

# RAPPORT DE STAGE DE FIN D'ETUDES

POUR L'OBTENTION DU DIPLOME DE LICENCE APPLIQUEE

EN TECHNOLOGIES DE L'INFORMATIQUE

Parcours : **SYSTEMES EMBARQUES ET MOBILES**

*Intitulé :*

Mise en place d'un système de sécurité  
embarquée

Organisme d'accueil

**Institut Supérieur des Etudes Technologiques de Sousse**

**Encadré par :**

Mr.Mokhtar Harrabi

**Elaboré par :**

Zakraoui Marwen

**Intitulé du stage :** Mise en place d'un système de sécurité embarqué

**Réalisé par les étudiants :** Zakraoui Marwen

**Organisme d'accueil :** Institut Supérieur des Etudes Technologiques de Sousse

**Mots-Clés :** système de sécurité embarquée, ESP32, ESP32-cam, Firebase, application mobile

**Résumé :**

Système de sécurité embarqué qui permet de sécuriser et contrôler la résidence avec un application mobile

**Nom de l'Enseignant  
Encadrant :**  
Mokhtar Harrabi

**Signature :**

**Nom de l'Encadrant  
Industriel :**

**Tampon & Sign.  
de l'Encadrant  
Ind. :**

## **Dédicace**

Nous dédions ce travail, à nos familles qui n'hésitaient pas à nous encourager et à nous supporter Leur soutien et leur amour nous ont donné le courage et la force de continuer et d'achever ce projet de fin d'études.

Nous le dédions aussi à nos chers professeurs en leurs expriment toute notre reconnaissance et notre considération respectueuse pour tous les efforts qu'ils ont fournis.

Nous n'oublierons pas de dédier ce travail à nos amis, avec qui nous avons de très beaux souvenirs et nous avons passé de très bons moments.

## **Remerciement**

Au terme de ce projet, nous tenons à remercier toutes les personnes qui nous ont tendu la main, qui nous ont encouragés et qui nous ont apporté un très grand soutien moral ou concret.

Nous remercions la direction de l'institut supérieur des études technologique de Sousse qui nous a donné l'occasion d'élaborer ce projet de fin d'études.

Un très grand merci à monsieur Harrabi Mokhtar pour la qualité de son encadrement et ses valeureux et pertinents conseils.

Enfin, nous remercions les membres des jurys qui ont accepté d'évaluer notre travail.

# Sommaire

<b>Introduction Générale</b> .....	7
<b>Chapiter1 : l'étude du projet</b> .....	8
<b>Introduction</b> .....	9
<b>I. C'est quoi le système de sécurité ?</b> .....	9
<b>II. Cahier de charges</b> .....	10
<b>III. schéma du projet :</b> .....	11
<b>Conclusion</b> .....	11
<b>Chapitre 2 : Méthodologie de travail (méthodologie agile scrum)</b> .....	12
<b>Introduction :</b> .....	13
<b>I. Méthodologie AGILE :</b> .....	13
<b>III. Méthodologie agile choisie :</b> .....	13
<b>III. Organisation du travail</b> .....	15
<b>Conclusion</b> .....	16
<b>Chapitre 3 : Release1(partie embarquée1)</b> .....	17
<b>Introduction</b> .....	18
<b>I. Organisation des sprints :</b> .....	18
<b>II. Réalisation</b> .....	19
1. Environnement logiciel .....	19
2. Environnement matériel .....	20
<b>III. Produit finale de release :</b> .....	26
<b>Conclusion</b> .....	26
<b>Chapitre4 : Release2(partie embarquée2)</b> .....	27
<b>Introduction</b> .....	28
<b>I. Organisation des sprints :</b> .....	28
<b>II. Réalisation</b> .....	28
1. Environnement logiciel .....	28
2. Environnement matériel .....	28
<b>III. Produit finale de release :</b> .....	31
<b>Conclusion</b> .....	31
<b>Chapitre 5 : Release3 (Application mobile)</b> .....	32
<b>Introduction :</b> .....	33
<b>I. Organisation de sprint :</b> .....	33
<b>II. Conception</b> .....	33
<b>III. Réalisation</b> .....	34

1.Environnement logiciel .....	34
2.Les interfaces d'application .....	36
<b>Conclusion</b> .....	37
<b>Conclusion générale</b> .....	38

## Table des Figures

Figure 1 :Schéma globale du système du projet.....	11
Figure 2 : Vue globale de scrum .....	14
Figure 3 : Equipe Scrum.....	15
Figure 4: Backlog du Produit .....	15
Figure 5 : Arduino .....	19
Figure 6: l'installation du module esp32 .....	19
Figure 7 : suite de l'installation du module esp32 .....	20
Figure 8 : Firebase.....	20
Figure 9 : carte ESP32.....	21
Figure 10 : Extrait de code pour connexion firebase.....	21
Figure 11 : Extrait de code d'écrire sur EEPROM .....	22
Figure 12 : montage de capteur DHT22 avec ESP32.....	22
Figure 13 : Extrait de code lire température et humidité.....	23
Figure 14: capteur de gaz .....	23
Figure 15: capteur de mouvement.....	24
Figure 16: Clavier matriciel 4x3 .....	24
Figure 17 : Écran LCD 16x2.....	25
Figure 18: montage esp-cam avec connecter le FTD .....	29
Figure 19 : récepteur laser .....	30
Figure 20: produit de release 2 .....	31
Figure 21 : Diagramme de cas d'utilisation de l'application .....	34
Figure 22 : Android studio .....	34
Figure 23 : Firebase.....	35
Figure 24 : java.....	35
Figure 25 : xml .....	35
Figure 26 : Interface de connexion.....	36
Figure 27 : Interface d'accueil .....	36
Figure 28 : Interface de compte utilisateur.....	37

## Liste des tableaux

Table 1 : Backlog des Sprints.....	16
Table 2 : Backlog des Sprints (Release 1).....	18
Table 3 : Backlog des Sprints (release2) .....	28
Table 4 : connexion entre esp-cam et FTDI .....	29
Table 5 : Tableau : Backlog des Sprints (release3) .....	33

# Introduction Générale

Les dernières années, la technologie est généralisée dans des nombreux domaines, en particulier dans le domaine de la sécurité domestiques.

Un système de sécurité est un ensemble des techniques d'électronique, de mécanique, d'automatismes, d'informatique et des télécommunications, qui sont utilisées dans les propriétés publiques ou privés tel que les entreprises et les habitations. Les systèmes de sécurité visent à apporter des fonctions de sécurité (comme les alarmes) et de communication (comme les commandes à distance ou l'émission de signaux destinés à l'utilisateur) que l'on peut retrouver dans les maisons, les hôtels et les lieux publics...

La sécurité revêt une importance primordiale pour toutes les résidences, en particulier le système de vidéosurveillance, le système de contrôle d'accès ou encore le système de protection contre les incendies. La surveillance a toujours s'été présente dans l'histoire humaine, elle peut être secrète ou évidente. Un système d'alarme contre l'intrusion et les risques d'incendies.

Alors et dans ce contexte que s'intègre Notre projet de fin d'études, qui a pour objectif de mise en place d'une solution embarquée dans le domaine de la sécurité.

Pour développer notre travail, nous avons organisés notre projet comme suit :

Le premier chapitre est consacré à la description globale de la sécurité, en mettons notre cahier de charge de notre projet. Le deuxième chapitre la méthodologie de travail qui guide ce projet. Le troisième chapitre étudié release 1 partie embarquée 1 en détailler l'environnement matérielle et logiciel. Le quatrième chapitre étudié release 2 partie embarquée 2 et nous finissons notre travail par Le chapitre aborder la conception et la réalisation de la partie mobile.



## **Chapiter1 : l'étude du projet**

## Introduction

Le début de ce chapitre est consacré à la recherche sur les systèmes de sécurité domestique existante par la suite on explique l'idée de projet, on met le cahier de charge et nous finissons ce chapitre par un schéma explicative.

### I. C'est quoi le système de sécurité ?

Après avoir recherché dans les systèmes de sécurité domestique qui existe, nous avons remarqué que le système de sécurité définie comme un ensemble d'éléments mis en place permettant de détecter une situation de danger potentiel et de prévenir les conséquences.

D'une manière générale, il se compose d'une pièce de détection et d'un avertisseur.

Généralement tous les systèmes se composent de :

#### ➤ Les caméras de surveillance :

Les systèmes de caméras de surveillances sont une excellente solution pour prévenir les incidents et aussi d'enregistrer des événements pouvant être importants.

#### ➤ Systeme d 'alarme:

Pour maximiser votre sécurité de votre résidence, il y a plusieurs systèmes d'alarme qui protège contre :

- Les fuites de gaz
- Changement de température
- Le cambriolage...

#### ➤ Systeme de contrôle d'accès

Dans le but d'optimiser la gestion et la sécurité des accès à vos installations il existe multiples types d'accès tel que :

- **Accès par badge**

Avec ce type de contrôle d'accès, un badge est fourni à chaque visiteur légitime. En approchant le badge à la borne, le lecteur détecte le badge individualisé et autorise l'accès.

- **Accès par Lecteur d'empreinte:**

Ce type de contrôle d'accès s'appuie sur la biométrie, qui est l'analyse d'un ou plusieurs caractères physiques uniques de chaque individu. Les empreintes digitales

Le fonctionnement de ces lecteurs est plutôt simple : la personne soumet son doigt (un ou plusieurs selon les technologies) ou son œil à l'analyse lors de l'entrée et dès que l'empreinte digitale est reconnue, l'accès est autorisé.

- **Accès Portier audio/vidéo :**

Le portier audio et/ou vidéo est différent par rapport aux systèmes précédents, d'où il permet une gestion centralisée de l'ouverture d'un grand nombre d'accès.

Le portier audio vidéo sera plus adapté à une installation sur un immeuble. De plus, certains portiers sont connectés, permettant une ouverture à distance depuis son téléphone portable.

L'installation est parfois compliquée, mais les systèmes sans fil atténuent cet inconvénient. Généralement le prix peut être élevé.

- **Accès par Clavier à code :**

En tapant un code sur un clavier à code, chaque utilisateur franchit le contrôle d'accès. La personnalisation des codes et des privilèges sur un même clavier est possible dans une certaine mesure.

Ces technologies ont l'avantage de la simplicité et l'efficacité

Un autre avantage c'est le prix d'où il est très faible par rapport aux autres.

Les techniques utilisées peuvent varier, mais le but est le même. La sécurité dans ce cadre est intégrée notre projet. Alors nous avons présenté le cahier de charge.

## **II. Cahier de charges**

Un bon projet qui répond au cahier des charges demandés, dans ce contexte Le but de projet est la mise en place d'un système de sécurité et de contrôler à base de microcontrôleur qui va sécuriser et contrôler votre résidence.

Nous avons fixé notre cahier des charges du projet dont les points suivants seront recouverts :

- Assurance de la residence:
  - Gestion d'ouverture sécurisée de la porte principale par un code de clavier.
  - Acquisition de la température et humidité par un capteur de température.
  - Détecter les fuites de gaz via un capteur de gaz.
  - Envoyer tous les valeurs (température, gaz) vers une base de données externe (firebase).
  - Gestion des flux d'ouverture/fermeture par capteur de mouvement et envoyer sur base de données.
  - Sécuriser l'environnement de résidence par barrière laser et camera (Esp-cam)
  - Afficher le temps réel des données sur une application mobile.
- Controle de la residence:
  - Changer le code d'accès par l'application mobile.

### III. schéma du projet :

Le schéma synoptique suivant nous permettre de mieux comprendre le fonctionnement global du système étudié :

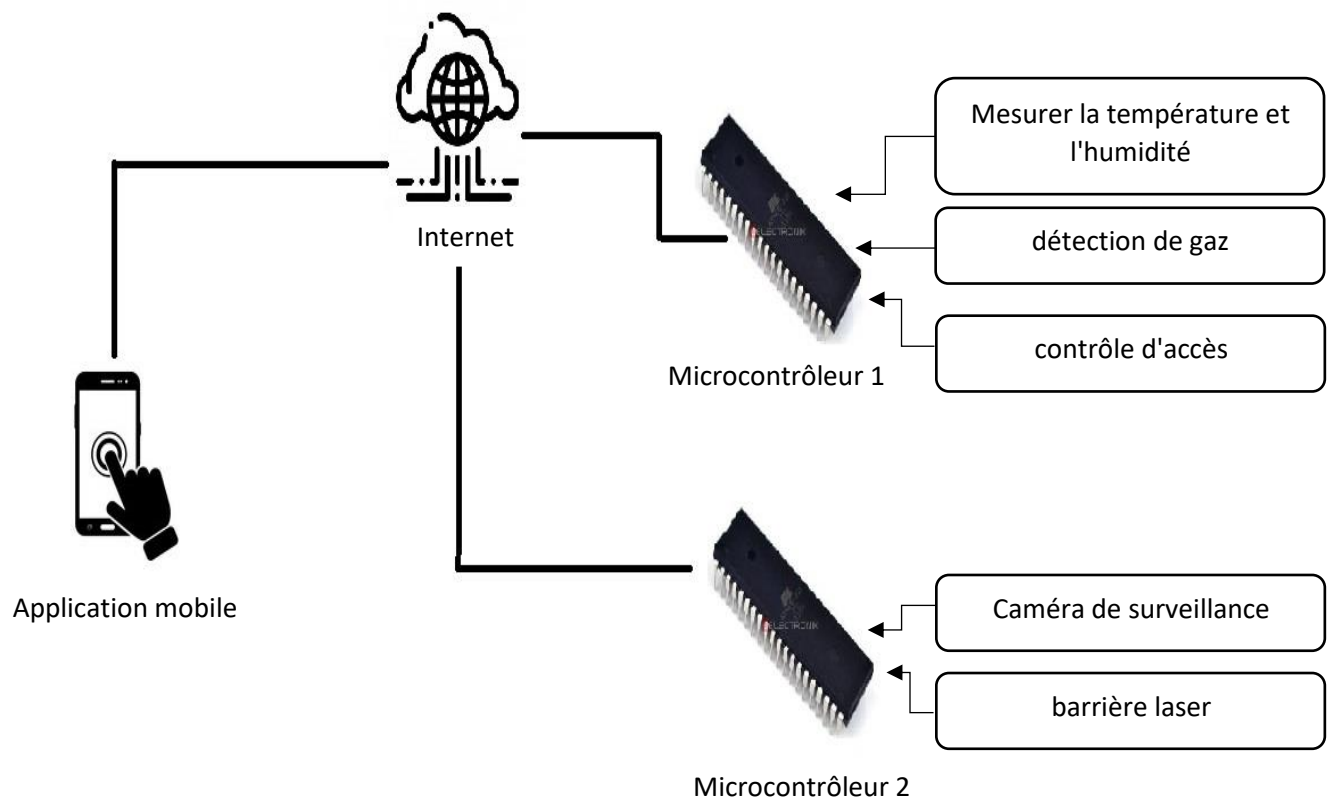


Figure 1 :Schéma globale du système du projet

### Conclusion

Dans ce premier chapitre d'introduction, nous faisons une étude générale du système de sécurité puis on a détaillé l'idée proposée.

## **Chapitre 2 : Méthodologie de travail (méthodologie agile scrum)**

## **Introduction :**

Le contrôle du quelque projet est relativement complexe, si nous ne suivons pas une approche, une méthodologie et une planification bien définie à l'avance.

Donc dans ce chapitre, nous présentons la méthodologie de travail. Puis nous allons définir Back log du Produit et Back log des Sprints.

## **I. Méthodologie AGILE :**

Une méthode Agile est une approche itérative et collaborative, capable de prendre en compte les besoins initiaux du client et ceux liés aux évolutions.

En effet, lorsque on emploie le terme « méthode agile » au singulier on parle d'un concept. Cependant il existe plusieurs méthodes agiles qui se différencient les unes aux autres.

## **Les méthodes agiles :**

➤ Extrême Programming (XP) : c'est une méthode agile de gestion de projet particulièrement bien adoptée aux projets de développement informatique. Le principe fondamental de la méthode XP est de faire collaborer étroitement tous les acteurs du projet est d'opter pour des interactions de développement très courtes. La planification des tâches reste très souple et l'estimation des charges simplifiée par des projections à très court terme.

➤ Scrum: La méthode agile Scrum est particulièrement destinée à la gestion des projets informatiques tient son nom du monde du rugby. Le principe de Scrum est de pouvoir modifier la direction prise par le projet au fur et à mesure de son avancement.

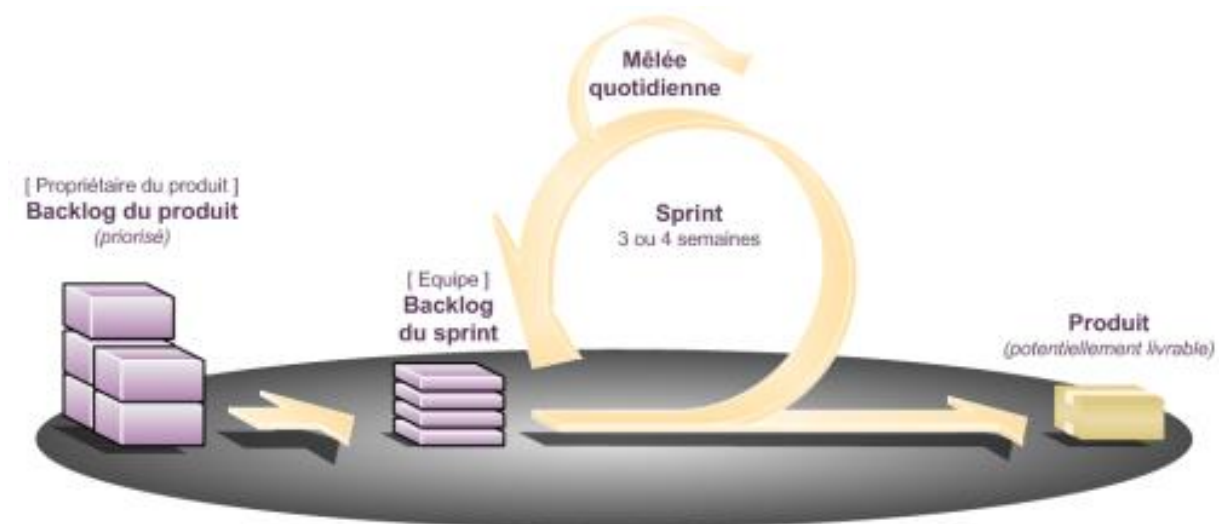
➤ Feature Driven Development (FDD) : Le Développement Dirigé par les Fonctionnalités, en français, est une méthode de gestion de projet basée sur la gestion des risques.

Les développements sont organisés en itération courtes autour de fonctionnalités testables par l'utilisateur..

## **III. Méthodologie agile choisie :**

Nous avons choisi SCRUM comme méthodologie afin de définir le déroulement des tâches ainsi que leurs répartitions. Scrum est un Framework qui est utilisé pour implémenter la méthode Agile de développement et de gestion de projet pour l'objectif d'améliorer la productivité de son équipe.

Afin de comprendre cette méthodologie on a défini des mots clés qui vont nous servir tout au long du projet et de ce rapport.



**Figure 2 : Vue globale de scrum**

Backlog du Produit : Définition des besoins fonctionnels sous forme «user story »

Backlog de Sprint : Liste des taches à implémenter dans un sprint, classées par importance et état.

Burn-Down Chart : Diagramme permettant à suivre l'état d'avancement du projet (Représentation du reste à faire).

Produit Partiel : Résulta du sprint testé et potentiellement livrable.

Scrum Quotidien : Le SCRUM meeting est une réunion d'une durée de 5 minutes, organisée quotidiennement, en position debout.

### **Les principales caractéristiques du modèle Scrum**

- Le **découpage d'un projet** en unités de temps courtes, nommés sprints
- Des **réunions régulières** pour faire le point.
- des **roles**:
  - Une **équipe** pluridisciplinaire (développeurs, architectes, designers, testeurs) ;
  - Le **product owner**, qui a la vision du produit et s'assure de la bonne **traduction des attentes du client à l'équipe projet**, en définissant les spécifications fonctionnelles (ou **specs**) et les **priorités** ;
  - Le scrum master, ou maître de mêlée, qui est le chef d'orchestre, le coordinateur de l'**équipe agile**, dont il fait partie intégrante.

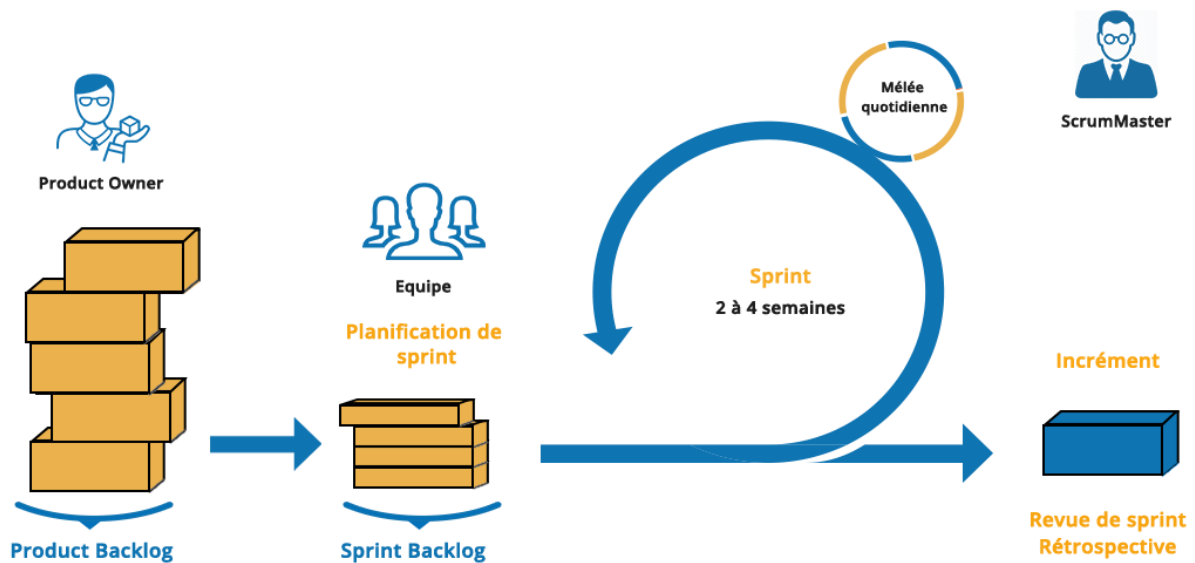


Figure 3 : Equipe Scrum

### III. Organisation du travail

Le Backlog de produit est la liste des fonctionnalités attendues d'un produit. Plus exactement, au-delà de cet aspect fonctionnel, il contient tous les éléments qui vont nécessiter le travail de l'équipe. Les éléments sont classés par ordre de priorité ce permet de définir l'ordre de réalisation. Le Backlog produit est élaboré avant le lancement des sprints, dans la phase de préparation. Il est utilisé pour planifier la release, puis à chaque sprint, lors de la réunion de planification du sprint pour décider du sous-ensemble qui sera réalisé. C'est donc un outil essentiel pour la planification.

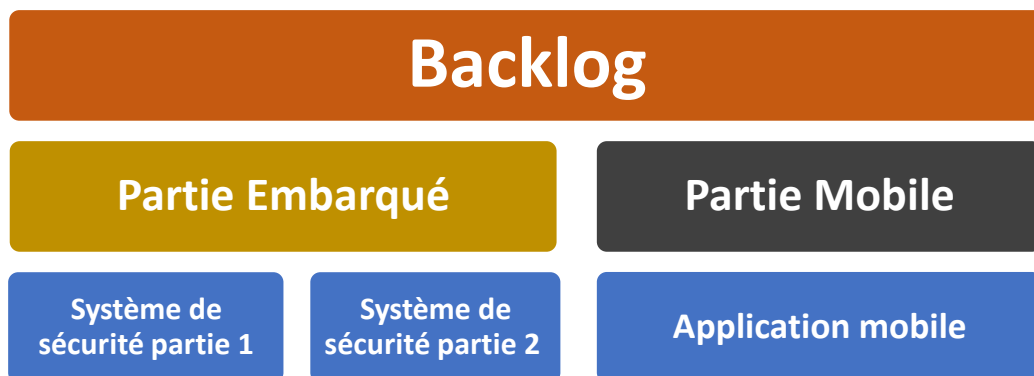


Figure 4: Backlog du Produit



<b>Id Sprint</b>	<b>Nom sprint</b>	<b>Id Story</b>	<b>User stories</b>	<b>Semaines</b>	<b>Release</b>
1	Préparation la structure et développer la partie embarquée1	1	Etude de l'environnement de travail		Release 1
		2	Faire la conception de la partie hardware		
		3	Choisir l'architecture de système et distinguer le matériel nécessaire		
		4	Télécharger tous les logiciels nécessaires		
		5	Développer le code sur Arduino		
		6	Tester le programme		
		7	Faire toutes les étapes nécessaires pour la réalisation des deux circuits		
		8	Tester les cartes		
2	Préparation la structure et développer la partie embarquée2	1	Installer l'environnement et Développer le code sur Arduino		Release 2
		2	Tester le programme		
		3	Faire toutes les étapes nécessaires pour la réalisation des deux circuits		
		4	Tester les cartes		
3	Préparation la structure et développer la partie mobile	1	Modéliser les interfaces des applications		Release 3
		2	Selection les principales fonctions		
		3	Installer les logiciels nécessaires		
		4	Création des interfaces		
		5	Création de la base de donner		
		6	Développer les programmes nécessaires		
		7	Tester les deux application		

**Table 1 : Backlog des Sprints**

## Conclusion

Dans ce deuxième chapitre nous avons présentés la méthodologie à appliquer.

Après avoir choisi Scrum comme méthodologie de conduite de projet, nous avons pu préparer un Backlog pour notre projet.

### **Chapitre 3 : Release1(partie embarquée1)**

## Introduction

Dans ce chapitre nous allons analyser et détailler la release 1 de partie embarquée par la spécification l'environnement matériel et logiciel.

### I. Organisation des sprints :

Ce release, comme il est mentionné ci-dessus, contient un seul Sprint qui est la préparation des structures basiques de la partie embarquée.

<b>Id Sprint</b>	<b>Nom sprint</b>	<b>Id Story</b>	<b>User stories</b>	<b>Semaines</b>	<b>Release</b>
1	Préparation la structure et développer la partie embarquée1	1	Etude de l'environnement de travail		Release 1
		2	Faire la conception de la partie hardware		
		3	Choisir l'architecture de système et distinguer le matériel nécessaire		
		4	Télécharger tous les logiciels nécessaires		
		5	Développer le code sur Arduino		
		6	Tester le programme		
		7	Faire toutes les étapes nécessaires pour la réalisation des deux circuits		
		8	Tester les cartes		

**Table 2 : Backlog des Sprints (Release 1)**

## II. Réalisation

### 1. Environnement logiciel

➤ **Arduino :**



Figure 5 : Arduino

Arduino est un espace de développement intégré (EDI) qui nous permet d'écrire, de compiler et d'envoyer du code sur le circuit imprimé.

Pour le développement de code nous avons utilisé l'Arduino avec un module complémentaire qui permet de programmer l'ESP32 à l'aide de l'IDE Arduino.

Un coup d'œil sur l'installation du module :

Ajouter "[https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json)" dans le champ «URL supplémentaires de BoardManager »

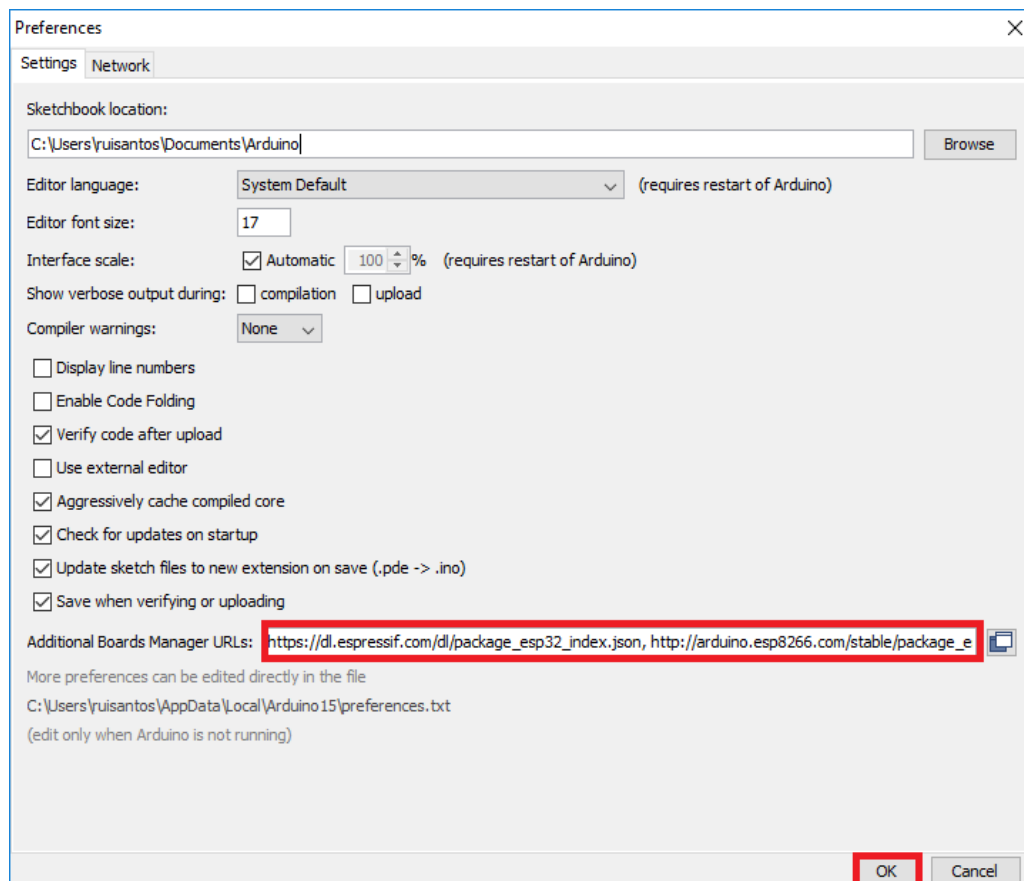


Figure 6: l'installation du module esp32

Puis Rechercher et installer ESP32 sur Gestionnaire de tableaux « ESP32 by Expressif Systems »

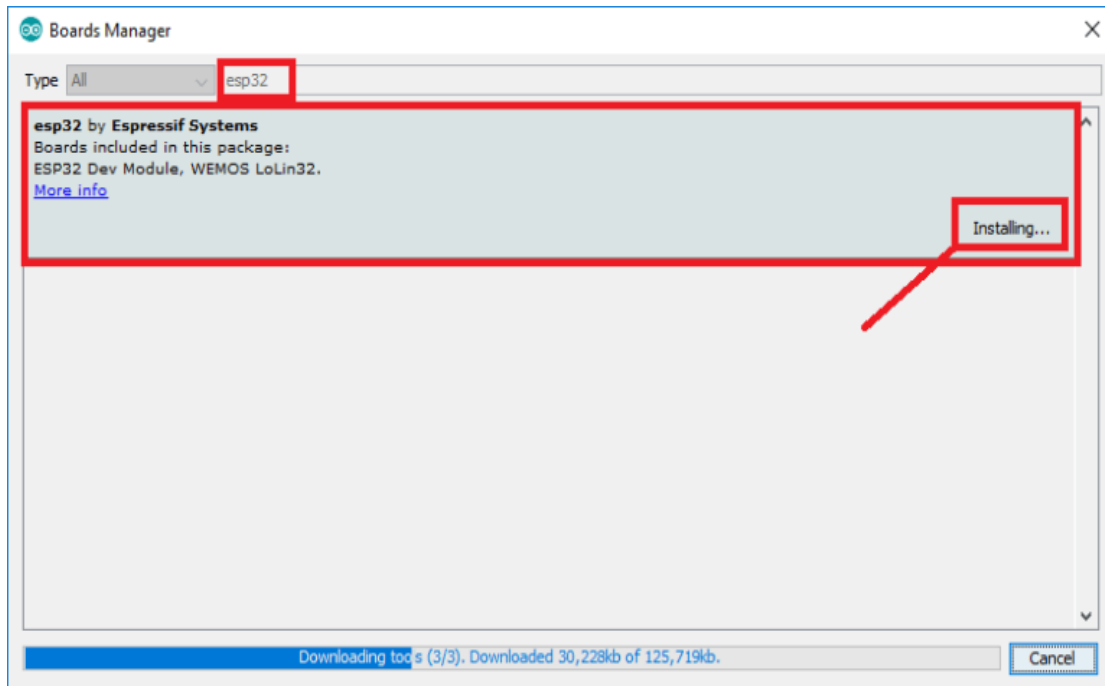


Figure 7 : suite de l'installation du module esp32

## ➤ Firebase :



Figure8 : Firebase

Nous utilisons le Firebase pour créer notre base de données et connecter notre partie embarquée par la partie mobile.

Firebase est un backend en tant que service très utile pour le développement d'applications mobiles.

Il offre de nombreuses fonctionnalités telles que l'authentification, la sécurité, la base de données en temps réel et le stockage de fichiers etc.

## 2. Environnement matériel

Pour notre projet nous choisirons le matériel suivant :

## Carte ESP32-bit



Figure 9 : carte ESP32

Notre projet est besoin a la connectivité internet donc nous utiliserons la module WIFI de la carte ESP32

La carte ESP32 est une carte de développement dédié à l'internet des objets (IoT) et les applications embarquées.

Voilà Extrait de code arduino pour connexion firebase:

```
Fichier Édition Croquis Outils Aide
sketch_apr26a $
1  #include <WiFi.h>
2  #include <FirebaseESP32.h>
3  #define FIREBASE_HOST "XXXXXX " // the pro;
4  #define FIREBASE_AUTH "ztRgolkRUMKEZzaKXmyAC2ZU1GcuWojXXXXXXXXXX"
5  #define WIFI_SSID "Ashish"
6  #define WIFI_PASSWORD "XXXXXXXXXX"
7  String fireStatus = "";
8
9  void setup() {
10     Serial.begin(9600);
11     delay(1000);
12     WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
13     Serial.print("Connecting to ");
14     Serial.println(WIFI_SSID);
15     while (WiFi.status() != WL_CONNECTED) {
16         Serial.print(".");
17         delay(500);
18     }
19
20     Serial.println();
21     Serial.print("Connected to ");
22     Serial.println(WIFI_SSID);
23     Serial.print("IP Address is : ");
24
25     Serial.println(WiFi.localIP());
26     Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
27     Firebase.setString("LED_STATUS", "OFF");
28
29 }
```

Figure10 : Extrait de code pour connexion firebase

Nous avons également besoin d'une EEPROM de cette carte dans laquelle nous mettons le code de sécurité.

EEPROM : Une mémoire morte est une mémoire utilisée pour enregistrer des informations qui ne doivent pas être perdues lorsque l'appareil qui les contient n'est plus alimenté en électricité.

```
EEPROM $
1 #include "EEPROM.h"
2 #define EEPROM_SIZE 4
3 String pass="";
4 void setup()
5 {
6   Serial.begin(115200);
7   Serial.println("start...");
8   if (!EEPROM.begin(EEPROM_SIZE))
9   {
10    Serial.println("failed to initialise EEPROM"); delay(1000000);
11  }
12  Serial.println(" bytes read from Flash . Values are:");
13  for (int i = 0; i < EEPROM_SIZE; i++)
14  {
15    Serial.print(byte(EEPROM.read(i))); Serial.print(" ");
16  }
17  Serial.println("from 0 to 255.");
18
19  println();
20  Serial.println("writing random n. in memory");
21  EEPROM.write(0,1);
22  EEPROM.write(1,2);
23  EEPROM.write(2,3);
24  EEPROM.write(3,4);
25
26  EEPROM.commit();
27
28
29 }
30 void loop()
31 {
```

Figure 11 : Extrait de code d'écrire sur EEPROM

## Les capteurs

### Capteur de température et humidité

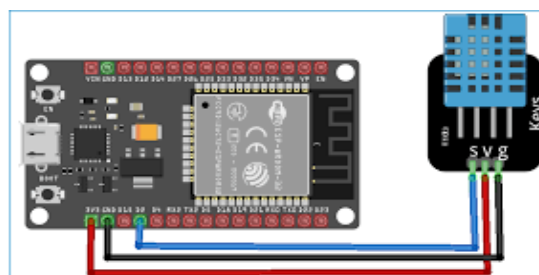


Figure 12 : montage de capteur DHT22 avec ESP32

La DHT22 permettant d'acquérir une température et une humidité ambiante d'une manière numérique. Il utilise un capteur d'humidité capacitif et une thermistance pour mesurer la température et l'humidité de l'air et la transmet d'une manière numérique sur un bus série. Les données sont actualisées toutes les 2 secondes.

La connexion de capteur est très simple, installer bibliothèque DHT22 pour ESP32 et relier le premier pin à gauche à l'alimentation (3V à 5V), le pin central sur une pin ESP-32 déclarée en entrée (INPUT) et le pin de droite à la masse (GND).

Et voilà un extrait de code

```
#include "DHTesp.h"
DHTesp dht;
int capteurTemHum = 15;

void setup() {
  Serial.begin(9600);
  dht.setup(15, DHTesp::DHT22);
}

void loop() {
  int capteur = analogRead(pin);
  float temperature = dht.getTemperature();
  float humidity = dht.getHumidity();
  Serial.println("Temperature: ");
  Serial.print(temperature);
  Serial.println("Humidity: ");
  Serial.print(humidity);
}
```

Figure 13 : Extrait de code lire temperature et humidite

### Capteur de gaz :



Figure 14:capteur de gaz

Le module de capteur de gaz Grove (MQ5) détecte les fuites de gaz (à la maison et dans les bâtiments). Il détecte le GPL, le gaz naturel, le gaz de ville, et bien plus encore. Des résultats rapides grâce à des temps de réponse rapides.



## Capture de mouvement:

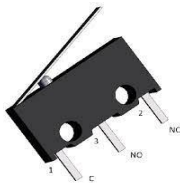


Figure 15: capteur de mouvement

Un bouton poussoir (comme un capteur) sur la porte pour détecter l'état de la porte (fermée ou ouverte)

## Clavier matriciel 4x3 :

Pour communiquer avec le système il doit un clavier, dans notre projet le clavier est principalement utilisé pour saisir un code de sécurité.

Pour la Connectivité entre l'Arduino et clavier Nous avons besoin d'installer la bibliothèque Keypad

Et voilà un extrait de code



Figure 16: Clavier matriciel 4x3

```
sketch_apr27b $
1  #include <Keypad.h>
2  const byte ROWS = 4;
3  const byte COLS = 3;
4  char keys[ROWS][COLS] = {
5      {'1','2','3'},
6      {'4','5','6'},
7      {'7','8','9'},
8      {'*','0','#'}
9  };
10 byte rowPins[ROWS] = {13, 12, 14, 32};
11 byte colPins[COLS] = {26, 25, 33};
12 Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
13
14 void setup() {
15
16 }
17
18
19 void loop() {
20     char key = keypad.getKey();
21
22     if(key){
23         Serial.println(key);
24     }
25
26 }
```

Figure17: extrait de code de keypad

### Écran LCD 16×2 :

Un des éléments permettant d'afficher des informations

L'écran LCD 16×2 présente 16 broches pour permettre la gestion de l'affichage et du contraste.



Figure 17 : Écran LCD 16×2

### Buzzer :

Buzzer est un composant électromagnétique qui transforme l'énergie électrique en vibration, donc en son.

Il est alimenté d'une tension d'alimentation (3.3V-5.0V)

Contient deux bornes : la borne (+) du buzzer à la borne numérique et la borne (-) à la GND de ESP32.

Le buzzer joue généralement le rôle de l'alarme



Figure19: Buzzer

Plusieurs tâches s'exécutent en même temps (mesurer le temp, Envoyer et recevoir des données de firebase, la saisie au clavier) sur le microcontrôleur. C'est ce qui conduit à retard de temps de saisie sur clavier pour éviter le problème nous avons utilisé les deux core de microcontrôleur.

L'ESP32 est livré avec 2 microprocesseurs : core 0 et core 1. Il s'agit donc d'un dual core. Lorsque nous exécutons du code sur Arduino IDE, par défaut, il s'exécute sur le noyau 1. Donc Nous répartissons les tâches sur core 0 et core 1.

### III. Produit finale de release :

Après étude de l'environnement matériel et logiciel, nous avons réalisé la première version de la partie embarquée

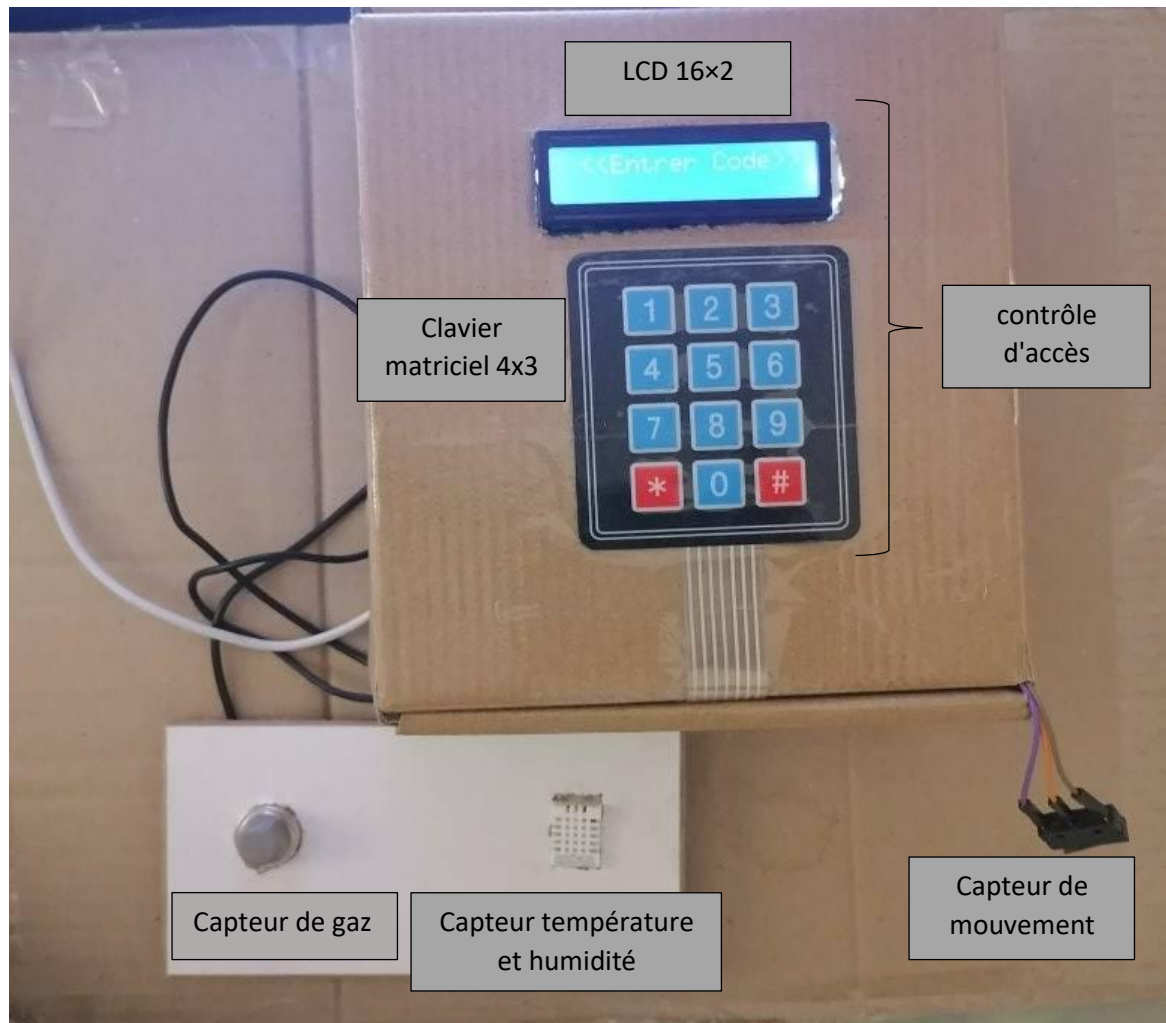


Figure20 : produit de release 1

### Conclusion

Dans ce chapitre, nous avons vu dans la première partie embarquée complète, en montrant la conception, les étapes de réalisation et les logiciels utilisés. Concernant le chapitre qui suit notre effort sera consacré à la production de notre deuxième release.

## **Chapitre4 : Release2(partie embarquée2)**

## Introduction

Dans ce chapitre nous allons analyser et détailler la release 2 de la partie embarquée 2 en détaillant l'environnement matériel et logiciel.

### I. Organisation des sprints :

Ce release, comme il est mentionné ci-dessus, contient un seul Sprint qui est la préparation des structures basique de la partie embraqué.

2	Préparation la structure et développer la partie embarquée2	1	Installer l'environnement et Développer le code sur Arduino		Release 2
		2	Tester le programme		
		3	Faire toutes les étapes nécessaires pour la réalisation des deux circuits		
		4	Tester les cartes		

Table 3 : Backlog des Sprints (release2)

## II. Réalisation

### 1.Environnement logiciel

Nous utilisons Arduino IDE pour programmer la carte ESP32-CAM. Donc, vous devez avoir installé Arduino IDE ainsi que le module complémentaire ESP32.

Nous l'avons fait dans le chapitre précédent.

### 2.Environnement matériel

#### ESP32-CAM :



Figure21 : ESP-cam

L'ESP32-CAM est un très petit module de caméra avec la puce ESP32-S . Outre l'appareil photo OV2640 et plusieurs GPIO pour connecter des périphériques, il dispose également d'un emplacement pour carte microSD qui peut être utile pour stocker des images avec l'appareil photo ou pour stocker des fichiers.

Les caractéristiques:

- Le plus petit module SoC Wi-Fi BT 802.11b/g/n
- CPU 32 bits basse consommation, peut également servir le processeur d'application
- Vitesse d'horloge jusqu'à 160 MHz, puissance de calcul récapitulative jusqu'à 600 DMIPS
- SRAM 520 Ko intégrée, 4MPSRAM externe

ESP32-CAM a besoin d'un connecteur USB pour télécharger le code

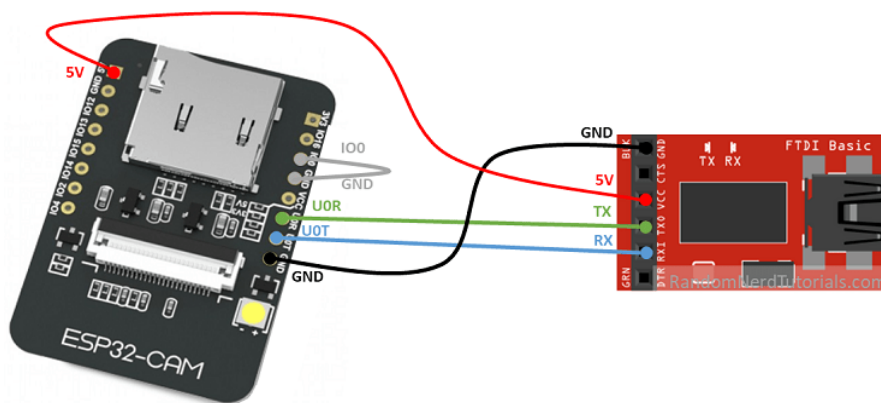


Figure 18: montage esp-cam avec connecter le FTD

ESP32-CAM	Programmeur FTDI
GND	GND
5V	VCC (5 V)
U0R	TX
U0T	RX
GPIO 0	GND

Table 4 : connexion entre esp-cam et FTDI

Avec cette carte et avec le modèle wifi, nous nous sommes connectés à Internet et nous avons envoyé l'image à Firebase.

### **Émetteur laser**



**Figure23: emetteur laser**

Ce module est un émetteur laser rouge pour Arduino, sa tension de fonctionnement est de 5V.

Les broches :

Broche – : GND (Masse)

Broche milieu : +5Vcc

Broche S : Signal

### **Récepteur laser**



**Figure 19 : récepteur laser**

Il peut détecter le laser avec le capteur transparent et envoyer une sortie haute ou basse à carte

Les broches :

Broche – : GND (Masse)

Broche OUT : recevoir de la valeur

Broche VCC : +5Vcc

### III. Produit finale de release :

Comme le chapitre précédent, nous avons réalisé la première version de la partie embarquée dans ce chapitre on doit réaliser la deuxième partie.

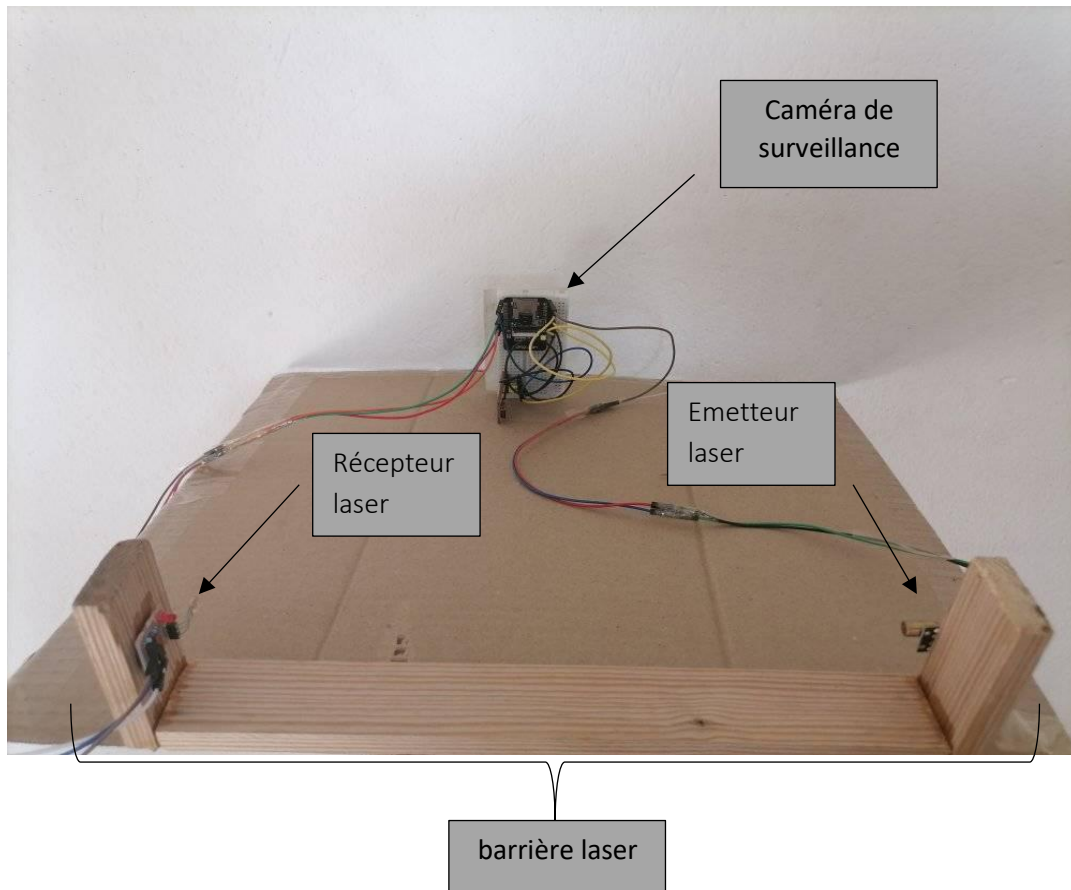


Figure 20: produit de release 2

### Conclusion

Dans ce chapitre, nous avons vu la deuxième partie embarquée complète, en montrant la conception, les étapes de réalisation et les logiciels utilisés. Dans le chapitre qui suit notre effort sera consacré à la production de notre troisième release.



## **Chapitre 5 : Release3 (Application mobile)**

## Introduction :

Nous avons, dans ce chapitre, analyser et détailler la partie mobile. En premier lieu, nous allons présenter la solution conceptuelle en exposant les différents diagrammes qui décrivent l'interaction entre le système et l'utilisateur afin d'atteindre le résultat désiré. Enfin, nous avons présenté la phase de la réalisation et de test.

## I. Organisation de sprint :

Ce release, comme il est mentionné ci-dessus, contient un seul Sprint, la Préparation des structure basic d'application mobile

3	Préparation la structure et développer la partie mobile	1	Modéliser les interfaces des applications		Release 3
		2	Sélection les principales fonctions		
		3	Installer les logiciels nécessaires		
		4	Création des interfaces		
		5	Création de la base de donner		
		6	Développer les programmes nécessaires		
		7	Tester les deux application		

**Table 5 : Tableau : Backlog des Sprints (release3)**

## II. Conception

Dans cette partie, nous allons présenter une description des besoins auxquels notre application doit rependre :

Diagramme de cas d'utilisation de l'application

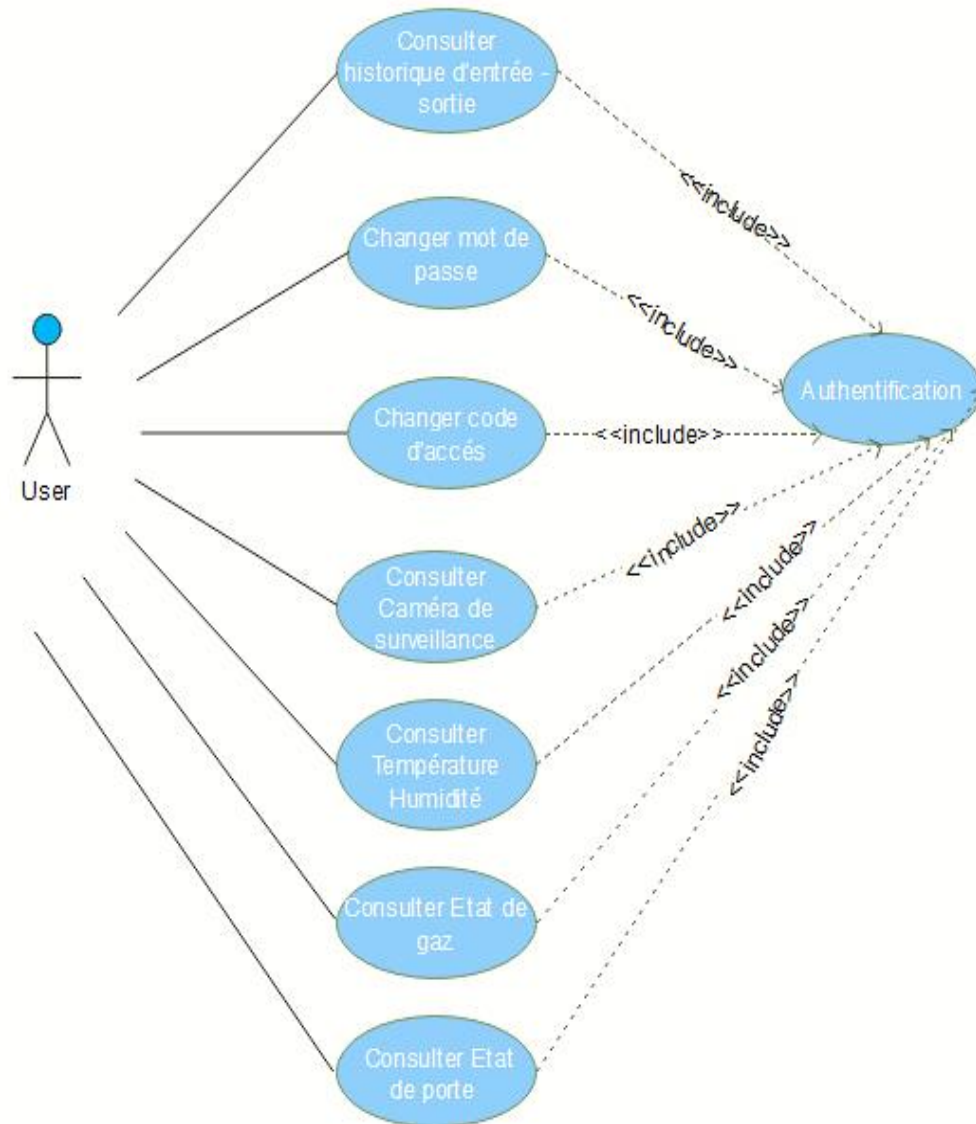


Figure21 : Diagramme de cas d'utilisation de l'application

### III. Réalisation

#### 1. Environnement logiciel

##### ➤ Android

Pour le développement de notre application mobile nous avons choisis Android studio qui est un environnement de développement pour développer des applications mobiles Android.



Figure 22 : Android studio

## ➤ Firebase

Pour la connexion de la partie mobile avec la partie embarquée et la création de notre base de données, on a utilisé Firebase qui est un ensemble de services d'hébergement pour n'importe quel type d'application. Il propose d'héberger en NoSQL et en temps réel des bases de données, du contenu, de l'authentification sociale (Google, Twitter, Facebook, GitHub), et des notifications, ou encore des services, tel que par exemple un serveur de communication temps réel.



Figure 23 : Firebase

## ➤ Les langages à utiliser

- **java** est un langage de programmation orienté objet  
Utilisé à grande échelle dans le monde entier pour le développement d'applications



Figure24 : java

- **xml**  
Le langage de balisage extensible est un langage de balisage qui définit un ensemble de règles de codage des documents dans un format à la fois lisible par l'homme et lisible par la machine. La spécification XML 1.0 du World Wide Web Consortium de 1998 et plusieurs autres spécifications connexes - toutes des normes ouvertes gratuites - définissent XML.



Figure25 : xml

## 2. Les interfaces d'application

### Interface d'authentification

L'accès à l'application est protégé par un système d'authentification. L'utilisateur doit saisir son adresse et mot de passe.

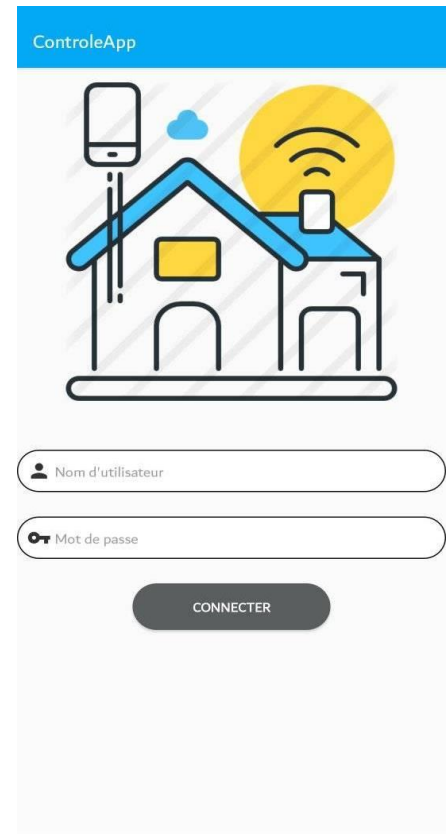


Figure 26 : Interface de connexion

### Interface d'accueil

Après avoir accédé à votre compte, cette interface est principale où l'on retrouve toutes les informations et nous permet de connaître les paramètres qui peuvent être fait

Grâce à cette page, il peut consulter la résidence, changer le code d'accès ou changer le mot de passe du compte, ainsi que voir les historiques d'entre sortie à la résidence.

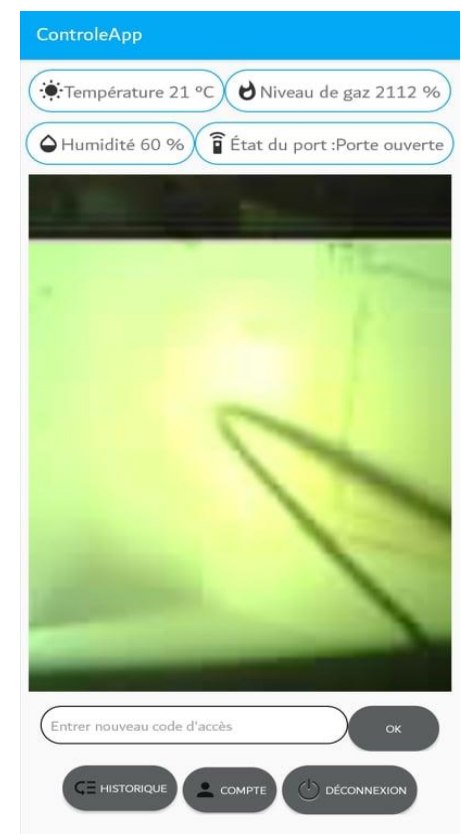


Figure 27 : Interface d'accueil

## Interface de compte utilisateur

A partir de cette interface l'utilisateur peut changer le mot de passe de compte.

ControleApp

Changer Mot de passe

Mot de passe Actuel

Nouveau mot de passe

Confirmer Nouveau mot de passe

ENREGISTER

Figure 28 : Interface de compte utilisateur

## Conclusion

Au long de ce chapitre, nous avons présenté la conception de la réalisation de l'application mobile qui fournissent des fonctionnalités précises et avec cette partie « Mobile » nous avons réussi à produire notre quatrième et dernière version.

## Conclusion générale

La réalisation de ce projet de fin d'études, nous donne une véritable occasion pour rencontrer les difficultés et l'environnement du travail qu'un informaticien devra affronter lors de la conception et de réalisation d'un projet.

Le travail que nous avons réalisé consiste à concevoir et à développer une application mobile et une partie embarquée qui s'intitule « Conception et réalisation d'une application mobile dans le domaine sécurité ».

Nous avons, par suite, situé le projet dans son cadre général tout en mettant l'accent sur les situations existantes en proposant une solution pour atteindre l'objectif donnée à travers une méthodologie précise « SCRUM ».

En effet, nous avons présenté les besoins fonctionnels et non-fonctionnels avec un cahier de charge bien détaillées.

Nous avons montré les outils nécessaires à notre travail, et par la suite, nous avons révélé la planification de notre travail qui nous a permis de détailler les différentes tâches de notre solution.

Ensuite, nous avons présenté le Backlog du produit de notre solution. Nous avons évoqué aussi l'analyse et la conception de chaque sprint. Et à la fin, nous avons présenté notre application en détaillant chaque sprint et en mettant l'accent sur l'implémentation.

Finalement, ce stage nous a offert l'opportunité de nous intégrer dans l'environnement de travail en groupe et d'améliorer nos capacités à la vie professionnelle. Il nous a aussi permis d'enrichir nos connaissances et notre formation théorique et pratique acquise tout au long des trois années de formation.

## **Webographie**

<https://firebase.google.com/>

<https://developer.android.com/>

<https://stackoverflow.com/>

<https://icons8.com/>

<https://www.ideematic.com/actualites/2015/01/methodes-agiles-definition/>

<https://www.appvizer.fr/magazine/operations/gestion-de-projet/scrum>

<https://randomnerdtutorials.com/esp32-cam-video-streaming-face-recognition-arduino-ide/>