









Criterion B: Design

Criterion B: Solution overview

Libraries used are apache poi excel library and javaOctave library:

 commons-collections4-4.1	9/8/2017 3:09 PM	Executable Jar File	734 KB
 commons-logging-1.2	9/8/2017 3:08 PM	Executable Jar File	61 KB
 curvesapi-1.04	9/8/2017 3:09 PM	Executable Jar File	97 KB
 javaoctave-0.6.0	1/31/2018 9:02 PM	Executable Jar File	134 KB
 poi-3.17	9/8/2017 3:17 PM	Executable Jar File	2,638 KB
 poi-ooxml-3.17	9/8/2017 3:17 PM	Executable Jar File	1,445 KB
 poi-ooxml-schemas-3.17	9/8/2017 3:17 PM	Executable Jar File	5,786 KB
 xmlbeans-2.6.0	9/8/2017 3:09 PM	Executable Jar File	2,667 KB

The apache poi excel library can be downloaded at the website:

<https://poi.apache.org/download.html>

The javaOctave library can be downloaded at the website:

<https://sourceforge.net/projects/javaoctave/>

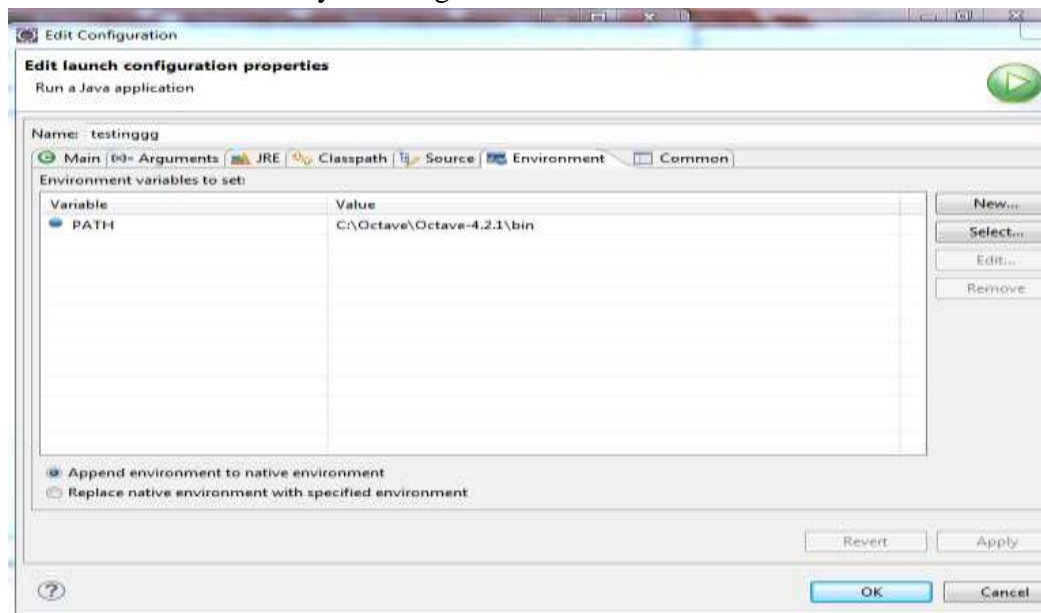
Some controlled variables:

- To figure out if the excel file format is supported by the apache poi library. This will require manual testing of many possible excel file formats. After some testing, it is evident that .xls and .xlt files can be read with HSSFWorkbook, while other files such as .xlsx, .xlsm, .xltx, .xltm and .xlam files can be read using XSSFWorkbook. Other older files format will not be read by the apache poi library.
- The excel file that the client uses need to be in a specific format so the program can read it correctly. We have decided that the excel format will look like:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1		ELL Sc	Sc 10	Sc 20	Sc 30	Sc 14	Sc 24	Bio 20	Bio 30	Chem 20	Chem 30	Phys 20	Phys 30	Math 10-C	Math 10-3	Math 20-1	Math 20-2	Math 20-3	Math 30-1	Math 30-2	Math 31
2	Student																				
3	names	their grades		grades can be left empty if they did not take the course																	
4	names																				
5	names																				

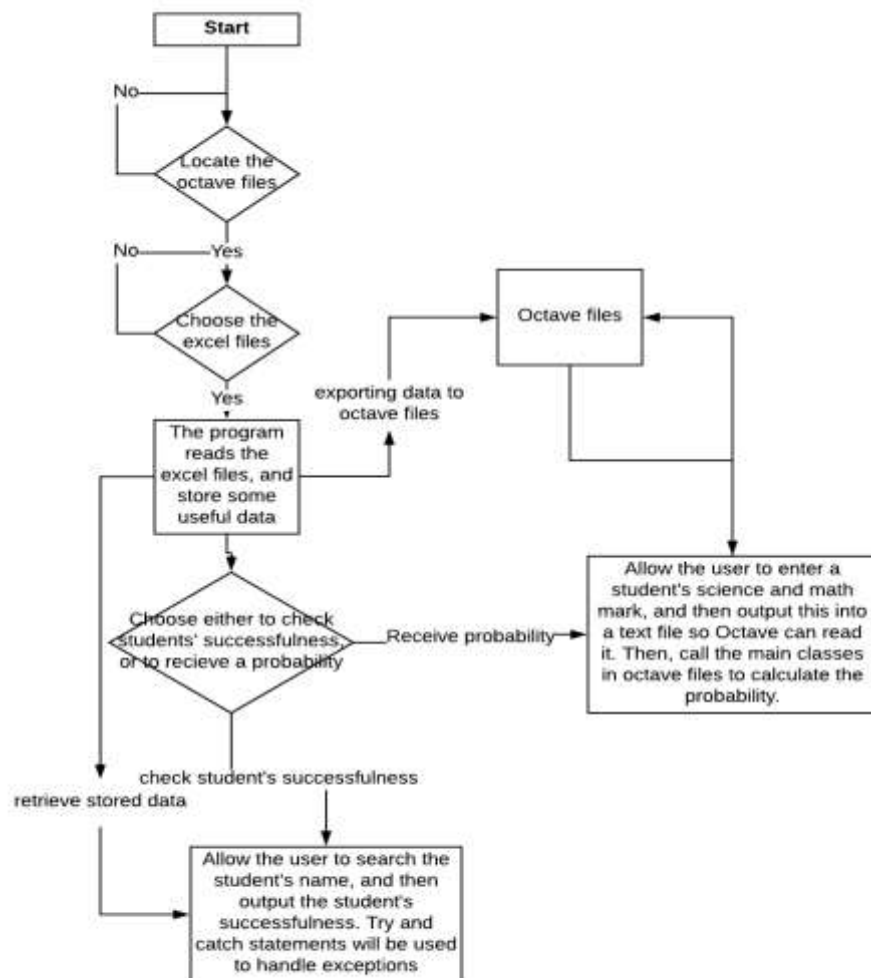
- For the octave files to run, they need to be in the same folder, which is similar analogy as all Java classes need to be in the same package. Therefore, the path of octave files needs to be configured at the beginning of the program.

- The Octave program folder name need to be the same as “C:\Octave\Octave-4.2.1” since I did not find a way to change the environmental variables in Java:

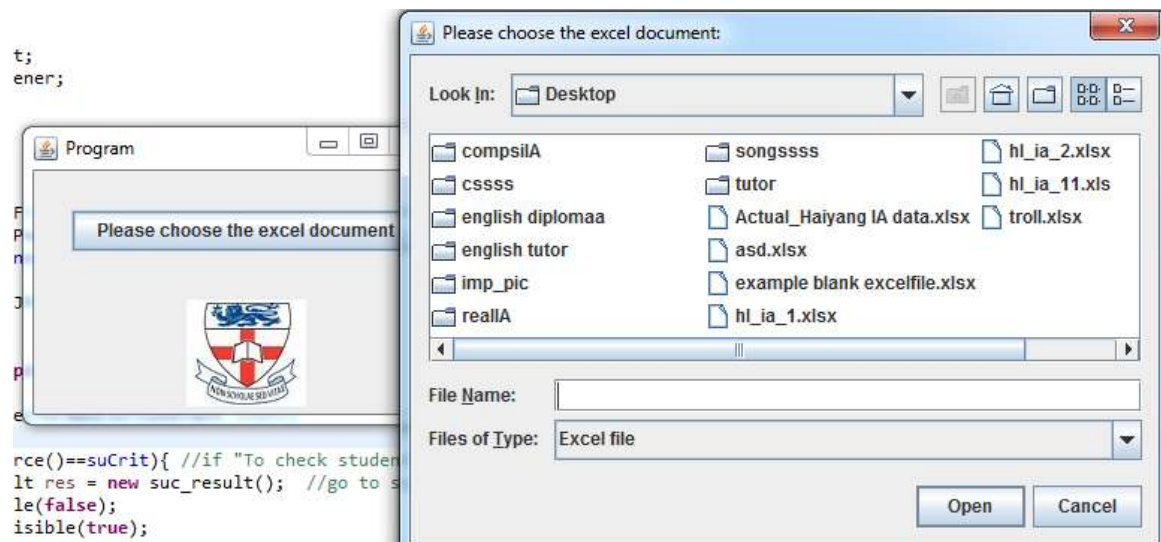
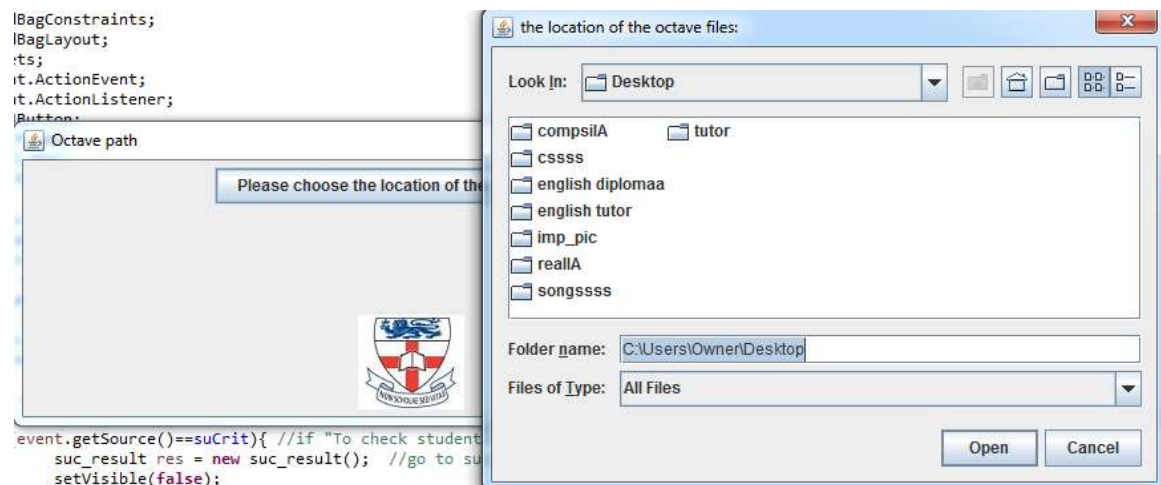
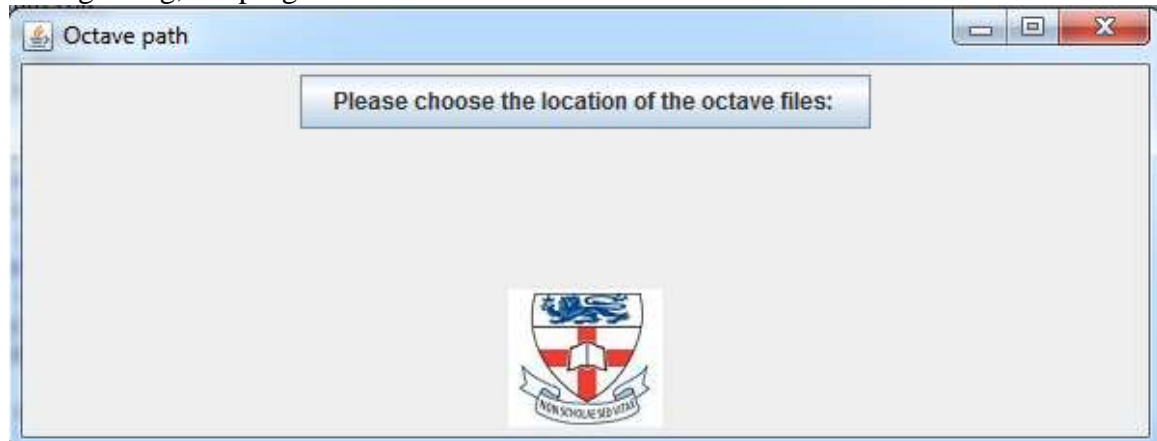


The designs of the solution:

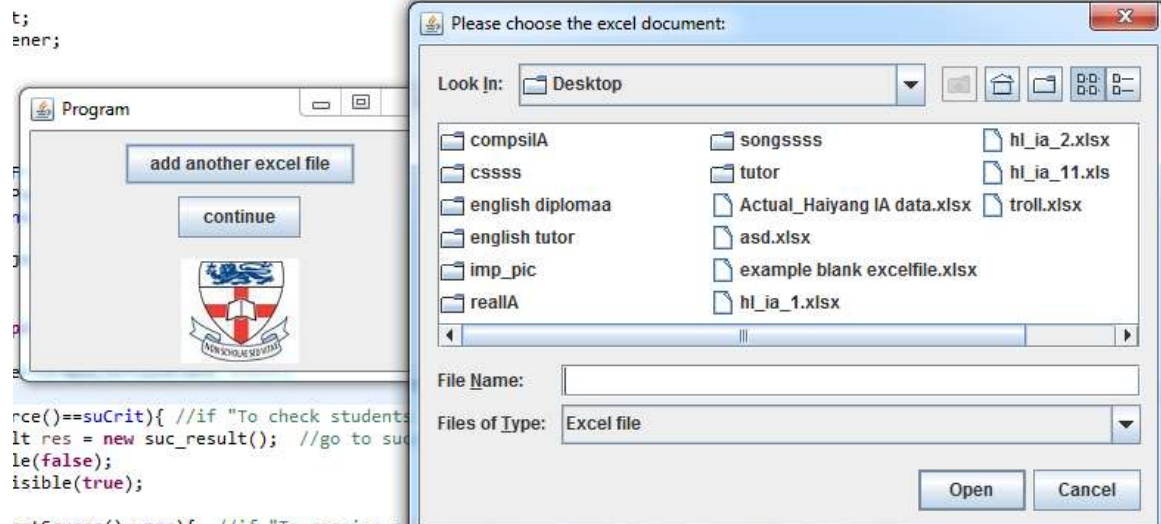
The basic overview of the program:



At the beginning, the program asks the user to choose the location of the octave files:



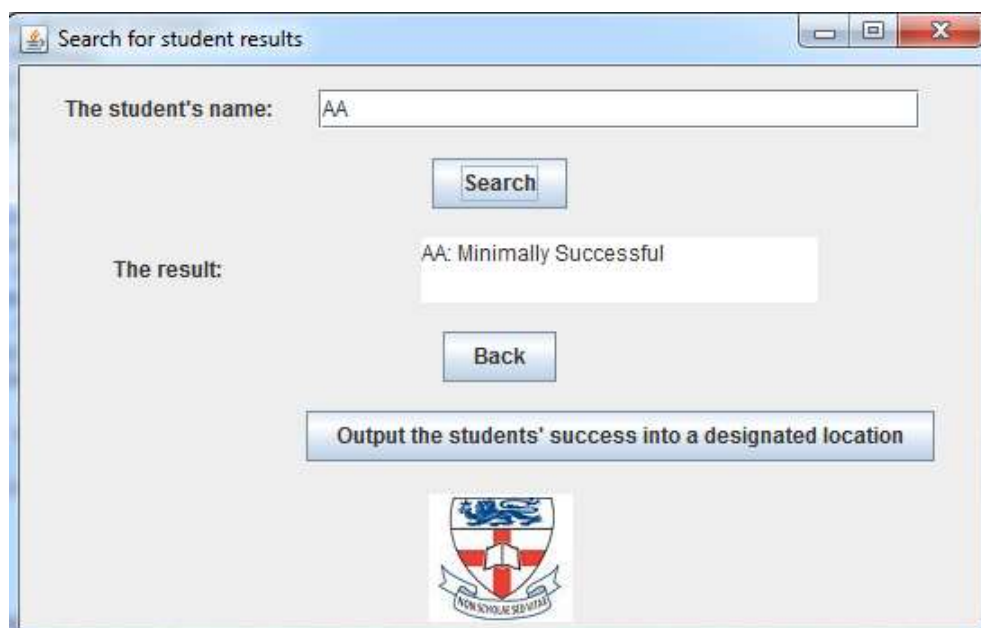
The user can choose multiple excel files:

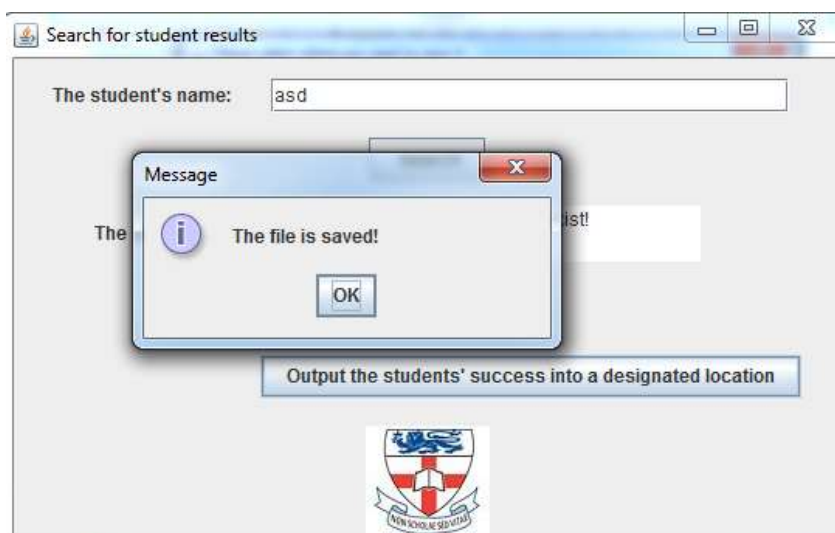
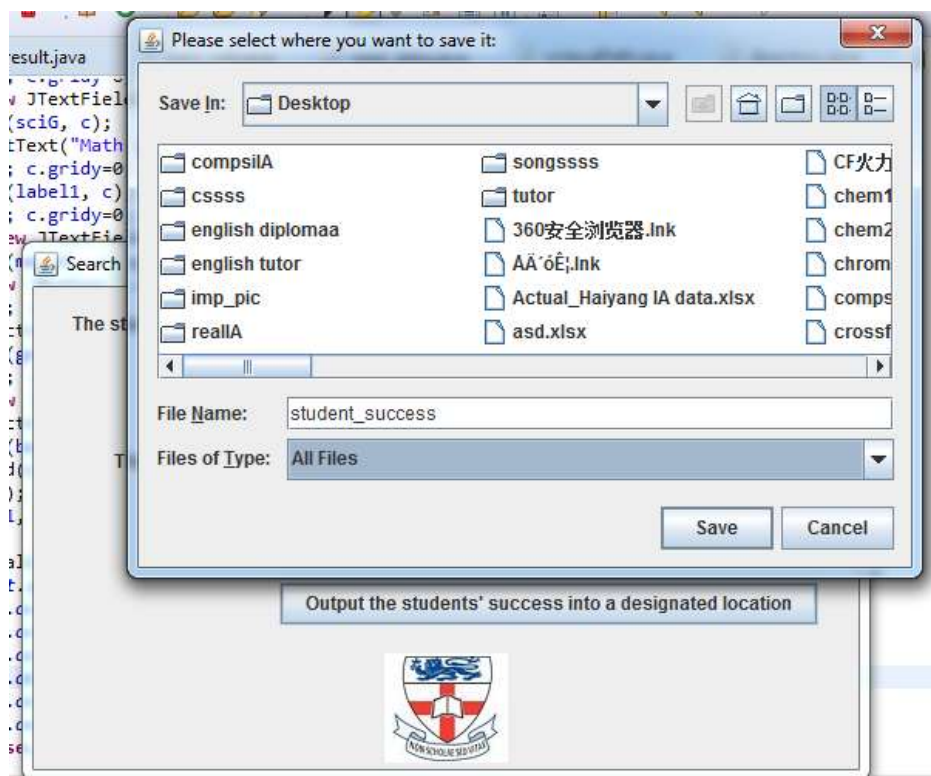
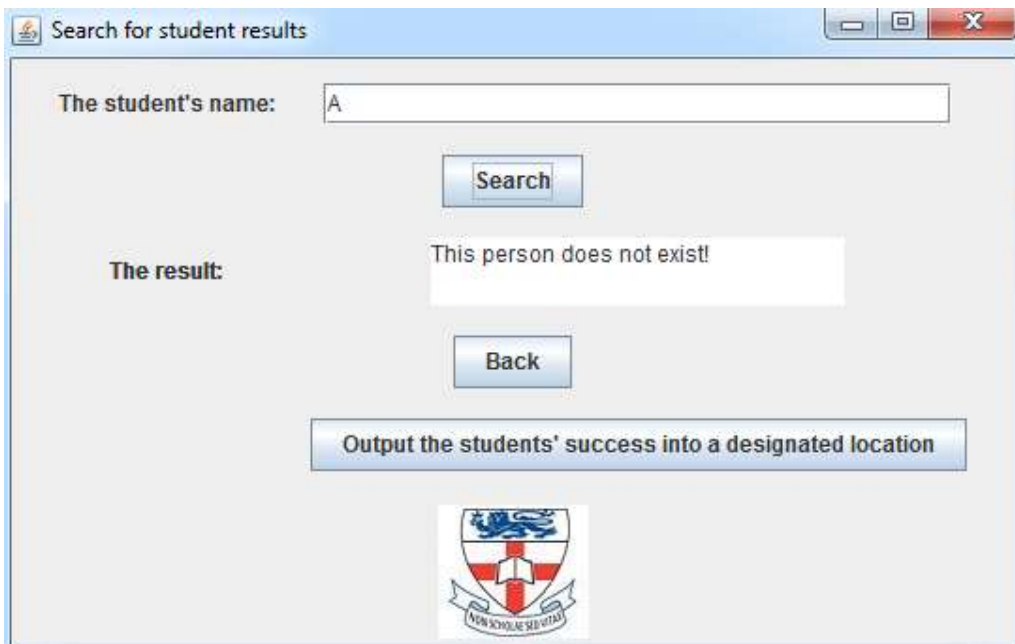


The user then needs to make either of two choices:



If the user wants to check students' successfulness:






```
student_success - Notepad
File Edit Format View Help
AA: Minimally Successful
AA1: Moderately Successful
CA: Moderately Successful
CD: Minimally Successful
CC: Moderately Successful
CP: Moderately Successful
FI: Minimally Successful
JW: Maximally Successful
KG: Minimally Successful
KJ: Moderately Successful
LY: Minimally Successful
LN: Moderately Successful
LL: Moderately Successful
```

If user wants to receive a recommendation:

Get recommendations for students


Science grade: Math grade:

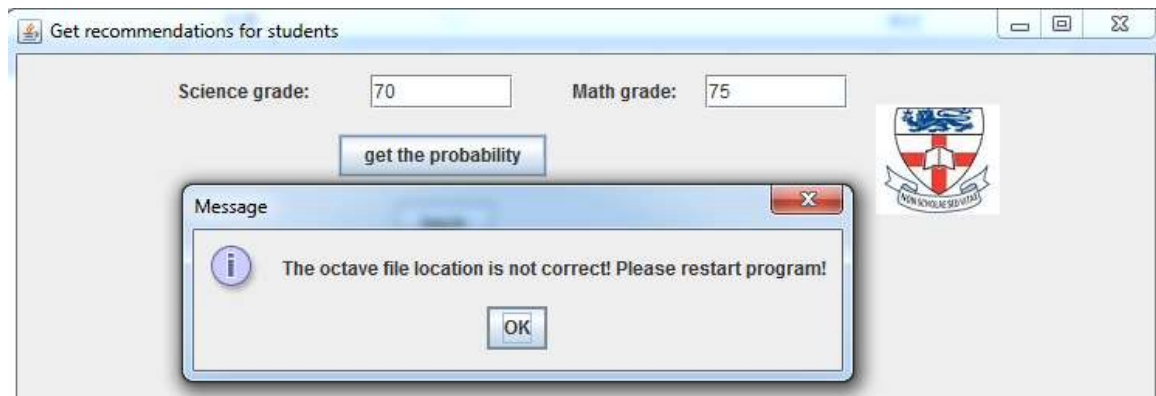
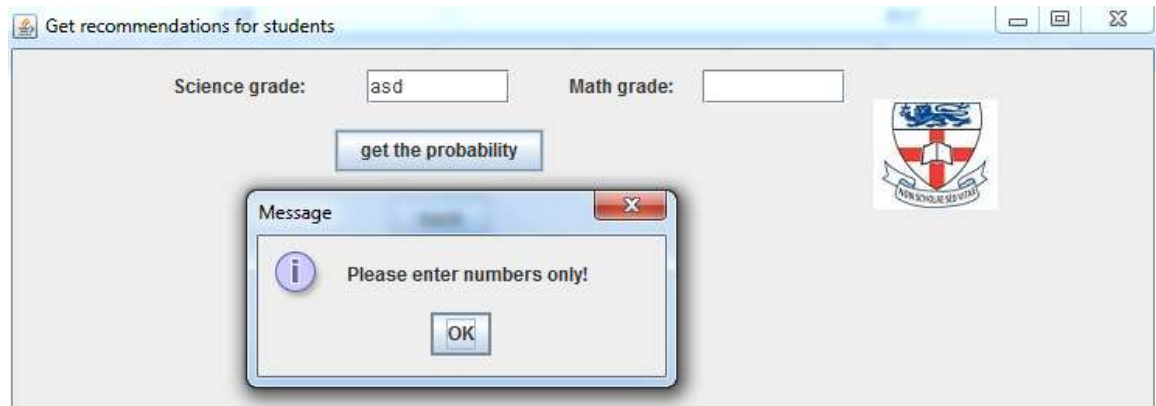
Probability of getting into Sc10:

The accuracy of this prediction:

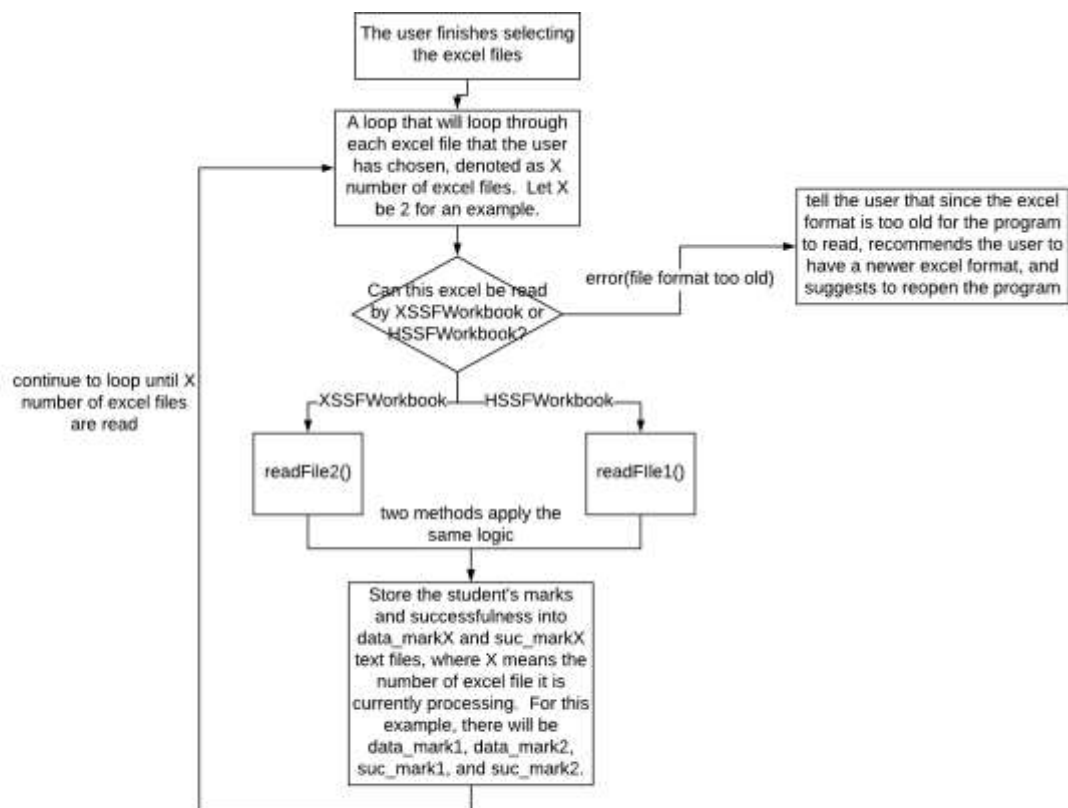
Probability of this student being moderately successful or above:

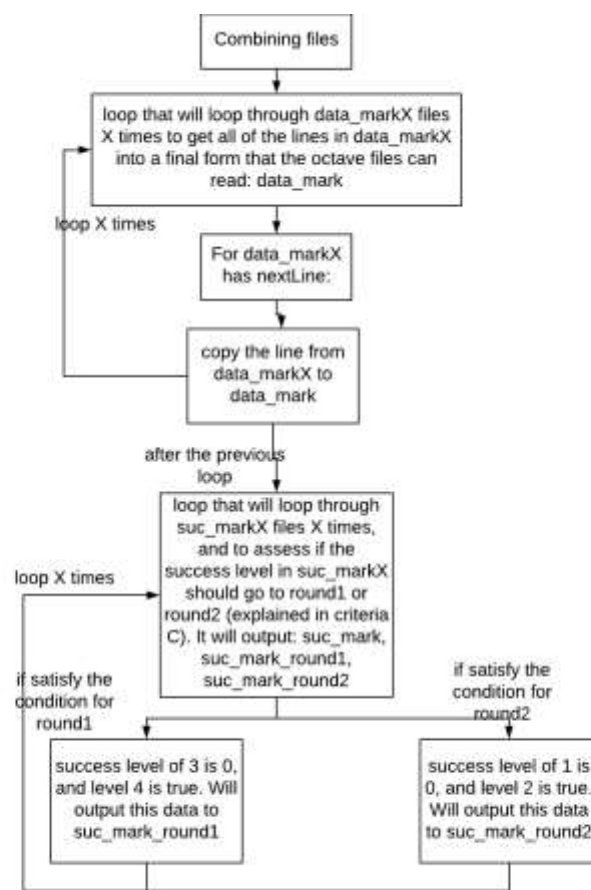
Probability of this student being maximally successful:





The process of reading and merging data:

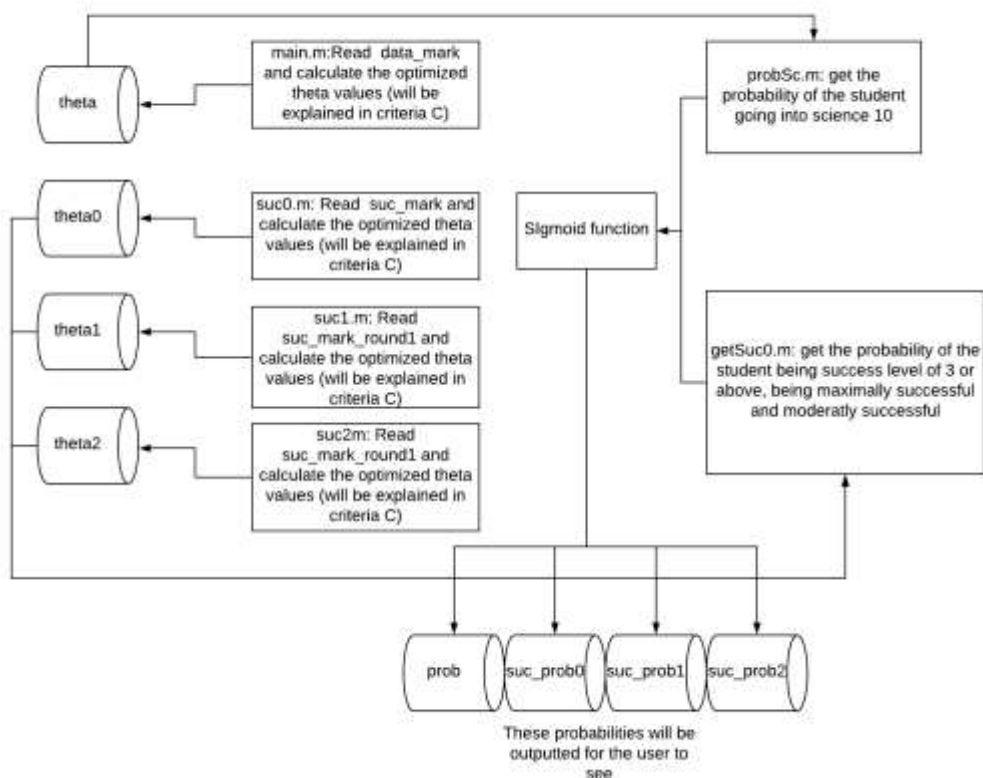


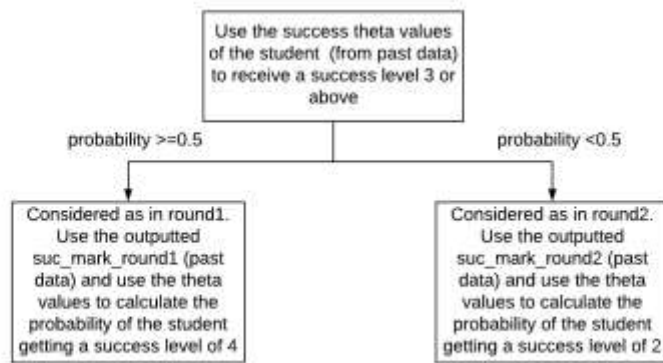


Success levels:

1. Not successful. (These students did not complete a 20-level science course.)
2. Minimally successful. (These students passed a 20-level science course to fulfill the minimum requirement for a high school diploma.)
3. Moderately successful. (These students passed a 30-level course with a mark below 75%.)
4. Maximally successful. (These students passed a 30-level course with a mark above 75%.)

Round1 and round2 will be clearer after showing the basic overview of the functioning of the machine learning algorithm and its connection with Java:





A more in-depth overview of the functionality of all Java classes:

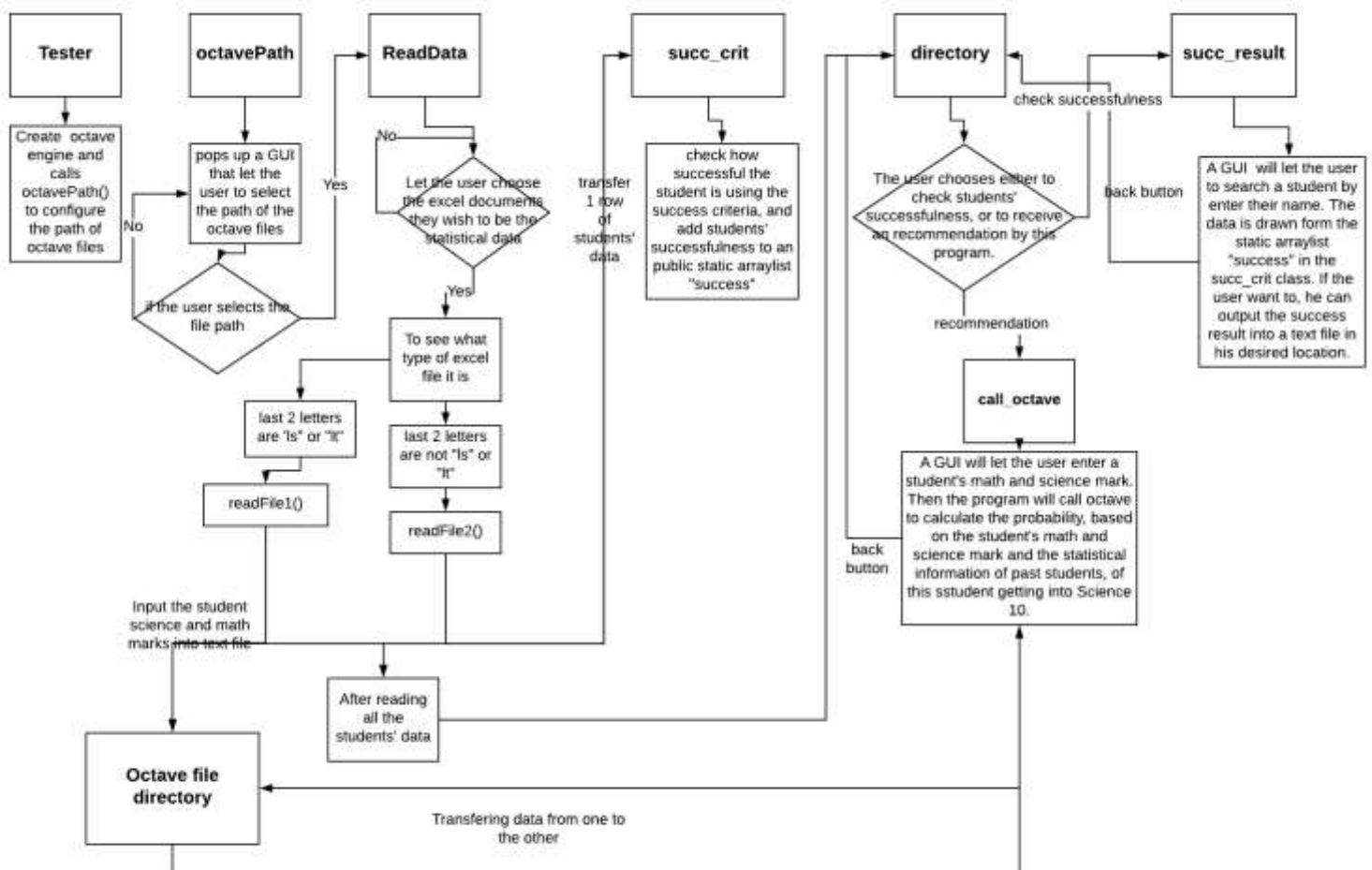
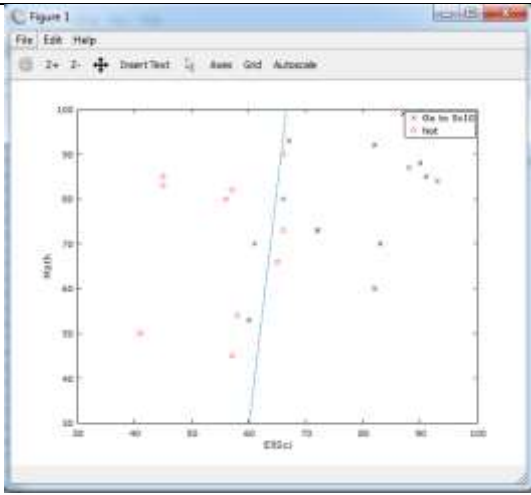


Table 6. Testing type, testing method, and examples.

Testing type	Example
The students' data in excel should be read correctly.	If one student's math mark is 50 and science mark is 70, the correct data printed out in the array should be student_name, 50, 70.
Have error handling for special cases.	If the octave file path is wrong, when Java calls octave there will be an error. Thus make a try and catch statement that

	will catch the OctaveEvalException error
The user can choose to location open and store files.	Outputting the file to a selected location and open it. Debug if it does not output to the correct location.
Students' successfulness is correctly considered	Make an answer key first manually, then compare the program calculated successfulness with the answer key
The client can search the successfulness of past students by searching their names.	Test if the student's result is correctly shown after searching.
Successfully use machine learning algorithms to produce a predictive model based on each student's grades.	 <p>This type of plot will be used to see if the prediction is correct or not. The line in the plot is the predicted cut off line.</p>
The client can receive the probabilities he asked for by input the student's Science and Math mark.	Refer to the plot to see if it is approximately the same.

Word count: 542

Marks	Awarded	Description
0		The response does not reach a standard described by the descriptors below.
1–2		The record of tasks and the design overview, including an outline test plan, are limited. From this information it is difficult to see how the product was developed.
3–4		The record of tasks and the design overview, including an outline test plan, are partially complete. They provide a basic understanding of how the product was developed.
5–6		The record of tasks and the design overview, including an outline test plan, are detailed and complete. From this information it is clear how the product was developed.