

Report on MCMP proposal:

i-SALUS: a web-based agronomic decision support system to help farmers optimize water and nutrients, increase resource use efficiency, and reduce environmental impact

Bruno Basso

Dept. Geological Sciences

Michigan State University

The objective of the project was to develop i-SALUS, a web-based agronomic decision support system designed to help farmers optimize water, nitrogen and phosphorous on their farms, increase resource use efficiency, and reduce environmental impact. The need for developing a useful and simple interface system like the i-SALUS comes from the necessity of increasing economic net return but at the same time reducing greenhouse gas emissions and groundwater contamination from nitrate and pesticides leaching.

The rationale for bringing SALUS crop model online is the fact that SALUS is not a very computationally demanding program. It can be run for a single spatial point in the field in “real time”. So it's feasible to run the model on the server with proper inputs from users and generate outputs within few seconds.

System Development Methodology and Instructions

The Python web framework Bottle was selected to as a foundation to develop our online system. Bottle is a fast, simple and lightweight WSGI micro web-framework. It's a single file module and has no dependencies other than the Python Standard Library. It has a clear-cut modular way for developers to implement a web-based system. Within this framework, Python code is used to take in user inputs from front-end browser-based user interface and generate proper input files and parameters that the model needs. Then the command line based SALUS model (C++ version) is called to run using these input files and parameters. The outputs of the model are all files that are being read by the Python script to produce HTML based outputs for client-side browser display. In order to provide security and future functionality (such a users saving their sessions) the Python module Cork was used to add user login/authentication functions. Cork

provides a simple set of methods to implement authentication and authorization in web applications running in the Bottle environment.

On the client side, JavaScript interactive mapping library Leaflet was used for the main mapping interface. Leaflet plugins for ArcGIS services were also used to read public facing web services from the ESRI corporation. jQuery and AJAX were used to implement user input interaction and model output display. HTML5 and CSS3 techniques were used to develop the main Login page (See figure 1). The user can sign up for the system with username, email and password(See figure 2). Once a user logs into the system, he or she can zoom in and click on any field in Michigan to run the SALUS model (or alternatively they can enter an address). Once the location is chosen, the JavaScript code in the client side will find the ID of nearest of the 450 weather “stations” for that location. Now a 'Set Management' dialog box is displayed for users to select their own specific inputs for the model. These inputs include soil texture, soil depth, soil tillage, plant population, planting date, irrigation and fertilizer application etc.(See figure 3). Users can change these inputs and then click the Run SALUS button to run the model on the server in real time. After the model run, the results will be displayed in the 'Results' dialog box. The major outputs include Yield, N Leaching, N Use Efficiency, CO2 Emission, N Fertilizer Efficiency, N2O Emission, ET, Water Use Efficiency, Yearly Drainage, Yearly Runoff etc.(See figure 4). If the user wants to run the model again for another set of inputs, he or she can just close the 'Results' dialog box by clicking on 'X' at the upper right corner. The 'Set Management' dialog box will be ready for user to change inputs again. If the user needs to run the model for another location, he or she can just click on the map again to select another location. Again the closest weather station will be located and the information sent to server for the model to read weather data for that station later when the user clicks the 'Run SALUS' button.

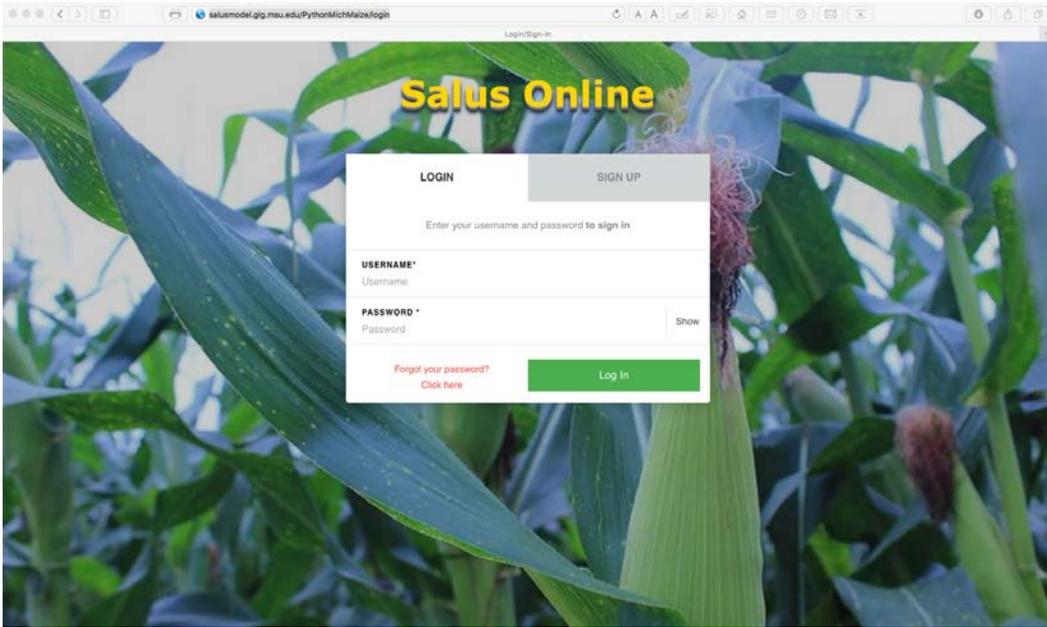


Figure 1. Login Page

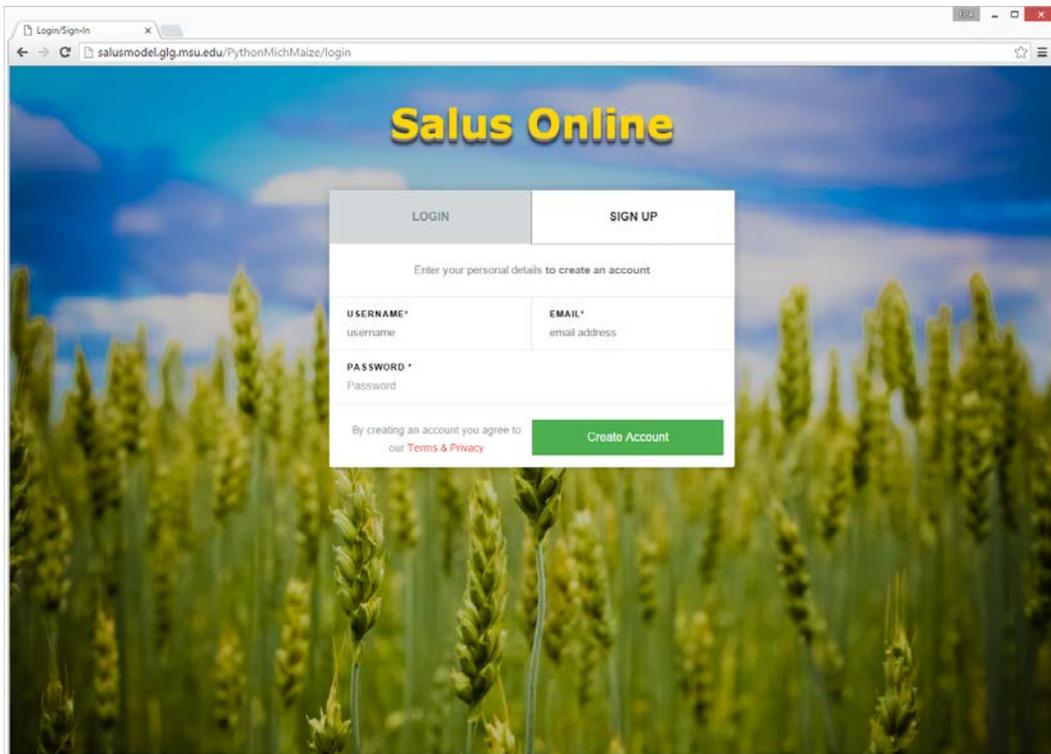


Figure 2. Sign Up Page

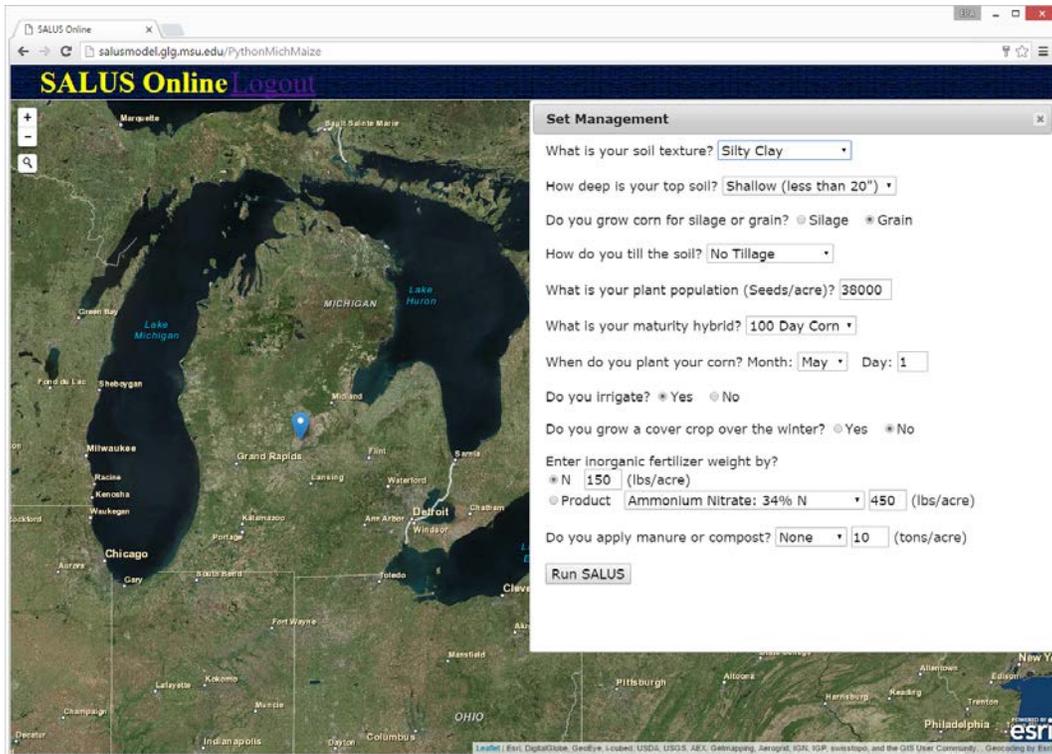


Figure 3. Set Management Dialog Box

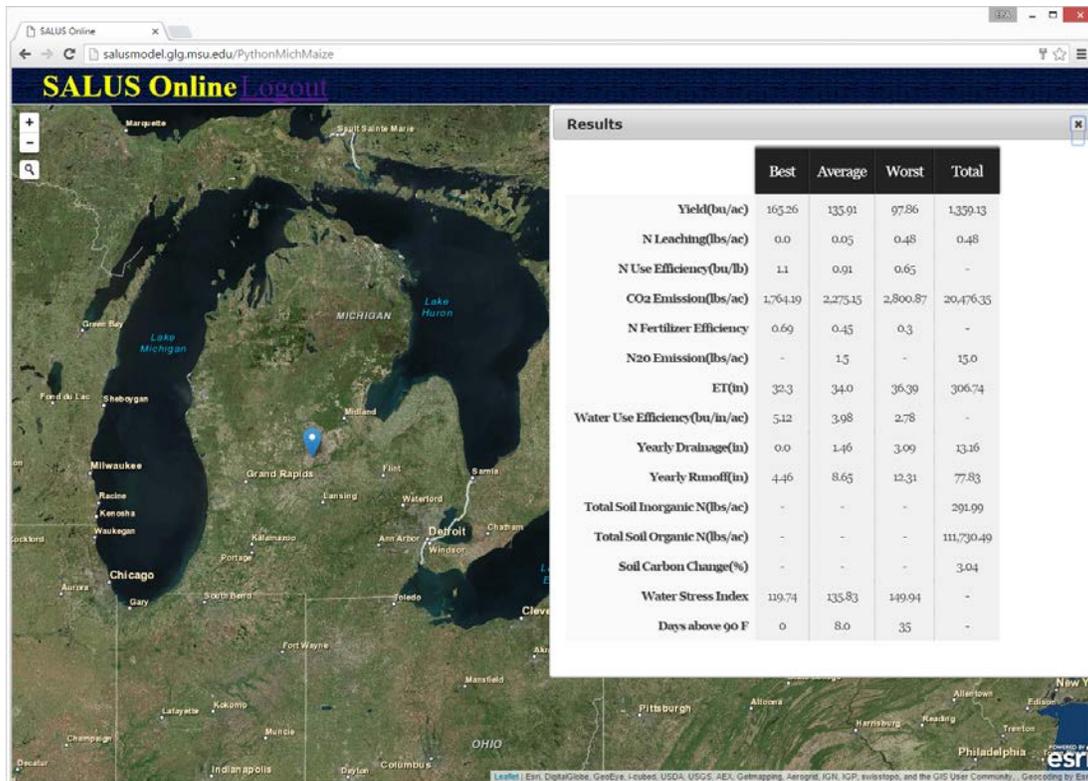


Figure 4. Results Dialog Box