**IFT 540 – Information Systems Development**

**Fall/ 2019**

**Arizona State University**

**IV Curve Automation Project**

**Project Report**

Submitted by:

Avinash Ravichandran

Submitted on : 15th December, 2019

**Contents**

1.Abstract............................................................................................................................................5

1.1 Summary........................................................................................................................... 5

1.2 Results............................................................................................................................... 5

2. Introduction..................................................................................................................................... 6

2.1 Authors............................................................................................................................... 6

2.1 Background........................................................................................................................ 6

2.2 Scope of the Project........................................................................................................... 6

2.3 Objective........................................................................................................................... 7

2.4 Status..................................................................................................................................7

3. Planning.......................................................................................................................................... 10

3.1 Project Description............................................................................................................ 10

3.2 Project Justification........................................................................................................... 11

3.3 Feasibility Analysis........................................................................................................... 11

3.4 Project Scope..................................................................................................................... 12

4. Analysis........................................................................................................................................... 13

4.1 Requirement Gathering...................................................................................................... 13

4.2 Requirement Determination.............................................................................................. 17

5. Design ............................................................................................................................................. 19

5.1 Use Case diagram.............................................................................................................. 19

5.2 Activity Diagram............................................................................................................... 19

5.3 Sequence Diagram............................................................................................................. 20

5.4 Class Diagram.................................................................................................................... 21

5.5 Conceptual Model.............................................................................................................. 22

5.6 Database Schema............................................................................................................... 23

6. Implementation.................................................................................................................................24

6.1 User Interface Design........................................................................................................ 24

6.2 Processing System............................................................................................................ 28

6.3 Output Design.................................................................................................................... 44

6.4 Physical Database Design.................................................................................................. 44

7. Maintenance .................................................................................................................................. 46

7.1 Future Features............................................................................................................... 46

8. Conclusion....................................................................................................................................... 47

9. Appendixes...................................................................................................................................... 48

10. References..................................................................................................................................... 50

**Abstract**

**1.1 Summary**

The current method implemented to calculate the ideal operating values of a solar module is a cumbersome process. The process involves collecting test data from the module at different scenarios, gathering the collected data and then manually calculating the values based on the test results. This process is highly ineffective and requires a change in the method of execution. The proposed system will be able to take in the test data as an input and execute all the manual calculation and provide results to the user and will also store the result in a back end database that would allow for the user to view the previous calculations done. Implementing this system will not only reduce the effort done to compute the values, it will also reduce the possible errors involved in the calculations.

**1.2 Results**

The completed system that takes inputs and the test data file from the output. Upon getting the required information the system would then process the data and present the output to the user. The data would also be sent to the back end sql database which could then be retrieved my database querying.

**Introduction**

**2.1 Authors**

Avinash Ravichandran: is a 1st year graduate student completing his Master's at Arizona State University. Proficient in Python, C, C++, DBMS and data analysis.

**2.2 Background**

The Current Voltage (IV) project deals with the operating values of a solar panel. Currently the process of finding the ideal operating values of a solar panel is done manually, where in the panel is subjected to various conditions and the readings are noted. After subjecting the panel to different condition, and once a rich data set is developed, the values are manually computed and the average operating values are retried.

This process of manually computing can be a cumbersome task as the number of the different values to be computed in a dataset can be vast and subjecting this calculation to be made manually can also result in error prone answers.

A new system is definitely required to handle this process in order to mitigate the complex calculations and also to reduce errors and provide an accurate answer with a shorter calculation period.

**2.3 Scope of the Project**

For this project we intend on developing a software that will help mitigate the complexities involved with the current method of computation. We intend to develop a software system that will help to calculate the average operating performance values for the list of collected values after subjecting the panel to different conditions.

Developing a separate user interface to allow the user to input the value seems redundant and will also disrupt the current method without providing any significant value and is hence out of the scope of this project.

**2.4 Objective**

The main objective of this phase is to explore the idea of the project and elaborating the methodologies that will be used to successfully complete the project.

Another main objective of conducting this phase is to provide the answers to the following questions :

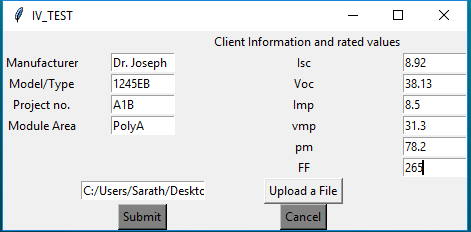
* What the project is about?
* Why to do the project?
* What are the risks involved in this project?
* What are the alternatives to this project?
* What are the estimated resources required for completing the project?
* What is the scope of the project?

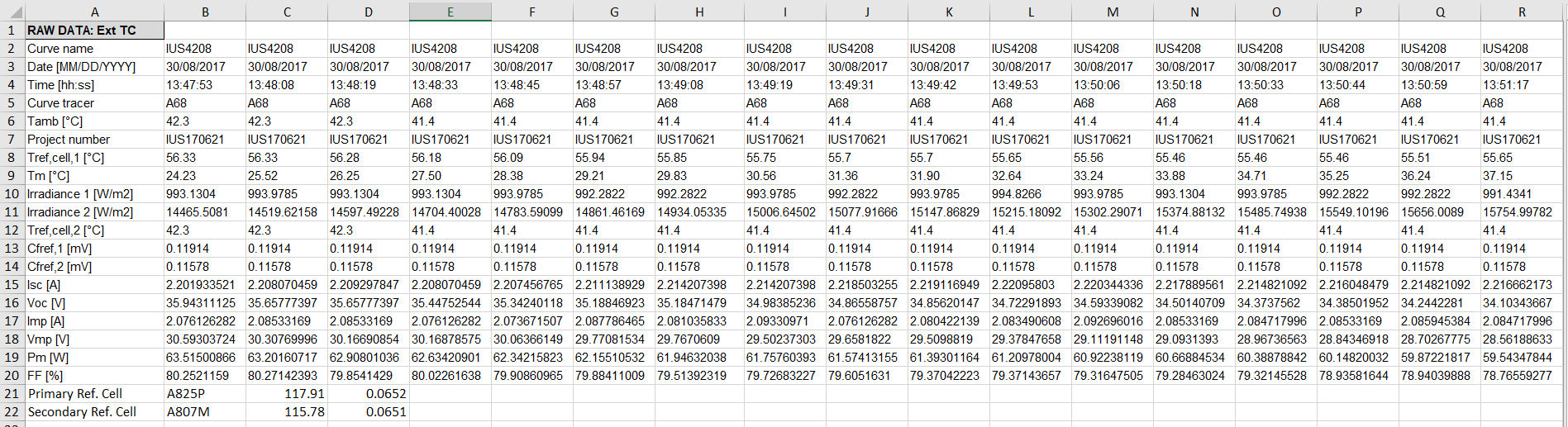
By answering the above questions in this phase, it will give us a better understanding of the direction in which to proceed with the project development.

**2.5 Status**

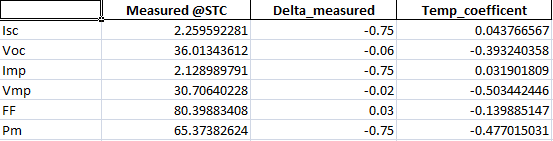
The project is implemented, all the different parts of the system is integrated and the entire system works smoothly. The system accepts the input from the user and is now able to produce appropriate results for the given inputs. The system adds the results to the back end database and also creates an excel file for the user. The system checks all of its functional requirements and is now open for future up-gradation and maintenance.

**Insert Input image**

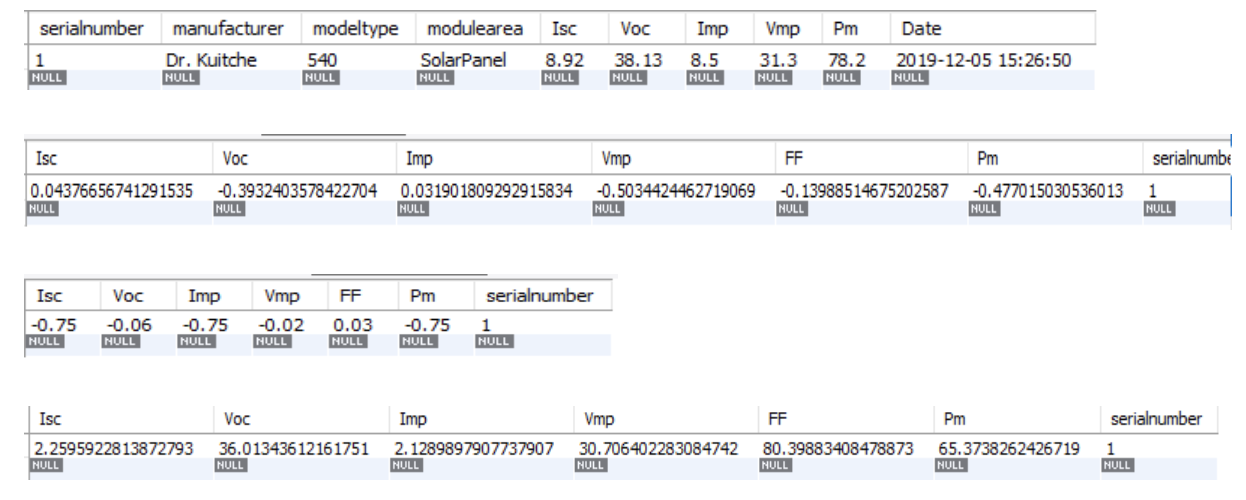
****

****

**Insert Output Report image**



**Insert database image**

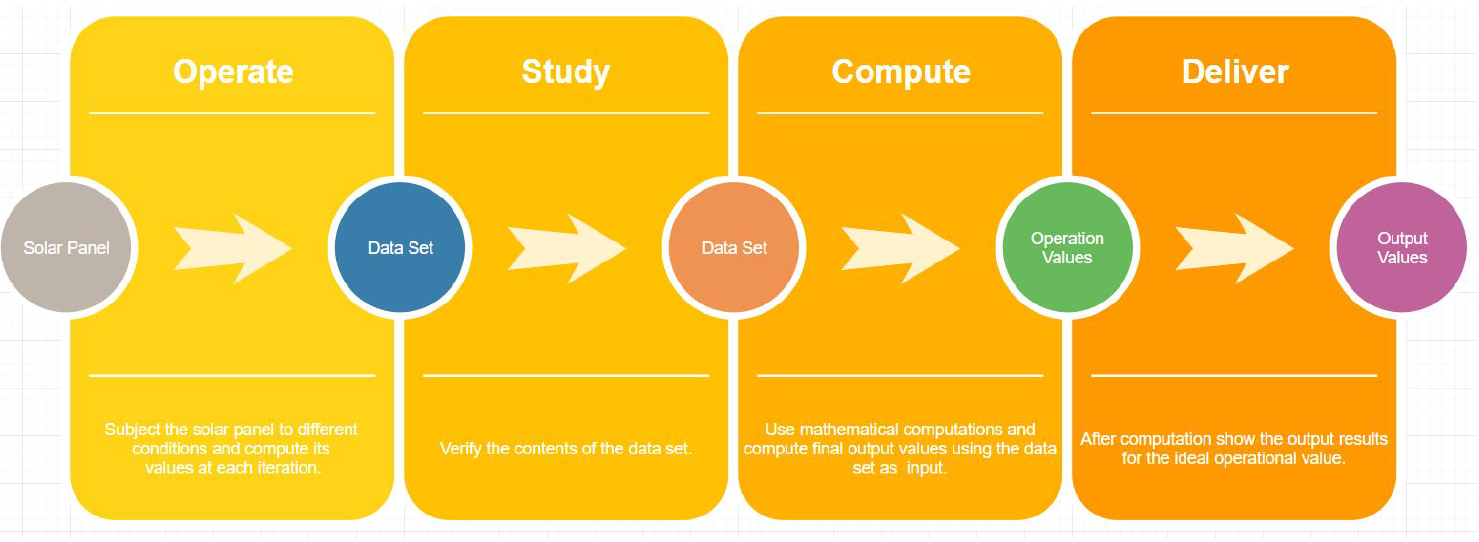


**Planning**

**2.1 Project Description**

Current System

The Current system utilizes a human calculation in order to compute the average operating values. The process starts with subjecting a solar panel to various conditions and finding its output values and tabulate these values. Once a rich data set is produced a human computes the average operating values by using the dataset. A practical dataset is a collection of hundreds of data for several component values. It is a tedious task for a human to be able to convert the entire dataset into a set of ideal operational value set.



Mission Statement

The main objective of this project is to build a software system that will process the tabulated data set to produce the average operating performance values of the solar panel.By developing such a software we will be able to simplify the current process, we will be able to reduce the computation time and the possible errors.

Strategies

The steps that should be followed along the project development cycle in order to achieve the end system are

• To study the computations required to derive the output values.

• Develop a data processing system that can read the input data set from an external database.

• Process the dataset with the appropriate mathematical computations.

• Develop the system to be able to display the necessary output.

All while stick to the strategies of :

• Sticking to the available resource.

• Documenting every phase of the project development.

• Studying and mitigating the risks.

• Staying within the scope of the project.

**2.2 Project Justification**

With this method, the process is lengthy, hectic and can be error prone, which is undesirable for real time applications and hence requires a new methodology to mitigate the problems with this current system.

This current system has a huge scope for improvement, using techniques such as data processing, we can easily compute the output ideal operating values of the solar panel by just feeding the tabulated operating values.

**2.3 Feasibility Analysis**

Project Estimate

Some of the factors that impact the financial cost of the project are :

• Software cost.

• Worker cost.

• Hardware cost.

The project will be developed in stages with each stage being able to produce one working component of the final system. The estimated time to complete the complete project is approximately 4 months.

Risk Analysis

The potential risks that could be involved in moving forward with the project are :

• Missing data will affect the results.

• A database crash would affect the entire system.

The risks that come with not developing the project are

• Inaccuracy results

• Cumbersome process

**2.4 Scope**

The following comes under the scope of this project

• A software that can interact with the dataset

• A software to process the dataset to compute ideal operating values.

• A software that can cleanse the dataset to reduce the possibility of errors.

• A software that can display the necessary output values.

The following does not come under the scope of this projects

• Multiple datasheet input.

• Output Comparison.

• Graphical Output Representation

Since this is an iterative project development approach the scope is ever changing as per the

requirements. The scope can grow or shrink to be able to complete the main objective with

maximum effectiveness.

**Analysis**

**4.1 Requirement Gathering**

**Document Analysis:**

|  |
| --- |
| **System Observed: IV** |
| **How It Works:**  **Step 1: The system has to be set properly for better results.**  **Step2: Prepare the module**   * **Expose the module to a minimum of 5KWh to reduce the effect of initial photon degradation, later cool the module.** * **Set the module on a tilt table, connect the module and take required I V curves when the module is at or below 25°C.** * **Verify the reference cell that is to be used in the test.** * **Ensure an operational check thermocouple that is to be tested.** * **Record the check in a log book.**   **Step3: Conduct the experiment**   * **Mount the module on a tilt table and connect the reference cell leads to the irradiance port of the curve tracer.** * **Connect the RS 232 cable from the Day Star Tracer to the computer.** * **Make other necessary connections to the curve tracker.** * **Ensure that the module operates at an ideal temperature value.** * **Note the I V values from the curve tracker.** * **Connect a calibrated DMM directly to the leads of the primary reference cell to read its voltage output.** * **Record all the readings on the cross check log book.** * **Download the curves from the laptop to the TUV PTL server.** * **Use manual computation strategies to compute the ideal operating values of the module taking the I V curve values as the input.**   **Step 4: Prepare the data**   * **The Data is downloaded from the computer and then it is uploaded to the TUV-PIL server.** * **The naming convention is used to save the data separately.** * **Do not delete any data from the hard disk of the laptop.**   **Step 5: After the data is transferred into an excel sheet, we manually compute the module’s ideal operating value.**  **Step 6: We use formulas for computing the database such as,**   * **Total irradiance:**      * **Adjust the reference cell calibration constant using:**      * **Correct the current at each point of the I-Vdata for irradiance using the following equation:**      * **Translate the uncorrected short-circuit current to RC**      * **Translate the uncorrected open circuit voltage to RC**      * **Translate each I Vdata point to RC**      * **Form a table of P verses V0 values by multiplying I0 by V0** * **Calculate the fill factor,FF, using the following equation:** |

**Interview:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Position | Purpose of the  Interview | Rational for  selecting this  person to interview. | Meeting date  and time |
| Ashwini | Lab Worker | • To understand the  experiment.  • To clarify the doubts  regarding the process.  • To find the problems with  the existing system  • To find the potential scope  of improvement. | She performs the  experiment and is  being working on  the experiment for  a long period of time. And  thus have an  immense  knowledge about  the same. | 24th September,2019  at 4:00pm |

|  |  |  |
| --- | --- | --- |
| Questions | Answers | Observations/Notes |
| How do you process the data  after the experiment? | The data are processed using mathematical equations through a manual process. | Manual  process is slow and  error prone. |
| How complex is the existing  system? | The system is fairly complex  especially to compute the final result. | Data processing requires a new simple system. |
| How do you maintain the data? | The data are maintained in a  database format | They use Excel sheet for  storing data. |
| How is the required data  generated? | The required data are generated by studying the values of the solar module on subjecting it to the I V test. | After the IV test is done on the solar panel. |
| What is the accuracy percentage of the existing  system? | There is no definite accuracy  percentage, however there is a possibility of a human error in data calculation. | Unresolved |
| What Values are considered for the experiment? | Some of the values which are  consider as: Isc, Voc, Imp, Vmp, FF, Etc | Uses many values for  calculation. |
| What type of intermediate data is produced during the  experiment | IV curves are initially generated.Corresponding IV values are then generated. | IV curves help us to generate  the required inp ut data. |
| How do you conduct the  experiment? | Subject the solar module and  find the IV values which are  then manually processed to  compute the ideal operating  values. | IV test is done on the solar  panel after which the data is  processed. |
| How much data is required to  improve the accuracy of the  final result. | Several curves are used to  improve the accuracy. | Higher the data ,higher the  accuracy. |
| What is the end goal of the  project? | To find the ideal operating  values of the solar module | To calculate the idea operating values. |
| What do we Infer from the IV  curve | We get the input data of the  solar module required for the  data processing. | It shows how the panel  operates to the IV test and  helps us find the input data. |
| Do we need a visual output | Yes | Yes |
| Why do we need to prepare the module | To reduce the effect of initial  photon degradation. | To ensure there is no error  input data. |
| Do we clean the data before  processing | No | No |

**4.2 Requirement Determination**

**Functional Requirements**

• A Software that can input data from the recorded data base.

• The software should be capable of processing the data using the provided formulas

• The software should be able to represent the results in a visual format.

**Non-Functional Requirements**

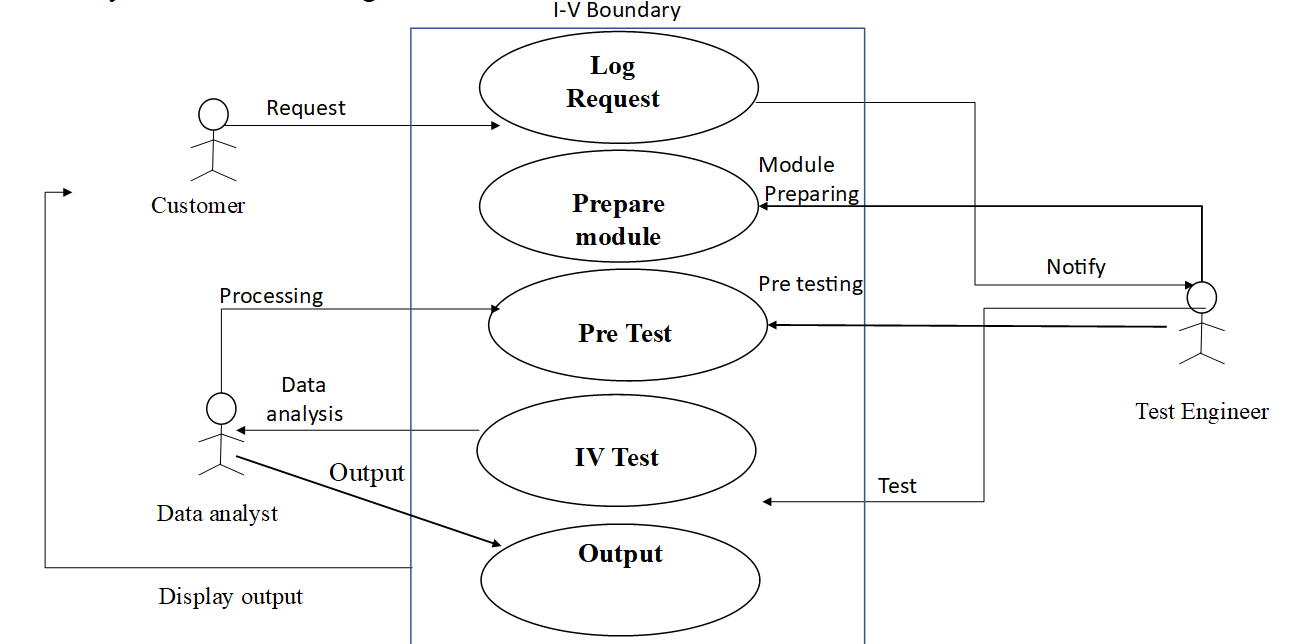
• The output results must be produced without any error.

• The software should be able to normalize the input data to improve the accuracy.

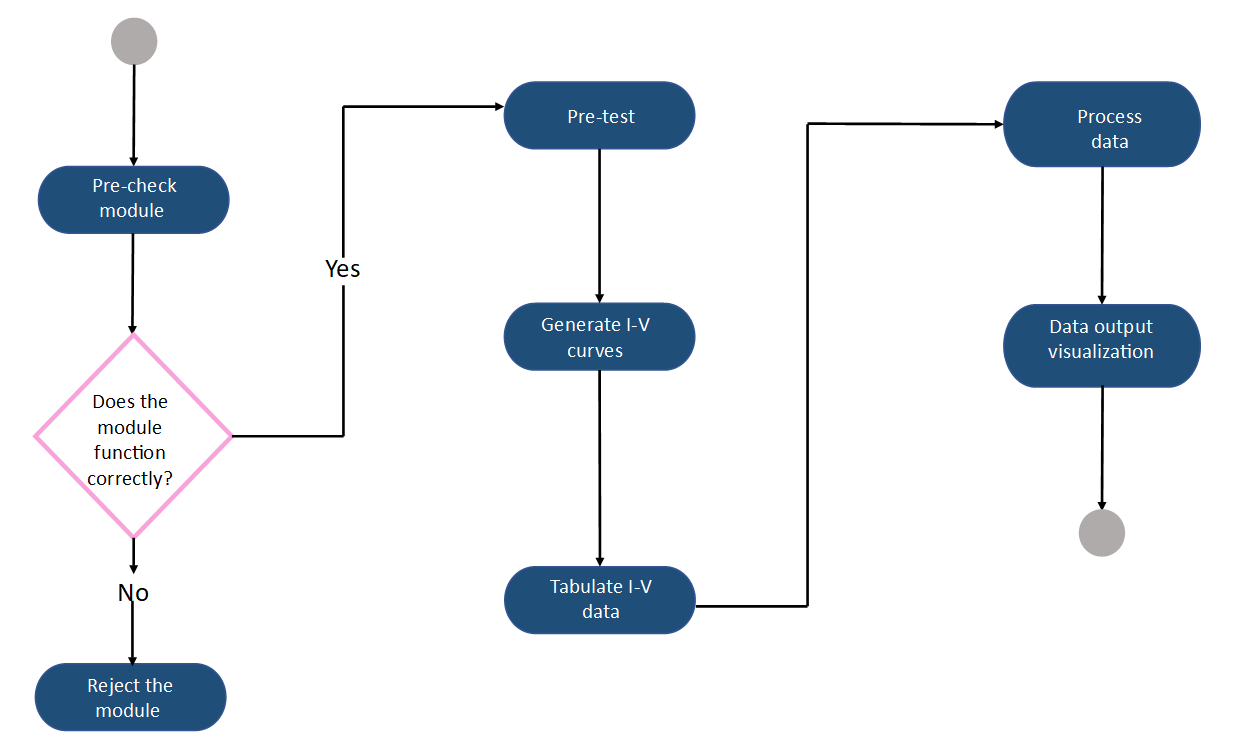
• The software should be secure that it doesn’t allow anyone to change values during computation.

**Design**

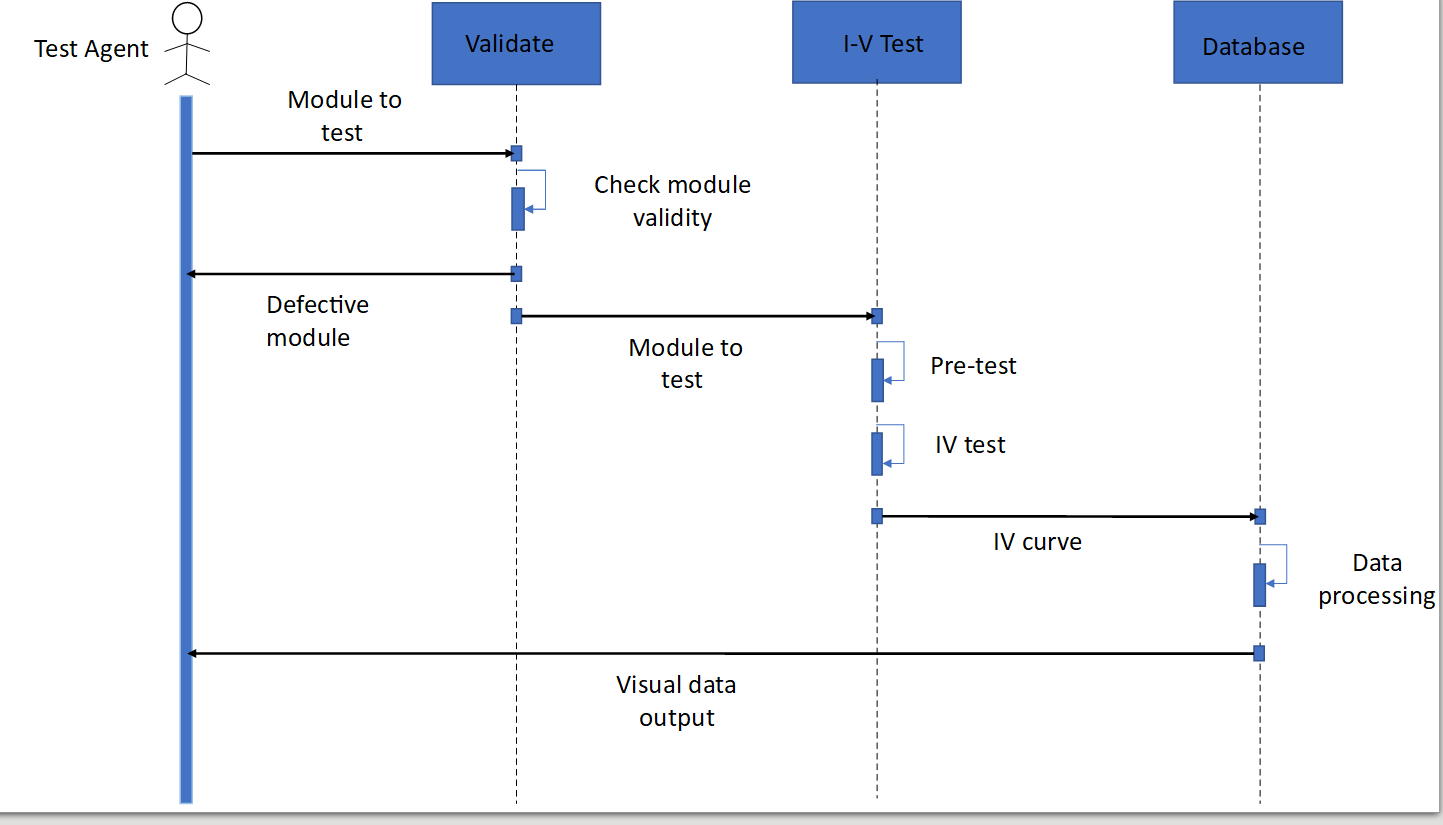
**5.1 Use Case Diagram**



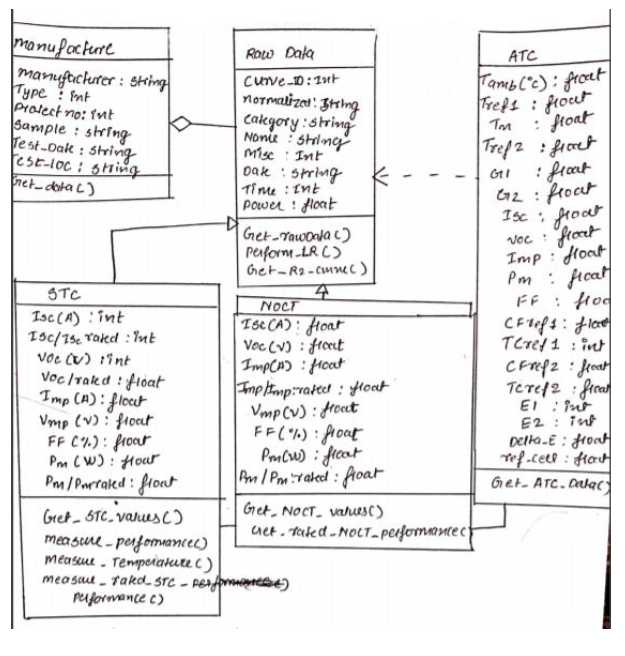
**5.2 Activity Diagram**



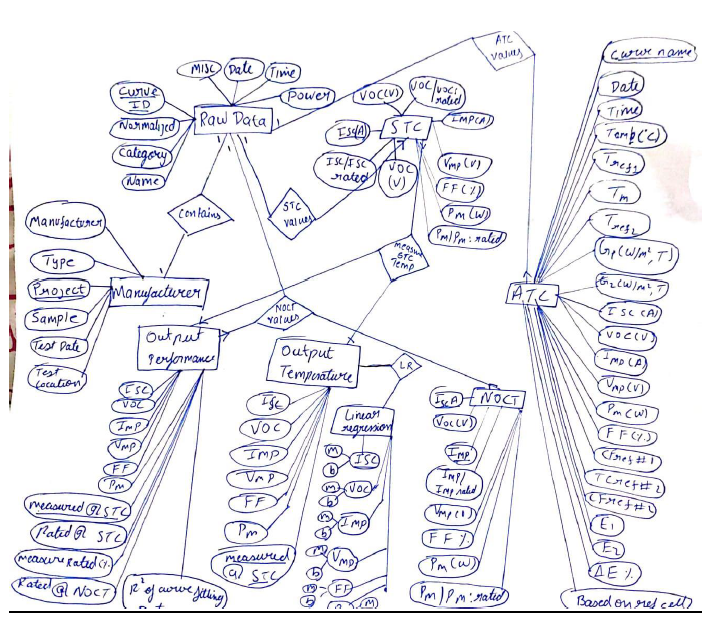
**5.3 Sequence Diagram**



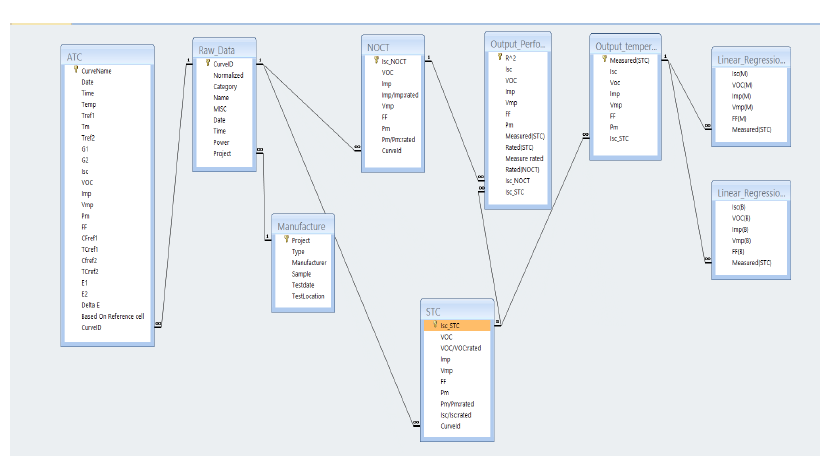
**5.4 Class Diagram**

****

**5.5 Conceptual Diagram**

****

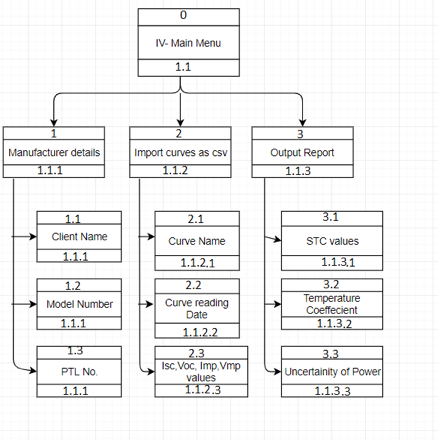
**5.6 Database Schema**



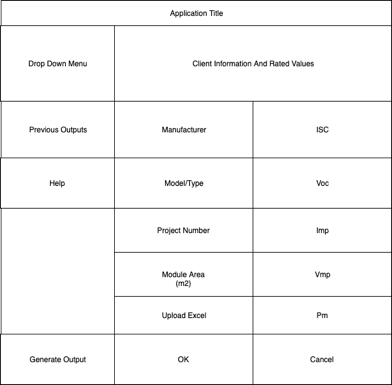
## **Implementation**

### **6.1 User Interface Design**

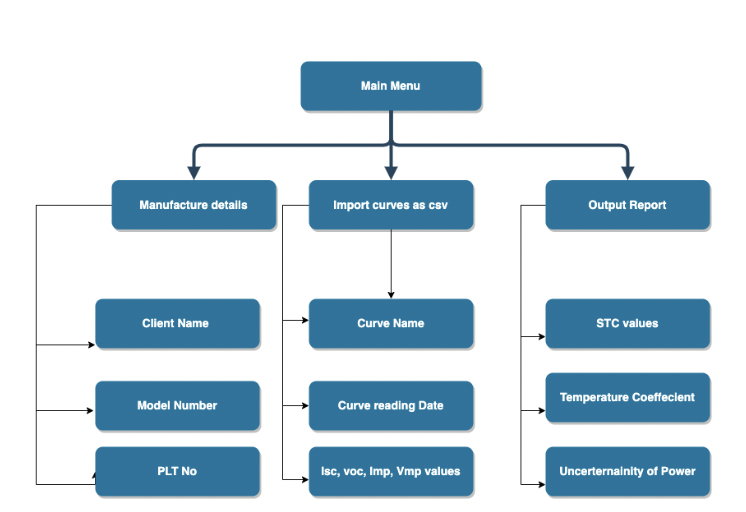
#### Interface structure design

****

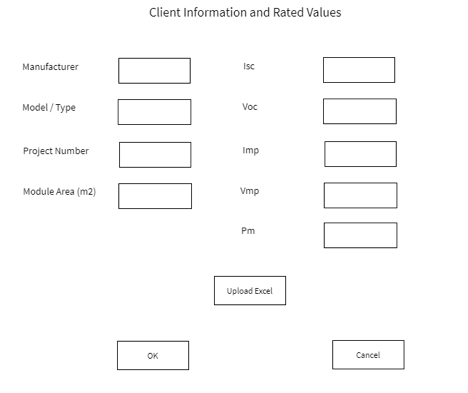
#### **Interface design prototyping**

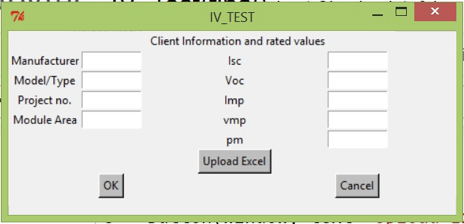
****

#### **Navigation Design prototype**

****

#### **Design Forms**

****



**6.2 Processing System**

User inputs manufacturer details and the data values for calculating the IV in the graphical user interface window. User should upload the csv file containing the values of the I-V curves from the computer using the browse function and then click OK to save the inputs. The output values for Temperature coefficient, Delta Measured and STC values are stored in database tables. Also, an excel report is generated with the output values.

import xlsxwriter

import pandas as pd

import numpy as np

import scipy as sc

from sklearn import \*

from tkinter import \*

from tkinter import messagebox

import tkinter.filedialog

import mysql.connector

from mysql.connector import Error

def CalculateOutput(manufacturerValues,Measured\_STC\_Rated,path):

df = pd.read\_excel (path, sheet\_name='Sheet1')

New\_data = df.values.tolist()

Curvename = New\_data[0]

Date = New\_data[1]

Time = New\_data[2]

CurveTracer = New\_data[3]

Tamb = New\_data[4]

Projectno = New\_data[5]

Trefcell1 = New\_data[6]

Tm = New\_data[7]

Irradiance1 = New\_data[8]

Irradiance2 = New\_data[9]

Trefcell2 = New\_data[10]

Cfref1 = New\_data[11]

cfref2 = New\_data[12]

Isc = New\_data[13]

Voc = New\_data[14]

Imp = New\_data[15]

Vmp = New\_data[16]

Pm = New\_data[17]

FF = New\_data[18]

PR = New\_data[19]

SR = New\_data[20]

Temp\_coefficent = []

Corrected\_i1 = []

for i in range (1,18):

if(FF[i]== 0):

Corrected\_i1.append(0)

else:

c = round(Irradiance1[i] \* PR[2]/( PR[2] + PR[3] \* (Trefcell1[i] -25)),2)

Corrected\_i1.append(c)

Corrected\_i2 = []

for i in range (1,18):

if(FF[i]== 0):

Corrected\_i2.append(0)

else:

c = round(Irradiance2[i] \* SR[2]/( SR[2] + SR[3] \* (Trefcell2[i] -25)),2)

Corrected\_i2.append(c)

Corrected\_i = []

for i in range (0,17):

if (PR[1] == "A825P"):

c = round(Corrected\_i1[i] \*(200.0 /1000.0),2)

else:

c = round(Corrected\_i2[i] \* (200.0/ 1000.0),2)

Corrected\_i.append(c)

x = []

for i in range(1,18):

if(FF[i]== 0):

x.append(0)

else:

c = round(((Tm[i] + 1.5) - 25),2)

x.append(c)

y = []

for i in range (0,17):

c=round(Isc[i+1]/Corrected\_i[i]\*200,2)

y.append(c)

Linear\_Regression\_m = []

Linear\_Regression\_b = []

X = np.array(x)

Y = np.array(y)

#isc linear reg

a = []

a = sc.stats.linregress(X,Y)

Linear\_Regression\_m.append(a[0])

Linear\_Regression\_b.append(a[1])

#voc Linear reg

Y = np.array(Voc[1:])

a = sc.stats.linregress(X,Y)

Linear\_Regression\_m.append(a[0])

Linear\_Regression\_b.append(a[1])

#imp linear reg

Imp\_200 = []

for i in range(0,17):

a = (Imp[i+1]/Corrected\_i[i])\*200.00

Imp\_200.append(round(a,2))

Y = np.array(Imp\_200)

a = sc.stats.linregress(X,Y)

Linear\_Regression\_m.append(a[0])

Linear\_Regression\_b.append(a[1])

#vmp linear reg

Y= np.array(Vmp[1:])

a = sc.stats.linregress(X,Y)

Linear\_Regression\_m.append(a[0])

Linear\_Regression\_b.append(a[1])

#ff linear reg

Y= np.array(FF[1:])

a = sc.stats.linregress(X,Y)

Linear\_Regression\_m.append(a[0])

Linear\_Regression\_b.append(a[1])

#pmax linear reg

Pmax\_200 = []

for i in range(0,17):

a = (Pm[i+1]/Corrected\_i[i])\*200.00

Pmax\_200.append(round(a,2))

Y= np.array(Pmax\_200)

a = sc.stats.linregress(X,Y)

Linear\_Regression\_b.append(a[1])

Linear\_Regression\_m.append(a[0])

i=0

while i < 6:

a = (Linear\_Regression\_m[i] / Linear\_Regression\_b[i]) \* 100

Temp\_coefficent.append(a)

i=i+1

Delta\_measured = []

j = 0

while j < 6:

a = round (((Linear\_Regression\_b[j] - Measured\_STC\_Rated[j]) / Measured\_STC\_Rated[j]), 2)

Delta\_measured.append(a)

j = j +1

try:

connection = mysql.connector.connect(host='localhost',

database='iv',

user='root',

password='root')

if (connection.is\_connected()):

db\_Info = connection.get\_server\_info()

print("Connected to MySQL Server version ", db\_Info)

cursor = connection.cursor()

cursor.execute("select database();")

record = cursor.fetchone()

print("You're connected to database: ", record)

mycursor = connection.cursor()

mycursor.execute("SELECT \* FROM TestData")

myresult = mycursor.fetchall()

for x in myresult:

print(x)

except Error as e:

print("Error while connecting to MySQL", e)

'''finally:

if (connection.is\_connected()):

cursor.close()

connection.close()

print("MySQL connection is closed")'''

mycursor = connection.cursor()

#for i in range(0,7):

man = manufacturerValues[0]

man1 = manufacturerValues[1]

man3 = manufacturerValues[3]

mea = Measured\_STC\_Rated[0]

mea1 = Measured\_STC\_Rated[1]

mea2= Measured\_STC\_Rated[2]

mea3 = Measured\_STC\_Rated[3]

mea4 = Measured\_STC\_Rated[4]

sql = "INSERT INTO ManufactureData (manufacturer, modeltype, modulearea, Isc, Voc, Imp, Vmp, Pm) VALUES (%s, %s, %s, %s, %s, %s, %s, %s)"

val = (man,man1,man3,mea,mea1,mea2,mea3,mea4 )

mycursor.execute(sql, val)

connection.commit()

sql = "INSERT INTO measuredatstc (Isc, Voc, Imp, Vmp, FF, Pm) VALUES (%s, %s, %s, %s, %s, %s)"

val = (Linear\_Regression\_b[0],Linear\_Regression\_b[1],Linear\_Regression\_b[2],Linear\_Regression\_b[3],Linear\_Regression\_b[4],Linear\_Regression\_b[5] )

mycursor.execute(sql, val)

connection.commit()

sql = "INSERT INTO temperaturecoefficient (Isc, Voc, Imp, Vmp, FF, Pm) VALUES (%s, %s, %s, %s, %s, %s)"

val = (Temp\_coefficent[0],Temp\_coefficent[1],Temp\_coefficent[2],Temp\_coefficent[3],Temp\_coefficent[4],Temp\_coefficent[5] )

mycursor.execute(sql, val)

connection.commit()

sql = "INSERT INTO deltaMeasured (Isc, Voc, Imp, Vmp, FF, Pm) VALUES (%s, %s, %s, %s, %s, %s)"

val = (Delta\_measured[0],Delta\_measured[1],Delta\_measured[2],Delta\_measured[3],Delta\_measured[4],Delta\_measured[5] )

mycursor.execute(sql, val)

connection.commit()

print("")

print("Linear\_regression")

print("")

print(Linear\_Regression\_m)

print("")

print("Measured @STC")

print(Linear\_Regression\_b)

print("")

print("Delta\_measured")

print("")

print(Delta\_measured)

print("")

print("Temp\_coefficent")

print("")

print(Temp\_coefficent)

parameters = ['Isc','Voc','Imp','Vmp','FF','Pm']

df = pd.DataFrame({'':parameters,'Measured @STC':Linear\_Regression\_b,'Delta\_measured':Delta\_measured,'Temp\_coefficent':Temp\_coefficent})

#with xlsxwriter.Workbook('test.xlsx') as workbook:

# worksheet = workbook.add\_worksheet()

writer = pd.ExcelWriter('IV\_Output.xlsx', engine='xlsxwriter')

df.to\_excel(writer,sheet\_name='IV',index=False)

writer.save()

#newwin = Toplevel(window)

#display = Label(newwin, text=STCvalues)

#display.pack()

#display2 = Label(newwin, text="Output2")

#display2.pack()

def submitCallBack():

inputValues = []

manufacturerValues = []

Manufacturer\_val = txt\_Manufacturer.get()

Model\_val = txt\_Model.get()

Project\_val = txt\_Project.get()

Module\_val = txt\_Module.get()

Isc\_val = float(txt\_Isc.get())

Voc\_val = float(txt\_Voc.get())

Imp\_val = float(txt\_Imp.get())

Vmp\_val = float(txt\_Vmp.get())

Pm\_val = float(txt\_Pm.get())

FF\_val = float(txt\_FF.get())

Path\_val = txt\_Path.get()

inputValues.extend([Isc\_val,Voc\_val,Imp\_val,Vmp\_val,Pm\_val,FF\_val])

manufacturerValues.extend([Manufacturer\_val,Model\_val,Project\_val,Module\_val])

CalculateOutput(manufacturerValues,inputValues,Path\_val)

messagebox.showinfo( "IV Test", "Output file generated")

def print\_path():

path = tkinter.filedialog.askopenfilename(

parent=window, initialdir='C:/ProgramData',

title='Choose file',

filetypes=[('Excel', '.xlsx')]

)

txt\_Path.insert(0,path)

def helloCallBack1():

messagebox.showinfo( "Hello Python", "file not saved")

window = Tk()

LBL\_ClientInfo = Label(window, text=" Client Information and rated values")

LBL\_ClientInfo.grid(column=3, row=0 )

window.title("IV\_TEST")

lbl\_Manufacturer = Label(window, text="Manufacturer")

lbl\_Manufacturer.grid(column=1, row=1)

txt\_Manufacturer = Entry(window,width=10)

txt\_Manufacturer.grid(column=2, row=1)

lbl\_Model = Label(window, text="Model/Type")

lbl\_Model.grid(column=1, row=3)

txt\_Model = Entry(window,width=10)

txt\_Model.grid(column=2, row=3)

lbl\_Project = Label(window, text="Project no.")

lbl\_Project.grid(column=1, row=5)

txt\_Project = Entry(window,width=10)

txt\_Project.grid(column=2, row=5)

lbl\_Module = Label(window, text="Module Area")

lbl\_Module.grid(column=1, row=7)

txt\_Module = Entry(window,width=10)

txt\_Module.grid(column=2, row=7)

lbl\_Isc = Label(window, text="Isc")

lbl\_Isc.grid(column=3, row=1)

txt\_Isc = Entry(window,width=10)

txt\_Isc.grid(column=4, row=1)

lbl\_Voc = Label(window, text="Voc")

lbl\_Voc.grid(column=3, row=3)

txt\_Voc = Entry(window,width=10)

txt\_Voc.grid(column=4, row=3)

lbl\_Imp = Label(window, text="Imp")

lbl\_Imp.grid(column=3, row=5)

txt\_Imp = Entry(window,width=10)

txt\_Imp.grid(column=4, row=5)

lbl\_Vmp = Label(window, text="vmp")

lbl\_Vmp.grid(column=3, row=7)

txt\_Vmp = Entry(window,width=10)

txt\_Vmp.grid(column=4, row=7)

lbl\_Pm = Label(window, text="pm")

lbl\_Pm.grid(column=3, row=9)

txt\_Pm = Entry(window,width=10)

txt\_Pm.grid(column=4, row=9)

lbl\_FF = Label(window, text="FF")

lbl\_FF.grid(column=3, row=10)

txt\_FF = Entry(window,width=10)

txt\_FF.grid(column=4, row=10)

txt\_Path = Entry(window,width=20)

txt\_Path.grid(column=2, row=12)

Upload = tkinter.Button(window, text='Upload a File', command=print\_path)

Upload.grid(column=3, row = 12)

btn\_Submit = Button(window, text="Submit", bg = "grey", command = submitCallBack)

btn\_Submit.grid(column=2, row = 14)

b = Button(window, text="Cancel", bg = "grey", command = helloCallBack1)

b.grid(column=3, row = 14)

window.mainloop()

**Database code:**

USE IV;

CREATE TABLE ManufactureData

(

serialnumber int NOT NULL auto\_increment primary key,

manufacturer varchar(60),

modeltype varchar(60),

modulearea varchar(60),

Isc double,

Voc double,

Imp double,

Vmp double,

Pm double,

Date datetime NOT NULL DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE measuredAtSTC

(

Isc double,

Voc double,

Imp double,

Vmp double,

FF double,

Pm double,

serialnumber int NOT NULL AUTO\_INCREMENT Primary key,

FOREIGN KEY (serialnumber) REFERENCES ManufactureData(serialnumber)

);

CREATE TABLE temperatureCoefficient

(

Isc double,

Voc double,

Imp double,

Vmp double,

FF double,

Pm double,

serialnumber int NOT NULL AUTO\_INCREMENT Primary key,

FOREIGN KEY (serialnumber) REFERENCES ManufactureData(serialnumber)

);

CREATE TABLE deltaMeasured

(

Isc double,

Voc double,

Imp double,

Vmp double,

FF double,

Pm double,

serialnumber int NOT NULL AUTO\_INCREMENT Primary key,

FOREIGN KEY (serialnumber) REFERENCES ManufactureData(serialnumber)

);

SELECT \* FROM TestData;

Desc testdata;

SELECT \* FROM measuredatstc;

select \* from temperaturecoefficient;

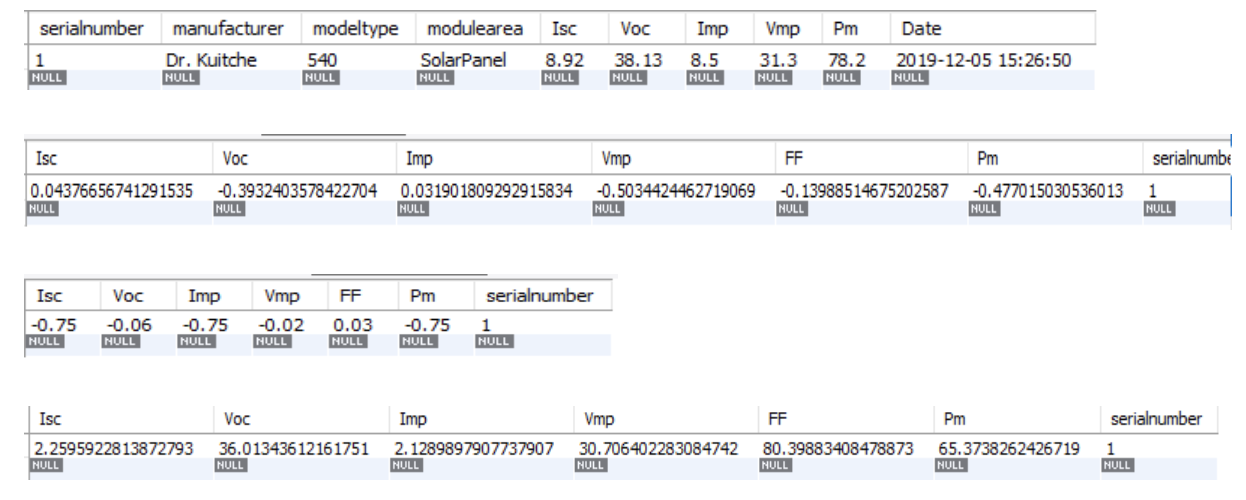
select \* from ManufactureData;

select \* from deltaMeasured;

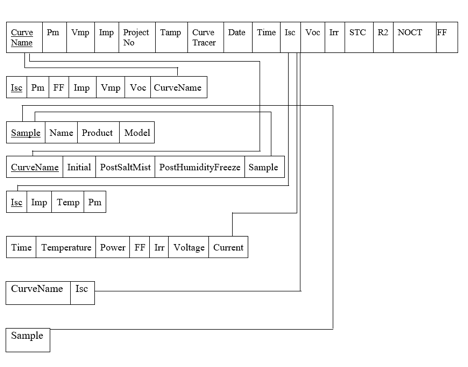
select \* from measuredatstc;

**6.3 Output Design**

The output will be presented in the form of an excel sheet for us to view and analyse



**6.4 Physical Database Design**

****

**Maintenance**

**7.1 Future Features**

The project has opportunities to grow in the future. Additional features such as:

1. Evaluate multiple modules at once
2. Provide a graphic view of the output
3. Compare the module’s output with the provided measure.

**Conclusion**

**8.1 Experience**

The main process of this stage was to develop a user interface which will make it easy for the end user to use the system. By having an interface as this, the user is easily able to enter the desired values into the system and is also able to upload the data file with ease. Apart from the UI design, we also had to connect the main system to the User interface making it possible for the entire system to function seamlessly.

**8.2 Difficulties**

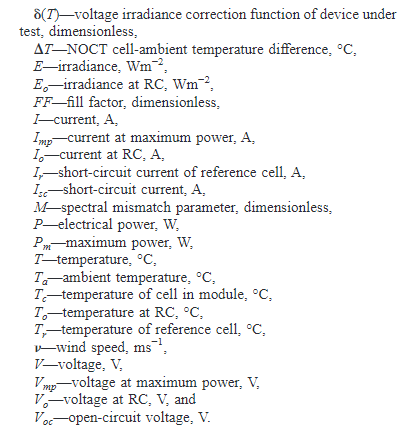
The main difficulties encountered in this stage was to connect the main system to the UI. The task of having various variable be able to correctly interpret the user value whilst validating that only the correct inputs are taken was quite challenging.

**Appendixes**

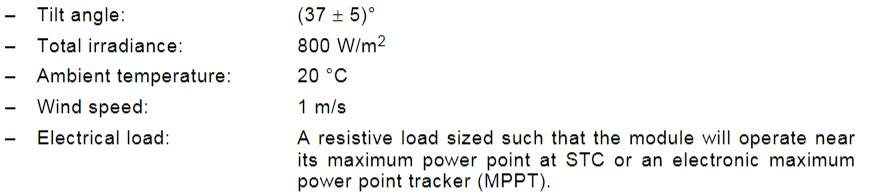
**IV Curve Tracer:**

I-V500w allows field detection of I-V Curve an of the main characteristic parameters both of a single module and of strings of modules for PV installations up to a maximum of 1500V and 10A or 1000V and 15A. For measuring I-V Curve, I-V500w manages an internal database of the modules, which can be updated at any time by the user and comparison between the measured data with the rated values allows immediately evaluating whether the string or the module fulfills the efficiency parameters declared by the manufacturer. I-V Curve can be measured also by decentralizing measurement of irradiation and temperature by using the optional remote unit SOLAR02, using the radio frequency connection (RF) with the master unit. Also for I-V500w, the display at the end of the test of I-V Curve is a clear indication about the compliance with the specifications declared by the panel manufacturer.

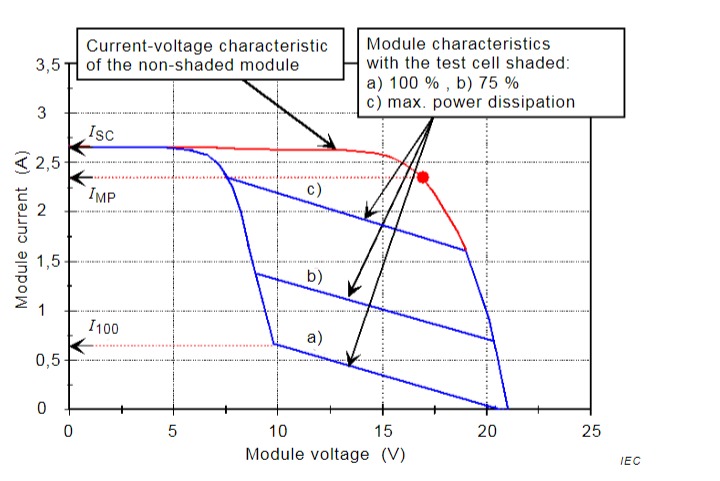
**IV PARAMETERS:**

****

**IV INSTRUMENT SETUP:**

****

**IV CURVE:**

****

**References**

1. http://www.alternative-energy-tutorials.com/energy-articles/solar-cell-i-v-characteristic.html
2. https://www.pveducation.org/pvcdrom/solar-cell-operation/iv-curve
3. https://www.ossila.com/pages/iv-curves-measurement
4. https://www.niser.ac.in/sps/sites/default/files/basic\_page/solar%20cell\_p344.pdf