**Specifications**

SMS Solution and Other Enhancements

**Vision**

1. To build SMS Solution which will send the fault description of a particular IOT device the mobile of the selected city, permitted the city has been allowed the SMS sending service.
2. To build some small enhancements/modiciations in the existing solution.

**Mission**

1. To develop EXE, which will schedule to send the SMS to a particular city depending on whether the city has been allowed the SMS sending service. The whole functionality is build round this EXE.
2. To develop small enhancements/modifications in reports etc.

**The Final Output**

The Final EXE. would look something like this. EXE, which will be running on the production server. The EXE

|  |
| --- |
| **SMS Engine** |
| SMS Engine Started… |
| Establishing Database Connections………. |
|  |
| Reading Device Fault Information………. |
|  |
| START Device Processing |
| Device ID 3667………. |
| Reading SMS Service Enability for this Device………. |
| SMS Service NOT Enabled for this Device………. |
| Processing the Data………. |
| END Device Processing |
|  |
| START Device Processing |
| Device ID 3668………. |
| Reading SMS Service Enability for this Device………. |
| SMS Service Enabled for this Device………. |
| Preparing SMS………. |
| Sending SMS to the Users of the Device………. |
| SMS sent Successfully to all Users………. |
| Logging the SMS Data………. |
| Processing the Data………. |
| END Device Processing |
|  |
| START Device Processing |
| Device ID 3668………. |
| Reading SMS Service Enability for this Device………. |
| SMS Service Enabled for this Device………. |
| Preparing SMS………. |
| Sending SMS to the Users of the Device………. |
| SMS sending failed for the users <username1>,<username2> |
| Logging the SMS Data………. |
| Processing the Data………. |
| END Device Processing |
|  |
| SMS Device Processing Completed…. |
| Closing Database Connections |
|  |
| Establishing Database Connections………. |
| ERRROR Establishing Database Connections… |
| Sending Error email to [gokul.tile@gmail.com](mailto:gokul.tile@gmail.com) |
| Closing the SMS Engine….. |

**Points to Ponder**

1. The final EXE is a scrolling desktop window on the production server
2. The EXE will take following input parameters in XML file
   1. Schedule Interval (in Mins)
   2. DATABASE Connection string
   3. Error Email
   4. MESSAGE Gateway
      1. SMS=1
      2. Whtsapp=0
3. The Exe will read this XML file on every startup (cold start) of the Exe.

**Introduction**

The IOT Devices are sending Data to the server continuously. Along with this data, the fault information is also sent. This fault information needs to be captured on realtime basis and send to the user via message gateway selected either SMS or whtsapp.

**The main Logic**

1. This SMS solution is to be focused on one Production server for now. The details of the production server as follows –
   1. *Server IP: will be sent to you separately*
   2. *Pwd: will be sent to you separately*
2. The Database it (SMS Engine) will read is **SLC.**
3. In SLC DB there is a table named SLCEvents, which has to be read primarily by the SLC Engine.
4. The SLCEvents has the following table structure and the type of values it will store.

***Table: SLCEvents***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr. | Field Names | Values it will store | Meaning of the Values | Remarks |
| 1 | Id | Number | Autonumber, Primary Key, Unique Key | Autonumber field |
| 2 | Slcid | Number | The previous SLCid for that deviceid | This will help detect the previous state of the faults. |
| 3 | Deviceid | Number | The original deviceid from the SLC table | From here u can check the city, area and location of the device. |
| 4 | Dt | Date Time | The date and time of the Record |  |
| 5 | Aux1 | 0/1 | N=0= OK N=1 = NOK | OK = Everything is normal NOK = fault. Check the SLCFault table for more description of the fault. |
| 6 | Aux2 | 0/1 |
| 7 | Aux3 | 0/1 |
| 8 | Aux4 | 0/1 |
| 9 | Aux5 | 0/1 |
| 10 | Aux6 | 0/1 |
| 11 | Aux7 | 0/1 |
| 12 | Aux8 | 0/1 |
|  | Fault\_ov | 0/1 |
|  | Fault\_uv | 0/1 |
|  | Fault\_OL | 0/1 |
|  | Fault\_UL | 0/1 |
|  | Fault\_OF | 0/1 |
|  | Fault\_UF | 0/1 |
|  | Fault\_OT | 0/1 |
|  | Fault\_GF | 0/1 |
|  | Fault\_PD | 0/1 |
|  | Fault\_PU | 0/1 |
|  | Fault\_ZV | 0/1 |
|  | Fault\_NV | 0/1 |
|  | N | 0/1 |
|  | S1 | 0/1 |
|  | S2 | 0/1 |
|  | S3 | 0/1 |
|  | S4 | 0/1 |
|  | Boost | 0/1 |
|  | R\_failure |  | N=0 = OK  N>0= NOK | OK = Everything is normal NOK = fault. Check the SLCFault table for more description of the fault. |
|  | Y\_failure |  |
|  | B\_Failure |  |
| 13 | R | 0/1 | R=Y=B=1 = Switched ON R=Y=B=0  Switched OFF | To detect ON/OFF of the LED Lights |
| 14 | Y | 0/1 |
| 15 | B | 0/1 |

1. There is already a mechanism in place which checks the SLC Table from its previous record and if any fault occurs, it copies the record in SLCEvents table.
2. You have to do the following tasks
   1. Read the deviceid of every inserted record.
   2. Fetch the cityid of the deviceid from the table ***SLCDevices***.
   3. Check this cityID in the table SMSAuthority to see if this city has SMS Sending Authority
   4. If the isSend value is 1 then check the corresponding smsgatewayid from the table ***smsGateway*\***
      1. Read the SMS Gateway of the city from table ***SMSGateway\****
      2. Read the Area and Location of the City from table ***SLCDevices\*\****
      3. Build the fault information of SMS as follows –
         1. Read SLCEvents,SLCFaults
      4. Build the SMS text of the SMS as follows:

STREET LIGHT ALERT

PROBLEM

<slcEvents.Date Time>

Zone: <slcdevices.zone> Area: <slcdevices.Area>-<slcdevices.location>

UnitID: <slcdevices.deviceid>

<Slcdevices.phase> Phase

Faults:

*Description of Faults from SLCEvents* and *SLCFaults from iii above*

Severity: High/Medium/Low from iii above

Original SLC EventID: *<slcEvents.Slcid>* prevous id of the same device

* + 1. Check the corresponding contact persons of the deviceid from the table ***CP.\*\****\*
    2. Log the SMS in table **smsSentHistory\*\*\*** with statusidfrom the table **statusMaster**\*\*\*\*.
    3. Update the isAlertProcessed field of ***SLCEvents*** to 1
  1. If the isSend value is 0 then
     1. Update the isAlertProcessed field of ***SLCEvents*** to 1 directly.

**\*SMSGateway**

The sms gateway is a new table with following fields

SMSGatewayID

SMSGatewayURL

SMSGatewayUserid

SMSGatewayPwd

Add a new field in table SMSAuthority called smsGatewayID

**\*\*SLCDevices**

SLCDevices has all the information about the device like cityid, area, location zone, phase etc

**\*\*\*CP**

The table CP has contact persons of the devices per deviceID

**\*\*\*\*StatusMaster**

The StatusMaster has status ID of SMS

Insert 0 for OK and 1 for Error status

**SMSSentHistory**

Has SMS sent date and time and statusid of the SMS

**SLCFaults**

New Table has to be created with the following field –

SLCFaultid – Primary,Autonumber,Unique

SLCFaultName

SLCFaultDescription

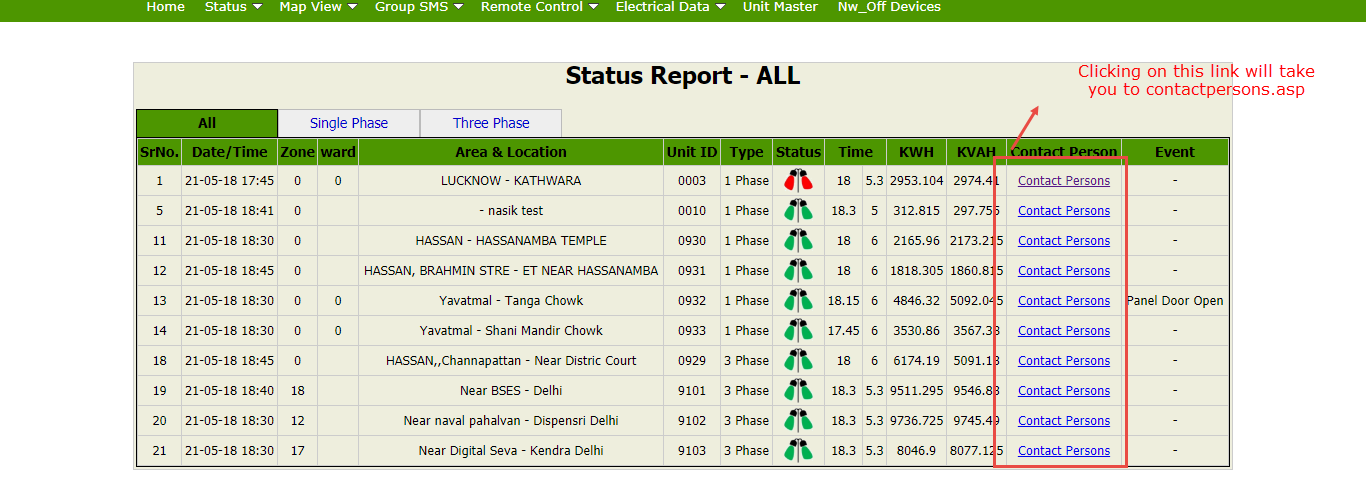
SLCFaultSeverity

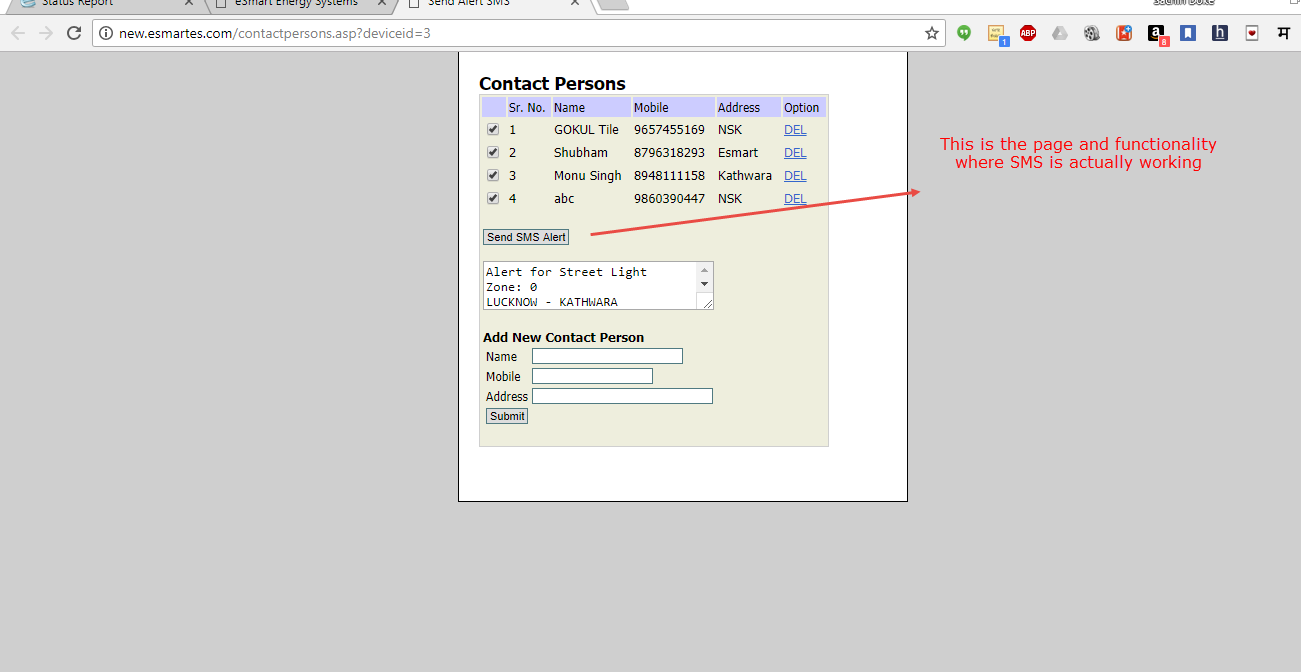
The SMS Functionality can be seen in the following page

<http://new.esmartes.com>

username: demo

pwd: demo123





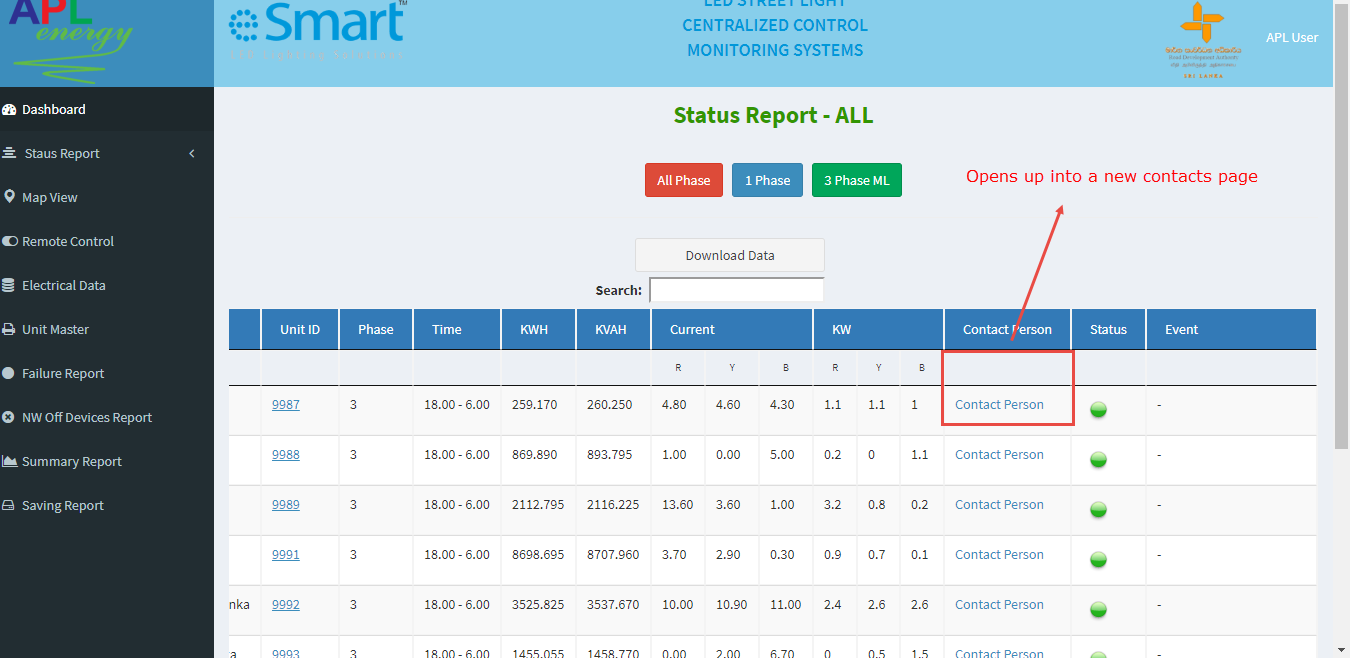
The **smscheckpage.asp** is attached with this mail. The fault information and SMS Gateway information is in this page (I think or it is in the contactpersons.asp page above)

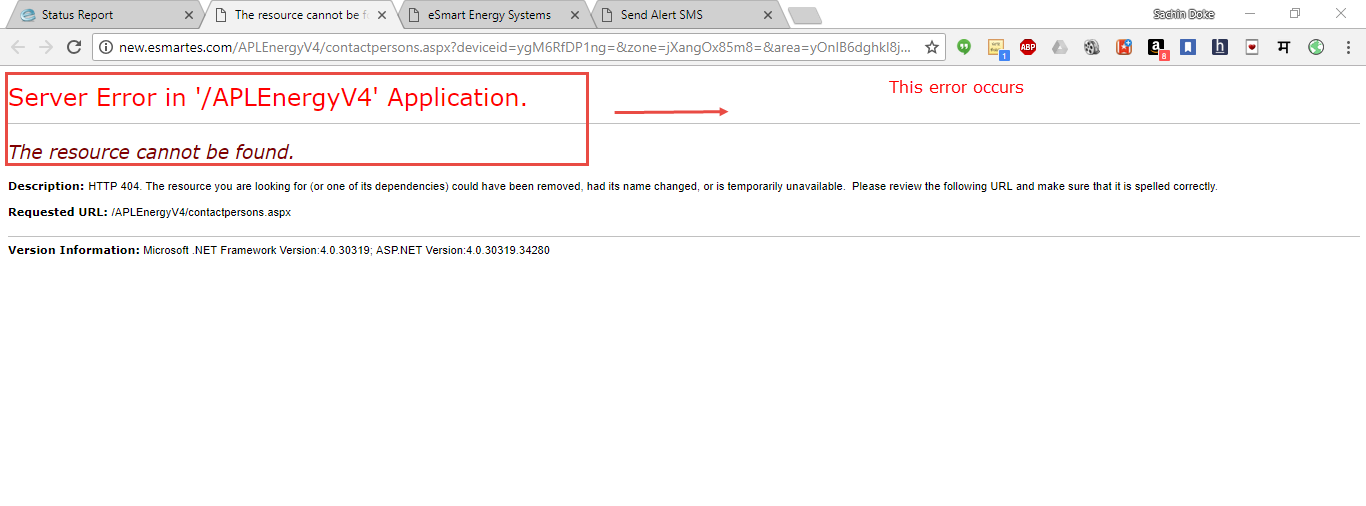
In APLEnergyV4 app the Contact persons page is not working. It gives a resource not found error. **Please fix this**

[**http://new.esmartes.com/APLEnergyV4**](http://new.esmartes.com/APLEnergyV4)

**username: srilanka**

**pwd: Srilanka@1234**

****

****

The Contacts Page needs to be copied from the following

***Ill send this to you later***

**Other Enhancements**

The %Glowing field needs to be added in the Login Report and Status Report.

%Glowing for Phase1 = (((KWR) \*1000)/slcdevices.totalload) \* 100

%Glowing for Phase1 = (((KWR + KWY + KWB) \*1000)/slcdevices.totalload) \* 100

The KWR,KWY and KWB are calculated fields and can be found in status report. See sreenshot below –

