Smart Parking - IOT Project

# Abstract :

We successfully implemented an IOT based smart parking system. With the help of individual nodes (proximity sensors) at every parking slot, we can make available the current status of the parking slot – ‘Available’ or ‘Occupied’

# Hardware implementation:

We plan to start with a small-scale implementation of the project i.e. simulate a real-life parking lot on a cardboard.

**Electronic Components:**

1. Raspberry Pi (Main control unit)
2. IR Sensor (Proximity Sensors)
3. RF id Reader
4. RF id Cards

**Why Raspberry Pi?**

In addition hosting the proximity sensor data to the internet we planned to send out mails regarding the in and out time of the parking lot users, hence we decided to use a Raspberry Pi which allows us to interface the sensors via python. Python is preferred for this because of its easy use as well as ability to write modular and object-oriented programs.

Moreover, after loading the Raspbian OS we were able to control it just like a desktop capable of interface sensors using GPIO pins.

# Features included:

1. Host slot availability to internet
2. Dedicated RF-ID to each user (with wallet maintained)
3. Mail based communication (for in/out time and wallet balance)
4. Autonomous registration for new user

# Software Implementation – Abstract:

The project has two different programs running simultaneously -

**1. RF-ID Tagging Module**

This program takes care of authentication of the RF-ID cards. Controls the micro servo motor (acts as a gate) and logs in/out time. This is the program that sends out mails based on the total time the user spends in the Parking lot. The customer will have to interact with this program and hence ease of use along with clarity of information was given importance.

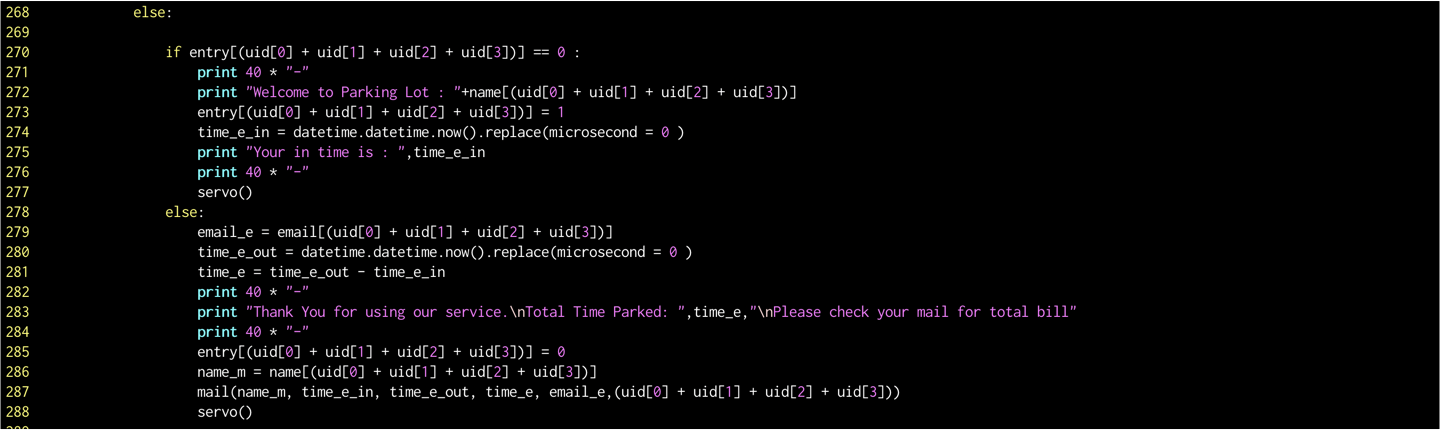
**2. Proximity Sensors Module**

This program reflects the current status of the sensors – ‘high’ or ‘low’. These sensors reflect the slot availability – ‘Available’ or ‘Occupied’. The Output is then dumped onto a text file, which is updated every second using the same python script. Furthermore, a HTML file reads the data from the text file and displays it onto the webpage. We then host the website using a hosting service called ‘ngrok’. Hence the server contains information about the availability status of the respective parking slots.

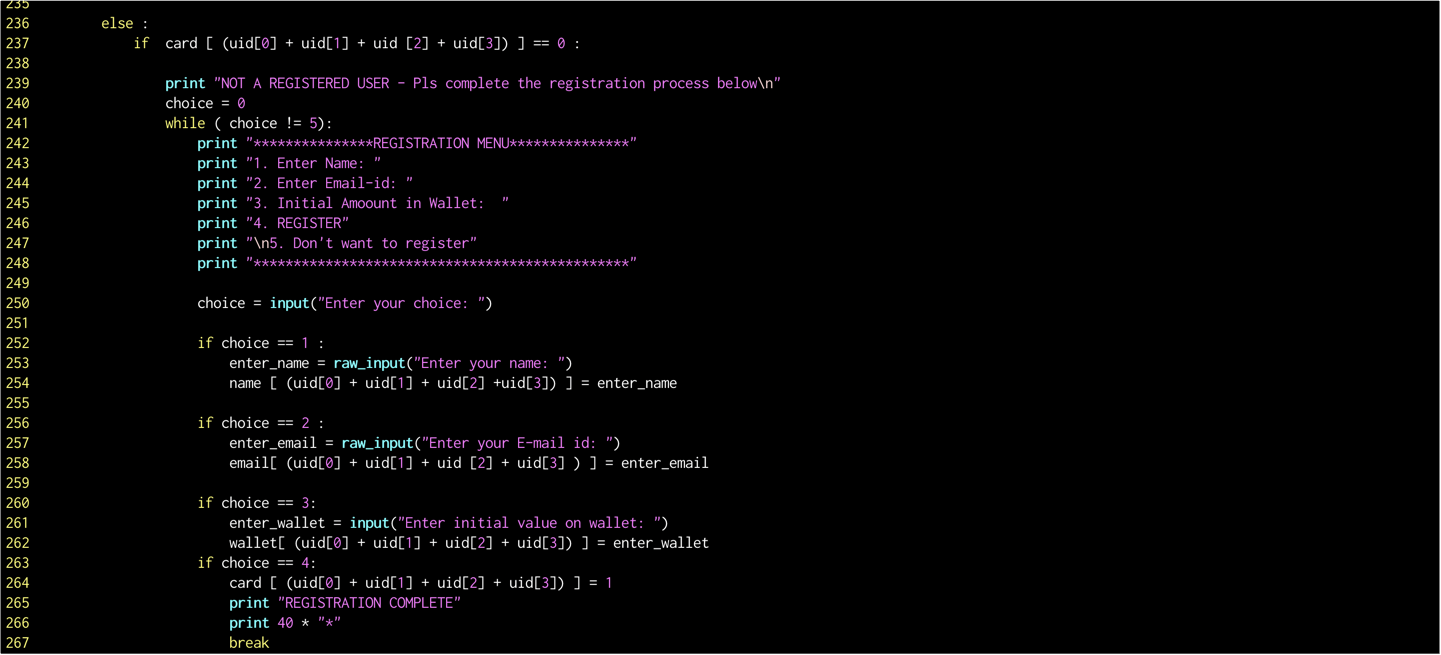
# Software Implementation - Code Snippets:

**RF-ID:** Accessing name, email-id and wallet amount corresponding to RF-ID number. In case the card hasn’t been resisted, a menu driven program runs to enter values of name, email id and wallet amount to corresponding RF-id cards.

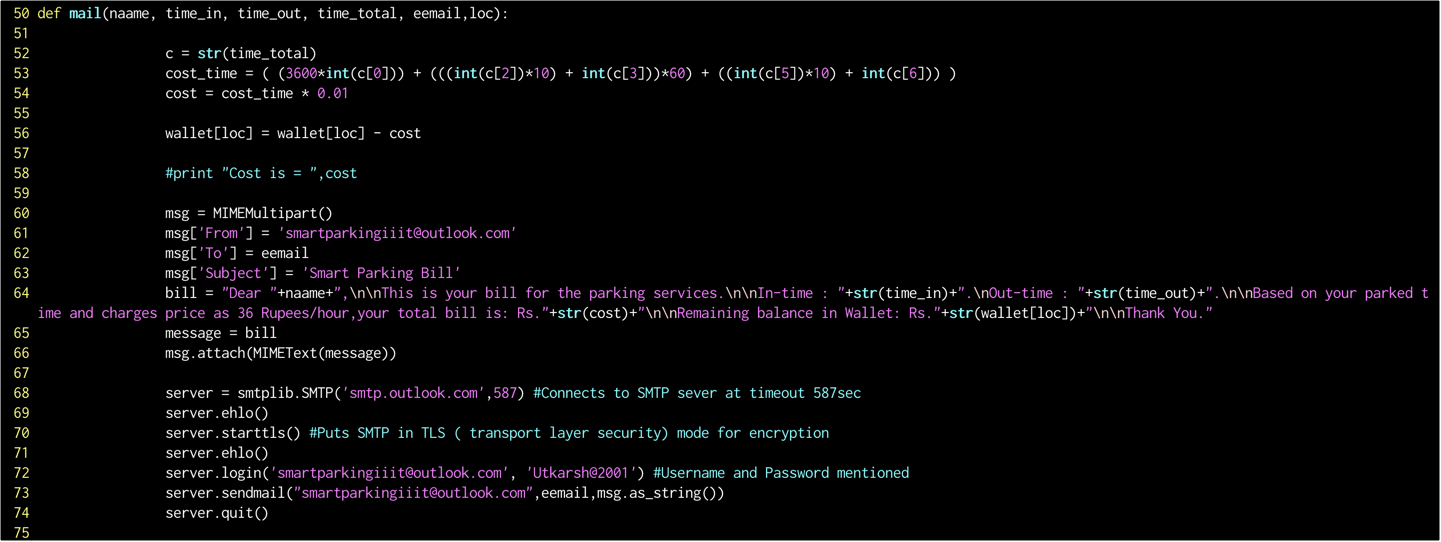
**If a registered user scans card:**



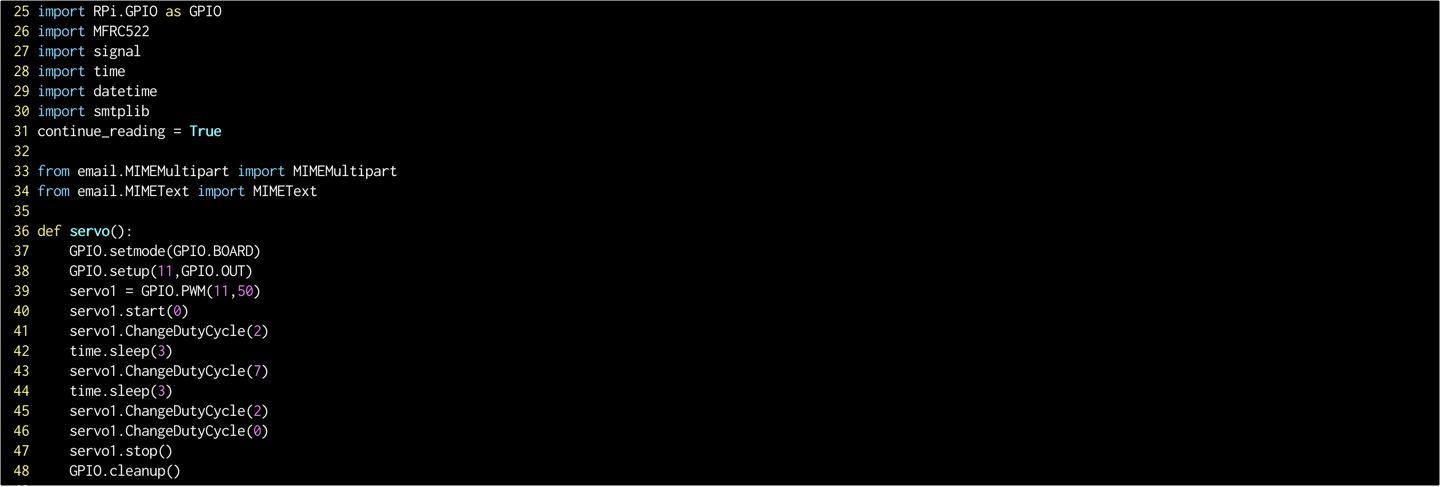
**If a non-registered user scans card:**



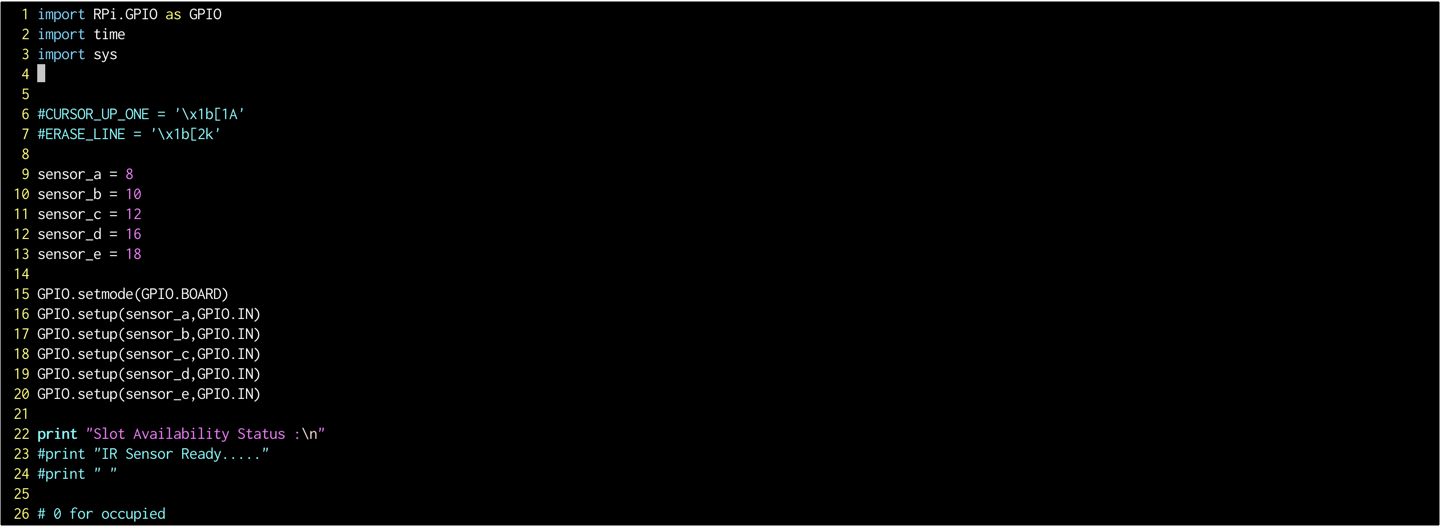
**Mail Module:** Using the SMTP function, we will send out mail to users upon exiting, which contain in-time, out-time and cost corresponding to total time parked. It will also contain the remain balance in the wallet.



# **Servo Motor:** This module controls the gate at the entrance of the parking lot. This gate opens only when a registered user scans their card.



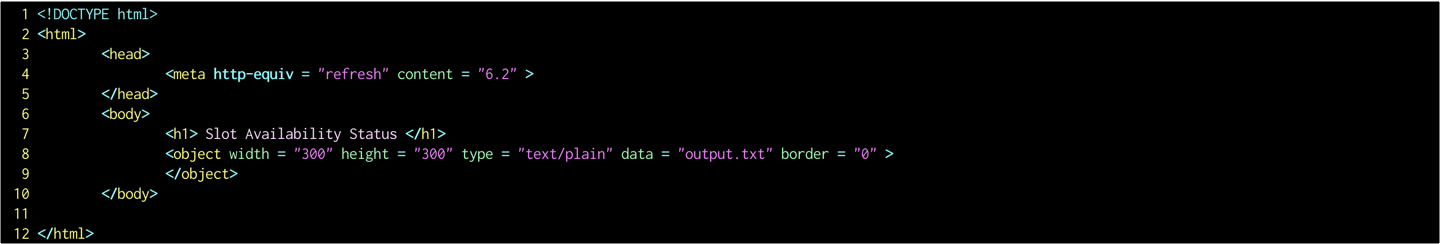
**Initialize Proximity Sensor:** Setting up the proximity so that the output from the sensor (high or low) can be read by the python program.



**Working of Proximity Sensor:** The current status of the sensors is written on to a file continuously with the gap of 1 second. The output in the terminal seems dynamic as the output is cleared a reprinted every 1 second (notice the same in the video).



**HTML File:** This html file takes the data from output.txt (contents of this file are written down by the python code running the sensors. There is an addition command to enable auto refresh the webpage. This html file is hosted on the internet using a third-party host: ‘ngrok’.

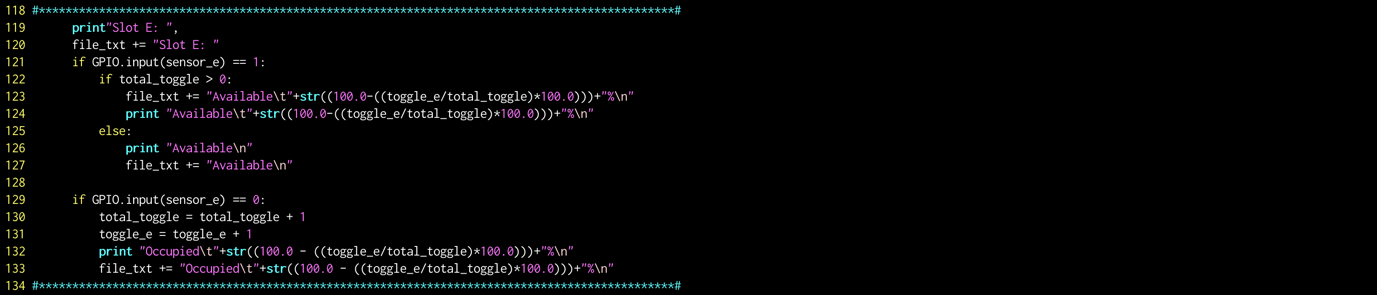


# Additional features:

**Slot Availability Probability**

We had added another feature which would indicate the probability that a slot will be available. This feature is an alternative to slot booking. The probability will allow the user to make a more calculated guess about finding a free space. The probability system might seem obvious in some real-world scenarios like in mall parking lots where slot close to the entrance have a lower probability of being ‘available’. Nevertheless, we believe probabilities will an added advantage.

This feature was implemented after we shot the project video, hence as a reference, I am pasting the code snippet below.



**IOT Team**

Tanmay Pathak – 2018102023

Utkarsh Mishra – 2018102020