# Smart Room Project

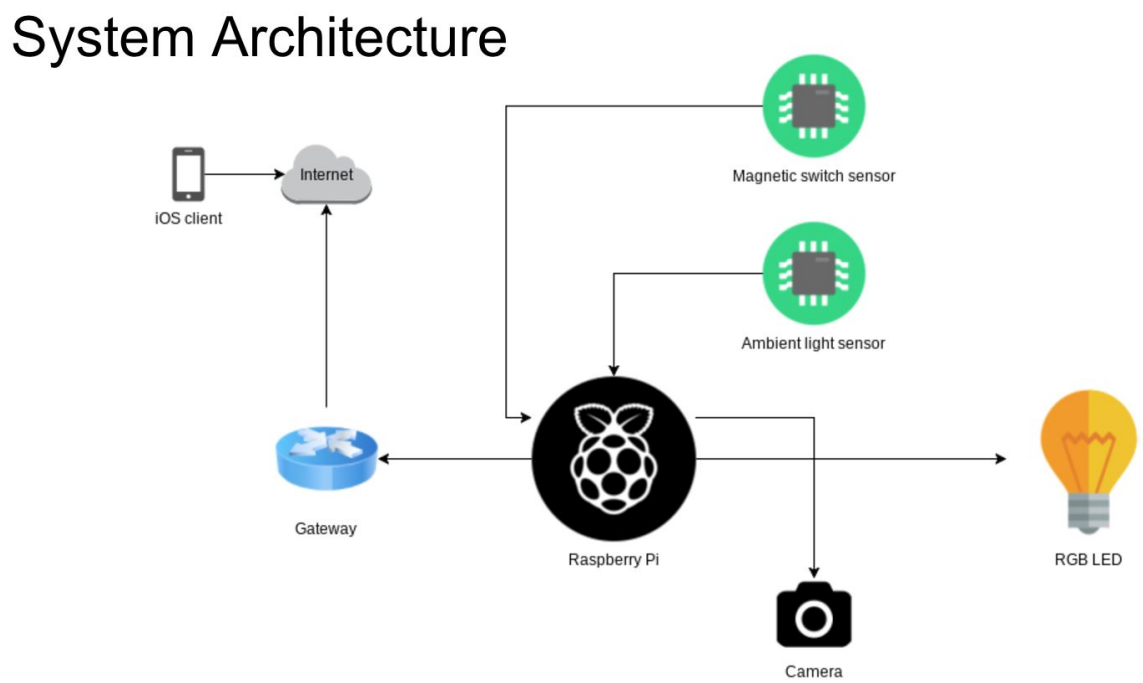
Students:
Bruno Mendivez (26874229)
Feixiang Li (26669900)

## Project description and features

The purpose of this project is to make a try to provide a comfortable living environment for the people. For instance, a function what we try to do in this project is using the RGB LED to automatically or manually adjust the brightness and colour of light so that the adjustment will give the resident some information. In general, we can use the mobile phone to set the bedtime with the clock, but sometimes the phone may be left at the bedroom or somewhere so that we can't hear it when we are at other room for the entertainment. Therefore, we design the pattern that the user can set a profile in app to specify the exact time of going to bed, then at that time the lightbulb will change its brightness or colour to tell the resident that it is time to go to bed. In addition, this project provides some more interesting, constructive and useful utilities as the following:

- Progressively dim the lights until desired level based on ambient light sensor.
- Progressively dim the lights based on schedule.
- Turn on (to current brightness) and off when enter or leave the room.
- Surveillance mode enabled when leaving the premises or when selected.
- Capture and delivery of snapshots when detection of intrusion.
- Alerts as push notifications when detection of intruders.
- Manual modes for both subsystems.

# IoT System Architecture Details




*ture as a whole*

For this project system, normally there are 4 parts ---the **sensors** (ambient light, magnetic switch), **actuators** (RGB led and webcam), **controller** (Raspberry Pi as also the **server**) and the **iOS client** (phone) for controlling the lights by user and presenting intrusion events (e.g the intrusion photo taken by camera) to user. Next, these parts will be introduced respectively.

## Devices

- **ADC**. Analog to digital converter. Used to represent numerically the analog signal coming from the ambient light sensor. It translates the current from the sensor to a number proportional to the amount of light it senses. The value varies from 0 to 1024 (10 bit resolution).

- **FULL COLOUR LED LAMP**
  The colour LED is used to simulate the lights installed in the resident room.

- **OPICTM Linear Output Ambient Light Sensor**
  The sensor is used to simulate the brightness. It can change its state in response to the setting of client profile. The analog signal it outputs is current.

- **Magnetic Reed Switch**
  The magnetic switch is used to simulate opening/ closing door. When the unexpected situation happens, the magnetic switch will automatically trigger the camera to take a photo to record, and the photo will be sent to the client.

- **Webcam**
  Used to take pictures when there is an intrusion event detected.
  The pictures are then sent to the client to verify them.

- **Raspberry Pi**
  The controller of the system. It's the responsible to handle the control loops defined by the architecture. Specifically, the control of the RGB led that depends on the ambient light sensor. It also controls the dimming and changing of color of the led done manually. Finally, this device also handles the communication with the iOS client through a REST API (for the retrieval of surveillance information) and an MQTT server (for the live exchange of messages regarding the state of the sensors and actuators).

- **iOS Client**
  User interface:

1. Home: Where the user can control the lights and surveillance system by choosing a light profile or handling the brightness and colour manually.
2. Profile Manager: Where the user can create and edit light profiles that specify the following: colour, if the light activates when the door open/closes, to which ambient light level the LED activates, the maximum brightness, the hour on which the light turns off and the time it takes to reach its maximum brightness.
3. Intrusion event log. Where the user can check all the intrusion events as they occur. The user can check images from each event to verify who's the intruder.

## Software libraries

1. **MQTT**. The iOS app makes use of the MQTT protocol to communicate with the Raspberry Pi and control both the sensor and actuators. When these devices change their states, real time communication takes place to update the visuals in the app. The library used was SwiftMQTT (https://github.com/aciidb0mb3r/SwiftMQTT). It includes the basic functionality to connecto to the broker, subscribe, publish and asynchronously obtain messages.
2. **Pi-blaster**. A PWM daemon for the Raspberry Pi. When dimming the light and changing the colour, we are varying the voltage of each LED (red, green and blue). A PWM pulse is useful for setting the intensity of each led from 0 to 1 . Ref: https://github.com/sarfata/pi-blaster

3. **Pi-blaster.js**. An API to control the daemon. It allows to vary the duty cycle of a pin from 0 to 1. Ref: https://github.com/sarfata/pi-blaster.js
4. **GPIO and SPI control for the Raspberry Pi**. A useful API for Node JS that allows to poll individual GPIO pins to sense an input. It also allows to read from SPI ports (like the one that requires the ADC MCP 3008 used in the project). Ref: https://github.com/jperkin/node-rpio
5. **MQTT broker for the Raspberry Pi**. The broker used was Mosquitto. The broker handles the subscriptions and MQTT message exchange between clients and the Raspberry Pi. Ref: http://www.switchdoc.com/2016/02/tutorial-installing-and-testing-mosquitto-mqtt-on-raspberry-pi/
6. **Node-cron**. In order to schedule the automatic dimming of light based on some timer or a predefined hour of the day, we use the cron library for Node JS. It allows to execute functions in specific periods of time. Ref: https://www.npmjs.com/package/node-cron
7. **Executing shell commands in Node JS**. Useful for activating the camera when an intrusion event has been detected. The operating system provides with a shell command to take pictures with any USB camera. In order to execute those commands from a Node thread we use this library. Ref: https://www.npmjs.com/package/node-cmd