

SIN

PROJET STI2D

**SESSION
2018**



**LYCÉE RENÉ GOSCINNY
500 ROUTE DES CROVES - 06340 DRAP**

**BACCALAURÉAT
SCIENCES ET TECHNOLOGIES DE L'INDUSTRIE
ET DU DÉVELOPPEMENT DURABLE**

**SPÉCIALITÉ
SYSTÈMES D'INFORMATION
ET NUMÉRIQUE**

PROJET:

Stores et Lumières Électriques Gérés Par Informatique

**RÉFÉRENCE:
S-AG-U-05**

**CANDIDAT
Valentin MINIÈRE**

**MEMBRES DE L'ÉQUIPE PROJET
Baptiste BELONNE
Alexandre MICHEL
José SRIFI**

Sommaire

I – Partie Générale.....	5
<i>A – Présentation du projet.....</i>	<i>5</i>
<i>B- Cahier des Charges.....</i>	<i>6</i>
<i>C – Documents usuels</i>	<i>7</i>
1 – Diagramme de Gantt	7
2 – Diagrammes de Cas D'utilisation	8
3 - Diagramme de séquence.....	9
4 - Diagramme des exigences,	10
<i>D – Matériel utilisé.....</i>	<i>11</i>
1 – Carte Raspberry Pi 3	11
2 – Carte Arduino Ethernet.....	11
3 – Shield Arduino Carte Moteur.....	11
4 – Shield Arduino Carte Relais.....	11
5 – Shield Arduino Carte Prototype.....	12
6 – Bandeau LED RGB	12
7 – Luxmètre.....	12
8 – Anémomètre.....	12
9 – Store.....	12
<i>E – Logiciels utilisés.....</i>	<i>13</i>
1 – Arduino	13
2 – Visual Studio Code	13
3 – Magic Draw.....	13
4 – Raspbian	13
5 – VNC Viewer.....	13
6 – HTML5 / CSS3 / PHP5 / JQuery/ Base de Données.....	14
<i>F – Lien avec le développement durable</i>	<i>15</i>
<i>G – Répartition des taches.....</i>	<i>16</i>
1 – Baptiste	16
2 – Alexandre.....	16
3 – Valentin (moi)	16
4 – José	16
II – Partie Personnelle.....	17
<i>A – Diagrammes et Algorithme.....</i>	<i>17</i>
<i>B – Code Arduino et php.....</i>	<i>17</i>
<i>C – Problème rencontrés.....</i>	<i>27</i>

I – Partie Générale

A – Présentation du projet

Nous sommes une équipe de quatre élèves de Terminale STI2D (Science et Technologies de l'Industrie et du Développement Durable) spécialité SIN (Science de l'Information et du Numérique). Nous avons choisi de réaliser un projet de domotique consistant à piloter un store et des lumières par informatique.

Le but de ce projet est de nous faire découvrir les aspects de l'informatique, de l'électronique et de la maîtrise des énergies. En effet, depuis le début de notre année de première, nous n'avons abordé que de manière succincte ces domaines lors des TP et nous voulions en découvrir beaucoup plus.

Nous avons le choix entre plusieurs projets, mais aucun d'entre eux ne nous intéressait. Nous avons donc soumis notre idée à notre professeur : « **Piloter un store et des lumières via un site internet** ». Ce dernier a accepté de présenter notre projet à l'académie. Après son accord, nous avons commencé à réfléchir sur comment le réaliser.

Nous avons consacré plusieurs heures à la recherche du matériel, des contraintes, des possibilités et aussi des documents techniques correspondant à nos attentes, tout en gardant à l'esprit que nous avons un budget limité.

Pour notre projet, nous avons choisi le nom de **SLEGPI** :

7



B- Cahier des Charges

La domotique permet de centraliser le contrôle des différents systèmes et sous-systèmes de la maison, du lycée ou de l'entreprise (chauffage, volets roulants, porte de garage, portail d'entrée, prises électriques ...).

Vous aurez à contrôler plusieurs dispositifs à savoir :

- Un store automatique,
- Les lumières intérieures et extérieures.

Pour atteindre cet objectif, vous développerez une liaison Serveur/Client internet. Vous mettrez en œuvre une carte Raspberry et des Arduino organisées de la façon suivante :

- Un module Serveur à l'aide de la carte Raspberry connectée sur le réseau pédagogique (la réalité physique de votre étude se déroulant au lycée).
- Ce module Serveur comprendra également un module radio (Emetteur/Récepteur) recevra les valeurs fournies par le dispositif d'acquisition des valeurs
- Plusieurs cartes Arduino chacune munie de son module radio (Emetteur/Récepteur) pour l'acquisition des données et l'actionnement des parties opératives.

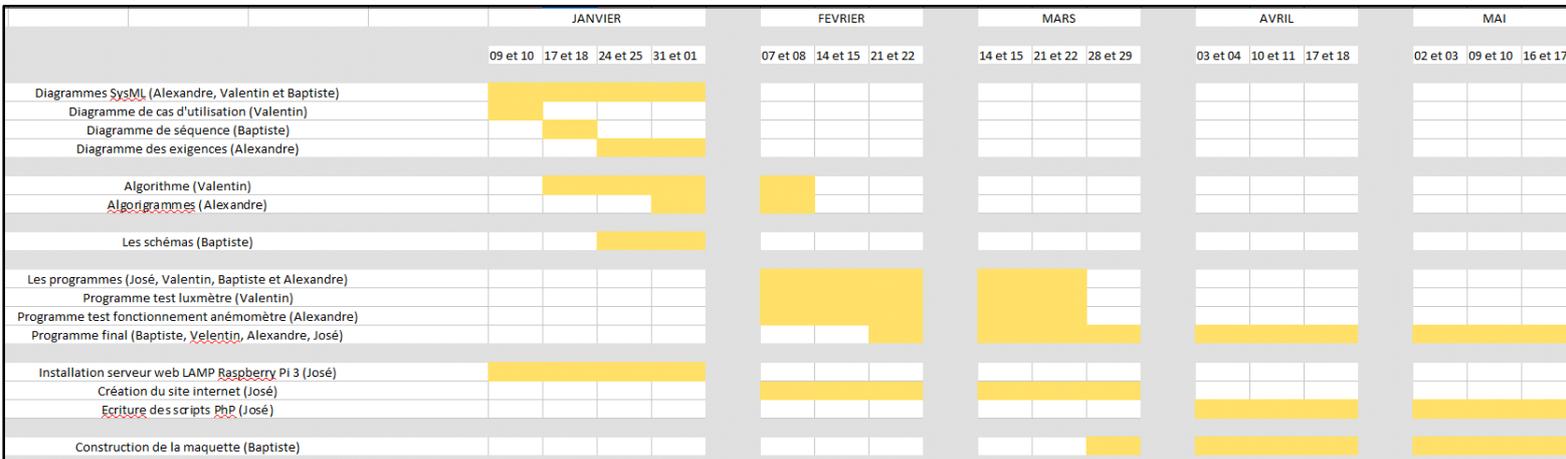
Vous aurez à réaliser le travail suivant :

- Concevoir et implanter un site internet dans le Serveur Raspberry (Serveur Web embarqué),
- Concevoir et implanter les programmes d'acquisition et de commande des parties opératives
- Réaliser les essais,
- Vérifier le bon fonctionnement de la liaison Serveur/Client,
- Rendre compte de toute cette étude dans votre dossier de présentation.

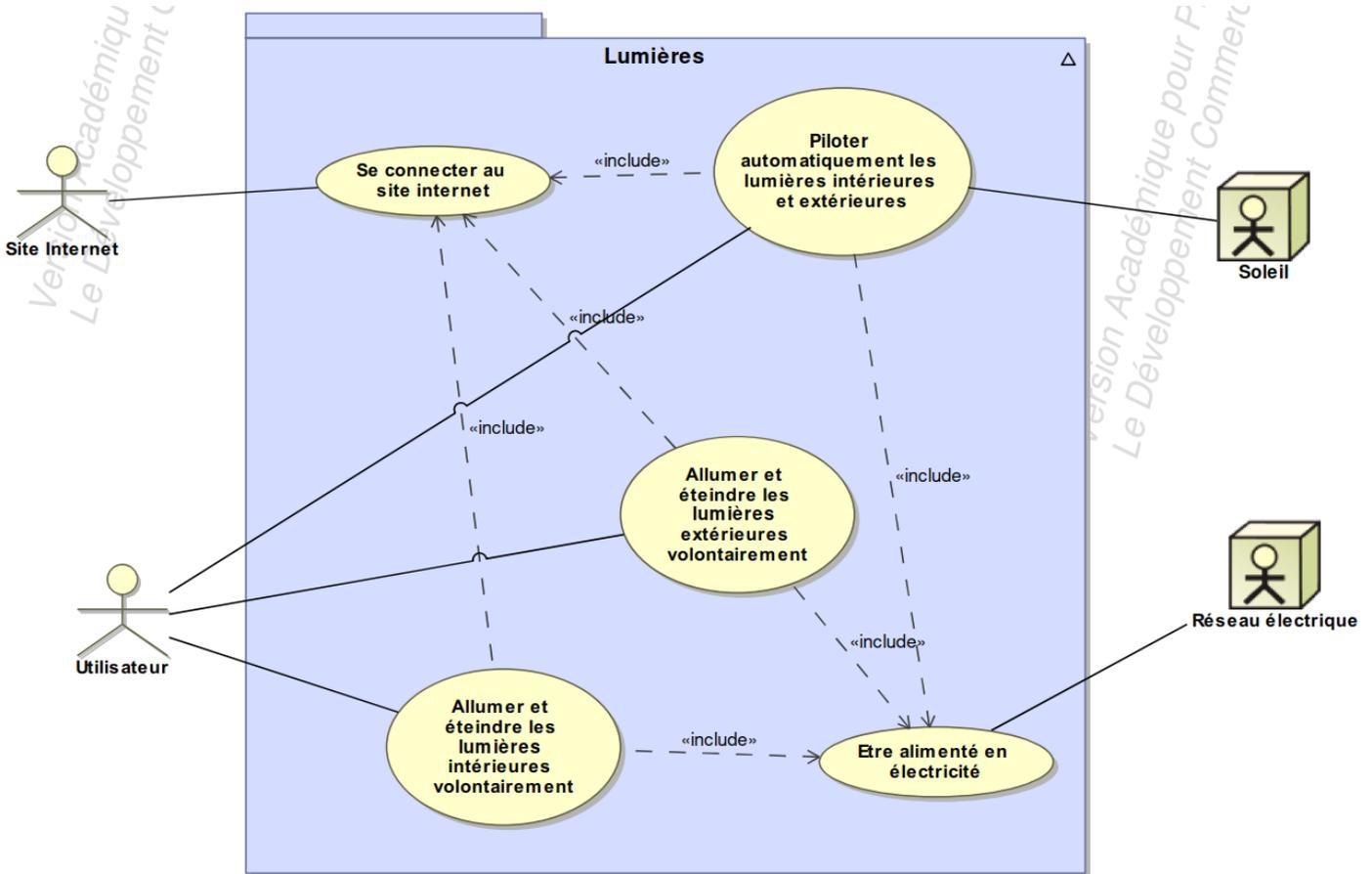
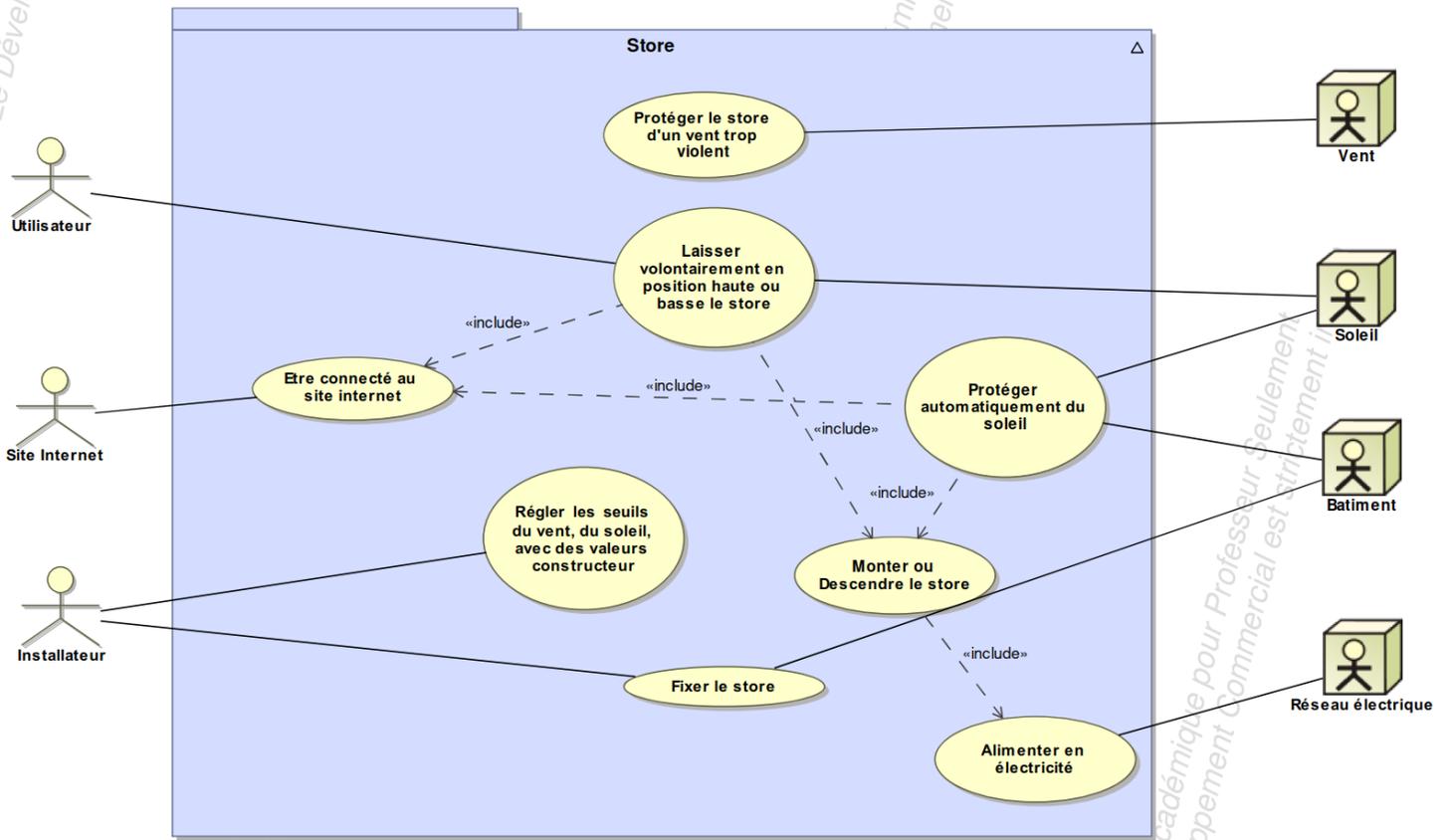
C – Documents usuels

Voici les documents usuels décrivant le fonctionnement du projet.

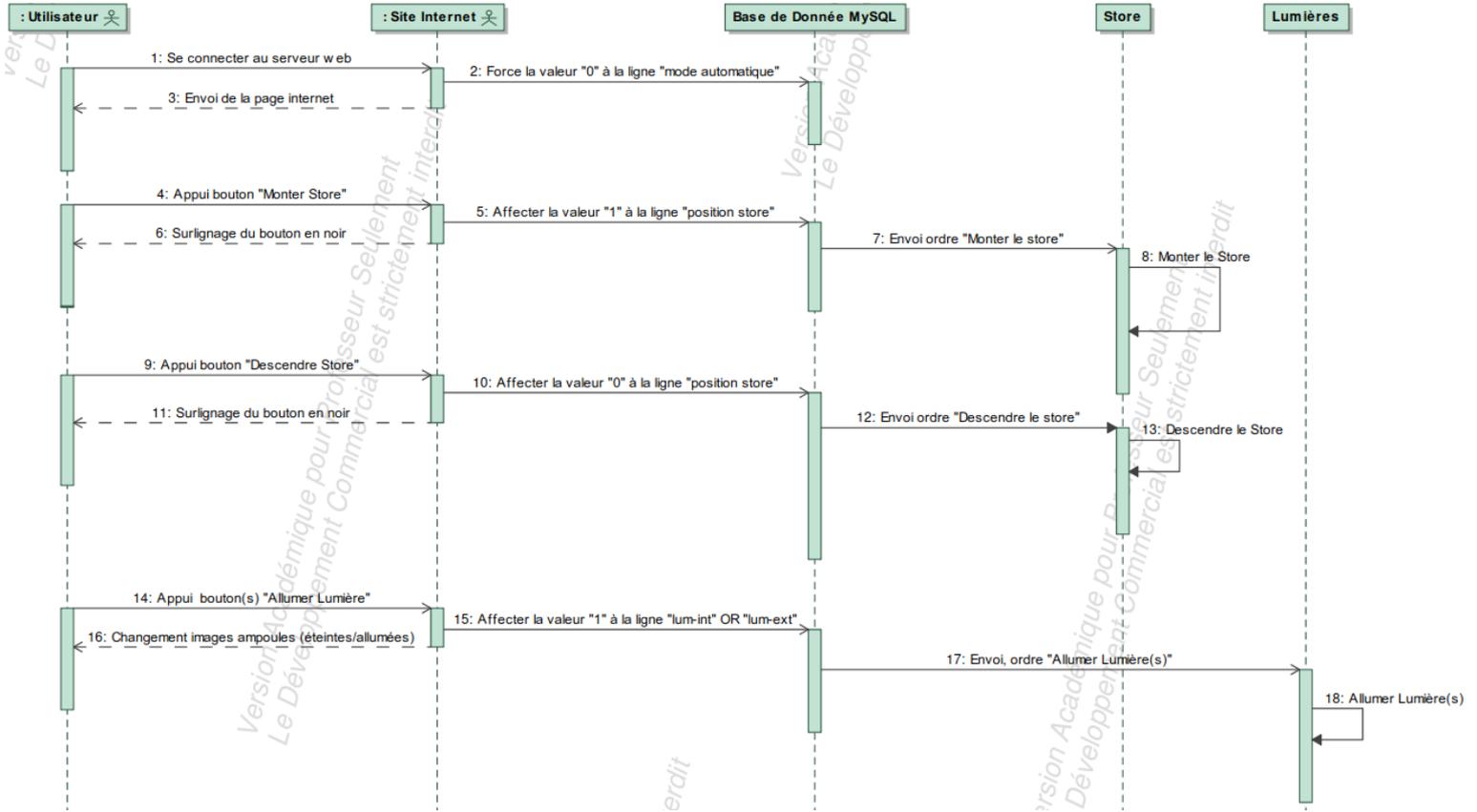
1 – Diagramme de Gantt



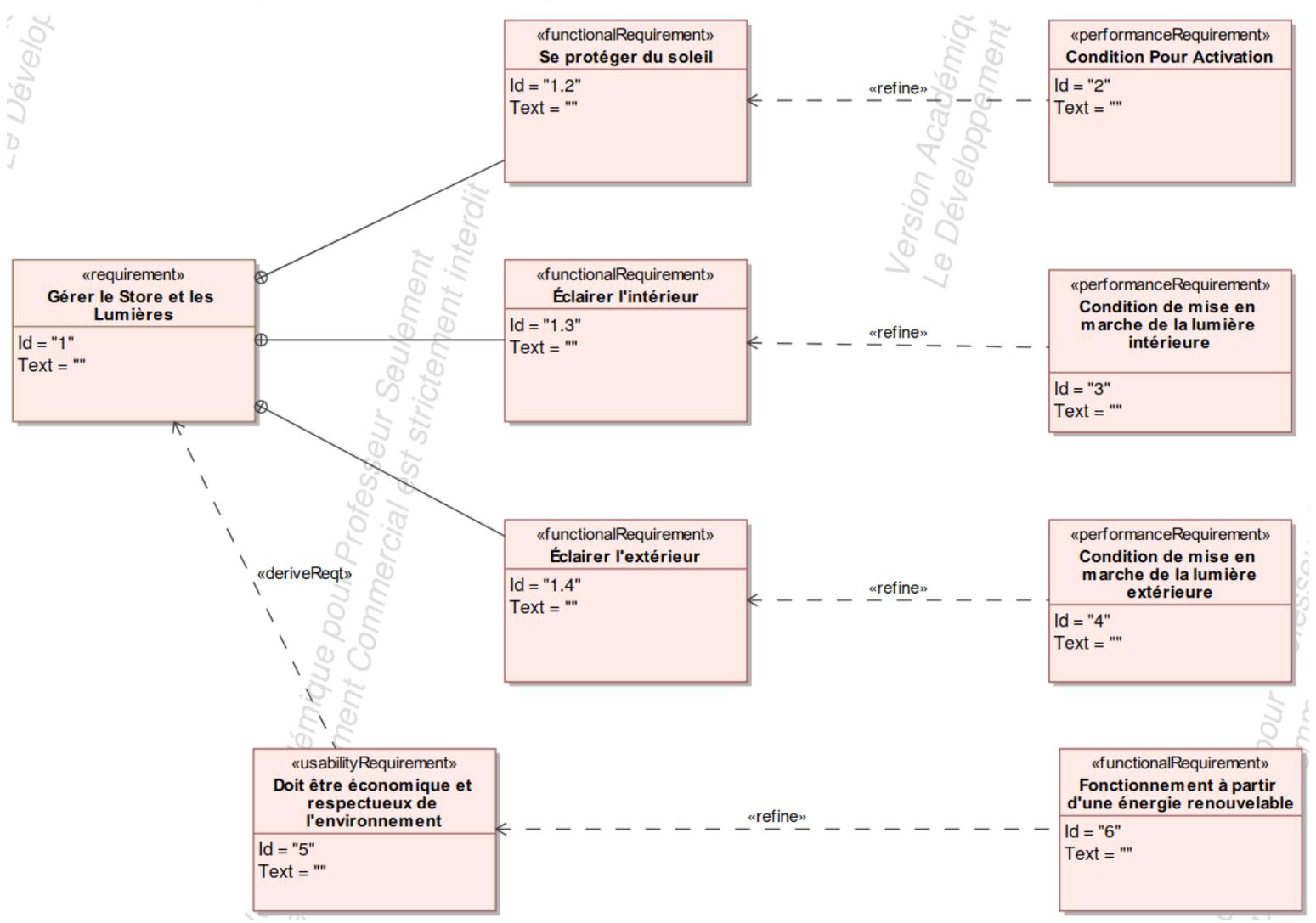
2 – Diagrammes de Cas D'utilisation



3 - Diagramme de séquence



4 - Diagramme des exigences,



D – Matériel utilisé

1 – Carte Raspberry Pi 3

Un Raspberry Pi est un nano-ordinateur (petit ordinateur) mono-carte. Cet ordinateur a la particularité d'avoir une taille proche de celle d'une carte de crédit. Cela nous a permis de la transporter facilement. Cette carte est principalement destinée pour l'apprentissage de la programmation informatique. Elle est compatible avec plusieurs OS (Système d'exploitation). Cela permet l'exécution de plusieurs distributions du système d'exploitation Open Source Linux (voir page 14).

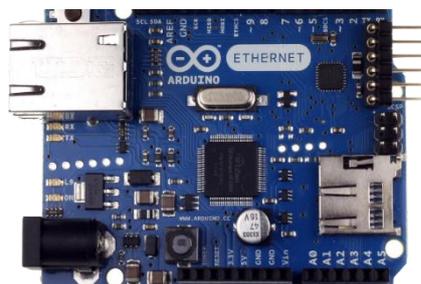


Nous avons utilisé cette carte pour pouvoir héberger notre serveur web.

2 – Carte Arduino Ethernet

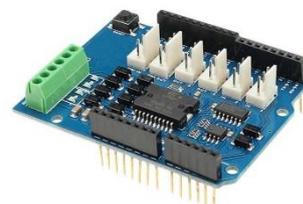
Un Arduino est une carte dotée d'un microcontrôleur qui peut être programmé de manière à effectuer des tâches très diverses comme la domotique, le pilotage d'un robot, de l'informatique embarquée, etc...

Nous avons choisi le modèle Ethernet pour pouvoir communiquer à l'aide d'un câble RJ45 à notre carte Raspberry Pi 3.



3 – Shield Arduino Carte Moteur

Un Shield carte moteur est un petit dispositif qui permet à une carte Arduino de gérer jusqu'à 4 moteurs continus et 2 moteurs pas-à-pas. Dans notre cas, il nous a permis de piloter le moteur de notre store.



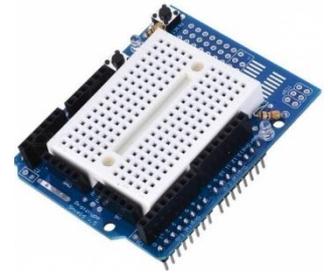
4 – Shield Arduino Carte Relais

Un relais est un organe électrique permettant la commutation de liaisons électriques. Il est chargé de transmettre un ordre depuis la partie opérative à la partie puissance d'un appareil électrique. Utile pour effectuer une commande de puissance de type tout ou rien, il nous a permis de transformer les 5 V de notre carte Arduino en 12 V pour nos Leds.



5 – Shield Arduino Carte Prototype

La carte Prototype est un moyen de tester nos capteurs avec Arduino sans souder de composants entre eux. Dans notre cas, il nous a permis de mettre en place les butées hautes et basses pour notre store.



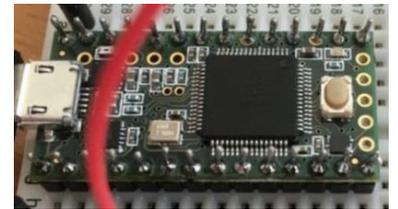
6 – Bandeau LED RGB

Nous avons utilisé des bandeaux LED RGB pour les lumières intérieures et extérieures car celles-ci fonctionnant en 12 V, elles sont compatibles avec notre carte relais.



7 – Luxmètre

Un luxmètre est un appareil permettant de mesurer la quantité de lumière dans l'environnement. Cette valeur s'exprime en lux ou en lumens. Dans notre cas, nous avons utilisé ce luxmètre pour mesurer la quantité de lumière avec notre carte Arduino afin de piloter automatiquement nos parties opératives.



8 – Anémomètre

Un anémomètre est un appareil permettant de mesurer la vitesse ou la pression du vent. Nous nous en sommes servis pour mesurer la quantité de vent avec notre carte Arduino afin de piloter automatiquement l'ouverture et la fermeture de notre store.



9 – Store

Nous avons choisi d'utiliser un store vénitien pour notre maquette car il s'agit du type de store qui pouvait le mieux s'assembler avec notre maquette.

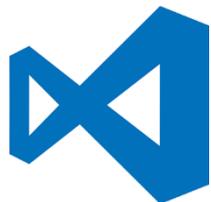
E – Logiciels utilisés

1 – Arduino

Arduino est un logiciel open source permettant de piloter le microcontrôleur de notre carte Arduino Ethernet. Il fonctionne en langage C. Dans notre cas, nous nous en sommes servis pour envoyer les informations nécessaires au microcontrôleur de la carte Ethernet afin de piloter notre store et nos lumières. Notre programme permet de lire et d'écrire dans notre base de données.



2 – Visual Studio Code



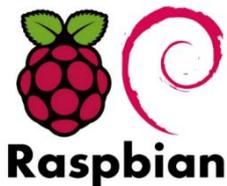
Visual Studio Code est un logiciel d'édition fichiers textuels (mais aussi de pages webs, de scripts js). Nous nous en sommes servis pour créer les pages de notre site internet ainsi que les scripts PHP et js.

3 – Magic Draw

Magic Draw est un logiciel de visualisation et de création de projets en SysML ou UML. Pour notre projet, nous nous en sommes servis pour créer nos « diagrammes de Cas d'utilisation » (page 8), « diagrammes de séquence » (page 9) et « diagramme des exigences » (page 10).



4 – Raspbian



Raspbian est une distribution open source basée sur Linux Debian et optimisée pour fonctionner sur un Raspberry Pi. Nous avons installé cette distribution sur notre Raspberry Pi pour sa stabilité.

5 – VNC Viewer

VNC Viewer est un logiciel de visualisation et de contrôle de l'environnement de bureau d'un ordinateur distant fonctionnant en SSH (Secure Shell). Il permet au logiciel client VNC de transmettre les informations de saisie du clavier et de la souris à l'ordinateur distant. Il nous a permis de travailler sur notre Raspberry Pi sans devoir monopoliser un écran, un clavier et une souris supplémentaires.



6 – HTML5 / CSS3 / PHP5 / JQuery/ Base de Données



Ce sont des langages informatiques. Nous nous sommes servis de HTML5, CSS3 et PHP 5 pour créer notre site internet, de Java Script pour récupérer des informations dans notre base de données afin de les afficher dans notre site internet. Nous avons créé et administré notre base de données MySQL à l'aide de PHPMysqlAdmin. Le jquery nous a permis de dynamiser le site avec un rafraichissement de page localisé.

F – Lien avec le développement durable

Le développement durable répond aux besoins de la population actuelle sans compromettre la capacité des générations futures à répondre aux leurs. Il s'agit d'une nouvelle conception de l'intérêt général, appliquée à la croissance économique et reconsidérée à l'échelle mondiale afin de prendre en compte les aspects environnementaux et sociaux d'une planète globalisée. Un projet s'installant dans le développement durable doit donc respecter les trois piliers fondamentaux de ce mode de développement, c'est à dire les piliers environnementaux (ne pas dégrader la biodiversité pour construire des usines, des résidences ...), économiques (ne doit pas être trop onéreux pour la population par rapport au type de prestations proposées) et sociaux (doit pouvoir être accessible à tous sans distinctions possible).

Notre projet respecte ces trois piliers. Un peu cher à l'achat mais rentable dans sur le long terme. Il fonctionne grâce à l'énergie solaire captée par un panneau solaire, énergie thermique convertie en énergie électrique qui sera ensuite stockée dans une batterie avant d'être utilisée pour faire fonctionner le système. Notre projet est destiné majoritairement pour les particuliers mais peut aussi être utilisé par des entreprises. De plus, il est peu encombrant et se fond bien dans le paysage, on ne le remarque quasiment pas.

G – Répartition des tâches

1 – Baptiste

Baptiste s'est occupé de la mise en place et du pilotage manuel des lumières intérieures et extérieures. Il a réalisé le diagramme de Séquence sur papier. Il aussi réalisé la maquette en bois avec des équerres, des planches de 30cm par 30 cm (soit 900 cm²) et des vis qu'il a achetées. Il a également fixé les planches en plexiglass que José a acheté. Il s'est occupé de mettre en place le pilotage du moteur pour notre store.

2 – Alexandre

Alexandre s'est occupé de la mise en place et du fonctionnement de l'anémomètre, a réalisé les diagrammes de Gantt et des exigences sur papier. Il s'est également occupé de mettre en place la base de données MySQL. Il a réalisé l'algorithme de notre projet et le schéma de câblage.

3 – Valentin (moi)

Je me suis occupé de la mise en place et du fonctionnement du luxmètre. J'ai également mis en place la connexion entre notre carte Arduino et notre base de données pour y écrire les informations pour que le site internet affiche en temps réel la quantité de vent et de lumière. J'ai mis en place la connexion entre notre carte Arduino et notre base de données pour y lire les informations nécessaires au pilotage manuel. J'ai réalisé sur papier le diagramme de cas d'utilisation et aussi l'algorithme du fonctionnement du projet.

4 – José

José s'est occupé de la mise en place du serveur web sur notre carte Raspberry Pi 3 et de la création du site internet. Il s'est également occupé de dessiner les diagrammes des documents usuels sur Magic Draw.

II – Partie Personnelle

A – Diagrammes et Algorithme

J'ai fait le diagramme de Cas d'utilisation que l'on peut voir un peu plus haut dans le dossier ce diagramme permet de savoir ce qui influe sur le projet, plus précisément le store et les lumières. J'ai également fait l'algorithme, cela nous permet de savoir comment notre projet marche, ce qu'il faut faire pour que le store se baisse ou se monte, que les lumières intérieur ou extérieur s'allument ou s'éteigne.

B – Code Arduino et php

Pour ma part comme dit précédemment, je me suis occupé de la mise en place du luxmètre et du bout de programme pour que la valeur reçu s'affiche dans le terminal Arduino :

```
int pin = 5;
unsigned long duration;
float T = 0;
int F = 1/T;

void setup()
{
  pinMode(pin, INPUT);
  Serial.begin(9600);
}
void loop()
{
  delay(1000);
  duration = pulseIn(pin, HIGH);
  Serial.println(duration);
  T = 2*duration;
  Serial.println(T);
  Serial.print("F: ");
  F = 1000000*1/T;
  Serial.println(F, DEC);
}
```

Je me suis également occupé de la liaison entre notre base de données et notre carte Arduino pour que les informations reçues par les capteurs dans le terminal Arduino s'écrivent dans la base de données. J'ai fait cela grâce à un code en php et un programme Arduino :

Le code en php :

```
<?php
    $host = "localhost";
    $user = "root"
    $pass = "root";
    $bdd = "captbdd";
echo"lux";
if (isset($_GET['lux']))
{
    $lux=$_GET['lux'];
    $connexion=mysql_connect($host,$user,$pass) OR die('<p>Connexion impossible à la base de données</p>');
    mysql_select_db($bdd);
    $query = "UPDATE table1 SET lux='$lux' WHERE id =3";
    $result = mysql_query($query);
    }
    else $lux="vide";
```

Et le code Arduino pour aller écrire dans la base de données

```
#include <SPI.h>
#include <Ethernet.h>
int pin = 9;
unsigned long duration;
float T=0;
int F = 1/T;
byte mac[] = { 0x90, 0xA2, 0xDA, 0x0D, 0x06, 0xAF };
byte ip[] = { 172, 19, 3, 110 };
char server[] = "10.66.240.81";
EthernetClient client;
void setup() {
    pinMode(pin, INPUT);
    Serial.begin(9600);
    if (Ethernet.begin(mac) == 0) {
        Serial.println("Failed to configure Ethernet using DHCP");
        for(;;)
            ;
    }
    delay(1000);
    Serial.println("connecting...");
}
void loop()
{
    if (client.connect(server, 80)) {
        Serial.println("connected");
        delay(1000);
        duration = pulseIn(pin, HIGH);
        T = 2*duration;
        Serial.print("Luminosite (lux): ");
        F = 1000000*1/T;
        Serial.println(F, DEC);
        client.print("GET /arduino/ecriture.php?lux=");
        client.print(F);
        client.println(" HTTP/1.1");
        client.println("Host: 10.66.240.81");
        Serial.println(F);
        delay(5000);
        Serial.println("Connection: close");
        client.println();
        client.stop();
        client.println();
    }
    else {
        Serial.println("connection failed");
    }
    delay(1000);
```

Je me suis occupé de la liaison Site Web et Arduino, je me suis occupé de la partie Arduino, pour qu'Arduino reçoive les informations que l'utilisateur veut envoyer via le Site Web :

Voilà la partie de code qui permet de recevoir les informations du site et de monter ou baisser le store et allumer ou éteindre les lumières. (En simulation).

```
byte mac[] = { 0x90, 0xA2, 0xDA, 0x0E, 0xD4, 0x22 };
byte ip[] = { 10,66,240, 59 };
int LE;
int LI;
int S;
char buffer[8];
String readString;
long int i=0;
EthernetServer server(80);

void setup()
{
  Ethernet.begin(mac, ip);
  server.begin();
  Serial.begin(9600);
}
void loop()
{EthernetClient client = server.available();
  if (client)
  {
    Serial.println("attente serveur");
    while (client.connected())
    {
      client.println("z");
      while (client.available())
      {
        char c = client.read();
        if (readString.length()<40 ) readString.concat(c);
        if (c == '\n' )
        {
          LI = 0;
          LE = 0;
          S = 0;
          Serial.println (readString);
          LI = readString.indexOf('Y');
          LE = readString.indexOf('Z');
          S = readString.indexOf('OUV');
          readString="";
          client.stop();
          Serial.println();
          if(LI >0)Serial.println("Lumiere interieur allumer");//Allumer les lumieres exterieurs
            else Serial.println("Lumiere interieur eteind"); //Eteindre les lumieres exterieurs
          if(LE >0)Serial.println("Lumiere exterieur allumer");//Allumer les lumieres interieurs
            else Serial.println("Lumiere exterieur eteind"); //Eteindre les lumieres interieurs
          if(S >0)Serial.println("Store en position haute");//Monter le Store
            else Serial.println("Store en position basse"); //Baisser le Store
```

Programme final :

```
#include <SPI.h>
#include <Ethernet.h>
#include <math.h>
#include <Servo.h>
#define WindSensorPin (2)
volatile unsigned long Rotations;
volatile unsigned long ContactBounceTime;
float WindSpeed;
byte mac[] = { 0x90, 0xA2, 0xDA, 0x0E, 0xD4, 0x22 };
byte ip[] = {10,66,240,59};
//char server[] = "spe.sin.apo.free.fr"; // nom d
char server[] = "10.66.240.81"; // @PC en local q

String readString;
Servo monServomoteur;
boolean drapeau= false;
int D;
int LE;
int LI;
int S;
int AM;
char c;
int pin = 9;
unsigned long duration;
int position = 0 ;
float T = 0;
int F = 1/T;
int Led = 7;
int bLed = 3;
int gLed = 4;
int rLed = 6;
int ST;

EthernetClient client;

void setup() {
  pinMode(pin, INPUT);
  pinMode(bLed, OUTPUT);
  pinMode(gLed, OUTPUT);
  pinMode(rLed, OUTPUT);
  pinMode(Led, OUTPUT);
  monServomoteur.attach(8);
  Serial.begin(9600); // mise en marche liaison séri
  //Ethernet.begin(mac, ip);
  if (Ethernet.begin(mac) == 0) {
```

```

if (Ethernet.begin(mac) == 0) {
  Serial.println("Failed to configure Ethernet using DHCP");
  // no point in carrying on, so do nothing forevermore:
  for(;;){
    delay(2000);
    Serial.println("connecting...");
  }

void loop(){
  //Luxmetre
  delay(500);
  duration = pulseIn(pin, HIGH);
  T = 2*duration;
  Serial.print("Luminosite (lux): ");
  F = 1000000*1/T;
  Serial.println(F, DEC);
  //Anemometre
  Rotations = 0;
  sei();
  delay (3000);
  cli();
  WindSpeed = (Rotations * 0.75) * 1.609344;
  Serial.print(Rotations); Serial.print("\t\t");
  Serial.println(WindSpeed);

  envoieBDD();
  LectureBDD();
  Traitement();
}

void isr_rotation () {

if ((millis() - ContactBounceTime) > 15 ) { // debounce the swi
Rotations++;
ContactBounceTime = millis();
}
}

void envoieBDD()
{
  if (client.connect(server, 80)) {
    Serial.println("connected");
    client.print("GET /SLEGPI/ecriture.php?lux=");
    client.print(F);
    client.println(" HTTP/1.1");
    client.println("Host: 10.66.240.81");
    Serial.println(F);
    Serial.println( "Connection: close" );
  }
}

```

```

    Serial.println( "Connection: close" );
    client.println();
    client.stop();
    client.println();
}
else {
    Serial.println("connection failed");
}

void LectureBDD()
{
    EthernetServer server(80);
    server.available();
    EthernetClient client = server.available();
    if (client)
    {
        Serial.println("attente serveur");
        while (client.connected())
        {
            client.println("z"); //pour la connexion
            while (client.available()) // Not if - you want to
            {
                char c = client.read();//lecture caractère par cara
                //on concatène les caractères et on ne sélectionne
                if (readString.length()<40 ) readString.concat(c);
                // détection de fin de ligne
                if (c == '\n' )
                {
                    Serial.println (readString);//écriture des 40 c
                    // index des recherches des mots "ON" et "OUV"
                    // si absent alors =-1
                    LI = readString.indexOf('Y');
                    LE = readString.indexOf('Z');
                    S = readString.indexOf('OUV');
                    AM = readString.indexOf('WW');
                    readString="";//initialisation de la variable c
                    client.stop();//deconnection de l'arduino
                    Serial.println();
                }
            }
        }
    }
}

```

```

void Traitement()
{
    if(AM >0){
        Serial.println("Mode automatique");

        if(F <1000){
            digitalWrite(bLed, HIGH);
            digitalWrite(gLed, HIGH);
            digitalWrite(rLed, HIGH);
            digitalWrite(Led, LOW);
            if (drapeau==true)
            {
                drapeau=false;
                position=0;
                monServomoteur.write(position);
                delay(15);
            }
            D=1;
            ST=0;
            //allumer

if (client.connect(server, 80) ) {
    Serial.println("connected");
    Serial.println("toto");
    client.print("GET /SLEGPI/ecriture.php?eInt=");
    client.print(D);
    Serial.println(D);
    client.println(" HTTP/1.1");
    client.println("Host: 10.66.240.81");
    Serial.println(F);
    Serial.println( "Connection: close" );
    client.println();
    client.stop();
    client.println();
}
else {
    Serial.println("connection failed");
}
    if (client.connect(server, 80) ) {
    Serial.println("connected");
    client.print("GET /SLEGPI/ecriture.php?eExt=");
    client.print(D);
    client.println(" HTTP/1.1");
    client.println("Host: 10.66.240.81");
    Serial.println(1);
}
}
}

```



```

        client.println("Host: 10.66.240.81");
        Serial.println(F);
        Serial.println( "Connection: close" );
        client.println();
        client.stop();
        client.println();
    }
    else {
        Serial.println("connection failed");
    }
    if (client.connect(server, 80)) {
        Serial.println("connected");
        client.print("GET /SLEGPI/ecriture.php?eExt=");
        client.print(D);
        client.println(" HTTP/1.1");
        client.println("Host: 10.66.240.81");
        Serial.println(l);
        Serial.println( "Connection: close" );
        client.println();
        client.stop();
        client.println();
    }
    if (client.connect(server, 80)) {
        Serial.println("connected");
        Serial.println("toto");
        client.print("GET /SLEGPI/ecriture.php?Store=");
        client.print(ST);
        Serial.println(D);
        client.println(" HTTP/1.1");
        client.println("Host: 10.66.240.81");
        Serial.println(F);
        Serial.println( "Connection: close" );
        client.println();
        client.stop();
        client.println();
    }
    }
    } //Allumer les lumieres exterieurs
else{
    Serial.println("Mode manuel");
        //Eteindre les lumieres exterieurs

    if(LI >0){
        digitalWrite(Led, LOW);

```


C – Problème rencontrés

J'ai rencontré qu'un seul gros problème c'est pour l'écriture dans la base de données, mon programme avait été validés par le professeur mais le programme Arduino et le programme php ne fonctionnait pas ensemble. J'ai donc revue les programmes et j'ai fini par trouver ce qui ne fonctionner pas et j'ai donc réussi à le réglé et à écrire les données des capteurs dans la base de données.