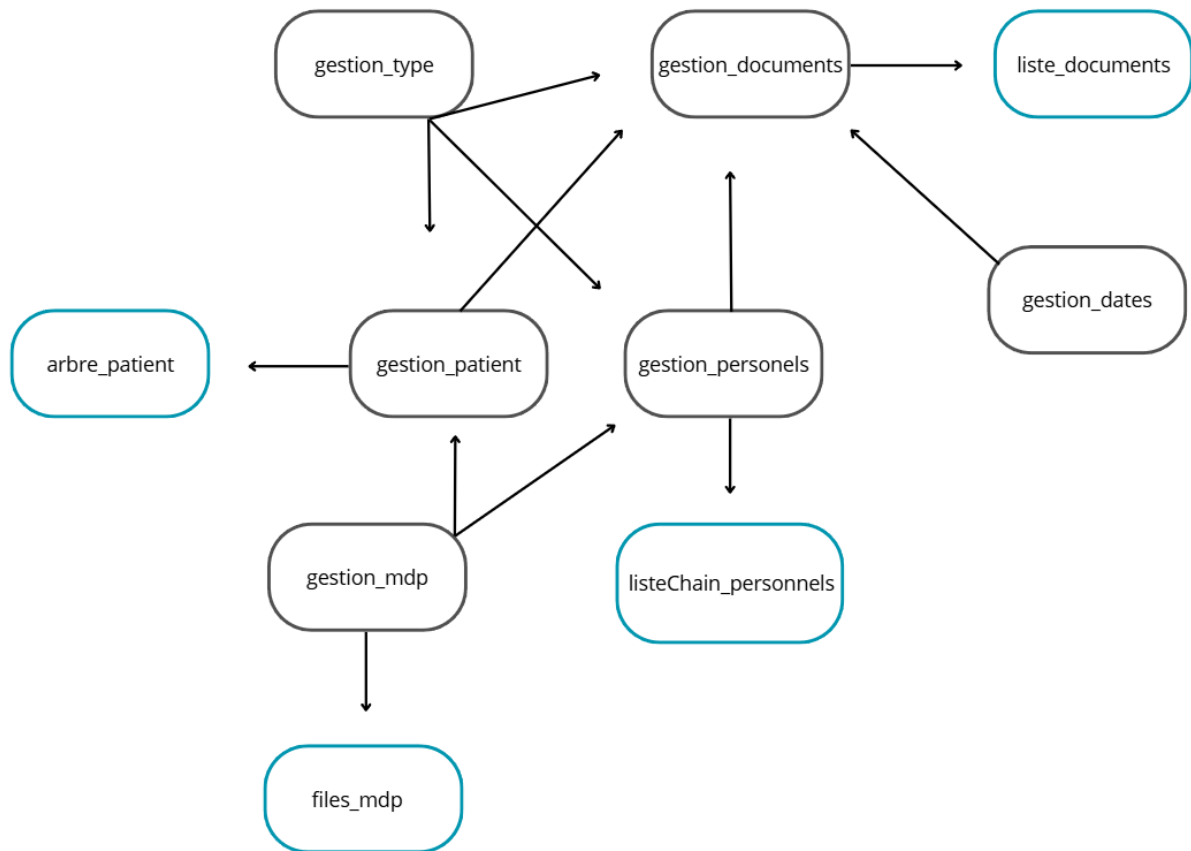


Structure de Donnée

- l'architecture envisagée pour le projet :



- Structure de données:

Package outils_Type

package outils is

-- Types pour les noms, prénoms, etc.

subtype T_mots is String(1..50);

Type T_identite is record

prenom,nom:T_mots (others=>' ');

k_nom ,k_prenom: natural:=0;

subtype T_Motlong String(1..50);

end record;

-- Types pour les erreurs

type Erreur is (login indispo, Mot_De_Passe_Incorrect, , Patient_Non_Trouve, Personnel_Non_Trouve);

end outils_Types;

Package Gestion_Dates

```
package gestion_dates is
  subtype T_int_mois is integer range 1..12;
  subtype T_int_jour is integer range 1..31;
  type t_date is record
    jour: T_int_jour;
    mois: T_int_mois;
    annee: natural;
  end t_date;
end gestion_dates;
```



Package Gestion_Personnel

```
with gestion_type;
use gestion_type;
```

```
package Gestion_personnel is
  type T_fonction is (médecin, secrétaire, administrateur)
```

```
  type T_personnel is record
    identité: T_idendite;
    nom_jfm_pers: T_mots;
    login_pers: T_Motlong;
    k_nom_jfm, k_login: natural:=0;
    empreint_pers: natural ;
    fonction: T_fonction;
  end record;
```



```
end gestion_personnel;
```

Package ListeChain_personnel

```
with gestion_personnel;
use gestion_personnel;
```

```
type T_cellule;
type T_pt_pers is access T_cellule
type T_cellule is record
  pers: T_personnels
  suivpers: T_pt_pers;
end record;
```



Package Gestion_Patient

with Types;

use Types;

package Patients is

```
type T_Patient is record
  pers: T_identite;
  Nom_Jeune_Fille_Mere :T_mots;
  k_nom_jfm: natural:=0;
  Login : T_Motlong;
  Empreinte_Mdp : Integer;
  Documents : T_pt_Doc
end record;
```



Package Arbre_Patient

with Gestion_Type;

use Gestion_Type;

```
type T_Noed_Patient ;
type T_arbre_Patients is access T_Noed_Patient;
type T_Noed_Patient is record
  cle : T_Patient;
  fg,fd : T_arbre_Patients;
end record;
end Arbre_Patients;
```



```
-- Opérations sur les patients
procedure Ajouter_Patient(Nom, Prenom, Nom_Jeune_Fille_Mere : in String;
Empreinte_Mdp : in Integer);
function Rechercher_Patient(Login : in Login_Type) return Patient_Record;
procedure Afficher_Patients;
```

Package Gestion_document

with Types;

use Types;

package Documents is

```
type T_Nat_Document is (Ordonnance, Certificat, Compte_Rendu, Resultats_Examens);
```

```
type T_Document is record
  Id : Integer;
  Titre_doc: T_Motlong:= (others=>' ');
  Nature : T_Nat_Document;
  Med : T_pt_pers;
  Lecture : Boolean ;
```

```
Date_Creation : T_date;  
Date_Derniere_Lect : T_date;  
Date_Derniere_Modif : T_date;  
end record;
```



Package Liste_document

```
with gestion_document;  
use gestion_document;
```

```
Type T_ListDoc  
Type T_pt is access T_ListDoc  
Type T_ListDoc is record  
Doc : T_Document;  
suivDoc: T_pt_Doc;  
end record;
```



```
end Liste_Documents;
```

-- Opérations sur les documents

```
Procedure Ajouter_Document(Titre : in String; Nature : in Nature_Document; Medecin : in  
Personnel.Personnel_Record; Patient : in Patients.Patient_Record);  
function Rechercher_Document(Identifiant : in Integer) return Document_Record;  
procedure Afficher_Documents;
```

Package Gestion_Mdp

```
with Types;  
use Types;
```

```
package Gestion_Mdp is
```

```
type Demande_Mdp_Oublie is record  
Login : T_Motlong;  
Nom_Jfm :T_mots;  
k_nom_jfm: natural:=0;  
Est_Patient : Boolean;  
end record;
```



package File_Mdp

```
with  
use  
type T_cellMdp
```

```

type T_pt is access T_cellMdp
type T_cellMdp is record
Mdp: Demande : Demande Mdp Oublie;
suiv:T_pT;
end record;
    type File_Mdp is record
tete ,fin :T_pt;
end record;

-- Opérations sur les demandes
procédure Ajouter_Demande(Login : in String; Nom_Jeune_Fille_Mere : in String;
Est_Patient : in Boolean);
function Traiter_Demande return Demande_Mdp_Oubli;
procedure Afficher_Demandes;

end Demandes_Mdp;

```

Package Gestion_Archive

```

with Types;
use Types;
type T_Document_Archive is record
    Id : Integer;
    Titre_doc: T_Motlong:= (others=>' ');
    Nature : T_Nat_Document;
    Med :T_identite;
    pat:T_identite;
    Lecture : Boolean ;
    Date_Creation : T_date;
    Date_Derniere_Lect : T_date;
    Date_Derniere_Modif : T_date;
end record;

package Gestion Archives is new .Ada_sequential_io(T_document_Archive);

-- Opérations sur les archives
procédure Archiver Document(Document : in Documents.Document Record);
procédure Afficher Archives;

end Archives;

```