SUP'COM

المدرسة العليا للمواصلات بتونس

ÉCOLE SUPÉRIEURE DES COMMUNICATIONS DE TUNIS

Scope Statement

Development of a mobile application supervises an intelligent alarm system

Realised By :

Malek FAKHFAKH

Nesrine KHAMMAR

Supervised By :

Dr. Mohamed Bécha KAANICHE

Academic year: 2021/2022

Table of Contents

Table of Figures

1. Introduction

In recent years and with great rapidity, intelligent systems continue to develop and believe. Which caused a kind of competition and improvement in terms of quantity and quality.
Currently, the trend in technologies is to use wireless techniques such as the Internet of Things which has been integrated into several fields such as: the military, industry, agriculture, etc.
Internet of Things is a very broad and rich term. It makes us imagine a whole world that is connected and can be communicated through the exchange of information between its objects.
In the field of security, protecting a home from theft or intrusion attempts is essential. But existing alarm systems are not precise. Because they often trigger false alarms following a gust of wind or following an animal passing.

Based on the previous elements, we can formulate the main objective of our work, which is the combination of the latest developments in the market of mobile technology, persons detection, face recognition.
The goal is to create a continuous and efficient video surveillance system. Our approach is based on the use of smartphones, with devices such as (camera, alarm, Raspberry Pi board..) as additional peripherals.

Thus, the main architecture of our work consists of a part in which we establish a reliable connection between the mobile device and the camera to collect continuous data. In another part, we analyze the data using different kinds of data analysis techniques and trigger an alert in case of danger.

2. Context

Figure 1: System modeling

The video surveillance system will be connected to a mobile application, from which the administrator is able to add easily images of people (family members, friends, etc.). They will be saved in the database.

Note that our intelligent video surveillance system is based on the persons detection, image and video analysis. It detects any foreign individual (his face does not exist in the database already populated by the administrator).
As a result, the owner of the house can consult the alerts generated, the video surveillance in real time, through the mentioned mobile application, etc.

3. Features

As already mentioned, the video surveillance system will be connected to a mobile application. This app will cover the following points :

Security :
Internet security is becoming an ever-increasing issue. Our application is storing personal and sensitive information so security is an absolute must.

Add person's picture to database :
The administrator is able to easily upload the images of the people he wants the surveillance camera to recognize them, through an interface within our application.

Push notifications :
To alert the administrator once an alarm is triggered, push notifications are more attractive than emails and less likely to end up in a spam folder.

4. Conception

The following image summarizes the architecture of our application.
Starting with the **Front-End** part, It communicates with the **middleware server** through HTTP requests. So that JSON is the format of the data exchanged between this two parts.

Also we will use Websockets for the notification push feature.

On the other hand, the **middleware** server communicates with the **Raspberry Pi board** throw MQTT protocol

Regarding data, they are stored in a NoSQL database : MongoDB.

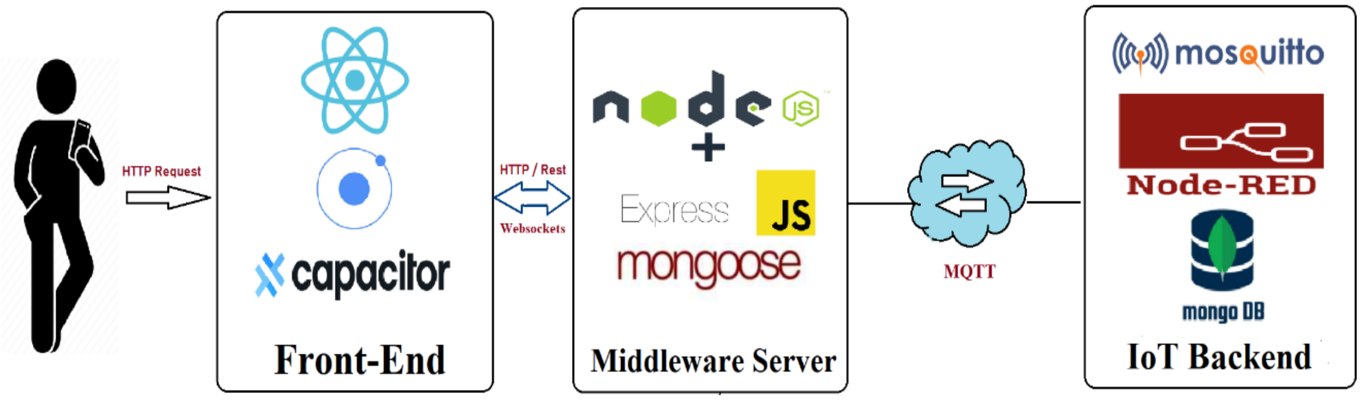

Figure 2: Global architecture of the mobile application

5. Technologies

Front-End :

For the Front-End part we will use the trio :

- **Ionic :**

Figure 3: Ionic Framework logo

Ionic Framework is an open source UI toolkit for building performant, high-quality mobile and desktop apps using web technologies (HTML, CSS, JavaScript) with integrations for popular frameworks like Angular, React, and Vue.

We have chosen Ionic because it is open source and it has a big and helpful community.

- ✦ **<u>React :</u>**

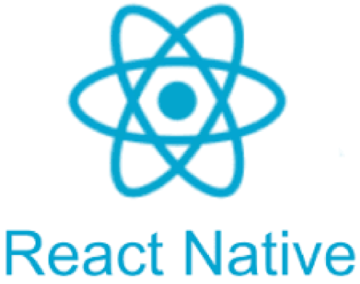

Figure 4: React logo

React Native is a framework for building native applications using JavaScript. React Native compiles to native app components, which makes it possible for you to build native mobile applications.

We decided to work with React Native to be able to realize the push notification functionality. And because it is gaining popularity, gaining community support, and gaining more market share. So it's an opportunity to work with this framework.

- ✦ **<u>Capacitor :</u>**

Figure 5: Capacitor logo

One of the limiting factors of the "hybrid" applications is having the majority of your application and user interfaces running inside of a web view embedded in a native application.
Capacitor is aiming to make the process of integrating native code into your Ionic projects a little more approachable.
So with Ionic Capacitor, adding native functionality to our apps can be easily done using a simple Plugin API.

We have chosen Ionic Capacitor to access the native features of react. And because it is the only native runtime to deliver first-class support for web apps and PWA Progressive Web Apps.

Middleware Server :

- **Node.js**

Figure 6: NodeJs logo

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment. It is used in particular as a web server platform.

Thanks to how easy it is to create an API in Node and how quick to start development, we have chosen Node.js

IoT Backend :

- **NoSQL Database :**

Figure 7: mongoDB logo

MongoDB is a source-available, cross-platform , document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

We have chosen mongoDB because it is the easiest NoSQL database

✦ Node-RED :

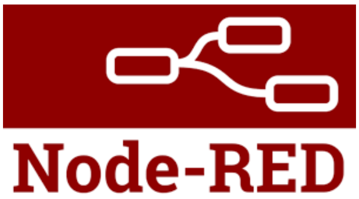

Figure 8: Node-RED logo

Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.

It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single-click.

✦ Mosquitto:

Figure 9: Eclipse Mosquitto logo

In MQTT protocol there are two major entities: a broker and clients. Eclipse Mosquitto is an open source message broker that implements the MQTT protocol. It is suitable for all situations from full power machines to embedded and low power machines.

We have chosen the Mosquitto MQTT broker since it supports SSL protocol, It supports webSocket. Its also managed, self hosted with a static IP and the installation process is not difficult.