**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.**

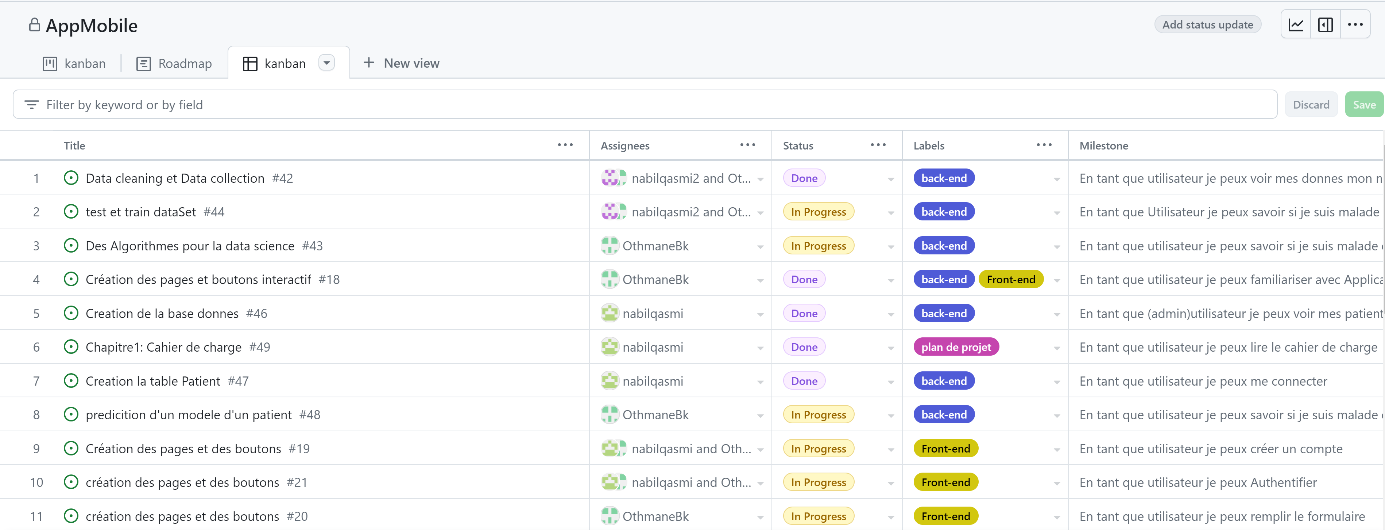
**Priorisation flexible : Les équipes peuvent facilement déplacer les tâches d'une colonne à l'autre sur le tableau Kanban pour refléter les changements de priorité ou les nouveaux besoins, ce qui permet une gestion flexible et réactive des travaux.**

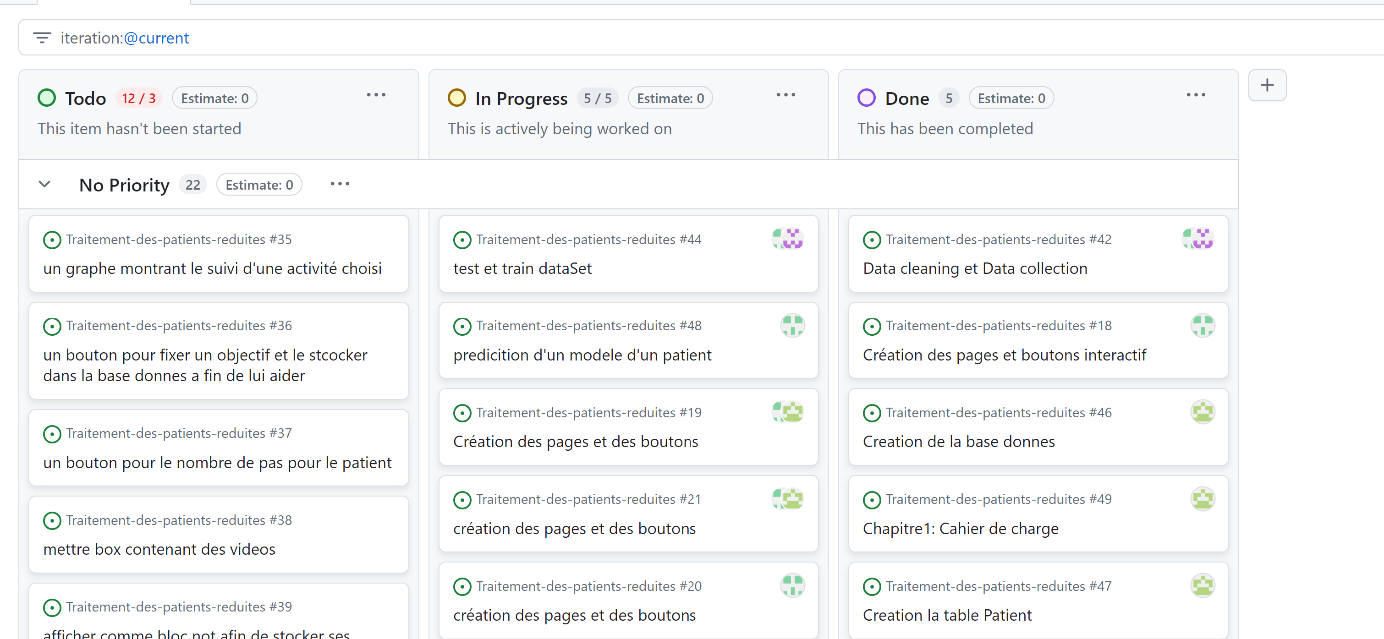
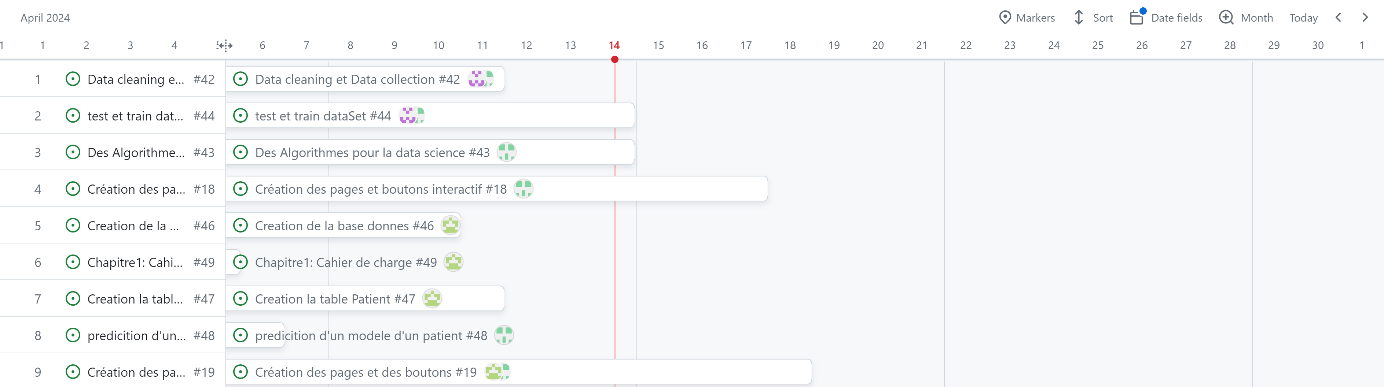
**Collaboration transparente : GitHub offre des fonctionnalités de commentaires et de révision de code intégrées dans les pull requests, ce qui permet aux membres de l'équipe de collaborer efficacement sur les tâches. Les commentaires peuvent être laissés directement sur les tâches du tableau Kanban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

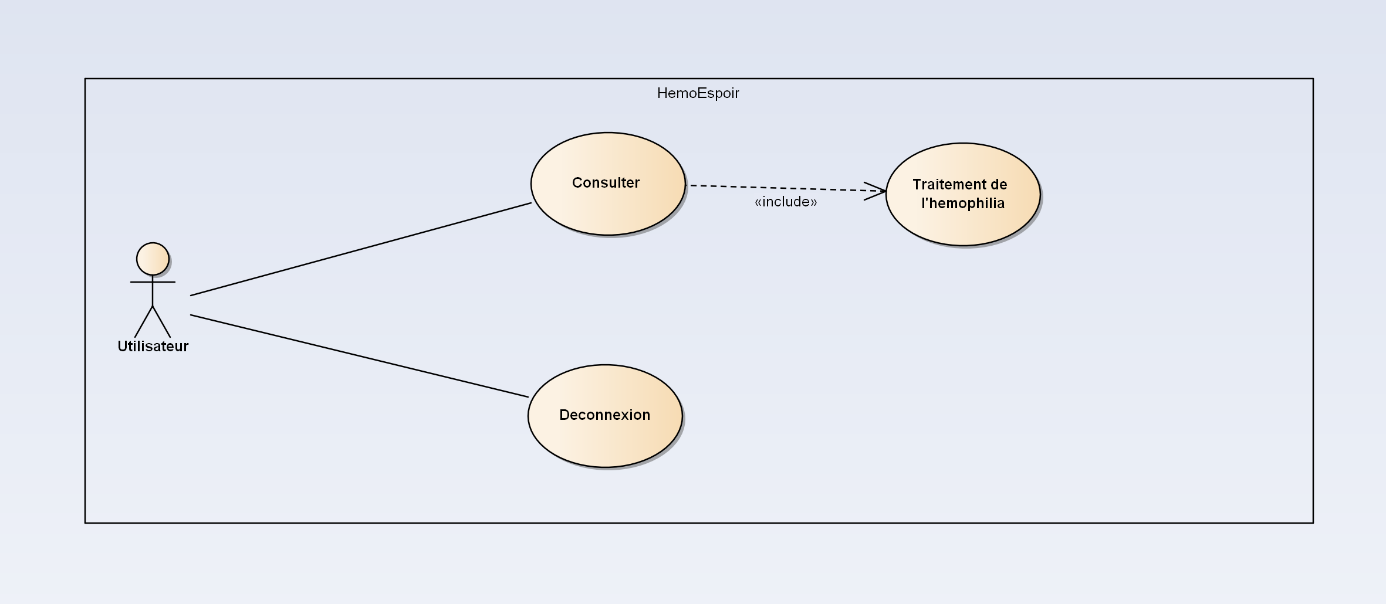
**a) Acteur /rôle/définition(Descriptif)**

|  |  |  |
| --- | --- | --- |
| **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** |

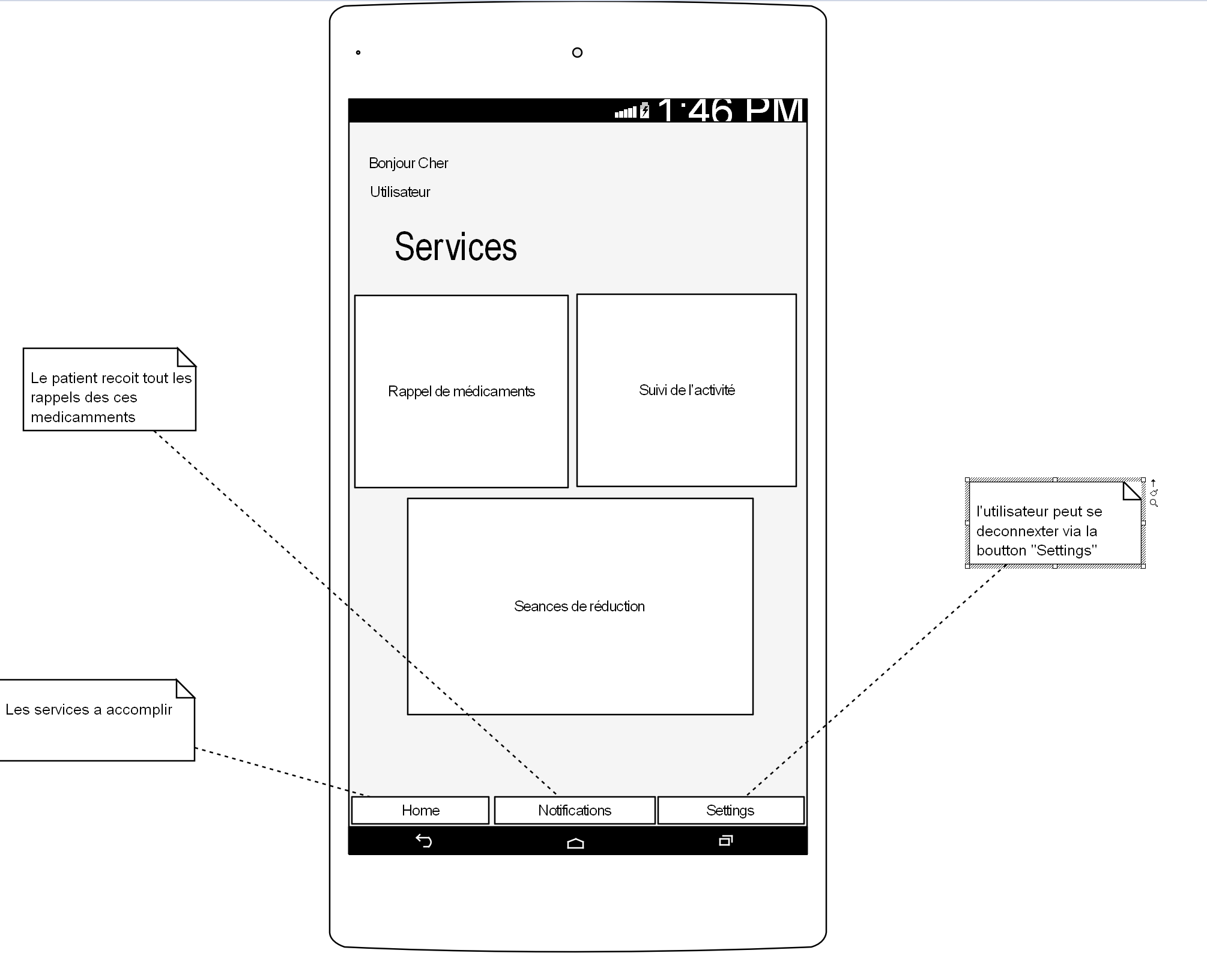
**B)** **Diagramme illustrant les cas d'utilisation de manière graphique.**

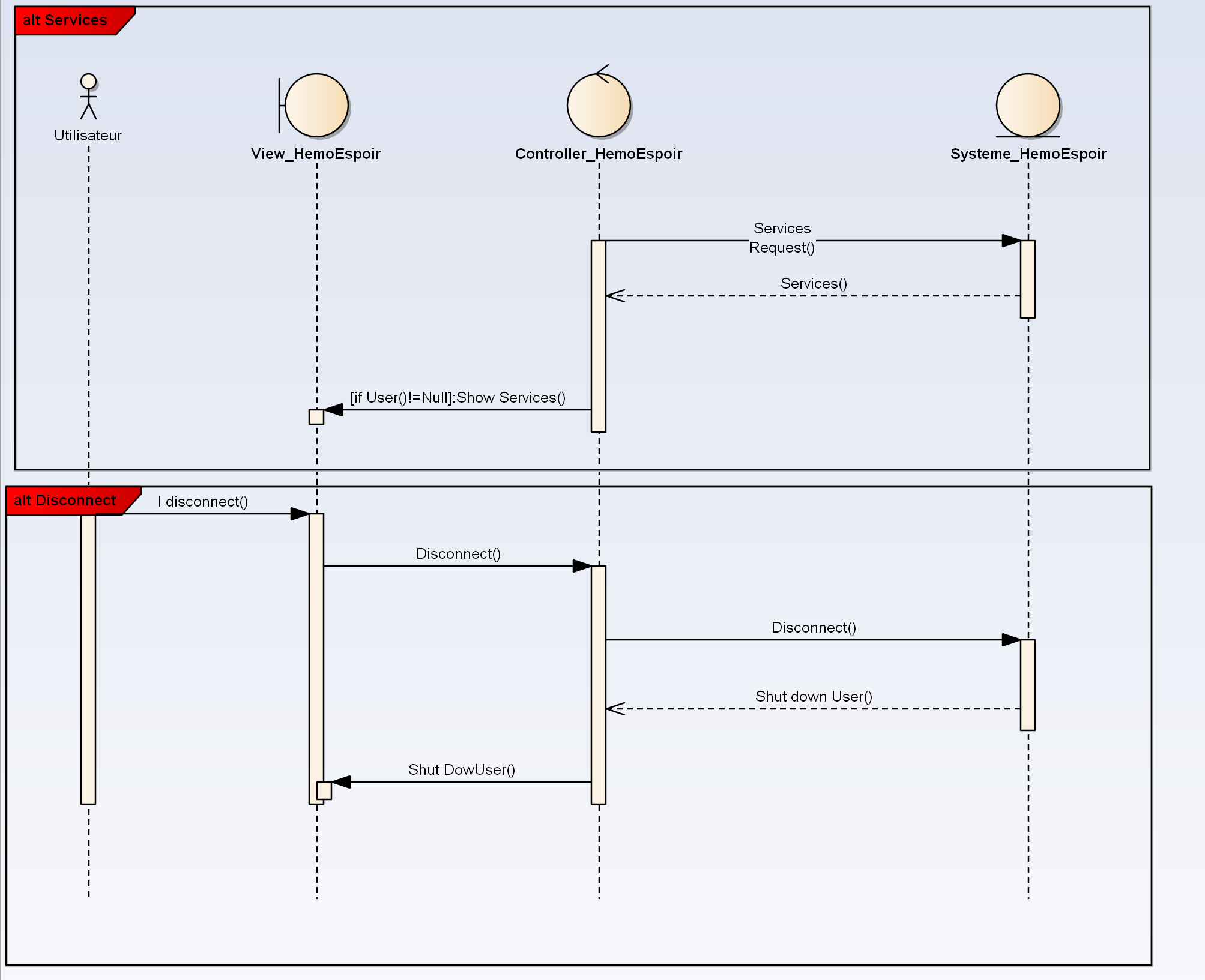
**En Langage de Modélisation Unifié (UML), un diagramme de cas d'utilisation synthétise les rôles des acteurs du système ainsi que leurs interactions avec celui-ci.**

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

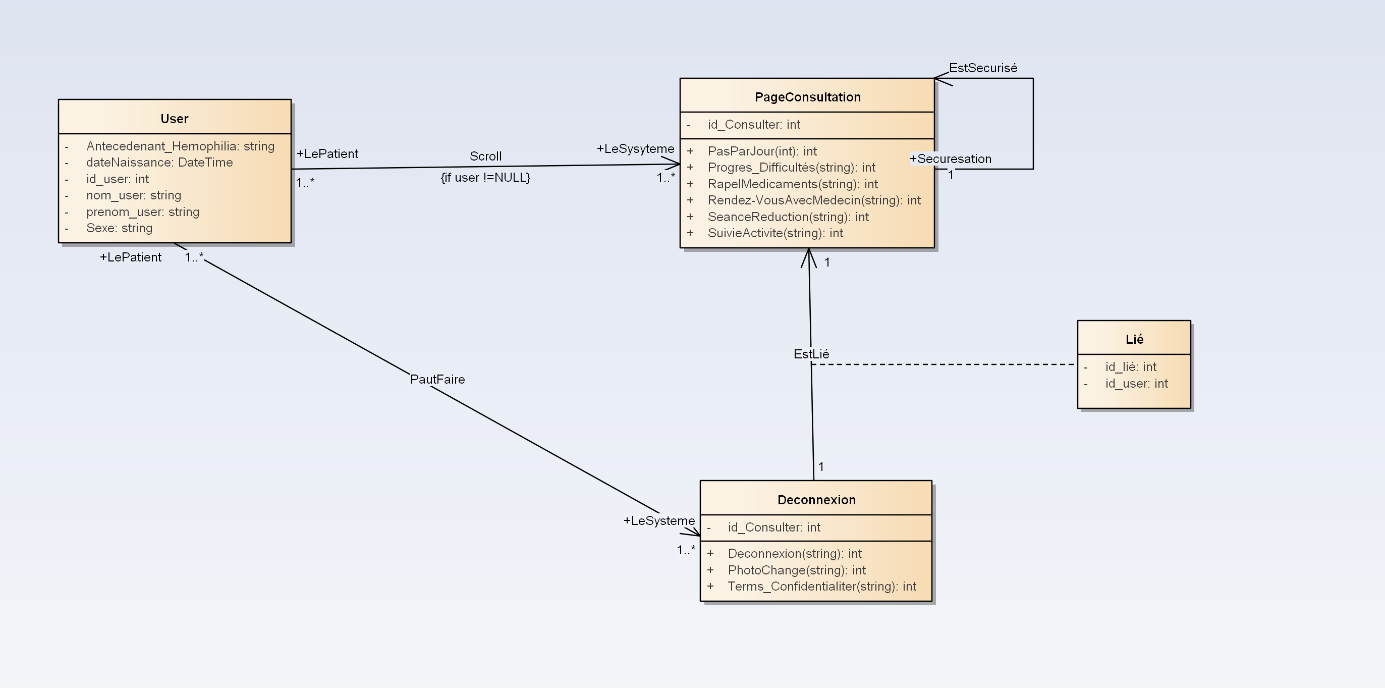
****

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

****

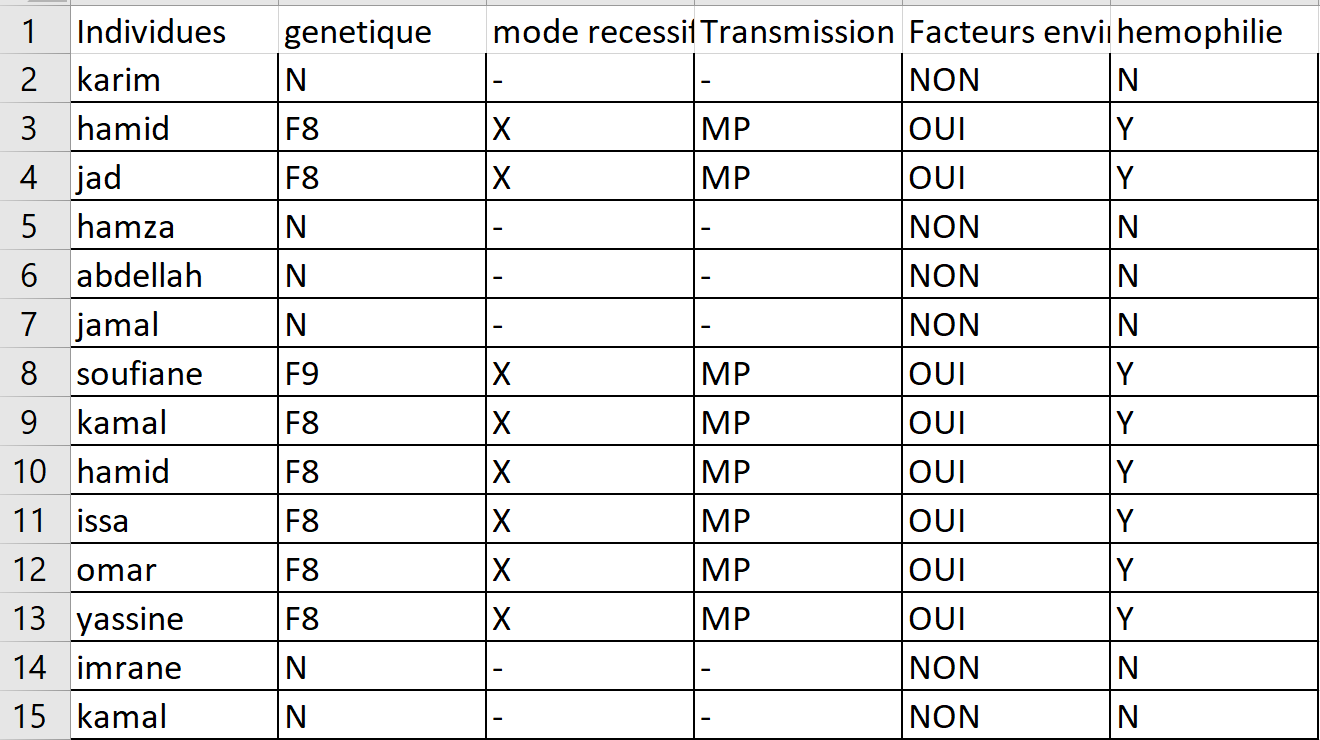
**3 : Un mode de vie intégrant les fonctionnalités de notre application.**

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

****



**III- DataScience**

****1- La Collection des donnes

L'hémophilie est un trouble héréditaire du sang qui affecte la capacité de coagulation. Les personnes atteintes d'hémophilie ont un déficit en facteur de coagulation, une protéine nécessaire pour que le sang coagule normalement. Il existe deux types principaux d'hémophilie : l'hémophilie A, causée par un déficit en facteur VIII, et l'hémophilie B, causée par un déficit en facteur IX.

Les symptômes de l'hémophilie incluent des saignements prolongés après une blessure ou une chirurgie, des ecchymoses fréquentes, des saignements de nez récurrents, des saignements des gencives, des saignements dans les articulations (hémarthrose) et des saignements internes graves.

Le traitement de l'hémophilie consiste généralement à remplacer le facteur de coagulation manquant à l'aide de concentrés de facteurs de coagulation ou de produits dérivés du plasma. Les personnes atteintes d'hémophilie peuvent également bénéficier de médicaments pour stimuler la production de facteurs de coagulation ou pour prévenir les saignements.

La prise en charge de l'hémophilie comprend également des mesures pour prévenir les blessures et les saignements, ainsi que pour traiter rapidement les saignements lorsqu'ils surviennent. Un suivi médical régulier est important pour surveiller l'état de santé et ajuster le traitement si nécessaire.

Mode ressecif : Le terme "mode récessif paternel" ne correspond pas à une caractéristique spécifique de l'hémophilie. En général, l'hémophilie est héritée de manière récessive liée à l'X, ce qui signifie que le gène responsable de l'hémophilie est situé sur le chromosome X. Les hommes ont un chromosome X et un chromosome Y, tandis que les femmes ont deux chromosomes X.

Si un homme hémophile a des enfants, toutes ses filles hériteront du chromosome X normal de leur mère et seront des porteuses saines. Ses fils hériteront du chromosome Y de leur père et ne seront pas atteints d'hémophilie. Cependant, si une femme porteuse d'un gène de l'hémophilie a des enfants avec un homme non porteur, il y a 50% de chances que ses fils héritent de la mutation et développent l'hémophilie.

Le terme "mode récessif paternel" pourrait être une confusion avec la manière dont l'hémophilie est héritée, mais il n'a pas de signification précise dans le contexte de l'hémophilie.

Mère porteuse:

Une mère porteuse de l'hémophilie est une femme qui possède un gène de l'hémophilie sur l'un de ses chromosomes X, mais qui ne présente pas elle-même les symptômes de l'hémophilie. Comme l'hémophilie est une maladie récessive liée à l'X, les femmes porteuses ont généralement un chromosome X normal qui compense le chromosome X porteur de la mutation.

Les mères porteuses de l'hémophilie ont un risque de 50% de transmettre le gène de l'hémophilie à leurs enfants. Si elles transmettent le chromosome X porteur de la mutation à leur fils, celui-ci aura un risque élevé de développer l'hémophilie. En revanche, si elles transmettent le chromosome X normal à leur fils, celui-ci ne sera pas atteint d'hémophilie, mais sera porteur du gène et pourra le transmettre à ses propres enfants. Les filles des mères porteuses ont également un risque de 50% d'être porteuses du gène de l'hémophilie.

F9/f8: Les facteurs VIII (F8) et IX (F9) sont des protéines de coagulation sanguine. L'hémophilie A est causée par un déficit en facteur VIII, tandis que l'hémophilie B est causée par un déficit en facteur IX.

- \*Facteur VIII (F8)\* : Le facteur VIII est une protéine nécessaire à la coagulation du sang. En cas de déficit en facteur VIII, la capacité du sang à coaguler normalement est compromise, ce qui entraîne des saignements prolongés après une blessure ou une chirurgie, des ecchymoses faciles, des saignements de nez fréquents et d'autres symptômes.

- \*Facteur IX (F9)\* : Le facteur IX est une autre protéine de coagulation importante. L'hémophilie B, également appelée hémophilie de Christmas, est causée par un déficit en facteur IX. Les symptômes de l'hémophilie B sont similaires à ceux de l'hémophilie A, avec des saignements prolongés et des ecchymoses.

Ce qui concerne le genetique d’une personne, on distingue trois types :

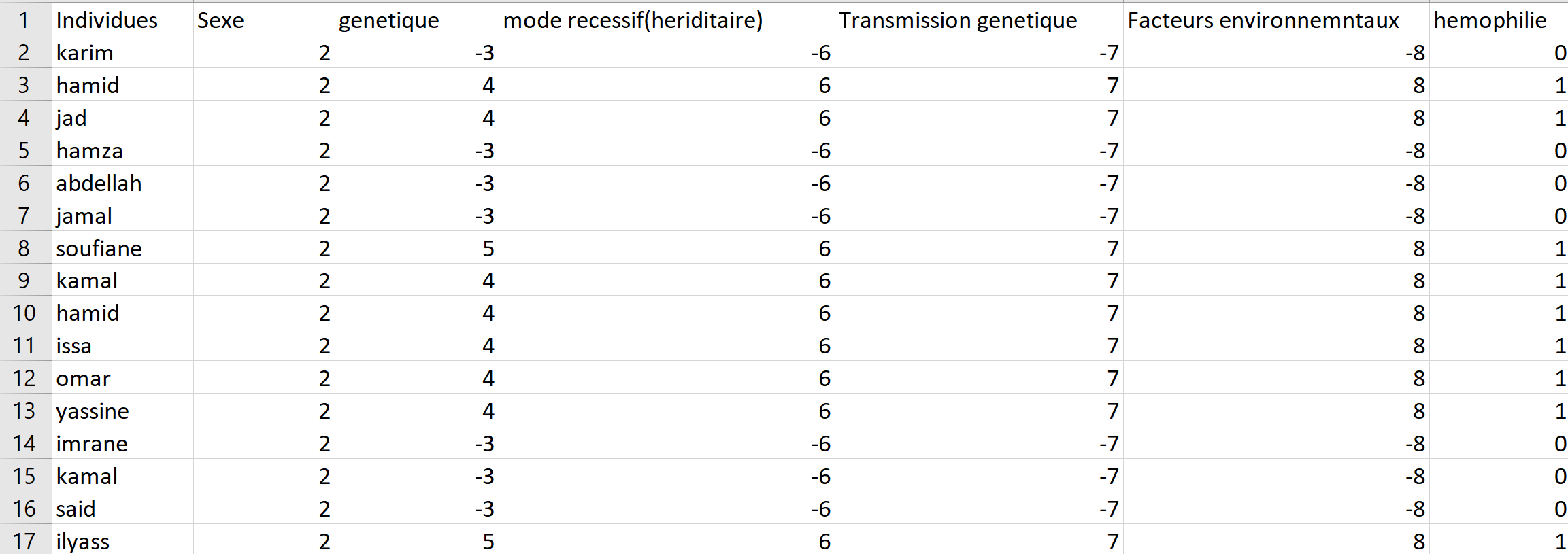
Genetique N signifie, que la personne est saine, Gnetique F8 signifie que la personne a l’hemophilia de type A, par contre F9 est associé a type B

En résumé, la principale différence entre l'hémophilie A (déficit en facteur VIII) et l'hémophilie B (déficit en facteur IX) réside dans le facteur de coagulation spécifique qui est déficient. Les deux types d'hémophilie présentent des symptômes similaires et sont traités de manière similaire, avec des injections de facteurs de coagulation manquants pour aider à contrôler les saignements.

2- Traitement des donnes

Traitement des donnes

La légende de la dataSet



Hémophile -> 0(non malade)

-> 1(malade)

Facteurs environnementaux -> 8(OUI)

-> -8(NON)

Transmission génétique-> -7(absence -)

-> 7(mère porteuse)

Mode récessif(héréditaire) -> -6(absence -)

-> 6(X)

Génétique -> -3(N)

-> 4(F8)

-> 5(F9)

Sexe -> 2(H)

-> -2(F)

3- Des outils de la Science des donnes

Pendant la phase des donnes, nous avons l’oportunites de decouvrir des nouvelles modules tels :

from sklearn.model\_selection import train\_test\_split

from sklearn.neighbors import KNeighborsClassifier

from sklearn.naive\_bayes import GaussianNB

from sklearn.tree import DecisionTreeClassifier

qui serviront a la science de donnes, pour mieux prédire le model cad quand un nouvau patient s’ajoute, il faut tester si cet element est malade ou non

1. Models

Modele 1

1. #train\_test\_split=la manipulation de données dans l'apprentissage automatique
2. #test\_size=0.2 c'est pour les donne de test et 0.8 c'est pour les donnes d'entrainement
3. X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y.values.ravel(), test\_size=0.2, random\_state=101)
4. model = GaussianNB()
5. #CROSS\_VALIDATION
6. cv=KFold(5,random\_state=0,shuffle=True)
7. print("Cross Validation: ",cross\_val\_score(model,X\_train,y\_train,cv=cv,scoring='accuracy').mean())

Modele 2

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y.values.ravel(), test\_size=0.1, random\_state=101)

dtree = DecisionTreeClassifier()

#CROSS\_VALIDATION

cv=KFold(5,random\_state=0,shuffle=True)

print("Cross Validation: ",cross\_val\_score(dtree,X\_train,y\_train,cv=cv,scoring='accuracy').mean())

dtree = dtree.fit(X, Y)

Molele 3

features=['age','Sexe','hemorragie','Transmission genetique','mode recessif(heriditaire)']

X=data[features]

Y = data['hemophilie']

#train\_test\_split=la manipulation de données dans l'apprentissage automatique

#test\_size=0.2 c'est pour les donne de test et 0.8 c'est pour les donnes d'entrainement

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y.values.ravel(), test\_size=0.1, random\_state=101)

model = KNeighborsClassifier()

#CROSS\_VALIDATION

cv=KFold(5,random\_state=0,shuffle=True)

D’apres ces modeles, on distingue le meilleur modele, puisuque notre dataSet est clean tous ces modeles ont une forte resultat sur la dataSet

b- PredictionModel

import pandas as pd

from sklearn.model\_selection import train\_test\_split

import time

from sklearn.neighbors import KNeighborsClassifier

data = pd.read\_excel("C:\\Users\\kella\\Desktop\\Traitement-des-patients-reduites\\Back-end\\dataSet\\dataSetNumbers.xlsx",sheet\_name=1)

features=['age','Sexe','Transmission genetique','mode recessif(heriditaire)']

X=data[features]

Y=data['hemophilie']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size=0.1, random\_state=101)

dtree = KNeighborsClassifier()

dtree = dtree.fit(X, Y)

def decision():

    print('Your name:')

    name= input()

    print('Enter your Sexe:')

    Sexe= input()

    print("What's your age:")

    age= input()

    print('DO you have a chromosome MP heridite from your mother:')

    MP= input()

    print('DO you have a chromosome X heridite from your father:')

    modeRecessif= input()

    SexeNumber = {

        'HOMME': 2,

        'FEMME': -2

        }.get(Sexe.upper(), None)

    modeRecessifNumber = {

        'NO': -6,

        'YES': 6,

        }.get(modeRecessif.upper(), None)

    MPNumber = {

        'NO': -7,

        'YES': 7,

        } .get(MP.upper(), None)

    listOfInformation=[age,SexeNumber,MPNumber,modeRecessifNumber]

    test\_x = pd.DataFrame([listOfInformation], columns=['age','Sexe','Transmission genetique','mode recessif(heriditaire)'])

    print("Attendez pour votre resultat...")

    time.sleep(5)

    try:

        if dtree.predict(test\_x)==[1]:

            print("la resultat est: vous etes malade")

        else:

            print("la resultat est: vous n'etes pas malade")

    except Exception:

        print("Error information")

if \_\_name\_\_=='\_\_main\_\_':

    decision()

Cote Back-End, L’utilisateur Saisie les coordonnes et on retourne la resultat (vous etes malade/vous n’etes pas malade)

c- Statique Descriptives de la DataSet

import pandas as pd

data = pd.read\_excel("C:\\Users\\kella\\Desktop\\Traitement-des-patients-reduites\\Back-end\\dataSet\\dataSetNumbers.xlsx",sheet\_name=1)

#moyenne, colleration, medium, les notions statistiques, plot,..

features=['age','Sexe','Transmission genetique','mode recessif(heriditaire)']

X=data[features]

Y=data['hemophilie']

print(data)

print(data.head())

print(data.describe())

print(data.mean)

new\_dataX=data[features]

print(new\_dataX.corr())

print(data[features[0]].value\_counts())

print(data[features[1]].median())

print(data[features[2]].mode())

print(data[features[0]].std())

print(data[features[2]].cov(data[features[3]]))

data\_sans\_valeurs\_manquantes = data.dropna()

print(data\_sans\_valeurs\_manquantes)

On affiche le data, en affichant head suelement les 4 premier, on fait le description de la DataSet,

On fait la correlation, on compte nombre d’age, on fait la covariance entre deux colonne, on detecte les valeurs manquates