**CHAPITRE 2 : ANALYSE ET CONCEPTION**

**Réalisée par: QasmiNabil/OthmaneBennanikella :**

1. **Méthode choisie.**

**Avant de commencer le développement du projet, nous avons pris soin de conceptualiser notre approche et de la formaliser à l'aide de la méthode UML. Nous présenterons en détail cette démarche au cours du troisième chapitre, lors de notre examen approfondi des aspects technologiques.**

1. **méthode Kanban :**

**Les tâches à réaliser sont représentées sous forme de cartes sur un tableau Kanban, avec des colonnes représentant les différentes étapes du processus. Chaque colonne peut représenter des états tels que "À faire", "En cours" et "Terminé". Les membres de l'équipe tirent des cartes de la colonne "À faire" et les déplacent à travers les différentes colonnes du tableau jusqu'à ce qu'elles soient complètement réalisées**

**-Exemple d utilisation :**

**Intro GitHub/kanban :**

**L'utilisation de la méthode Kanban avec GitHub présente plusieurs avantages :**

**Gestion visuelle du flux de travail : GitHub permet de créer des tableaux Kanban pour visualiser le flux de travail des tâches. Cela offre une vue d'ensemble claire de l'état actuel des travaux, des tâches en cours et de celles qui sont en attente.**

**Suivi des tâches en temps réel : Les tableaux Kanban de GitHub sont mis à jour en temps réel, ce qui permet à toute l'équipe de voir instantanément l'état des tâches et de suivre leur progression.**

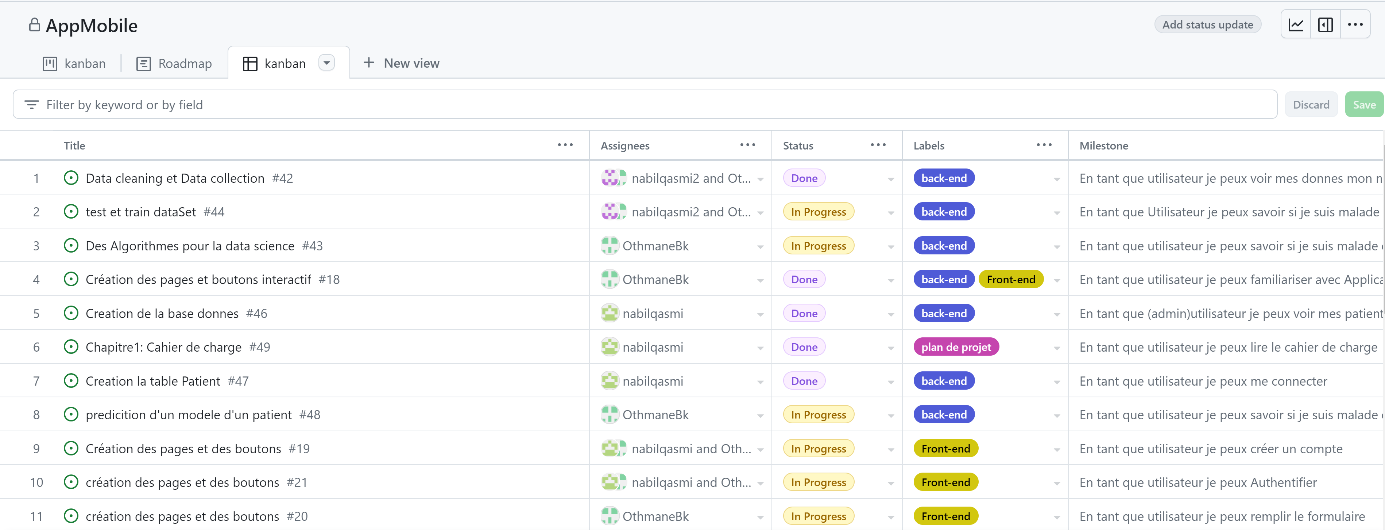
**Priorisation flexible : Les équipes peuvent facilement déplacer les tâches d'une colonne à l'autre sur le tableau Kanban pour refléter les changements de priorité ou les nouveaux besoins, ce qui permet une gestion flexible et réactive des travaux.**

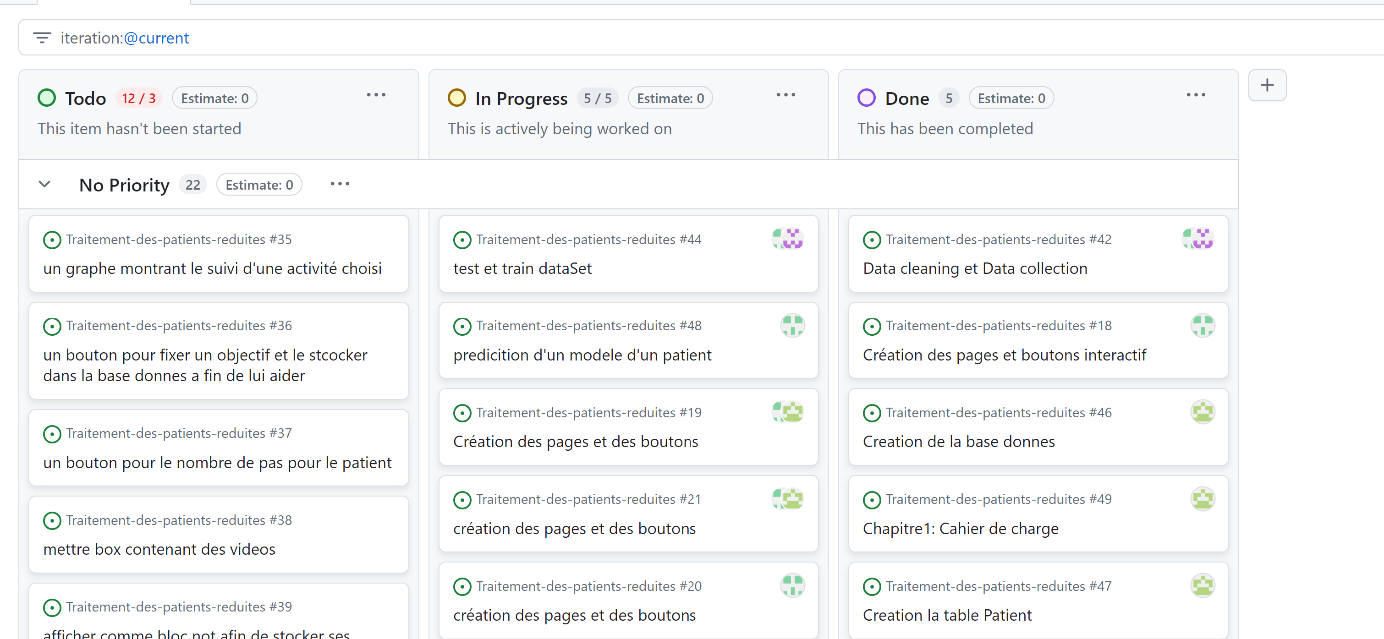
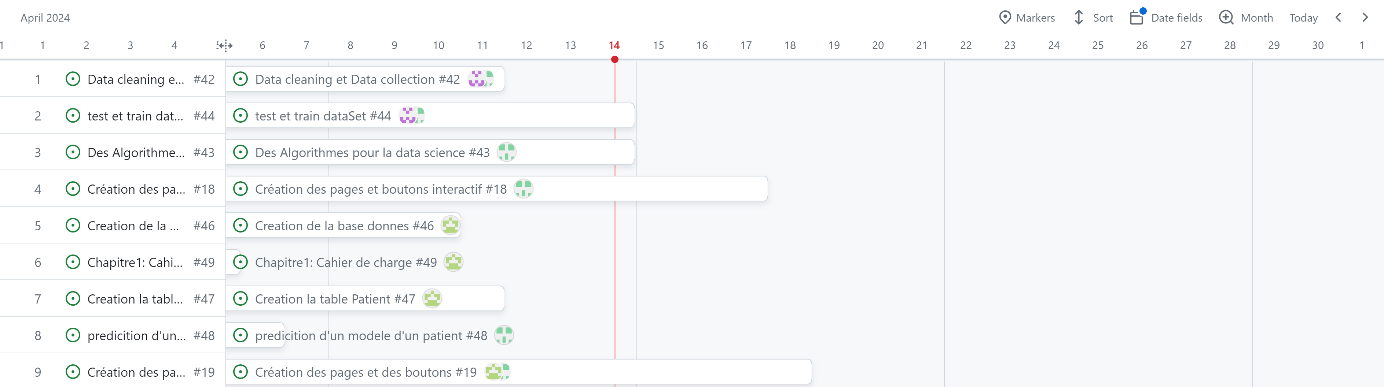
**Collaboration transparente : GitHub offre des fonctionnalités de commentaires et de révision de code intégrées dans les pull requests, ce qui permet aux membres de l'équipe de collaborer efficacement sur les tâches. Les commentaires peuvent être laissés directement sur les tâches du tableau Kanban pour discuter des détails ou des problèmes rencontrés.**

**Amélioration continue : La méthode Kanban encourage l'amélioration continue en identifiant et en éliminant les goulots d'étranglement et les inefficacités dans le processus de développement. GitHub fournit des métriques et des analyses pour évaluer les performances de l'équipe et identifier les domaines à améliorer.**

**En résumé, l'utilisation de la méthode Kanban avec GitHub facilite la gestion du flux de travail, la collaboration entre les membres de l'équipe et l'amélioration continue du processus de développement. Cela conduit à une meilleure visibilité, une plus grande efficacité et une livraison plus rapide des projets logiciels.**

**.**

****

****

1. **Conception du projet.**

**Dans ce chapitre, nous examinerons tous les cas d'utilisation du système. Pour ce faire, nous nous pencherons sur les aspects suivants :**

**Qui sont les utilisateurs du système ?**

**Quelles sont leurs attentes vis-à-vis de celui-ci ?**

**1.Analyse fonctionnelle**

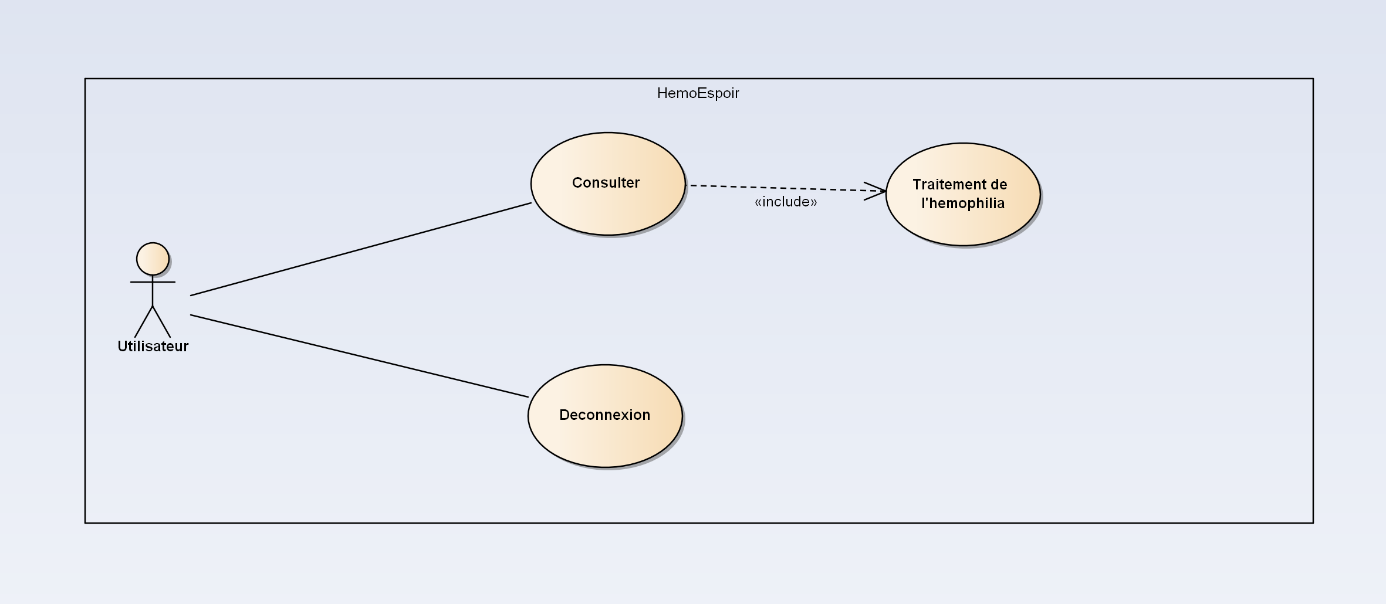
**a) Acteur /rôle/définition(Descriptif)**

|  |  |  |
| --- | --- | --- |
| **Acteur :**  **L utilisateur** | **Définition :**  **Les utilisateurs téléchargeant l'application bénéficieront d'un service leur permettant d'accéder à un suivi médical complet et fiable.** | **Rôle :**  **fournir un service de suivi médical complet et fiable aux utilisateurs qui la téléchargent** |

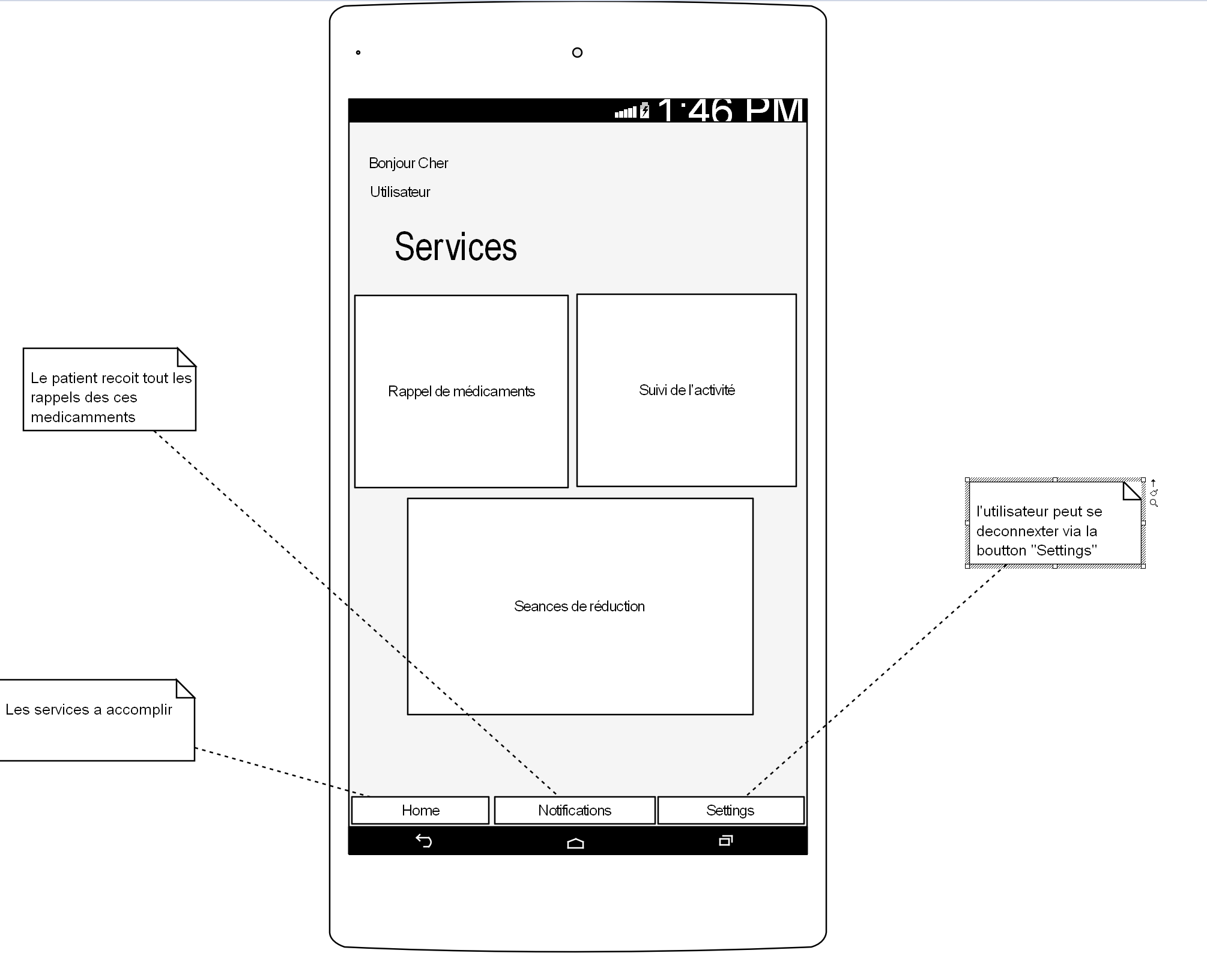
**B)** **Diagramme illustrant les cas d'utilisation de manière graphique.**

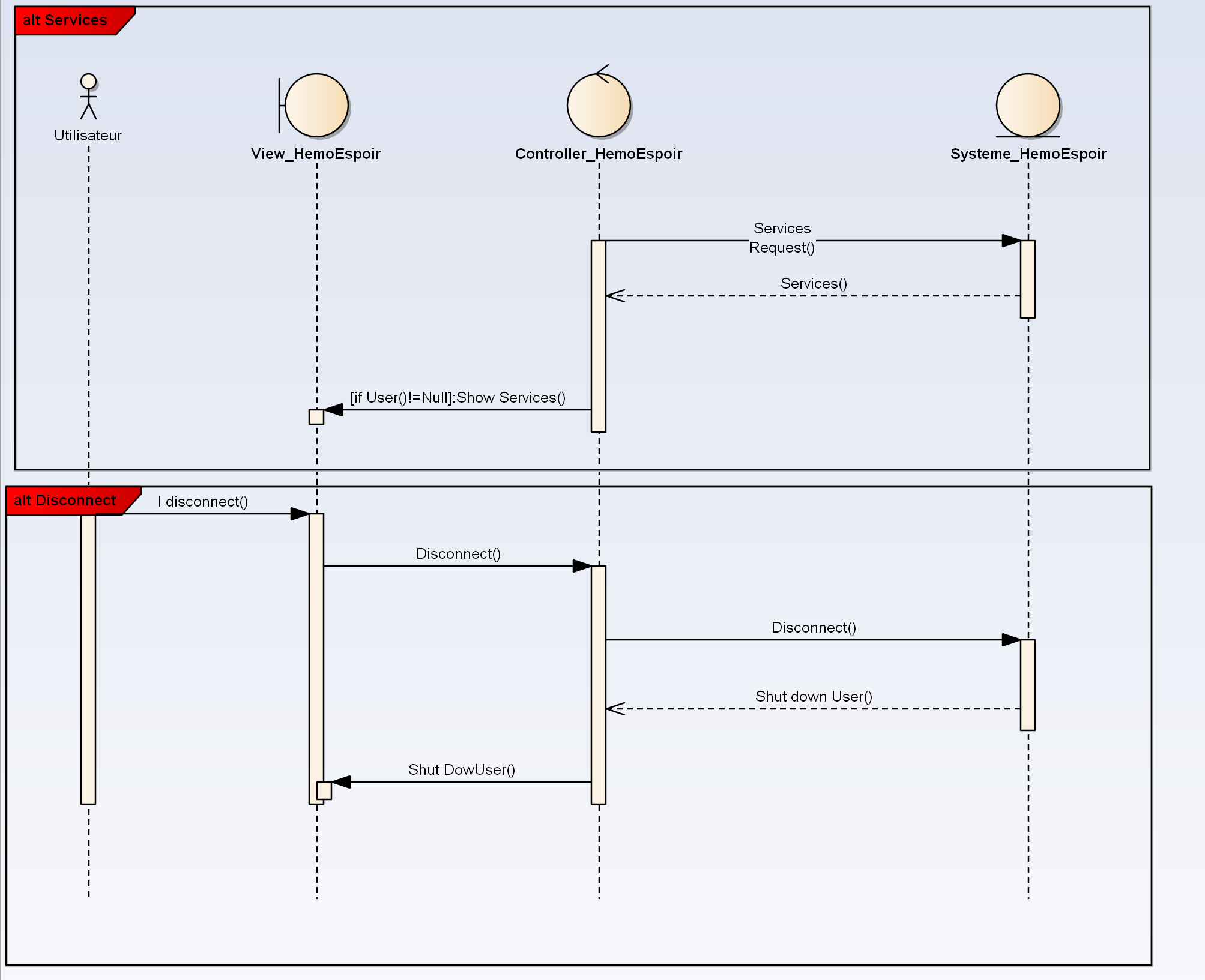
**En Langage de Modélisation Unifié (UML), un diagramme de cas d'utilisation synthétise les rôles des acteurs du système ainsi que leurs interactions avec celui-ci.**

**1 :Les fonctionnalités que l'utilisateur peut effectuer depuis la page d'accueil.**

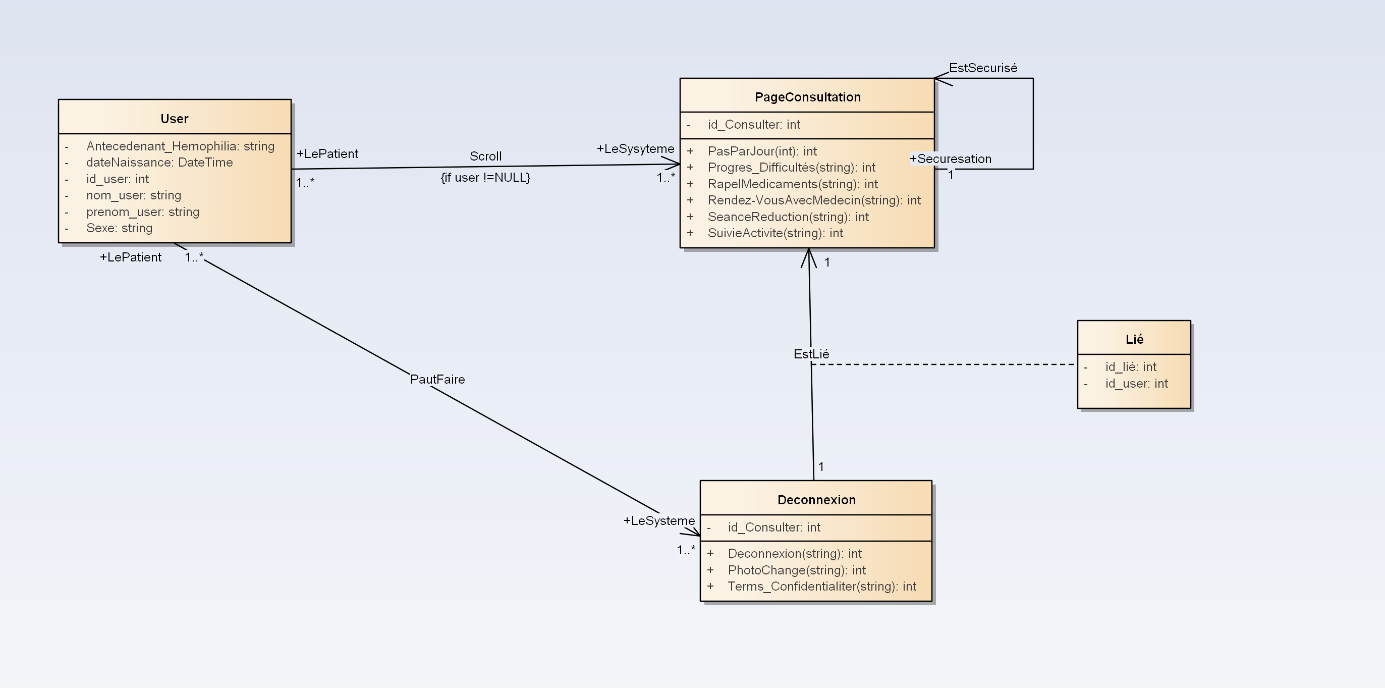
****

**2 : Voici les services fournis par notre application :**

****

**3 : Un mode de vie intégrant les fonctionnalités de notre application.**

**4-Voici Une Ilustration(Diagramme de Classe)**

****

**Conclusion :**

**À travers ce chapitre, toutes les interrogations relatives à la conception du système à développer ont été clarifiées. Le résultat obtenu est un modèle graphique (ensemble de diagrammes) prêt à être implémenté. Dans le chapitre suivant, nous examinerons de manière approfondie les langages utilisés lors de la phase de développement.**