Monitoring Vehicular Traffic Inside Campus

Entry/Exit and Database Management Program

Need Statement

Every Higher Education Institute in the country, especially the ones located in Metropolitan Cities have a very high number of visitors daily. These visitors vary from regular visitors such as Students and Permanent Faculty members to the occasional visitor such as an Industry Representative or a Guest Speaker. At any given time, a considerably large number of People are present inside the Campus. As a result, for the sake of security of Students and Staff it becomes essential to keep track of visitors inside the campus and also store these records for the foreseeable future.

Abstract

Presently, the track of these records is done manually by the Security Guards in Separate Entry/Exit Registers which are maintained at every Gate. This current system to track records has some major flaws such as:

- Different Gates have different sets of Registers i.e. a lack Global Database system which makes looking for data a tedious process.
- Manual Method of storing data increases chances of errors in records.
- A vehicle which might enter the college regularly needs to stop at the Guard Post every time to verify itself and the passengers thereby taking more time.

Our Entry/Exit System uses a Centrally maintained Database which makes the data far more user-friendly and accessible and real time synchronization of data over wireless network between all the gates in the campus. A Digital Database helps in decreasing the number of errors and increasing the longevity of data in the records. After a First Time Only Registration the vehicles are allowed faster Entry and Exit inside or outside the campus without compromising the integrity of records. Moreover, because these records are maintained on a Central Database, these records can be accessed anywhere by anyone with permission to access these records.

Working

The system consists of Raspberry Pi(s) (One Per Gate) and Computers in the Guard Rooms which are all interconnected to each other with the help of a wireless network. The User (Security Guard) gives a command on a GUI-enabled software installed on the Guard Room Computer, which then instructs the RaspberryPi to click a Photograph (through the Web Cams attached to it) of the vehicle.

The Database is maintained on a Server Computer in the form of many different files. These files contain information about all registered users, vehicles in the campus at any given time, and their entry and exit times. The system also only allows the access of data at the Server Computer and therefore unauthorized access of data is controlled. The License Plate Number is recognized from the photograph using an API based script of OpenALPR. The system always requires internet connection because of the OpenALPR script.

Once the image is processed and a Number is obtained, the number is then searched in the existing records for a match. If a match is found, the user is allowed Entry/Exit which is signaled by a prompt on the Guard Room Computer, and physically through a Bulb mounted on the gate. If no record is found, a case which can only arise at the time of entry, a Form opens on the Guard Room Computer, where details such as Name, Phone Number and type of Visitor are stored in the database. The whole process (in the case of a New User) takes approximately 7 to 8 seconds.

Implementation

There are markings in the on both sides of the gate where the vehicle stops, and the cameras are placed on either side of the Gate which always remains closed. The light bulbs which are mounted on the gates, light up to signal Entry or Exit. The whole setup is installed on the gate inside a splash proof casing so that it doesn't obstruct the motion of the gates.

Software

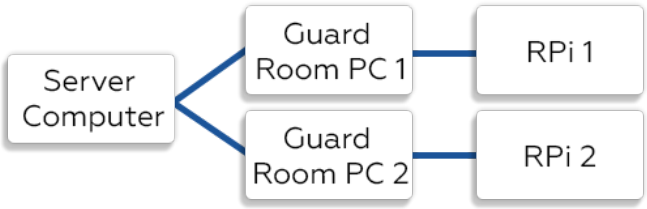
The front-end of the software is installed on the Computer inside the Guard Room. The front-end is a GUI Enabled Python Script which controls the Exit and Entry Cameras and is also used to manually update the central database at the time of 'New User Registration'. The back-end on the other hand consists of Two scripts one running on a Raspberry Pi and the other a Server Script which controls and maintains the network on which the system works.

The GUI program helps also in verifying the captured number plate and in case of some error it also has a provision for the Guard to manually enter the Number Plate or capture the photograph again so that it can be processed once again by the Open ALPR script. The GUI script is made using Python's tkinter library and every button leads to a new or previous frame.

The Raspberry Pi runs two scripts, the OpenALPR script which processes the photograph and extract the license plate using a cloud-based API. The second script is the client-side script which takes in commands from the GUI script running on the Guard Room's Computer and takes photographs and sends them back to the Guard Room PC.

The Server and the Client side make use of the sockets library to transfer data between them. The OpenALPR script requires an active internet connection to function.

The whole system network has the following structure.

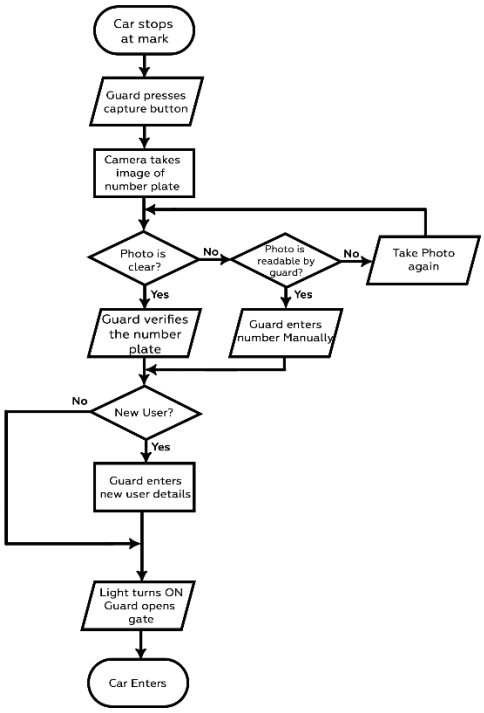


Hardware

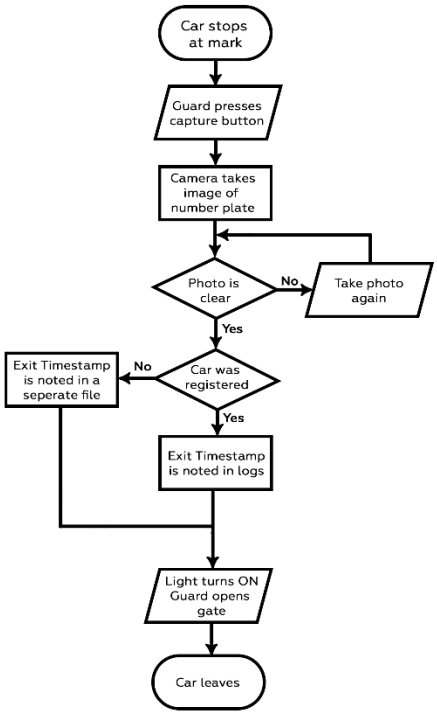
We have used the following pieces of hardware to successfully able to implement the system:

- Raspberry Pi(s) (One at every Gate)
- 5V Relay module
- Buck Boost converter module
- 720p camera
- 220V bulbs

System Flow

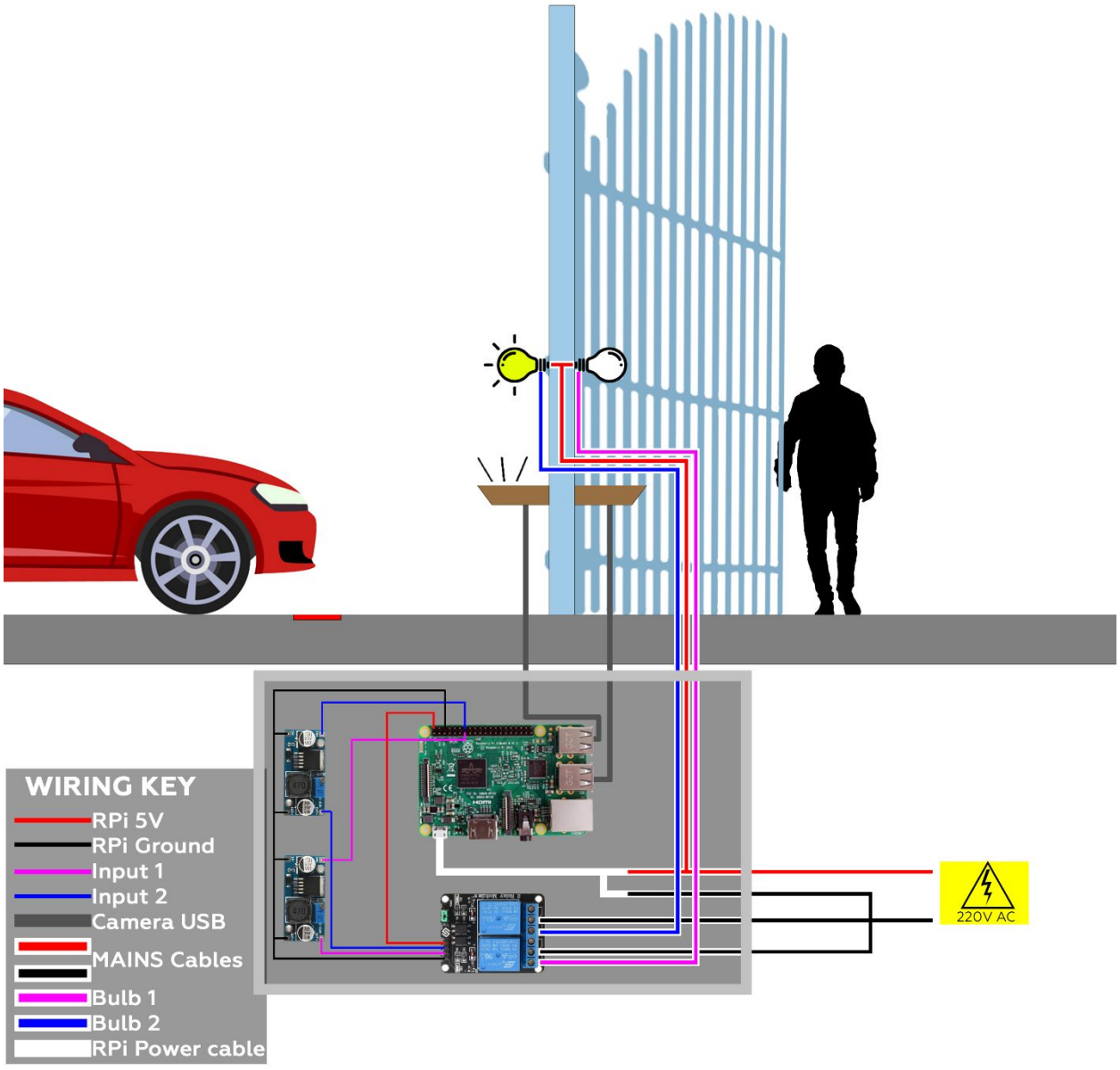


Entry Flow



Exit Flow

Circuit Diagram



Basic Circuit Diagram

Error Handling

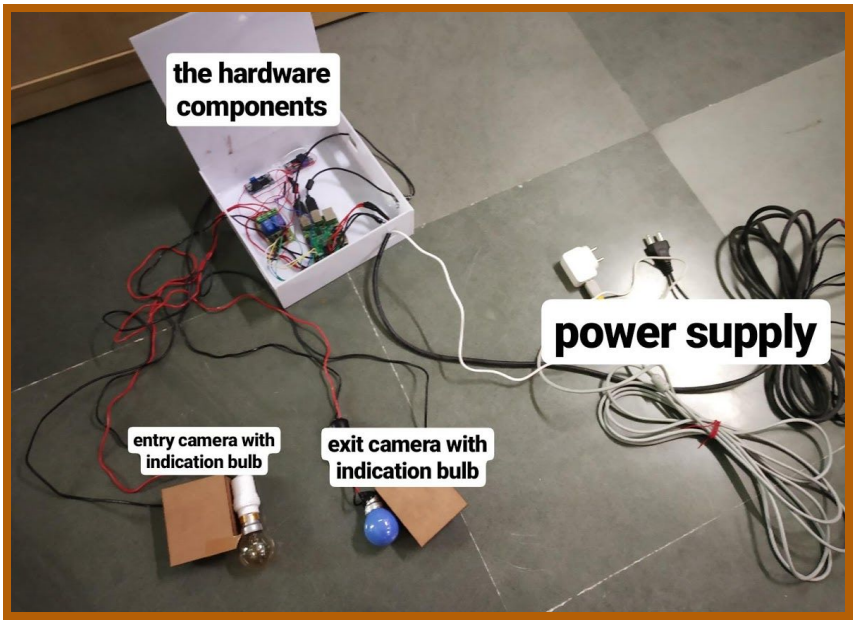
We have minimized the possibility of error as much as we could as of now. The major fallouts that we have successfully tackled are:

- Double entry in records by one vehicle
- Wrongly read number plate
- Incorrect data

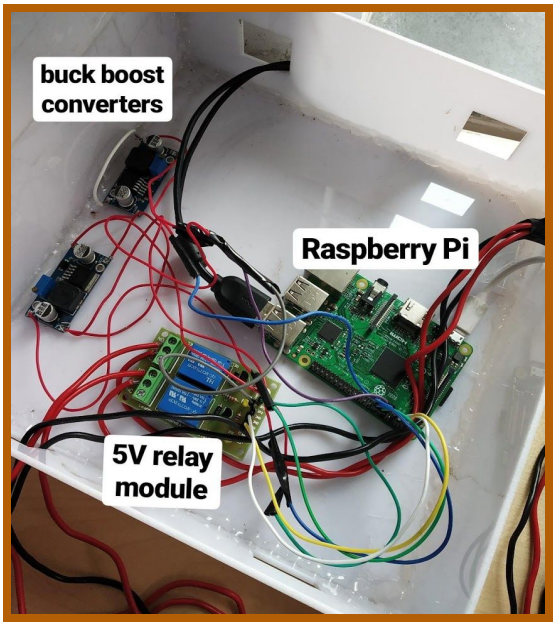
Constraints

- The system currently takes in account only four-wheeled vehicles.
- As the system relies on information taken from photographs, any dust or other obstructions in front of the camera can hamper the system's performance
- The system requires constant power and a stable internet connection to function properly

Photographs



Final Setup



Hardware Components

Close up Image of Camera Used

Setup on Gate