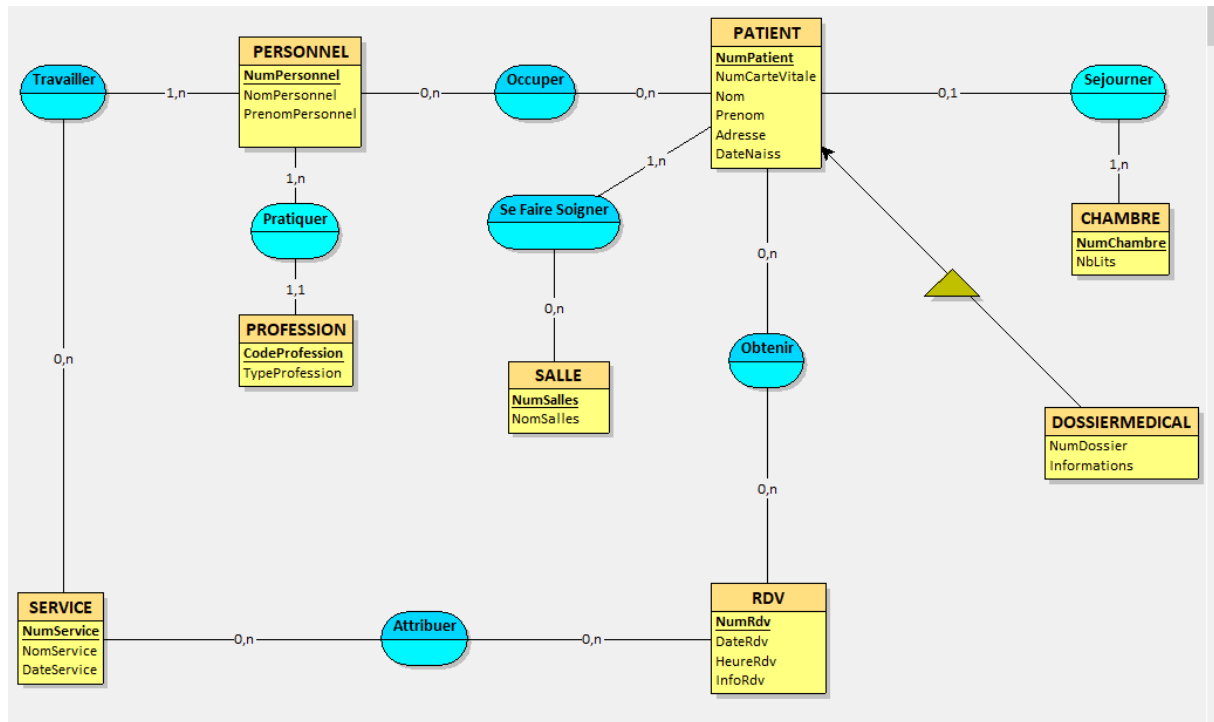


Document technique

1.mcd du cas de l'hôpital

Dans un premier temps nous avons effectuer le mcd afin d'avoir un premier point de vu su sujet

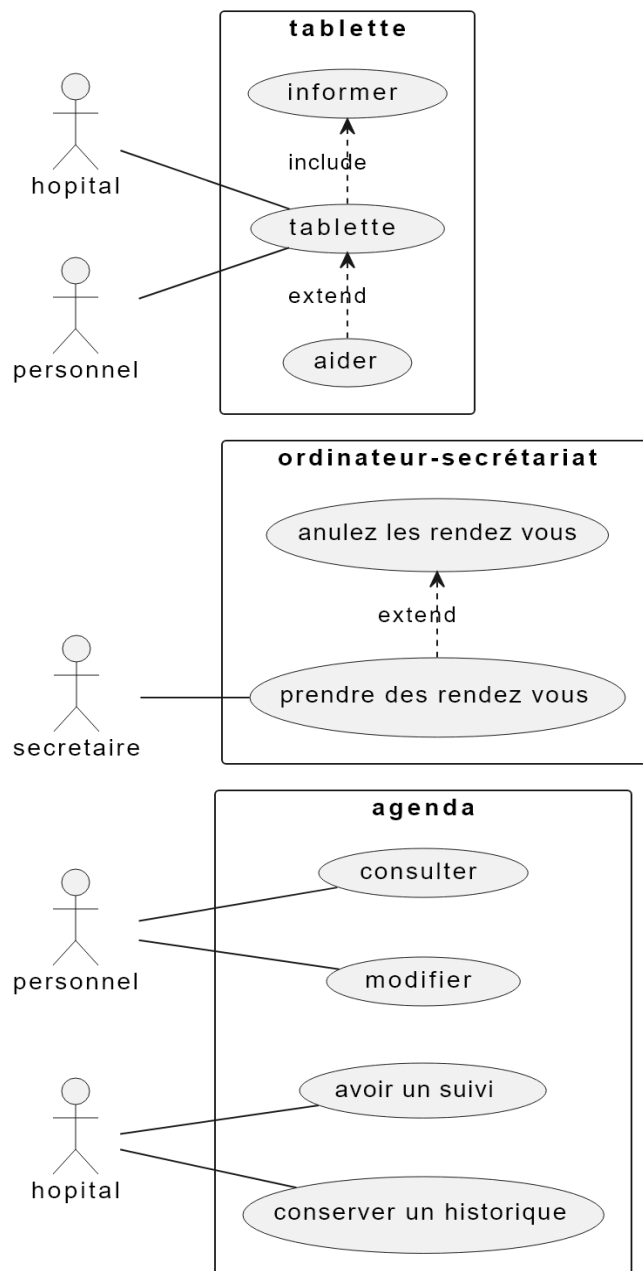


Le mcd nous montre les différentes tables que l'on aura dans notre base de données ainsi que les objets qu'ils contiennent

Nous avons pu voir un héritage de la classe patient sur la classe dossiermedicale

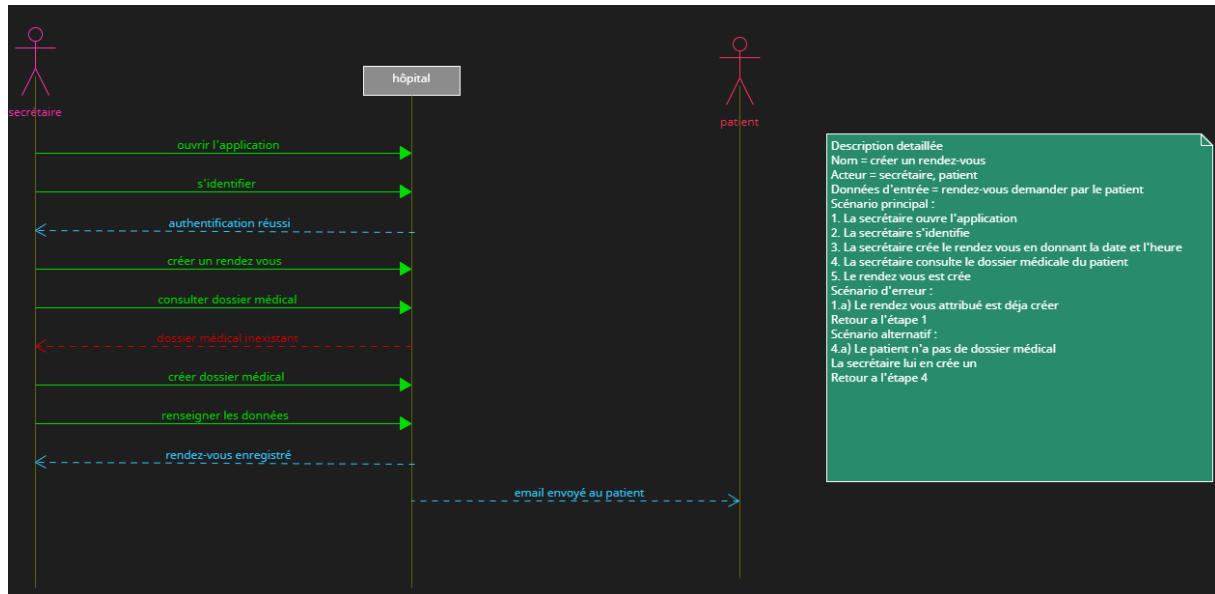
2. Cas d'utilisation

Après le mcd nous avons pu effectuer un cas d'utilisation qui nous montré les besoins de chaque personne de l'hôpital



3. diagrammes de séquences

Dans un dernier de temps de l'analyse nous avons créé un diagramme de séquence dans lequel nous avons montré un point dans lequel un interlocuteur crée un événement jusqu'à en devenir un organisateur



4. Début de création du site

Nous avons décidé de commencer à effectuer l'index du site dans lequel nous avons mis tout ce que les médecins auront besoin .



5. création de notre base de données

Grace au mcd que vous pouvez voir sur notre première page nous avons pu crée notre base de données sur SQL server.

```
CREATE TABLE PERSONNEL(  
    NumPersonnel INT IDENTITY,  
    NomPersonnel VARCHAR(50),  
    PrenomPersonnel VARCHAR(50),  
    PRIMARY KEY(NumPersonnel)  
);  
CREATE TABLE RDV(  
    NumRdv INT IDENTITY,  
    DateRdv DATE NOT NULL,  
    HeureRdv TIME,  
    InfoRdv VARCHAR(500),  
    PRIMARY KEY(NumRdv)  
);  
CREATE TABLE SERVICE(  
    NumService INT IDENTITY,  
    NomService VARCHAR(20) NOT NULL,  
    DateService DATETIME2,  
    PRIMARY KEY(NumService)  
);  
CREATE TABLE CHAMBRE(  
    NumChambre INT IDENTITY,  
    NbLits SMALLINT,  
    PRIMARY KEY(NumChambre)  
);  
CREATE TABLE SALLE(  
    NumSalles INT IDENTITY,  
    NomSalles VARCHAR(50) NOT NULL,  
    PRIMARY KEY(NumSalles)  
);  
CREATE TABLE PROFESSION(  
    CodeProfession INT IDENTITY,  
    TypeProfession VARCHAR(50),  
    NumPersonnel INT NOT NULL,  
    PRIMARY KEY(CodeProfession),  
    FOREIGN KEY(NumPersonnel) REFERENCES PERSONNEL(NumPersonnel)  
);  
CREATE TABLE PATIENT(  
    NumPatient INT IDENTITY,  
    NumCarteVitale INT,  
    Nom VARCHAR(50) NOT NULL,  
    Prenom VARCHAR(50),  
    Adresse VARCHAR(100),  
    DateNaiss DATE,  
    NumChambre INT,
```

```

    PRIMARY KEY(NumPatient),
    FOREIGN KEY(NumChambre) REFERENCES CHAMBRE(NumChambre)
);
CREATE TABLE DOSSIERMEDICAL(
    NumPatient INT,
    NumDossier SMALLINT NOT NULL,
    Informations VARCHAR(500),
    PRIMARY KEY(NumPatient),
    FOREIGN KEY(NumPatient) REFERENCES PATIENT(NumPatient)
);
CREATE TABLE Occuper(
    NumPersonnel INT,
    NumPatient INT,
    PRIMARY KEY(NumPersonnel, NumPatient),
    FOREIGN KEY(NumPersonnel) REFERENCES PERSONNEL(NumPersonnel),
    FOREIGN KEY(NumPatient) REFERENCES PATIENT(NumPatient)
);
CREATE TABLE Se_Faire_Soigner(
    NumPatient INT,
    NumSalles INT,
    PRIMARY KEY(NumPatient, NumSalles),
    FOREIGN KEY(NumPatient) REFERENCES PATIENT(NumPatient),
    FOREIGN KEY(NumSalles) REFERENCES SALLE(NumSalles)
);
CREATE TABLE Travailler(
    NumPersonnel INT,
    NumService INT,
    PRIMARY KEY(NumPersonnel, NumService),
    FOREIGN KEY(NumPersonnel) REFERENCES PERSONNEL(NumPersonnel),
    FOREIGN KEY(NumService) REFERENCES SERVICE(NumService)
);
CREATE TABLE Obtenir(
    NumPatient INT,
    NumRdv INT,
    PRIMARY KEY(NumPatient, NumRdv),
    FOREIGN KEY(NumPatient) REFERENCES PATIENT(NumPatient),
    FOREIGN KEY(NumRdv) REFERENCES RDV(NumRdv)
);
CREATE TABLE Attribuer(
    NumRdv INT,
    NumService INT,
    PRIMARY KEY(NumRdv, NumService),
    FOREIGN KEY(NumRdv) REFERENCES RDV(NumRdv),
    FOREIGN KEY(NumService) REFERENCES SERVICE(NumService)
);

```

Dans la base de données nous pouvons voir toutes les tables émis dans le mcd ainsi que leurs attributs

6. configuration de notre serveur

```
ktexier@SLAM2-IP-199:~$ ssh 172.16.199.254
[BTSSIO]SLAM2
Linux debiansio 5.10.0-18-amd64 #1 SMP Debian 5.10.140-1 (2022-09-02) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Oct 18 11:47:18 2022 from 172.16.199.3
root@debiansio:~#
```

Nous avons pu configurer notre serveur afin que nous puissions avoir accès à notre ssh ainsi qu'à notre dépôt git et donc de push ,commit etc...

7. Choisir un traitement et développer avec **Java Android**

Nous avons commencé la mission dans laquelle nous devons choisir un traitement et le développer avec java

```
public class Patient{
    private int NumPatient;
    private String Nom;
    private String Prenom;
    private String Adresse;
    private String DateNaiss;
    private String NumCarteVitale;

    public int getNumPatient()
    {
        return NumPatient;
    }

    public String getNom()
    {
        return Nom;
    }
    public String getPrenom()
    {
        return Prenom;
    }
    public String getAdresse()
    {
        return Adresse;
    }
    public String getDateNaiss()
    {
        return DateNaiss;
    }
    public String getNumCarteVitale()
    {
        return NumCarteVitale;
    }
    public void setNumPatient(int NumPatient){
        this.NumPatient=NumPatient;
    }
    public void setNom(String Nom){
        this.Nom=Nom;
    }
    public void setPrenom(String Prenom){
        this.Prenom=Prenom;
    }
    public void setAdresse(String Adresse){
        this.Adresse=Adresse;
    }
}
```

```

    }
    public void setDateNaiss(String DateNaiss){
        this.DateNaiss=DateNaiss;
    }
    public void setNumCarteVitale(String NumCarteVitale){
        this.NumCarteVitale=NumCarteVitale;
    }

    /**
     *
     * @param NumPatient
     * @param Nom
     * @param Prenom
     * @param Adresse
     * @param DateNaiss
     * @param NumCarteVitale
     */
    public Patient(int NumPatient,String Nom,String Prenom,String
Adresse,String DateNaiss, String NumCarteVitale){
        this.NumPatient=NumPatient;
        this.Nom=Nom;
        this.Prenom=Prenom;
        this.Adresse=Adresse;
        this.DateNaiss=DateNaiss;
        this.NumCarteVitale=NumCarteVitale;
    }
    @Override
    public String toString()
    {
        return "Patient: NumPatient = " + this.NumPatient + " Nom = " +
this.Nom + " Prenom = " + this.Prenom + " Adresse = " + this.Adresse + "
DateNaiss = " + this.DateNaiss + " NumCarteVitale = " + this.NumCarteVitale;
    }
}

```

Ci-dessus nous avons la classe patient dans lequel nous lui avons donner ses attributs comme donnée dans la table patient du mcd , la classe est constituer de getter et de setter ainsi que son constructeur et de sa méthode toString()

Et ensuite nous avons son Test pour vérifier si toute la classe marcher bien

```

public class TestPatient{
    public static void main(String[] args) {
        Patient patient=new Patient(1,"monkey d","luffy","2rue du
sunny","17/07/2003","1111224451");
        System.out.println(patient);
    }
}

```