University of Arkansas

Office of Sustainability

Department of Computer Science and Computer Engineering

Environmental Footprint Calculator For Poultry Producers

User's Manual

May 2016

Tables of Contents

1. Introduction 3
2. Capabilities of the Calculator 5
3. Quick Start 7
  3.1 How to Install/launch the Program 7
  3.2 How to Run a Saved Farm Data 10
  3.3 How to Build a Farm from Scratch (the very beginning?) 11
4. Features and Screens 12
  4.1 Home Screen 12
  4.2 Bird Screen 12
  4.3 Feed Screen 14
    4.3.1 Ingredients 15
    4.3.2 Shipping 19
  4.4 Barn Screen 20
    4.4.1 Location and Size 21
    4.4.2 Heating and Cooling 22
    4.4.3 Water 23
    4.4.4 Lighting 24
  4.5 Waste Screen 25
9. Deign and Technical Documentation 26
  9.1 High Level Design 26
  9.2 System Architecture 29
  9.3 Swing Component Layout 31

Figures Content

Figure 1. Folder structure of the program 8
Figure 2. Home Screen shows the program is executed correctly. 9
Figure 3. Run a Saved Farm Data 10
Figure 4. Create a new Project 11
Figure 5. Bird Screen overview 12
Figure 6. Bird Screen variables data types 13
Figure 7. Two sub panels of Feed Screen Ingredients and Feed Shipping Data 14
Figure 8. Feed Screen - Ingredients view 15
Figure 9. Feed Screen - Ingredients variables data types 16
Figure 10. Feed Screen - Shipping view 19
Figure 11. Feed Screen - Shipping variables data types 19
Figure 12. Four sub panels of Barn Screen 20
Figure 13. Barn Screen - Location and Size panel view 21
Figure 14. Barn Screen - Location and Size variables data types 21
Figure 15. Barn Screen - Heating and Cooling screen 22
Figure 16. Barn Screen - Heating and Cooling variables data types 22
Figure 17. Barn Screen - Water screen 23
Figure 18. Barn Screen - Water screen variables data types 23
Figure 19. Barn Screen - Lighting screen view 24
Figure 20. Barn Screen - Lighting screen variables data types 24
Figure 21. Waste Screen view 25
Figure 22. Waste Screen variables data types 25
Figure 23. Essential function and flow of our GUI 26
Figure 24. Program Input Flowchart 27
Figure 25. Program Input variables 28

Tables Content

Table 1. List of Ingredients for Feed screen 16
Table 2. Table of system architecture 30
Table 3. Table of Visual Layouts Components 32

1. Introduction

Chicken farmers must make decisions on their farms that require the use of energy sources, food intake, and production skills throughout their day to produce poultry at a justifiable cost. Generally, chicken farmers must maintain multiple commodities to manage their chickens. These include heating, light sources, food, air circulation, to name a few. Optimizing all of these commodities will help the farmers to reduce the cost of operation while maximizing the production of their poultry. The calculator created through this implementation should allow for poultry industry workers to analyze possible optimizations they can make to producing poultry with a usable user interface. This should allow for poultry farmers and the like to create better goals for themselves regarding their practices.

The US Poultry and Egg Association has implemented a sustainability strategy for the industry based on a continuous improvement framework that has been adopted also by the US pork, beef and dairy sectors. In 2015 the University of Arkansas developed a strategy for implementing this framework. This strategy includes a process-based model that will simulate environmental and economic impacts of real life, on-farm decisions by poultry producers. The model can be used by these producers to help them lessen the impact the industry has on the environment, as well as guide them to making more informed decisions based on various factors. The importance of this problem is wide reaching, given the size of the industry and its impact on the environment as a whole.

The *Environmental Footprint Calculator For Poultry Producers* will use existing algorithms from scientific literature and in software form will need to have a professional user interface that is both easy to use and comprehensive in its delivery of pertinent information.

This program calculates the necessary actions in order maximize the efficiency of the farm, water usage, land usage and operating costs associated with a swine production operation. This farm-level model simulates operations of animals and time at very high resolution. The combination of costing and footprinting capabilities enables economic analyses of strategies to reduce the environmental impacts of swine production. The model is designed to be used by producers and researchers to identify sources of significant environmental issues in the operation, how these impacts are affected by changes in operation procedures and/or hardware, and the economic impact of reductions.

Most environmental calculators in agriculture and industry require that the user input the amount consumed per year for fuels, electricity, water, feed, etc. These calculators then essentially multiply these amounts by impact factors and report the resulting footprints. The *Environmental Footprint Calculator For Poultry Producers* works at a more fundamental level by predicting the yearly use of consumables from a detailed description of the farm including information such as herd size, feed composition, manure handling systems, farm location, barn sizes, and characteristics of the heating/cooling systems. This description of the operation is processed through fundamental models of chickens nutrition, growth and excretion, chemical reactions in the manure handling systems, barn heating and cooling, and water usage to calculate the consumption of utilities and commodities. It then uses these amounts with the applicable impact and price factors to calculate the environmental and cost impacts. In this way it is possible to evaluate operation and management practices that will minimize negative environmental effects from the farm.

This program was written by Capstone students in Department of Computer Science and Computer Engineering at the University of Arkansas. For any technical questions and comments should be addressed to the Ms. Heather Sandefur, *hsandef@email.uark.edu*. Ms. Sandefur has a B.S. in Biological Engineering (2012) and a M.S. in Chemical Engineering (2015) from the University of Arkansas. She is a senior researcher at Paradigm Sustainability Solutions. She has authored several papers and projects in the environmental engineering field and is responsible for the management of research efforts totaling over $1M in combined grants as a researcher for the University of Arkansas.

2. Capabilities of the Calculator

Chicken farmers must make decisions on their farms that require the use of energy sources, food intake, and production skills throughout their day to produce poultry at a justifiable cost. Generally, chicken farmers must maintain multiple commodities to manage their chickens. These include heating, light sources, food, air circulation, to name a few. Optimizing all of these commodities will help the farmers to reduce the cost of operation while maximizing the production of their poultry. The calculator created through this implementation should allow for poultry industry workers to analyze possible optimizations they can make to producing poultry with a usable user interface. This should allow for poultry farmers and the like to create better goals for themselves regarding their practices.

Developed by our very own university, there is also the Pig Production Environmental Calculator (PPEC), which has been mentioned previously in the paper. It was also mentioned that we plan to use this as a loose base for our project. This program is a calculator that offers a predictive model, showing estimation models of various variables in pork production based off of certain characteristics of the operation being modeled. This program was developed at the University of Arkansas, funded by the National Pork Board and the U.S. Department of Agriculture. With the PPEC (Pig Production Environmental Calculator), there are many design flaws as discussed in our meetings with our sponsors:

- There are scaling issues in that you cannot actually change the size of the program's window.
- Each action in the program opens up a new window and destroys the previous one.
- Unable to directly handle (copy, paste) the program's output (any information about the farm system).
- No real walk through for new users; it assumes users know what they need to put in.
- Unable to make quick changes to the model's calculations.
- Confusing UI, buttons are all over the place with no real rhyme or reason.
- Information is non-persistent; information is only kept on the calculation screen, and then thrown away.

The objective of *Environmental Footprint Calculator For Poultry Producers* is to develop software around a model that simulates the environmental impacts of poultry production in the real world . The software will use a computational engine that is currently being developed by the University of Arkansas Office for Sustainability, and our goal is to create a detailed user interface that implements this engine. The back-end computations will be loosely based on an existing software project for the pork industry entitled “The Pig Production Environmental Calculator v2.X”. While the front end (GUI) of the software will resemble a native windows application with full functionality and a ‘wizard’ to help guide the user through the program. The look and feel of the software will be that of a native Microsoft Windows application. It will have internal windows which are re-sizable and manipulable, specifically allowing the user to copy/paste to and from the software. It will provide saving and loading of farm data and output as well.

Before any calculations of footprints or costs are made, a comprehensive model of the overall farm operation is performed in order to characterize its inputs and outputs. Because the model is so detailed in its approach, it is one of the most capable programs of its type. Its modular construction and use of sub-models from the literature enables it to be expanded in scope and/or modified for greater accuracy as new data on animal and farm unit operation response become available from swine researchers.

To run the calculator, a user enters a description of the farm operation that is centered on hardware facilities, herd characteristics and methods of barn and manure system operation. The user does not have to know the amount of feed that will be consumed, the electricity required, manure produced or fuels used, the model predicts these. This approach was chosen so it would be easier for a user to evaluate types of operations that do not yet exist and/or changes to current operations.

The software *Environmental Footprint Calculator For Poultry Producers* should take in various inputs such as poultry breed, ambient temperature and feed composition, as well as specifics for the facility such as number of houses and type of manure management system(s). Through various UI elements we will supply the user with comprehensive data about the model of their facility, and give detailed information about the carbon, water and land footprints these facilities produce. Once the data of the bird, feed, barn, and waste streams are calculated, the model uses these data to make decisions on their farms.

3. Quick Start

3.1 How to Install/launch the Program

The *Environmental Footprint Calculator For Poultry Producers* can be run on different operating system such as Windows or Linux (Ubuntu, Linux Mint, …). No additional software is needed to run this program. In order to make the program more efficient, we created several executable files. Those are *cpp_executable_unix.exe* and *cpp_executable_win64.exe*.

There is no need to run any installation programs or create any special system files. The model is not copy-protected and does not employ any form of digital rights management. It is free to use. The model can be run from a flash drive, an external hard drive or any writable media (Except DVDs). An internet connection is not required to run the calculator. However in order to receive updates – including more recent estimates of prices –an internet connection will be needed just to download the updated files. Inside the main folder there are three items:

- The program file *cpp_executable_unix.exe* or *cpp_executable_win64.exe*. Double-click this file to start the software.
- A folder called inputfiles. This is where the program saves all files and later finds farms, barns and feed mixes that have already been created.
- A user's manual file that includes how-to of the program as well as technical design of the program

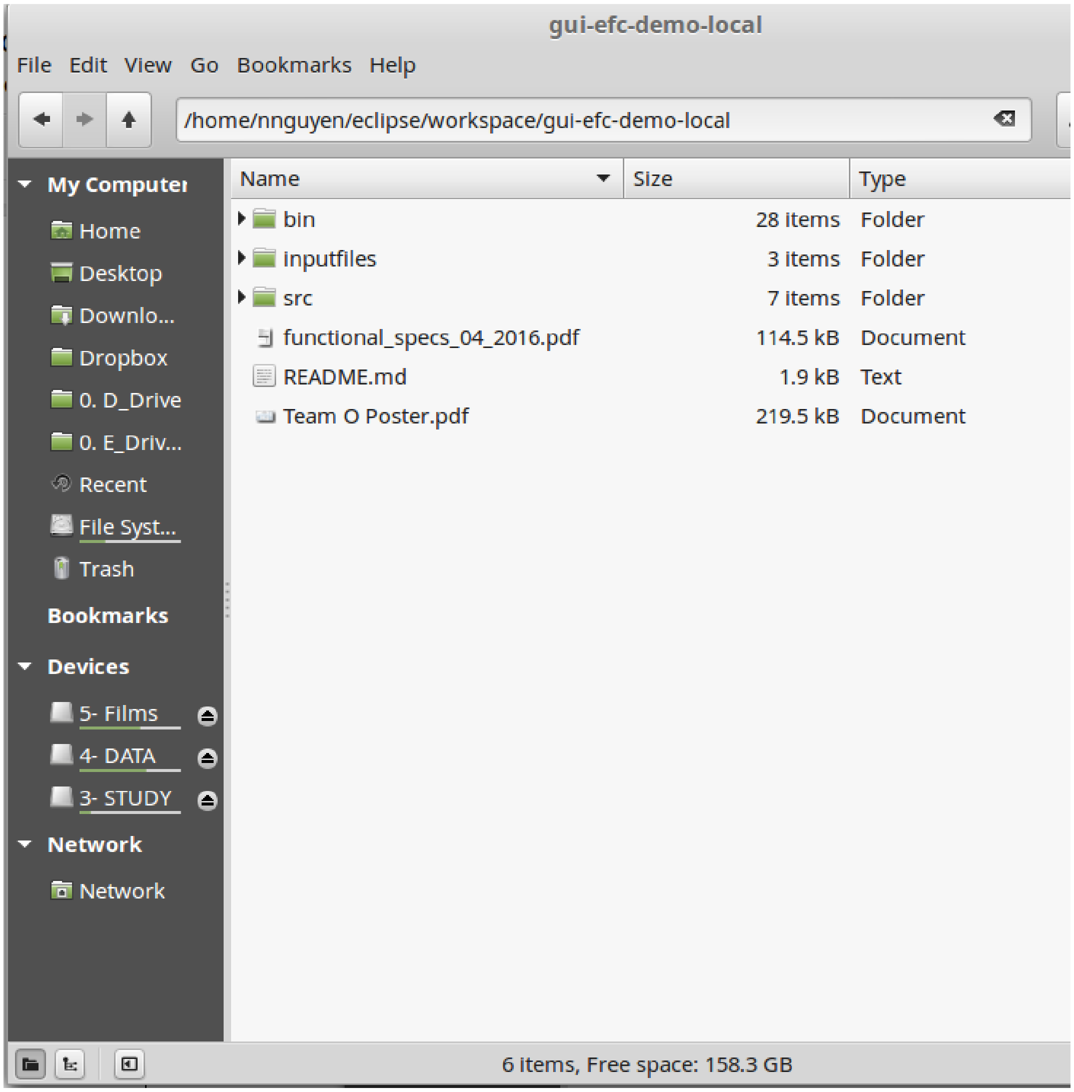

Figure 1. Folder structure of the program

In order to run program, double-click on cpp_executable_unix.exe or cpp_executable_win64.exe depending upon the operating system you are using. When you see the Home Screen (shown below) appears, then you could know the program is executed correctly

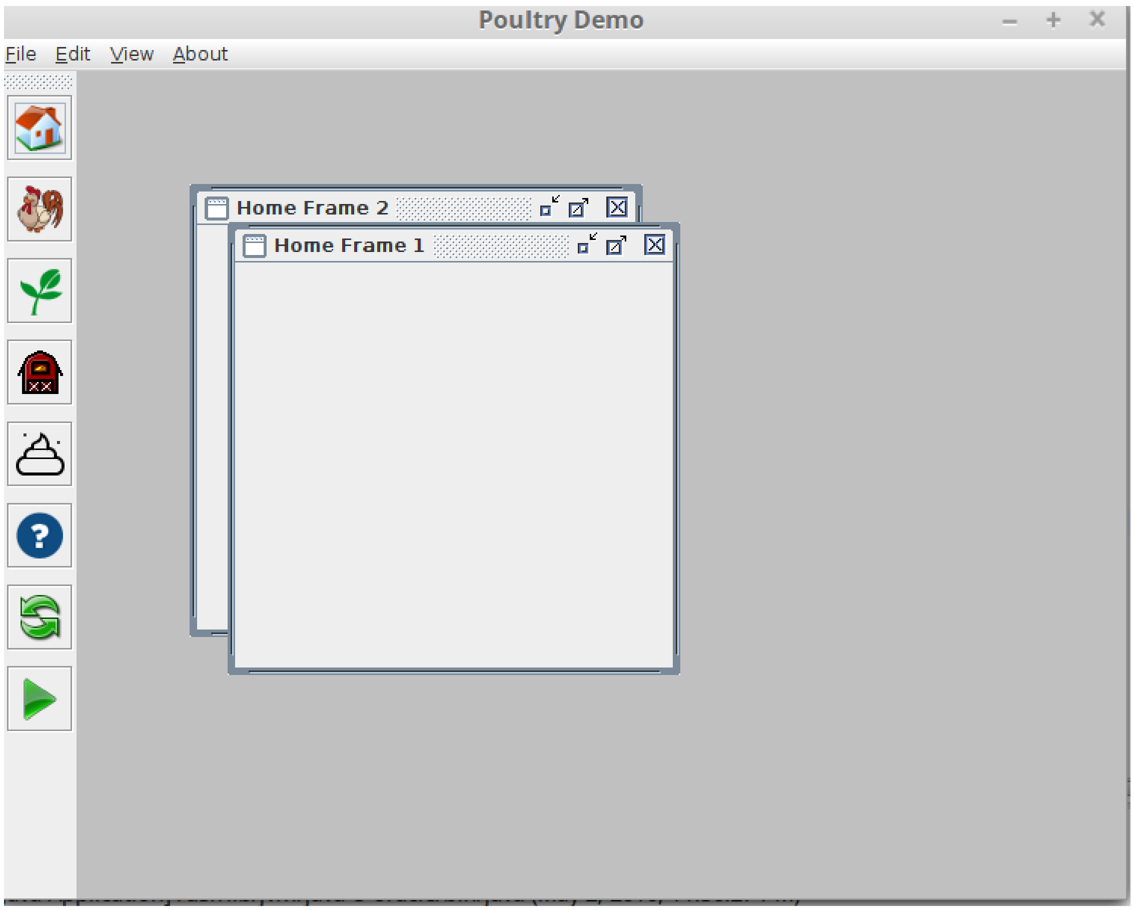

Figure 2. Home Screen shows the program is executed correctly.

3.2 How to Run a Saved Farm Data

The users could run a Saved Farm Data by loading the pre-existed data that was stored in the storage. This file could be created as a sample of the program or could be created by some other users for the purpose of using again. In other words, the user can store and retrieve a complete set of inputs and outputs for a farm. This section shows how to load an example farm that has already been saved, run it, and view the outputs.

From the Home Screen of the Program, we run menu File, then choose Load Project, then the Open project panel will appear. Please direct to the folder inputfiles if it was not automatically open on your machine. The following screenshot would appear and the users could choose from the input file (xml file)

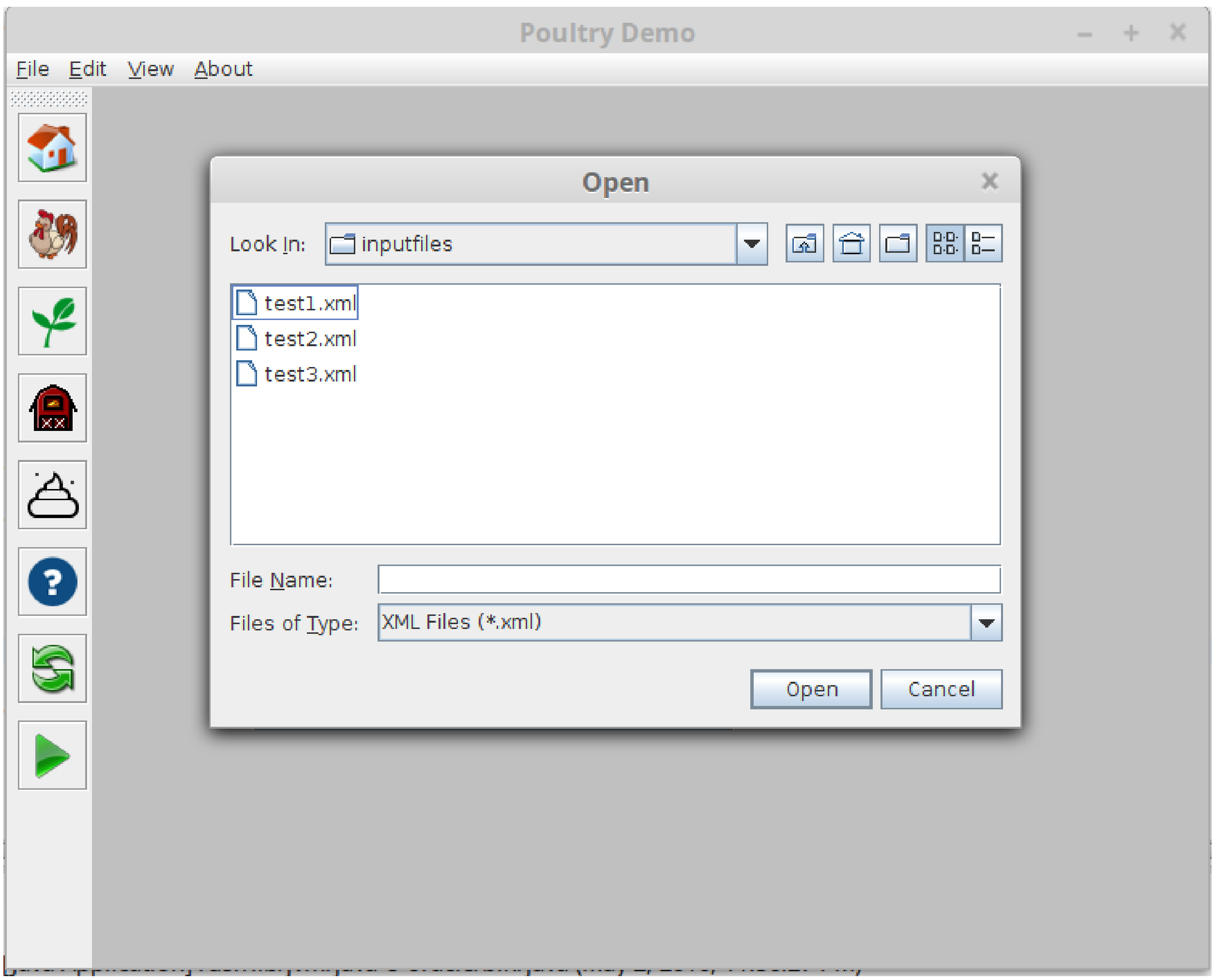

Figure 3. Run a Saved Farm Data

3.3 How to Build a Farm from Scratch (the very beginning?)

In case the program is using the wrong data or using the pre-load input data from existing files, the users could open a new project without closing and re-open the program.

From the Home Screen of the Program, we run menu File, then click on New Project, the New project with blank data would appear so users could use.

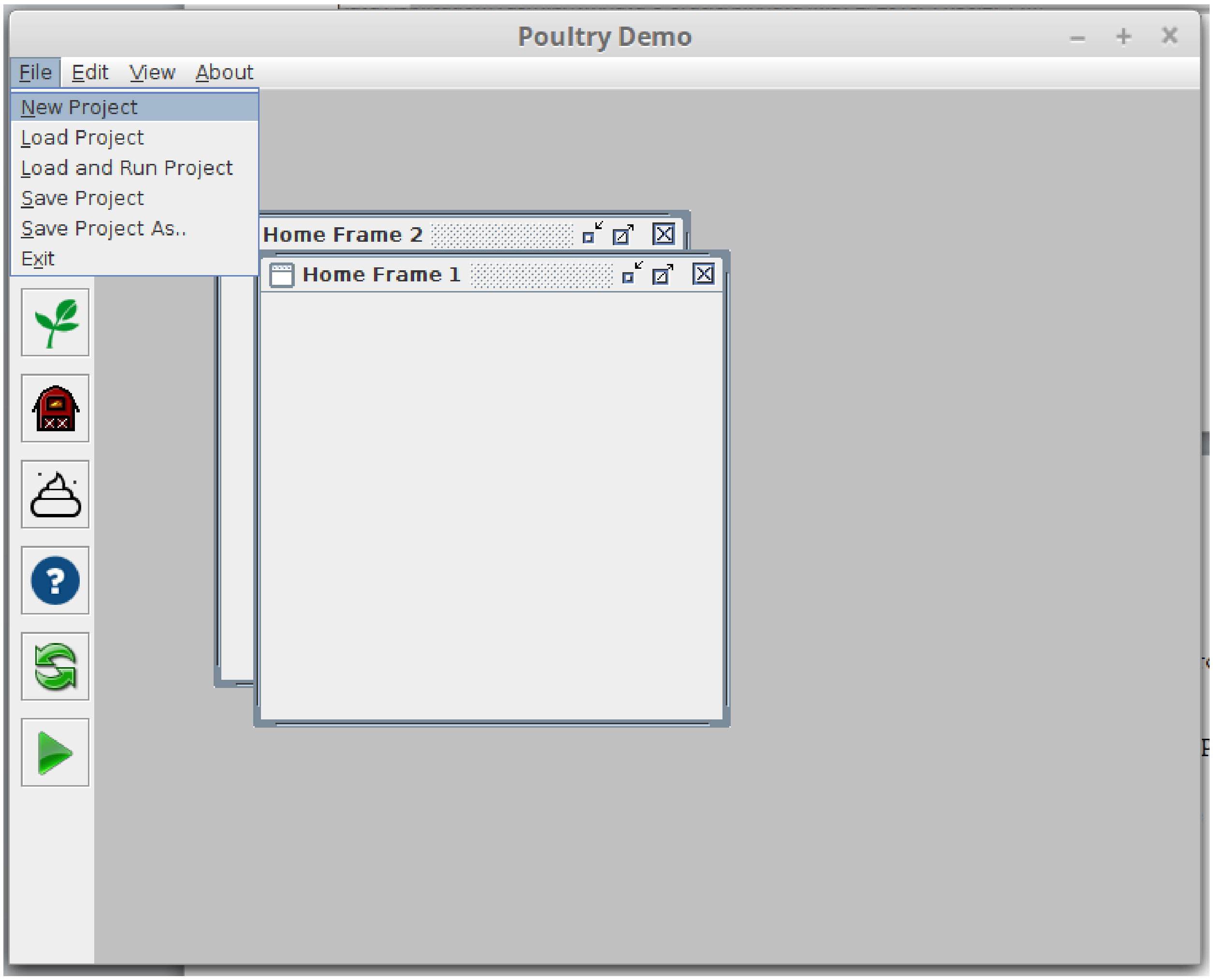

Figure 4. Create a new Project

4. Features and Screens

4.1 Home Screen

Should provide access to a welcome screen, directions to instructions, and help in creating a new project or loading an ongoing one.

4.2 Bird Screen

The first screen is the Bird screen. The primary purpose of this screen is to set the Bird size. Enter the breed, target Weight, number of broilers, number of fatalities. Once the herd size is entered (on the next screen) the model sets the herd size as the number of chicken spaces in the barn. The bird screen has one Drop-down Menu required for Breed Selection. The bird screen only contains one panel, which is Bird data. The choices for Breed are Cobb 500, Cobb 700, Ross 308, or Ross 708

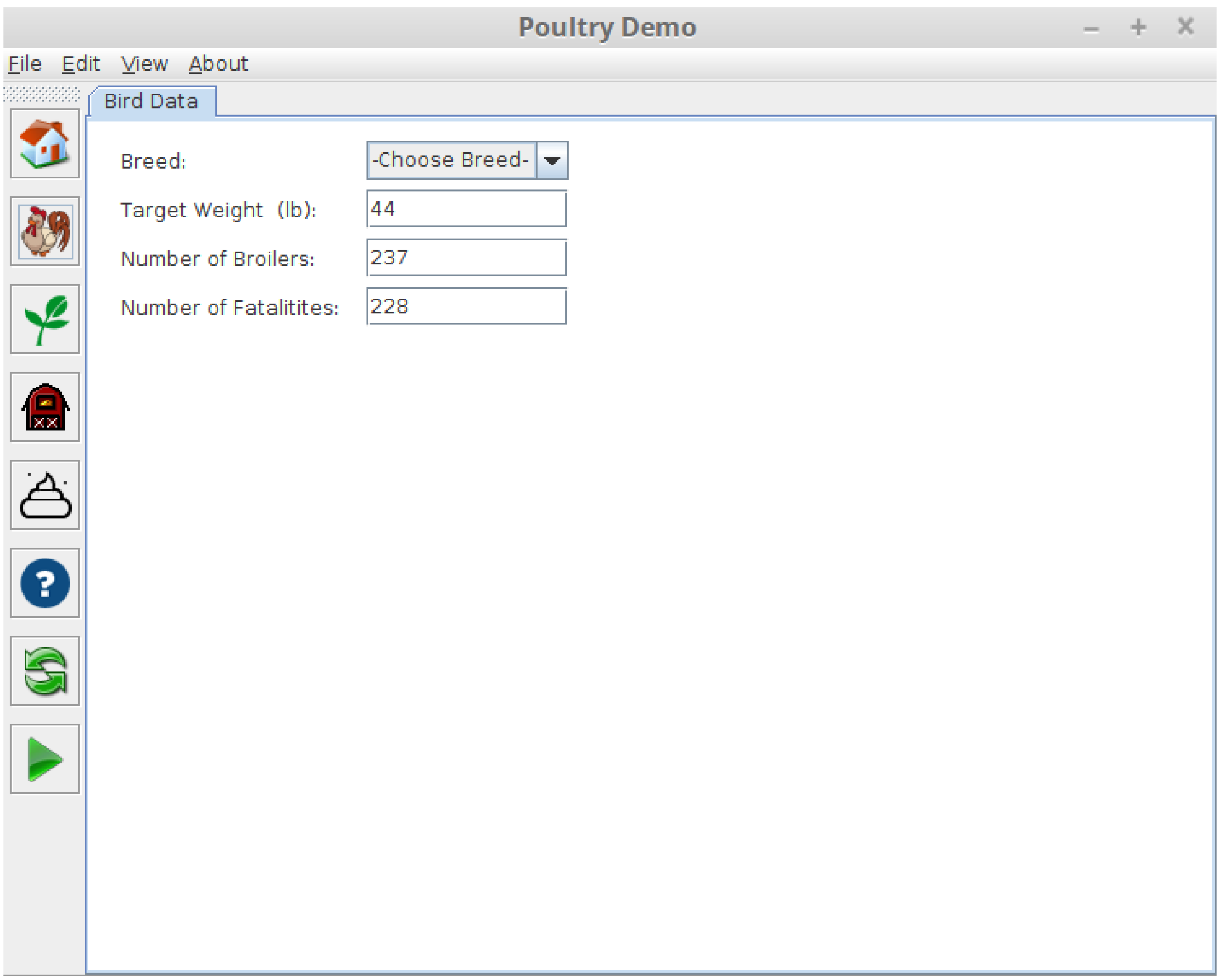

Figure 5. Bird Screen overview

The technical aspect of Bird Data screen would contains the following data type

		Variable Declarations
Breed:	**(Dropdown Menu)**	string birdBreed
	Cobb 500	
	Cobb 700	
	Ross 308	
	Ross 708	
Target Weight (lb):	(float)	float targetWeight
Number of Broilers:	(int)	int numBroilers
Number of Fatalities:	(int)	int numFatalities

Figure 6. Bird Screen variables data types

4.3 Feed Screen

The Feed Screen has two sub panels for input, which are ingredients and Feed Shipping Data. The discussion of each panel would be described in more detail in the sections below.

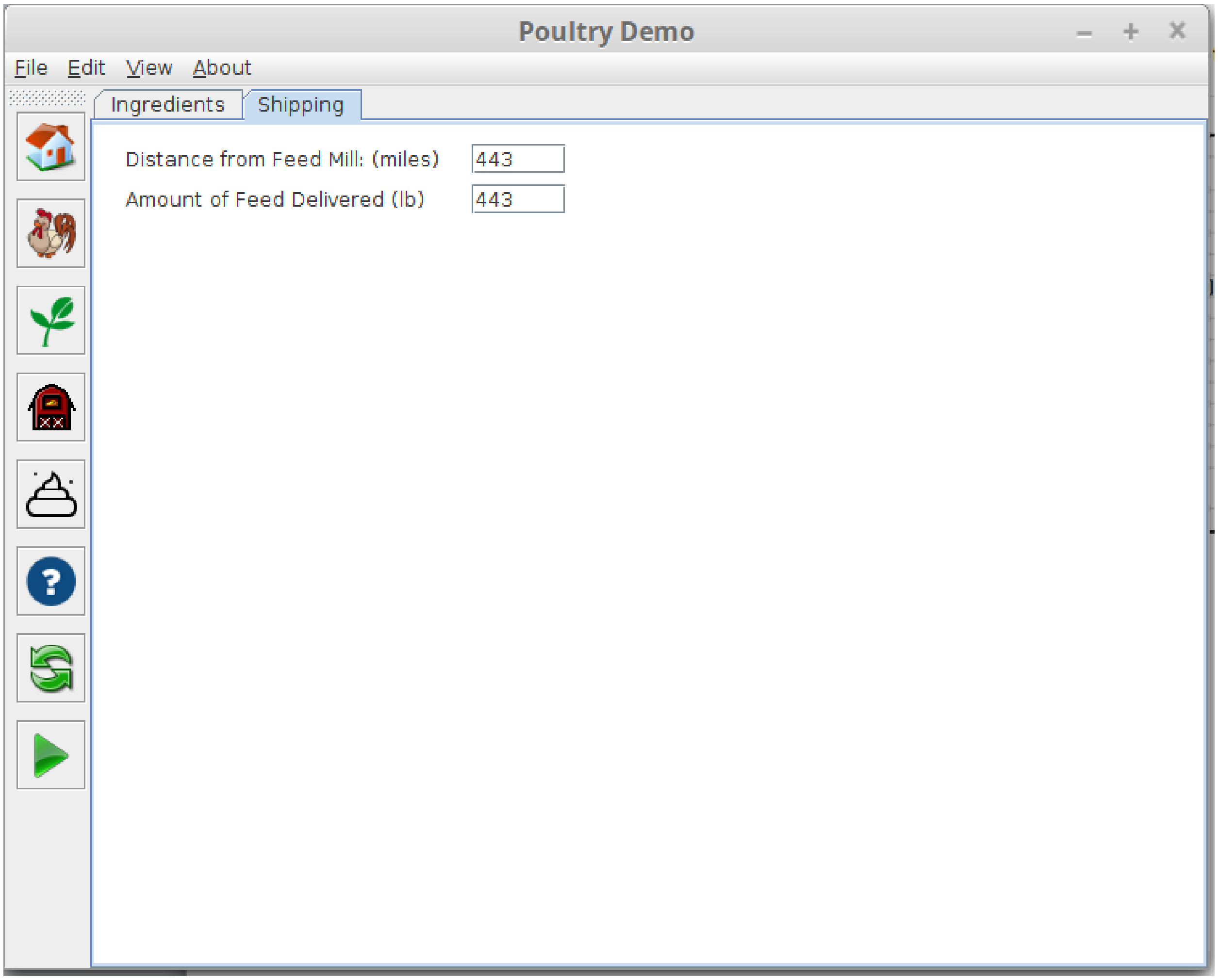

Figure 7. Two sub panels of Feed Screen Ingredients and Feed Shipping Data

4.3.1 Ingredients

In these barn types, the total number of ingredients used across both diets cannot exceed 30 ingredients. Feed ingredients are entered in terms of percent of diet on a dry matter basis in the "%" columns. The model will automatically convert each percent into pounds in the "lb./ton" columns. A user can enter diets different ways. The user can choose to input the example diets by choosing Load Example Inputs on the right side of the screen.

Choose the number of timing and phases as well as the ingredient mix for the barn on the Feed screens. There is a list of ingredients from which to choose from when building a feed mix. A list of these ingredients and their characteristics are listed below

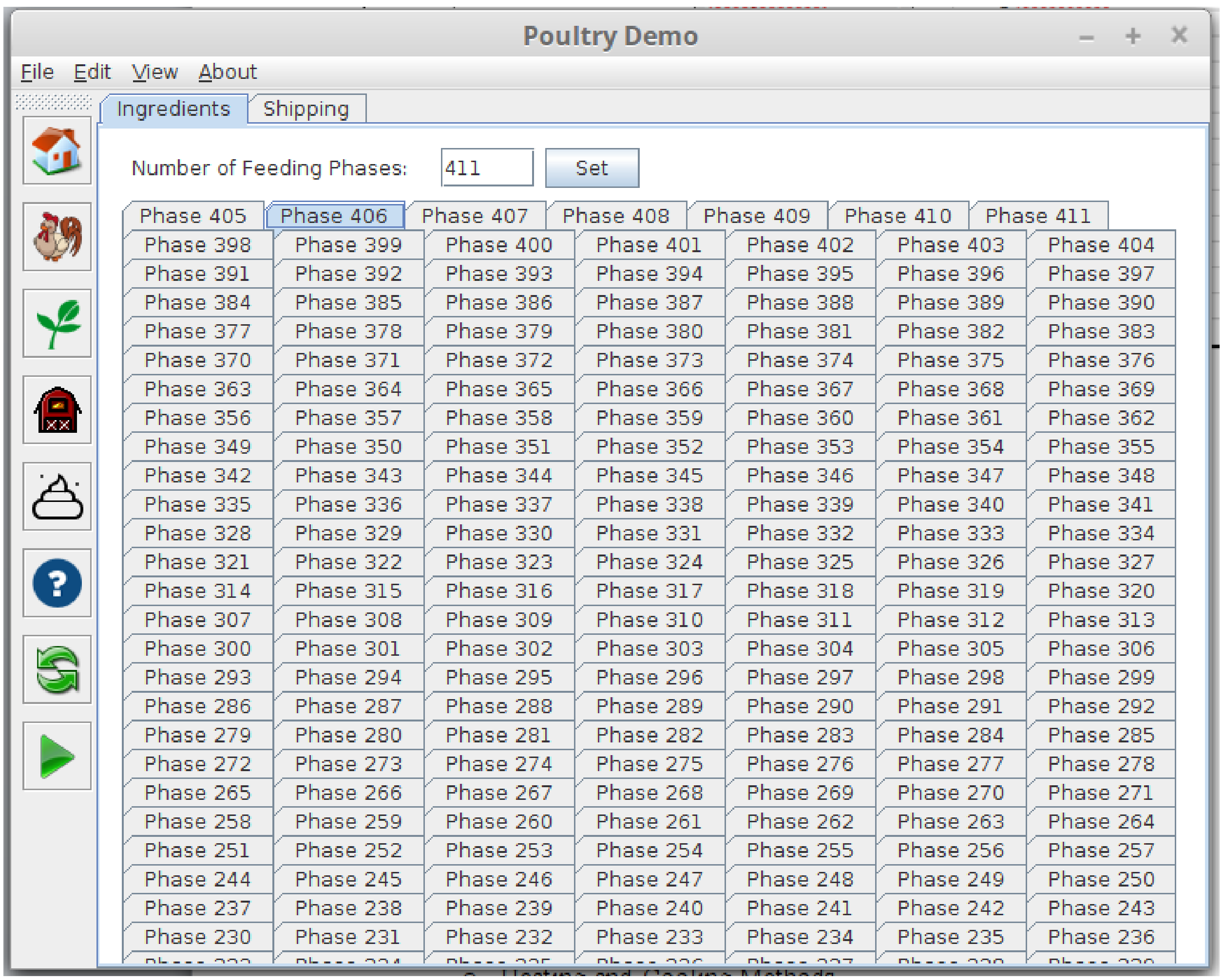

Figure 8. Feed Screen - Ingredients view

The technical aspect of Feed Screen - Ingredients would contains the following data type

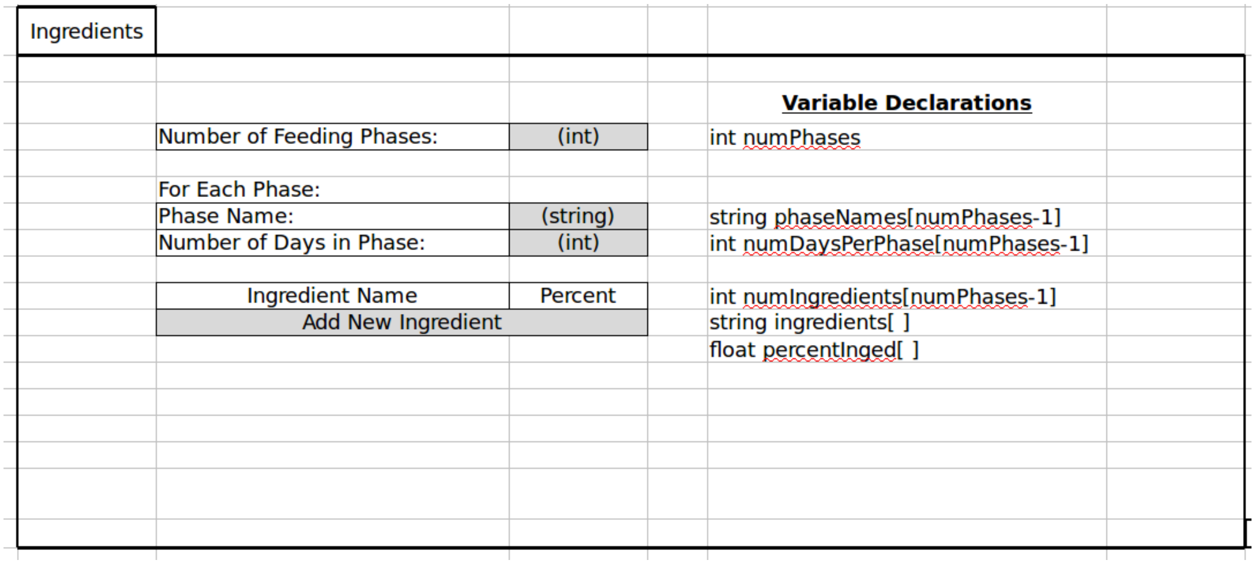

Figure 9. Feed Screen - Ingredients variables data types

The *Environmental Footprint Calculator For Poultry Producers* utilizes a database of ingredients information to make the calculation. The feed Ingredients for each phase would determine the amount of nutrients that the Birds have taken over the time, which would be used in the calculator

Table 1. List of Ingredients for Feed screen

Feed Ingedient Names	**Percent**
Alfalfa meal dehydrated, 17% protein	Specified by Users
Alfalfa meal dehydrated, 20% protein	
Bakery waste, dehydrated	
Barley *Hordeum vulgare* grain	
Barley grain, Pacific coast	
Broadbean, *Vicia faba* seeds	
Bloodmeal, vat dried	
Bloodmeal, spray dried	
Brewer's grains, dehydrated	
Buckwheat, common *Fagopyrum sagittatum* grain	
Canola *Brasica napus-Brasica campstris* [a]	
Casein dehydrated	
Casein precipitated dehydrated	
Cattle skim milk, dehydrated	

Coconut *Cocus nucifera* kernels[b]	
Corn distillers' grains, dehydrated	
Corn distillers' grains with solubles, dehydrated	
Corn distillers' solubles, dehydrated	
Corn gluten, meal, 60% protein	
Corn gluten with bran	
Corn grain	
Corn grits byproduct (hominy feed)	
Cotton seed meal, mechanically extracted, 41% protein	
Cotton seed meal, prepressed solvent extracted, 41% protein	
Cotton seed meal, prepressed solvent extracted, 44% protein	
Fish solubles, condensed	
Fish solubles, dehydrated	
Fish, Anchovey meal mechanically extracted	
Fish, Herring meal mechanically extracted	
Fish, Menhaden meal mechanically extracted	
Fish, White Gadidae meal mechanically extracted	
Gelatin by-products	
Livers meal	
Meat meal rendered	
Meat with bone, meal rendered	
Millet Pearl *Pennisetum glaucum* grain	
Millet, Proso *Panicum miliaceam* grain	
Oats *Avena sativa* grain	
Oats grain, Pacific coast	
Oats hulls	
Pea *Pisum* spp. seeds	
Peanut meal, mechanically extracted	
Peanut meal, solvent extracted	
Poultry by-product meal (visera with feet and heads)	
Poultry feathers, meal hydrolysed	
Ricebran with germ	
Rice grain, polished and broken	
Rice polishings	
Rye *Secale cereale* grain	
Safflower seeds, meal solvent extracted	
Safflower seeds witout hulls, meal solvent extracted	
Sesame seeds, meal mechanically extracted	
Sorghum grain, 8-10% protein	

Sorghum grain, > 10% protein	
Soybean flour by-product	
Soybean protein concentrate, more than 70% protein	
Soybean seeds, heat processed	
Soybean seeds, meal solvent extracted	
Soybean seeds without hulls, meal solvent extracted	
Sunflower common seeds, meal solvent extracted	
Sunflower seeds without hulls, meal solvent extracted	
Triticale *Triticale hexaploide* grain	
Wheat *Triticum aestivium* bran	
Wheat red dog	
Wheat middlings	
Wheat shorts	
Wheat grain, hard red winter	
Wheat grain, soft white winter	
Whey *Bos taurus*, dehydrated	
Whey *Bos taurus*, low lactose dehydrated	
Yeast, Brewer's dehydrated	
Yeast, Torula, dehydrated	

4.3.2 Shipping

The screen view for the Feed Screen – Shipping data is shown below

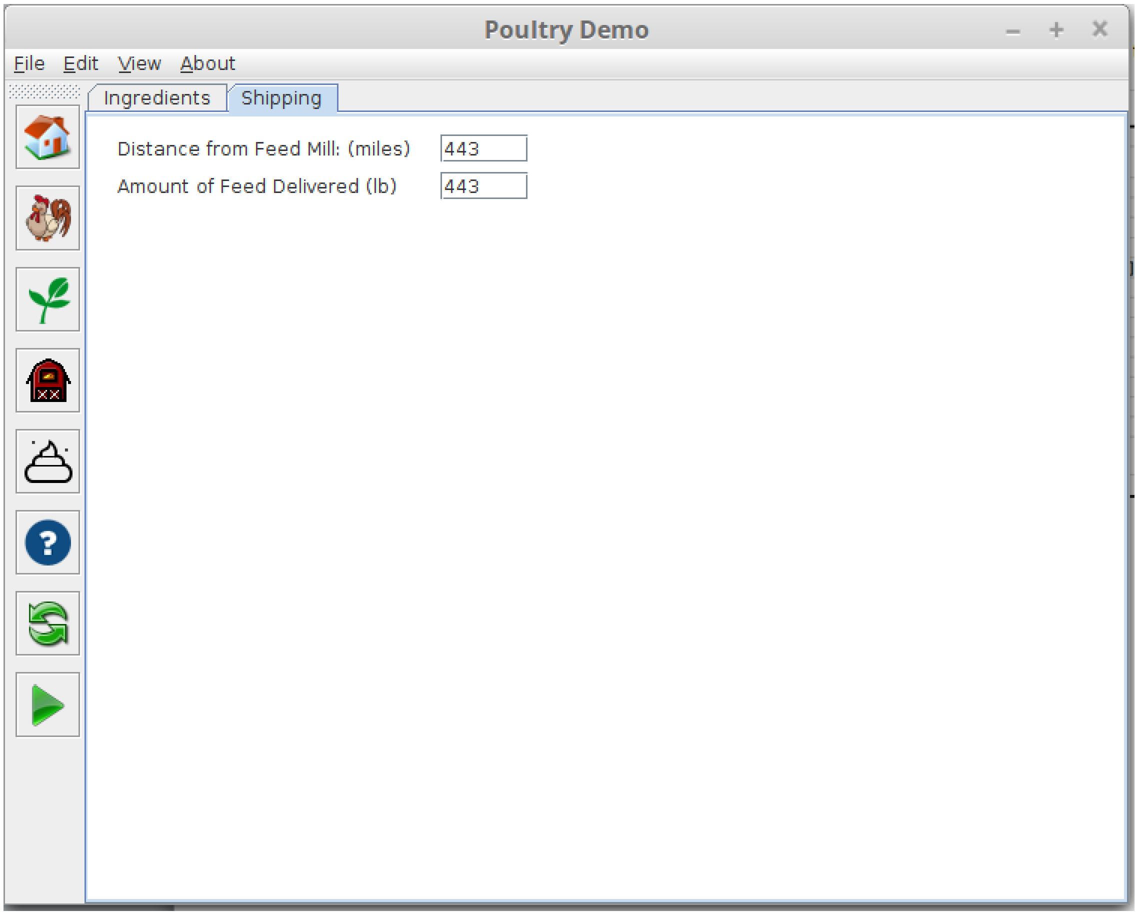

Figure 10. Feed Screen - Shipping view

The technical aspect of Feed Screen - Shipping would contains the following data type

Shipping		
		Variable Declarations
Distance from Feed Mill (miles):	(float)	float feedDistance
Amount of Feed Delivered (lb)	(float)	float feedMassDel

Figure 11. Feed Screen - Shipping variables data types

4.4 Barn Screen

The Barn Screen has four sub panels for input, which are Location and Size, Heating and Cooling Methods, Water Usage, and Lighting Usage. The discussion of each panel would be described in more detail in the sections below.

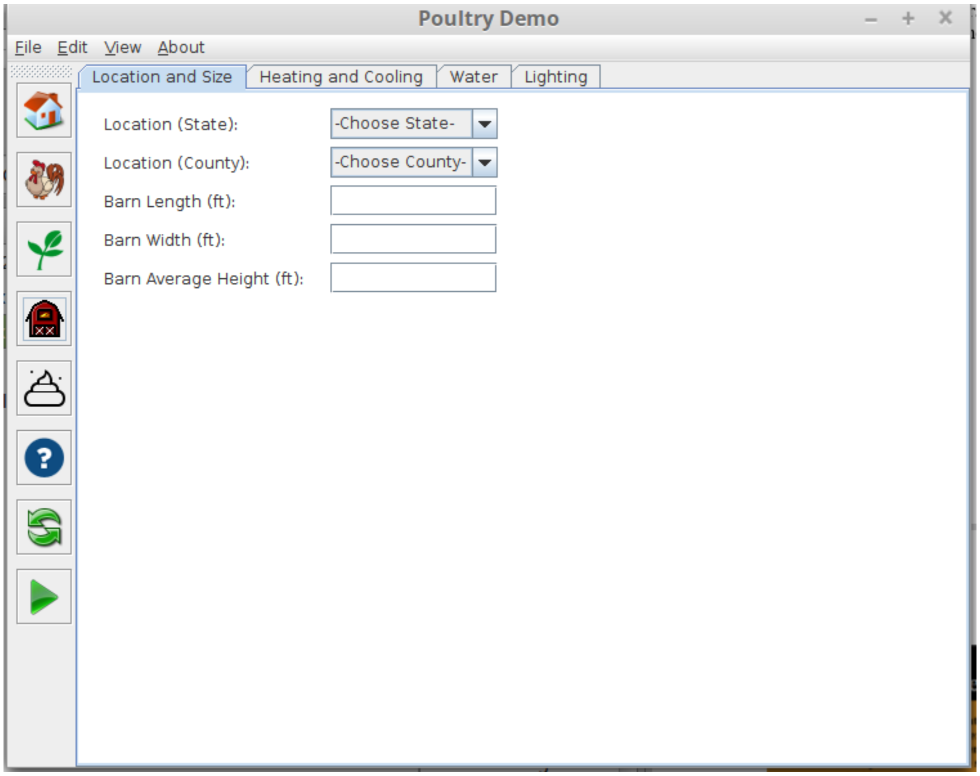

Figure 12. Four sub panels of Barn Screen

4.4.1 Location and Size

The farm’s location is important because the model uses it to determine the relevant local weather throughout the year and to find suggested local prices for feed ingredients, utilities 26(electricity, water), commodities (fuels), and farm operations (dead animal disposal, IC injections, manure management).

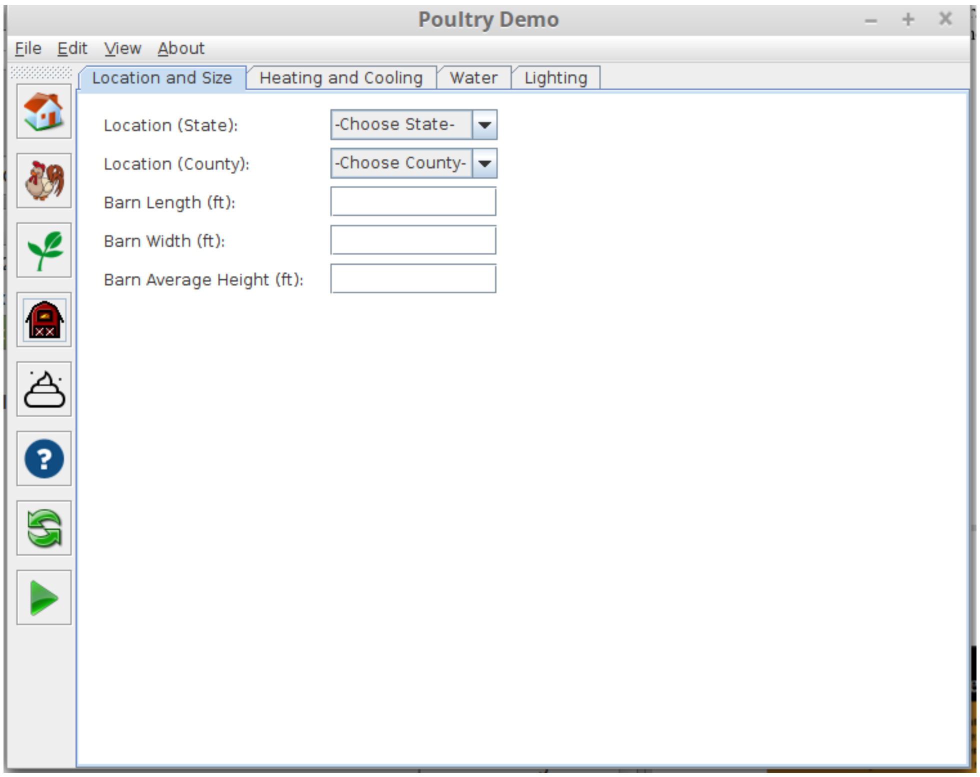

Figure 13. Barn Screen - Location and Size panel view

The technical aspect of Barn Screen - Location and Size would contains the following data type

Location and Size		
		Variable Declarations
Location (State):	(Dropdown)	string state
Location (County):	(Dropdown)	string county
Barn Length (ft):	(float)	float barnLength
Barn Width (ft):	(float)	float barnWidth
Average Barn Height (ft):	(float)	float barnHeight

Figure 14. Barn Screen - Location and Size variables data types

4.4.2 Heating and Cooling

The temperature in the barn is one of the most important parameters to the growth performance of a chickens. The NRC equations that are used in this model reflect this dependency by modulating the amount of feed consumed and the resulting growth rates as a function of barn temperature. The model has a detailed heat balance around the barn, calculated once an hour throughout the year, with the purpose of estimating realistic barn temperatures.

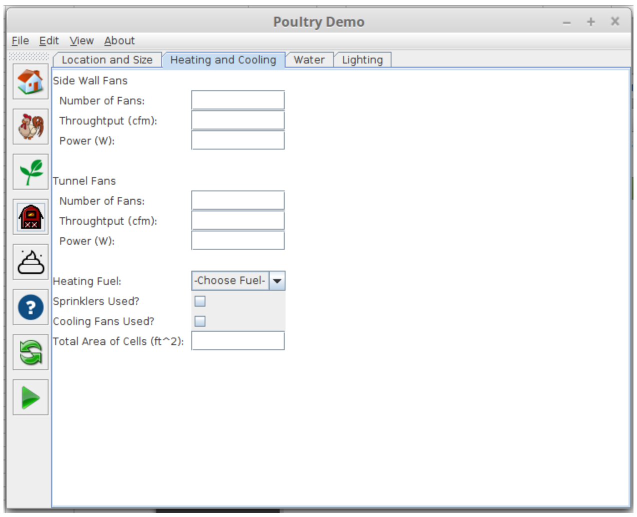

Figure 15. Barn Screen - Heating and Cooling screen

The technical aspect of Barn Screen – Heating and Cooling would contain the following data types:

	Heating and Cooling	Variable Declarations
Side Wall Fans:		
Number of Fans:	(int)	int numSideWall
Throughput (cfm):	(float)	float sideThroughput
Power (W):	(float)	float sidePower
Tunnel Fans:		
Number of Fans:	(int)	int numTunnel
Throughput (cfm):	(float)	float tunnelThroughput
Power (W):	(float)	float tunnelPower
Heating Fuel:	(Dropdown)	bool propaneUsed
	Propane	
	Natural Gas	
Cooling Fans Used?:	(Check Box)	bool fansUsed
Total area of cells (ft2):	(float)	float fanArea
Sprinklers Used?:	(CheckBox)	bool sprinklersUsed

Figure 16. Barn Screen - Heating and Cooling variables data types

4.4.3 Water

This screen shows the percentage of water used on the farm that is coming from well, piped in and surface water sources. The total for all sources must equal 100%.

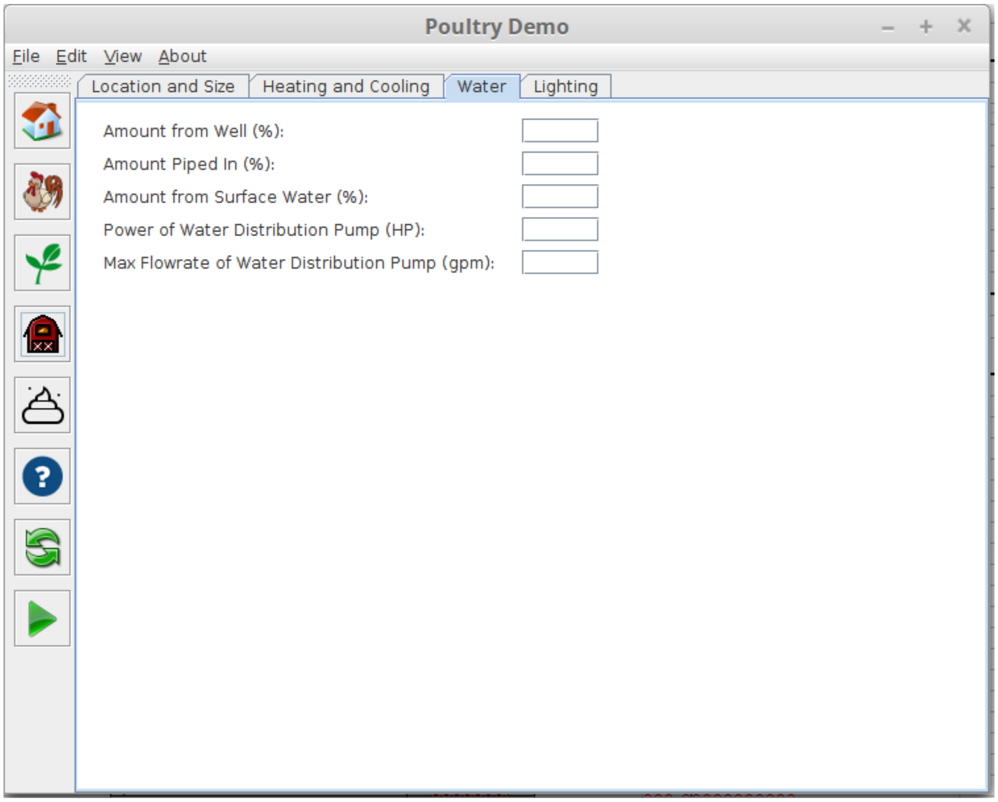

Figure 17. Barn Screen - Water screen

The technical aspect of Barn Screen – Water would contain the following data types:

		Water
		Variable Declarations
Water Sources:		
Amount from Well (%):	(float)	float waterWell
Amount Piped In (%):	(float)	float waterPiped
Amount from Surface Water (%):	(float)	float waterSurface
Power of Water Distribution Pump (HP):	(float)	float pumpPower
Max Flowrate of Water Distribution Pump (gpm):	(float)	float pumpFlowrate

Figure 18. Barn Screen - Water screen variables data types

4.4.4 Lighting

The model asks for total wattages instead of the number of bulbs of each wattage so a barn could have an arbitrary mix. All power to the lights is counted as heat by the heating and cooling code.

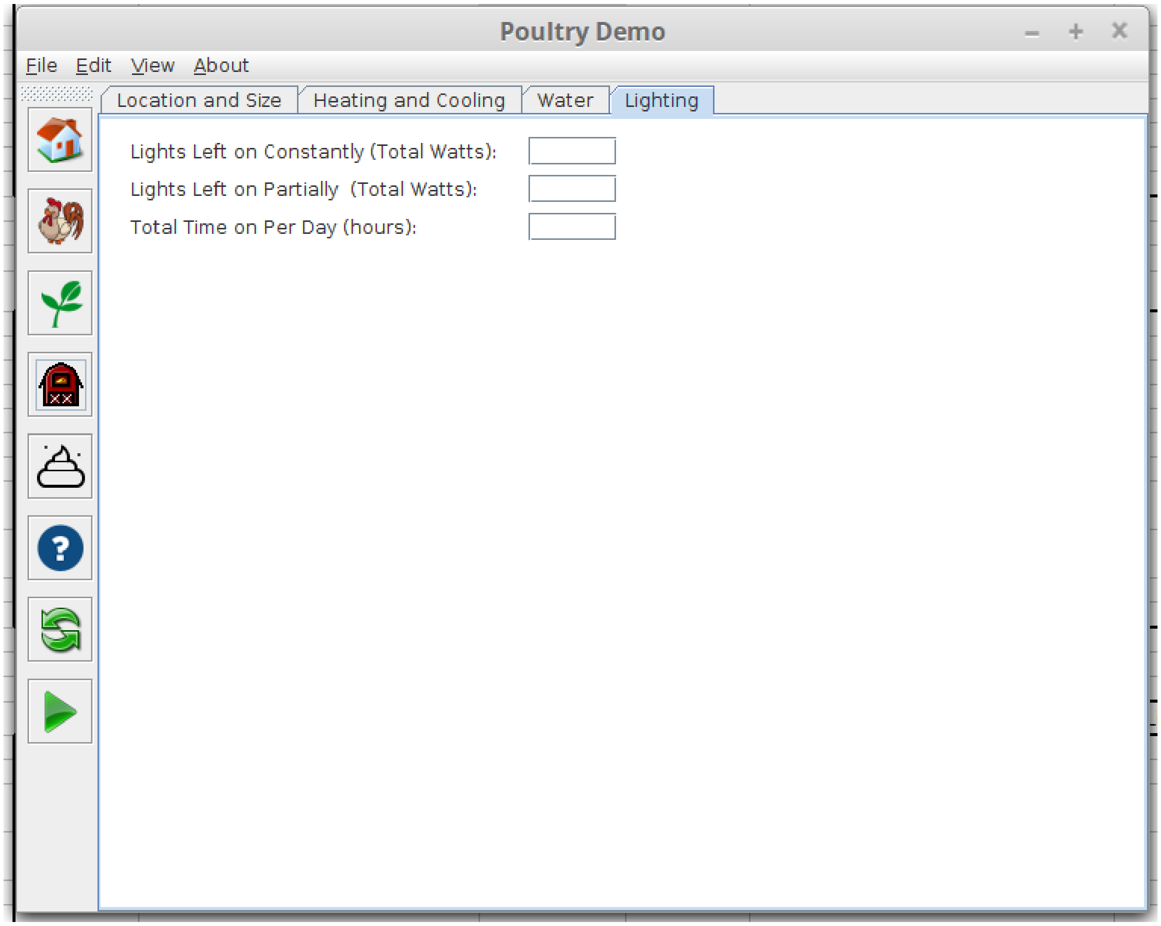

Figure 19. Barn Screen - Lighting screen view

The technical aspect of Barn Screen – Lighting would contain the following data types:

					Lighting
				Variable Declarations	
	Lights Left on Constantly (Total Watts):	(float)		float lightsConstant	
	Lights Left on for Part of the Day (Total, Watts):	(float)		float lightsIntermittent	
	Total Time on Per Day (hours):	(float)		float lightsTimeOn	

Figure 20. Barn Screen - Lighting screen variables data types

4.5 Waste Screen

The screen view for the Waste screen data is shown below

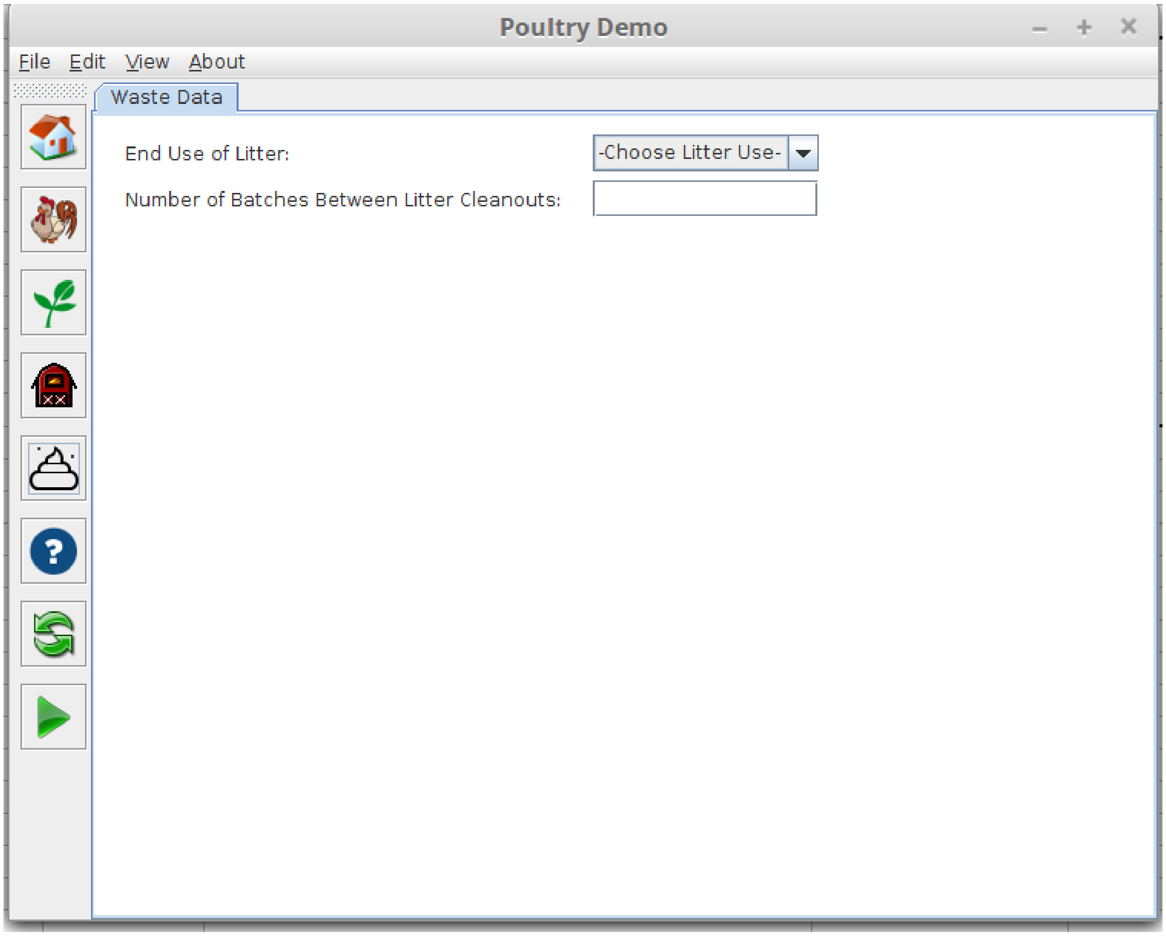

Figure 21. Waste Screen view

The technical aspect of Waste Screen would contain the following data types:

		Variable Declarations
End Use of Litter:	**(Dropdown Menu)**	string litterUse
	Fertilizer	
	Animal Feed	
	Fuel Source	
Number of Batches Between Litter Cleanou	(int)	int numLitterPhases

Figure 22. Waste Screen variables data types

9. Deign and Technical Documentation

The *Environmental Footprint Calculator For Poultry Producers* has undergone rigorous testing; however, as in any big software project, problems can arise. As such, we would attach the design architecture of the software as well as technical design that was used. This way, the developers of the next generation could go through and make improvement to the program.

9.1 High Level Design

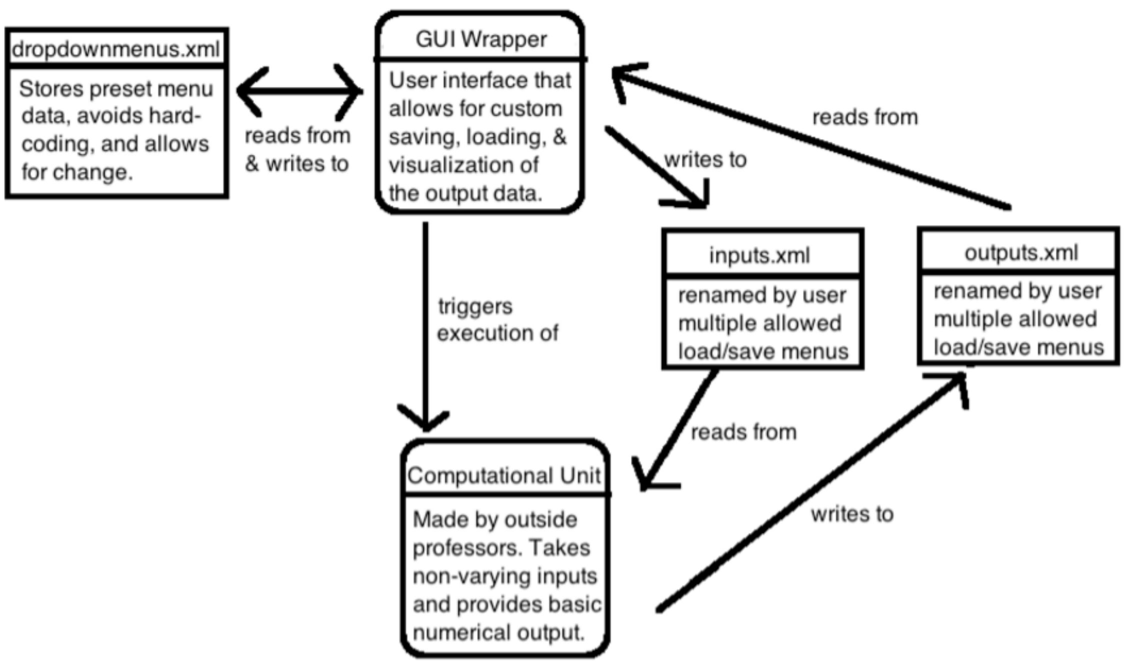

Figure 23. Essential function and flow of our GUI

The GUI Wrapper is the only coding responsibility for our team, as the computational engine is handled by the sponsors, but is is in our hands to handle the computational engine. That means that every requirement for the engine to run must be met and that responsibility falls on us, as our GUI is responsible for gathering data, error checking data, and converting drop-down selections into the multiple value formats that the engine requires. All saving and interfacing can be seen to work through XML, which is an industry standard in passing data between programs and processes. The only direct interaction between the programs is just the order for the computational engine to begin.

When new hands pick up this project there is the issue of waiting the right amount of time after triggering the computational engine to look for output. Given that there is no computational engine yet, that hadn't been discussed with the sponsors about how to pair our programs in that way. One possible solution is for the computational engine to add the date and time to the XML it writes. Then, the GUI can wait an appropriate number of milliseconds, grab the XML and check that the date and time written is past the date and time that the GUI triggered the calculation engine. Again, it is a future issue.

As drop-down menus are convenient for linking a label to data, which sometimes has many components (such as nutritional values of a feed), there must be a way to edit values and add items, and so drop-down menu information is also contained in an XML file.

While our GUI allows for the user to freely choose any category and fill out the inputs in any order, our GUI was created with a flow in mind. The ordering of the input categories in the GUI imply a process following the order shown above for ease of use. The "Home" tab will eventually introduce a "New User" wizard that will guide users through the process of using the GUI.

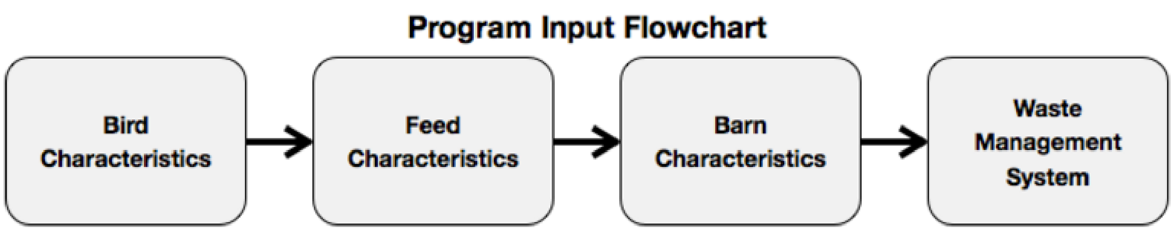

Figure 24. Program Input Flowchart

The following image is a sampling of our input variables for the GUI (as provided by our champion, Heather). These input variables have been included in the GUI as various forms of Java input methods, including text boxes, check boxes, and drop down menus.

Input Variables	
Bird Characteristics	Breed
	Number of Birds
	Number of Bird Fatalities
	Finished Weight
Feed Characteristics	Number of Feeding Phases
	Feed Ingredients
	Distance from Feed Mill
	Frequency of Feed Deliveries
Barn Characteristics	Location (State)
	Location (County)
	Barn Type
	Percentage of Water from Well
	Percentage of Water piped in
	Percentage of Water from Surface
	Power of Water Distribution Pump
	Max Flowrate of Distribution Pump
	Heating/Cooling Method
	Dead Animal Disposal Method
Waste Management System	Litter Processing Method
	Frequency of Litter Processing
	Litter Disposal Method

Figure 25. Program Input variables

9.2 System Architecture

PoultryDemo.java creates the initial instances of the JFrames and JPanels, and within these creation methods further methods are called from loadInputMethods.java to populate them. This is the Main file for compiling and running.

IOClass.java is a custom class that reads/saves to xml file format to/from the variables of the UI. It is used for iteracting with the calculator via these XML files.

Calculator_Call.java is a placeholder that is called when the 'run' button is pressed. It finds which operating system is being used (Windowns, mac, linux/unix) and then executes a temporary C++ executable (which will eventually be the calculator executable).

The table of system architecture is shown on the next page.

Table 2. Table of system architecture

File Name	Description	Methods() : return type
PoultryDemo.java	Main class Component creation	createToolBar : JToolBar createMenuBar : JMenuBar createAboutMenu : void createViewMenu: void createEditMenu : void createFileMenu : void createWastePanel : JTabbedPane createBarnPanel : JTabbedPane createBarnPanel : JTabbedPane createFeedPanel : JTabbedPane createBirdPanel : JTabbedPane createHomePane : JDesktopPane createHomeFrame : void createDesktopPane : JDesktopPane
Calculator_Call.java	Method for running an external process (.exe)	main : void output : String isWindows: bool isMac : bool isUnix : bool
loadInputMethods.java	Population methods for tabs in tabbed panes	loadBirdDataPanel : void loadFeedIngredPanel : void loadFeedShippingPanel : void loadBarnLocSaizePanel : void loadBarnHeatCoolPanel : void loadBarnWater : void loadBarnLighting : void loadWastePanel : void
IOclass.java	References all input components for saving/loading from file	addinput : void loadinputs : void saveinputs : void
KeyEvent.java	Utility class for keyboard input	getKeyCodeForChar(char) : int getTypeString(int) : String
/resources/*.png	Icon files for buttons 32x32 png	
/resources/cpp*.exe	Dummy C++ executables	

9.3 Swing Component Layout

The GUI has three main components contained in the encompassing frame window topFrame. They are the main desktop pane desktopPane, the menu bar topMenuBar, and the tool bar topToolBar.

Panels displayed on desktopPane are switched using the buttons in topToolBar. topMenuBar contains shortcuts to file saving, loading, as well as future features not yet implemented.

The current build has the following visual structure of Java Swing componets (javax.swing.*). User input components (text fields, checkboxes, etc.), are sent to the IOclass for use in file input/output. The IOclass keeps arraylists of inputs based on their type, i.e. ArrayList<JTextField>. IOclass uses these to load/save from/to xml files.

The table of Visual Layouts Components is shown on the next pages.

Table 3. Table of Visual Layouts Components

Visual Layout Components			
Component Name	**Component Type**	**Contained In**	**Creation Method Layout Type**
PoultryDemo()			
topFrame	JFrame	PoultryDemo class	PoultryDemo construct BorderLayout
desktopPane	JDesktopPane	topFrame	createDesktopPane() CardLayout
topMenuBar	JMenuBar		createMenuBar()
topToolBar	JToolBar		createToolBar()
createHomePane()			
homePane	JDesktopPane	desktopPane	createHomePane()
createBirdPanel()			
birdPanel	JTabbedPane	desktopPane	createBirdPanel() FlowLayout
birdDataPanel	JPanel	birdPanel	loadBirdDataPanel() GroupLayout
birdBreedLabel breedInputBox	JLabel JComboBox<String>	birdDataPanel	
targetWeightLabel targetWeightField	JLabel JTextField		
numBroilersLabel			

numBroilersField numFatalitiesLabel numFatalititesField			
Component Name	**Component Type**	**Contained In**	**Creation Method Layout Type**
createFeedPanel()			
feedPanel	JTabbedPan e	desktopPane	createFeedPanel() FlowLayout
feedIngredientsPanel	JPanel	feedPanel	loadIngredPanel() GroupLayout
numPhasesLabel numPhasesField	JLabel JTextField	feedIngredientsPane l	
phaseNameLabel phaseNameField			
numDaysPerPhaseLabe l numDaysPerPhaseField			
feedShippingPanel	JPanel	feedPanel	loadFeedShippingPanel() GroupLayout
feedDistanceLabel feedDistanceField	JLabel JTextField	feedShippingPanel	
feedMassDelLabel feedMassDelField			
createBarnPanel()			
barnPanel	JTabbedPan e	desktopPane	createBarnPanel()
BarnLocationSize	JPanel	barnPanel	loadBarnLocSizePanel() GroupLayout

<table>
<tr><td>barnLocationState
barnStateInputBox</td><td>JLabel
JComboBox
<String></td><td rowspan="5">BarnLocationSize</td><td></td></tr>
<tr><td>barnLocationCounty
barnCountyInputBox</td><td>JLabel
JComboBox
<String></td><td></td></tr>
<tr><td>barnLength
barnLengthField</td><td rowspan="3">JLabel
JTextField</td><td></td></tr>
<tr><td>barnWidth
barnWidthField</td><td></td></tr>
<tr><td>barnHeight
barnHeightField</td><td></td></tr>
<tr><td>BarnHeatCool</td><td>JPanel</td><td>barnPanel</td><td>loadBarnHeatCoolPanel(
)
FlowLayout</td></tr>
<tr><td>SideFanAmt
SideFanAmtField</td><td rowspan="6">JLabel
JTextField</td><td rowspan="6">BarnHeatCool</td><td rowspan="6"></td></tr>
<tr><td>SideFanThroughput
SideFanThroughField</td></tr>
<tr><td>SideFanPower
SideFanPowerField</td></tr>
<tr><td>TunnelFanAmt
TunnelFanAmtField</td></tr>
<tr><td>TunnelFanThroughput
...ThroughputField</td></tr>
<tr><td>TunnelFanPower
TunnelFanPowerField</td></tr>
<tr><td>HeatingFuel
HeatingFuelDrop</td><td>JLabel
JComboBox
<String></td><td></td><td></td></tr>
<tr><td>CoolFanUsed
CoolFanCheck</td><td>JLabel
JCheckBox</td><td></td><td></td></tr>
</table>

CellTotalArea CellAreaField	JLabel JTextField		
SprinklersUsed SprinklerCheck	JLabel JCheckBox		
BarnWater	JPanel	barnPanel	loadBarnWater() GroupLayout
WellAmount WellAmountField	JLabel JTextField	BarnWater	
PipedAmount PipedAmountField			
SurfaceWaterAmount SurfaceWaterField			
WaterPumpPower WaterPumpField			
MaxFlowrate FlowrateField			
BarnLighting	JPanel	barnPanel	loadBarnLighting() GroupLayout
ConstantLight ConstantLightField	JLabel JTextField	BarnLighting	
PartialLight PartialLightField			
TotalTime TotalTimeField			
Component Name	**Component Type**	**Contained In**	**Creation Method Layout Type**
createWastePanel()			
wastePanel	JTabbedPan	desktopPane	createWastePanel() GroupLayout

	e		
wasteData	JPanel	wastePanel	loadWastePanel() GroupLayout
LitterUse LitterUseDrop	JLabel JComboBox <String>	wasteData	
LitterCleanout LitterCleanoutField	JLabel JTextField		
createMenuBar()			
mainMenuFile	JMenu	topMenuBar	createFileMenu()
mainMenuEdit			createEditMenu()
mainMenuView			createViewMenu()
mainMenuAbout			createAboutMenu()
createToolBar()			
home button home.png	JButton	topToolBar	
bird button bird.png			
feed button feed.png			
barn button barn.png			
waste button waste.png			
help button help.png			
run button run.png			

End of User's Manual

University of Arkansas

Office of Sustainability

Department of Computer Science and Computer Engineering

Environmental Footprint Calculator For Poultry Producers