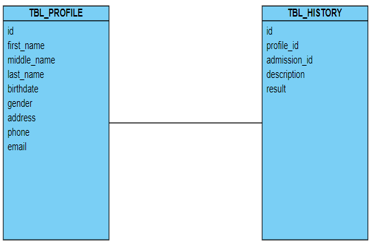
**Micro Service Patient :**

1. **Description Générale :**

Ce micro-service a comme but de gérer les patients de notre hôpital Covid19. On peut inscrire une patient et voir l’historique de chaque patient déjà subir une action dans notre hôpital par exemple « un test de covid19, réservation chambre , opération … »

Ce micro service fonctionne d’une façon indépendante des autres micro services en communiquant avec sa propre base de donnée.

1. **Diagrammes :**
   1. **Diagramme de classe :**



**Description des attributs :**

* **TBL\_PROFILE :**
* Id :

Type : Integer | Description : identifier par incrémentation pour un nouveau patient.

* First\_name :

Type : String | Description : le prénom du patient.

* Middle\_name :

Type : String | Description : le deuxième prénom du patient.

* Last\_name :

Type : String | Description : le nom du patient.

* Birthdate :

Type : Date | Description : la date de naissance du patient.

* Gender :

Type : Char | Description : ‘M’pour Monsieur, ‘F’ pour Madame.

* Address :

Type : String | Description : la description du lieu où vive le patient.

* Phone :

Type : String | Description : le numéro de téléphone du patient.

* Email :

Type : String | Description : Email adresse du patient.

* **TBL\_HISTORY :**
* Id :

Type : Integer | Description : identifier par incrémentation pour un historique pour un patient.

* Profile\_id :

Type : Integer | Description : identifier d’un profile.

* Admission\_id :

Type : Integer| Description : identifier d’une admission.

* Description :

Type : String | Description : une description détaillée à propos la visite du patient.

* Result :

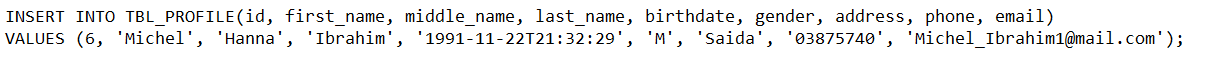
Type : String | Description : le résultat final du service.

**Exemple des données :**

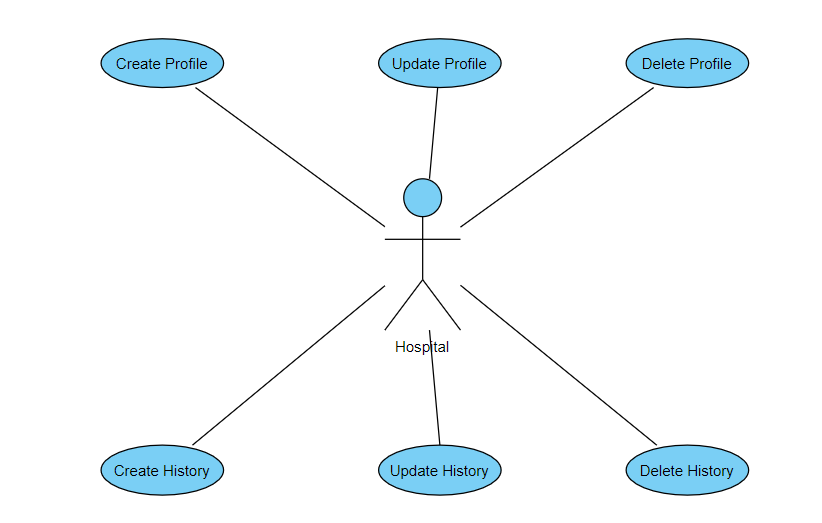
* TBL\_PROFILE :



* TBL\_HISTORY :

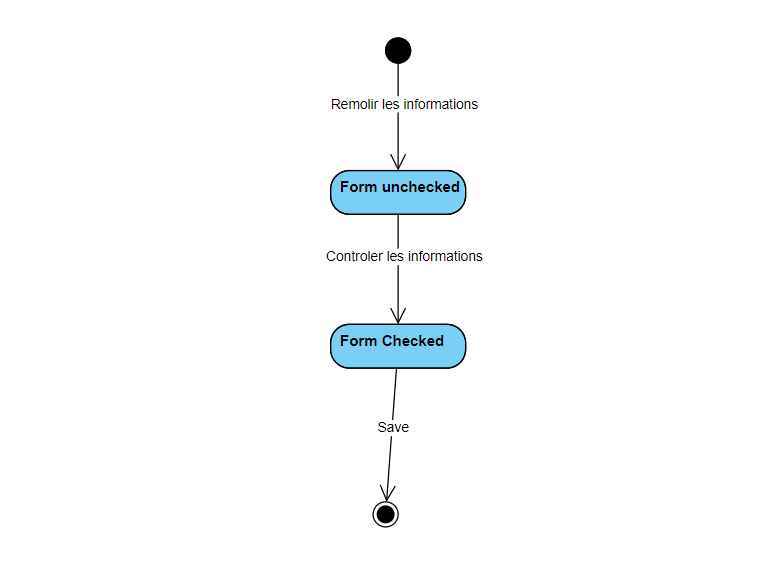


* 1. **Cas d’utilisation :**



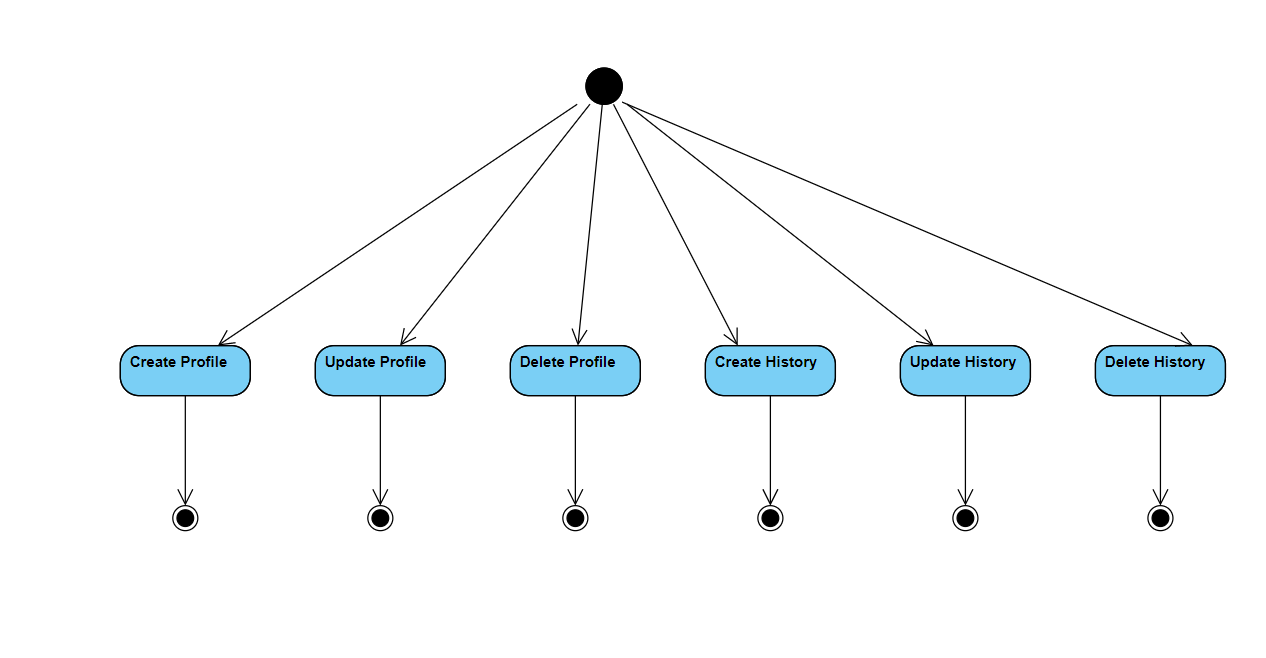
L’employée en prenant les informations du patient est capable de faire les Crud SQL opérations (Create / Update / Delete) ou bien pour un profile ou bien pour un history en introduisant les informations nécessaires.

* 1. **État-transition :**



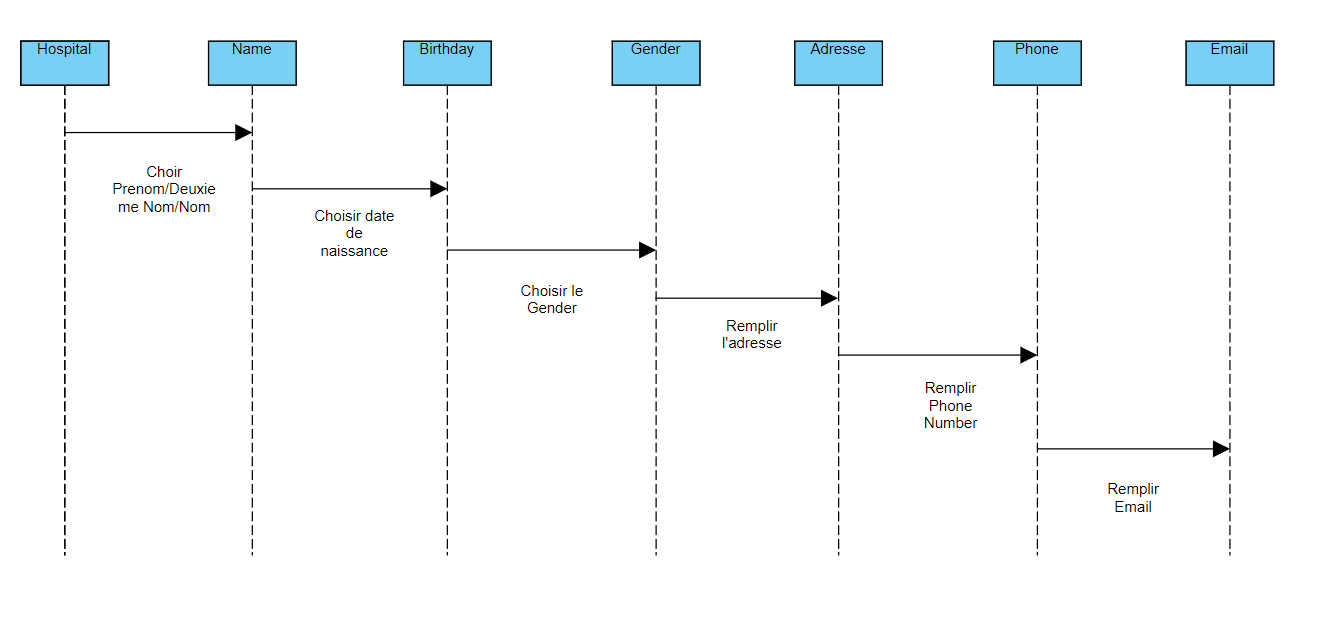
Le formulaire du patient doit être remplie par l’employée. Lorsque l’employée ajoutent les informations du patient, le formulaire devient sous forme Unchecked. Après les contrôles déjà faits sur les Fields, le formulaire devient checked. Enfin, les informations sont enregistrées dans la base de données.

* 1. **Diagramme d’activité :**



L’employé peux subir les Crud Operations (Create ou Update ou Delete) pour un Profile ou pour une History.

* 1. **Diagramme de séquence :**

Les informations nécessaires pour créer un patient sont : prénom, deuxième nom, nom, date de naissance, gender, l’adresse, numéro de téléphone, Email.

**Fonctions :**

**Profile :**

GetProfileEntityById : Get les informations d’un profile par son ID.

CreateProfileEntity : Create un nouveau profile en insérant tous les informations.

UpdateProfileEntity : Update les informations d’un profile.

DeleteProfileEntity : Supprimer un profile.

GetResultByPatient : Récupérer le résultat par ID Profile.

GetHistoryByPatient : Récupérer l’historique par ID Profile.

**History :**

GetHistoryEntityById : Get les informations d’un historique par son ID.

CreateHistoryEntity : Create un nouvel historique en insérant tous les informations.

UpdateHistoryEntity : Update les informations d’un historique.

DeleteHistoryEntity : Supprimer un historique.