**N-Track Online Tool Documentation**

Table of Contents

[**1** **Summary** 2](#_Toc474768310)

[**2** **General Flowchart of the program** 2](#_Toc474768311)

[**3** **Web Interface Flow Chart** 3](#_Toc474768312)

[**4** **Flow Chart for Server Side Process** 4](#_Toc474768313)

[**5** **Description of Files** 5](#_Toc474768314)

[**5.1** **Index.php** 5](#_Toc474768315)

[**5.2** **Mysql.php** 5](#_Toc474768316)

[**5.2.1** **Database fields** 6](#_Toc474768317)

[**5.3** **Register.php** 7](#_Toc474768318)

[**5.4** **main.php** 7](#_Toc474768319)

[**5.5** **daemon.php** 8](#_Toc474768320)

[**5.5.1** **Source Code Location** 8](#_Toc474768321)

[**5.5.2** **The Program (ntracker\_run.m) Structure** 9](#_Toc474768322)

[**5.5.2.1** **Summary** 9](#_Toc474768323)

[**5.5.2.2** **Description** 9](#_Toc474768324)

[**5.5.2.2.1** **ntracker\_run.m:** 9](#_Toc474768325)

[**5.5.2.2.2** **import\_userdata.m:** 9](#_Toc474768326)

[**5.5.2.2.3** **soil.m:** 10](#_Toc474768327)

[**5.5.2.2.4** **weather.m:** 10](#_Toc474768328)

[**5.5.2.2.5** **process\_filex.m:** 11](#_Toc474768329)

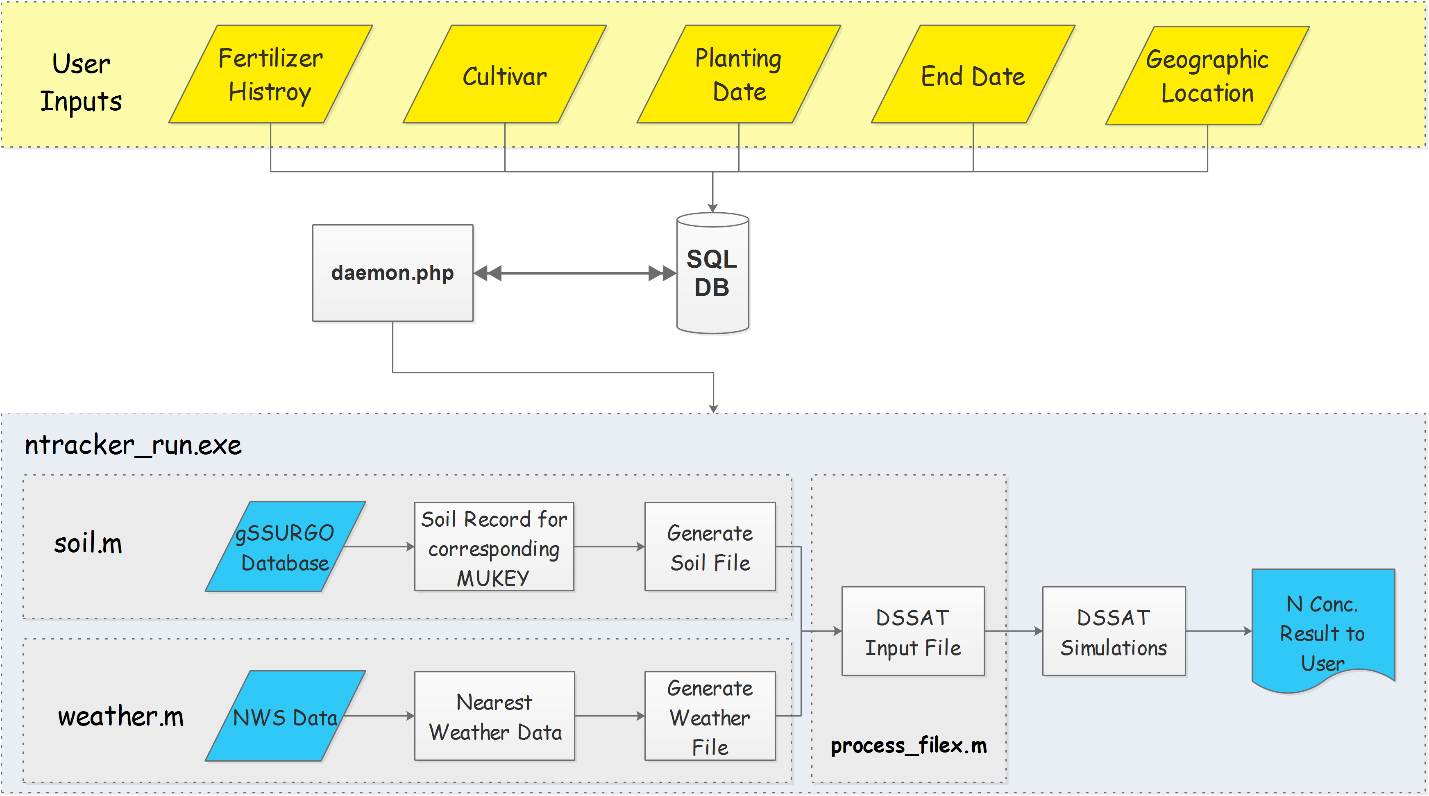
# **Summary**

The N-tracking tool is an online tool for simulating soil nitrogen dynamics for a user-defined location in cornfields within Illinois. DSSAT software, version 4.6, is used for N concentration calculations based on the soil and meteorological data stored on the server. The DSSAT program is run for user specified duration and location, and the average nitrogen concentration within 0-1 and 1-2 feet is calculated. The graphical output web links are sent to the users’ email address.

The overall implementation of the tool can be described as follows:

* + User submits the nitrogen tracking calculation request through the online tool website.
  + Each request is stored in a queue, and the online tool is ready to receive other requests. All the requests are also stored in another requests log table.
  + The queued requests are processed on a First-In-First-Out (FIFO) basis.
  + After the successful processing, the request is deleted from the queue, and an email is sent to the requester with the input parameters and the corresponding nitrogen profile link.
  + After sending the email, another request in the queue is processed. This process repeats until the queue is empty.

# **General Flowchart of the program**

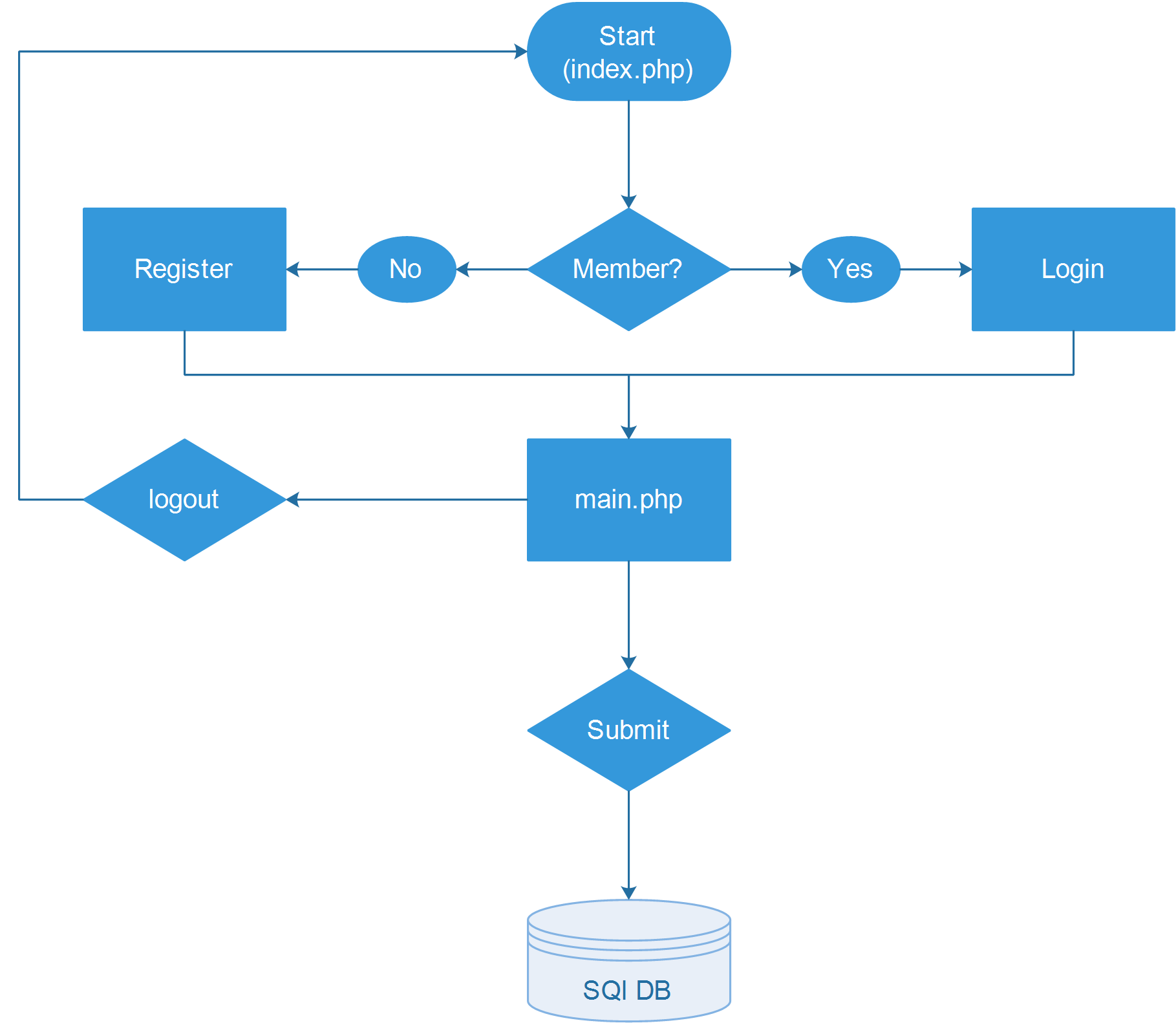


The online tool can be divided into two major functional components:

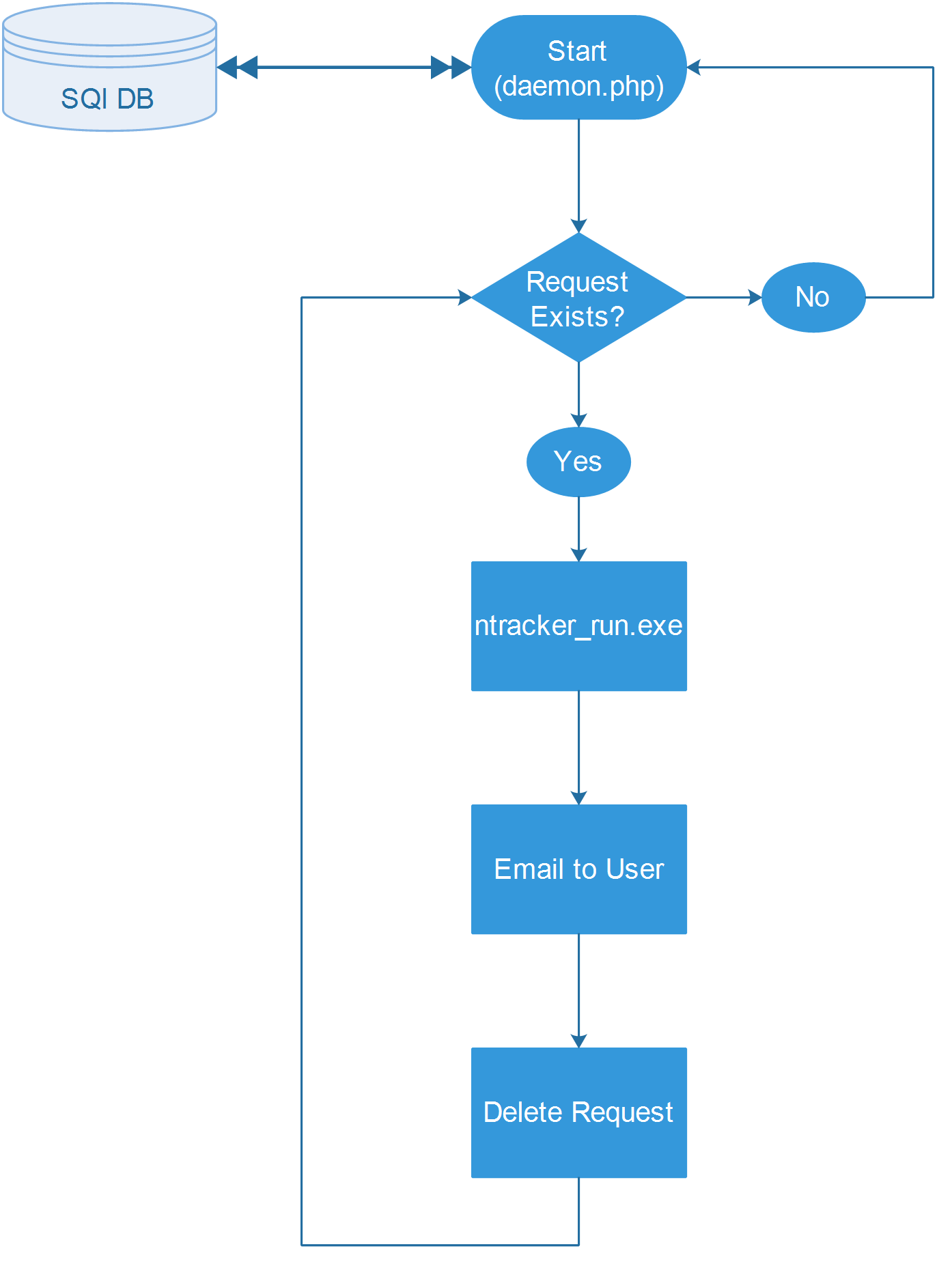
* Web interface
* Background process

The following flowcharts explain the functional flow of the components.

# **Web Interface Flow Chart**



# **Flow Chart for Server Side Process (daemon.php)**



# **Description of Files**

The description of the different files used in the system is as follows:

## **Index.php**

This is the homepage of the online tool. A user can either sign in using an existing user id and password or sign up for a new account. This page directs to main.php after the user clicks on Login button. If Create Account is clicked, the required fields are shown for creating a new account. A successful new account registration will also direct the user to the main.php.

## **Mysql.php**

It contains information about the database in which the existing user information is stored (“$db\_name”). It also contains database login and password. Additionally, names of the tables used to store the parameters submitted by the clients (“$tbl\_name1”) are also included. If daemon.php finds that the table doesn’t exist, it will automatically create one. If you want to change the table structure, you can modify the “daemon.php” file.

$sql="CREATE TABLE IF NOT EXISTS $tbl\_name1 (

timestamp varchar(50) DEFAULT NULL,

email varchar(255) DEFAULT NULL,

latitude float DEFAULT NULL,

longitude float DEFAULT NULL,

month tinyint(4) DEFAULT NULL,

day tinyint(4) DEFAULT NULL,

year int(4) DEFAULT NULL,

myusername varchar(255) DEFAULT NULL,

id int(10) unsigned NOT NULL AUTO\_INCREMENT,

emonth tinyint(4) DEFAULT NULL,

eday tinyint(4) DEFAULT NULL,

eyear int(4) DEFAULT NULL,

ndate date DEFAULT NULL,

nrate int(5) DEFAULT NULL,

ndepth tinyint(4) DEFAULT NULL,

second\_date date DEFAULT NULL,

second\_rate int(5) DEFAULT NULL,

second\_depth tinyint(4) DEFAULT NULL,

third\_date date DEFAULT NULL,

third\_rate int(5) DEFAULT NULL,

third\_depth tinyint(4) DEFAULT NULL,

cultivar text DEFAULT NULL,

rmi int(10) DEFAULT NULL,

PRIMARY KEY (id)

) ENGINE=InnoDB DEFAULT CHARSET=latin1";

### **Database fields**

Following are the fields in database (ntrack\_params and ntrack\_log on \\swsrsetserver) to store the query request parameters:

Timestamp 🡪 for storing time and date of the request.

Email 🡪 for storing email of request submitter. Needed to send the output files.

Latitude 🡪 latitude of the location where N tracking in required

Longitude 🡪 longitude of the location where N tracking in required

Month 🡪 Planting month

Day 🡪 Planting day

Year 🡪 Planting year

myusername 🡪 Username

emonth 🡪 End month for calculations

eday 🡪 End day for calculations

eyear 🡪 End year for calculations

ndate 🡪 First nitrogen application date

nrate 🡪 First nitrogen application rate

ndepth 🡪 First nitrogen application depth

second\_date 🡪 Second nitrogen application date

second\_rate 🡪 Second nitrogen application rate

second\_depth 🡪 Second nitrogen application depth

third\_date 🡪 Third nitrogen application date

third\_rate 🡪 Third nitrogen application rate

third\_depth 🡪 Third nitrogen application depth

cultivar 🡪 Name of the corn cultivar, either selected from the dropdown list or entered in the text field

rmi 🡪 Cultivar’s relative maturity index (RMI) in number of days.

## **Register.php**

This file is called when the user creates a new account through index.php. It performs data checks on the user’s inputs and if everything is correct, the information provided in the signup form is stored in the database for storing the login information. After successful registration, this page directs the user to main.php.

## **main.php**

Using this page, user can submit the N-tracking requests. User provides the simulation parameters including the corn cultivar used, planting and simulation end dates, fertilizer applications information, and latitude and longitude of the desired location. It must be ensured that the following rules are followed or else, the request submission will fail.

**Checks on user input data:**

1. If user choses to add new cultivar:
   1. Cultivar name cannot be blank
   2. Cultivar’s RMI cannot be blank
2. Plant date, end date, nitrogen date, nitrogen rate, nitrogen depth, Latitude, and Longitude cannot be blank
3. Nitrogen application dates should be in mm/dd/yy(yy) format, e.g. 04/14/16 or 04/14/2016.

**Check on Data Ranges:**

1. End date cannot be beyond 365 days of the planting year, e.g. for planting date 04/15/2016, user cannot select end date in 2017 after 05/15/2016, and so on.
2. End date cannot be in the future (after the current date).
3. Fertilizer application cannot be more than six months before planting date for fall applications
4. Fertilizer date cannot be after simulation end date.
5. Latitude and Longitude cannot be outside of Illinois boundary

The location can be selected either from the Google-maps API shown in the form or it can be inserted manually in the text fields. **Note that** **this tool works only for the Illinois region because the server has soil data for the Illinois region only.** After the user specifies the simulation parameters and submits the request, the user’s request is added to the queue. User will be redirected to confirmation.php, and the request will be processed on a First-In-First-Out basis.

## **daemon.php**

This is the heart of the system. It runs as a background process, and can be customized and configured in the task scheduler. It retrieves the user request parameters from the database and begins the execution of the main N-tracking program (ntracker\_run.exe) using those parameters.

### **Source Code Location**

The source codes of the program (ntracker\_run.exe) is located in the following directory on [\\swsrsetserver](file:///\\swsrsetserver):

C:\Program Files (x86)\NMSU\RSET\wwwroot\ntrack\ ntracker.rar

The code may also be downloaded from the link:

// Add a link here (if desired, although not recommended)

### **The Program (ntracker\_run.m) Structure**

#### **Summary**

A batch file daemon.bat located in “C:\Program Files (x86)\NMSU\RSET\wwwroot\ntrack” calls the file daemon.php which in turn calls the program exe file (ntracker\_run.exe) whenever there is a new request in the SQL database to prepare input data, run DSSAT simulation, and sends output back to the user as an email. The source code of the program is located in ‘C:\Users\Administrator\My Documents\MATLAB’ with the main file being ‘ntracker\_run.m’.

#### **Description**

The source code of the exe file contains the following functions. The functions appear in order as they are called:

* Ntracker\_run.m
  + Import\_userdata.m
  + Soil.m
  + Weather.m
  + Process\_fileX.m
    - Cultivar\_exists.m
    - Write\_cultivar.m
      * Calculate\_params.m
    - Get\_index.m
    - Get\_file\_id.m
  + Plot\_dssat.m
    - Jl2normaldate.m

Some of the main functions and their roles are described in the following sections.

##### **ntracker\_run.m:**

This function is the main entry point for the N-tracking program. The function defines hard coded directories for the following:

1. NWS daily weather data (main directory, e.g. ‘E:/2017Data/NWS/’)
2. DSSAT exe file (default is, 'C:/DSSAT46/')
3. Soil file (processed from gSSURGO, currently located on [\\swsrsetserver](file:///\\swsrsetserver) in the directory 'E:/SunY/Soildata/IL\_Soil\_Data\_Base\_Das.txt')

It then passes all the user defined variables to the import\_userdata.m (described below) for preparing data to be used to further steps. The following functions are called by the ntracker\_run.m and are presented in a sequence.

##### **import\_userdata.m:**

This function creates a new structure with all the user data in DSSAT’s acceptable format. User defined input data that was stored in the SQL database is processed into the program useable format and it outputs a structure with the parameters that will be used in later functions. The followings are the processes that are performed in this function:

1. Planting date, end date, and fertilizer application dates are converted to ‘yyddd’ format, where ‘yy’ is the two digits year and ‘ddd’ is the three-digit number for day of the year. For example, 05/15/2017 will be converted to 17135 because May 15 is the 135th day of the year.
2. Nitrogen application rates are converted from lbs/acre to kg/ha.
3. Stores cultivar name, rmi, lat/ long as is in a structure object.

##### **soil.m:**

The soil function has twofold role. In the first part, the function uses lat/long coordinates to locate soil’s Map Unit Key (MUKEY) of the selected location. This is done by utilizing R’s spatial analysis capabilities. MATLAB’s soil function runs the R code to get the corresponding MUKEY. The source code for this is located at 'C:\Users\Administrator\Documents\R\extract\_mukey.R' on [\\swsrsetserver](file:///\\swsrsetserver). This code searches the MUKEY for the provided location.

**Notice: This file needs MUKEY raster which is located on \\swsrsetserver (E:/SunY/Soildata/** **MapunitRaster\_IL\_10m1.tif). In addition, the ArcGIS needs to be installed on the system and the script will need to be update according to the current ArcGIS version installed on the system it is running on. To do that, open extract\_mukey.R with any text editor and change the version according to the installed version on line 38. For example for ArcGIS 10.4, change the ArcGIS10.3 (in red) to ArcGIS10.4.**

**Line 38: rpygeo.env <-rpygeo.build.env(python.path="C:/Python27/ArcGIS10.3",…**

The MUKEY is used to search the soil file for the corresponding soil records, which is then used to generates a soil file (UI.SOL) in “C:\DSSAT46\Soil\”. Not all of the MUKEYs have corresponding records in the soil database, so if the program fails to find a record against a MUKEY, it’ll search for the closest soil record to the user’s location and use that information to write the soil file.

The format and variable explanation in soil file (UI.SOL) can be found here:

<http://rsetserver.sws.uiuc.edu/docs/N%20Tracking%20Project/Soil%20Data/Champaign%20soil%20data%20sample.xlsx>

Next, the program decides the simulation start date. It is based on either planting date or first nitrogen application date. The program sets simulation date to 5 days prior to the planting or fertilizer application date, whichever is earliest. This simulation start date is also used in weather module to prepare daily weather data file for DSSAT simulations.

##### **weather.m:**

Based on the simulation start and end dates for the user defined location, the weather module generates a weather file (UINT1601.WTH) in “C:\DSSAT46\Weather\” for the entire simulation period. For each day of each month of each year within the simulation period, the program reads a csv file ‘yyyy-mm-dd.csv’ inside of each day’s folder and find a record corresponding to the user’s selected location and writes that information to the DSSAT’s weather file, UINT1601.WTH. This continues until the simulation end date is reached.

The file structure and explanation of each variable in the weather file (UINT1601.WTH) is explained in the doc file “DSSAT Input Weather Data Format.docx” that can be downloaded from:

<http://rsetserver.sws.uiuc.edu/docs/N%20Tracking%20Project/Weather%20Data/WEATHER%20DATA%20file%20format.docx>

In addition to soil and weather data, DSSAT requires information about initial soil conditions at the site where simulations are needed to be run. This information however, is not readily available. The N-Track program has collected initial conditions for six sites scattered across Illinois and therefore complete DSSAT input files (FileX) exists only for these six sites. The program is set to use the initial conditions from one of the six sites that is nearest to the user’s location. This can be changed later as more data is available in future. The FileX from the nearest station is used in the process\_filex.m module to update the soil, weather, location information, along with planting date, end date, and fertilizer applications in the FileX.

##### **process\_filex.m:**

This function will update required information in the DSSAT’s input file (FileX). The information includes cultivar ID and name, soil and weather information, lat/long, fertilizer application, planting date, and simulation starting date among other. The output from this function is a file ready to be used in the DSSAT.

The processes performed in this module include:

1. Checking existence of the selected cultivar in the database. If it is a new cultivar, a new record will be added to the cultivar file (C:/DSSAT46/Genotype/MZCER046.CUL).
2. Update the cultivar ID and name in the FileX
3. Update soil and weather IDs in the fileX
4. Add latitude and longitude to the FileX
5. Update fertilizer application(s) dates, rates and depths
6. Update planting and simulation start dates
7. Update initial condition date
8. Update simulation end date (harvest date)

The FileX’s file structure and explanation of variables are provided in the help document that can be accessed from the following link:

<http://rsetserver.sws.uiuc.edu/docs/N%20Tracking%20Project/FileX%20format.docx>

The program calls DSSAT’s exe file to run the simulations using the updated FileX. The output of the simulations are stored in the default DSSAT directory (‘C:\DSSAT46\Maize\’)

##### **Plot\_dssat.m:**

This function reads in the output file ‘SoilNi.OUT'’ from 'C:\DSSAT46\Maize\ and calculates average N concentrations in 0-1, 1-2, and 0-2 ft depths and plots the output in the user’s folder in the ntrack root folder.

Finally, daemon.php generates an email with user’s inputs and a link to the output file and sends the email to the user at their registered email address.