



Rapport De Clôture

Du Projet De Programmation



Gestion de rendez-vous

Seddik

Sakhri

Maskhi

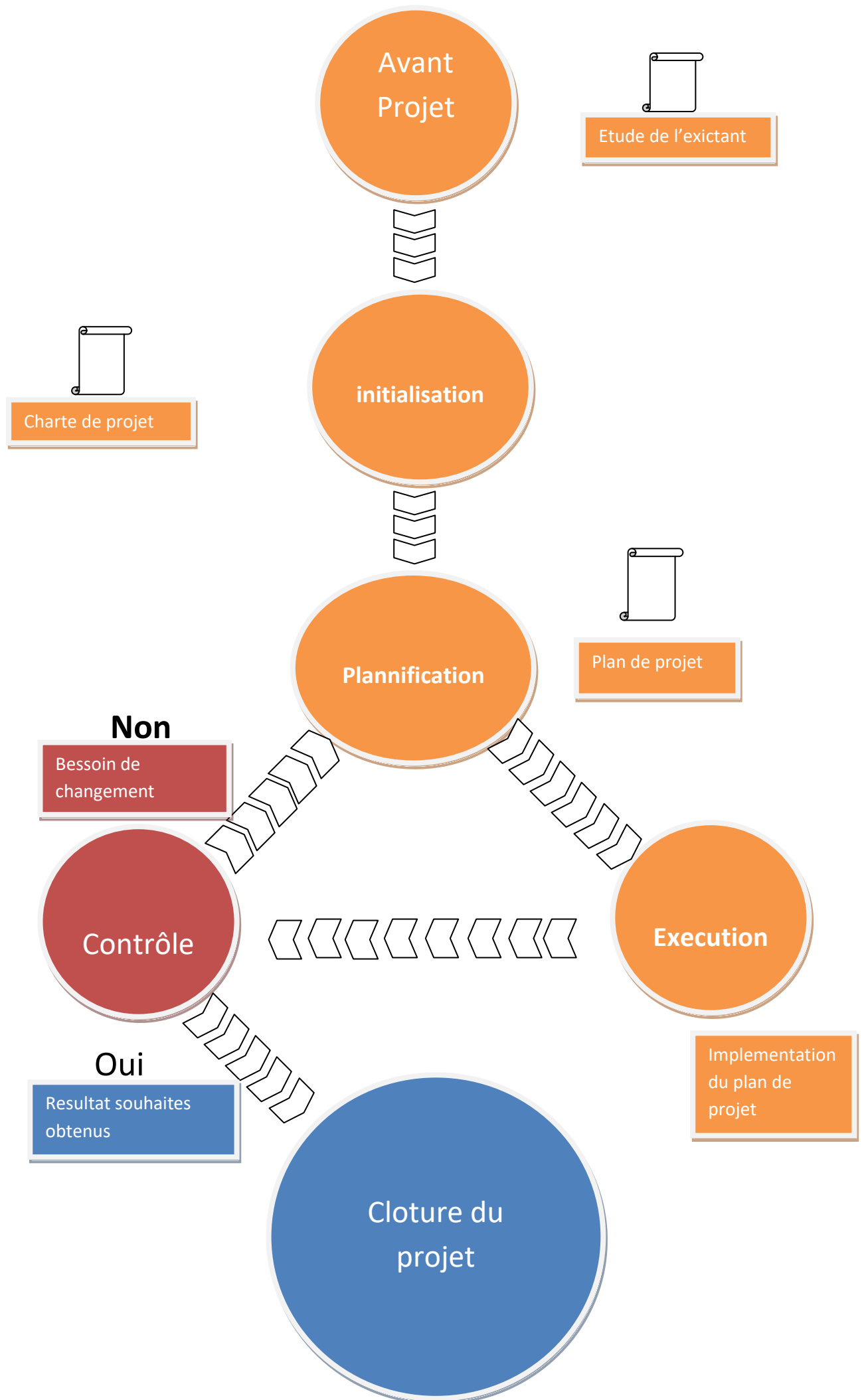
Wael

Wissal

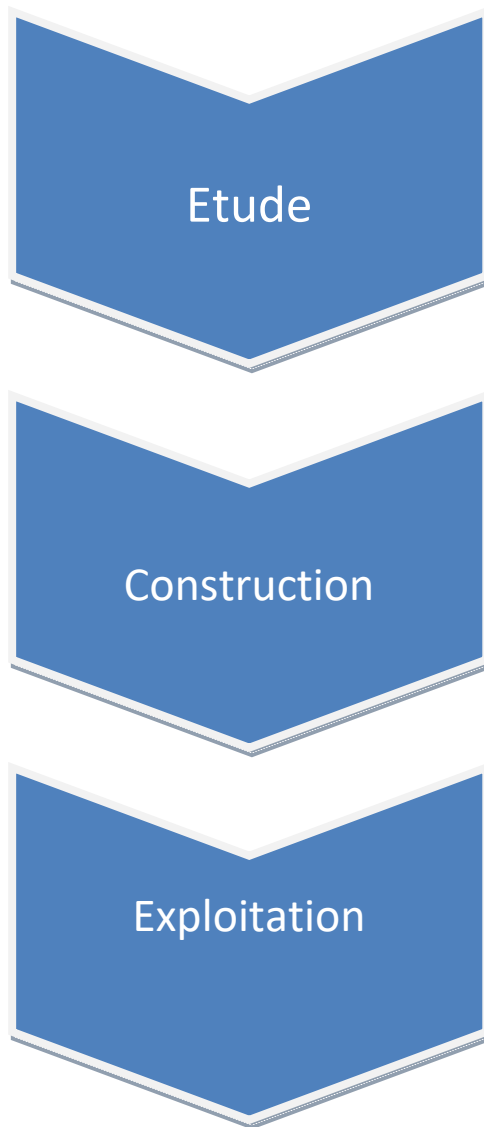
Aymen

li1 A Groupe 5

2020-2021



Plan du cloture

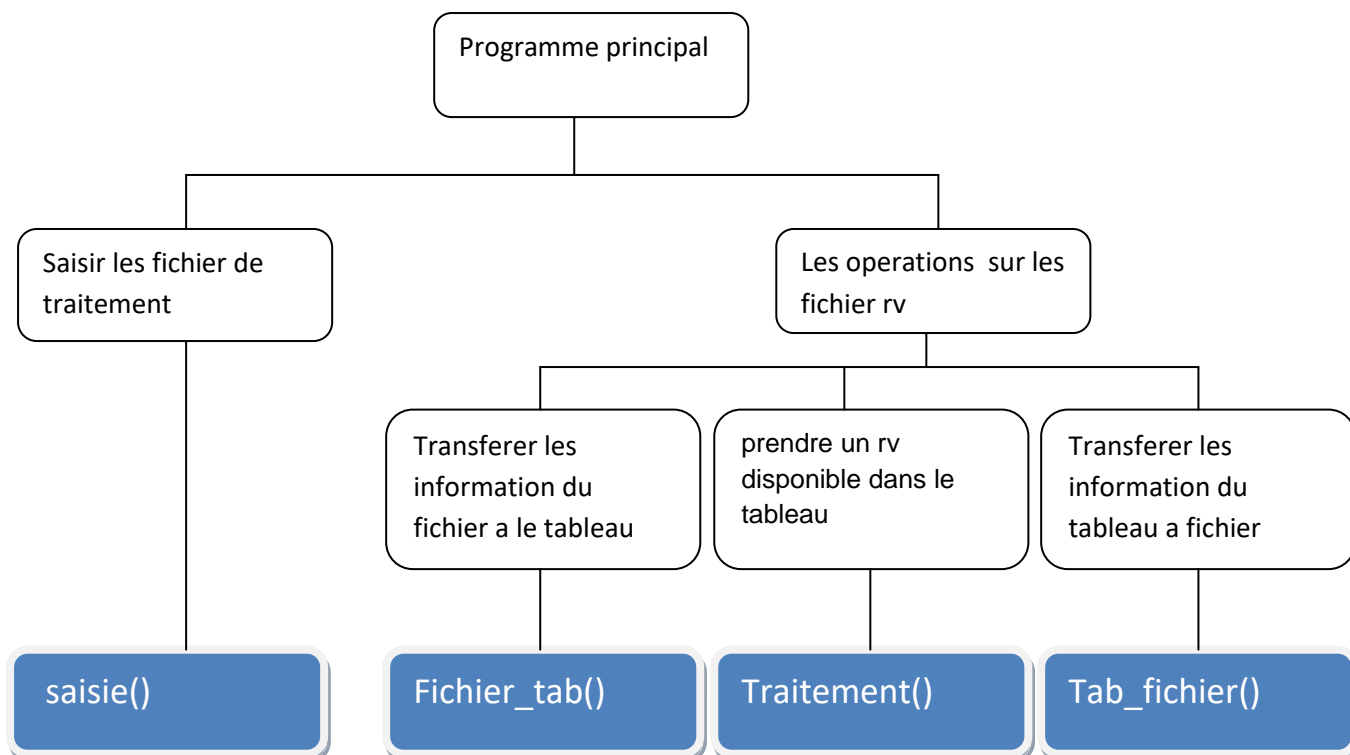
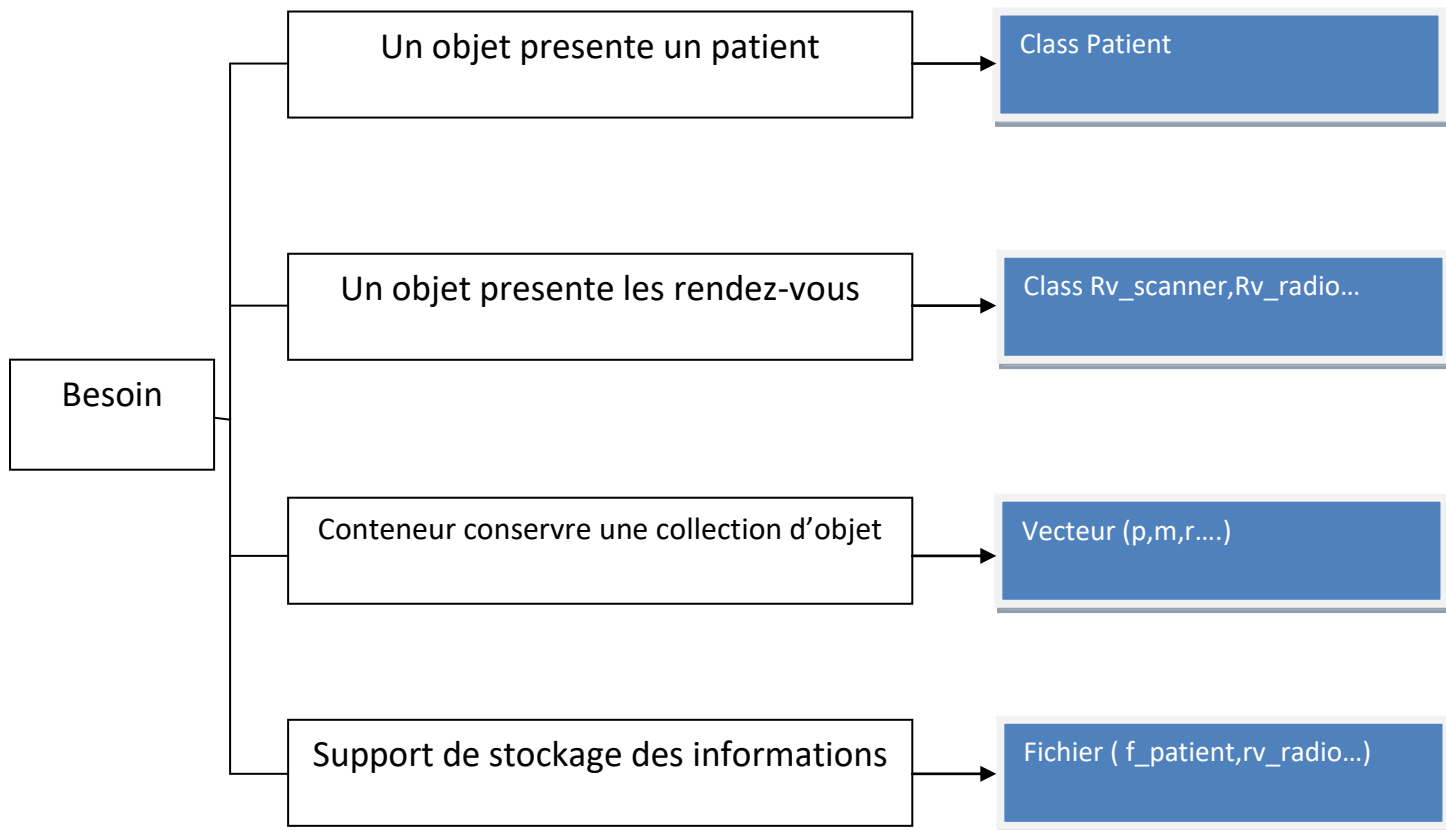


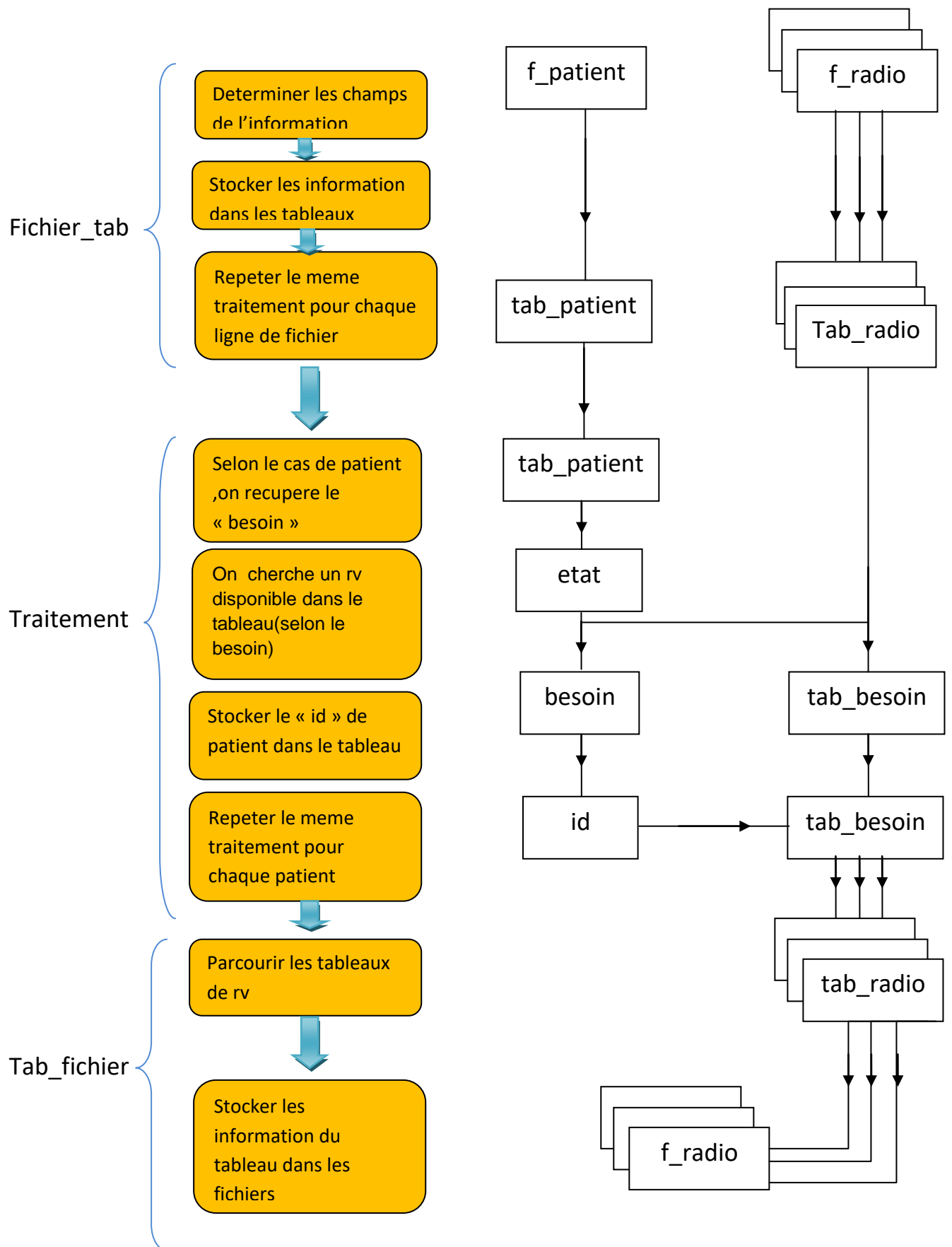
Introduction :

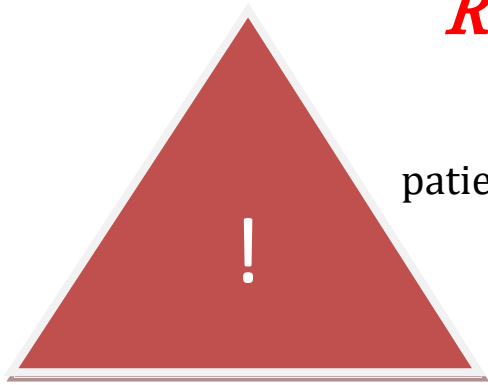
Dans le cadre de notre projet de programmation nous avons eu pour tâche la réalisation d'un projet de gestion de rendez-vous. Notre objectif était de réaliser une démarche bien ordonnée des emplois des rendez-vous pour une bonne organisations.

Après une étude approfondie des situations des cancéreux on se sentit un bain d'affection, si puissant et si fort, et ça nous a encouragé davantage de développer cette idée pour que leur souffrance, causé par la désinvolture des hôpitaux publics, pourrait être partiellement indemnisés.

A- analyse de la situation :







Rq :

→ **id** represente le numero de dossier de patient . chaque patient a sa numero de dossier
→ On a choisie de stocker l id dans les fichiers Pour eviter tout confusion de nom ou de prenom

B-analyse fonctionnelle :

dans cette partie nous allons expliciter certains parties délicates de notre code .

1- Fonction : Fichier_tab()

Cette fonction est une fonction fondamental de la programme principale

objectif : stocker les informations du fichiers dans des tableaux.

But	Idee
L'affectation des valeur aux différentes attributs des instances d'après les fichiers	Une fonction split qui stocke dans un vecteur t les différentes valeurs insérés dans les fichiers et qui sont délimiter par des ":" puis chaque attribut prend sa valeur par ordre
Un usage multiple de vecteur de ce vecteur t du fonction split peut nous causer un désordre	D'où on détruit la totalité de son contenu par la fonction "clear" après chaque usage
On peut stocker à la fois plusieurs instance	Alors on a ajouté un compteur , dès que l'opération s'est terminé une nouvelle se crée

Code :

```
void split(const string& chaine, char delimitateur, std::vector<string>& elements)
{
    stringstream ss(chaine);
    string sousChaine;
    while (getline(ss, sousChaine, delimitateur))
    {
        elements.push_back(sousChaine);
    }
}
```

```
void fichier_tab_p(std::ifstream& f_patient, std::vector<patient>& p)
{
    string ligne ;
    std::vector<string> t ;
    int i(0);
    while(getline(f_patient, ligne))
    {
        patient pp ;
        split(ligne, ':', t);

        pp.id = t[0] ;
        pp.nom = t[1];
        pp.prenom = t[2] ;
        pp.nom_med = t[3] ;
        pp.etat = t[4] ;
        pp.besoin = t[5];
        p.push_back(pp);
        t.clear();
        i+=1 ;
    }
}

void fichier_tab_s(std::ifstream& f_scanner, std::vector<scanner>& s)
{
}

void fichier_tab_r(std::ifstream& f_radio, std::vector<radio>& r)
{
}

void fichier_tab_m(std::ifstream& f_mammo, std::vector<mammo>& m)
{
}

void fichier_tab_a(std::ifstream& f_analyse, std::vector<analyse>& a)
{
}

void fichier_tab_c(std::ifstream& f_controle, std::vector<controle>& c)
{
}
```


2- Fonction : traitement()

C'est le noyau et le plus important partie de ce programme

Objectif : chercher un rv disponible dans le tableau(selon le besoin) pour chaque patient selon leur etat.

But	Idee
Manipulation du tableau patient selon des conditions de priorité bien précis	1 ^{er} parcours en prenant état = 'A' 2eme parcours en prenant état = 'B' 3eme parcours en prenant état = 'C' 4eme parcours en prenant état = 'D'
Remplissage du tableau correspondant	Pour chaque parcours on teste le besoin
Le remplissage doit être significatif	On remplace le premier champs vide du chaque tableau par l ID du patient

Code :

```

void affect_scannerAB(patient& pp ,std::vector<scanner>& ss)
{
    bool aff = false;
    for(int i(0);i<ss.size() && (aff==false) ; i++)
    {
        if (ss[i].rv1=="vide")
        {
            ss[i].rv1=pp.id ;
            aff=true;
        }

        else if (ss[i].rv2=="vide")
        {
            ss[i].rv2=pp.id ;
            aff=true;
        }
        else if (ss[i].rv3=="vide")
        {
            ss[i].rv3=pp.id ;
            aff=true;
        }
        else if (ss[i].rv4=="vide")
        {
            ss[i].rv4=pp.id ;
            aff=true;
        }
        else if (ss[i].rv5=="vide")
        {
            ss[i].rv5=pp.id ;
            aff=true;
        }
        else if (ss[i].rv6=="vide")
        {
            ss[i].rv6=pp.id ;
            aff=true;
        }
        else if (ss[i].rv7=="vide")
        {
            ss[i].rv7=pp.id ;
            aff=true;
        }
        else if (ss[i].rv8=="vide")
        {
            ss[i].rv8=pp.id ;
            aff=true;
        }
        else if (ss[i].rv9=="vide")
        {
            ss[i].rv9=pp.id ;
            aff=true;
        }
        else if (ss[i].rv10=="vide")
        {
            ss[i].rv10=pp.id ;
            aff=true;
        }
    }
}

```

```

void affect_radioAB(patient& pp ,std::vector<radio>& rr)
{

void affect_mammoAB(patient& pp ,std::vector<mammo>& mm)
{

void affect_analyseAB(patient& pp ,std::vector<analyse>& aa)
{

void affect_controleAB(patient& pp ,std::vector<controle>& cc)
{

    void affect_scannerCD(patient& pp ,std::vector<scanner>& ss)
{

    void affect_radioCD(patient& pp ,std::vector<radio>& rr)
{

    void affect_mammoCD(patient& pp ,std::vector<mammo>& mm)
{

    void affect_analyseCD(patient& pp ,std::vector<analyse>& aa)
{

    void affect_controleCD(patient& pp ,std::vector<controle>& cc)
{

```

3-Fonction :Tab_fichier() :

Objectif : Stocker les information du tableau dans les fichiers correspondant.

But	Idee
Gaspillage de mémoire lors de creation un nouveau fichier de stockage apres chaque traitement	On utilise le meme fichier en ecrasant le contenu du fichier par le nouveau contenu avec (of_open) ou lieu (f_open)
Utilisation du caractère de separation defferent	On choisie le « : » pour separer les informations

l'information dans chaque fichier peut etre desordonner	On choisie un ordre bien determiner pour separer les informations pour facilite le travailler de la fonction fichier_tab()
---	--

On choisie un ordre bien
determiner pour separer les
informations pour facilite le
travailler de la fonction
fichier_tab()

Code :

```
void tab_fichier_s(std::vector<scanner>& s, std::ofstream& of_scanner)
{
    for (int i(0); i<s.size(); i++)
    {
        of_scanner << s[i].date << ":"<<s[i].rv1 << ":"<<s[i].rv2 << ":"<<s[i].rv3 << ":"<<s[i].rv4 << ":"<<s[i].rv5 << ":"<<s[i].
    }
}

void tab_fichier_r(std::vector<radio>& r, std::ofstream& of_radio)
{
}

void tab_fichier_m(std::vector<mammo>& m, std::ofstream& of_mammo)
{
}

void tab_fichier_a(std::vector<analyse>& a, std::ofstream& of_analyse)
{
}

void tab_fichier_c(std::vector<controle>& c, std::ofstream& of_controle)
{
}
```

4-fonction : saisie()

Objectif : saisir les fichiers de traitement

Code :

```
void saisie_nf(std::string& nf_patient, std::string& nf_scanner, std::string& nf_radio, std::string& nf_mammographie)
{
    cout << "Saisir votre fichier patient sous la forme: C:/fichier/nom_fichierpatient.txt ";
    cin >> nf_patient;
    cout << "Saisir votre fichier scanner sous la forme: C:/fichier/nom_fichier_scanner.txt ";
    cin >> nf_scanner;
    cout << "Saisir votre fichier radio sous la forme: C:/fichier/nom_fichier_radio.txt ";
    cin >> nf_radio;
    cout << "Saisir votre fichier mammo sous la forme: C:/fichier/nom_fichier_mammo.txt ";
    cin >> nf_mammographie;
    cout << "Saisir votre fichier analyse sous la forme: C:/fichier/nom_fichier_analyse.txt ";
    cin >> nf_analyse;
    cout << "Saisir votre fichier controle sous la forme: C:/fichier/nom_fichier_controle.txt ";
    cin >> nf_controle;
}
```

5-Programme principale :

declaration des variables :

```

string nf_patient,nf_scanner,nf_radio,nf_mammo,nf_analyse,nf_controle ; // declaration de
ifstream f_patient,f_scanner,f_radio,f_mammo,f_analyse,f_controle ; // declaration des st
ofstream of_patient,of_scanner,of_radio,of_mammo,of_analyse,of_controle ; // declaration

vector<patient> p ;
vector<scanner> s ;
vector<radio> r ; // declaration des tableaux dynamiques de type de nos classes
vector<mammo> m ;
vector<analyse> a ;
vector<controle> c ;

```

Appel du fonction saisie ()

```
saisie_nf(nf_patient,nf_scanner,nf_radio,nf_mammo,nf_analyse,nf_controle); /
```

Ouvrir les fichier nécessaire en mode lecture

```

f_patient.open(nf_patient.c_str());
f_scanner.open(nf_scanner.c_str());
f_radio.open(nf_radio.c_str());
f_mammo.open(nf_radio.c_str());
f_analyse.open(nf_analyse.c_str());
f_controle.open(nf_controle.c_str());

```

Appel du fonction fichier_tab() :

```

fichier_tab_p(f_patient,p);
fichier_tab_s(f_scanner,s);
fichier_tab_r(f_radio,r);
fichier_tab_m(f_mammo,m); /
fichier_tab_a(f_analyse,a);
fichier_tab_c(f_controle,c);

```

Appel du fonction traitement () :

```
traitement(p,s,r,m,a,c) ;
```

fermeture des fichier en mode lecture et les ouvrir en mode ecriture

```

f_patient.close();
f_scanner.close();
f_radio.close();
f_mammo.close();           // fermeture
f_analyse.close();
f_controle.close();

of_scanner.open(nf_scanner.c_str());
of_radio.open(nf_radio.c_str());
of_mammo.open(nf_mammo.c_str());
of_analyse.open(nf_analyse.c_str());
of_controle.open(nf_controle.c_str());

```

Appel de la fonction tab_fichier() :

```

tab_fichier_s(s, of_scanner);
tab_fichier_r(r, of_radio);
tab_fichier_m(m, of_mammo);  /
tab_fichier_a(a, of_analyse);
tab_fichier_c(c, of_controle);

```

fermeture des fichiers en mode écriture


```

of_scanner.close();
of_radio.close();
of_mammo.close();
of_analyse.close();
of_controle.close();

```

C- Exploitation :

Nous allons commencer par constater le contenu de fichier scanner (ou radio ou...) avant de faire les traitements

 scanner - Bloc-notes

Fichier Edition Format Affichage Aide


```
2/05/2021:vide:vide:vide:vide:vide:vide:vide:vide:vide:vide
3/05/2021:vide:vide:vide:vide:vide:vide:vide:vide:vide:vide
4/05/2021:vide:vide:vide:vide:vide:vide:vide:vide:vide:vide
5/05/2021:vide:vide:vide:vide:vide:vide:vide:vide:vide:vide
8/05/2021:vide:vide:vide:vide:vide:vide:vide:vide:vide:vide
9/05/2021:vide:vide:vide:vide:vide:vide:vide:vide:vide:vide
10/05/2021:vide:vide:vide:vide:vide:vide:vide:vide:vide:vide
11/05/2021:vide:vide:vide:vide:vide:vide:vide:vide:vide:vide
12/05/2021:vide:vide:vide:vide:vide:vide:vide:vide:vide:vide
15/05/2021:vide:vide:vide:vide:vide:vide:vide:vide:vide:vide
16/05/2021:vide:vide:vide:vide:vide:vide:vide:vide:vide:vide
17/05/2021:vide:vide:vide:vide:vide:vide:vide:vide:vide:vide
18/05/2021:vide:vide:vide:vide:vide:vide:vide:vide:vide:vide
19/05/2021:vide:vide:vide:vide:vide:vide:vide:vide:vide:vide
20/05/2021:vide:vide:vide:vide:vide:vide:vide:vide:vide:vide
22/05/2021:vide:vide:vide:vide:vide:vide:vide:vide:vide:vide
23/05/2021:vide:vide:vide:vide:vide:vide:vide:vide:vide:vide
24/05/2021:vide:vide:vide:vide:vide:vide:vide:vide:vide:vide
25/05/2021:vide:vide:vide:vide:vide:vide:vide:vide:vide:vide
26/05/2021:vide:vide:vide:vide:vide:vide:vide:vide:vide:vide
26/05/2021:vide:vide:vide:vide:vide:vide:vide:vide:vide:vide
```

On saisie les informations de patient pour ajouter un nouveau objet de type patient et le stocker dans le fichier f_patient (saisie direct dans le fichier).

On prend l'exemple suivant :

Id=0 ;nom=wael ;prenom=seddik ;doctor =mohamed ;etat=A ;besoin=scanner

Après la saisie de l'information de patient , voici le contenu du fichier f_patient

 patient - Bloc-notes

Fichier Edition Format Affichage Aide


```
1:wael:seddik:mohamed:A:scanner
```

On va lancer maintenant notre programme

On commence par mentionner les fichiers nécessaire pour stocker les informations comme suit :

```
Saisir votre fichier patient sous la forme: C:/fichier/nom_fichierpatient.txt f_patient
Saisir votre fichier patient sous la forme: C:/fichier/nom_fichier_scanner.txt f_scanner
Saisir votre fichier patient sous la forme: C:/fichier/nom_fichier_radio.txt f_radio
Saisir votre fichier patient sous la forme: C:/fichier/nom_fichier_mammo.txt f_mammo
Saisir votre fichier patient sous la forme: C:/fichier/nom_fichier_analyse.txt f_analyse
Saisir votre fichier patient sous la forme: C:/fichier/nom_fichier_controle.txt f_controle
```

Après la saisie des noms de fichiers, le contenu de fichier scanner(ou radio...) sera écrasé de manière suivante :

 *scanner - Bloc-notes

Fichier Edition Format Affichage Aide

```
2/05/2021:1:vide:vide:vide:vide:vide:vide:vide:vide:vide
3/05/2021:vide:vide:vide:vide:vide:vide:vide:vide:vide:vide
4/05/2021:vide:vide:vide:vide:vide:vide:vide:vide:vide:vide
5/05/2021:vide:vide:vide:vide:vide:vide:vide:vide:vide:vide
8/05/2021:vide:vide:vide:vide:vide:vide:vide:vide:vide:vide
```

On peut prendre un exemple plus complexe comme la suite

```
1:wael:seddik:mohamed:A:scanner
2:aymen:maskhi:saber:B:mammo
3:wissal:sakhri:adel:D:radio
4:ala:tlili:mohamed:A:controle
5:sami:gafsi:taher:C:analyse
6:hsan:masoudi:fedi:B:mammo
7:mahdi:wedhrfi:moez:B:scanner
8:saber:dhaouedi:moez:A:scanner
9:zouhair:khelifa:mohamed:D:radio
10:bilel:khfifi:mohamed:C:controle
11:hamza:abdi:adel:B:analyse
12:ali:agrebi:mohamed:D:scanner
13:iskander:benamor:mohamed:A:mammo
```


2/05/2021:1:8:31:35:42:65:7:16:34:41

3/05/2021:50:68:68:vide:vide:vide:vide:vide:vide:vide

4/05/2021:vide:vide:vide:vide:vide:vide:vide:vide:vide

5/05/2021:vide:vide:vide:vide:vide:vide:vide:vide:vide

8/05/2021:vide:vide:vide:vide:vide:vide:vide:vide:vide

9/05/2021:vide:vide:vide:vide:vide:vide:vide:vide:vide

10/05/2021:vide:vide:vide:vide:vide:vide:vide:vide:vide

11/05/2021:vide:vide:vide:vide:vide:vide:vide:vide:vide

Conclusion :

L'aide de ce projet nous avons pu comprendre et expérimenter les différentes étapes de la conception d'un programme de gestionnaire des fichiers. En plus d'être un projet pédagogique il nous aide d'améliorer nos compétences de travail en équipe et de gestion de temps .

Finalement nous avons atteint notre objectif désiré de révolutionner le secteur sanitaire et surtout la nécessité de prêter attention aux patients atteints de cancer à travers cette digitalisation innovante.

Remerciement :

Nos remerciements s'adressent également à notre chère Madame **Yemna Sayeb** , tes conseils tout au long du projet nous ont été d'un apport précieux. Trouves ici l'expression de nos sincère reconnaissance.