

CHAPITRE 3 CONSTRUCTION ET TRANSITION

Nous voilà enfin dans cette dernière partie de notre travail qui reprendra la construction et la transition, c'est-à-dire, respectivement nous donnerons plus vie à notre nouveau système par les diagrammes qui décrivent dynamiquement son comportement, nous réaliserons le système, et nous le déploierons pour une utilisation correcte après test de celui-ci. Nous allons concrètement mettre en œuvre les fonctionnalités de notre système dans la Construction et le préparerons pour le déploiement, soit à l'hébergement dans la partie Transition.

3.1 CONSTRUCTION DU SYSTEME

Cette section sera essentiellement concentrée sur la finalisation du système et la mise en œuvre ou la concrétisation de celui-ci.

3.1.1 Modélisation statique

La modélisation statique nous permettra de représenter la structure de notre système à un moment fixe dans le temps. Pour ce faire, nous mentionnerons le diagramme de classe, le modèle relationnel, ainsi que le diagramme de composant.

a. Diagramme de classes

Ce diagramme est impératif lors de la modélisation d'un système comme le nôtre. Il nous permettra de représenter la structure statique du système en termes de classes et de relations.

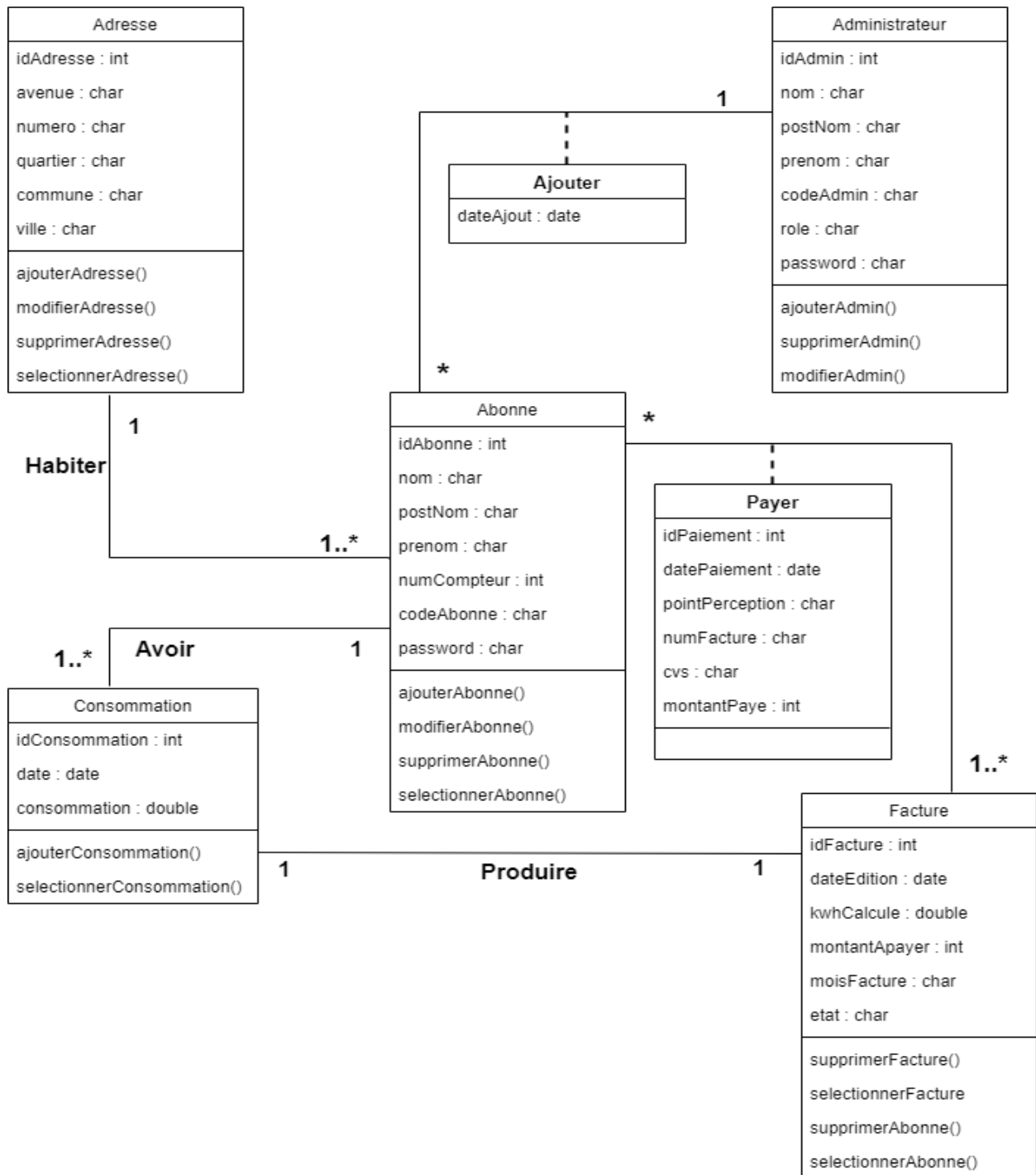


Fig. III – 1. Diagramme de classes

b. Modèle relationnel

Le modèle relationnel est une structure de nos classes vues précédemment sous une forme de tables et de relations entr'elles. Ce modèle nous conduira directement à la mise en œuvre de notre base des données.

Partant du diagramme de classes ci-dessus, nous noterons que les règles au même chiffre, et ayant de relations « plusieurs » les deux (2), vont produire une table ; auparavant une relation, et elle recevra les deux (2) clés primaires de ses tables de relation. Celles à la relation « Une » resteront inchangés, et enfin celles aux relations « Plusieurs » et « Une » enverra la clé primaire de la table « Plusieurs » à « Une » pour en faire une clé étrangère. Ce sont ces règles qui sont appliquées pour produire le modèle relationnel ci-après :

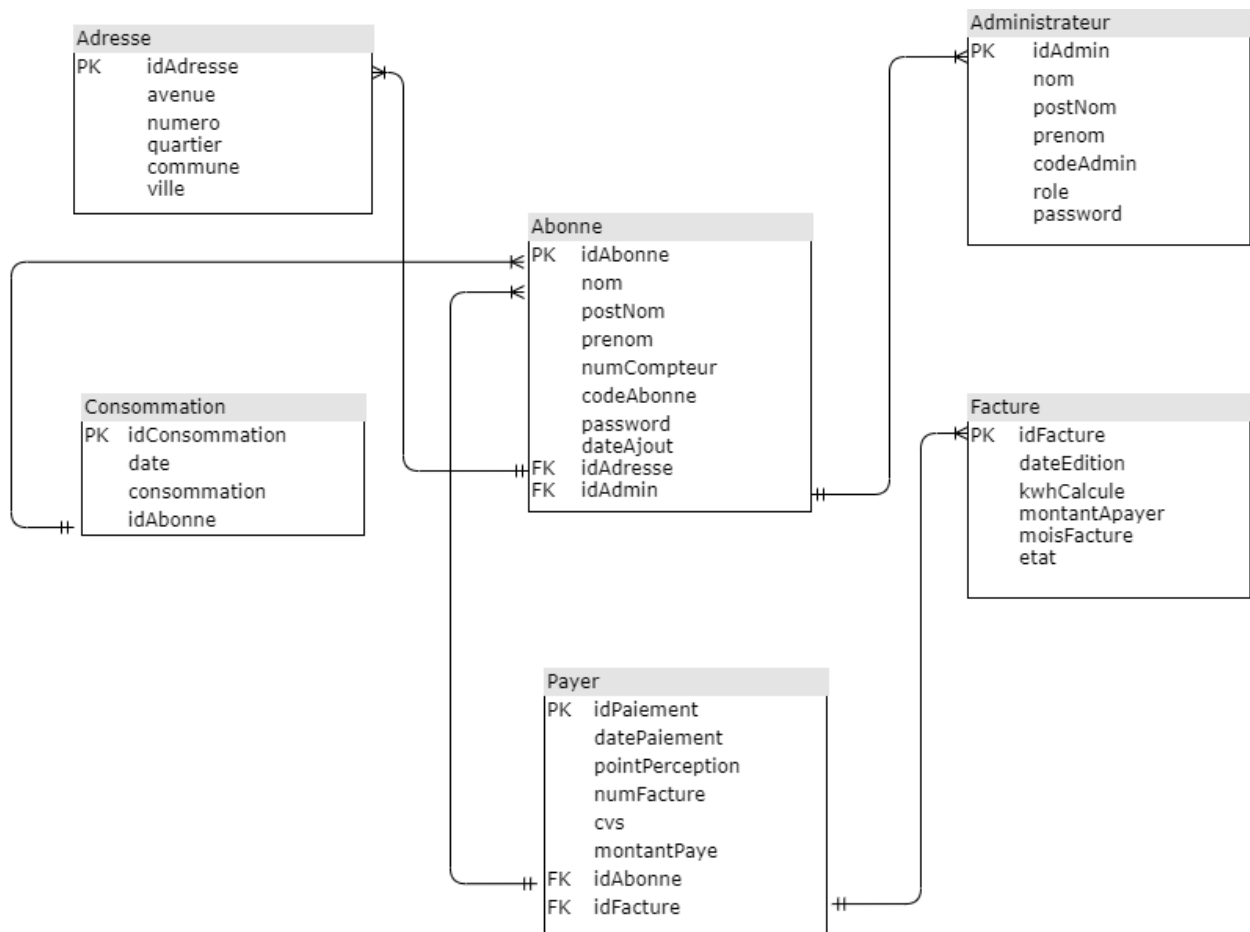


Fig. III – 2. Modèle relationnel

```

CREATE TABLE Adresse (
  idAdresse int(11) NOT NULL AUTO_INCREMENT,
  avenue varchar(25) NOT NULL,
  numero varchar(10) NOT NULL,
  quartier varchar(25) NOT NULL,
  commune varchar(25) NOT NULL,
  ville varchar(25) NOT NULL,
  PRIMARY KEY ( idAdresse )
)
  
```

```
CREATE TABLE Administrateur (  
    idAdmin int(11) NOT NULL AUTO_INCREMENT,  
    nom varchar(20) NOT NULL,  
    postNom varchar(20) NOT NULL,  
    prenom varchar(20) NOT NULL,  
    codeAdmin varchar(10) NOT NULL,  
    role varchar(10) NOT NULL,  
    password varchar(15) NOT NULL,  
    PRIMARY KEY ( idAdmin )  
)  
  
CREATE TABLE Abonne (  
    idAbonne int(11) NOT NULL AUTO_INCREMENT,  
    nom varchar(25) NOT NULL,  
    postNom varchar(25) NOT NULL,  
    prenom varchar(25) NOT NULL,  
    numCompteur int(15) NOT NULL,  
    codeAbonne varchar(25) NOT NULL,  
    password varchar(15) NOT NULL,  
    dateAjout date NOT NULL,  
    idAdresse int(11) NOT NULL,  
    idAdmin int(11) NOT NULL,  
    PRIMARY KEY ( idAbonne ),  
    FOREIGN KEY (idAdmin) REFERENCES Administrateur (idAdmin),  
    FOREIGN KEY (idAdresse) REFERENCES Adresse (idAdresse)  
)  
  
CREATE TABLE Consommation (  
    idConsommation int(11) NOT NULL AUTO_INCREMENT,  
    date date NOT NULL,  
    consommation double (5,5) NOT NULL,  
    idAbonne int(11) NOT NULL,  
    PRIMARY KEY ( idConsommation ),  
    FOREIGN KEY (idAbonne) REFERENCES Abonne (idAbonne)  
)  
  
CREATE TABLE Facture (  
    idFacture int(11) NOT NULL AUTO_INCREMENT,  
    dateEdition date NOT NULL,  
    kwhCalcule double (5,5) NOT NULL,  
    montantApayer int(11) NOT NULL,  
    moisFacture varchar(15) NOT NULL,  
    etat varchar(10) NOT NULL,  
    PRIMARY KEY ( idFacture )  
)  
  
CREATE TABLE Payer (  

```

```

    idPaielement int(11) NOT NULL AUTO_INCREMENT,
    datePaielement date NOT NULL,
    pointPerception varchar(35) NOT NULL,
    numFacture varchar(25) NOT NULL,
    cvs varchar(35) NOT NULL,
    montantPaye int(11) NOT NULL,
    idAbonne int(11) NOT NULL,
    idFacture int(11) NOT NULL,
    PRIMARY KEY ( idPaielement ),
    FOREIGN KEY (idAbonne) REFERENCES Abonne (idAbonne),
    FOREIGN KEY (idFacture) REFERENCES Facture (idFacture)
)

```

c. Diagramme de composants

Ce diagramme nous permettra de voir clair sur les dépendances qui existent entre les composants, et certaines interfaces. Nous avons représenté ici les composants et les interfaces principaux de notre système.

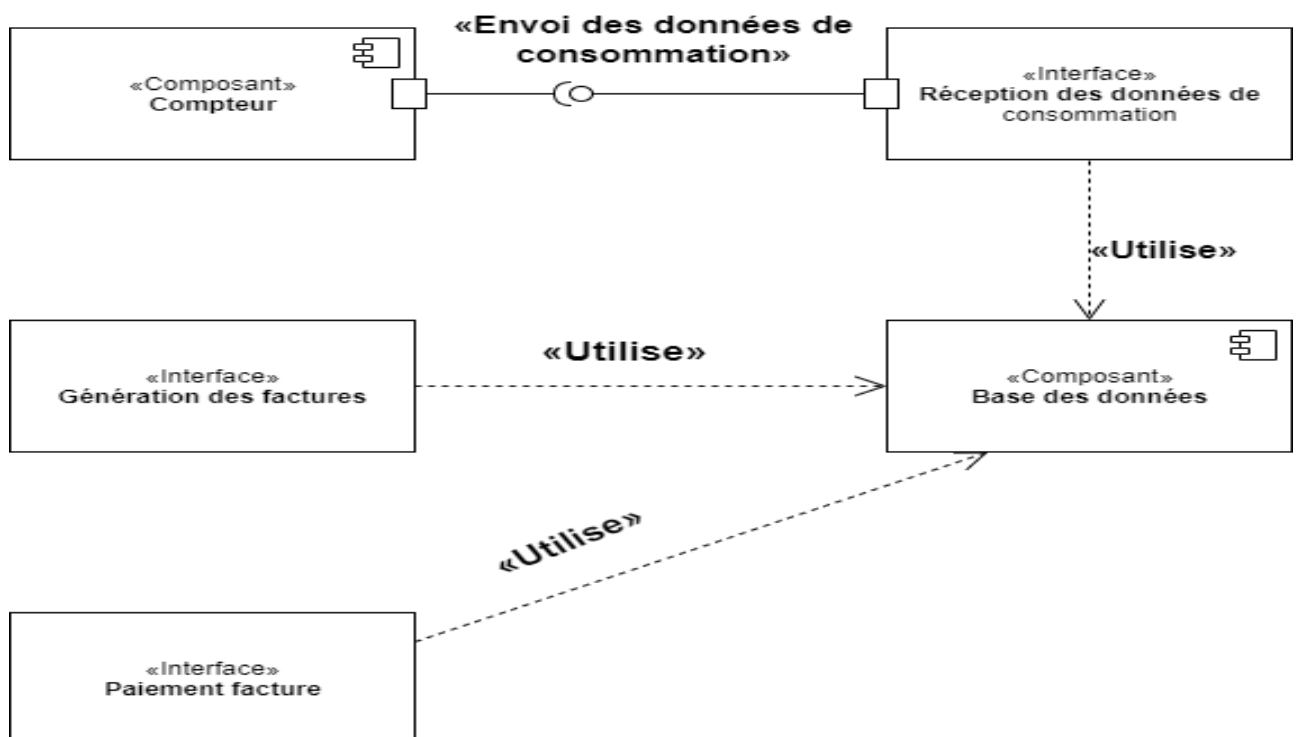


Fig. III – 3. Diagramme de composants

3.1.2 Modélisation dynamique

La modélisation dynamique nous permettra d'exprimer le comportement du système et les différents traitements qui y seront effectués, comment ils changeront d'état d'une situation à l'autre. Deux diagrammes nous serviront dans cette partie : Les diagrammes d'activité et d'état-transition.

a. Diagramme d'activités

Les diagrammes activités va particulièrement mettre l'accent sur les traitements et va représenter un enchaînement d'activités au sein d'une opération ou d'un cas d'utilisation.

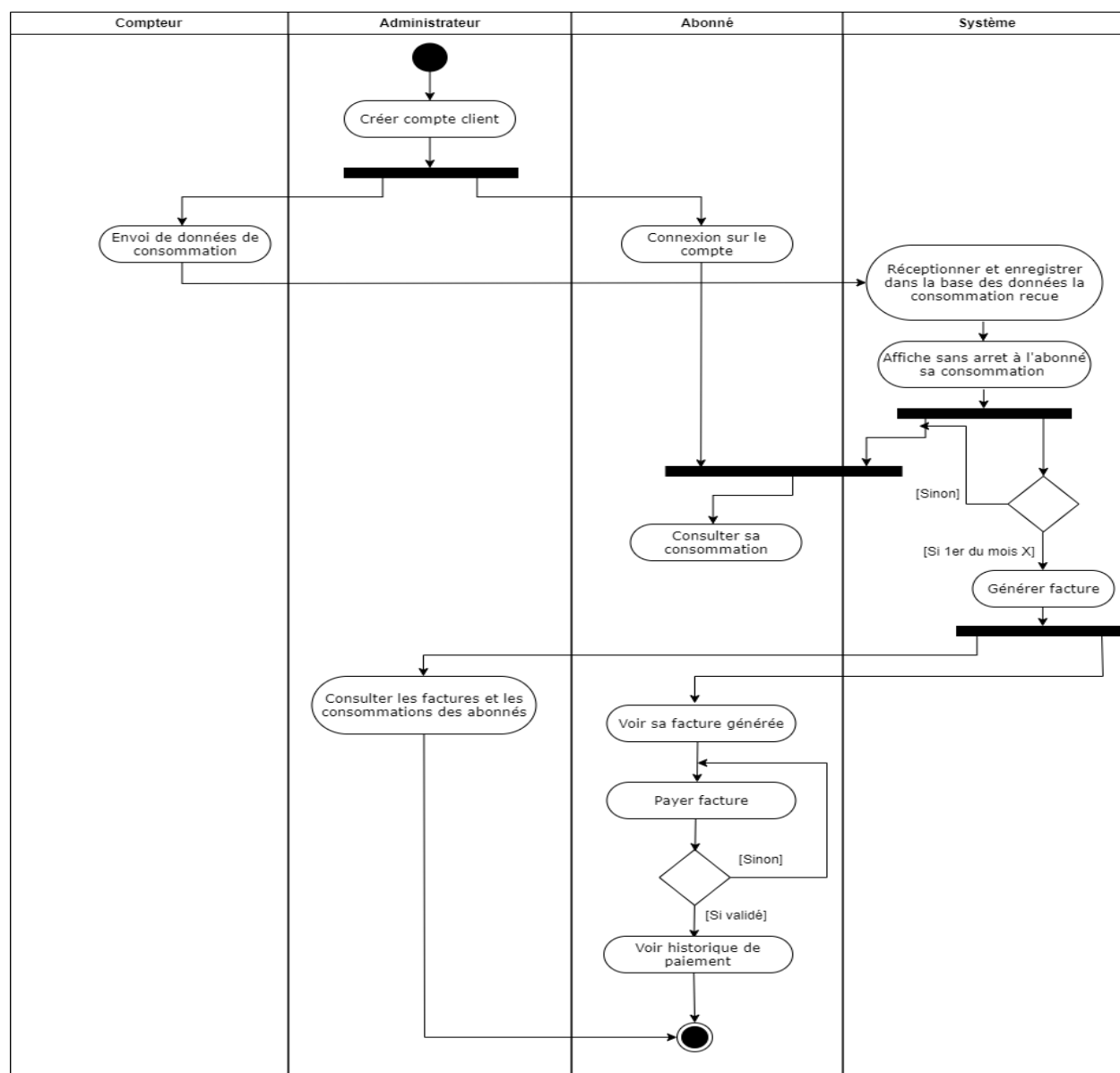


Fig. III – 4. Diagramme d'activités

b. Diagramme d'état-transition

Comme le dit son nom, le diagramme d'état-transition nous aidera à voir les changements d'états des objets et les transitions qui conduisent à ces changements. Dans la transition, nous ferons mention de conditions, dans certains cas, qui peuvent bien mener à un état quelconque.

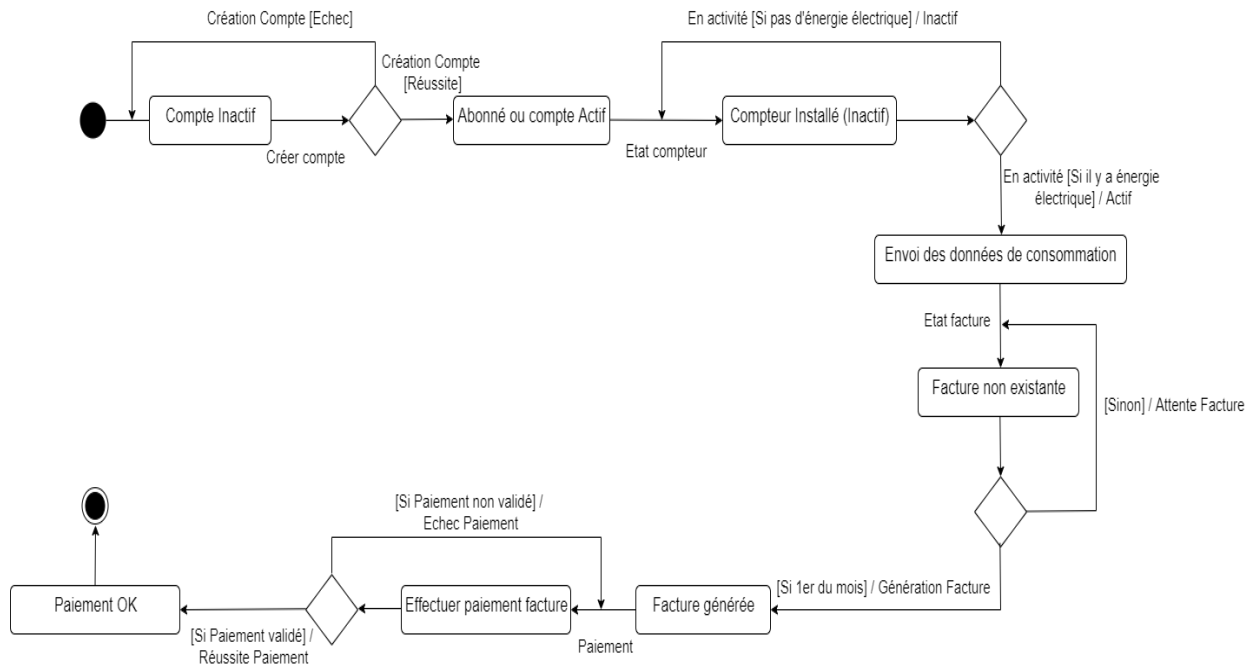


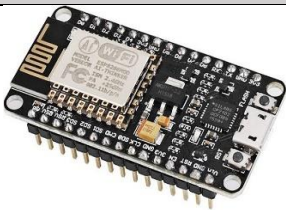


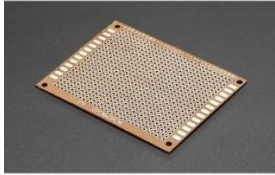





Fig. III – 5. Diagramme d'état-transition

3.1.3 Implémentation du système

Ce travail est composé de deux grandes parties : La partie technique ou électronique pour réaliser le compteur dont on a besoin, le Smart meter, ainsi que la partie logicielle, le système en application qui recevra les données du compteur et dans lequel les abonnés et les administrateurs pourront trouver leur soit étanchée en répondant précisément à leurs besoins.

Ceci étant, nous présenterons alors les équipements électroniques qui nous ont aidés, les logiciels ou les technologies utilisées, les langages de programmation, le SGBD choisi dans notre cas.

a. Présentation des équipements matériels pour le compteur

EQUIPEMENTS		EQUIPEMENTS	
ESP8266 : Microcontrôleur Wifi		Fils électriques mâles et femelles	
SCT013 Sensor : Capteur de courant		Plaque à essai	
Fusible		Résistances	
Boîtier électronique		Condensateur	
Prise électrique			

Tab. III - 1. Équipements matériels pour la réalisation du compteur

- ✓ **ESP8266** : C'est un microcontrôleur avec un module wifi intégré. Grâce à lui, nous avons pu envoyer les données de consommation au serveur hébergeant notre application.
- ✓ **SCT013 Sensor** : C'est un capteur de courant conçu pour mesurer des courants alternatifs jusqu'à 100A. Couramment utilisé dans des projets des IoT (Internet of Things), il peut mesurer la consommation d'énergie des appareils électriques. Il nous a donc permis de mesurer la consommation passant par le compteur.
- ✓ **Fusible** : C'est un dispositif de sécurité qui protège un circuit électrique des éléments comme la surintensité... C'est justement pour cela que nous l'avons mis.

- ✓ **Fils électriques mâles et femelles** : Ils nous ont permis de raccorder nos différents équipements pour le mettre dans le circuit.
- ✓ **Plaque à essai** : Nous y avons réalisé un petit montage pour le SCT013 Sensor principalement.
- ✓ **Résistance** : Un composant électronique qui s'oppose au passage du courant et qui peut le réguler dans un circuit.
- ✓ **Boîtier électronique** : Un boîtier dans lequel on peut placer des composants électroniques pour les protéger physiquement et y monter un système bien à l'abri.
- ✓ **Condensateur** : Un composant électronique qui stocke de l'énergie sous forme de charge.
- ✓ **Prise électrique** : Un dispositif qui peut connecter un appareil électrique au réseau électrique.

b. Système de gestion de base des données utilisé (SGBD)

En effet, nous avons à notre disposition plusieurs SGBD chacun présentant ses avantages. Mais pour sa portabilité, son caractère open source, sa popularité et bien d'autres caractéristiques, notre choix s'est porté sur **MySQL**. C'est un SGBD relationnel qui répond au mieux aux besoins de notre système. Il utilise le langage SQL pour la manipulation de notre base des données.

c. Langages de programmation et autres technologies utilisés

- ✓ **PHP** : **Php HyperText Preprocessor** est un langage de programmation principalement utilisé pour produire des pages Web dynamiques et qui s'exécute via un serveur web.
- ✓ **C++** : C'est un langage de programmation réputée dans plusieurs applications et très performant. C'est un langage aussi utilisé dans la plateforme et IDE Arduino afin de nous permettre de manipuler nos cartes microcontrôleurs.
- ✓ **Arduino** : C'est la marque d'une plateforme de prototypage open-source qui permet aux utilisateurs de créer des objets électroniques interactifs à partir de cartes électroniques matériellement libres sur lesquelles se trouve un microcontrôleur. C'est une plateforme basée sur une interface entrée/sortie simple.
- ✓ **WampServer** : C'est une plateforme de développement web sous windows pour des applications web dynamiques à l'aide des outils qu'ils proposent :

Serveur Apache, du langage PHP et d'une base des données MySQL que nous avons pu gérer grâce à son outil puissant appelé '**PhpMyAdmin**'.

- ✓ **Architecture MVC** : C'est une architecture logicielle qui repartit les responsabilités d'une application en trois (3) composants : Le modèle, le vue et le contrôleur. Le modèle s'occupe des données, la vue est l'interface pour utilisateur et le contrôleur s'occupe du traitement et la logique pure de l'application. Il est l'intermédiaire entre le modèle et la vue.

Noms	Images
Arduino IDE	
Visual Studio Code	

Tab. III - 2. Les environnements de développement intégré (EDI)

d. Présentations des codes sources

e. Présentation des interfaces

3.2 TRANSITION DU SYSTEME

Dans cette dernière section de notre travail, nous allons montrer brièvement de quelle façon notre système sera implanté, son déploiement, pour son utilisation. Nous allons, pour ce faire, utiliser le diagramme de déploiement.

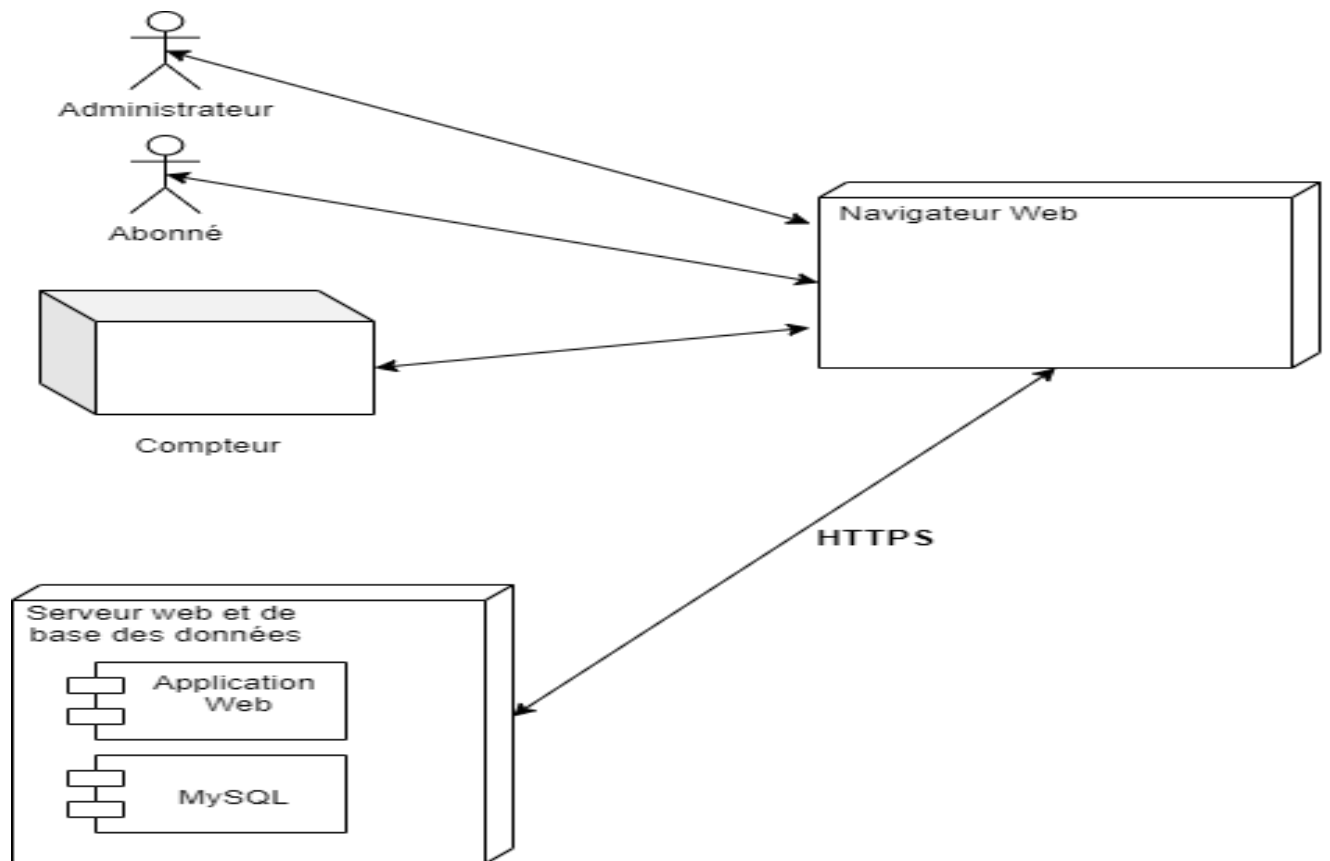


Fig. III – 6. Diagramme de déploiement