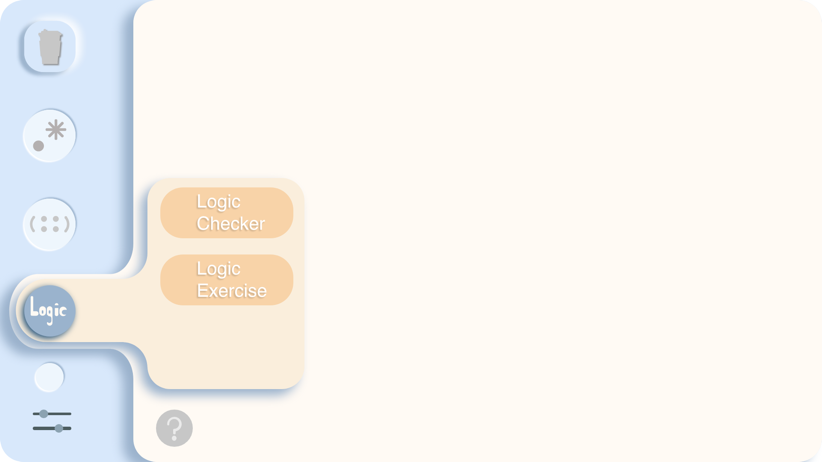
ENGF002 Scenario1

## Plan of implementing logic proposition learning tool.

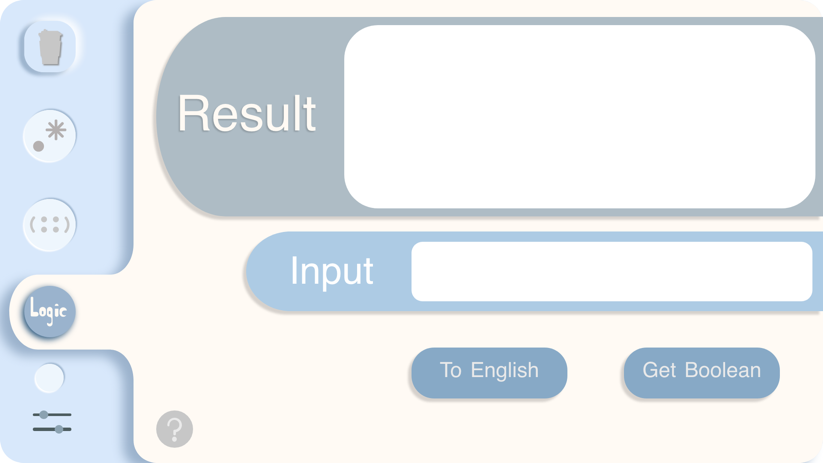
This learning tool can be used to convert logic proposition to English sentence which can deepen learner understanding of propositional logic. In addition, the software can also produce a final result of

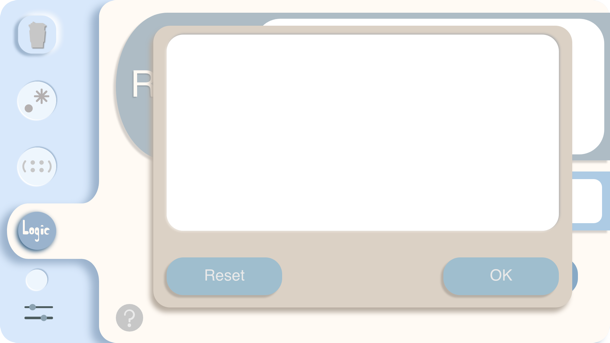
A formular (True or False). The functions above contain in the page logic proposition generation. In general, there are three parts available for user to interact with, which is logic proposition generation, logic proposition checker (provide some formular to user and let them fill in the truth table the computer will check and correct), additional information page (some basic equivalent equation or property about logic proposition,)

(the general layout of logic part)



## For the Logic propositional generation:





There are two things can be generated:

1: learner given a proposition logic formula in form (Such as A-> (B ^C), B<-->A, e.t) and then user need to choose a kind of interpretation way( by pressing “to English“ or ”get Boolean” button. Next the software will pop a window to let learner input the meaning of each letter (For instance: what is the meaning of A formular? will be cast, he likes eating apple is a proposition may be input and bind with A, when each letter is matched, whole formular will be translated into English Sentence by software.)

Firstly each letter may be substitute by the learner specified meaning , so that each declarative sentence can be stored. And follow the precedence of proposition logic to interpret which is ()> - >and, or>implication, bi-implication.

The input whole formula just in String form to receive user input But then it will be stored in Java Array list by transforming to char array and add it one by one. so that the connective key words in English form can be easily added. And each logic proposition character will just be considered as a single String in Arraylist(Such as “->” “->>-”)

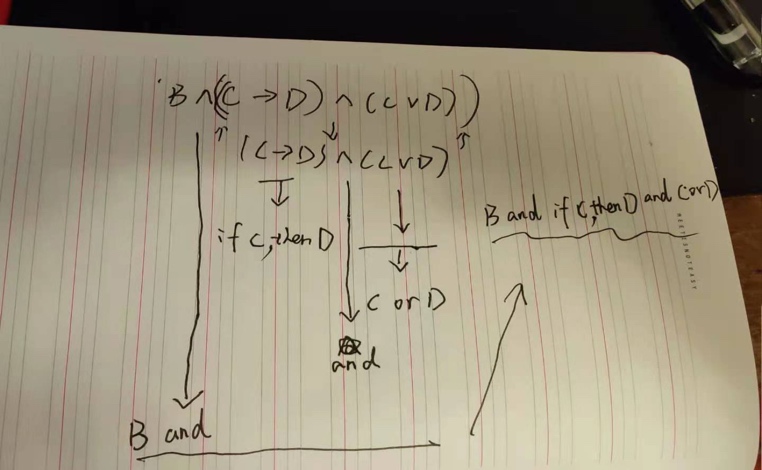
#### To analyze the structure of propositional logic my algorithm is:

First To dealing the proposition inside the parenthesis. I use stack to find a balanced closed parenthesis complex in the Arraylist(transformed from user input logic formular). Iterate the Arraylist, the first time “(“ is found (record the index) push into stack when a”)” is found pop a “(“in the stack. When the stack become empty, it means that I found a closed parenthesis complex also record the last “)” index in the parenthesis, remove the outer parenthesis as no parenthesis exist in an English sentence. Using recursion to go into the parenthesis keep stripping parenthesis based on the recorded interval and return when no more parenthesis in the interval and it will go to the remaining uninterpreted parenthesis complex in previous layer.

And the base interpret order is based on the precedence of logic character: just scan the given interval different interpret method will be given in different logic character (for example: for ^: just substitute it with “and”, And for the “-><-”,” only if” should replace “-><-”.

Finally, each logic symbol will be correct interpreted in its original meaning considering () and the precedence of symbol.

###### The main recursive interpret structure is:



2: learners can also calculate the result of a formular by inputting the value of each formular (Letter will be replaced by true or false here) And use the logic character in java programing language to substitute some of proposition logic character (& to ^,! to -). For the logic character like implication and bi-implication, rewrite it with (! And |), based on the amino:

1. (P←→Q) ＝ (P→Q) ∧(Q→R)

2.P→Q ＝ ┐P∨Q

## For the propositional logic checker:

This page is for learner do some exercise about propositional logic which involve randomly given a specidied size of proposisitonal logic(in the below question form),let the learner draw the truth tavle after finishing the answer in each colum of turth table just press the check button the feedback of your fillin answer will apper in result.

Push a button then the outputlogic() called in cotroller so that the model output a random formular.And to create a turth table the model will finally call a method called drawtable() in view class so that a truth table will be drawn automatically in needed row and colum.acorrding to the precedence of proposition logic to decide which formular will be compute first and be written into the row title of turth table.and use the caculator of a proposition fomular above to check if the correct answer is the same as what user input.

## The logic checker Layout:

A picture containing calendar

Description automatically generated

Every time the ahead formular result is caculated it will be rewritten to substitue the

Caculated formular to keep track of each individual formular correct boolean value to check untill the whole formular boolean is getten.

## 3 Helper page:

Text, letter

Description automatically generated