que internaute je peux consulter les Bons Plans | 4,1,1 | Création de l'interface d'accueil javaFx() | 10h | Ahmed |
| 4,1,2 | Développer la méthode de consultation des bons plans | 2h | Chaima |
| 4,1,3 | Développer la méthode de consultation des évènements | 3h | Lina |
| 4,1,4 | Tester les deux méthodes | 1h | Chaima |
| 4.2 | En Tant que internaute je peux chercher un plans | 4.2.1 | Création de l'interface de recherche | 1h | Chaima |
| 4.2.2 | Intégration de l'API MAP avec l'ajout des marqueurs | 8h | Chaima |
| 4.2.3 | Création des champs et outils de recherche avancée | 3h | Ahmed |
| 4.2.3 | Développer les méthodes de recherche dynamique avec des mots clés prédéfinies | 5h | Ahmed |
| 4.2.4 | Tester la rechercher | 2h | Chaima & Ahmed |
| 5.1 | En tant qu'un client, je peux ajouter un commentaire sur un plan | 5.1.1 | Création de l'interface des commentaires du plan javaFx() | 2h | Omar |
| 5.1.2 | Développer la méthode ajouter commentaire () | 2h | Omar |
| 5.1.3 | Tester la méthode d'Ajout d'un commentaire | 2h | Omar |
| 5.2 | En tant qu'un client, je peux modifier un commentaire sur un plan | 5.2.1 | Création de l'interface de modification des commentaires du plan javaFx() | 2h | Omar |
| 5.2.2 | Développer la méthode modifier commentaire () | 2h | Omar |
| 5.2.3 | Tester la méthode de modification d'un commentaire | 2h | Omar |
| 5.3 | En tant qu'un client, je peux supprimer mon commentaire sur un plan | 5.3.1 | Création de l'interface de suppression des commentaires du plan javaFx() | 2h | Omar |
| 5.3.2 | Développer la méthode supprimer commentaire () | 2h | Omar |
| 5.3.3 | Tester la méthode supprimer d'un commentaire | 2h | Omar |
| 5.4 | En tant qu'un client, je peux consulter la liste des commentaires sur un plan | 5.4.1 | Création de l'interface des commentaires du plan javaFx() | 2h | Omar |
| 5.4.2 | Développer la méthode afficher commentaire () | 2h | Omar |
| 5.4.3 | Tester la méthode d'affichage d'un commentaire | 2h | Omar |
| 5.5 | En tant qu'un client, je peux noter un plan | 5.5.1 | Création de l'interface de l'évaluation des commentaires du plan javaFx() | 2h | Omar |
| 5.5.2 | Développer la méthode d'évaluation commentaire () | 2h | Omar |
| 5.5.3 | Tester la méthode d'évaluation d'un commentaire | 2h | Omar |
| 5.6 | En tant qu'un client, je peux signaler un plan | 5.6.1 | Création de l'interface de signalisation des plans javaFx() | 2h | Omar |
| 5.6.2 | Développer la méthode signaler plan () | 2h | Omar |
| 5.6.3 | Tester la méthode de signalisation d'un plan | 2h | Omar |
| 6,1 | En Tant que Client je peux ajouter une publicité | 6.1.1 | Création de l'interface de gestion des publicités avec SceneBuilder ("Mes publicités") | 3h | Ahmed |
| 6.1.2 | Développer la méthode d'ajout d'une publicité avec l'intégration des images | 5h | Ahmed |
| 6.1.3 | Tester l'ajout dans le tableau "publicité" de notre base de données | 1h | Ahmed |
| 6,2 | En Tant que Client je souhaite Consulter la liste de mes publicités | 6.2.1 | Création du tableView dans l'interface de gestion des publicités | 6h | Ahmed |
| 6.2.2 | Développer la méthode d'affichage des publicités dans le tableau | 2h | Ahmed |
| 6.2.3 | Tester la réussite d'affichage des champs nécessaires | 1h | Ahmed |
| 6,3 | En Tant que Client je peux Modifier une publicité qui existe dans ma liste | 6.3.1 | Développer la méthode de remplissage des champs avec les infos existants concernant la publicité choisie | 4h | Ahmed |
| 6.3.2 | Développer la méthode qui fait la mise à jour de ces champs suivant les modifications faites par l'utilisateur | 3h | Ahmed |
| 6.3.3 | Tester la réussite de mise à jour des infos et des photos qui caractérisent la publicité modifiée | 1h | Ahmed |
| 6,4 | En Tant que Client je peux Supprimer une publicité qui existe dans ma liste | 6.4.1 | Développer la méthode de suppression des publicités | 1h | Ahmed |
| 6.4.2 | Tester la suppression dans la base de données aussi que dans le tableau affiché pour l'utilisateur d'une manière simultanée | 1h | Ahmed |
| Ahmed |
| 6,5 | En Tant que Client je peux partager ma publicité sur les réseaux sociaux | 6.4.1 | Intégration de l'API SHARE sur les réseaux sociaux | 5h | Ahmed |
| 6.4.2 | Tester l’API et le partage avec succès sur les réseaux sociaux choisis par l'utilisateur | 2h | Ahmed |
| Ahmed |

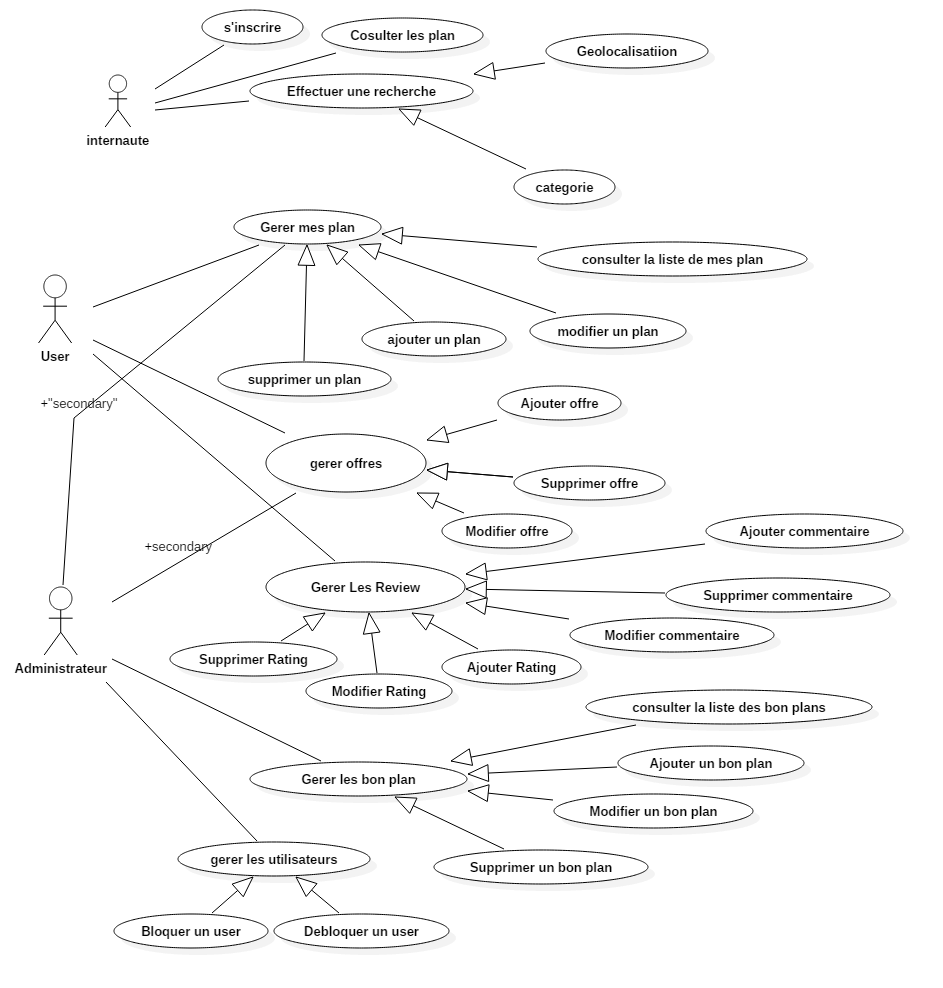
# Diagramme de Cas d’utilisation :

Le diagramme de cas d’utilisation est le premier diagramme UML qui permet de présenter

les acteurs du système ainsi que les fonctionnalités. C’est un diagramme simple et facile à lire

qui donne une vue globale sur le système.

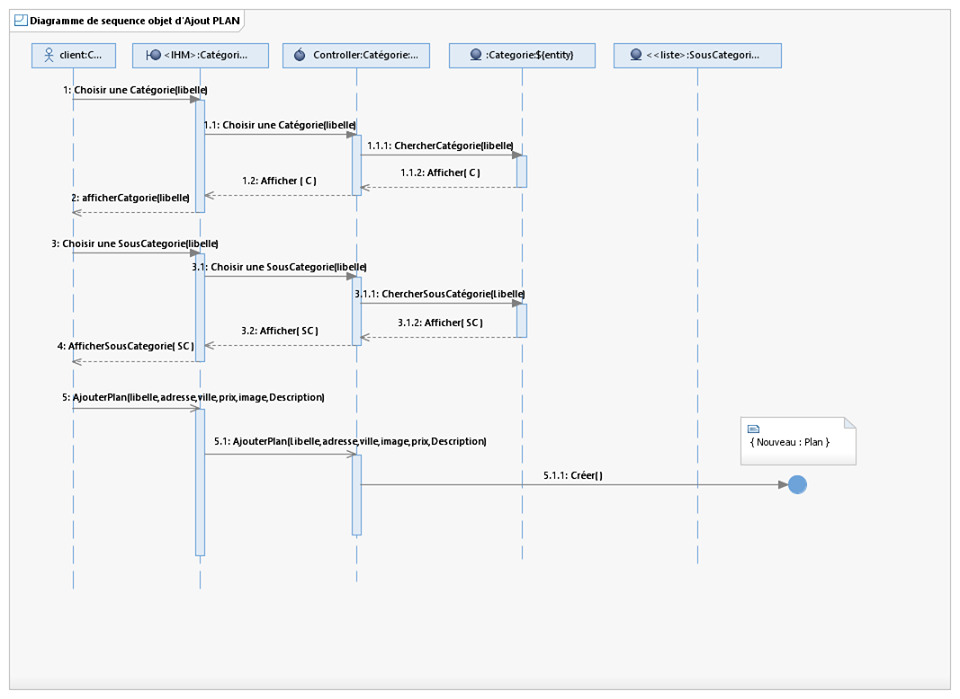
La figure ci-dessus présente le diagramme de cas d’utilisation globale de notre application.

****

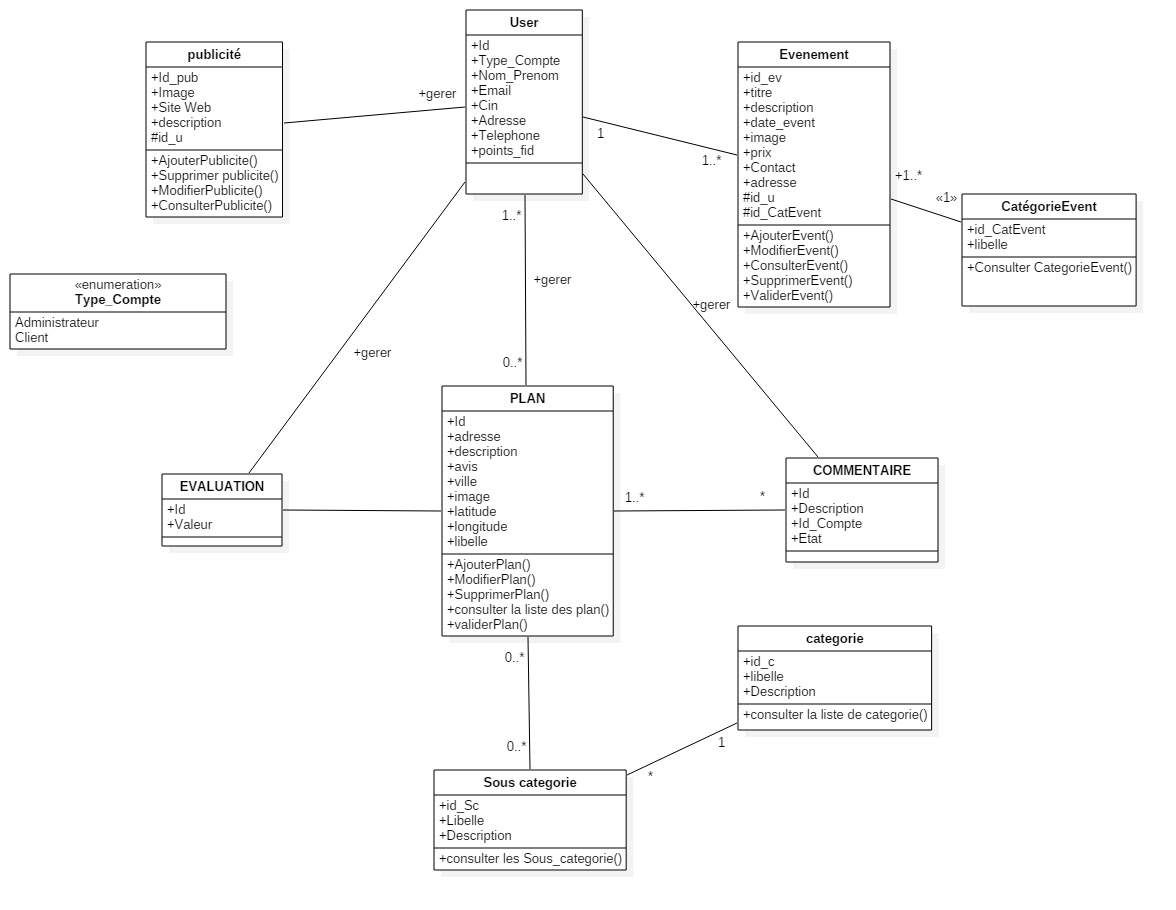
# Conception de vue dynamique

La conception de vue dynamique permet de montrer le fonctionnement du système via les diagrammes de séquence objet et les diagrammes d’activité.

## Diagramme de séquence objet :

****

## 3.2. Diagramme de Classe  :



# Burndown chart :

Dans le cas du premier Sprint, tous les taches de la semaine 1 ont été achevés à temps. Cela n’empêche que certaines tâches ont été sous-estimées au niveau de l’effort à consacrer. A la fin de sprint, une réunion de rétrospective a eu lieu afin de discuter les raisons du retard de l’accomplissement des tâches et dont l’objectif est d’améliorer les estimations d’effort pour atteindre des trajectoires d’effort restant presque idéales dans les prochains sprints.