

Institut Galilée
Université Sorbonne Paris Nord – Paris 13
M1 Informatique - Internet of Things

ALARME D'INCENDIE

Group: Legendary Enigma

Membre: Minh Hao LE

Encadré par: M. Aomar Osmani

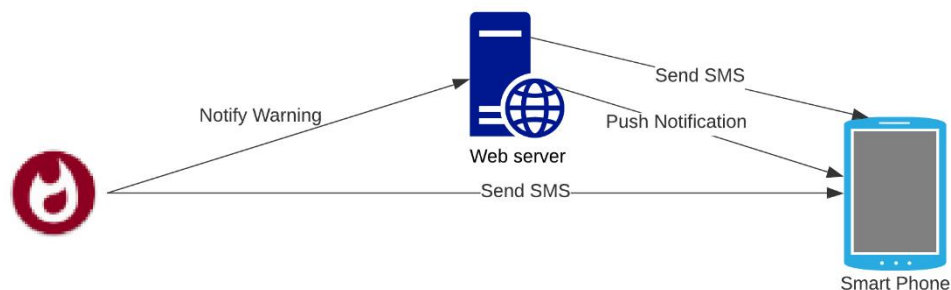
M. Massinissa Hamadi

I. Description de projet

Actuellement, on a souvent vu qu'il y a beaucoup d'alarmes d'incendie dans les maisons parce que les gens soucient de leur sécurité.

Ces alarmes ont le haut-parleur et le feu d'avertissement. Mais il n'y a pas beaucoup d'alarmes d'incendie qui ont l'alerte sms. Quand le client n'est donc pas à la maison, le problème devient sérieux. Les alarmes d'incendie qui ont les alertes sms sont très chers comme PARADOX WC588P - 140\$, XFM70627901ES – 400\$, ...

Notre projet est de développer un alarme d'incendie qui a normalement un le haut-parleur et le feu d'avertissement. De plus, il peut envoyer les alertes sms quand la maison a ou n'a pas de Wifi, de signal téléphonique. En plus, on voudrait diminuer le prix de produit.



Il y a quatre méthodes principales pour informer le client : la sonnerie retentit à volume élevé, le feu d'alerte, envoyer un sms par SIM800L, envoyer un sms par AWS SNS.

D'abord, nous avons développé un prototype dans le confinement, avec l'utilisation des matériels nécessaires et le structure de projet inclut les services de AWS.

II. Use case diagram & Fonctionnalité

Les fonctionalites de produit sont presentees en Use case diagram suivant:



Le dispositif peut détecter du gaz, de la fumée et il pourra provoquer la sonnerie eleve et clignoter la LED. En même temps, il enverra un sms au telephone du client et un message au Webserver. De plus, le dispositif peut publier la temperature au AWS IoT chaque 30s.

Webserver peut verifier le connexion de l'application mobile. Il stocke aussi les informations de dispositif, les donnees de captures. En d'autre terme, il peut envoyer un sms au telephone du client par le message par le dispositif quand la fumee est detecté. Webserver peut fournir les donnees en temps reel d'un dispositif.

Le client peut utiliser l'application mobile pour se connecter, déconnecter, configurer les parametres comme (Wifi, numero de telephone). Le client peut aussi tester le device ou regarder la temperature en temps reel. Chaque dispositif a un nom unique et le mot de passe, ces informations sont écrites sur les étiquettes qui sont collées sur le dispositif.

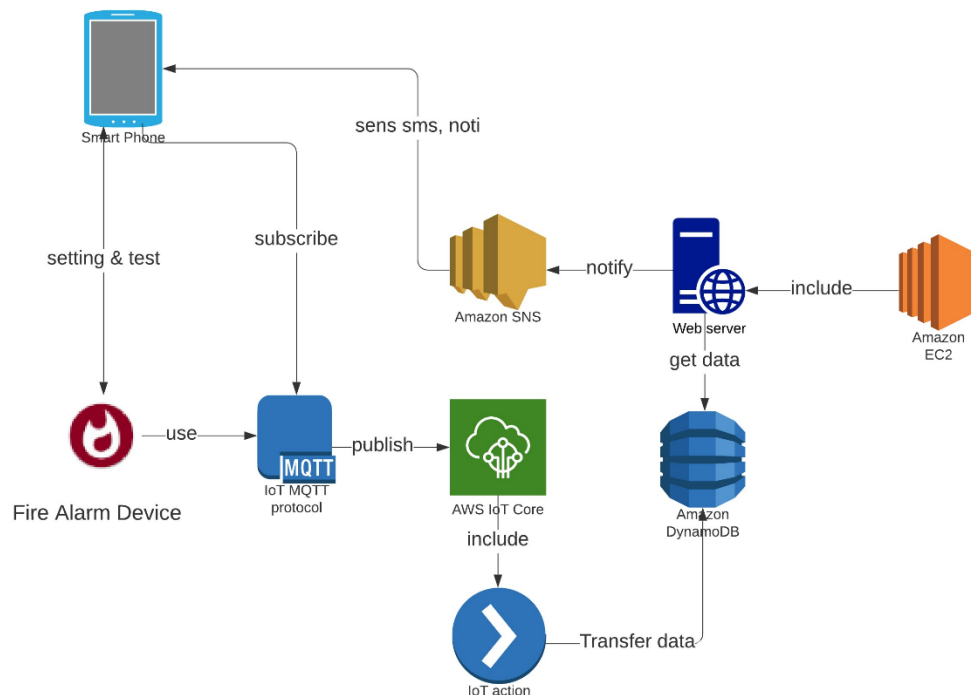
III. Project Structure

Nous avons utilisé quelques services de AWS comme

- AWS EC2 : le container de Backend
- AWS IoT : le service de IoT fournit la méthode MQTT(s'inscrire & publier) de collecter les données de capture.
- AWS DynamoDb : le service est de stocker les données de capture.
- AWS SNS : le service est d'envoyer un message au téléphone.

Fire Alarm _ Project Structure

minhhaok35 | April 21, 2020



D'abord, le dispositif peut publier les données au AWS IoT par méthode MQTT.

Après, AWS IoT les transfèrera au AWS DynamoDb par IoT Action pour les stocker dans AWS DynamoDb.

De plus, Amazon EC est un service de fournir la machines virtuelle avec un IP publique. On l'utilise afin de déployer notre Webserver. Webserver en Nodejs peut récupérer les données et calculer les chiffres statistiques de la température.

En fin, AWS SNS fournit le service d'envoyer des messages, de publier des notifications au téléphone. Webserver peut l'utiliser pour envoyer des messages ou des notifications au client qui a un problème de gaz.

Afin d'éviter la douleur de l'argent (chez un étudiant), AWS EC2 est éteint tous les soirs.

IV. La réalisation de l'appareil

Les matériels nécessaires que nous avons utilisé pour le l'appareil

- Arduino UNO R3 : Micro controller
- Capture MQ2 : Détecteur de gaz, de fumée
- LED : Feu d'alerte
- Buzzer : Parleur d'alerte
- ESP32 Dev Kit : Connecter entre Wifi, Bluetooth et le dispositif
- SIM800L : Envoyer une alerte au téléphone
- Thermistor : Capture de température
- Potentiometer : Diminuer l'électrice pour adapter SIM800L
- Expansion Board, Resistor, Wire.

Les communications entre les parties :

- ESP32 – Arduino UNO : Serial
- SIM800L – Arduino UNO : SerialSoftware
- MQ2 – Arduino : Analog, Digital - MQ2 a une porte Analog (détecter la fumée) et une porte Digital (détecter le gaz)
- Thermistor – Arduino : Analog

Chez SIM800L, La tension de fonctionnement de la puce est comprise entre 3,4V et 4,4V. Alors nous avons utilisé Potentiometer pour assurer le voltage est 4.1V (parfait pour SIM800L).

Nous mettons les matériels de produit sur un sketch en Fritzing

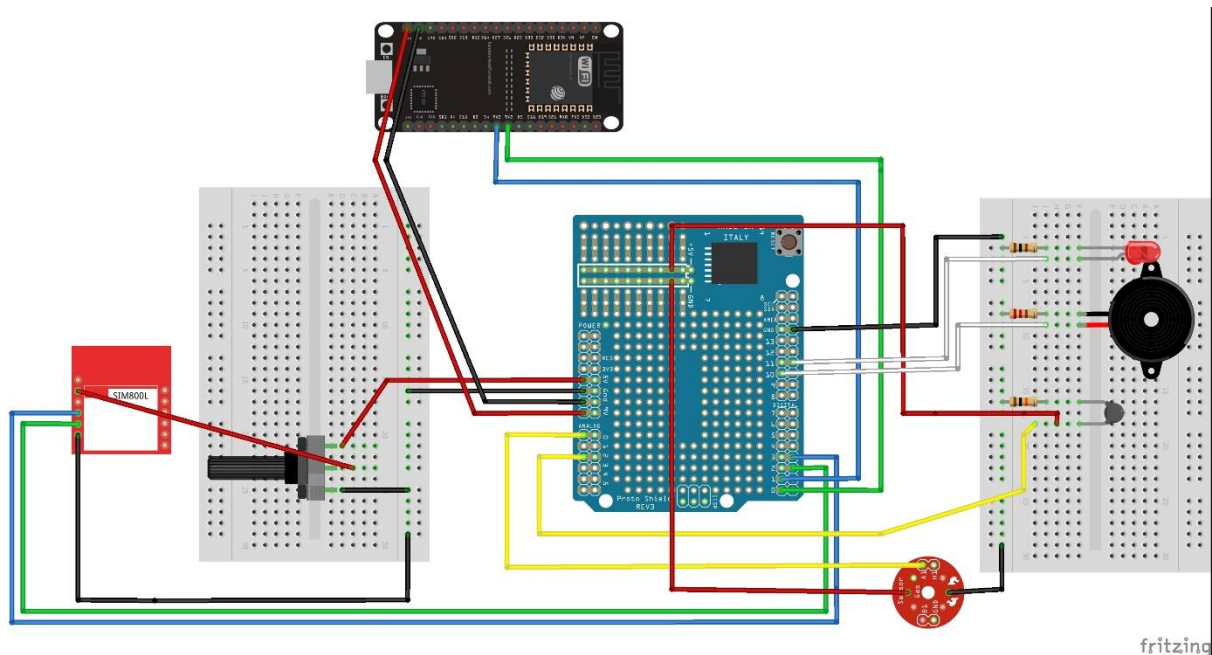
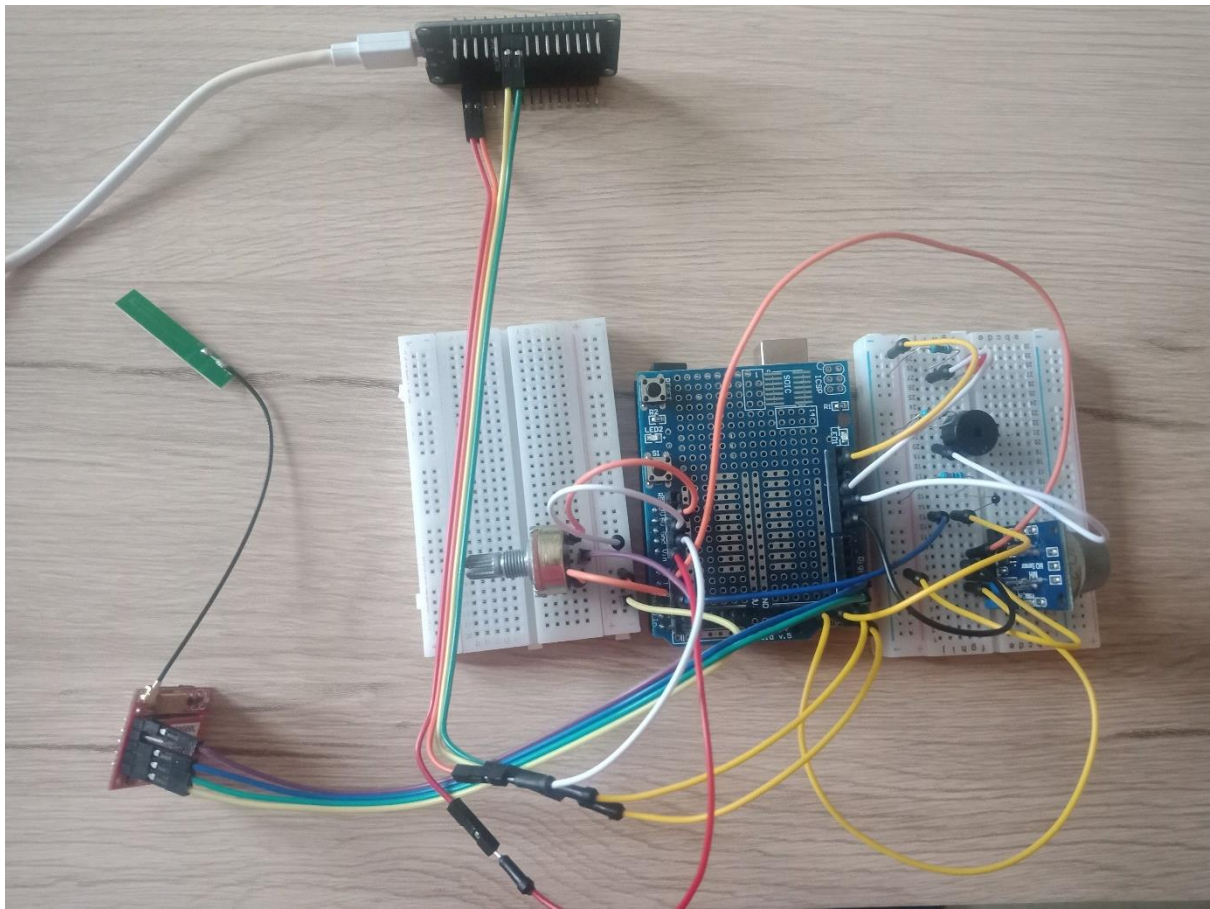


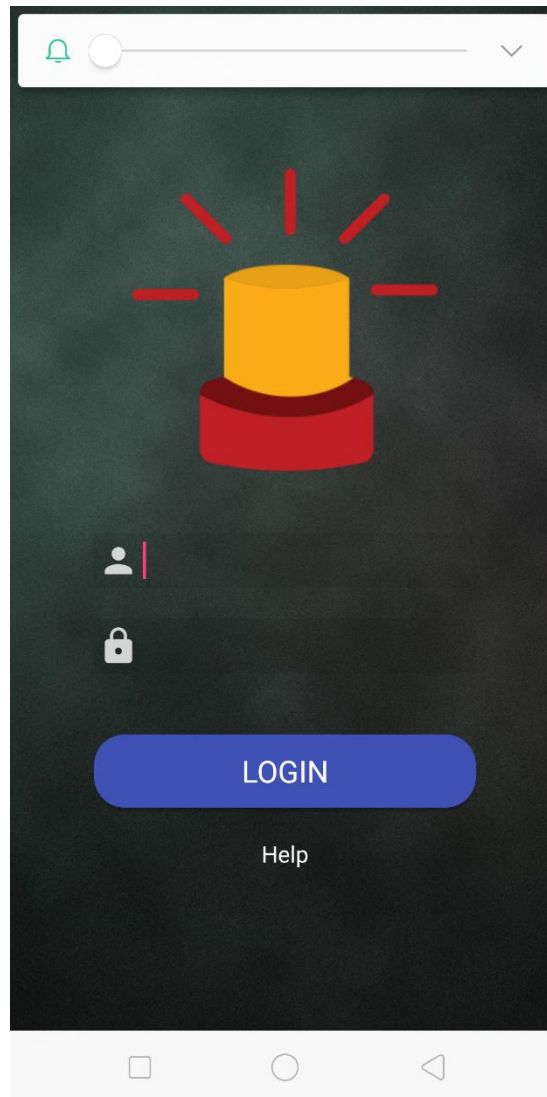
Photo réelle du produit



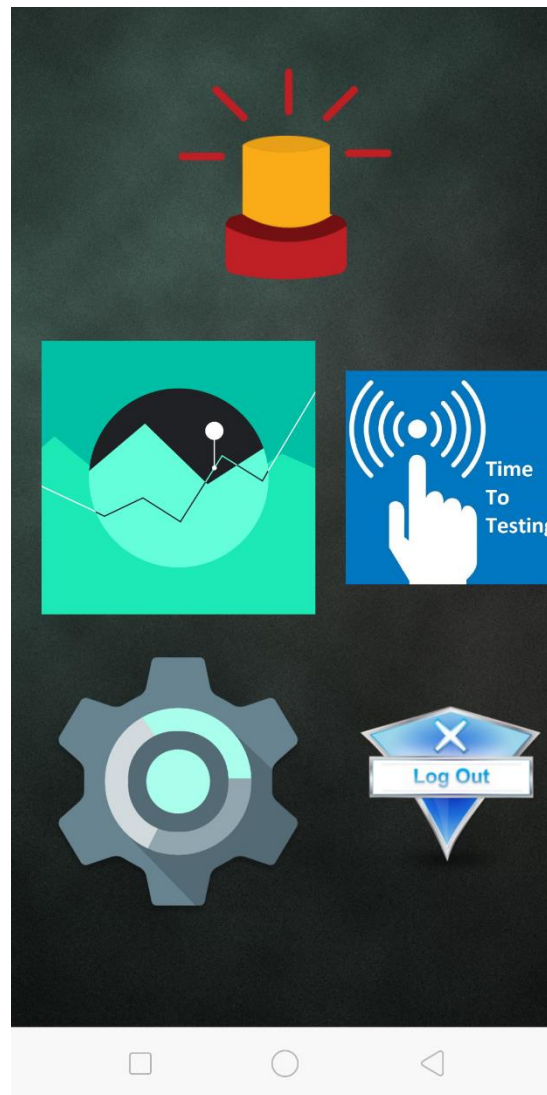
V. L'application mobile

Il y a quatre fonctionnalités dans l'application mobile :

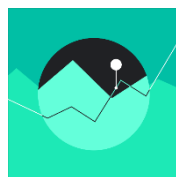
- Se connecter
- Configurer les paramètres (Wifi, numéro de téléphone d'alerte)
- Regarder les données en temps réel
- Tester l'alerte du dispositif (LED & Buzzer, SMS, Notification)



Après le client se connecte par le compte de device, c'est l'écran d'accueil



Quand le client clique cette image :

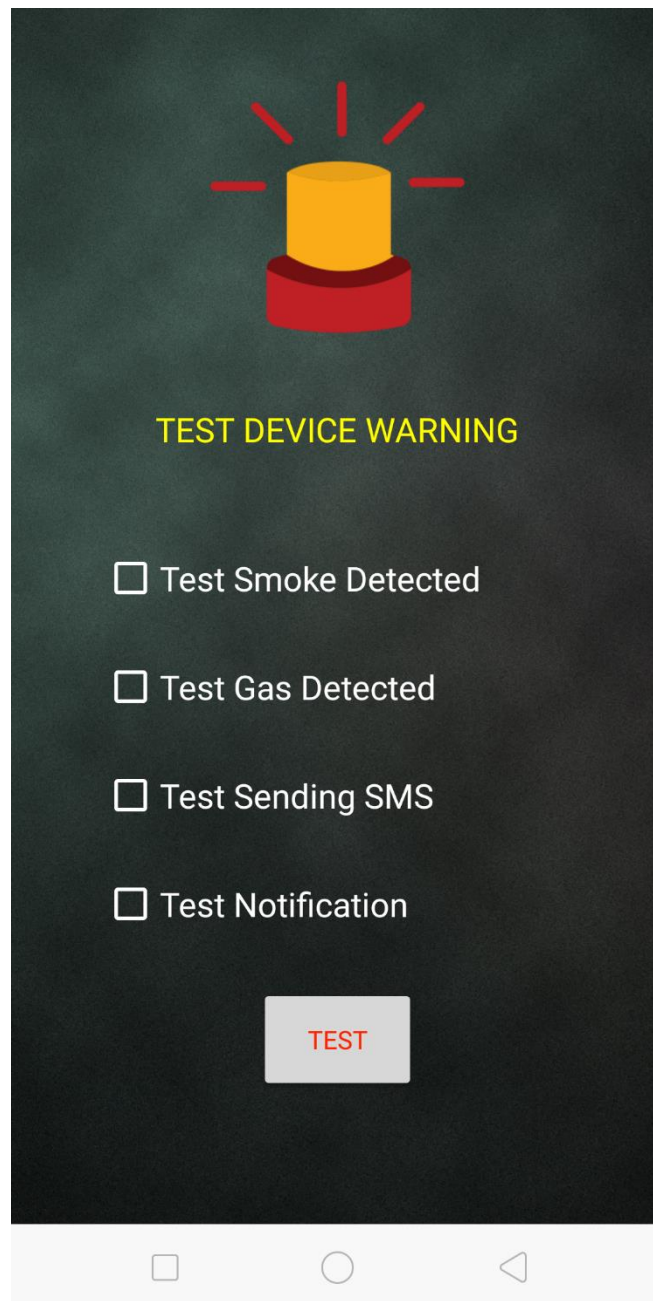


Le client peut regarder les données de dispositif en temps réel. L'application va s'inscrire un topic de AWS IoT pour récupérer les données dans le temps de regarder. Après, l'application va supprimer l'inscription.

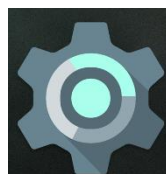
Quand le client clique cette image :



Le client peut tester la fonctionnalité principale de l'appareil

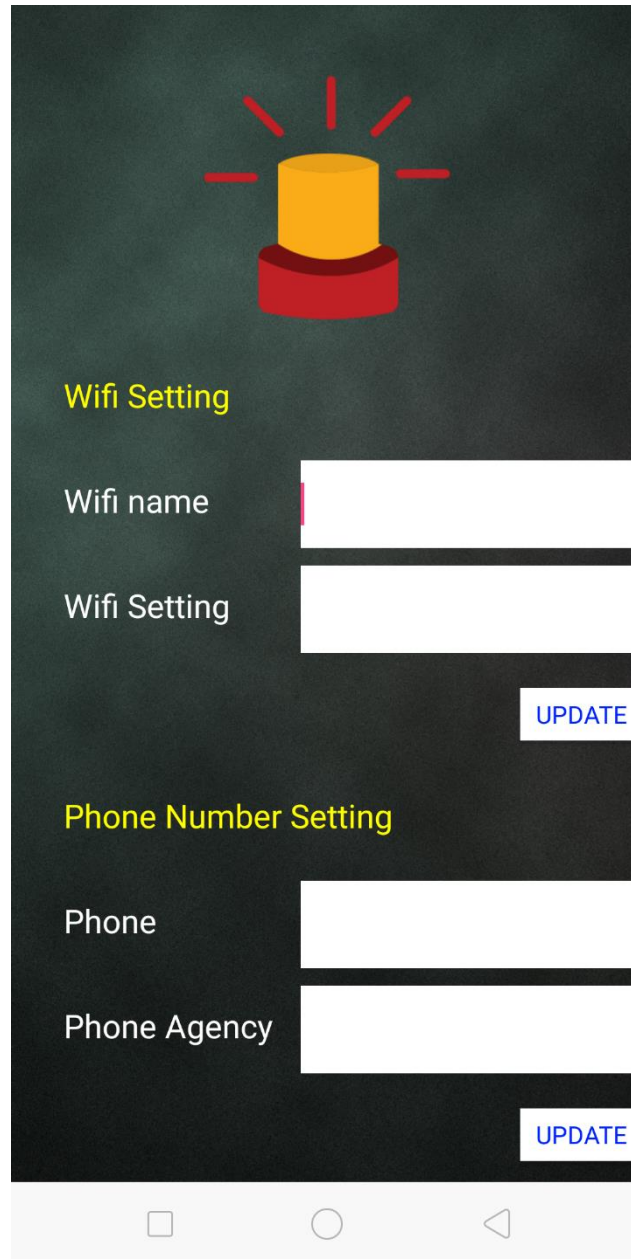


Quand le client clique l'image :



Le client peut configurer les paramètres comme WiFi, le numéro de téléphone d'alerte. Ces données sont aussi stockées dans EEPROM de ESP32 pour récupérer après un reboot.

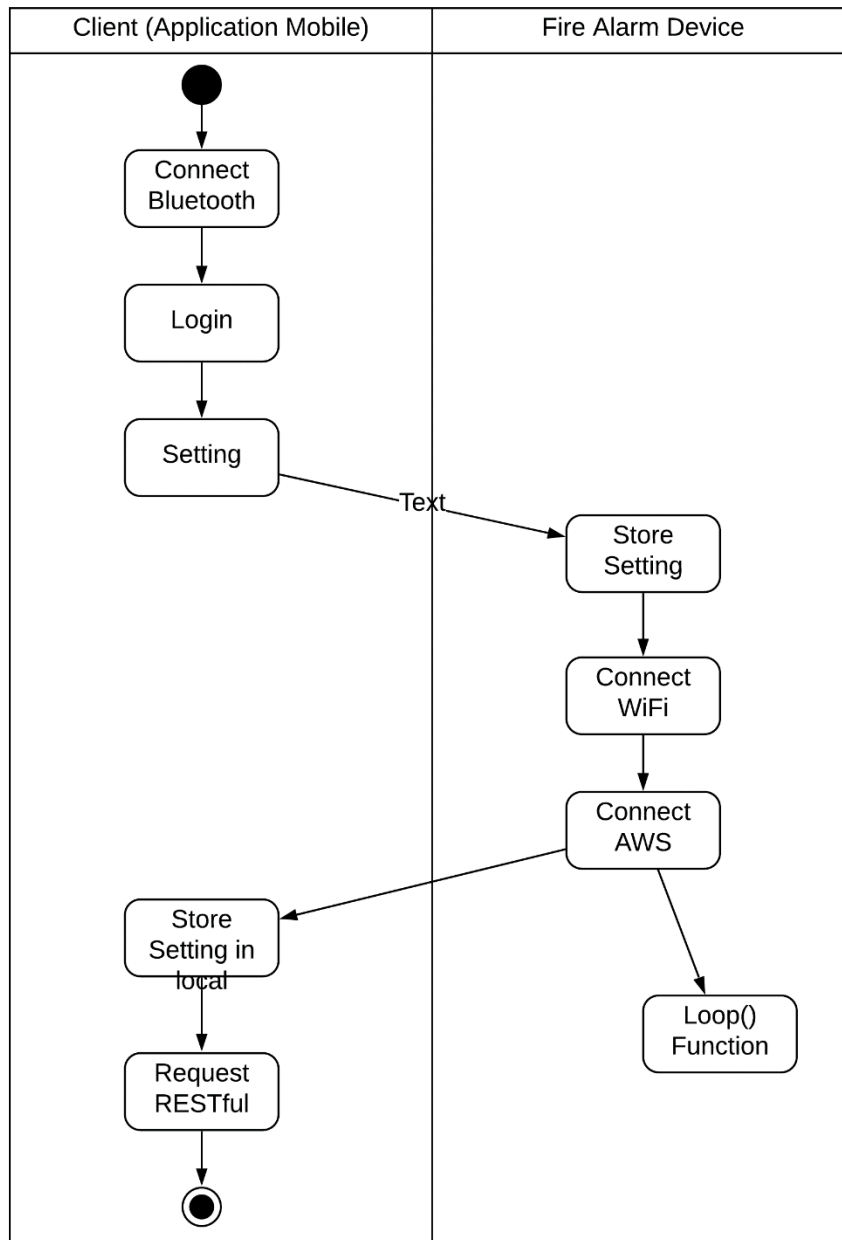
L'écran de configurer est apparu :



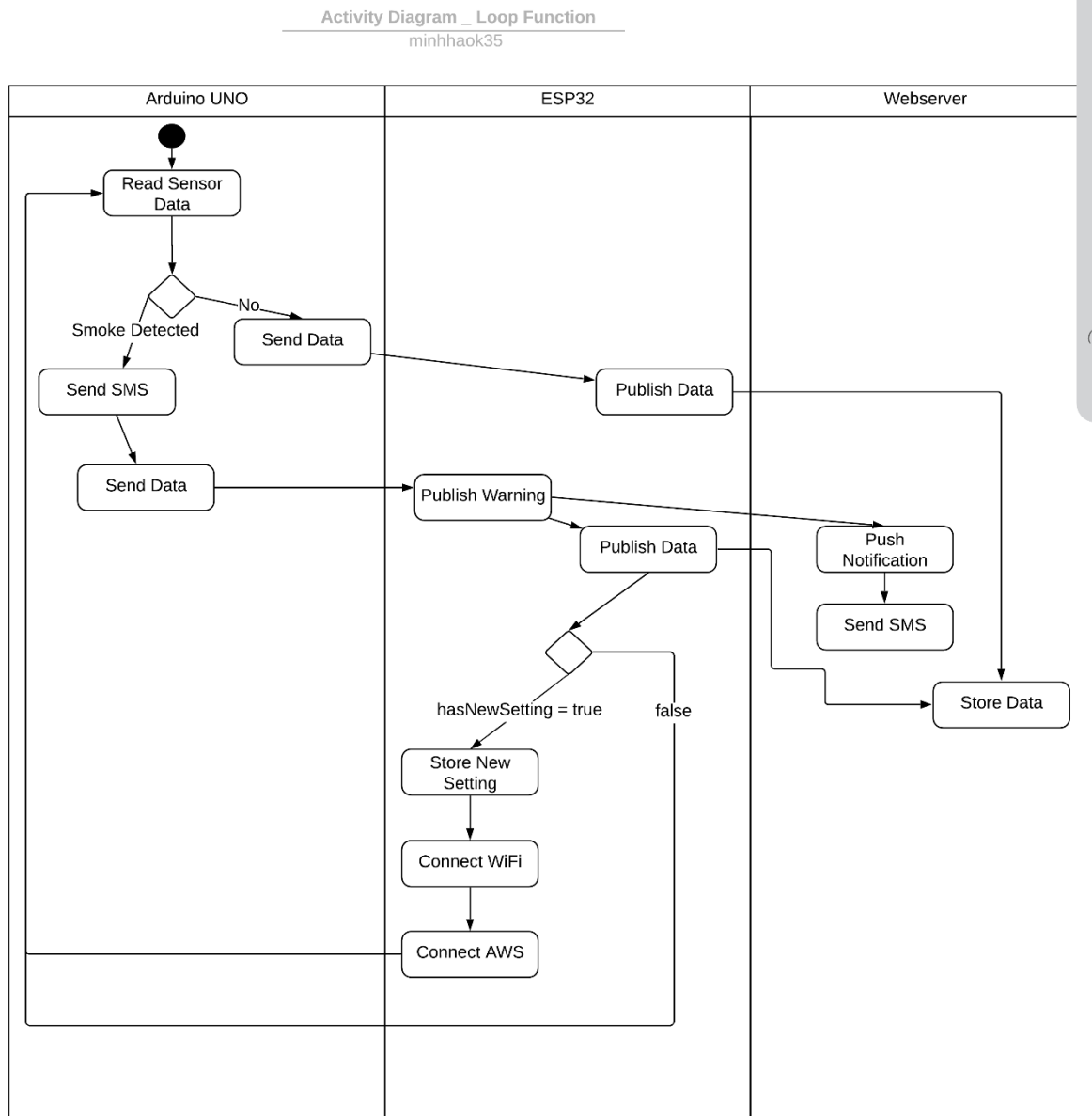
The screenshot shows a mobile application interface for configuration. At the top, there is a yellow cylindrical light icon with red rays emanating from it. Below the icon, the text "Wifi Setting" is displayed in yellow. Underneath, there are two white input fields: the first is labeled "Wifi name" and the second is labeled "Wifi Setting". To the right of the second input field is a blue button labeled "UPDATE". Below this section, the text "Phone Number Setting" is displayed in yellow. Underneath, there are two more white input fields: the first is labeled "Phone" and the second is labeled "Phone Agency". To the right of the second input field is another blue button labeled "UPDATE". At the very bottom of the screen, there is a white bar containing three standard Android navigation icons: a square, a circle, and a triangle.

VI. Diagramme d'activité :

C'est le diagramme d'activité de la première connexion. Le client doit configurer les informations pour que la fonctionnalité d'envoyer sms peut bien fonctionner.



C'est le diagramme d'activite de fonction LOOP dans le produit



VII. Conclusion

Pendant le développement du produit, nous avons rencontré un manque de matériels comme Wire, BME280. Alors c'est le premier prototype incomplet mais les fonctionnalités bases sont développées

Dans la future, après bien développer ces fonctionnalités. Nous allons développer les autres fonctionnalités suivant :

- Multi-language: L'application mobile a plusieurs langues et la forme SMS aussi
- Régions and Locations: Localiser l'emplacement de l'appareil et proposer le numéro d'agence de pompiers
- Collecter et connecter l'agence de pompiers