

# Centralized Attendance System

## Summer Project

-by-

*Ajinkya Gorad*

*Avineil Jain*

*This page is intentionally left blank*

## About

This document describes the implementation of Centralized Attendance System (CAS) based on rfid tags and biometric (Fingerprint) .Using Ethernet (Local Network) to check those id on a local server & notify the user about the same.

This CAS has plug and play compatibility of input sensors ,RFID and Fingerprint.

# Index

## Contents

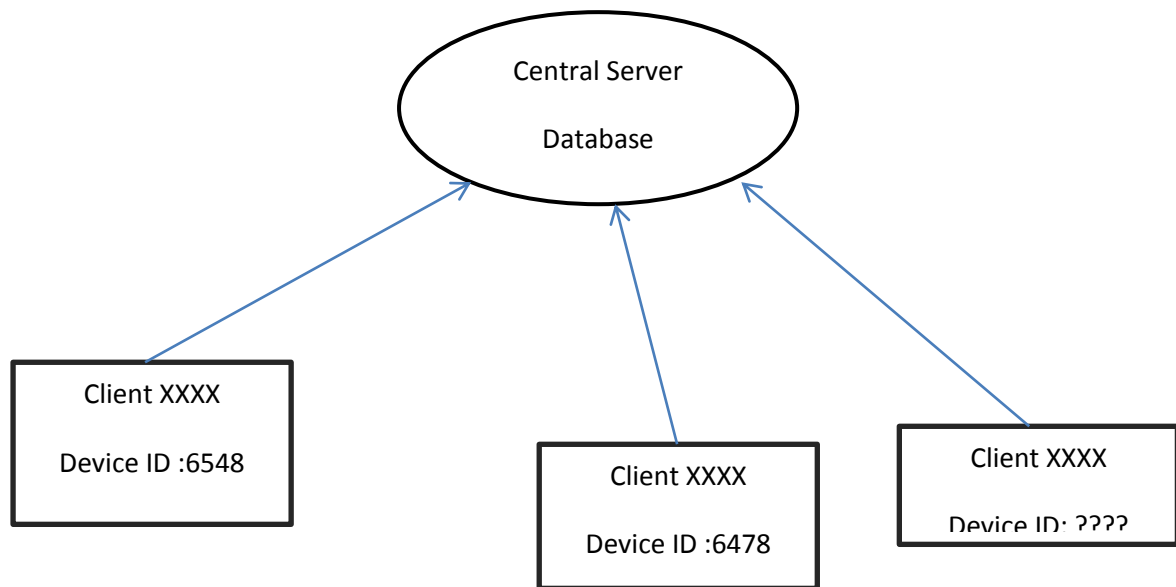
Centralized Attendance System .....	1
About.....	3
Index.....	4
Abbreviations.....	5
Introduction .....	6
Hardware .....	7
Software .....	10
Schematic Circuit Diagram .....	10
Modules .....	10
Central Processor .....	10
Communication.....	11
User Notifier.....	11
ID Reader.....	12
Miscellaneous .....	13
Program Flow .....	14
Device Settings.....	14
Ethernet .....	14
Server .....	15
Communication Protocol .....	15
User Transaction .....	15
<i>Method</i> .....	15
Request Template for User Transaction .....	17
Startup Transaction.....	17
PHP files .....	18
Problem Faced During Development.....	18
Design Flukes .....	18
Future Work /Features Not Implemented.....	19
References/Links.....	19

## Abbreviations

<b>uC</b>	<b>Microcontroller</b>
<b>ISP</b>	<b>In System Programming</b>
<b>GPS</b>	<b>Global Positioning System</b>
<b>CAS</b>	<b>Centralized Attendance System</b>
<b>RTC</b>	<b>Real Time Clock</b>
<b>UID</b>	<b>User ID</b>
<b>RFID</b>	<b>Radio Frequency ID</b>
<b>ID</b>	<b>Identification</b>
<b>TCP</b>	<b>Transmission Control Protocol</b>
<b>RAM</b>	<b>Random Access Memory</b>
<b>UART</b>	<b>Universal Asynchronous Receiver Transmitter</b>
<b>SPI</b>	<b>Serial Peripheral Interface</b>
<b>I2C</b>	<b>Inter Integrated Circuit</b>
<b>IP</b>	<b>Internet Protocol</b>
<b>LCD</b>	<b>Liquid Crystal Display</b>
<b>LED</b>	<b>Light Emitting Diode</b>
<b>MAC</b>	<b>Media Access Control</b>
<b>DNS</b>	<b>Domain Name System</b>
<b>DHCP</b>	<b>Dynamic Host Configuration Protocol</b>
<b>EEPROM</b>	<b>Electronically Erasable Programmable Read Only Memory</b>
<b>SD</b>	<b>Secure Digital</b>
<b>GPRS</b>	<b>Global Packet Radio Service</b>
<b>GSM</b>	<b>Global System for Mobile communication</b>
<b>SIM</b>	<b>Subscriber Identity Module</b>

## Introduction

Centralized Attendance System (CAS) is the attendance system with central server with many clients connected to it. It has plug and play compatibility of user ID input sensors.



Where each client consists of a 4 main parts with other subparts

- ➔ Central processor : [ATmega2560 , Arduino mega board]
- ➔ ID reader : [RFID/ FINGERPRINT]
- ➔ User Notifier : [LCD, Buzzer, RGB LED]
- ➔ Communication maker : [Ethernet , W5100 Ethernet Chip]
- ➔ Miscellaneous : [GPS receiver, RTC clock]

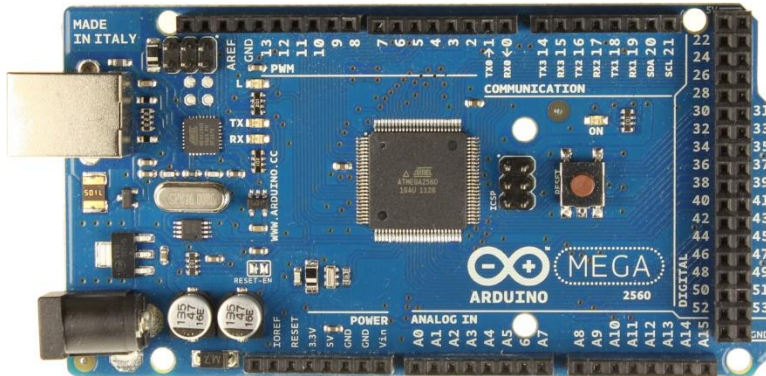
And server consists of

- ➔ Database consisting **userid** corresponding to **type**
- ➔ database of registered devices
- ➔ A startup configuration file (PHP script) to optionally configure device for setting date/device id.

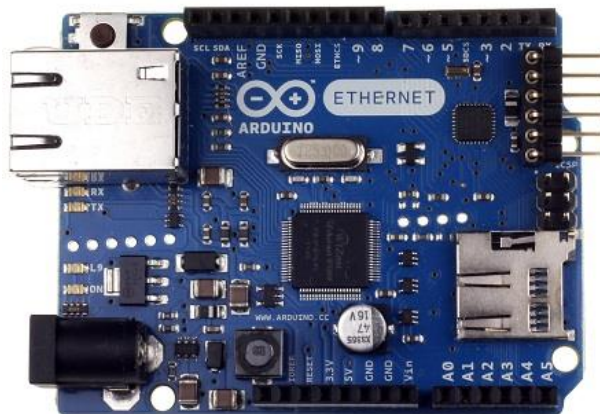
## Hardware

Following hardware was used

- Arduino Mega : (<http://www.arduino.cc/en/Main/ArduinoBoardMega2560>)



- Arduino Ethernet shield (based on wiznet chip w5100) : (<http://www.arduino.cc/en/Main/ArduinoBoardEthernet>)



- Hitachi HD44780 LCD : (<https://www.sparkfun.com/datasheets/LCD/HD44780.pdf>)



- Tiny RTC module for Real Time Clock (based on DS1307) : (<http://smart-prototyping.com/Arduino-Tiny-RTC-I2C-real-time-clock-module-24C32-storage-DS1307.html>)



- RGB Led : (<https://www.sparkfun.com/products/105>)



- Buzzer :



- EM-18 RFID sensor : (<http://www.tomsonelectronics.com/uploads/1430561217EM-18-RFID-Reader.pdf>)





- R305 Fingerprint sensor : ([http://amateurworld.in/download/products\\_extra\\_files/fingerprint-module.pdf](http://amateurworld.in/download/products_extra_files/fingerprint-module.pdf))



- SIM28 GPS module : ([http://www.vis-plus.ee/pdf/SIM28@SIM68R@SIM68V\\_NMEA\\_Messages\\_Specification\\_V1.01.pdf](http://www.vis-plus.ee/pdf/SIM28@SIM68R@SIM68V_NMEA_Messages_Specification_V1.01.pdf) )



## Software

Following software were used

- Atmel studio 6 (Version: 6.2.1153) ,( <http://www.atmel.com/tools/atmelstudio.aspx>)
- Visual Micro Plugin for atmel studio for arduino 1.6.1  
(<http://visualmicro.com/downloads/Releases/ArduinoForVisualStudio.msi>)
  - From (<http://www.visualmicro.com/>)
- Arduino 1.6.1 (<http://arduino.cc/download.php?f=/arduino-1.6.1-windows.exe>)
  - From (<http://www.arduino.cc/en/Main/OldSoftwareReleases>)
- Wamp Server (<http://www.wampserver.com/en/>)

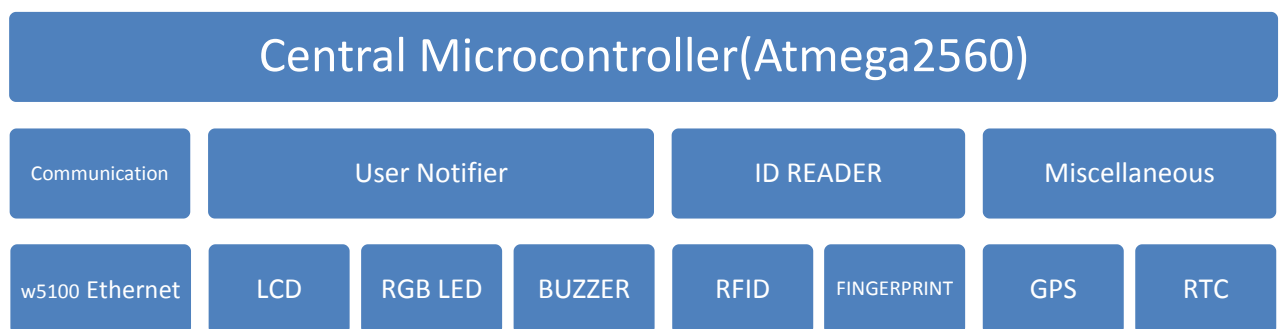
Following software were used for debugging purposes

- Realterm Serial Terminal : ( from <http://realterm.sourceforge.net/>)
- Wireshark : (from <https://www.wireshark.org/>)
  - For debugging get request packets and testing connection issues

## Schematic Circuit Diagram

See file cas\_schematic.pdf

## Modules



## Central Processor

Central Microcontroller ( ATmega2560 on Arduino Mega) controls every other module . It works on 16 MHz with instructions rate approaching 1 MIPS.

Here is link to the datasheet of ATmega2560 uC <http://www.atmel.com/images/doc2549.pdf>

Specification :

- 256Kbyte in system Flash
- 8Kbyte RAM
- 4x UART, 1x SPI, 1x I2C
- (refer datasheet )

## Communication

### *Wiznet Ethernet Module*

- Wiznet w5100 Ethernet module/chip has a on chip TCP/IP stack with IPv4 support.
- It has all the TCP task management functions hardcoded on it. Giving easy access to internet embedded systems.
- Physically connected to SPI port of the ATmega2560 with Chip Select pin as 10.
- Managed by class 'wiznet' in 'wiznet.h'.
- Uses Arduino's 'Ethernet.h' library.

### User Notifier

This notifies the user about current operation status of device.

### *RGB LED*

- This LED indicates the status of a transaction.
- In practice it can show variety of colors, but 3 colors are chosen according to the transaction result.
  - Transaction Successful: User Registered -> GREEN
  - Transaction Successful: User Not Registered -> RED
  - Transaction Failed : -> BLUE
- Colors can be changed for respective status in the software by function 'set' in 'RGBLED' class.
- Physically connected to pins 5, 6, 7 corresponding to R, G, B.

### *LCD*

- This shows the textual status of transaction when initiated, else by default shows current date and time.
- Managed by class '**LCD**' in '**lcd.h**'. Support for default LCD text is also added while displaying current date & time.
- Uses arduino's '**LiquidCrystal.h**' library for Hitachi HD44780 lcd displays
  - Has flexibility to choose any LCD size (currently used is 16 x 2).
- Physically connected to pins from 22 to 28 on Arduino Mega.

## **BUZZER**

- Gives the sonic indication to user about the result of a transaction process.
- Programmed with 3 different kind of sounds for each type of transaction result , and one default for swipe.
- Uses arduino's function '**tone**' to generate frequency for specific duration of specified pin.
- Physically connected to pin 13.

## **ID Reader**

- This unit gets the device ID from the device ( although there is no physical layer externally ) this is on software level.
- And managed by class '**idClass**' in '**id.h**'.
- Physically connected to Serial1 (or UART\_1) and SENSOR\_SELECT pin on 29.
- **SENSOR\_SELECT pin** decides which **sensor to be used** and is connected to the module boards for taking the desired input from sensor.
  - **HIGH** on this pin -> **RFID SENSOR**
  - **LOW** on this pin -> **FINGERPRINT SENSOR**

## **RFID**

- RFID (Radio Frequency Identification) uses sensor **EM-18** sensor for sensing 12 byte RFID sequence from a RFID tag.
- In software, managed by class '**RFID**'

## **FINGERPRINT**

- Fingerprint module uses sensor R305 to do all the task of registering/ checking/ deleting on the device.
- A demo software to add/delete fingerprint is available on <https://dl.dropboxusercontent.com/u/87066238/Sunrom-3935.zip> from <http://www.sunrom.com/p/finger-print-sensor-r305>
- Storage capacity is 250 fingerprints.
- For more info refer above link.
- In software, managed by class **fingerPrintClass**.
- Uses Arduino library '**Adafruit\_Fingerprint.h**' .

## Miscellaneous

### *GPS*

- GPS (Global Positioning System) is added to get the global position of the device( upto certain accuracy).
- Uses GPS sensor SIM28 from SIMCOM.
- Managed by 'GPSClass' in 'GPS\_NMEA.h'.
- Physically connected to Serial3 (or UART\_3)
- Can work with any other GPS sensor which outputs a NMEA sequence 9600 bauds (default) without change in code.
- Baud rate can be configured in 'getLatLon' function in 'GPSClass'
  - As `Serial3.begin(_baud_rate_as_integer);`

### *RTC*

- RTC (Real Time Clock) is added to get real time from module
- Uses DS1307 RTC chip from Dallas.
- Accuracy in time keeping and drifting dependent on type of xtal used.
- Managed by 'timeClass' in 'time.h'
- Physically connected to SCL and SDA pins of ATmega2560.
- Requires a Li-ion battery for backup power for DS1307.
- Time can be set by altering configurations in 'startupSet.php' while device power ups (or resets).

## Program Flow

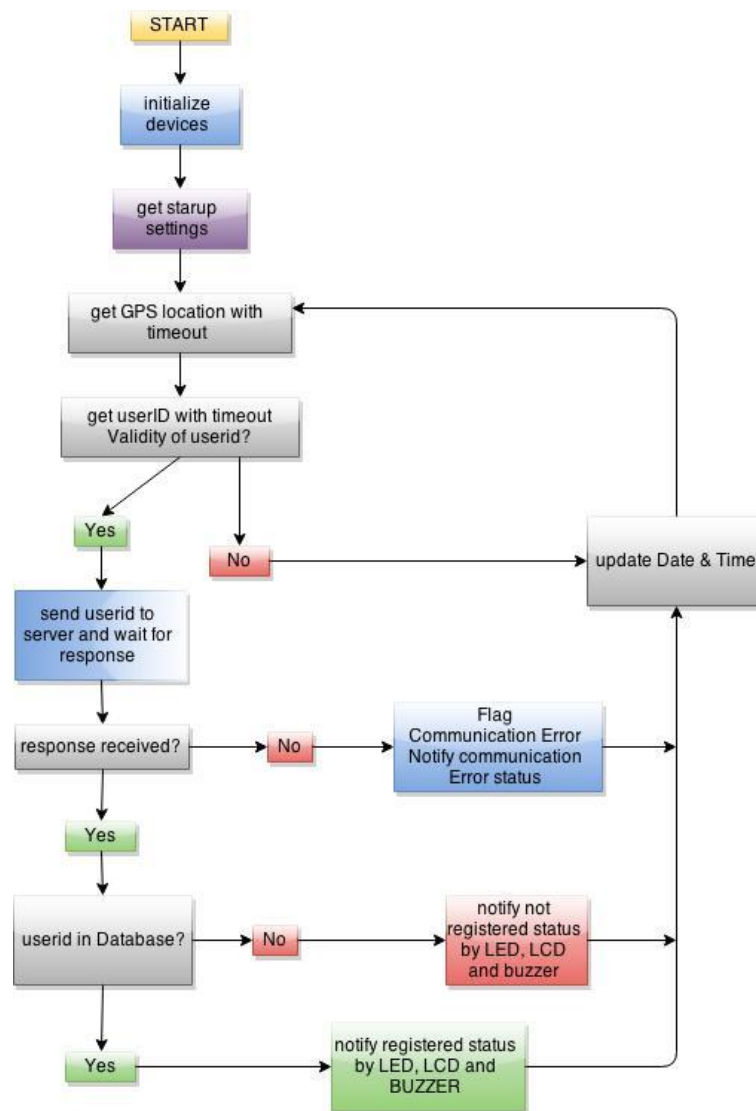


Fig : Program Flow Chart

## Device Settings

Many device options may need to be configured before using it on another node or network. This includes software on server as well as code on the device.

### Ethernet

Following configuration options are available in 'wiznet.h' and 'wiznet.cpp' files

- Setting IP address, MAC address, subnet mask, DNS IP for a device .
- Setting Port of the server (default is set to 8080), but can be any possible port.
- ! Arduino supports DHCP but at a cost of increased memory usage but can be implemented.

- Change the 'startupFile' and 'checkFile' in 'wiznet.cpp' as
  - 'starupFile' string contains the link to file <startup\_file.php>, the file which is stored on server.
    - This file is used for getting startup settings on first booting of device.
  - 'checkFile' string contains the link to file <check\_file.php>, the file which is stored on the server.
    - This file is used for checking a specific userid (UID) with all other transmitted data with the server's database to return a response of 'id\_found' or 'id\_not\_found'. Which states that user is registered or not.

## Server

Following configurations to be changed (tested in Windows 8.1 with using Wamp server)

- Set port as programmed in device (default is 8080)
- Enable/ Add firewall rules for Inbound & Outbound for that port
- Make sure server is at sufficiently top on the routers network to allow connection from client, like public for direct access.

## Communication Protocol

Communication is over the Ethernet, initiated only by device only via GET commands over http.

## User Transaction

Various types of data fields (but fixed in number) has to be transmitted through the request in each transaction.

These fields [key] => [value] are as follows : ( for the steady state of the system)

- METHOD=CHECK (can be added for ADD/DELETE)
- TYPE=RFID/FINGERPRINT
- UID=<ID CORRESPONDING TO <TYPE>>
- DEVID=<UNIQUE ID OF DEVICE>
- TIME=<TIME IN "HHMMSS" FORMAT">
- DATE=<DATE IN "DDMMYYYY" FORMAT">
- LAT=<LATTITUDE IN STRING>
- LON=<LONGITUDE IN STRING>

**\*Latency** meaning to be taken as ability to work in xx conditions.

## Method

It indicates the type of process to do on sever (currently only CHECK)

- Check
  - Check if UID exist in the databases
  - Latency : both online and offline

### *Type*

This indicates the type of sensor was used to take the UID (userid)

- RFID
  - UID of type RF Identification
  - 12 byte data
    - E.g. 18008DD45918,...
- FINGERPRINT
  - UID of type obtained from fingerprint
  - String preceeding with "ID"
    - E.g. 'ID9','ID45',...

### *UID*

It indicates the UserID , which a user will have, and will be stored on database of server.

- Can be any string/ specific data and unique to a DEVID.

### *DEVID*

It holds the uniquely assigned ID to a device

- This ID needs to be specified on first operation using startupSet.php
- It is stored on EEPROM of the ATmega2560 with at address 0x0000 as 'IDXXXX'
- But if ID is "IDxxxx" data transmitted will be DEVID=xxxx

### *Time*

It holds the time of getting of UID

- In 'HHMMSS' format (string)

### *Date*

Date of getting of UID

- In 'DDMMYYYY' format (string)

### *LAT*

Holds the latitude of the device when UID was taken

- In metric units as 'ddd.dddddddd' degrees (string)

### *LON*

Holds the Longitude of the device when UID was taken

- In metric units as 'ddd.dddddddd' degrees (string)



## Request Template for User Transaction

- Following is the general template for sending GET request with minimum numbers of header options

**GET <url>?key1=value1&key2=value2...keyN=valueN HTTP/1.1**

**Host: <server>**

**Connection: Close**

- Example : ( transmitting only UID though)

**GET /cas/check.php?METHOD=CHECK&UID=18008DD45918 HTTP/1.1**

**Host: 10.3.192.78**

**Connection: Close**

- Result from the server to the device is an http response with string 'id\_found' or 'id\_not\_found' .

## Startup Transaction

- Startup transaction occurs after booting of device after initialization of each device. The startup function is in class 'wiznet' in 'wiznet.h' & is called in a setup with timeout.
- Following is the startup request sent by the device

- **GET <startupFile>?DEVID=<device\_id> HTTP/1.1**

**Host: <server>**

**Connection: Close**

where device\_id is the allotted device id to the device , although server can override this and assign a different one by following. By default if id has not been assigned device\_id will be '0000'.

NOTE: startupFile is the file on the server which sends startup settings stored in another file (startup.txt)

- Response to be expected is http with following data after header :

- **DEVID=XXXX**
- **METHOD=<method>**
- **TEXTLCD=<text\_lcd>**
- **DATETIME=<dateTime in format YYYYMMDDHHMMSS ie YmdHis format >**

There is no restriction on the permutations of the above parameters and even parameters can be absent. (see 'setStartupSettings' in 'wiznet.cpp')

NOTE : date time to set using time zone (in PHP).

Currently only CHECK method is supported

- Only the parameters specified are modified/applied

- Example result type to be expected is

- **METHOD=CHECK**

**DATETIME=20150531063225**

Parameters modified in this are method and dateTime

- **DEVID=5454**

**DATETIME=20150531063225**

**TEXTLCD=HELLO**

### **METHOD=CHECK**

All parameters are modified in this method.

- In case of communication error default method of 'CHECK' is chosen.

## **PHP files**

Test files used on server (on Win8.1 using Wamp)

- check.php : checks whether given UID with <TYPE> is on the server or not( for that <DEVID>)
- client.txt : stores the list of registered devices ( only their DEVIDs)
- id.txt : stores the list of registered UID with <TYPE>
- log.txt : any debugging data or transaction is appended here
- startup.txt : stores startup configurations
- startup.php : used when a device boots up and it sets the configuration from 'startup.txt' and execute accordingly)
- startupSet.php : actually html which stores the response from user (server ) to file startup.txt ( interface)
- \*delete.php : handles when DELETE method is given
- \*delete.txt : stores data to be deleted by delete.php as "<TYPE>,<UID> [CRLF]"

Note: marked with '\*' (asterisk) are not implemented but can be thought of future work.

## **Problem Faced During Development**

Following were the troublesome problems faced during the course of the project

- **Ethernet Shield** – Arduino shield uses SPI interface from ISP connector .
- **Problems after creating server using Wamp** – Port 80 was not working on Windows, since it was being used by other program, so connection had to be shifted to port 8080 (any other port other than 80 could be used). Also firewall settings are changed.
- **With Visual Micro** –Latest version of Visual Micro with latest Arduino IDE 1.6.4 had bugs.
- **Fingerprint Sensor** – Adarfruit Library uses the array size of 'packet' undefined (auto set) causing the runtime errors on Arduino Mega by resetting it.

## **Design Flukes**

Following are the design considerations which make it vulnerable

- String is used (handy though) which allots dynamic memory, so uC can go out of memory if it is not used wisely , ie. if string operations are subjected on randomly generated inputs (e.g. responses from the server).
- Uses the polling method to check for Fingerprint and RFID data, decreases the probability by getting a card swipe.
- Device resets after plugging of Fingerprint module .

## Future Work /Features Not Implemented

Following features were not implemented

- Solving Design Flukes .
- Cache memory in case of communication failure
  - Planned with SD card, Ethernet shield has a SD card holder
  - Can use Arduino's SD card library
- GPRS with GSM module
  - Planned with SIM300/900,
    - Requires a simcard with internet packet connection
    - Requires a public server; as client has public IP address for direct access.
- Delete/Add features with Fingerprint and RFID without taking out modules for individual registration; ie. Server based registration.
- Adding SQL based support on server.
- Data Encryption to avoid false transactions and protect the data.

## References/Links

- Project Github Link : [https://github.com/ajinkyagorad/Centrallised\\_Attendance\\_System](https://github.com/ajinkyagorad/Centrallised_Attendance_System)
- Atmel studio 6 <http://www.atmel.com/tools/atmelstudio.aspx>
- Visual Micro Plugin for atmel studio for arduino 1.6.1  
<http://visualmicro.com/downloads/Releases/ArduinoForVisualStudio.msi>
  - From <http://www.visualmicro.com/>
- Arduino 1.6.1 <http://arduino.cc/download.php?f=/arduino-1.6.1-windows.exe>
  - From <http://www.arduino.cc/en/Main/OldSoftwareReleases>
- Wamp Server <http://www.wampserver.com/en/>
- Arduino Mega : <http://www.arduino.cc/en/Main/ArduinoBoardMega2560>
- Arduino Ethernet shield : <http://www.arduino.cc/en/Main/ArduinoBoardEthernet>
- Hitachi HD44780 LCD : <https://www.sparkfun.com/datasheets/LCD/HD44780.pdf>
- Tiny RTC module: <http://smart-prototyping.com/Arduino-Tiny-RTC-I2C-real-time-clock-module-24C32-storage-DS1307.html>
- EM-18 RFID sensor : <http://www.tomsonelectronics.com/uploads/1430561217EM-18-RFID-Reader.pdf>
- RGB Led : <https://www.sparkfun.com/products/105>
- R305 Fingerprint sensor : [http://amateurworld.in/download/products\\_extra\\_files/fingerprint-module.pdf](http://amateurworld.in/download/products_extra_files/fingerprint-module.pdf)
- SIM28 GPS module : [http://www.vis-plus.ee/pdf/SIM28@SIM68R@SIM68V\\_NMEA\\_Messages\\_Specification\\_V1.01.pdf](http://www.vis-plus.ee/pdf/SIM28@SIM68R@SIM68V_NMEA_Messages_Specification_V1.01.pdf)
- Realterm Serial Terminal : from <http://realterm.sourceforge.net/>
- Wireshark : (from <https://www.wireshark.org/>)
- Software for fingerprint <https://dl.dropboxusercontent.com/u/87066238/Sunrom-3935.zip>  
from <http://www.sunrom.com/p/finger-print-sensor-r305>