Research by:

572481 - Adu, Stephen

570027- Andreicha, Semida

563453 - Buaron, Tal

573967 - Cholodov, Andrej

566536 - Domela Nieuwenhuis, Jens

With the assistance of Harald Drillenburg and Koos van Tubergen

09-01-2017

research report – feedback version 12

PROJECT APPLICATION DEVELOPMENT - InHolland University of Applied Science

Abstract

This research report explores the development of an application that results from a possible solution, to a problem taken from an existing ecological system, by using a mathematical model to show the predicted effects of the proposed measure on the ecosystem.

Table of Contents

[1. Introduction 2](#_Toc472273987)

[1.1 Context 2](#_Toc472273988)

[1.2 Purpose 2](#_Toc472273989)

[2. Central research question and sub-questions 2](#_Toc472273990)

[2.1 Research question 2](#_Toc472273991)

[2.2 Research sub-questions 3](#_Toc472273992)

[3. Research Methodology 3](#_Toc472273993)

[3.1 Data Collection 3](#_Toc472273994)

[3.2 Mathematical Model 4](#_Toc472273995)

[3.3 Software Development 6](#_Toc472273996)

[3.4 Connection 10](#_Toc472274001)

[4. Results 11](#_Toc472274002)

[4.1 Results Data Collection 11](#_Toc472274003)

[4.2 Results Mathematical Model 13](#_Toc472274004)

[4.3 Result Software Development 14](#_Toc472274005)

[4.4 Result Connection between Application 17](#_Toc472274006)

[5. Conclusion 17](#_Toc472274007)

[6. Discussions 18](#_Toc472274008)

[7. Bibliographies 20](#_Toc472274009)

[8. Appendices 21](#_Toc472274010)

[8.1 Competition Coefficient Calculations 21](#_Toc472274011)

[8.2 UML Diagrams 23](#_Toc472274012)

# Introduction

Mathematical engineering is based on developing application models for making predictions and find desirable solutions to real-life problematic situations. While working towards the development of an application, one comes across several challenges which changes the approach of the development. Challenges can be encountered when determining the model of the application, due to the observation and collection of data which is relevant to use as variables in a mathematical formula. Thus, parameters are going to be divided into categories, such as the essential and trivial factors, then later implemented to improve the model. Therefore, due to the level of complexity, that the model may hold, it is mandatory to not only implement an application which gives an output statistical numbers but visuals as well, such as graphs.

## 1.1 Context

Oostvaardersplassen, a preserved ecological system, which has many animals living there including the main three large herbivores: wild horses, wild cattle, and deer; and other foremost species such as geese and birds of prey. Assuming that no major predators are present, in the ecological system, and the herbivores being prohibited to migrate. The consequence of an unbalance ecosystem has been established in the Oostvaardersplassen preservation.

Complicating the situation is the fact that there are a lot of geese, particularly during winter, and they consume the same type of food as the large herbivores do. The competition and interaction between the herbivores and the geese for food can be quite severe, leading some to extreme situations such as extinction.

## 1.2 Purpose

The intention of this paper is to work towards a possible solution of a problem, which has been taken from an existing ecological system, Oostvaardersplassen. The solution will be implemented based on a mathematical model on a system, resulting in application which will be the one doing the number calculations. The outcome of the application will be graphical illustrated, and will hold the predicted effects of the proposed measures on the ecosystem.

The language suggests someone speaking English well, but not a native speaker, and trying to make it too polished. Don’t; you’re making too many mistakes that way, making the text hard to read. Use your own level of English, it’s more than good enough.

# Central research question and sub-questions

## 2.1 Research question

The main research question for the programme is going to be divided into two parts. For the reason that, two applications are needed in order for the answer to be found.

Main Research Question of the project at the programme level is:

What will happen to the populations of deer, cattle, horse and geese if releasing a number of foxes in the preserve (with the number of foxes being the free variable)?

The Main Research Question for this research paper, project level, derives from the programme research question, based on the competition between the herbivores:

How do the herbivores and the geese compete for the grass in an enclosed area?

## 2.2 Research sub-questions

The research sub-questions are questions which derive from the main research question, and provide the path to the answer of the research paper.

Sub-Research Questions for this research paper:

* + 1. What is the rate of grass consumption of each animal?
    2. How do animal populations change as a result of grass availability?
    3. What are the populations of each animal?
    4. How can the start data and historical data help shape existing mathematical models to an equation which fits the problem in Oostvaardersplassen?
    5. How can the application hold all the implementations that are being asked?

# Research Methodology

In order to answer the main research question for this programme, this research paper will start by focusing on answering the research sub-questions of this project. Then will attempt to answer the research question of this paper, which subsequently will lead to an answer to the programme research question. The methodology explored in this project reflects the competition for food, taking place between herbivores and geese, in an enclosed ecosystem, Oostvaardersplassen. By means of collecting data and modelling equation based on the data, the research will then attempt to provide an application which has as an outcome accurate prediction based on fluctuating data.

The process of development has several phases that are being implemented to get a final result. One of the phases, which kicks off this research, is based on **data collection**. This is the point at which all the information necessary to proceed with any of the further steps, is going to be gathered. The second phase of the research then fully focus on the development of the **mathematical models**, which are the key aspects that bring out the outcome wanted for this research paper. In the third phase multiple developments are in view, those are the **development of the software**, which includes both the frontend work as well as the backend work. The last phase of the development incorporates the **connection**, which will be established, between the two software applications. Subsequently, each one of these steps, will help answer the research sub-questions which in turn will provide a solution to this paper’s research question.

## Data Collection

The first step into development begins with **data collection**, which is based on the information found on the Website of Oostvaardersplassen (Startpagina Staatsbosbeheer. Het zit in onze natuur. n.d.). There information regarding the *population statistics* of the three main herbivores and geese, during different seasons, *birth rate and death rate,* the *amount of grass they eat*, and any *other related data*.

The way in which existing *population numbers* of herbivores and geese, from Oostvaardersplassen, are going to be deduced is by using given resources, that were received, and information, that is on the Oostvaardersplassen website (Startpagina Staatsbosbeheer. Het zit in onze natuur. n.d.).

As well an excel sheet was provided, with the population numbers and *total births rates and deaths rates* for each of the three-herbivore species. From there, calculations based on the individual birth/death-rate for each species were involved. Using SPSS’ [[1]](#footnote-1)Curve Fit functionality, the correlation between the population and individual birth/death-rate, gave some of the variables needed to be used in the mathematical model. A full list of these variables can be found in the appendix.

However due to lack information, about the *grass consumption* of each animal, more research had to be conducted. The way about to the amount of grass consumption is to find the amount that each of the animals, in Oostvaardersplassen, needs to continue living. There are a lot of resources with information in regards to the grass consumption of the herbivores. Unlike the herbivores, the geese were lacking resource information and sources. Thus, this research deals with only the most similar specie that resembles the herbivores found in Oostvaardersplassen. Then data will be taken from those species and followed with an implementation in the application.

## 3.2 Mathematical Model

Then the process of development shifted towards the second step which was to develop a **mathematical model**. The *equation*, used in the mathematical model, is a *modified version*, shown in Equation 1.1, of the competitive Lotka-Volterra model (Gotelli, Nicholas J. 2008) (Equation 1.2). This model calculates the *population growth rate* of a specific species, given the specific species’ *grass availability*, *intrinsic rate of increase*, *carrying capacity*, competing species’ population and their competition coefficients.

*Modified version* of the equation of competitive Lotka-Volterra model, for N species.

Competitive Lotka-Volterra model, for N species.

The out of the eight, variables which the mathematical model from Equation 1.1, seven variables are used in both mathematical equations. The values for the variables are going to be deduced from the research done during the data collection. One such a variable is , which refers to the *population growth rate* of species *i*. Where is the change in population size that occurs in which is a small interval of time. A second such a variable, is the *intrinsic rate of* *increase,* referred to as in the formulais a species’ growth rate without changes in birth/death rates, due to population numbers. Which is calculated as , and denotes birth rate (b) minus the death rates (d) of a species. A third variable present in the model is refers to the population of species *i*. While the variables found in the parenthesis represent the unused portion of the carrying capacity. This is calculated using five variables, in the mathematical model for this research paper while the original model had only four variable. The fourth variable implemented in the model is the *value of* . Then the summation, sigma, uses , which refers to the total number of competing species, and , which refferes to the other population. NOTE that does NOT mean the total population of all competing species. is a fifth variable implemented in the model, and stand for to the *competition coefficient* of species *j* upon *i*. This refers to the extent that each individual of species *j* affects species *i*’s growth rate. Calculation of the competition of species *j* on species *i,* is done as shown in the Expression 1.1. The variable refers to the relative utilization of resource *h* by species *i*, computed as a fraction of the total utilization of all resources for species *i*. A NOTE is to be made which is that is equal to 1.

The sixth variable is , and is the population of *species j*. The final *carrying capacity*, , which is the maximum amount of a specific species that an environment can support. This is calculated as shown in Expression 1.2, and has the inputs as birth rate (b), death rate (d), crowding effect of birth (a), and crowding effect of death (c).

The *modification* made to this model was the introduction of an extra variable (Equation 1.1), which measures the grass availability. To find the last variable, the competition coefficient, the formula used a separate equation7.7 (Schoener, T. 1974). This involved researching each species’ dietary distribution and comparing them against each other. Then multiplying these numbers by the species’ relative consumption amounts, so that it would account for the differences in consumption between species as well. The current *grass amount,* noted with the variable , is calculated effectively as shown in Expression 1.3, with the baseline being the current amount available. An example outcome for this can be seen in Matrix 1.1. More information can be found in appendix 8.1.

***Matrix 1.1:*** *Competition coefficient numbers*

Finally, a second mathematical model is to be used with an equation which is used to calculate the total grass consumption for each animal’s specie. While the model did not need this data as a variable, however, it was required to have in order to answer the main research question of this project. The second equation, shown in Equation 2.1, was introduced as a straight forward equation.

Equation 2.1 takes the total available amount of grass in kilograms (GKg) calculated using the equation showed in Expression 2.1. After converting the area of dry grass (DG) into kilograms and deducting from it the amount being eaten by all herbivores (EH) in one year in kilograms.

To convert the area of DM[[2]](#footnote-2) grass in Oostvaardersplassen, it was needed to do a research of how to convert the area to weight. One of the sources (Co, n.d.), is a company who sell grass for golf clubs, and they sell it in pallets and a standard grass height. Considering that the area of DM grass in Oostvaardersplassen the pallets of grass to cover (*Reference*2016) all this area are deduced, and calculation are made by using the equation found in Expression 2.2.

The methods used to create the mathematical models are based on an assumptions. One such one is that each day, the animals that live in Oostvaardersplassen, consume the same amount of the food. More assumptions can be found under the Discussion section.

## 3.3 Software Development

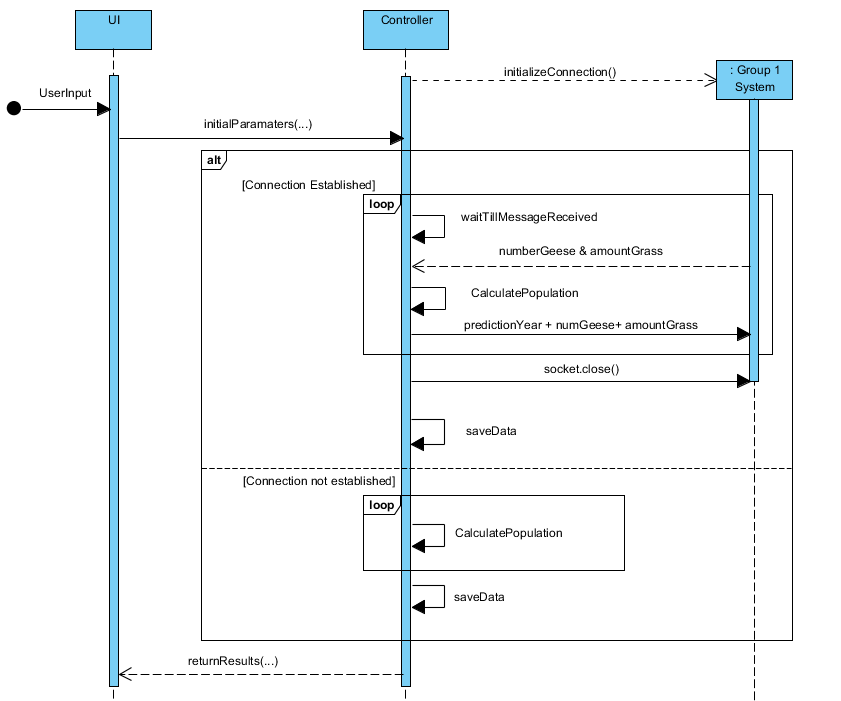
The tertiary step into development is the **software development**, which is divided into two parts the *backend work* and the *frontend work*. However the process of development in this stage begins with neither but by creating *UML[[3]](#footnote-3) diagrams and models*, which include: *Feature List, Domain Model*, *Use Cases Description, Use Case Diagram, Sequence Diagram* and *Class Diagram*. This will help to visualize how the application will look like and what it should do. As soon as, the UML model for the application is completed the focus then turns to the back-end work, implemented using Java. After having the engine and the whole data of the Oostvaardersplassen in the program, the application will then be developed to display the prediction in a GUI[[4]](#footnote-4), via using JavaFX[[5]](#footnote-5).

Thus the process of developing the software is begun by the UML models and diagrams. Firstly, a *feature list* will be made, in this list the specifications of a product can be found. The complete list of features combined together makes the product. The feature list for the application built, during the time period of the research, are to be found in List 3.1.

|  |
| --- |
|  |
| 1. Ability to tweak/customize model via UI  2. Displaying a table with predictions.  3. Showing the prediction in a graph.  4. Add/Edit/Remove data according to user need.  5. Displaying historical data. |

The *domain model* shows the conceptual classes and how they interact to predict population changes and update the UI. It shows the associations necessary between classes that form the relationships between them. The domain model can be seen at Diagram 3.1

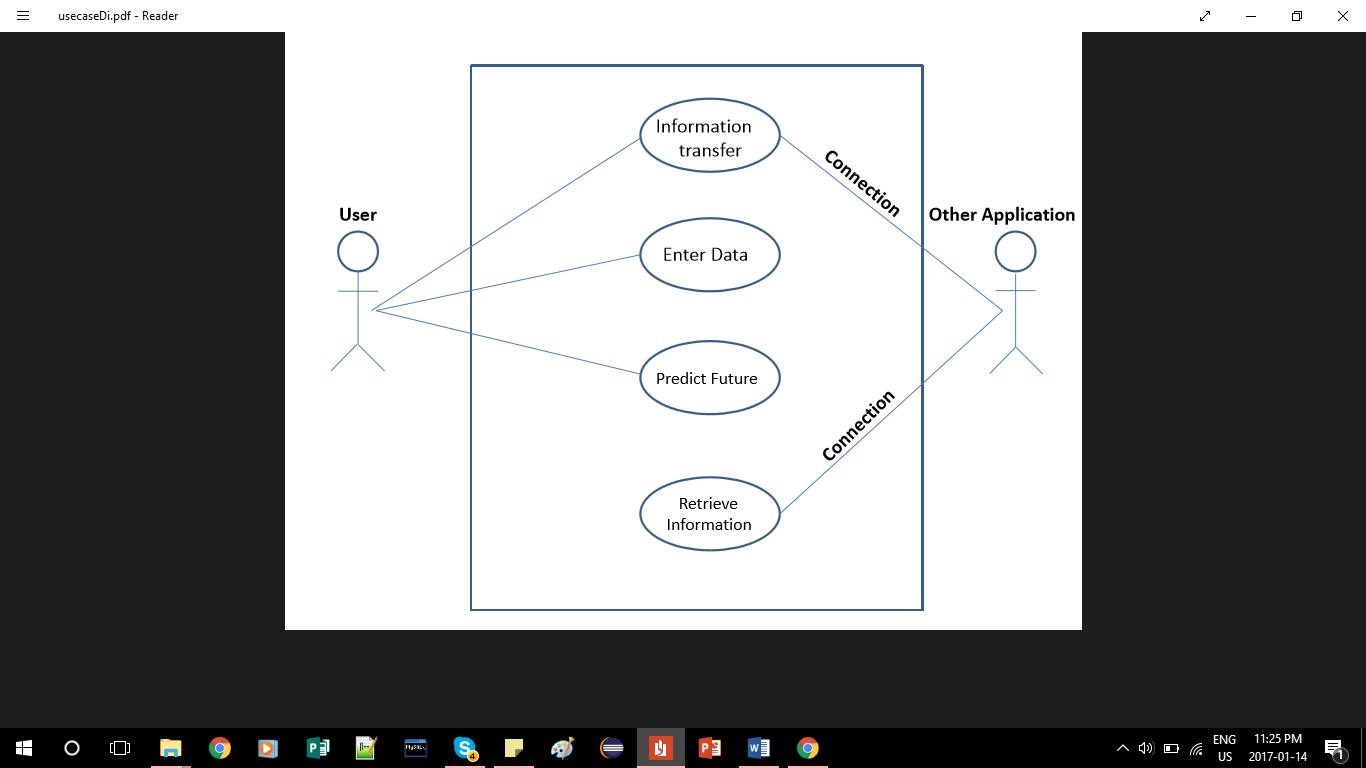
The *sequence diagram* has three lifelines, UI[[6]](#footnote-6), competition and the other Group’s system. Diagram 3.2 shows the sequence diagram for the application built during this research. The application has been structured so that a user can input data into the user-interface and change it to their liking. A class called competition handles the predications and the communication to the other group’s system. The competition class calculates a prediction using mathematical models, waits to retrieve information on the amount of geese and grass from the other Group’s system, calculates the new values, sends a message to the other Group’s system, then saves the data and repeats the process until it has calculated the amount of years issued by the user. The data is then sent back to the UI to be drawn in a graph for the user.



Following the process of development of the UML, the *Use Case* description will follow. Here user goals, sub-goals or intentions are expressed in every step of the basic flow or extensions. The use case description, for this research paper, has only the “happy path”, which means that it is only been taken in consideration a success scenario. The use case description is found under Description 3.1.

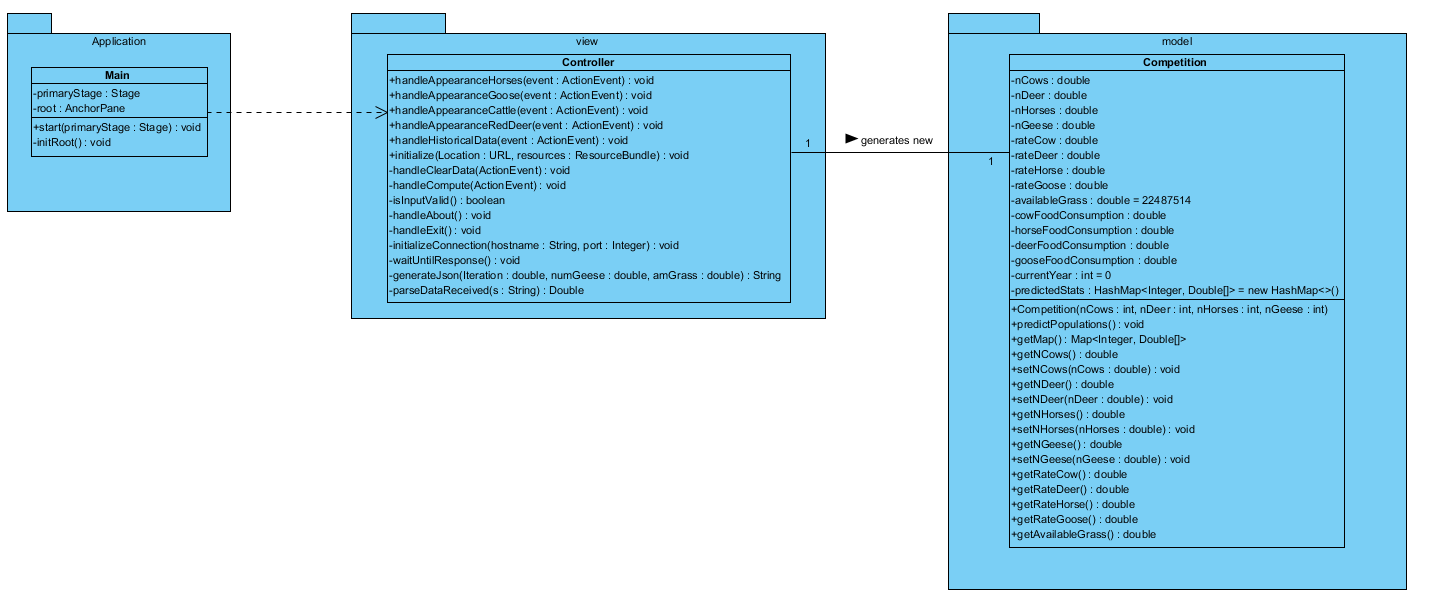
|  |
| --- |
| **Description 3.1** |
| Primary actor: user/client  Stakeholder and Interests:   * User: interested in future prediction. * The other Group: wants to know the population of geese and the amount of grass left, after each prediction.   Precondition:  Correct parameters.  Post-condition:  Prediction is displayed as an outcome on the graph. Information is sent to the other application.  Main Success Scenario:   1. User opens the application. 2. System asks for parameters. 3. User enters variables. 4. System validates variables. 5. System processes variables in the formula. 6. System shows results. 7. System sends results to the other active application. 8. User closes the program. |

The *Use Case Diagram* is referred to the behaviorused to describe a set of actions that some system or systems should or can perform in collaboration with one or more external users of the system. The use case diagram, for this research paper, is found under Diagram 3.3.



In addition to all the diagrams one more is being added which concludes the UML for this paper, which is the *Class Diagram*. This diagram is a reflection the description the structure of a system by showing the system's classes, their attributes, methods, and the relationships among objects.

As can be seen in Diagram 3.4 the class diagram shows the classes used to generate the predictions. The project utilizes MVC [[7]](#footnote-7)software design pattern, as it gives a clear and expandable structure to the project.



Since the application is built dynamically in the *front-end development*, the best choice for creating the Graphical User Interface (GUI) was by utilizing Scene Builder[[8]](#footnote-8). Scene Builder was a shaped fit, in regards to the application’s needs, not only because of the affluent maintainability but also due to its support; as well as the trivial factor of previous work experience with this tool.

The GUI was designed with ease of use in mind, to be simple yet effective. It consists of Menu Bar, Text Fields, Buttons, Radio Buttons and Graph, which visually represent the data.

Inside the *Menu Bar,* a primary functionality, which is needed to be implemented here, is one that allows the user to exit the application. Another functionality, that the GUI will provide, are *Text Fields* where the user can enter custom data, numerical values specifically, that will be used in the Prediction model. Then another functionality refers to the predominant *buttons*, for the application. They are the ones which influence the most the outcome of a prediction, sending the entered data to the mathematical models. The *Radio Buttons* have the functionality which manipulate the chart view. Nonetheless the *Graph View* is another important feature in the GUI, since the application displays its results visually; by drawing charts which then will be compared and analysed by the user*.*

In the *backend work*, the development of the software side, the language the code was written in is Java. By using classes and methods, which are referred to by name and invoked at any point in a program to get a functionality.

The main facilities that the backend is working on is to make the prediction possible. Besides the fact that the key aspect of the backend is to make the GUI work, a lot of other factors have an effect on this work. One of the functionality is to take all the text fields inputs, from the GUI, and process them inside the formula. Also printing to the console, the amount of grass left each year, is yet another feature which is going to be helpful to have implemented. The historical data of the nature is a feature that can be considered optional, however for development purposes this is a very important aspect. By means of this feature, comparison can be done, between the mathematical models created and the past data, with which it can be noted if the predictions are not similar. Then the functionalities in regards to the GUI’s buttons, are also important because they control the opacity of the lines on the graph.

## 3.4 Connection

The last step into development is the **connection** between the two applications. This is done by using communication between the two applications, which is done via the use of sockets. The sockets send JavaScript Object Notation (JSON) [[9]](#footnote-9)objects containing values used in calculating a prediction. The object will be sent, the application will wait until it receives a response from the other application, then the following year’s prediction will commence. The process of communication will be similar for both applications.

# Results

The results of the development of the application, which results in a prediction of the ecosystem, has been developed in sequential steps which involved: **data collection**, which gathered information for the **mathematical models** to be built. Then followed by the **development of the software** was worked on which then led to a **connection**, between two applications.

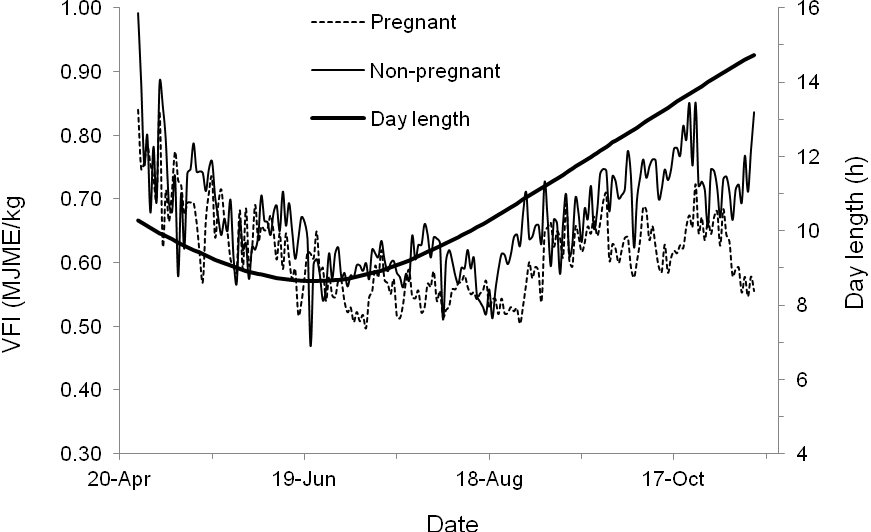
## Results Data Collection

During the **data collection** session a lot of information has been collected, which was to be further used in the development of the project.

The first results, in this research, deal with the consumption of the animals living the nature preserve. The figures with the dietary distribution in percentages, that were found out during the development of the research can be found under the List 4.1.

|  |
| --- |
| **Cattle diet** (Mcinnis, Michael L., and Martin Vavra 1987):  Graminoids 88%  Forbs (Herbaceous Plants) 3%  Shrubs (Woody plants) 8%  **Red Deer diet** (Wieren, Se Van. 1995)**:**  Graminoids 20%  Forbs (Herbaceous Plants) 24%  Ferns (Herbaceous Plants) 10%  Browses (Woody plants) 44%  **Konik Horse diet** (Cosyns, Eric, Tine Degezelle, Else Demeulenaere, and Maurice Hoffmann 2001)**:**  Graminoids 86%  Herbaceous Plants 12%  Woody plants 2%  **Geese diet** (Middleton, Beth A., and A. G. Van Der Valk 1986)**:**  Graminoids 83%  Herbaceous Plants 15%  **Other animals:**  Insects/Snails 2% |

In concordance to the website (Intake requirements n.d.), in regards to the data collection concerning with the food consumption of the *Red Deer*, it shows a distribution of a 50/50 rate of male/female dears. By means of data shown in a graph (Graph 4.1), it was deducted that the average deer weights about 110kg, and consumes about 2.5kg of Dry Matter each day.



Per the website (Konik 2016), used to collect data about *Horse* food consumption, the average horse consumes every day around 2.5% of DM; depending on the horse‘s body weight. Per the source, the horses in Oostvaardersplassen weigh between 350kg-400kg, it was decided on taking an average body weight of a horse of 375kg, which consume 9.4kg of DM each day.

From the source (Cattle 2017), used to generate information about the food consumption of a *Cow*, can be deduced that the average cow consumes 2.5% of its body weight in DM each day. The average weight of a cow in Oostvaardersplassen is 600kg, which means that each cow consumes 15kg of DM on average every day.

Due to lacking information in the data collection in regards to the food consumption of a “wild *Goose*”, a farm goose was decided to be taken as a source for indication. Per the source *(E. Joyner, N. Jacobson, & D. ARTHUR, n.d.)*, the average goose consumes everyday around 0.15kg of DM.

To convert the area of DM grass in Oostvaardersplassen, it was needed to do a research of how to convert the area to weight. Calculation are made by plugging in the values into the equation found in Section 3.2 under Expression 2.2. The calculations are to be displayed under Expression 2.2.1.

The primary source (Co, n.d.), is a company who sell grass for golf clubs in pallets, where each pallet covers 450feet of the area (137.16m) and weigh 2250lbs (1020.583kg). A factor which needs to be taken into consideration that the grass height when being sold, to the customer, is at 0.3cm height. Considering that the area of DM grass in Oostvaardersplassen is 20,000 square meter, which means it is required to have 479 pallets of grass to cover all this area.

The grass at Oostvaardersplassen is at around 14cm per 2015, which means the available amount of food at Oostvaardersplassen. Expression 2.3.1 holds the calculations with the values inserted the equation, can be found in Section 3.2 under Expression 2.3.

What makes this method valid is the fact that in order to get the total yearly amount of grass consumption by each animal, the daily consumption has to be multiplied by 365 days in order to make it a total consumption per year. The reason for changing the daily consumption into a yearly consumption is because the application itself is working in years and not in days. This method is laying over the research that has been done over the food consumption of each herbivore and the geese.

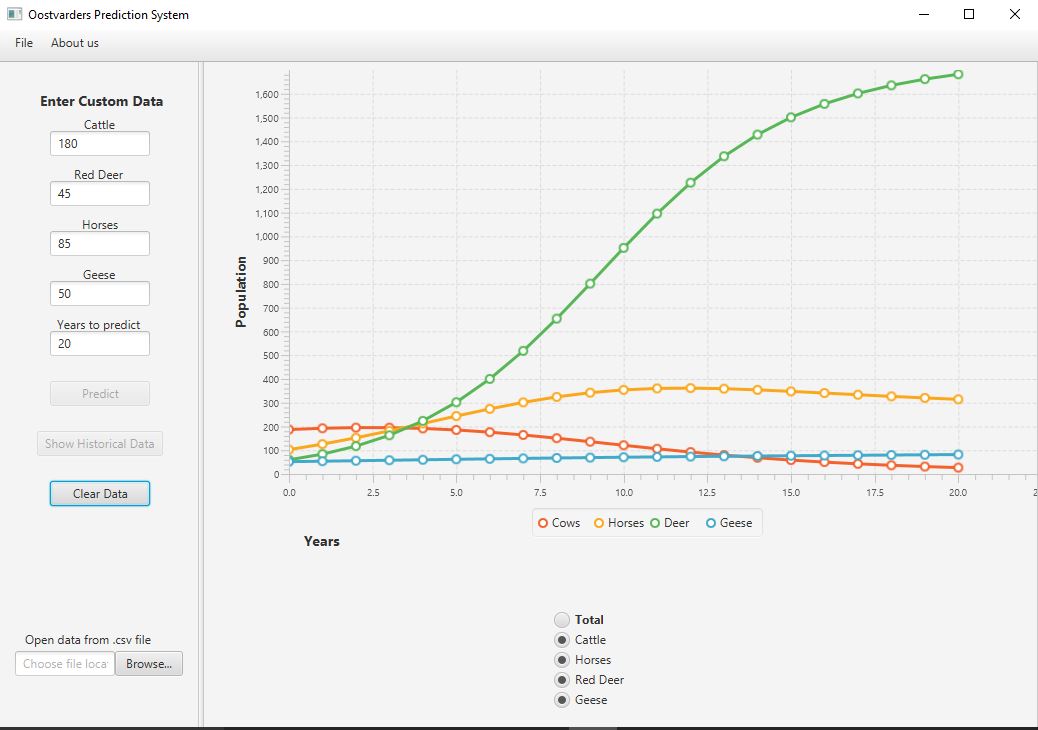
The population’s amounts of the herbivores was also found, fith the help of the most recent reference which was May per 2015. After some research with calculations and some small assumption, it is now known what the estimated herbivores numbers in accordance to 2016.

Per May 2015, there were 250 cattle, or cows, up to May 2016, 80 cows died. The assumption made is that there were 60 cattle born on average for the last few years. This means that there are an approximate 230 cows which are now living in Oostvaardersplassen preservation. In accordance with May 2015, the number of Konik horses was an estimate of 1250, up until May 2016. Out of which 380 horses died, and under supposition is the number of new born which is 250, on average per last few years. Thus, resulting in a population of 1120 horses living in the conservation. Lastly, the Red deer population is counted to be 3200 up to May 2016, in accordance to May 2015. The death rate indicates that 970 deer died, during the past few years. There is an estimate of 700 dears which were born average for the last few years, which means the deer population coming to a numeral of 2930 deer. The Geese are estimated at about 30,000 each year, not taking into account migration.

## Results Mathematical Model

The modified competitive Lotka-Volterra **mathematical model**, was used to develop the application for predictions in regards to future animal’s population. In order to calculate the population growth rate of a specific species several variables had to be found. Those being: the specific species’ population, intrinsic rate of increase, carrying capacity, species’ competing populations, and their competition coefficients.

In order to check the accuracy of the formula, the group made many tests that were successful. One of the test, that was taken, was taking the figures of 1992 of the historical data and put into the application to test the outcome prediction. The figures that were used in the test prediction were: 170 cows, 45 deer, 85 horses (excluding the geese due to the fact they migrate, but taking their food consumption into account), the prediction was for 20 years. Just as expected, the prediction, which can be seen in Figure 1, came out very similar to the reality of the historical data, by the fact the deer population raised enormously and the cows were reaching towards extinction. Few of the only things the application itself can’t predict is the weather effects, such as cold winter/warm summer, on the herbivores and geese migration. Therefore, the accuracy of this application is sufficient for this research’s purpose.

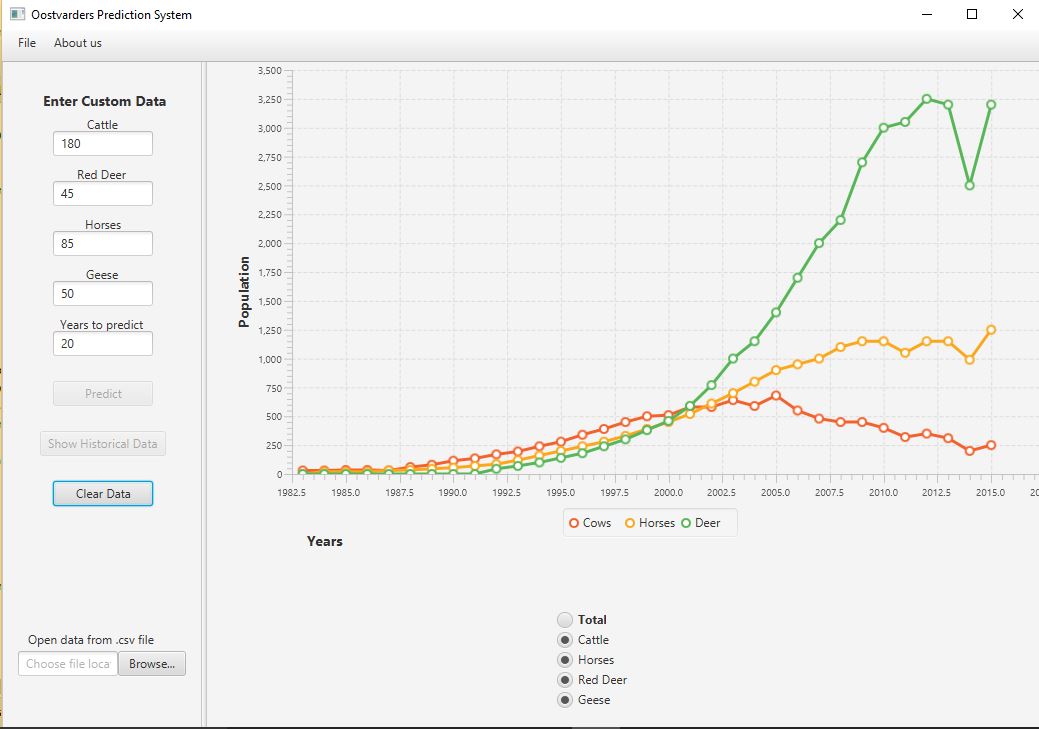


After running several tests with the formula of the competitive Lotka-Volterra model together with new introduced variable G (Grass), several results were discovered. On one hand, the result was showing that cows are reaching adjacent to extinction, after about twenty years from the prediction date. On the other hand, the deer population is reproducing at a much greater speed than any other herbivore. In the first five years, the deer reproduce slower, however, after those five years, they start to take over the area of the nature preserve. This means that after fifteen years the deer population becomes four times greater than the first five years. Unlike the two herbivores, the horses start to decline downwards after a period of thirteen years.

## 4.3 Result Software Development

To implement all the information that is collected the **software development** is done in both frontend development and backend development, with Java and JavaFX. With the intention of showing predictions in a graph, which’s outcome is based on the input that will be inputted by the user.

Due to the dynamic built of the application the best choice for creating the *front-end work,* or the GUI, was by utilizing Scene Builder. The GUI is designed to be simple yet effective interface. It consists of Menu Bar, Text Fields, Buttons, Radio Buttons and Graph, which visually displays the data. The GUI of the application can be seen in Figure 2.



Inside the *Menu Bar,* three main items can be found. A primary item is the close functionality, which allows the user to exit the application, and is found under “File -> Close”. Another item is in “About us”, which opens a window with a brief introduction about the application creators.

“Cattle”, “Red Deer”, “Horses”, “Geese” and “Years” are the *Text Fields* which prompt the user to enter Custom Data, numerical values specifically, that will be used in the Prediction model. The fields are checked by application for correct entry; if the user enters a word value instead of a numerical one, or leaves area empty – an error message will be displayed.

The predominant *buttons*, for the application, are the ones which influence the most the outcome of a prediction. “Predict” is one of the buttons, and its task to take entered data and sends it to Model to execute calculation, then display results in View. In addition to the prediction, the “Predict” Button would also print a message to the console about the grass availability at the end of the calculation. A second button is “Show Historical Data”, which draws the chart of historical data for cattle, horses and deer from 1982 to 2015. The last button resets the Text Fields and Graph view so that the user can renter data for a new prediction, entitled “Clear Data”.

The main purpose of the *Radio Buttons* is to manipulate the chart view. The user can select to display one or more preferred animal graphs or use “Totals” button to show all graphs together.

Since the application displays its results visually, by drawing charts which then will be compared and analysed by the user, the *Graph View* takes up the largest part of the application window. The chart will automatically adjust its scale to given timeframe and graphs. Each animal is assigned a unique colour for better readability.

The *backend work* implemented on the software side, is developed in Java code. By using classes and methods, which are referred to by name and invoked at any point in a program to get a functionality. Below can be found a brief description of the main classes and methods that make the application functional.

Following are descriptors of the class used in the package, fund under Application entitled Model, with the most important aspects. The class is the Competition Class, this class will handle all the predictions*.*

It is composed of a general constructor entitled *Competition (),* and has initial data the following parameters: nCows (The initial number of cows), nDeer (The initial number of Deer), nHorses (The initial number of Horses), and nGeese (The initial number of Geese).

One of the methods used in this class is*PredictPopulations(),* this method will calculate the competition of the animals. A next method used is *getMap(),* which returns a map containing the year as a key-value and values. An instance of an output for this method can be found under Example 1.

Method *GenerateJson()* generates a JSON object to be sent to the other application. Example 2 shows an instance of the format.

The method *parseDataReceived(String s)* takes a string in a json format and takes the data out and applies it to relevant variables that needs to be updated. A simple example is: When application one sends the Grass and Geese amounts application two updates the Grass and Geese variables.

The View package, located inside the Application, contains one of the most important classes the Controller Class. This class makes possible the connection between the backend work and the frontend work.

The method *HandleClearData(),* clears the LineChart and the Series Objects, the Series objects contain the points to make up each line drawn. It checks if clearData has been button has been armed if so, un-disables compute and showHistory buttons.

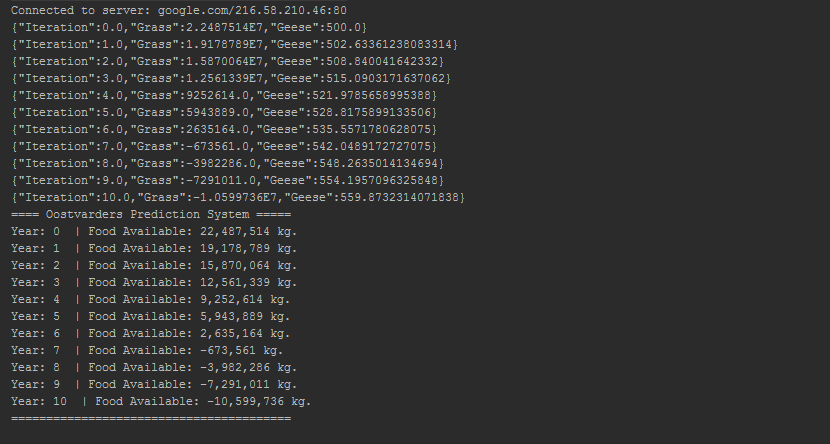
Then follows methods which in essence do similar tasks but for different types. They are the *handleApperanceCattle(), handleAppearanceGoose(), handleApperanceHorse(),* and *handleApperanceRedDeer.* All these methods removes the Series (Object that contains cords for a specific animal line) from the chart, if the radio button is not selected. As well as has the option to add them back if it is selected again.

The next method is *handleCompute()*, which has several tasks performing init. Firstly it takes numbers from input and makes them variables. Secondly creates a map to store data, followed by the creation of a Competition object and fills parameters with variables made. Then a loop, initialized from 0 until the year inputted by the user, calls the predictPopulations method inside it and fills the map to store data. Formerly loops through the map and adds the data to a Series (Object that holds information to draw the lines for each animal). Subsequently the method then disables compute and showHistory buttons. The final steps are printing information, from the data looping (i.e each predicted year), and adds all the Series into the LineChart object.

The *isInputValid()* method checks that the text fields only contains numbers, and no other characters. *handleHistoricalData()* is a method which adds historical data to Series Objects then adds those to LineChart. Dialog box is an alert about creators of application, and is activated by the *handleAbout()* method. The *handleExit()* is the method which helps the user to exit the application.

## 4.4 Result Connection between Application

The **communication** between the two applications is done via the use of sockets, which send JavaScript Object Notation (JSON) [[10]](#footnote-10)objects containing values used in calculating a prediction. The object will be sent, the application will wait until it receives a response from the other application, then the following year’s prediction will commence. The process of sending the JSON object is as follows: firstly one application will be waiting to receive the data, which is the amount of grass and number of geese, from the other application, from group one. Then calculations will kick in to process the new data. Following a new JSON object will be sent with the information to the other application. This information is the amount of Grass, the population of Geese and the YearOfThePrediction, referred to as the iteration. The outcome in the console, which proofs that the connection works, can be found in Diagram 4. This process will repeat until the last year that is predicted is reached. This process will be similar for both applications.

The work to implement the connection is by the backend work, and is found under the Controller Class. One of the methods that handle this task is the*initialize()* method, whichinitializes variables and the connection. While the method *initializeConnection()* initializes the connection to the socket. Another method that is involved, in the process of the communication between the applications, is the *waitUntilResponse*(). This method makes thread which wait 250ms until it receives something from the socket. It will make the program look like it is not responding at times, but this is “normal”.

It is to be noted that the tests for the connection has only been done on the application that this research report has been working on. Thus a real connection between the two applications has yet to be done.

# Conclusion

Rewrite. In a conclusion, you need to

* Repeat the main question and subquestions
* Answer the subquestions one by one, referring to previous chapters (no new facts here)
* Answer the main question, referring to the answers to subquestions if necessary
* Answer the main question from the programme

Deriving from the programme research question the Main Research Question for this research paper, at the project level, has in focus the competition between the herbivores.

How do the herbivores and the geese compete for the grass in an enclosed area?

2.2 Research sub-questions

The research sub-questions are questions which derive from the main research question, and provide the path to the answer of the research paper.

Sub-Research Questions for this research paper:

1. What is the rate of grass consumption of each animal?

2. How do animal populations change as a result of grass availability?

3. What are the populations of each animal?

4. How can the start data and historical data help shape existing mathematical models to an equation which fits the problem in Oostvaardersplassen?

5. How can the application hold all the implementations that are being asked?

Although confidence was to be found in the formulas and data as presented, there were some issues with the data collection that could not be solve. An instance is the lack of available data for certain species. As such, certain assumptions were made to account for these issues. A full list of issues and assumptions with regards to the data collection can be found in the appendix.

The **Equation research** is mostly focused on the equations of the grass growth. The research conducted to find equations is based on literature (Gotelli, N. J. 2008). Transforming grass amount into actual weight so a comparison can be made to the amount of food the herbivores and geese are eating.

During the project development, several limitations may occur, that could harm the result of the application. Some of those **Limitations** for this research are the time frame which project needs to be finished. The information availability constraints which might be lacking data or not accurate at all. Another challenge is the interdependence on the other group that is investigating relating variable. This research has limited scope to Oostvaardersplassen preservation.

To answer the **second sub-question**, research needs to be done about what is the formula that most fit the situation. To deduce how the availability of grass would affect the animal's population that are leaving in the ecological preservation. A book of the ecological system was given as a reference in order to figure out which formula would best fit to the situation being faced.

The general conclusion that can be drawn, from this research, is that after about twenty years, from now, the cows in Oostvaardersplassen will reach extinction. Per the prediction, the cows reach their peak point after seven years, and from that point, they decline downwards towards extinction. Another conclusion is directly related to the deer’s population. The deer will keep populate if they can, and if the will not be grass an issue in this competition. In relation to the graph, the deer population can reproduce in twenty years by thirty-four times. This means that the deer can reach its peak point after twenty years. In contrast to the deer population, when it comes to the horses, the result shows that the horses are reaching their peak point after thirteen years from the prediction day. That point insinuates a slight declining in the population size each year, but extinction will not be reached.

Per the conclusion and the result that was faced, several important recommendations can be made towards the client of Oostvaardersplassen. One of the most important recommendations that can be given is to expand the territory of the dry land, and potentially to dry more land to allow cows to reproduce in an open environment. Another suggestion that can be made is to bring more cows to the nature preserve, and considering to get rid of at least half of the deer population. The reason for such can explain that in nature the deer reproduce too fast, and means that they are taking away an important amount of food from the cows; due to the fact, they share the most surface for food with them cows.

A new variable was introduced into this formula which is the grass, in order to test the effect of the grass on the herbivores living there. The improved formula now takes the grass consumption of the herbivores together with the available grass in Oostvaardersplassen. To test the effect of how animal populations, change, because of grass availability, the application needs to confirm all the inputs given. Then the system will use the formula to predict each year, protracted until there is no available grass to be consumed. The application will draw a graph of the effect of the grass availability on the herbivores, showing the effect on the population over years if there is available grass to consume. Therefore, the application will not continue the prediction for the rest of the years where there is no available grass to consume by the herbivores. Which means that starvation will be reached by the animals.

# Discussions

Below is a list of issues with the data collection, followed by the assumptions made for them where needed.

* One of the assumptions was a linear correlation between grass availability and carrying capacity, which may not hold in real-life scenarios.
* The formula for competition coefficients uses percentages of dietary overlap but does not consider differences in the amount of consumption.
  + Multiplying the coefficients by the relative consumption amounts to find a more accurate number.
    - Using daily Dry Matter intake for relative consumption amounts, so this calculation does not account for the difference in digestive efficiency.
* The model does not account for the separate types of geese.
  + By using the three species of geese present and averaged them as one species.
* For all species, the dietary information was not available for the specific region modelled.
  + By using the dietary information from other locations.
* Spring data was unavailable for Konik horses,
  + - Averaging the summer-winter combined since the research displayed seasonal percentages.
* Terminology was not completely consistent
  + Categorized was done by using the Konik Horse’s data from descriptions.
    - Not being experts in biology errors cannot rule out in dietary descriptions completely.
* The numbers used to calculate the competition coefficient don’t add up to 100% due to rounding.
* The dietary information for the specific types of geese was too difficult to find.
  + Thus, the research used dietary information for Greylag and Barheaded geese combined.
* The only available dietary information for geese was limited to the months of November-April
  + Finding dietary information for other seasonal periods proved impossible, however, the research uses it anyway.
* The dietary information for Heck Cattle was problematic to find.
  + Thus, dietary information for Wild Cattle is used instead.

# Bibliographies

Startpagina Staatsbosbeheer. Het zit in onze natuur. (n.d.). Retrieved November 23, 2016, from <https://www.staatsbosbeheer.nl/>

Gotelli, N. J. (2008). *A Primer of Ecology* (4th ed.). Sunderland, Massachusetts: Sinauer Associates.

Intake requirements. (n.d.). Retrieved January 06, 2017, from <http://deernz.org.nz/deerhub/feeding/feeding-deer/intake-requirements>

Konik (2016). In *Wikipedia*. Retrieved from <https://en.wikipedia.org/wiki/Konik>

SARAH, R. N. (2016). Nutritional requirements of horses - management and nutrition - veterinary manual. Retrieved January 6, 2017, from <http://www.merckvetmanual.com/mvm/management_and_nutrition/nutrition_horses/nutritional_requirements_of_horses.html>

Cattle. (2017). How much grass does a cow eat each day? Retrieved January 6, 2017, from <http://www.answers.com/Q/How_much_grass_does_a_cow_eat_each_day>

E . Joyner, D., N. Jacobson, B., & D. ARTHUR, R. Retrieved January 1, 2017, from Nutritional characteristics of grains fed to Canada Geese, <http://wildfowl.wwt.org.uk/index.php/wildfowl/article/viewFile/773/773>

Schoener, T. (1974). Some Methods for Calculating Competition Coefficients from Resource-Utilization Spectra. *The American Naturalist,* *108*(961), 332-340. Retrieved from <http://www.jstor.org/stable/2459895>

Co, F. G. Sod installation | Fulton, Arkansas | Fulton grass Co. Retrieved January 8, 2017, from <http://www.fultongrass.com/faq>

Mcinnis, Michael L., and Martin Vavra. "Dietary Relationships among Feral Horses, Cattle, and Pronghorn in Southeastern Oregon." *Journal of Range Management*40.1 (1987): 60. Web. < <http://oregonstate.edu/dept/eoarc/sites/default/files/publication/309.pdf> >.

Wieren, Se Van. "Composition and Cell Wall Content of Diets Selected by Roe Deer, Red Deer, Highland Cattle and Konik Ponies." *Annales De Zootechnie* 44.Suppl. 1 (1995): 121. Web. <<http://rcin.org.pl/Content/12794/BI002_2613_Cz-40-2_Acta-T43-nr6-77-94_o.pdf>>.

Cosyns, Eric, Tine Degezelle, Else Demeulenaere, and Maurice Hoffmann . "Feeding Ecology of Konik Horses and Donkeysin Belgian Coastal Dunes and Its Implicationsfor Nature Management." N.p.,(Dec. 2001). Web.<<https://www.researchgate.net/publication/242600674_Feeding_ecology_of_Konik_horses_and_donkeys_in_Belgian_coastal_dunes_and_its_implication_for_nature_management> >

Middleton, Beth A., and A. G. Van Der Valk. "The Food Habits of Greylag and Barheaded Geese in the Keoladeo National Park, India." *The Food Habits of Greylag and Barheaded Geese in the Keoladeo National Park, India | Middleton | Wildfowl*. N.p., (1986). Web. <<http://wildfowl.wwt.org.uk/index.php/wildfowl/article/view/774> >

"How Much Does a Pallet of Sod Weigh?" *Reference*. N.p., n.d. Web. Dec. 2016. [https://www.reference.com/home-garden/much-pallet-sod-weigh-390e9d062e1a93a8#](https://www.reference.com/home-garden/much-pallet-sod-weigh-390e9d062e1a93a8)

# Appendices

## 8.1 Competition Coefficient Calculations

Competition coefficient calculations, **BEFORE** calculating the difference in total consumption.

CATTLE = 1 DEER = 2 HORSES = 3 GEESE = 4

A\_11 = 1

A\_12 = = = = 0.2832

A\_13 = = = = 0.9748

A\_14 = = = = 0.9401

A\_21 = = = = 0.6340

A\_22 = 1

A\_23 = = = = 0.6346

A\_24 = = = = 0.6214

A\_31 = = = = 1.010

A\_32 = = = = 0.2937

A\_33 = 1

A\_34 = = = = 0.97

Competition coefficient calculations, **AFTER** calculating the difference in total consumption.

CATTLE = 1 DEER = 2 HORSES = 3 GEESE = 4

A\_11 = 1

A\_12 = 0.2832 \* = 0.0472

A\_13 = 0.9748 \* = 0.6109

A\_14 = 0.9401 \* = 0.009401

A\_21 = 0.6340 \* = 3.804

A\_22 = 1

A\_23 = 0.6346 \* = 2.386

A\_24 = 0.6214 \* = 0.03728

A\_31 = 1.010 \* = 1.612

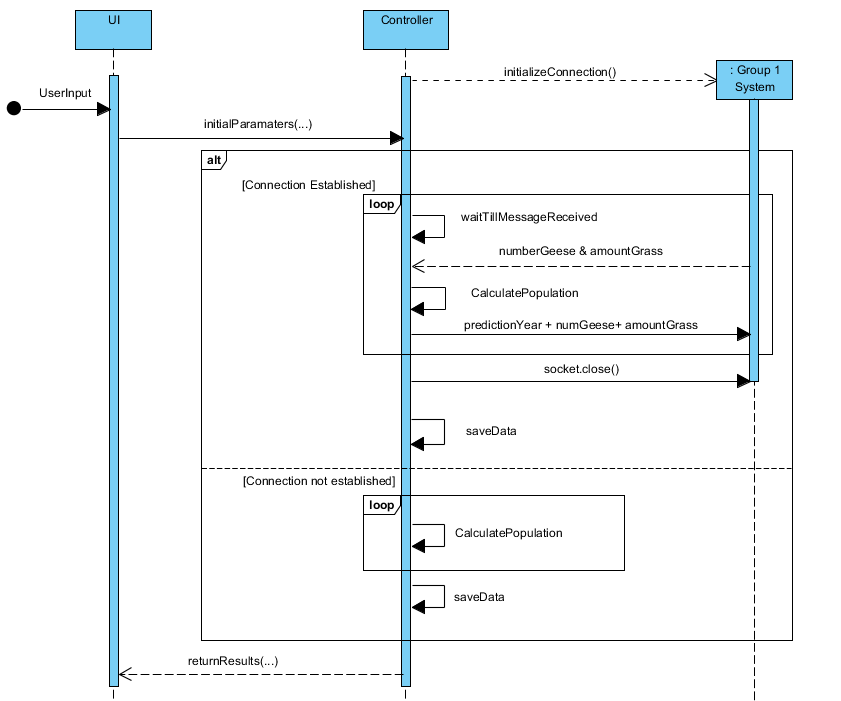
A\_32 = 0.2937 \* = 0.07811

A\_33 = 1

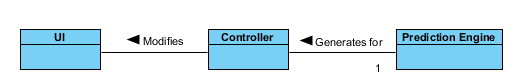
A\_34 = 0.97 \* = 0.01548

## 8.2 UML Diagrams

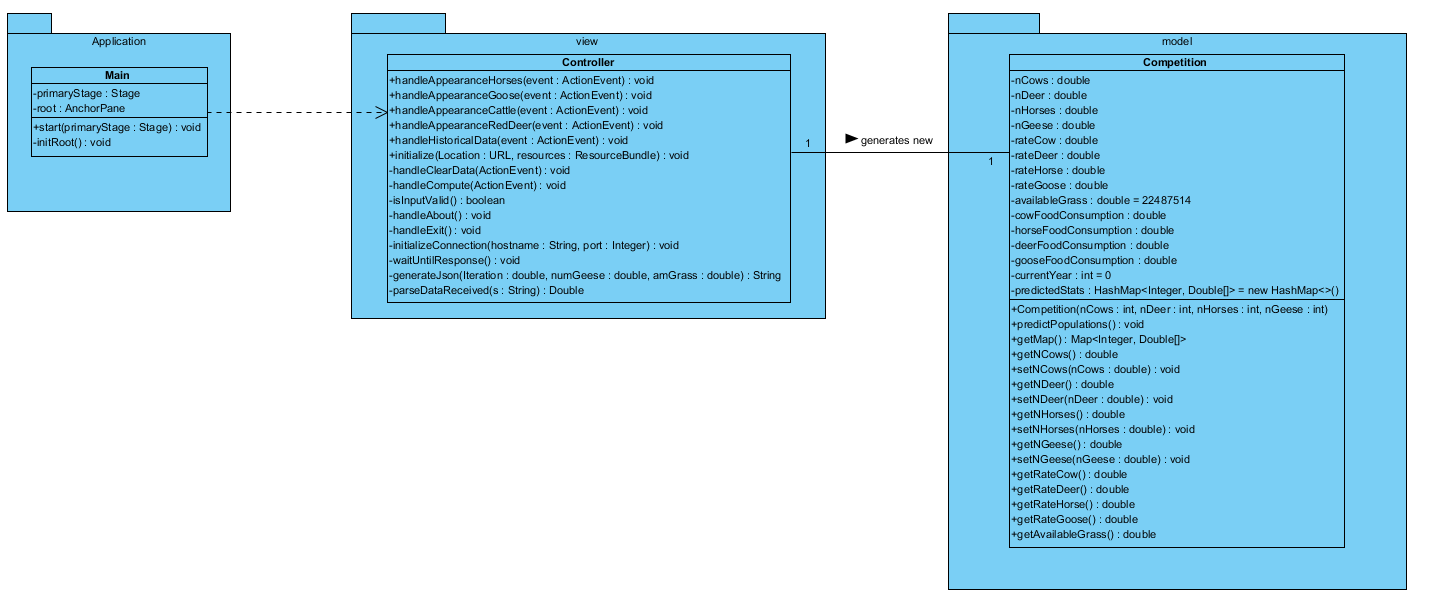
**Sequence Diagram**

****

**Domain Model**



Whatever this is, it is not a domain model.

****

**Class Diagram**

1. SPSS – Statistics is a software package used for logical batched and non-batched statistical analysis. [↑](#footnote-ref-1)
2. DM – abbreviation for dry matter; [↑](#footnote-ref-2)
3. UML – Unified Modeling Language is a general-purpose, developmental, modeling language in the field of software engineering, which intends to provide a standard way to visualize the design of a system. [↑](#footnote-ref-3)
4. GUI – Graphical User Interface is a type of user interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, instead of text-based user interfaces, typed command labels or text navigation. [↑](#footnote-ref-4)
5. JavaFX - software platform for creating and delivering desktop applications, intended as the standard use of GUI library for Java. [↑](#footnote-ref-5)
6. UI – the user interface or UI, is everything designed into an information device with which a person may interact with. [↑](#footnote-ref-6)
7. MVC – the model view controller or MVC is one of the pattern of ASP.NET; and is used to build web applications. [↑](#footnote-ref-7)
8. Scene Builder – is a visual layout tool for JavaFX applications. The JavaFXScene lets a user rapidly project a JavaFX application user interfaces, without coding. [↑](#footnote-ref-8)
9. JavaScript Object Notation (JSON) – is a lightweight data-interchange format. [↑](#footnote-ref-9)
10. JavaScript Object Notation (JSON) – is a lightweight data-interchange format. [↑](#footnote-ref-10)