**Conception**

1. **Diagramme de classe**
2. **Diagramme de cas d’utilisation**
3. **Diagramme de séquence**
4. **Diagramme d’états-transitions**
5. **Description des principaux algorithmes**
6. **Analyse**
   1. genererClassement(e:Empreinte,d:dictionnaire): Correspondance[]

On calcule les correspondances entre l'empreinte donnée en paramètre et toutes celles du dictionnaire pour obtenir un tableau des 10 correspondance les plus importante. On ne compte pas les cas où il n’y a pas de risque (“pas de risque”)

* 1. calculerProbabilité(e: Empreinte, m:Maladie)

On compare les attribue de e avec les empreintes de la maladie m un à un. Soit a le nombre d’attributs identiques et n le nombre d’attribut total. La probabilité calculé est égale à : (a/n)\*100, on ne garde dans le tableau de paire correspondance que la plus haute probabilité avec la maladie comparée.

1. **Empreinte**
   1. ajouterAttribut(a:Attribut)

On ajoute un attribut à la liste d’attribut des empreintes

* 1. lancerAnalyse(): Analyse

Appelle la méthode générerClassement de la classe Analyse pour l’empreinte. Cette analyse est ensuite retourner

1. **Maladie**
   1. ajouterEmpreinet(e:Empreinte): boolean

On ajoute un empreinte à la liste d’empreinet de la maladie. Retourne true si l’empreinte a été ajouté sinon false (si cette empreinte exister déjà dans le dictionnaire pour cette maladie par exemple).

* 1. getEmpreinteById (id: Long) : Empreinte

Retourne l’empreinte ayant l’identifiant id. Si l’empreinte n’existe pas, retourne null.

1. **Dictionnaire**
   1. ajouterMaladie(emp: Empreınte): boolean

Vérifie que la maladie n’existe pas dans le dictionnaire avant de l’ajouter avec son empreinte. Si elle est déjà présente, on regarde si l’empreinte donnée pour cette maladie est déjà répertoriée dans le dictionnaire. Si ce n’est pas le cas on ajoute cette empreinte à la maladie dans le dictionnaire.

* 1. getMaladieById(id: Long): Maladie

On retourne la maladie qui possède l'identifiant id. Si la maladie n’existe dans le dictionnaire, on retourne null.

1. **FichierPatient**
   1. analyserEmpreinte(): Analyse []

Pour chaque empreinte de cette classe, on appelle la méthode lancerAnalyse. Chacune des analyse retourner est regroupé dans un tableau que l’on retourne

* 1. ajouterEmpreinte(e:Empreinte): boolean

On vérifie que l’empreinte n’existe pas déjà dans la liste d’empreinte de la classe avant de l’ajouter dans la liste.

1. **FichierEmpStream**
   1. lireFichierPatient(sourceFichier: String): FichierPatient

On ouvre le fichier sourceFichier.txt et on le lit grâce à un objet ifstream. On appelle la classe verifierExtension et verifierSignature. Si les deux retourne True, pour chaque ligne du fichier on crée une Empreinte correspondante que l’on ajoute à un FichierPatient. On retourne ce dernier.

* 1. lireDictionnaire(sourceFichier: String): Dictionnaire

On ouvre le fichier sourceFichier.txt et on le lit grâce à un objet ifstream. On appelle la classe verifierExtension et verifierSignature. Si les deux retourne True, pour chaque ligne du fichier on crée une Maladie correspondante que l’on ajoute au Dictionnaire. On retourne ce dernier.

* 1. verifierExtension(sourceFichier: String): boolean

On récupère l’extension en manipulant le nom du fichier.Retourne vrai si il s’agit d’un extension .txt (celle choisie dans notre cas) sinon false.

* 1. verifierSignature(): boolean

On lit et récupère la signature du fichier et on l’a compare avec la signature renseigner lors du premier chargement de fichier. Retourne true si elles sont égale, sinon false. Dans le cas où il s’agit du premier chargement (signature = null) on retourne forcément true.

1. **main**
   1. renseignerDictionnaire(sourceFichier: String): Dictionnaire

On crée un FichierEmpStream à partir duquel on appelle la méthode lireDictionnaire avec comme paramètre le fichier sourceFichier.

* 1. renseignerFichierPatient(sourceFichier: String): FichierPatient

On crée un FichierEmpStream à partir du quel on appelle la méthode lireFichierPatient avec comme paramètre le fichier sourceFichier.

* 1. afficherTop10(d: Dictionnaire,fp: FichierPatient)

On appelle ma méthode analyserEmpreinte pour le FichierPatient en paramètre fp avec en paramètre le Dictionnaire en paramètre d. Pour chacune des analyse du tableau d’analyse retournée par cette méthode on affiche les maladies des 10 premières paires du tableau correspondance. (Plus ou moins de 10 maladie peuvent être affichées : voir les besoins fonctionnels du TP1)

* 1. afficherMeilleurCorresp(d: Dictionnaire,fp: FichierPatient)

On appelle ma méthode analyserEmpreinte pour le FichierPatient en paramètre fp avec en paramètre le Dictionnaire en paramètre d. Pour chacune des analyse du tableau d’analyse retournée par cette méthode on affiche la maladie de la première paire du tableau correspondance. (plus ou moins d’une maladie peut être affichée : voir les besoins fonctionnels du TP1)

* 1. afficherDictionnaire(d: Dictionnaire)

On affiche toutes les maladies contenues dans la liste de maladie du dictionnaire d donné en paramètre

* 1. afficherDetailMaladie(id: Long, d: Dictionnaire)

On affiche la maladie dont l’identifiant dans le dictionnaire d est id.

1. **Structuration des données**

Dans notre application nous retrouvons 3 structures de données.

* Le Dictionnaire, pour stocker les maladies, possède une map
* Le FichierPatient, pour stocker ses empreintes, possède une multimap (car il peut y avoir plusieurs fois la même empreinte à analyser
* L’analyse, pour stocker les résultats, possède un tableau de pair <Long, double> pour avoir une paire de maladie avec la probabilité qui lui correspond.

**Plan de tests d'intégration**

Pour chaque service que nous proposons dans notre application nous mettons en place un test d’intégration mis en place dans la classe main.

1. **Renseigner un Dictionnaire**
   1. Lancer la méthode renseignerDictionnaire avec un fichier valide (bonne extension et bonne signature) non vide, au début de l'exécution de l’application (premier dictionnaire renseigner).

→ Les maladies sont chargées et stockées dans la classe Dictionnaire

* 1. Lancer la méthode renseignerDictionnaire avec un fichier valide (bonne extension et bonne signature) non vide, au milieu de l'exécution de l’application.

→ Les maladies sont chargées et stockées dans la classe Dictionnaire en plus des maladie déjà presnetes.

* 1. Lancer la méthode renseignerDictionnaire avec la bonne extension, au milieu de l'exécution de l’application, mais qui ne possède pas la signature demandée

→ Un message d’erreur est affiché

* 1. Lancer la méthode renseignerDictionnaire avec un fichier possédant la mauvaise extension (dans notre cas cela signifie différente de .txt)

→ Un message d’erreur est affiché

* 1. Lancer la méthode renseignerDictionnaire avec un fichier valide (bonne extension et bonne signature) vide.

→ Un message d’erreur est affiché

1. **Renseigner un Fichier Patient**
   1. Lancer la méthode renseignerFichierPatient avec un fichier valide (bonne extension et bonne signature) non vide, au début de l'exécution de l’application (premier fichier renseigner).

→ Un message d’avertissement est afficher pour signaler qu’il n’y a pas encore de dictionnaire.

* 1. Lancer la méthode renseignerFichierPatient avec un fichier valide (bonne extension et bonne signature) non vide, au milieu de l'exécution de l’application.

→ Les empreintes sont chargées et stockées dans une classe FichierPatiente. Les précédentes empreintes sont supprimées.

* 1. Lancer la méthode renseignerFichierPatient avec la bonne extension, au milieu de l'exécution de l’application, mais qui ne possède pas la signature demandée

→ Un message d’erreur est affiché

* 1. Lancer la méthode renseignerFichierPatient avec un fichier possédant la mauvaise extension (dans notre cas cela signifie différente de .txt)

→ Un message d’erreur est affiché

* 1. Lancer la méthode renseignerFichierpPatient avec un fichier valide (bonne extension et bonne signature) vide.

→ Un message d’erreur est affiché

1. **Afficher le Top 10 des maladie**
   1. Lancer la méthode afficherTop10 avant que le Dictionnaire et/ou le FichierPatient ne soit renseigné.

→ Un message d’avertissement est affiché

* 1. Lancer la méthode afficherTop10 avec un FichierPatient qui possède une empreinte et un dictionnaire qui présente plus de 20 maladies →Les 10 maladie les plus probables pour cette empreinte sont affichées. Si des maladie présente une probailité superieur à 70%, un message « Une analyse supplémentaire pour vérification est conseillée. » est rajouté.
  2. Lancer la méthode afficherTop10 avec un FichierPatient qui possède plusieurs empreintes et un dictionnaire qui présente plus de 20 maladies

→Les 10 maladie les plus probables pour chacune des empreintes sont affichées

* 1. Lancer la méthode affciherTop10 avec un FichierPatient qui possède une ou plusieurs empreintes et un dictionnaire qui présente moins de 10 maladies.

→ Les maladies les plus probables (moins de 10) sont affichées pour chacune des empreintes.

* 1. Lancer la méthode afficherTop10 avec un FichierPatient présantant une empreinte et un dictionnaire tel qu’aucune maladie du dictionnaire ne présente de risque pour l’empreinte du patient (probabilité < 20%).

→ Un message d’informations est affichées.

1. **Afficher la meilleure correspondance de maladie**
   1. Lancer la méthode afficherMeilleurCorresp avant que le Dictionnaire et/ou le FichierPatient ne soit renseigné.

→ Un message d’avertissement est affiché

* 1. Lancer la méthode afficherMeilleurCorresp avec un FichierPatient qui possède une empreinte et un dictionnaire qui présente plus de 20 maladies

→Les 10 maladie les plus probables pour cette empreinte sont affichées. Si des maladie présente une probailité superieur à 70%, un message « Une analyse supplémentaire pour vérification est conseillée. » est rajouté.

* 1. Lancer la méthode afficherMeilleurCorrespavec un FichierPatient qui possède plusieurs empreintes et un dictionnaire qui présente plus de 20 maladies

→La maladie la plus probable pour chacune des empreintes est affichée.

* 1. Lancer la méthode afficherMeilleurCorrespavec un FichierPatient présantant une empreinte et un dictionnaire tel qu’aucune maladie du dictionnaire ne présente de risque pour l’empreinte du patient (probabilité < 20%).

→ Un message d’informations est affichées.

1. **Afficher le dictionnaire**
   1. Lancer la méthode afficherDictionnaire avec un dictionnaire possédant des maladies.

→ Toutes les maladies contenue dans le dictionnaire sont affichées

* 1. Lancer la méthode afficherDictionnaire avec un dictionnaire vide

→ Un message d’avertissement prevenant que le dictionnaire est vide, est affiché.

1. **Afficher le détail d’une maladie**
   1. Lancer la méthode afficherDetailMaladie en donnant l’id d’une maladie qui existe dans le dictionnaire qui possède une empreinte.

→ La maladie est affiché (id, nom, empreinte)

* 1. Lancer la méthode afficherDetailMaladie en donnant l’id d’une maladie qui existe dans le dictionnaire qui possède plusieurs empreintes.

→ La maladie est affiché (id, nom, lise d’empreintes)

* 1. Lancer la méthode afficherDetailMaladie en donnant l’id d’une maladie qui n’existe pas dans le dictionnaire.

→ Un message d’avertissement est affiché.

**Plan des Tests Unitaires**

**Planning (ébauche)**