**Preface:**

*We wanted to build a system that is reliable and secure enough to protect a company from intruders. We wanted to alert the manager and the police and get the intruders recorded. The intruders must be identified, the system must identify whether they are workers, servants, or animals.*

**Requirements specification:**

**Functional Requirements**

|  |  |
| --- | --- |
| **REQUIREMENTS** | **COMMENTS** |
| *Motion sensor will detect motion* |  |
| *Raspberry pi 3 is the main controller that is able to send messages to the Arduino and the camera* | *There is another controller that will control the GSM Module* |
| *The camera module will stream video in local area and will record video for 20 sec and mail it through pi 3 that is connected to the network* |  |
| *GSM module will receive signals from the Arduino that had received a signal from pi3 to call the police and the manager* |  |
| *GSM can receive any message from external source, thus the pi3 will control the circuit depending on the message* | *There was a problem in identifying the message coming from the external source using python but it was very easy using Arduino* |
| *the pi3 will give the intruder 5sec to identify his face if he didn’t the alarm will go up and recording starts* |  |
| *If needed when opening the case, the customer could enter any number for the* GSM *to call and to identify if there is any problem with the* GSM *or not using lcd and a keypad* | *Requirements testing* |
| *The Arduino Uno board has one UART that you can use either with a USB cable or from the RX/TX pins* | *Asynchronous serial communication using UART with Arduino and PI3b with USB cable* |

**Non-Functional Requirements**

*Product req:*

1.the GSM module is around 2A during transmission burst, and requires from 3.7 to 4.2 v

2.we should use a power supply and a mobile charger to deliver power to the boards and a 4v lithium battery for the GSM

3.the system is available 24/7 if connected to the power supply and the battery should be recharged after 24 hours working

4. We cannot directly connect Rx pin on module to Arduino’s digital pin as Arduino Uno uses 5V GPIO whereas the SIM800L module uses 3.3V level logic and **is NOT 5V tolerant**. This means the Tx signal coming from the Arduino Uno must be stepped down to 3.3V so as not to damage the SIM800L module.

5.Arduino output high voltage ranges from 3.3v to 5v and receives from 3v to 5v as high, GSM receives 3.7v and sends 3.7v as output high

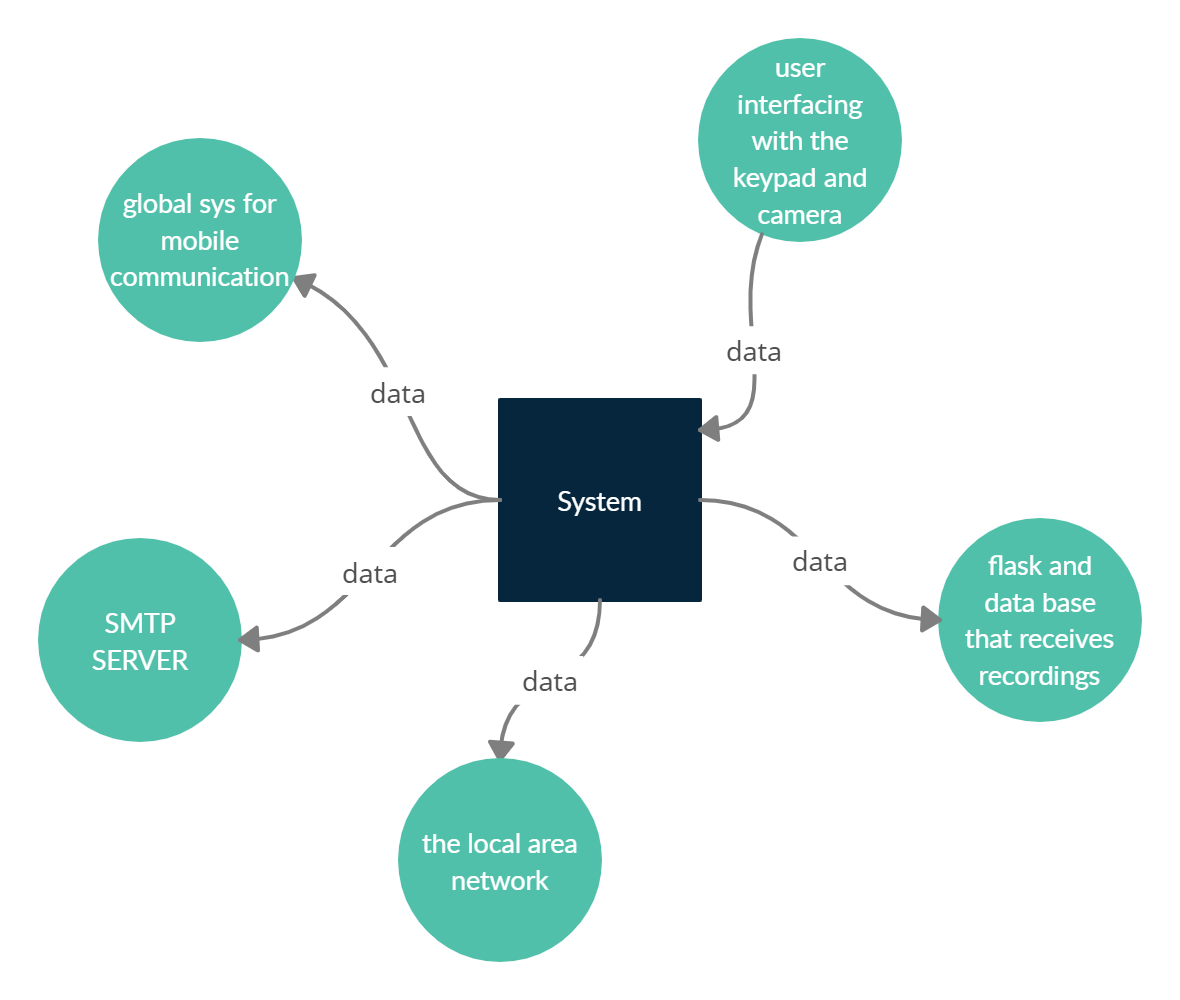
5.the project is using 5v level shifter form the transmitter (GSM) to the Receiver (Arduino)

Organizational requirements:

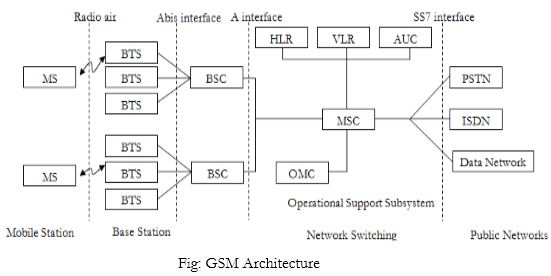
1.we used c for the GSM and python for the rest

2.as soon as the system is supplied with power it starts working and streaming

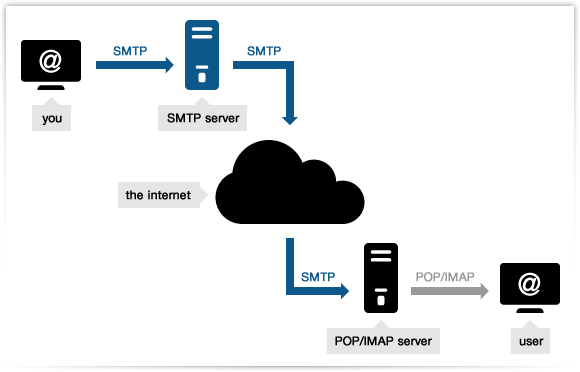
**System Context Model**



**GSM Architecture**



**SMTP Server**

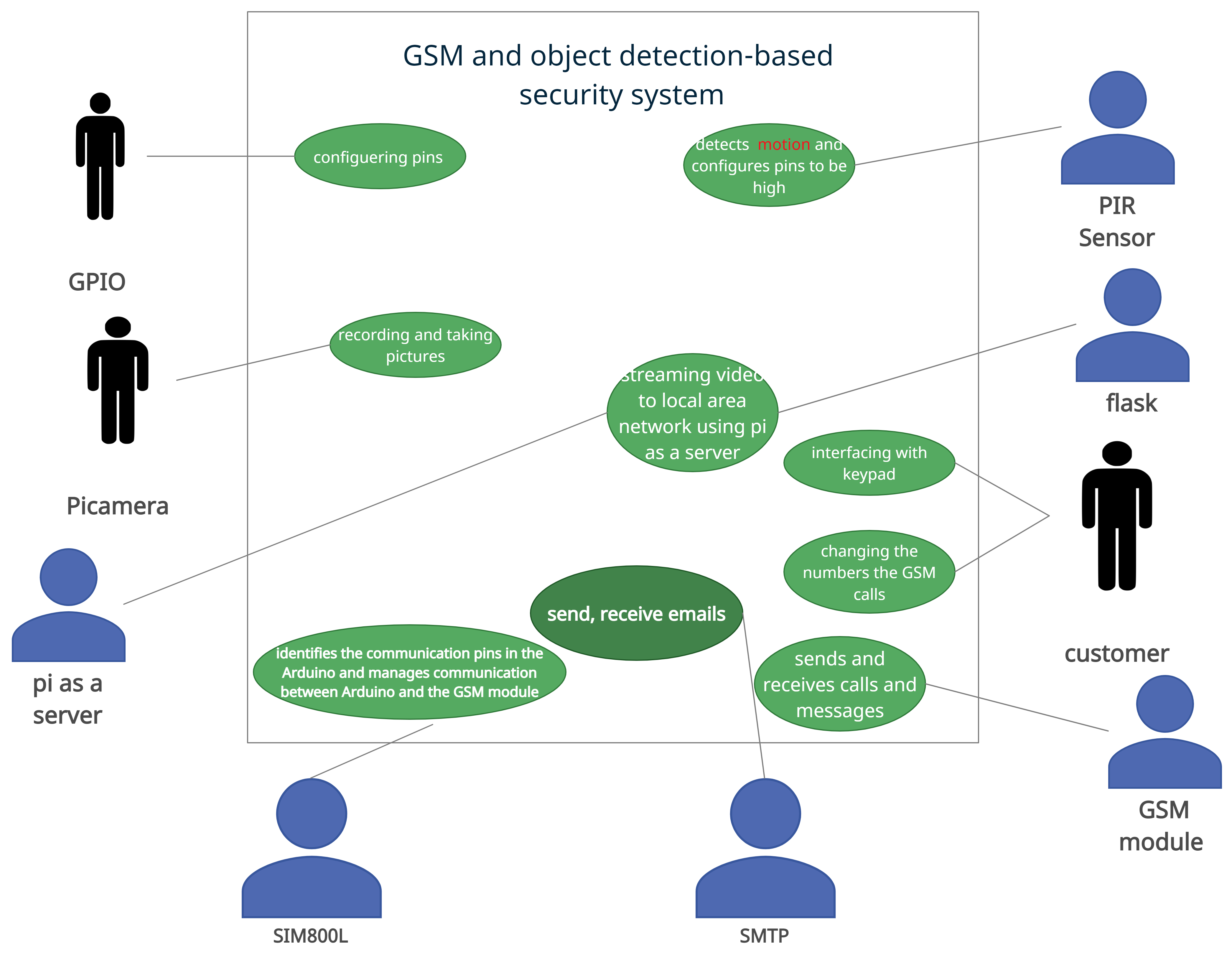


It’s the contraction of **Simple Mail Transfer Protocol**, and defines the method that handles the process of **email exchange and delivery** across Ips or it’s an application used by mail servers to send, receive, and/or relay outgoing mail between email senders and receivers.

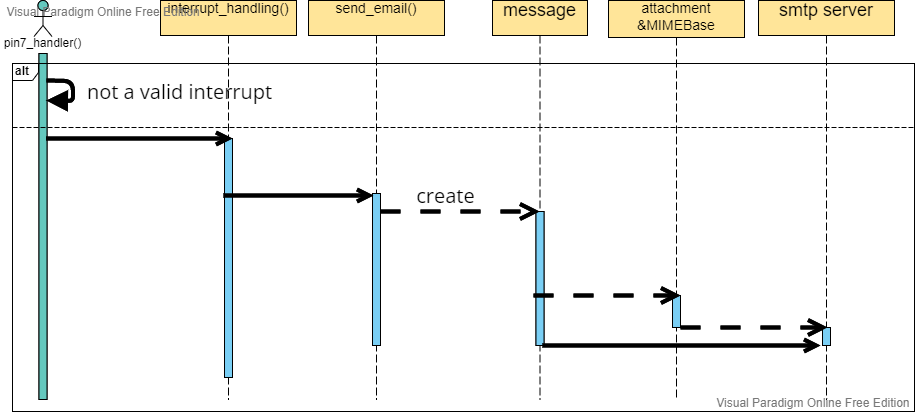
To put it simply, when you send a message to a friend using a mail client, it’s picked by an [**outgoing server**](https://serversmtp.com/en/outgoing-email-server) (called **SMTP server**) which starts a conversation with your friend’s incoming one.

During this process the machines “talk” SMTP: the protocol provides a standard, reliable set of guidelines to make servers identify themselves and communicate to understand who is the sender, who the recipient, where the content must go etc. In a few words, how to properly **deliver the email** (and also take care of possible issues). Any normal email provider (like Gmail or Hotmail) has an associated SMTP server or SMTP host devoted to the message delivery process.

**Use Case Diagram**

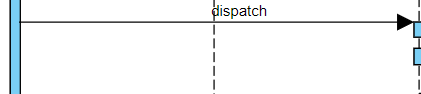


**Sequence Diagram**

****

**There must be a video to be recorded and to be sent if the interrupt is valid but there is no reply just making specific tasks**

When an interrupt occurs pin7\_handler function is called where interrupt is being checked and camera object records some videos and holds name for them(after finishing with camera object), then send\_email function is called where message object holds the body, username, and the recipient, the attachment & MIMEBase objects have the job of holding, encoding and encrypting the attachments (videos or pictures) ,finally after all objects are filled with information the SMTP Server will send the mail.



The arrow means the current function or object has to execute the next function or method to reach the final goal

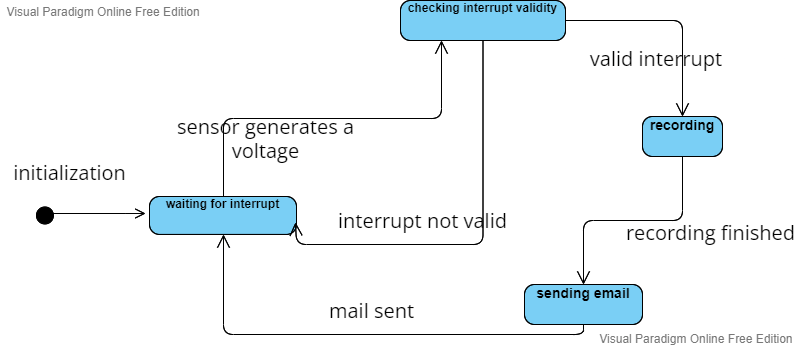
Message object: holds the body, recipient, transmitter and header

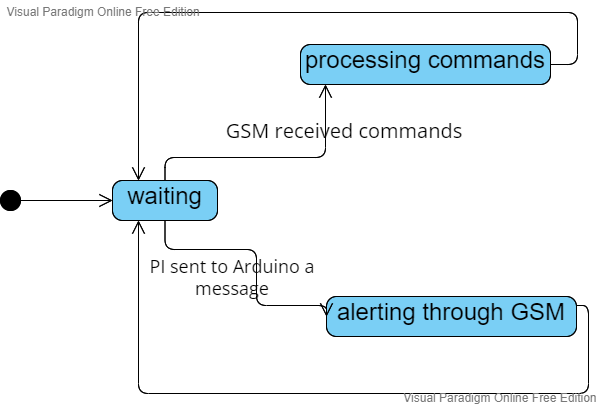
Attachment object: holds the path, the name and the extension of the picture or video

MIMEBase Object: encodes and encrypts the picture or the video

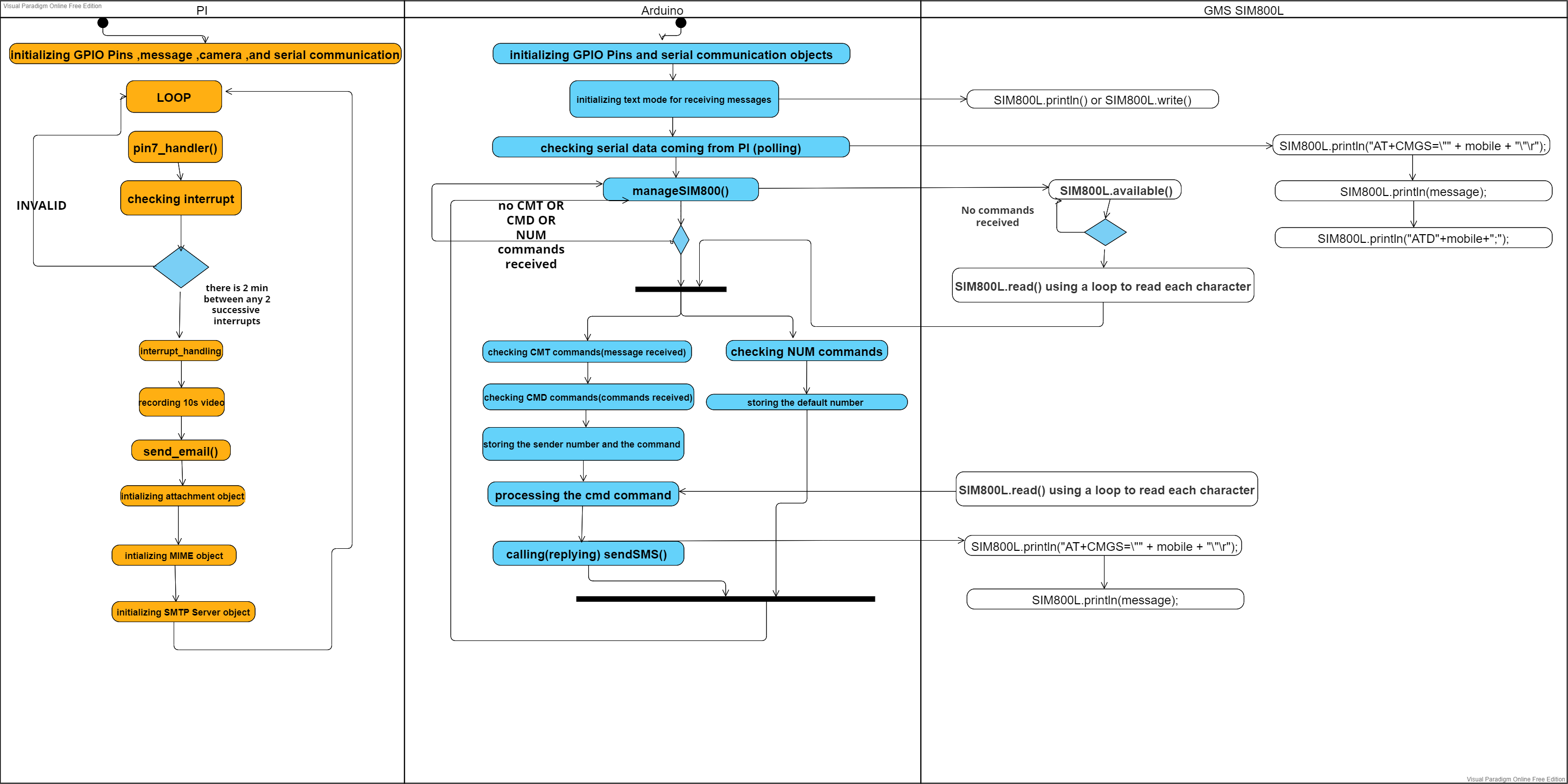
SMTP Server: sends the email to the recipient

**State diagram**



****

**Activity Diagram**

****