OBJECT ORIENTED PROGRAMMING

PROJECT REPORT

Submitted To: Ma’am Anum

Submitted By:

Faizan Ejaz

DE-43 Syndicate-A Computer Engineering

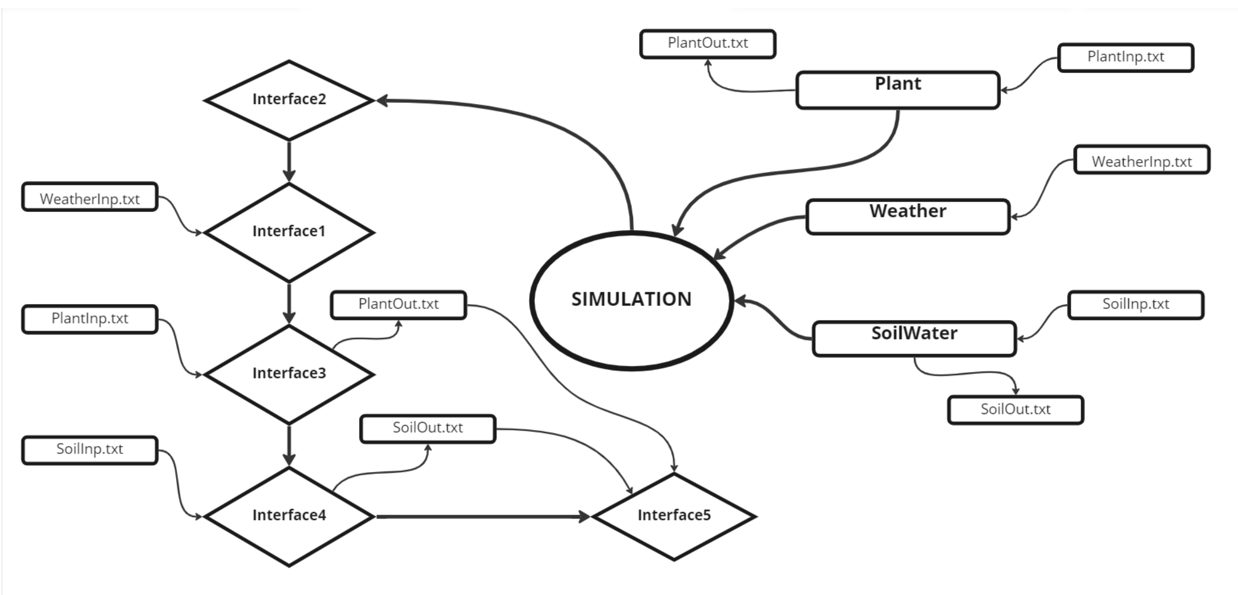
Introduction:

This given program is a good way to simulate the growth of a plant with the aid of a computer. This program enables the user to just input values through GUI and see the calculated growth factors of a plant.

Description:

We were required to simulate a plant growth model using object-oriented programming. For this, we were provided with a tentative solution design. This design had the description of the main problem and some of the formulas required to simulate the growth of a plant were also given. But much of its part was open ended, and we were required to completely simulate the growth of a plant. So, we write the formulas with the help of analysis and internet.

Flow Chart:



Approach:

The main classes were the Plant class, Soil Water Class, Weather Class, and the Simulation Class. We only used association(aggregation) in our program to make the link with Soil Water Class, Plant Class, and Weather Class.

1. **Weather Class:**

- There are 6 variables in this class.

- Four of them are static because their values are to be used in other classes i.e., Plant and Soil Water
- In the initialization function, the file ‘WeatherInp.txt’ is opened, and the values of the attributes of Weather class are read from the file. The file is then closed.

2. <u>Soil Water Class:</u>

- All the required variables from the given table are made attributes of this class.
- In addition, the variables that are being calculated later in the methods of the class are also made attributes, because their values are needed in various parts of the class.
- Other than the main functions like Initialization, Rate Calculations, Integration and Output, there are also other functions.
- These functions are used to calculate various variables. These functions are then called in the Rate Calculation function.
- There are also setters and getters in this class.
- In the initialization function, a file ‘SoilInp.txt’ is opened and the values of various variables are read from the file by clicking on enter button through GUI. Then the file is closed.
- The output function opens 2 files, ‘Soilout.txt’ and ‘IRRIG.txt’.
- ‘Soil.txt’ is opened to write all the calculated variables by clicking on simulate button in GUI.
- ‘IRRIG.txt’ is opened to write the two calculated variables SWFAC1 and SWFAC2.
- SWFAC1 and SWFAC2 are written to another text file because their values are needed in the Plant class.

3. <u>Plant Class:</u>

- Like the Soil Water Class, the Plant class also has the same main functions i.e., Initialization, Rate Calculations, Integration and Output.
- The Initialization class opens two files, ‘PlantInp.txt’ and ‘IRRIG.txt’.
- SWFAC1 and SWFAC2 are read from ‘IRRIG.txt’ file through GUI.
- All the other variables are read from ‘PlantInp.txt’ through GUI.
- Other than the four main methods, this class has other methods like PTS,PGS,LAIS etc.
- In the Rate Calculations function, all the methods that were calculating various variables are called.
- The integration class integrates the rates of change of different variables to give those variables.

- In the output method, ‘PlantOut.txt’ is opened and the calculated variables are written to this file through GUI. Then the file is closed.

4. **Simulation Class:**

   - In this class we have created objects of all classes and then override main methods by calling initialization, rate calculations, integration, and output of each class in respective methods.

5. **Main Class:**

   - In this class we have made object of simulation class and called the methods of simulation class.

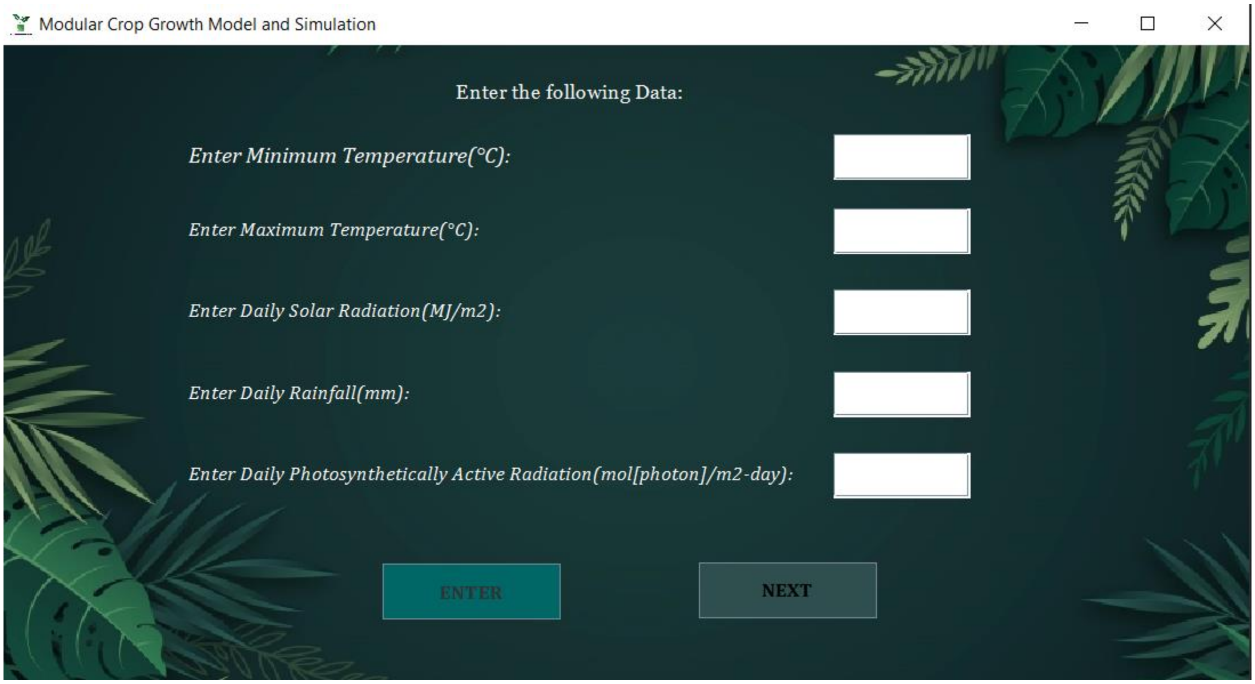
GUI Design:

Interface 2:

In this GUI we have created title page by using JLABELS and and Button named continue which opens next GUIs named Interface1, Interface3, Interface4, and Interface5.

Interface1:

We have made this GUI for weather class in which it takes all input values of initialization method by using JLABELS and JTEXTFIELD and and after pressing enter button we can read them from file then after pressing next button, the next interface opens. We have added background and foreground image from our directory by using Jlabel.

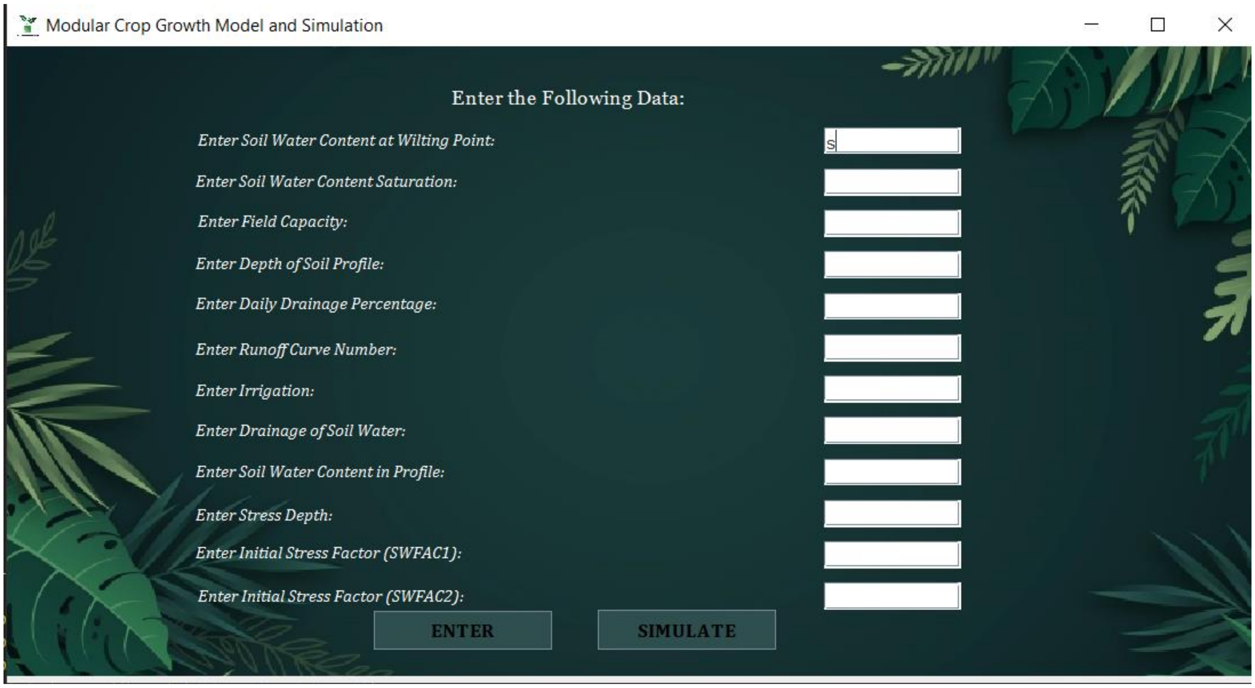


Interface3:

We have made this GUI for plant class in which it takes all input values of initialization method by using JLABELS and JTEXTFIELD and after pressing enter button we can read them from input file then after pressing next button, the next interface opens. We have added background and foreground image from my directory by using Jlabel.

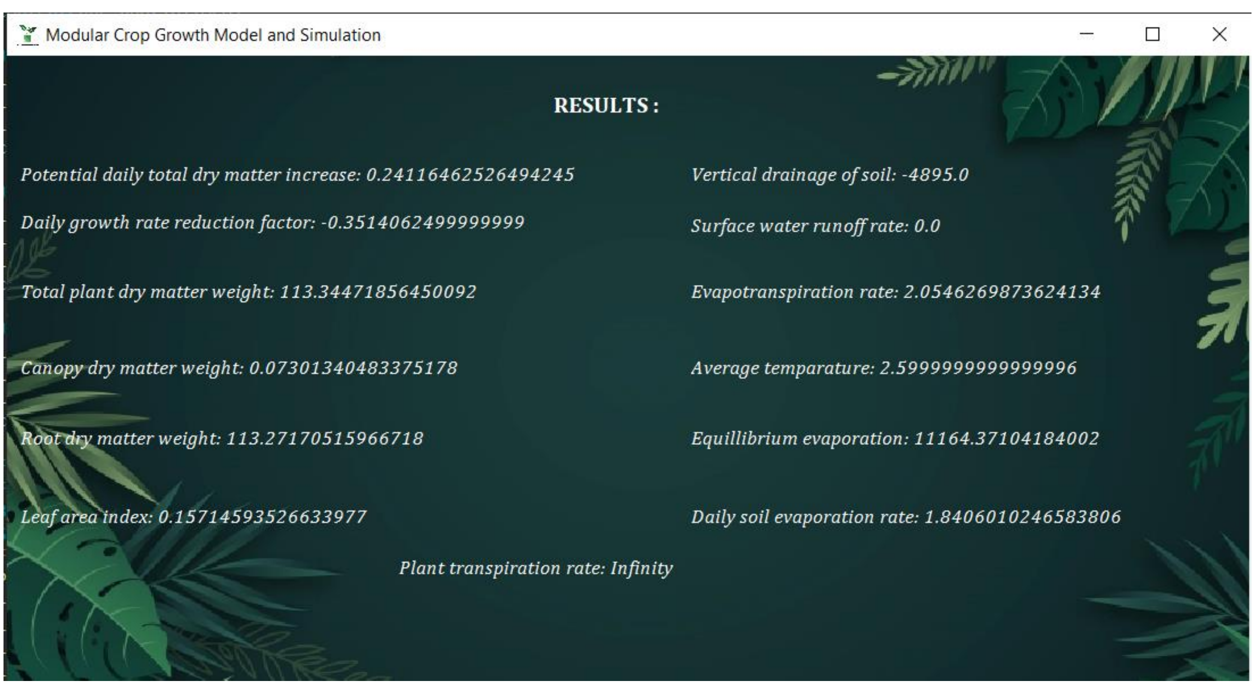
Interface4:

We have made this GUI for soil water class in which it takes all input values of initialization method by using JLABELS and JTEXTFIELD and after pressing enter button we can read them from input file then after pressing simulate button, all the calculated values are written to the output file. We have added background and foreground image from my directory by using Jlabel.

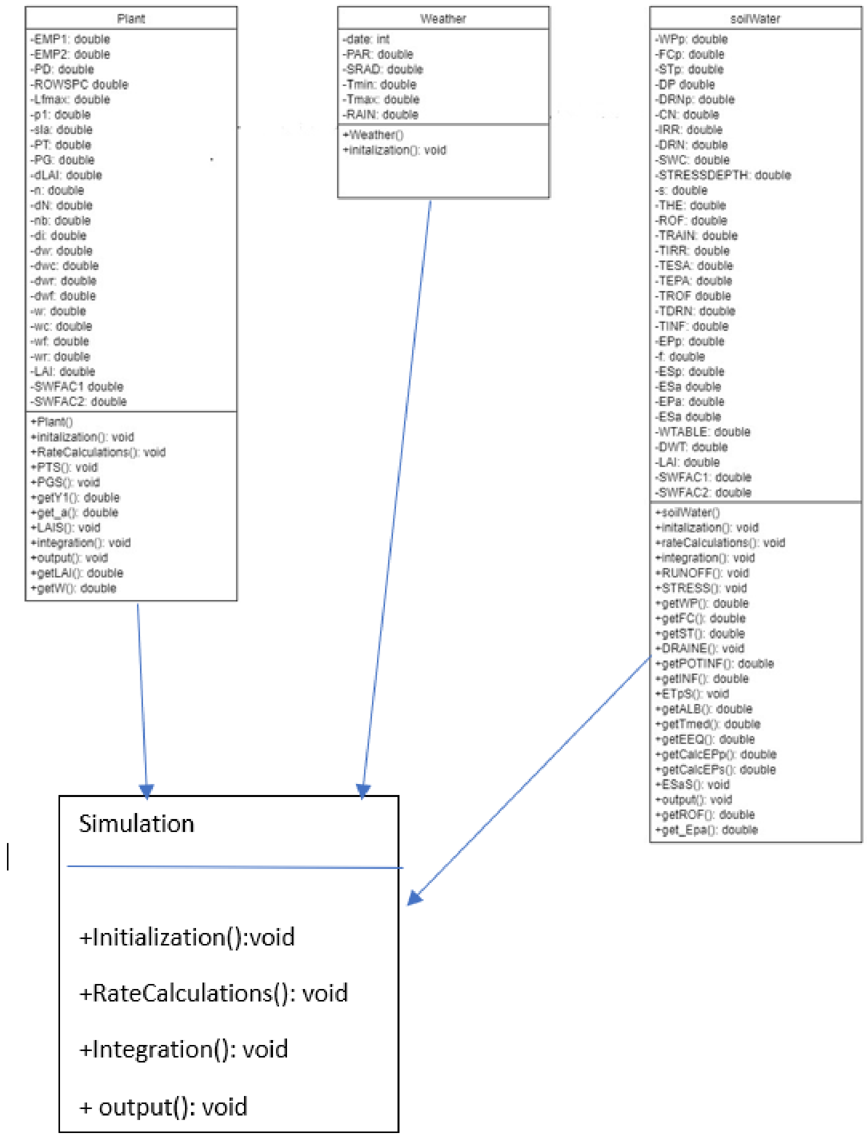


Interface5:

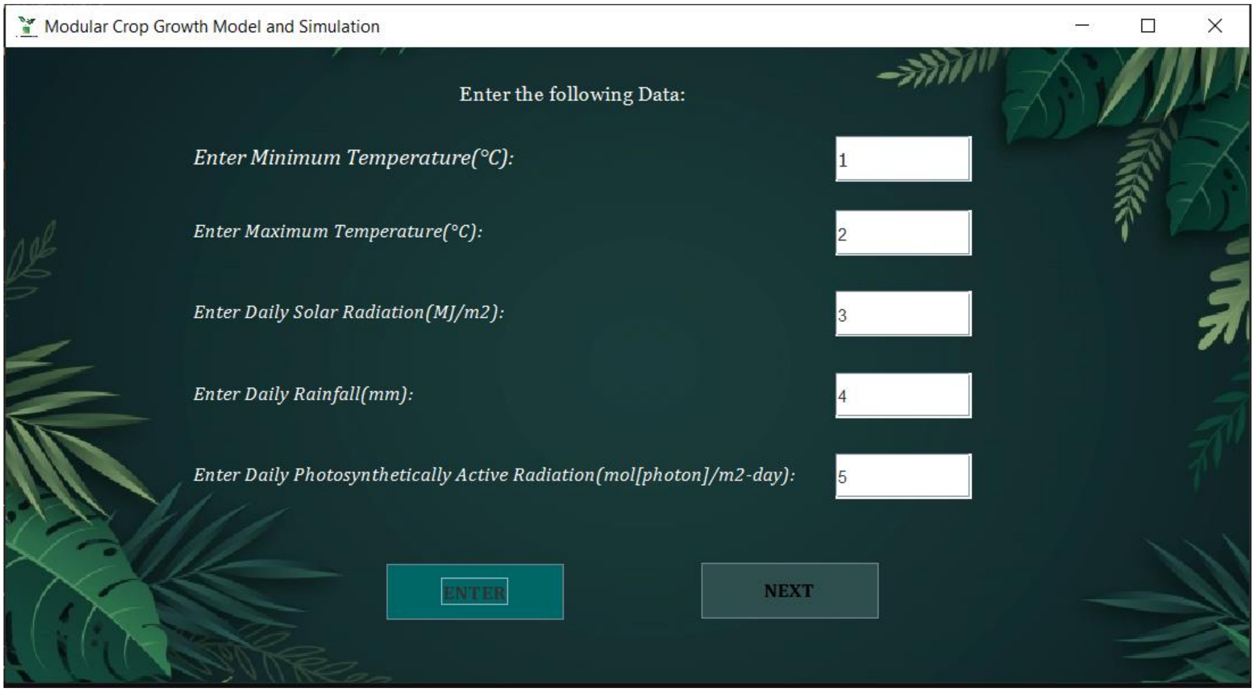
In this interface we have just displayed all the calculated values with the help of Jlabels and Jtextfields.

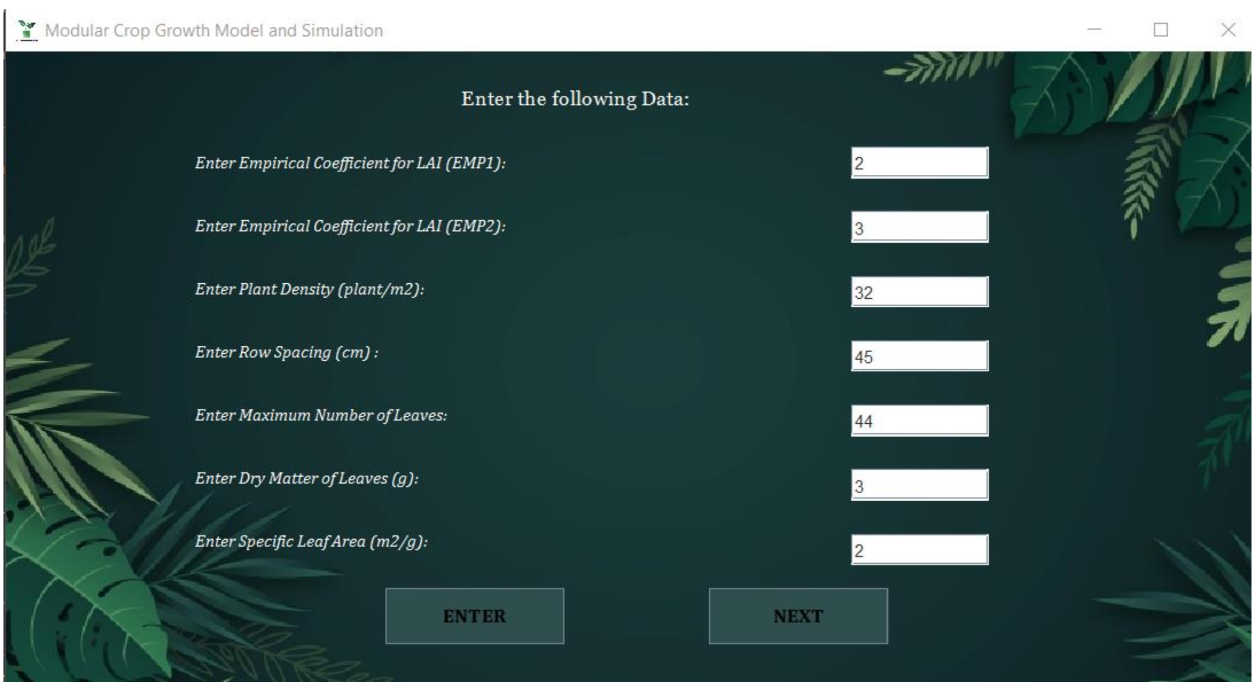


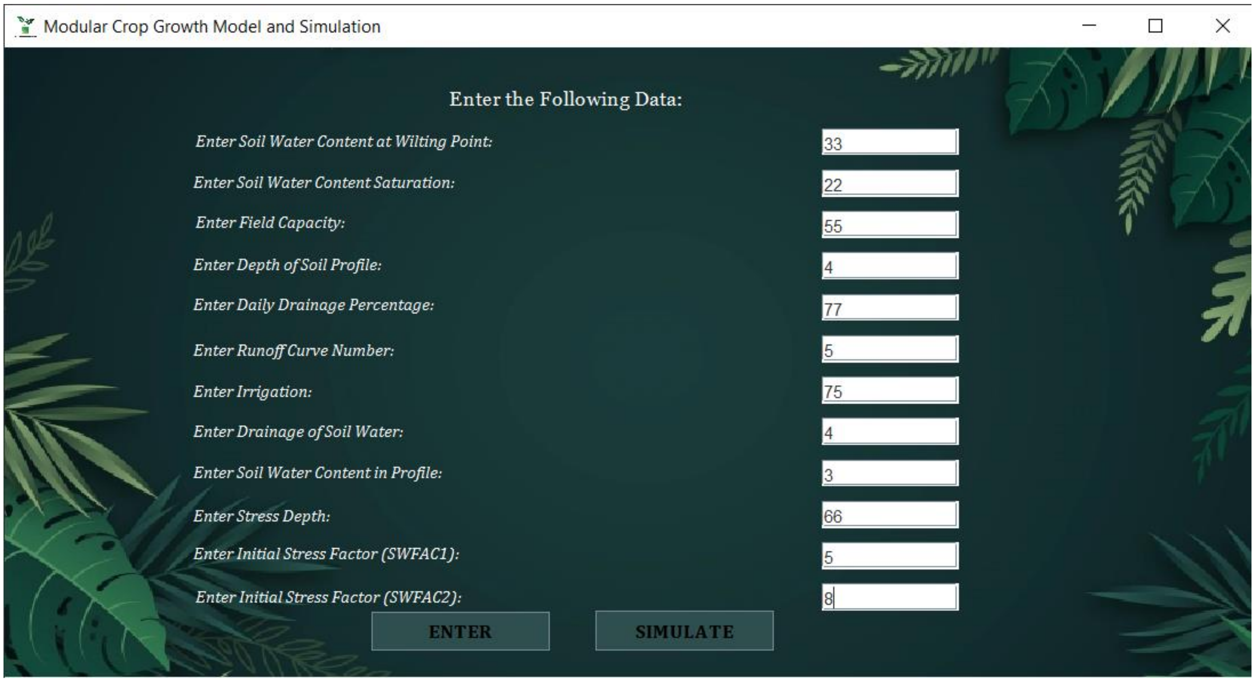
Detailed UML:

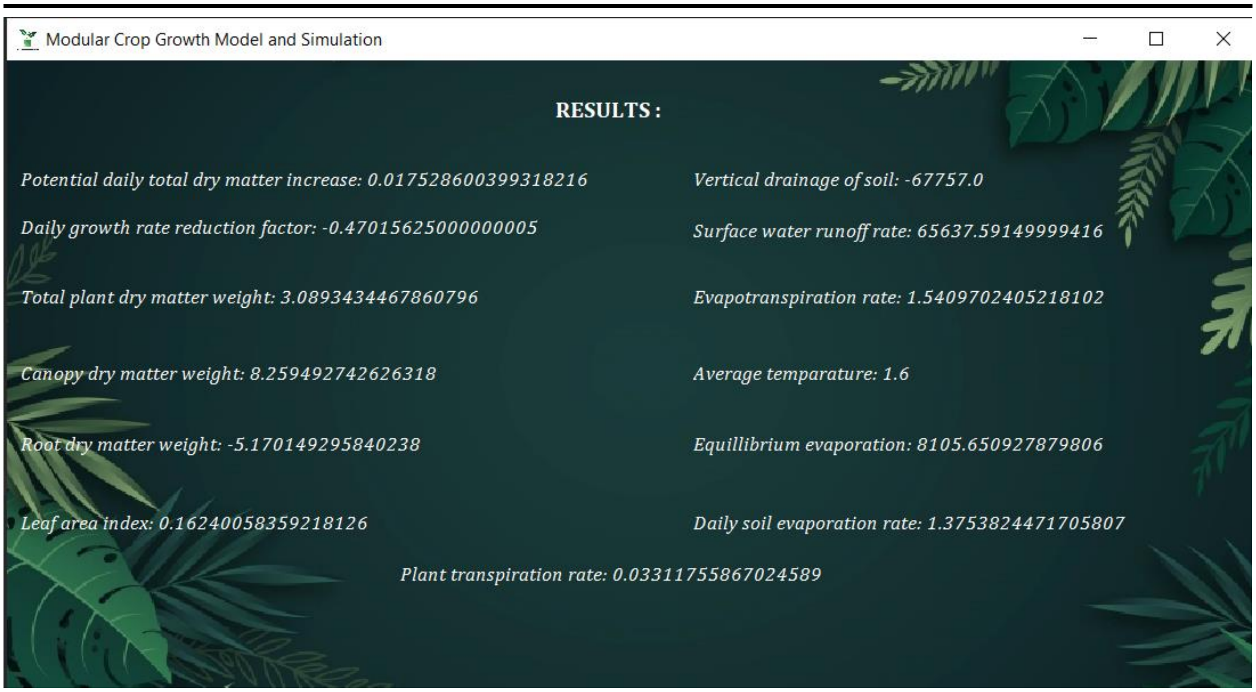


Output:


Modular Crop Growth Model and Simulation
Enter the following Data:
Enter Minimum Temperature(°C):
1
Enter Maximum Temperature(°C):
2
Enter Daily Solar Radiation(MJ/m2):
3
Enter Daily Rainfall(mm):
4
Enter Daily Photosynthetically Active Radiation(mol[photon]/m2-day):
5
ENTER
NEXT


Modular Crop Growth Model and Simulation
Enter the following Data:
Enter Empirical Coefficient for LAI (EMP1):
2
Enter Empirical Coefficient for LAI (EMP2):
3
Enter Plant Density (plant/m2):
32
Enter Row Spacing (cm) :
45
Enter Maximum Number of Leaves:
44
Enter Dry Matter of Leaves (g):
3
Enter Specific Leaf Area (m2/g):
2
ENTER
NEXT


Modular Crop Growth Model and Simulation
Enter the Following Data:
Enter Soil Water Content at Wilting Point: 33
Enter Soil Water Content Saturation: 22
Enter Field Capacity: 55
Enter Depth of Soil Profile: 4
Enter Daily Drainage Percentage: 77
Enter Runoff Curve Number: 5
Enter Irrigation: 75
Enter Drainage of Soil Water: 4
Enter Soil Water Content in Profile: 3
Enter Stress Depth: 66
Enter Initial Stress Factor (SWFAC1): 5
Enter Initial Stress Factor (SWFAC2): 8
ENTER
SIMULATE


Modular Crop Growth Model and Simulation
RESULTS :
Potential daily total dry matter increase: 0.017528600399318216
Vertical drainage of soil: -67757.0
Daily growth rate reduction factor: -0.47015625000000005
Surface water runoff rate: 65637.59149999416
Total plant dry matter weight: 3.0893434467860796
Evapotranspiration rate: 1.5409702405218102
Canopy dry matter weight: 8.259492742626318
Average temparature: 1.6
Root dry matter weight: -5.170149295840238
Equillibrium evaporation: 8105.650927879806
Leaf area index: 0.16240058359218126
Daily soil evaporation rate: 1.3753824471705807
Plant transpiration rate: 0.03311755867024589

WBALOUT - Notepad

File Edit Format View Help

```
Changes in soilwater content are not equal to cumulative inflows and outflows
```

WeatherInp - Notepad

File Edit Format View Help

```
1.0
2.0
3.0
4.0
111
5.0
```

SoilOut - Notepad

File Edit Format View Help

```
Vertical drainage of soil: -67757.0
Surface water runoff rate: 65637.59149999416
Evapotranspiration rate: 1.5409702405218102
Average temparature: 1.6
Equillibrium evaporation: 8105.650927879806
Daily soil evaporation rate: 1.3753824471705807
Plant transpiration rate: 0.03311755867024589
```

SoilInp - Notepad

File Edit Format View Help

```
33.0
22.0
55.0
4.0
77.0
5.0
75.0
4.0
3.0
66.0
```

PlantOut - Notepad

File Edit Format View Help

```
Potential daily total dry matter increase: 0.017528600399318216
Daily growth rate reduction factor: -0.47015625000000005
Total plant dry matter weight: 3.0893434467860796
Canopy dry matter weight: 8.259492742626318
Root dry matter weight: -5.170149295840238
Leaf area index: 0.16240058359218126
```

PlantInp - Notepad

File Edit Format View Help

```
2.0
3.0
32.0
45.0
44.0
3.0
2.0
```

IRRIG - Notepad

File Edit Format View Help

```
5.0
0.0
```

Limitations:

We have not used exception handling and threading in our project.

Mathematical Modeling:

We have made these formulas with help of internet and senior's project.

```
dw = (p1 * n * Math.exp(3)) * (0.05* PG);
     dwc = dLAI / (Math.pow(sla, 2.7)) * (0.4*PG);
     dwr = dw - dwc;
     dwf = PG * PD;

w = Math.abs(Math.exp(4.069) * dw);
     wc = Math.exp(1) * dwc;
     wr = Math.pow(dwc, 0) + w - wc - 1;
     wf = dwf;
     n = dN / 1000;
```

Conclusion:

We have made five classes and five graphical user interfaces in our project and relation between our classes is aggregation.

Our code was simulated successfully. In this project we have learned that how methods of different classes execute by making their object in another class. And how we can enter and display values through graphical user interface by just drag and drop.

References:

- Stack overflow
- Geeks for Geeks
- Senior's Project