WASHINGTON STATE UNIVERSITY
VANCOUVER

CS 320 Course Project - Software Design Document

for

Discrete Probability Calculator

Version 1.0

Prepared by

Group Name: Group 2

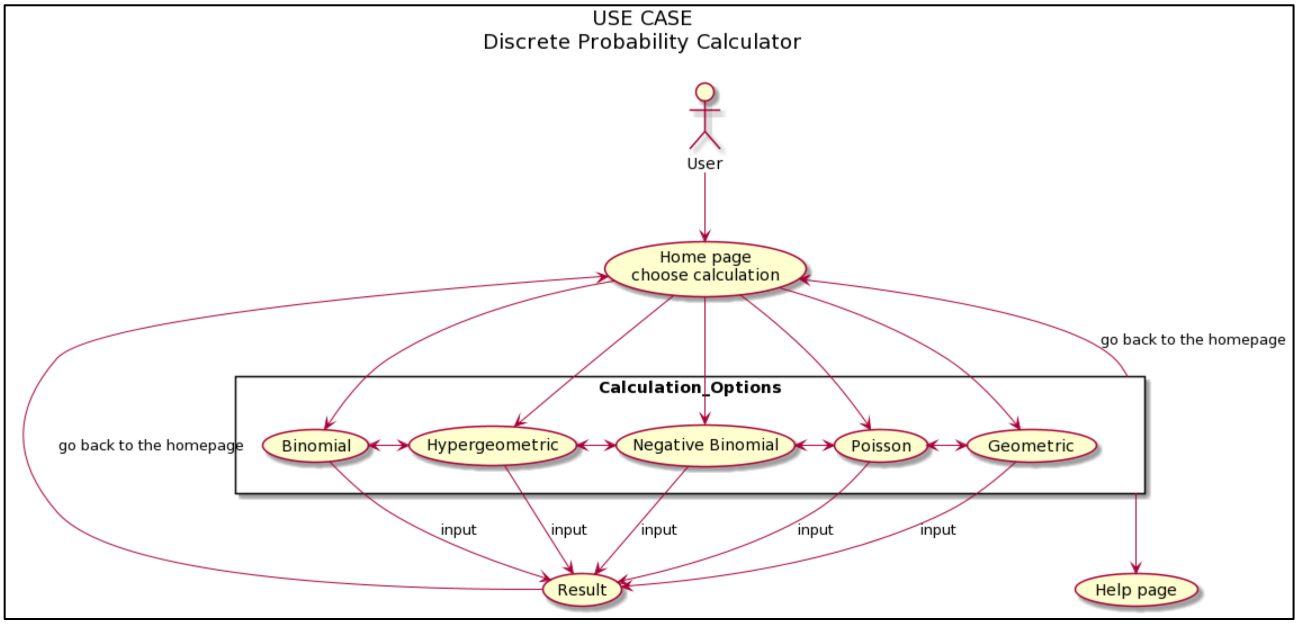
Patrick Tsai	**011709316**	**patrick.tsai@wsu.edu**
Zhicheng Zhou	**011735296**	**zhichengzhou123@gmail.com**
Vuochlang Chang	**011706492**	**vuochlang.chang@wsu.edu**

Date: **11/21/2020**

CONTENTS

1 Introduction

The Discrete Probability Calculator is a simple website-based application that users can use to calculate the probability of the discrete distribution. There are multiple ways where the user can access the calculations for each distribution. As shown in *Figure 1*, the user can access each distribution mainly from the homepage and directly from the other distributions as well.



(Figure 1)- Use Case Diagram

The overall goals of the system are:

1. To provide easy access to the users to calculate discrete probabilities; Users can access DPC website through any browsers whether from their smartphones or computers
2. To provide as another option from a TI-84 calculator or other versions
3. To help users to understand each probability better with the Help sections
4. To improve our understanding of building a web application that interacts with user input.

The DPC system is intended for study purposes, its intended use is for student/academic questions. However, the DPC system is NOT intended to be used as a tool to make any real-life decisions, as that any probabilities derived from the program should not be used to make important life decisions.

1.1 Project Overview

In this Software Design Document, it is covered over the DPC system requirements using system modeling. The SDD is divided into three parts such as:

1. The activity diagram-this section focuses on the external perspective on the context of each scenario of DPC
2. The class diagram-this section focuses on the structural perspective on the organization of each scenario of DPC
3. The behavioral diagram-this section focuses on the behavioral perspective on the dynamic behavior of the DPC system. Both sequence diagram and state diagram apply to our project, the data driven sequence diagram of our project shows the flow of the program with different inputs (correct and incorrect), and the event driven state diagram of our project shows how each distribution work under certain circumstances.

1.2 Definitions, Acronyms and Abbreviations

TERMS	DEFINITION
SDD	Software Design Document
DPC	Discrete Probability Calculator
Input	The given data from the User
User	The person who is using the DPC software
Discrete Probability Distribution	Discrete Probability distribution describes the occurrences of every possible value of a random discrete variable, discrete means that the random variable can only take on some non-negative, non-decimal integer value such as: 0 -> positive infinity
Binomial Distribution	Binomial distribution counts the number of successes in some repeated independent experiments, each trial can only consist two possible outcomes: SCCUESS OR FAILURE. The parameters for Binomial distribution are: 1. The number of trials. 2. The number of successes. 3. The probability of a success on a given trial (the probability of a success may not vary)

Hypergeometric Distribution	Hypergeometric distribution is similar to Binomial distribution but with the variance in the probability of a success. In Hypergeometric distribution, the given probability changes for each trial due to the replacement of samples. The parameters for Hypergeometric distribution are: 1. The number of items in total. 2. The number of items that are classified as succuss in total items. 3. The number of items in the chosen sample. 4. The number of items that are classified as succuss in the chosen sample. 5. The number of success of failure that needs to be found.
Negative Binomial Distribution	Negative Binomial distribution is looking for the probability of the number of trials takes to produce n successes. The parameters for Negative Binomial distribution are: 1. The number of successful trials. 2. The kth trial where the number of successful trials happen on.
Poisson Distribution	Poisson distribution is counting the probability of some number of outcomes can occur during a given time interval (usually represented as μ) The parameters for Poisson distribution are: 1. The mean (expected value) of the random variable. 2. The number of outcomes.
Geometric Distribution	Geometric distribution is counting the probability of the number of trials on which the first succuss can happen. The parameters for Geometric distribution are: 1. The number of trials that takes for the first success to happen. 2. The probability of one success can happen.

1.3 References and Acknowledgments

"How to Cite References: IEEE Documentation Style", IEEEDataPort, Available: https://ieee-dataport.org/sites/default/files/analysis/27/IEEE%20Citation%20Guidelines.pdf*.* [Accessed Nov 12, 2020].

2 Activity Diagrams

The activity diagram section will be divided into five parts where each of the activity diagrams will focus on their own significant requirements. The five activity diagrams are based on each scenario within the Calculation_Options as shown in the *Figure 1*. Those five scenarios are:

1. Activity diagram for Binomial distribution
2. Activity diagram for Geometric distribution
3. Activity diagram for Hypergeometric distribution
4. Activity diagram for Negative Binomial distribution
5. Activity diagram for Poisson distribution

Note: the following activity diagrams are following a similar pattern; the only differences are the required input and the synchronized testing rules to ensure that the user has provided reasonable data to compute.

2.1 Activity diagram for Binomial Distribution

In this activity diagram, we are assuming the user is in the Binomial distribution section in the start state. The start and the end state indicate using the black-rounded circle.

Once the user is in the start state, the user is automatically considered as browsing the site **(Browse).** At this point, the user has 4 options listed as:

1. **(Home page)** - the user can go back to the home page
2. **(Choose other distribution calculations)**- the user can choose other calculations and jump directly to that page from the current calculation
3. **(Help page)**- the user can go to the help page to understand more about the distribution if they have any doubts
4. **(input the required data)** - the user can use the designed box specifically to calculate the probability of the distribution

While options 1 to 3 will take the user to the other page and operate differently, option 4 will keep the user to remain on the current page and go through the calculation.

If the user chooses option 4, the user will have to input the required data such as:

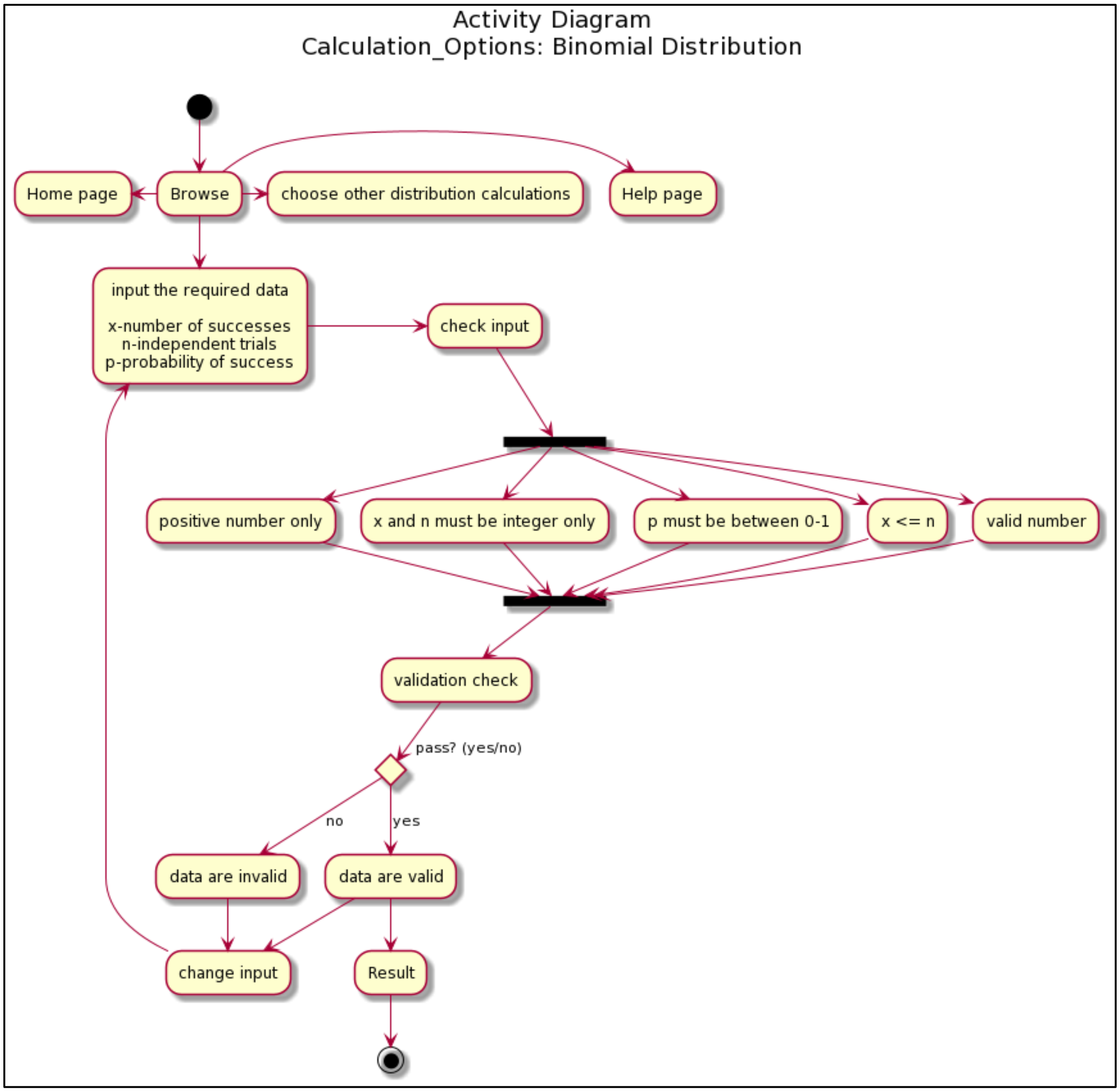
1. **x**-number of successes
2. **n**-independent trials
3. **p**-probability of success

When the user is inputting the data, the data will go through the checking section **(check input).** In that checking section, multiple tests are running synchronically to ensure that all the data are valid and are ready to be computed.

For Binomial distribution, those multiple tests are:

1. test to ensure that all inputs are in number form
2. test to ensure that all inputs are positive
3. test to ensure that x and n must be integer number
4. p must be between 1 and 0
5. x must be less than or equal to n

If any of the tests have failed, then it will prevent the system from doing the computation until the user input the correct data. Once the data are valid, then the user can click on the calculate button to start computing, or change the data as needed.



2.2 Activity diagram for Geometric Distribution

In this activity diagram, we are assuming the user is in the Geometric distribution section in the start state. The start and the end state indicate using the black-rounded circle.

Once the user is in the start state, the user is automatically considered as browsing the site **(Browse).** At this point, the user has 4 options listed as:

1. **(Home page)** - the user can go back to the home page
2. **(Choose other distribution calculations)**- the user can choose other calculations and jump directly to that page from the current calculation
3. **(Help page)**- the user can go to the help page to understand more about the distribution if they have any doubts
4. **(input the required data)** - the user can use the designed box specifically to calculate the probability of the distribution

While options 1 to 3 will take the user to the other page and operate differently, option 4 will keep the user to remain on the current page and go through the calculation.

If the user chooses option 4, the user will have to input the required data such as:

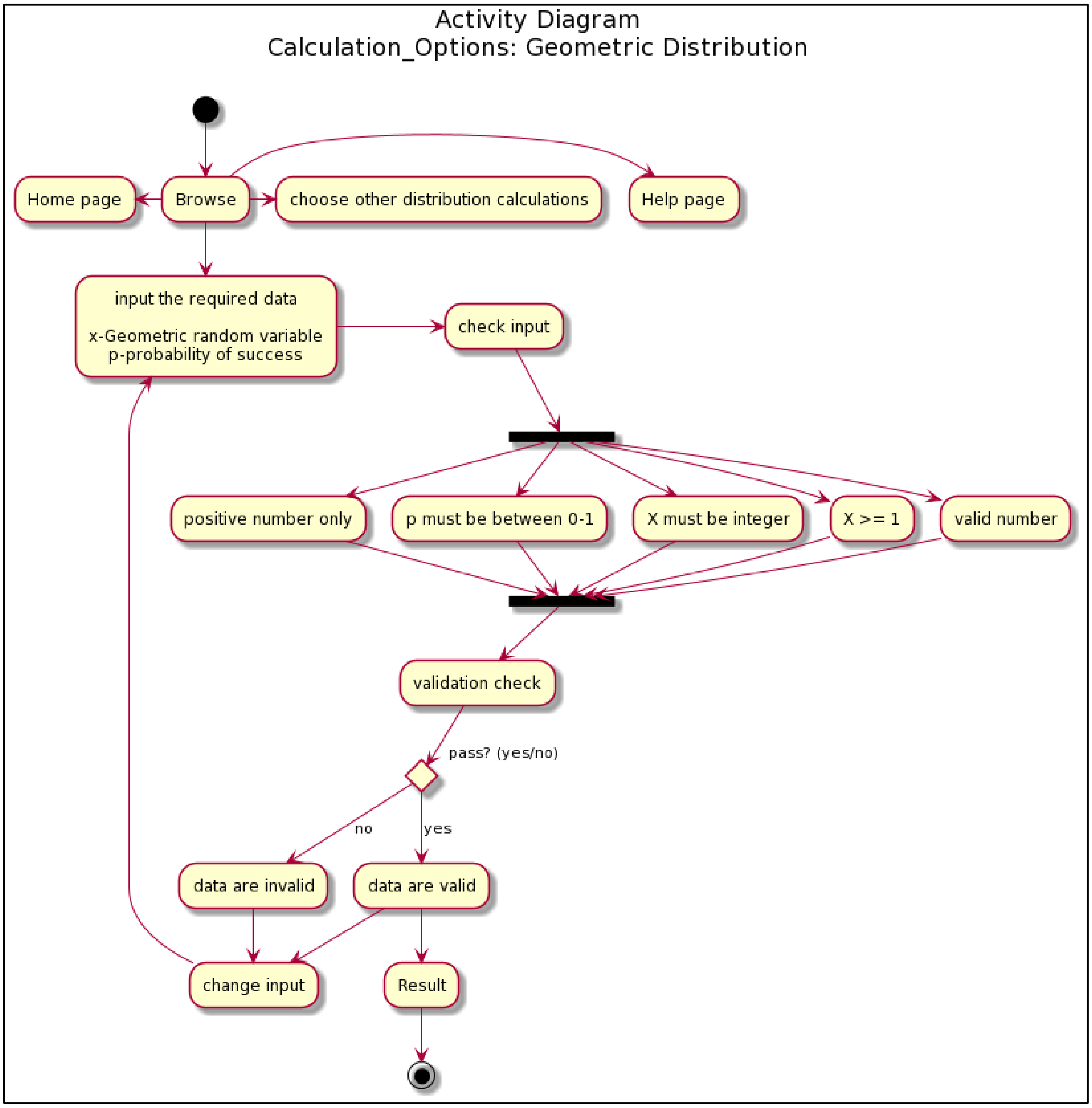
1. x-Geometric random variable
2. p-probability of success

When the user is inputting the data, the data will go through the checking section **(check input).** In that checking section, multiple tests are running synchronically to ensure that all the data are valid and are ready to be computed.

For Geometric distribution, those multiple tests are:

1. test to ensure that all inputs are in number form
2. test to ensure that all inputs are positive
3. test to ensure that x must be integer number
4. p must be between 1 and 0
5. x must be less than or equal to 1

If any of the tests have failed, then it will prevent the system from doing the computation until the user input the correct data. Once the data are valid, then the user can click on the calculate button to start computing, or change the data as needed.



2.3 Activity diagram for Hypergeometric Distribution

In this activity diagram, we are assuming the user is in the Hypergeometric distribution section in the start state. The start and the end state indicate using the black-rounded circle.

Once the user is in the start state, the user is automatically considered as browsing the site **(Browse).** At this point, the user has 4 options listed as:

1. **(Home page)** - the user can go back to the home page

2. **(Choose other distribution calculations)**- the user can choose other calculations and jump directly to that page from the current calculation
3. **(Help page)**- the user can go to the help page to understand more about the distribution if they have any doubts
4. **(input the required data)** - the user can use the designed box specifically to calculate the probability of the distribution

While options 1 to 3 will take the user to the other page and operate differently, option 4 will keep the user to remain on the current page and go through the calculation.

If the user chooses option 4, the user will have to input the required data such as:

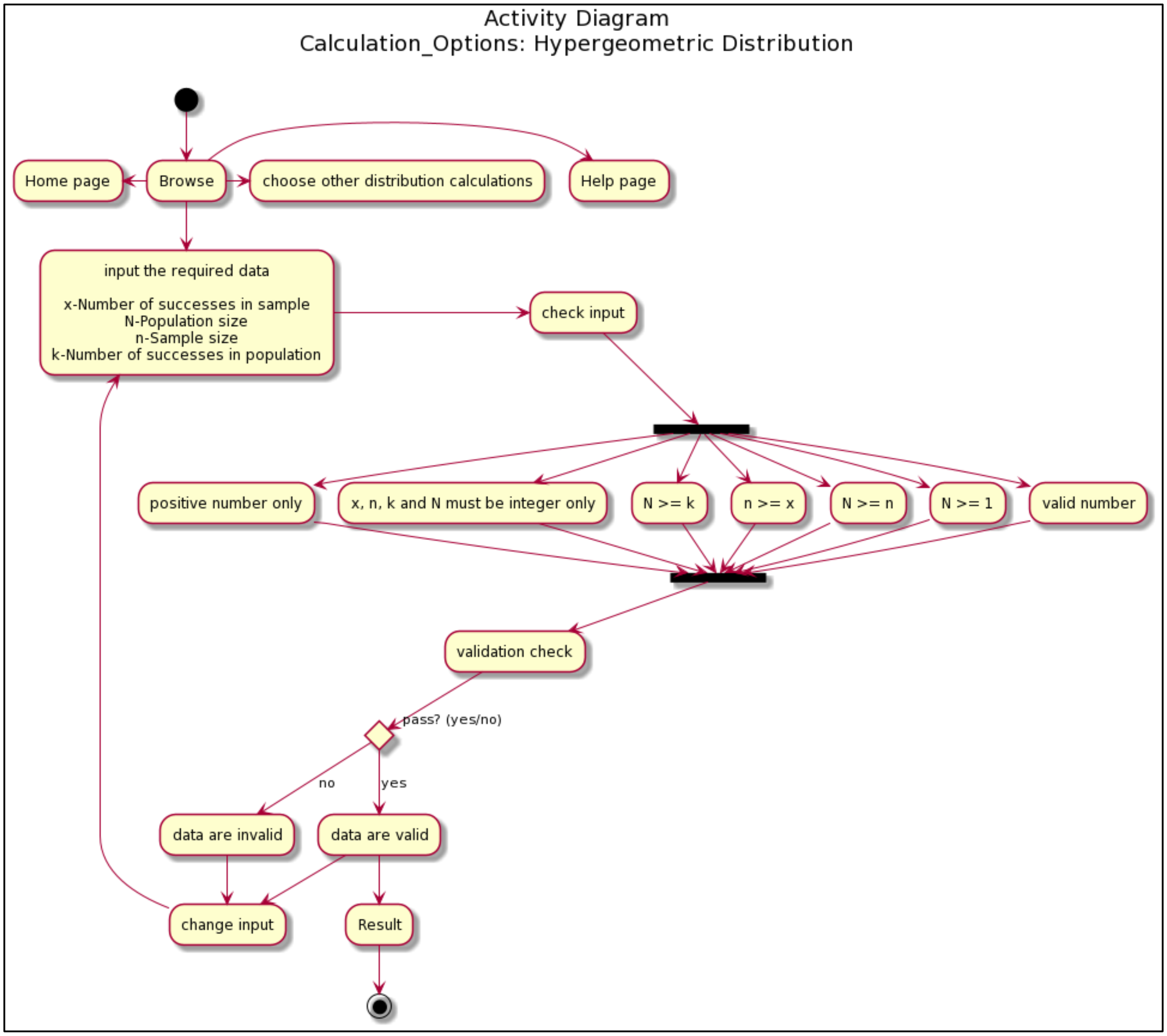
1. **x-**Number of successes in sample
2. **N**-Population size
3. **n**-Sample size
4. **k**-Number of successes in population p-probability of success

When the user is inputting the data, the data will go through the checking section **(check input).** In that checking section, multiple tests are running synchronically to ensure that all the data are valid and are ready to be computed.

For Hypergeometric distribution, those multiple tests are:

1. test to ensure that all inputs are in number form
2. test to ensure that all inputs are positive
3. test to ensure that x, n, k, and N must be integer number
4. N is greater or equal to k
5. n is greater or equal to x
6. N is greater or equal to n
7. N must be greater than or equal to 1

If any of the tests have failed, then it will prevent the system from doing the computation until the user input the correct data. Once the data are valid, then the user can click on the calculate button to start computing, or change the data as needed.



2.4 Activity diagram for Negative Binomial Distribution

In this activity diagram, we are assuming the user is in the Negative Binomial distribution section in the start state. The start and the end state indicate using the black-rounded circle.

Once the user is in the start state, the user is automatically considered as browsing the site **(Browse).** At this point, the user has 4 options listed as:

1. **(Home page)** - the user can go back to the home page
2. **(Choose other distribution calculations)**- the user can choose other calculations and jump directly to that page from the current calculation

3. **(Help page)**- the user can go to the help page to understand more about the distribution if they have any doubts
4. **(input the required data)** - the user can use the designed box specifically to calculate the probability of the distribution

While options 1 to 3 will take the user to the other page and operate differently, option 4 will keep the user to remain on the current page and go through the calculation.

If the user chooses option 4, the user will have to input the required data such as:

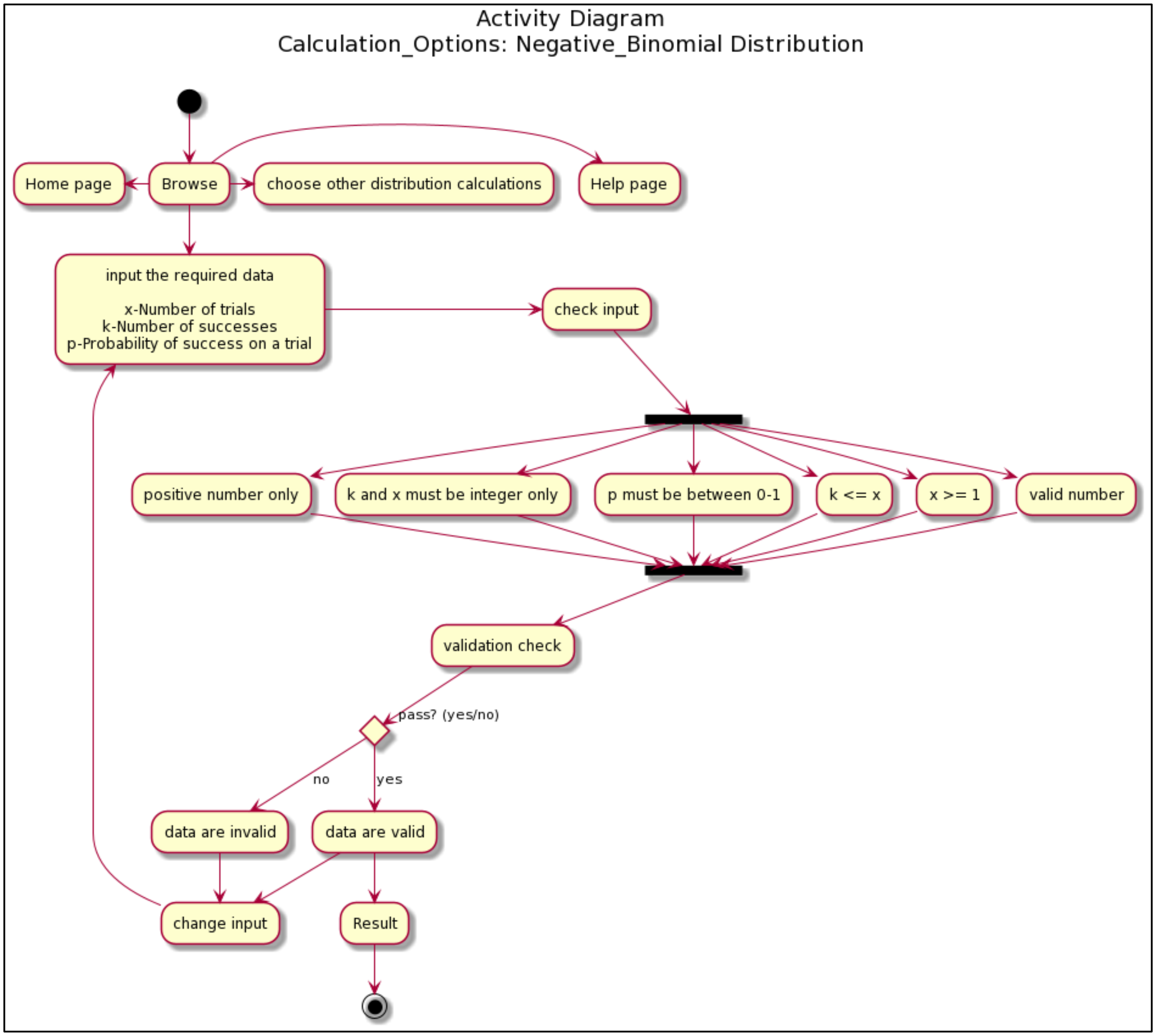
1. **x**-Number of trials
2. **k**-Number of successes
3. **p-**Probability of success on a trial

When the user is inputting the data, the data will go through the checking section **(check input).** In that checking section, multiple tests are running synchronically to ensure that all the data are valid and are ready to be computed.

For Negative Binomial distribution, those multiple tests are:

1. test to ensure that all inputs are in number form
2. test to ensure that all inputs are positive
3. test to ensure that k, and x must be integer number
4. p must be between 0 to 1
5. x is greater or equal to k
6. x is greater or equal to 1

If any of the tests have failed, then it will prevent the system from doing the computation until the user input the correct data. Once the data are valid, then the user can click on the calculate button to start computing, or change the data as needed.



2.5 Activity diagram for Poisson Distribution

In this activity diagram, we are assuming the user is in Poisson distribution section in the start state. The start and the end state indicate using the black-rounded circle.

Once the user is in the start state, the user is automatically considered as browsing the site **(Browse).** At this point, the user has 4 options listed as:

1. **(Home page)** - the user can go back to the home page
2. **(Choose other distribution calculations)**- the user can choose other calculations and jump directly to that page from the current calculation

3. **(Help page)**- the user can go to the help page to understand more about the distribution if they have any doubts
4. **(input the required data)** - the user can use the designed box specifically to calculate the probability of the distribution

While options 1 to 3 will take the user to the other page and operate differently, option 4 will keep the user to remain on the current page and go through the calculation.

If the user chooses option 4, the user will have to input the required data such as:

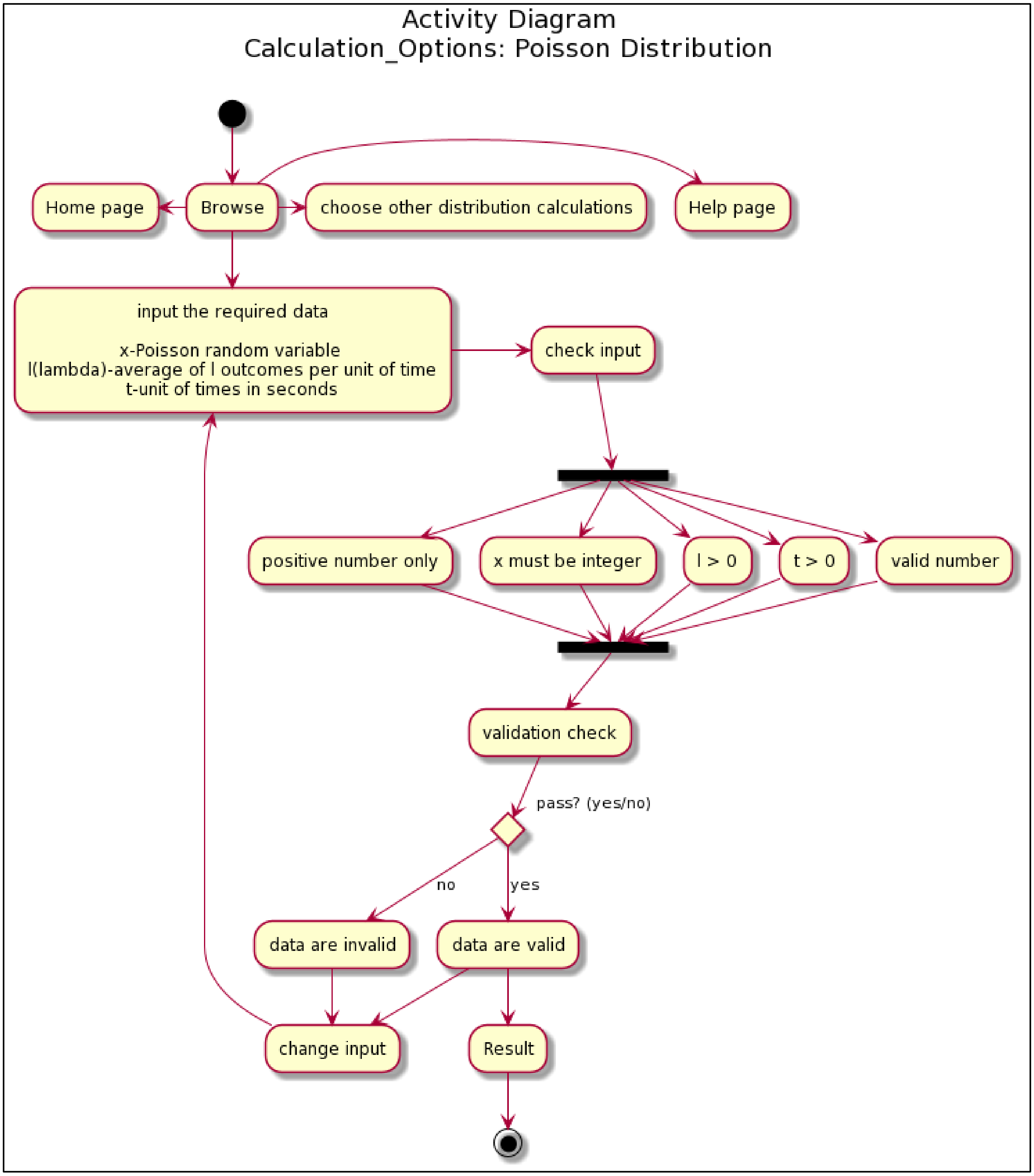
1. **x**-Poisson random variable
2. **l**-average of l outcomes per unit of time
3. **t**-unit time in second(s)

When the user is inputting the data, the data will go through the checking section **(check input).** In that checking section, multiple tests are running synchronically to ensure that all the data are valid and are ready to be computed.

For Poisson distribution, those multiple tests are:

1. test to ensure that all inputs are in number form
2. test to ensure that all inputs are positive
3. test to ensure that x must be integer number
4. l is greater than 0
5. t is greater than 0

If any of the tests have failed, then it will prevent the system from doing the computation until the user input the correct data. Once the data are valid, then the user can click on the calculate button to start computing, or change the data as needed.

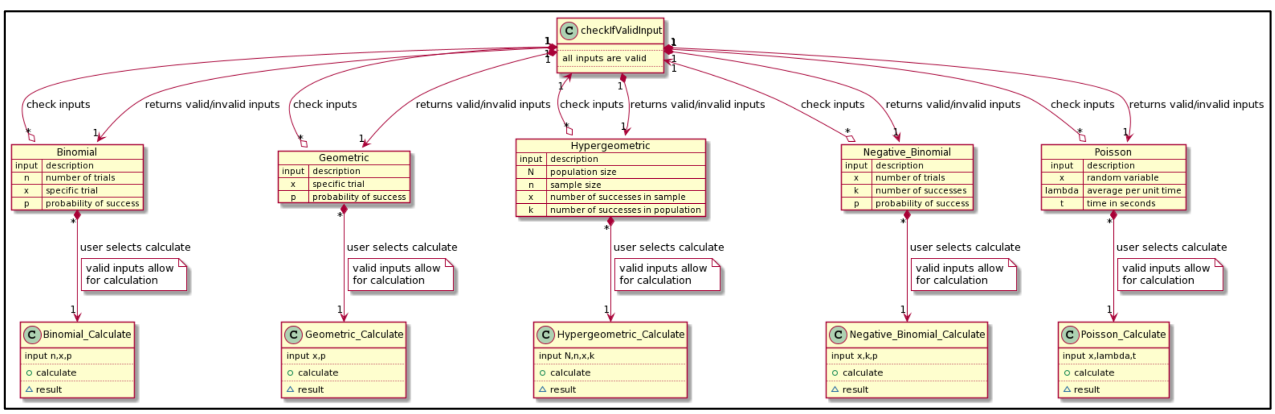

Activity Diagram
Calculation_Options: Poisson Distribution
Home page
Browse
choose other distribution calculations
Help page
input the required data
x-Poisson random variable
l(lambda)-average of l outcomes per unit of time
t-unit of times in seconds
check input
positive number only
x must be integer
l > 0
t > 0
valid number
validation check
pass? (yes/no)
no
yes
data are invalid
data are valid
change input
Result

3 Class Diagrams

3.1 Class Overview

The Discrete Probability Calculator v1.0 will have classes specific to each distribution differentiated by inputs and used for calculation (binomial class, geometric class, hypergeometric class, negative binomial class and poisson class). We also use a class for validating the inputs. In total the DPC uses 6 classes.

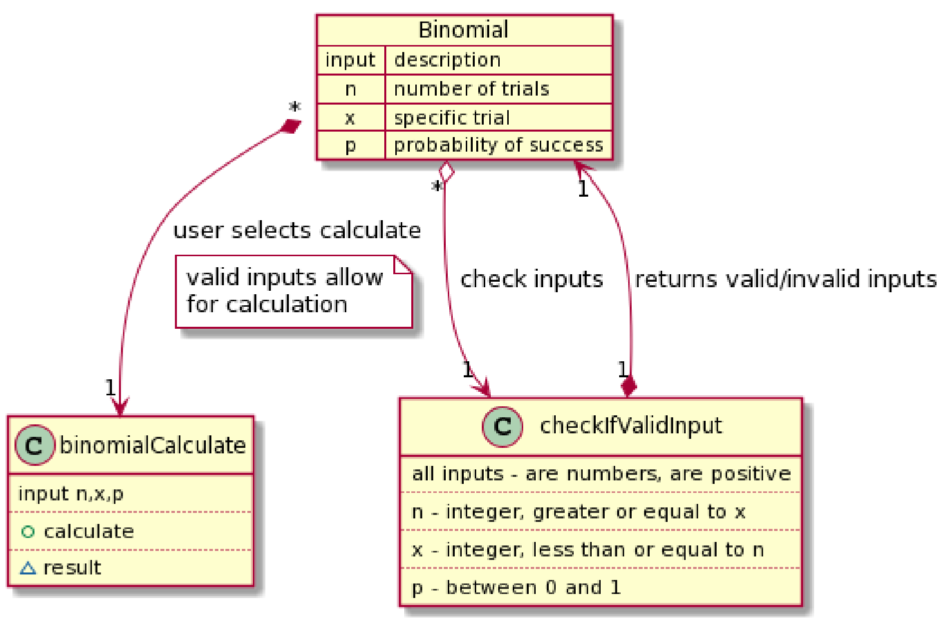
Class	Description
checkIfValidInput	This class validates the user input, it makes sure the input is valid for calculating the specific distribution.
binomialCalculate	Takes the valid inputs for n,x,p, calculates using the binomial distribution and returns the probability
geometricCalculate	Takes the valid inputs for x,p, calculates using the geometric distribution and returns the probability
hypergeometricCalculate	Takes the valid inputs for N,n,x,k, calculates using the hypergeometric distribution and returns the probability
negativeBinomialCalculate	Takes the valid inputs for x,k,p, calculates using the negative binomial distribution and returns the probability
poissonCalculate	Takes the valid inputs for x,lambda,t, calculates using the poisson distribution and returns the probability



3.2 Binomial Specific Diagram

The diagram below is a breakdown of the interaction between webpage and class function calls for the Binomial distribution. Aggregation is used between the Binomial inputs and the checkIfValidInputs class due to the ability to use checkIfValidInputs class for every distribution type. The use of composition for between Binomial inputs and the binomialCalculate class is due to the class being dependent on a single set of Binomial inputs specific to the Binomial distribution. The multiplicity is based on there being one set of validation rules for all (or any set *) of inputs, and one set of calculations for all (or any set *) of inputs.

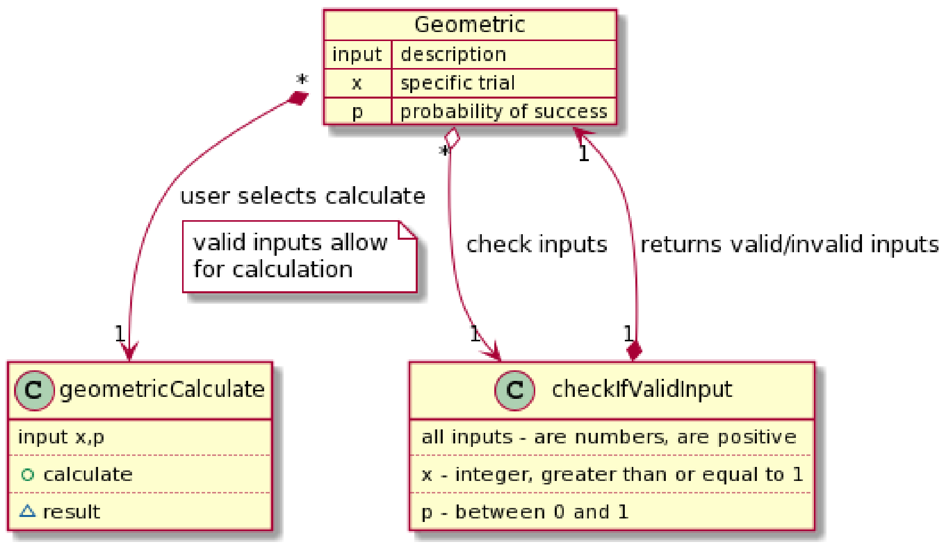
<table>
<tr><th>Class</th><th>Description</th></tr>
<tr><td>checkIfValidInput</td><td><table>
<tr><th>Input</th><th>Description of Validity Test</th></tr>
<tr><td>All inputs</td><td>Are numbers, are positive</td></tr>
<tr><td>n</td><td>Is integer, greater or equal to x</td></tr>
<tr><td>x</td><td>Is integer, less than or equal to n</td></tr>
<tr><td>p</td><td>Between 0 and 1</td></tr>
</table></td></tr>
<tr><td>binomialCalculate</td><td>Calculates the probability based on inputs and returns the result.</td></tr>
</table>



3.3 Geometric Specific Diagram

The diagram below is a breakdown of the interaction between webpage and class function calls for the Geometric distribution. Aggregation is used between the Geometric inputs and the checkIfValidInputs class due to the ability to use checkIfValidInputs class for every distribution type. The use of composition for between Geometric inputs and the geometricCalculate class is due to the class being dependent on a single set of Geometric inputs specific to the Geometric distribution. The multiplicity is based on there being one set of validation rules for all (or any set *) of inputs, and one set of calculations for all (or any set *) of inputs.

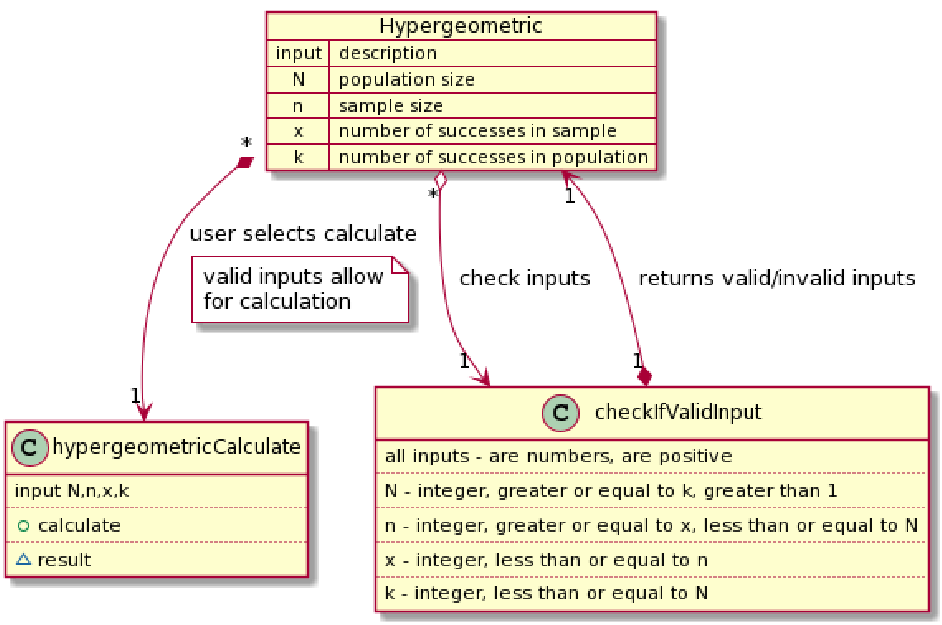
<table>
<tr><th>Class</th><th>Description</th></tr>
<tr><td>checkIfValidInput</td><td><table><tr><th>Input</th><th>Description of Validity Test</th></tr><tr><td>All inputs</td><td>Are numbers, are positive</td></tr><tr><td>x</td><td>Is integer, greater or equal to 1</td></tr><tr><td>p</td><td>Between 0 and 1</td></tr></table></td></tr>
<tr><td>geometricCalculate</td><td>Calculates the probability based on inputs and returns the result.</td></tr>
</table>



3.4 Hypergeometric Specific Diagram

The diagram below is a breakdown of the interaction between webpage and class function calls for the Hypergeometric distribution. Aggregation is used between the Hypergeometric inputs and the checkIfValidInputs class due to the ability to use checkIfValidInputs class for every distribution type. The use of composition for between Hypergeometric inputs and the hypergeometricCalculate class is due to the class being dependent on a single set of Hypergeometric inputs specific to the Hypergeometric distribution. The multiplicity is based on there being one set of validation rules for all (or any set *) of inputs, and one set of calculations for all (or any set *) of inputs.

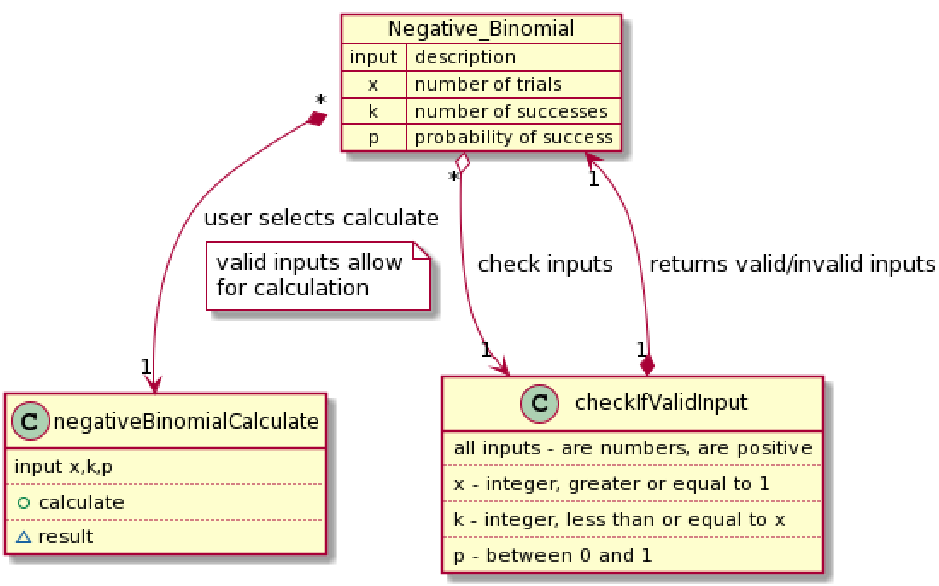
<table>
<tr><th>Class</th><th>Description</th></tr>
<tr><td>checkIfValidInput</td><td>
<table>
<tr><th>Input</th><th>Description of Validity Test</th></tr>
<tr><td>All inputs</td><td>Are numbers, are positive</td></tr>
<tr><td>N</td><td>Is integer, greater or equal to k, greater than 1</td></tr>
<tr><td>n</td><td>Is integer, greater or equal to x, less than or equal to N</td></tr>
<tr><td>x</td><td>Is integer, less than or equal to n</td></tr>
<tr><td>k</td><td>Is integer, less than or equal to N</td></tr>
</table>
</td></tr>
<tr><td>hypergeometricCalculate</td><td>Calculates the probability based on inputs and returns the result.</td></tr>
</table>



3.5 Negative Binomial Specific Diagram

The diagram below is a breakdown of the interaction between webpage and class function calls for the Hypergeometric distribution. Aggregation is used between the Negative Binomial inputs and the checkIfValidInputs class due to the ability to use checkIfValidInputs class for every distribution type. The use of composition for between Negative Binomial inputs and the negativeBinomialCalculate class is due to the class being dependent on a single set of Negative Binomial inputs specific to the Negative Binomial distribution. The multiplicity is based on there being one set of validation rules for all (or any set *) of inputs, and one set of calculations for all (or any set *) of inputs.

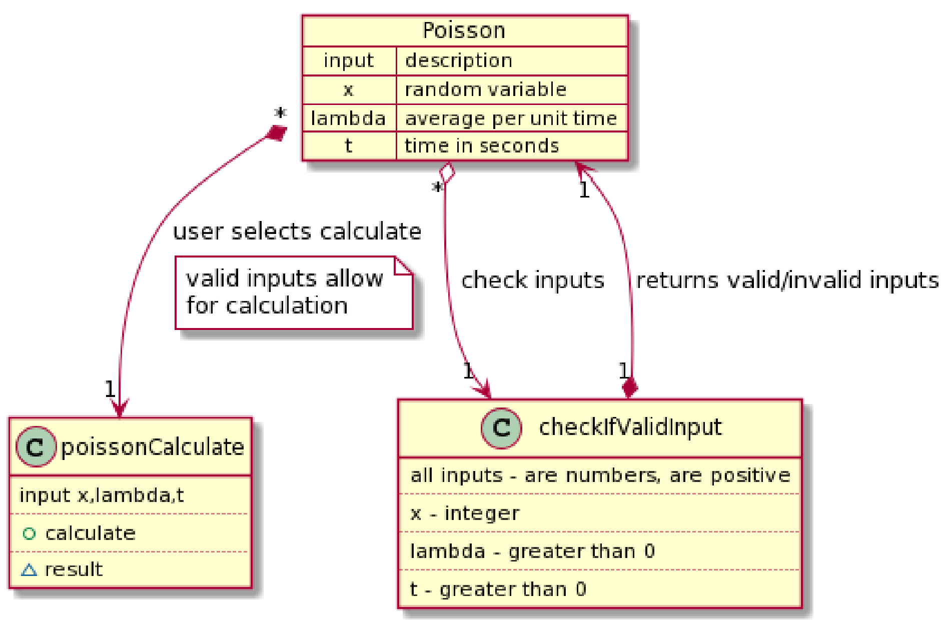
Class	**Description**	
checkIfValidInput	*Input*	*Description of Validity Test*
	All inputs	Are numbers, are positive
	x	Is integer, greater or equal to 1
	k	Is integer, less than or equal to x
	p	Between 0 and 1
negativeBinomialCalculate	Calculates the probability based on inputs and returns the result.	



3.6 Poisson Specific Diagram

The diagram below is a breakdown of the interaction between webpage and class function calls for the Hypergeometric distribution. Aggregation is used between the Poisson inputs and the checkIfValidInputs class due to the ability to use checkIfValidInputs class for every distribution type. The use of composition for between Poisson inputs and the poissonCalculate class is due to the class being dependent on a single set of Poisson inputs specific to the Poisson distribution. The multiplicity is based on there being one set of validation rules for all (or any set *) of inputs, and one set of calculations for all (or any set *) of inputs.

<table>
<tr><th>Class</th><th>Description</th></tr>
<tr><td>checkIfValidInput</td><td><table><tr><th>Input</th><th>Description of Validity Test</th></tr><tr><td>All inputs</td><td>Are numbers, are positive</td></tr><tr><td>x</td><td>Is integer</td></tr><tr><td>lambda</td><td>Greater than 0</td></tr><tr><td>t</td><td>Greater than 0</td></tr></table></td></tr>
<tr><td>poissonCalculate</td><td>Calculates the probability based on inputs and returns the result.</td></tr>
</table>

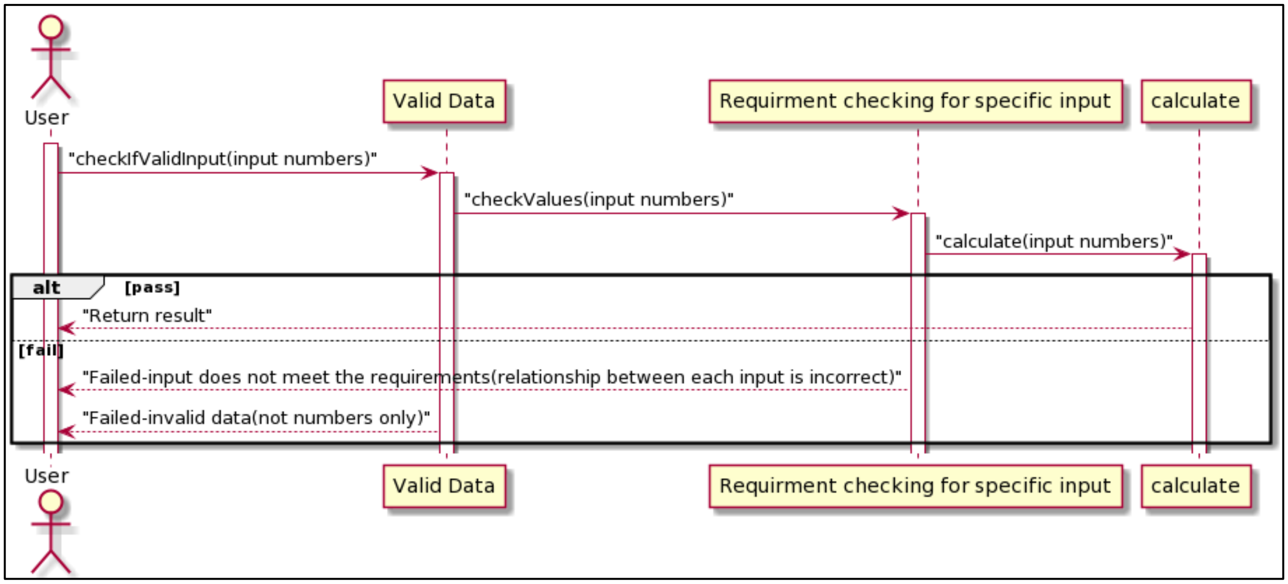


4 Behavioral Diagram(s)

4.1 Sequence Diagram

Both sequence diagram and state diagram apply to our project.
The following is a **sequence diagram.**

Description: First the program will ask the user for some inputs corresponding the distribution they want to calculate, the program then read in each input and check if they are numbers only, if not it will return false and prevent the user from clicking on the "result" button; if they are numbers, the program then check if the numbers are within the correct range and if their relationship is correct, note that for discrete distribution to work, the inputs must be within the range and the relationship must be correct, that's why we have different requirements for different distributions. If the numbers follow the criteria, the program will calculate it and return the result.



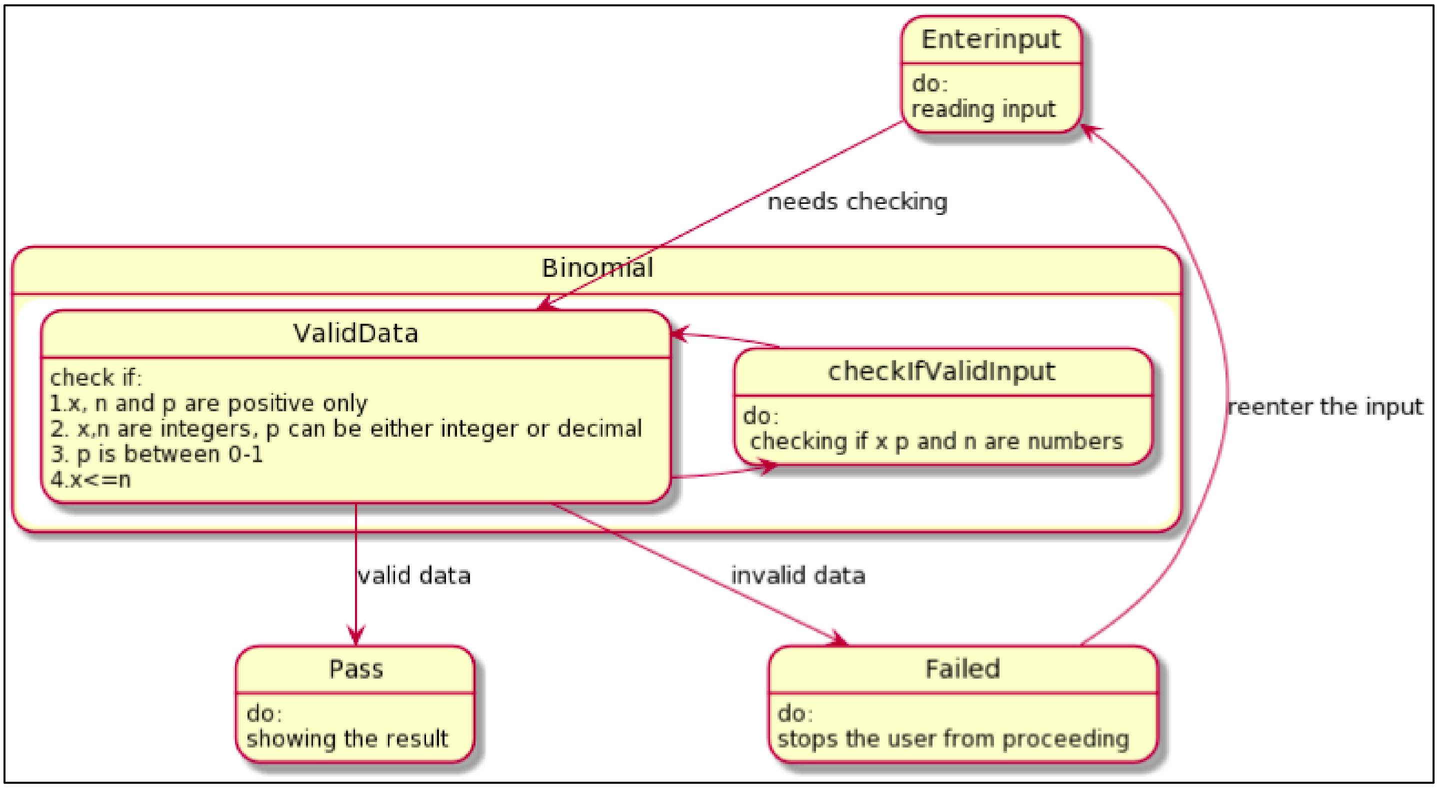
4.2 State Diagram for Binomial Distribution

This is a state diagram for Binomial distribution, our project calculates under 5 different discrete distribution, so **there will be a state diagram for each**.

Description: This state diagram is describing binomial distribution, after the user entered the input, it's in the needing checking state to validate if all the data are correct,
we then check if they are

1. numbers only
2. x, n and p are positive only
3. x, n are integers
4. p can be either integer or decimal
5. p is between 0-1
6. x<=n

We calculate the result only when all the input data is valid, otherwise it will ask the user to reenter the data if anything is incorrect.



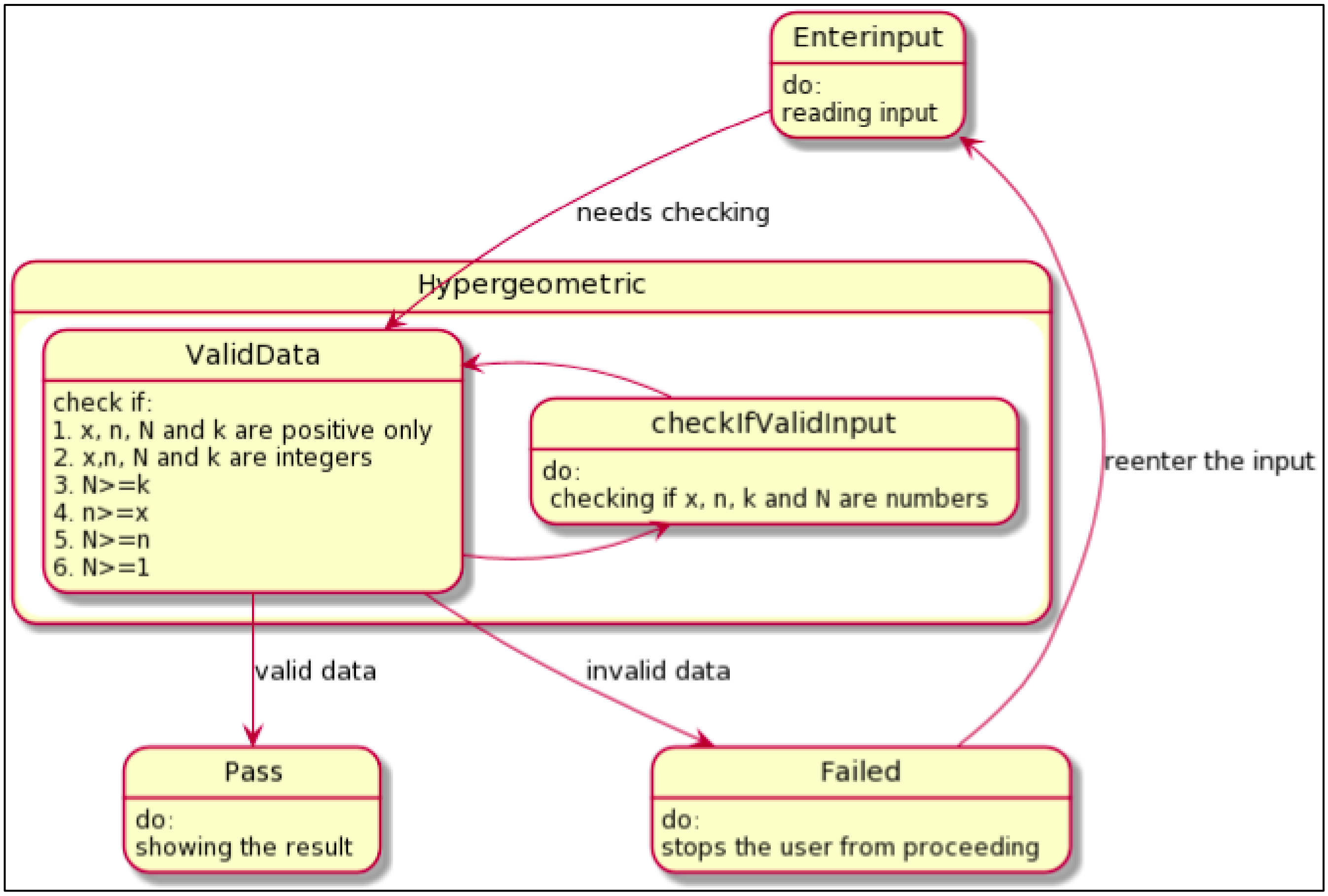
4.3 State Diagram for Hypergeometric Distribution

This is a state diagram for Hypergeometric Distribution.

Description: This state diagram is describing hypergeometric distribution, after the user entered the input, it's in the needing checking state to validate if all the data are correct,
we then check if they are

1. numbers only
2. x, n, N and k are positive only
3. x, n, N and k are integers
4. N>=k
5. n>=x
6. N>=n
7. N>=1

We calculate the result only when all the input data is valid, otherwise it will ask the user to reenter the data if anything is incorrect.



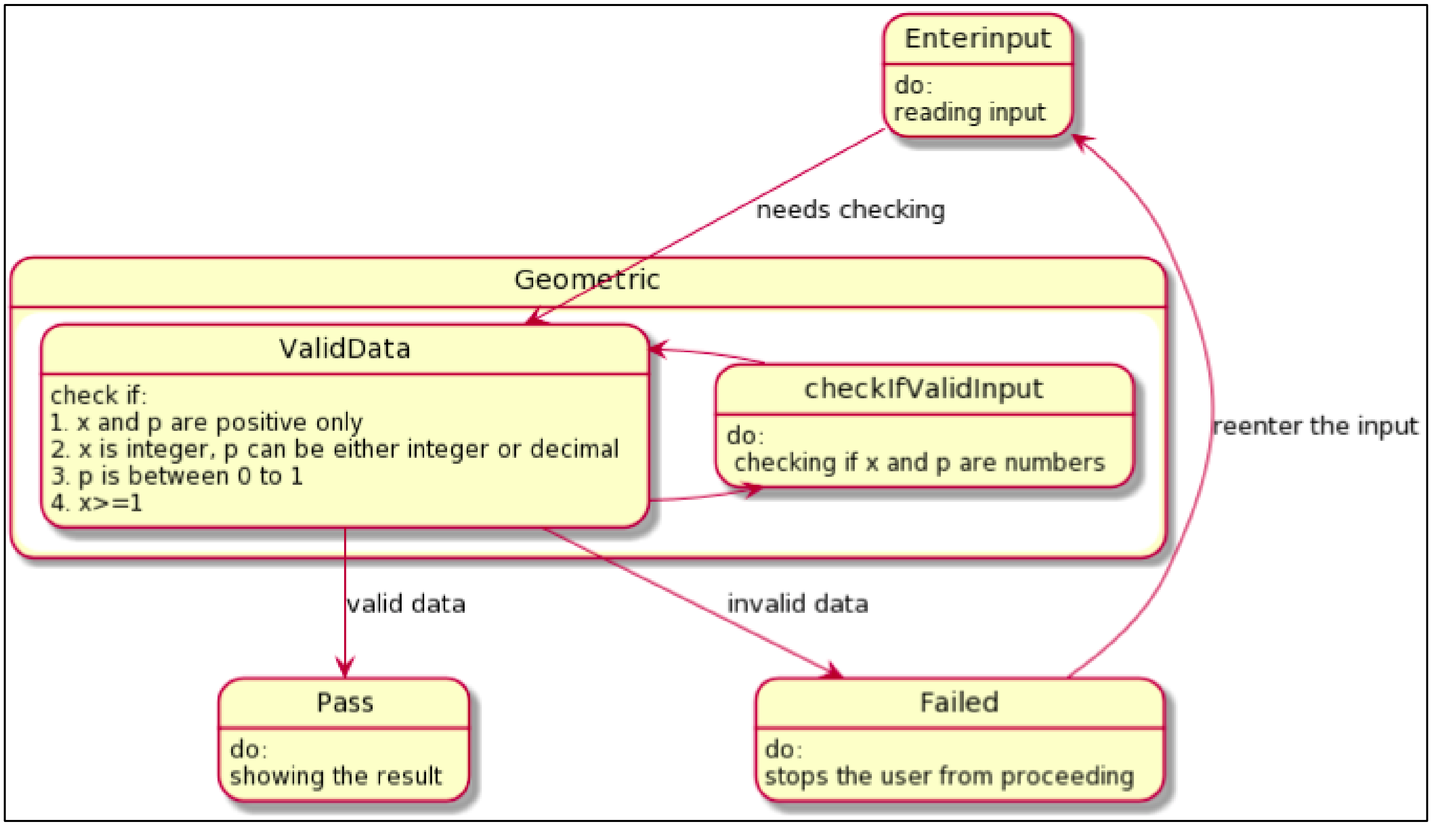
4.4 State Diagram for Geometric Distribution

This is a state diagram for Geometric Distribution.

Description: This state diagram is describing geometric distribution, after the user entered the input, it's in the needing checking state to validate if all the data are correct,
we then check if they are

1. numbers only
2. x and p are positive only
3. x is integer
4. p can be either integer or decimal
5. p is between 0-1
6. x>=1

We calculate the result only when all the input data is valid, otherwise it will ask the user to reenter the data if anything is incorrect.



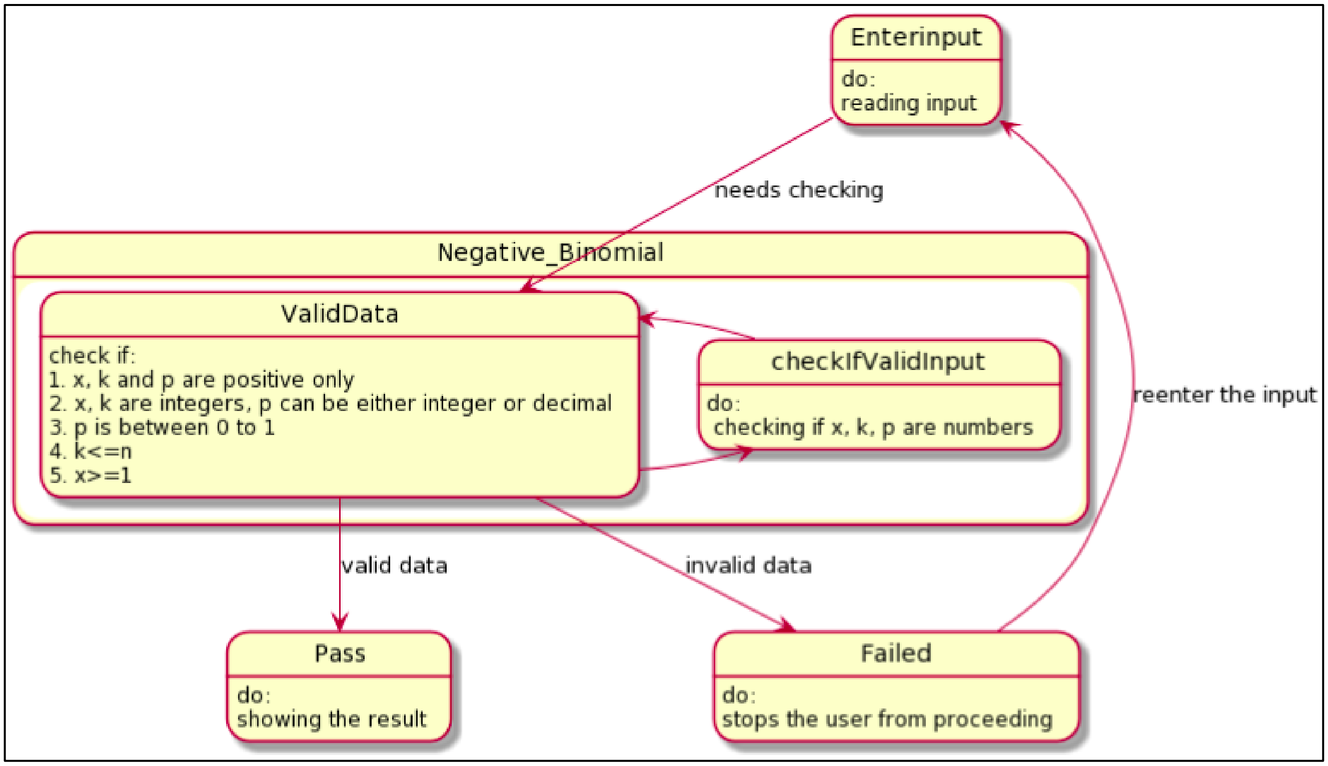
4.5 State Diagram for Geometric Distribution

This is a state diagram for Negative Binomial Distribution.

Description: This state diagram is describing negative binomial distribution, after the user entered the input, it's in the needing checking state to validate if all the data are correct,
we then check if they are

1. numbers only
2. x, k and p are positive only
3. x, k are integers, p can be either integer or decimal
4. p is between 0 to 1
5. k<=n
6. x>=1

We calculate the result only when all the input data is valid, otherwise it will ask the user to reenter the data if anything is incorrect.



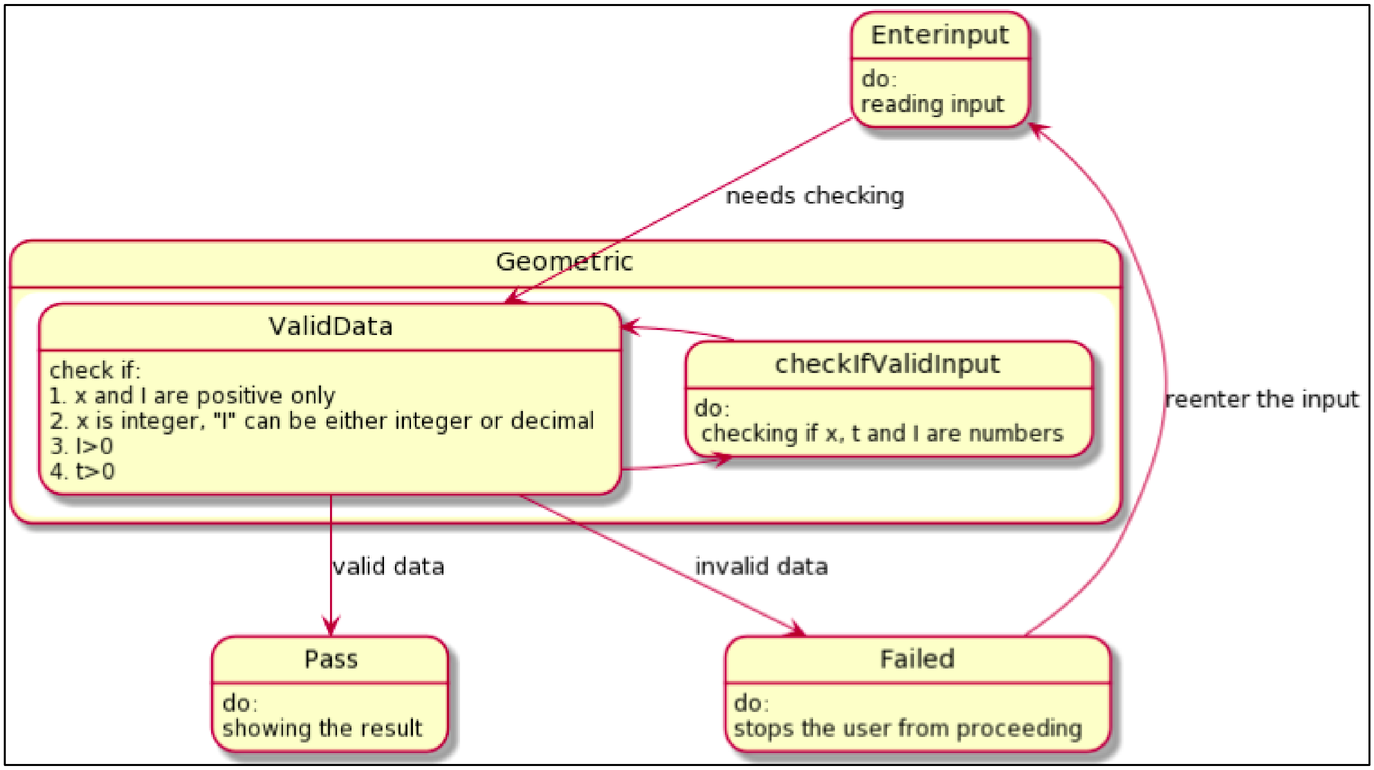
4.6 State Diagram for Poisson Distribution

This is a state diagram for Poisson Distribution.

Description: This state diagram is describing poison distribution, after the user entered the input, it's in the needing checking state to validate if all the data are correct,
we then check if they are

1. numbers only
2. x and l are positive only
3. x is integer, "l" can be either integer or decimal
4. l>0
5. t>0

We calculate the result only when all the input data is valid, otherwise it will ask the user to reenter the data if anything is incorrect.



EnterInput
do:
reading input
needs checking
Geometric
ValidData
check if:
1. x and l are positive only
2. x is integer, "l" can be either integer or decimal
3. l>0
4. t>0
checkIfValidInput
do:
checking if x, t and l are numbers
reenter the input
valid data
invalid data
Pass
do:
showing the result
Failed
do:
stops the user from proceeding

Appendix A - Group Log

CS320 – Meeting Notes

These are the Meeting Minutes for Project Milestone 2 ONLY – Previous meeting minutes are found in the Group Log Appendix in Project Milestone 1
11/12/2020 –Thursday, zoom meeting • Milestone 2 review • Looked over Anna's uml diagrams for 2.1 Activity diagrams, went over the validity checks for each distribution • Talked about 2.2 structural diagram and 2.3 behavioral modeling • We looked at the examples from lecture 12 sys modeling part2 • We divided the work: Section 2 - Anna, Section 3 - Patrick, Section 4 – Z • Anna showed us the code she's been working on, the binomial distribution and the valid checking functionality.
11/16/2020 –Monday, zoom meeting • We went over Milestone 2. • We all went over our diagrams. Anna and Z have completed their diagrams and descriptions. • Patrick has mostly finished diagrams, but hasn't finished descriptions. • We got Z access to the shared doc. • We went over Anna's validation code to get a better idea of how the diagrams should be structured. • We will meet at our regularly scheduled Wednesday meeting, at 2 pm. • This was a Zoom call from 6:30pm to 7:15 pm
11/18/2020 –Monday, zoom meeting Wednesday 2pm • We each talked about our diagrams and sections • Ran out of time to go over Anna’s code • Schedule a meeting for the next day to go over Anna’s code (6pm 11/19)
11/19/2020 - Meeting Minutes Zoom meeting 6pm - 7:15pm • Anna walked us through the html, css and JavaScript code she has worked on. • We divided up the remaining work: • Z will work on the geometric and hypergeometric distributions • Anna will work on the negative binomial distribution

• Patrick will work on the Poisson distribution and add descriptions and examples for each distribution. • We will not meet over Thanksgiving break, and will meet again on Wed. Dec. 2 at 2pm.