Islington College

Emerging Programming Platforms and Technologies

CS5004NI

Technical Report

Submitted By:

Lijala Shrestha - 15043033

Bachan Ghimire - 15043036

Amrit Shrestha - 15042984

Group: L2C3

Date: 26th December, 2016

Submitted To:

Mr. Rabin Regmi

Lecturer, IT Faculty

TABLE OF CONTENTS

TABLE OF FIGURES

INTRODUCTION

This is a documentation of our Group Coursework of the module, CS5004NI – Emerging Programming Platforms and Technologies. The development of the program is done in NetBeans Integrated Development Environment, which is a Graphical User Interface (GUI) Builder Interface, using Java programming language.

We were assigned to create a GUI of a Restaurant Menu Information System. The GUI must have certain labels, a combo box, radio buttons, a table, text fields and so forth, as components which has to perform task of setting the menu order in the table and to search respective menu order from the table according to the price as well as category of the dish.

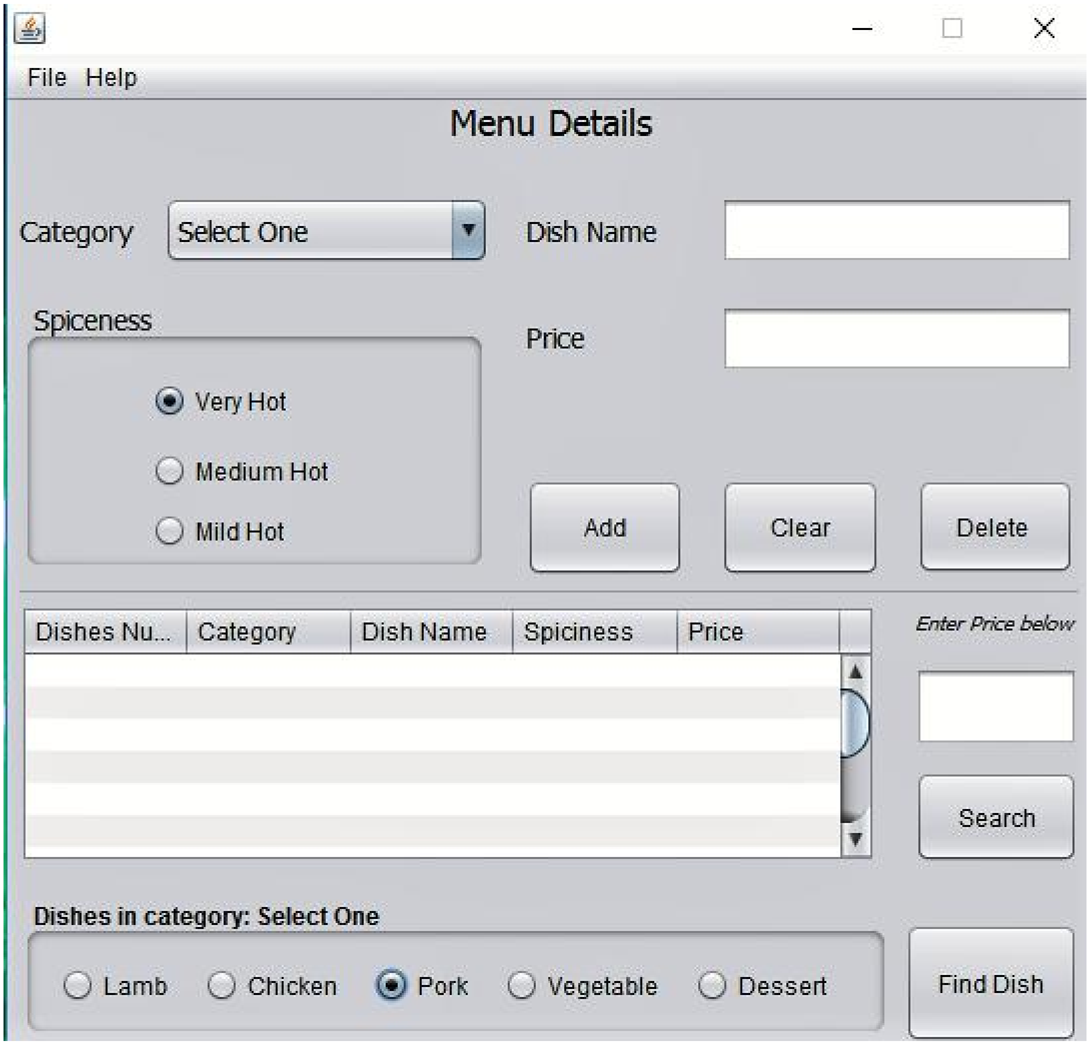


Figure 1:The GUI application

INDIVUDIAL TASKS

Bachan Ghimire – 15043036

- Created the structure of the GUI application
- Developed the core program
- Did application validation
- Finalised the documentation

Lijala Shrestha – 15043033

- Wrote the core group report.
- Updated the GUI structure
- Development of sorting method in the program.
- Critically tested the application.

Amrit Shrestha – 15042984

- Researched on the algorithms implied
- Did the referencing
- Created files used in the GUI application
- Tested the application

REPORT

In this section, we discuss about the GUI application. We will talk about how it was developed, what methods in the class does, and a few testing that demonstrates the GUI application.

SYSTEM DEVELOPMENT

❖ How we used NetBeans for developing our GUI:

We used NetBeans primarily for the development of the JFrame from, for it was really convenient to drag and drop components and give them size, and coordinates for their placement.

We first created a project in NetBeans. Then, a JFrame form Class withing the package of the project was created. The components that were to be placed in the form were dragged and dropped in their respective positions in the form, from a palette provided. There was an option of setting properties of each component. That too was done.

Wherever Action Event was required, we would assign Action Listeners to components. The NetBeans IDE automatically generated the code for creating the GUI and methods for each of the action events. We then wrote the "to do" code in the methods created.

❖ Short descriptions about the methods created in MenuInfo class:

public MenuInfo():

This is a the class constructor of the class MenuInfo. It initialises the components and also initialises a few attributes.

searchButtonActionPerformed(java.awt.event.ActionEvent evt):

This method is called when the search button is clicked. It takes Action Event as parameter and performs the task of searching dish by price.

clearButtonActionPerformed(java.awt.event.ActionEvent evt):

This method is called when the clear button is clicked. This method clears the current selection in GUI input components and sets them to null or empty.

exitMenuItemActionPerformed(java.awt.event.ActionEvent evt):

This method is called when exit menu item is clicked. It will exit the GUI application.

openFileMenuItemActionPerformed(java.awt.event.ActionEvent evt):

This method is called when a file menu item named “Open” is clicked. This method will open a file that introduces the application to the user.

helpMenuMenuKeyPressed(javax.swing.event.MenuKeyEvent evt):

This method is called when a key is pressed within the menu bar. This method opens a help file that provides instruction on using the application.

addButtonActionPerformed(java.awt.event.ActionEvent evt):

This method is called when the add button is clicked. This methods takes all the input from the GUI and stores it in an array. Then the array is looped till the data is filled in the table.

categorySearchButtonActionPerformed():

This method is called when the Find Dish button is clicked. This method takes all the dishes available in a category that is selected and then it displays the dishes that are available.

deleteButtonActionPerformed(java.awt.event.ActionEvent evt):

This method is called when the delete button is pressed. This method takes the selected row from the table and then deletes it.

sort(ArrayList<Integer> array):

This method is called when the search button is clicked, because it is invoked there. It takes an ArrayList of integer type as a parameter and then returns the same array after sorting it.

search(ArrayList<Integer> array, int low, int high, double key):

This method is the implementation of the binary search algorithm. It accepts an array and 3 numbers as parameter, them being lowest and highest index of array, and one being the price entered by the application user.

It searches for the price given my user by implementing binary search algorithm and returns the matched value if the corresponding dish is found. If non matches, then it returns -1 as integer value in default.

❖ How Binary Search Algorithm was used for search function:

Binary search algorithm is a logic in computer science that works on the principle of searching for an element from midpoint of total elements to be searched. In the context of our coursework, a search function was to be performed for price that a user inputs. The search would be done form a table full of dishes and its corresponding prices.

First, the all the dishes' prices were retrieved from the table containing dishes and was stored in an array. The order of array was the same as order of lists in the table. The unsorted array is then passed into a method that sorts the array in ascending order. This method then sorts the array and returns the same array but by sorting it.

Once we have the sorted array, we send the array as a parameter onto a method that does the core searching task. It would also take the least and highest element index number from the array, as well as would take the price entered by user in the GUI. By implementing the mathematical idea of median value, the middle index of array was determined. If the price entered by user matched the value of middle index, it would return the value itself. Whereas, if the value in middle index is higher than the entered price, it would return the method itself, as the method for searching is a method that returns some value in default. When the method itself is returned, the method is called again. The same procedure happens again but the lowest index now as a parameter will be the middle index number of the previously called method. On the contrary, if the value in middle index is lower than the entered price, the method itself is returned as well, but this time the highest index as a parameter will be the the middle index number of the previously called method.

This procedure will keep on going till the entered price matches an element in the array. If the entered price is not found in the array, then the method that does searching returns a value of its' return type in default. Further, this can be represented in diagrammatic way in the following way:

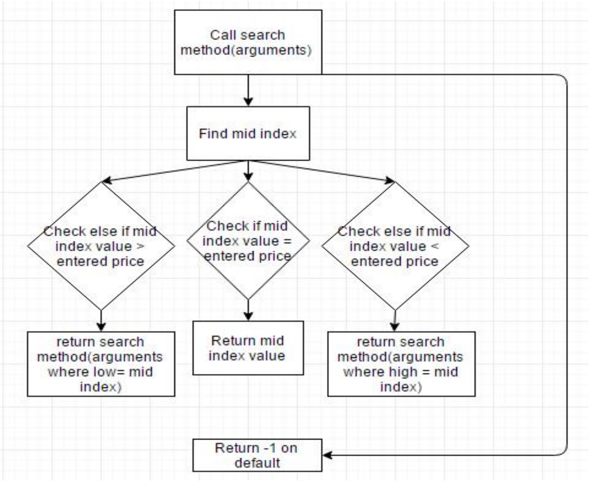
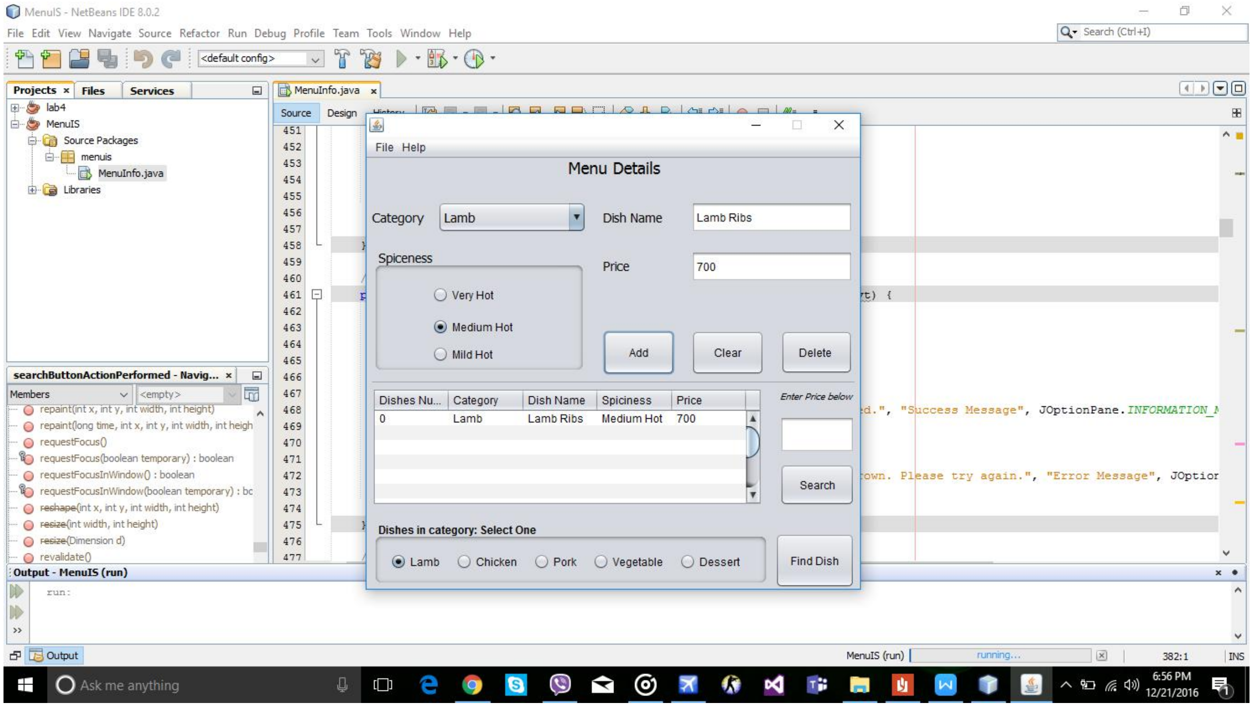

Call search method(arguments)
Find mid index
Check else if mid index value > entered price
Check if mid index value = entered price
Check else if mid index value < entered price
return search method(arguments where low= mid index)
Return mid index value
return search method(arguments where high = mid index)
Return -1 on default

Figure 2: Binary search flow chart

TESTING

Here, we can quite well see that the GUI application is running in the NetBeans IDE.



Menu Details

Category: Lamb
Dish Name: Lamb Ribs
Spiceness: ○ Very Hot ● Medium Hot ○ Mild Hot
Price: 700

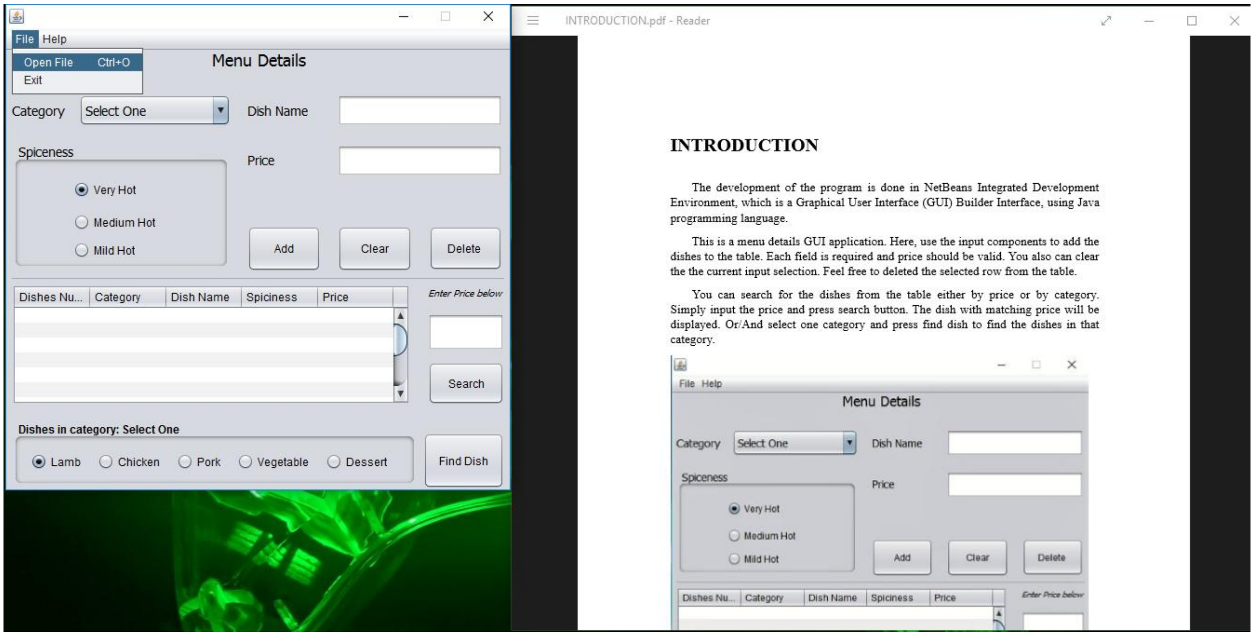
Dishes Nu...	Category	Dish Name	Spiciness	Price
0	Lamb	Lamb Ribs	Medium Hot	700

Dishes in category: Select One
● Lamb ○ Chicken ○ Pork ○ Vegetable ○ Dessert

```
d.", "Success Message", JOptionPane.INFORMATION_M
own. Please try again.", "Error Message", JOptior
```

```
run:
```

Figure 3: Program running in NetBeans



INTRODUCTION

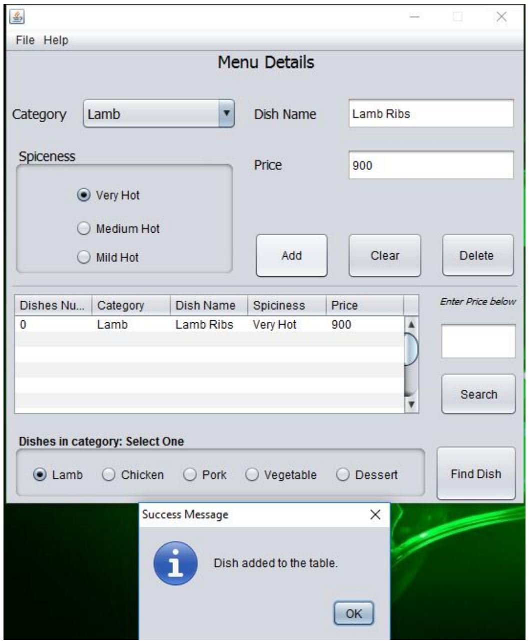
The development of the program is done in NetBeans Integrated Development Environment, which is a Graphical User Interface (GUI) Builder Interface, using Java programming language.

This is a menu details GUI application. Here, use the input components to add the dishes to the table. Each field is required and price should be valid. You also can clear the the current input selection. Feel free to deleted the selected row from the table.

You can search for the dishes from the table either by price or by category. Simply input the price and press search button. The dish with matching price will be displayed. Or/And select one category and press find dish to find the dishes in that category.

Figure 4: Opening file from the application

Adding dishes to the table by using add button



Category: Lamb

Dish Name: Lamb Ribs

Spiceness: ● Very Hot ○ Medium Hot ○ Mild Hot

Price: 900

Dishes Nu...	Category	Dish Name	Spiciness	Price
0	Lamb	Lamb Ribs	Very Hot	900

Dishes in category: Select One

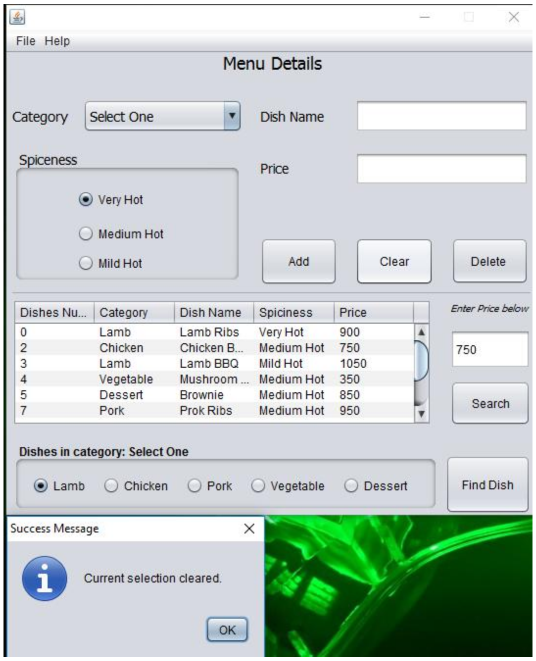
● Lamb ○ Chicken ○ Pork ○ Vegetable ○ Dessert

Success Message

Dish added to the table.

Figure 5: Adding dish to the table

The clear button next to add button has cleared the current selection as seen in the figure below.



Category: Select One

Dish Name:

Spiceness: ● Very Hot ○ Medium Hot ○ Mild Hot

Price:

Dishes Nu...	Category	Dish Name	Spiciness	Price
0	Lamb	Lamb Ribs	Very Hot	900
2	Chicken	Chicken B...	Medium Hot	750
3	Lamb	Lamb BBQ	Mild Hot	1050
4	Vegetable	Mushroom ...	Medium Hot	350
5	Dessert	Brownie	Medium Hot	850
7	Pork	Prok Ribs	Medium Hot	950

Enter Price below: 750

Dishes in category: Select One

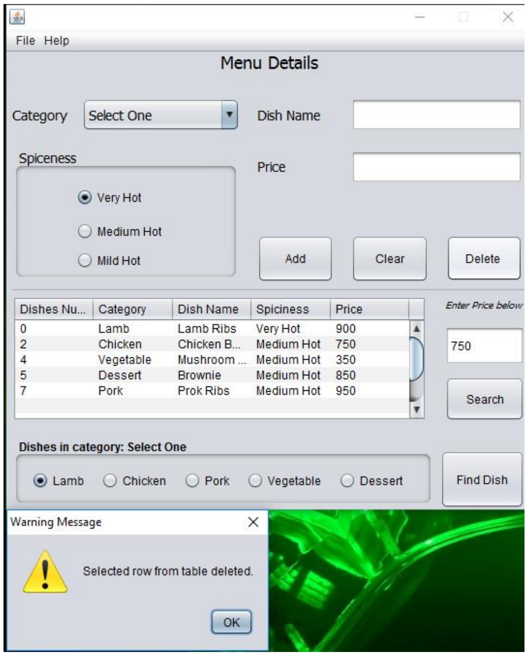
● Lamb ○ Chicken ○ Pork ○ Vegetable ○ Dessert

Success Message

Current selection cleared.

Figure 6: Clearing of current selection

Here, we can see that a dialog box has appeared that says the currently selected row has been deleted.



Menu Details

Category: Select One

Dish Name:

Spiceness: ● Very Hot ○ Medium Hot ○ Mild Hot

Price:

Dishes Nu...	Category	Dish Name	Spiceness	Price
0	Lamb	Lamb Ribs	Very Hot	900
2	Chicken	Chicken B...	Medium Hot	750
4	Vegetable	Mushroom ...	Medium Hot	350
5	Dessert	Brownie	Medium Hot	850
7	Pork	Prok Ribs	Medium Hot	950

Enter Price below: 750

Dishes in category: Select One

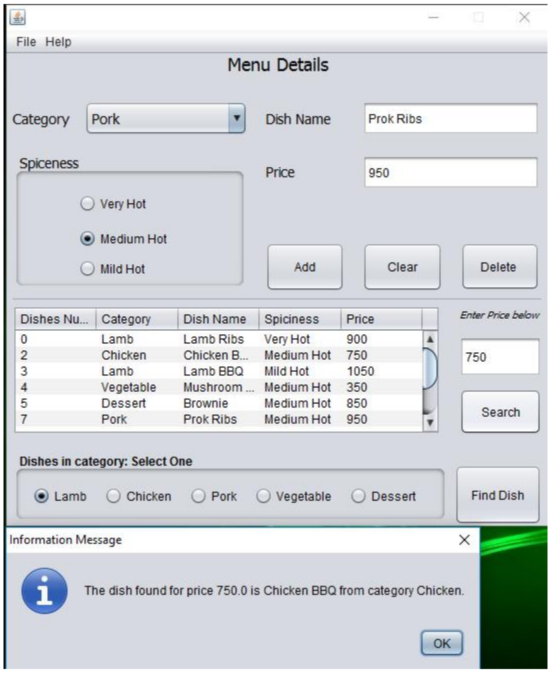
● Lamb ○ Chicken ○ Pork ○ Vegetable ○ Dessert

Warning Message

Selected row from table deleted.

Figure 7: Deleting selected row

We can see in the figure below that, when price is entered just above search button and when it is pressed, it checks in the table for the dish with matching price. Here, it was found.



Menu Details

Category: Pork

Dish Name: Prok Ribs

Spiceness: ○ Very Hot ● Medium Hot ○ Mild Hot

Price: 950

Dishes Nu...	Category	Dish Name	Spiceness	Price
0	Lamb	Lamb Ribs	Very Hot	900
2	Chicken	Chicken B...	Medium Hot	750
3	Lamb	Lamb BBQ	Mild Hot	1050
4	Vegetable	Mushroom ...	Medium Hot	350
5	Dessert	Brownie	Medium Hot	850
7	Pork	Prok Ribs	Medium Hot	950

Enter Price below: 750

Dishes in category: Select One

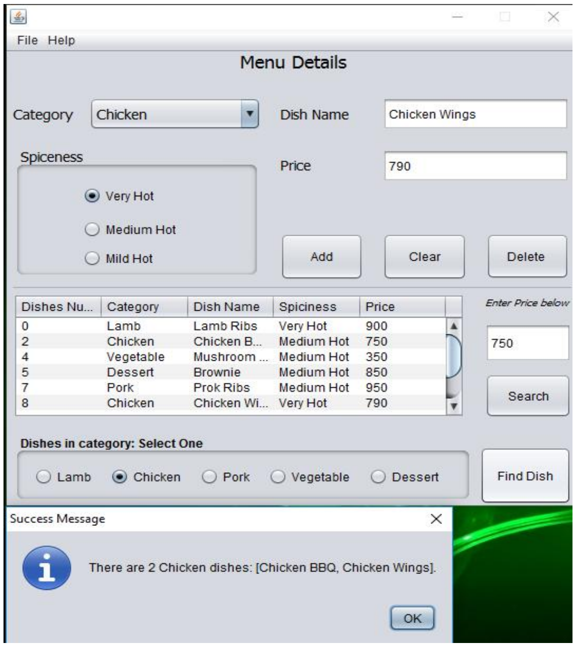
● Lamb ○ Chicken ○ Pork ○ Vegetable ○ Dessert

Information Message

The dish found for price 750.0 is Chicken BBQ from category Chicken.

Figure 8: Searching by price

The searching of dish by category is done. Here, for the category chicken, the program goes through the table to find the dishes with category that is selected.



Menu Details

Category: Chicken
Dish Name: Chicken Wings
Spiceness: ● Very Hot ○ Medium Hot ○ Mild Hot
Price: 790

Dishes Nu...	Category	Dish Name	Spiciness	Price
0	Lamb	Lamb Ribs	Very Hot	900
2	Chicken	Chicken B...	Medium Hot	750
4	Vegetable	Mushroom ...	Medium Hot	350
5	Dessert	Brownie	Medium Hot	850
7	Pork	Prok Ribs	Medium Hot	950
8	Chicken	Chicken Wi...	Very Hot	790

Enter Price below: 750

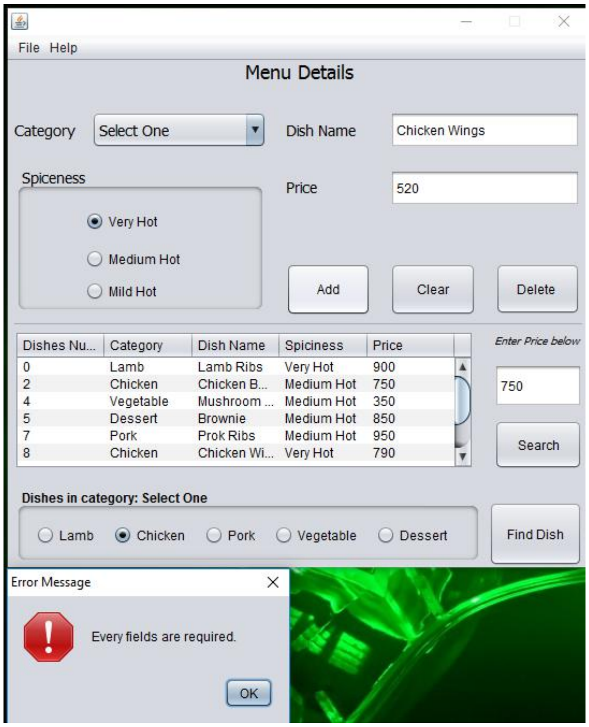
Dishes in category: Select One
○ Lamb ● Chicken ○ Pork ○ Vegetable ○ Dessert

Success Message

There are 2 Chicken dishes: [Chicken BBQ, Chicken Wings].

Figure 9: Searching by category

Now, let us demonstrate validation of the system. It will let user know all kind of possible errors that might occur during the usage of the system.



Menu Details

Category: Select One
Dish Name: Chicken Wings
Spiceness: ● Very Hot ○ Medium Hot ○ Mild Hot
Price: 520

Dishes Nu...	Category	Dish Name	Spiciness	Price
0	Lamb	Lamb Ribs	Very Hot	900
2	Chicken	Chicken B...	Medium Hot	750
4	Vegetable	Mushroom ...	Medium Hot	350
5	Dessert	Brownie	Medium Hot	850
7	Pork	Prok Ribs	Medium Hot	950
8	Chicken	Chicken Wi...	Very Hot	790

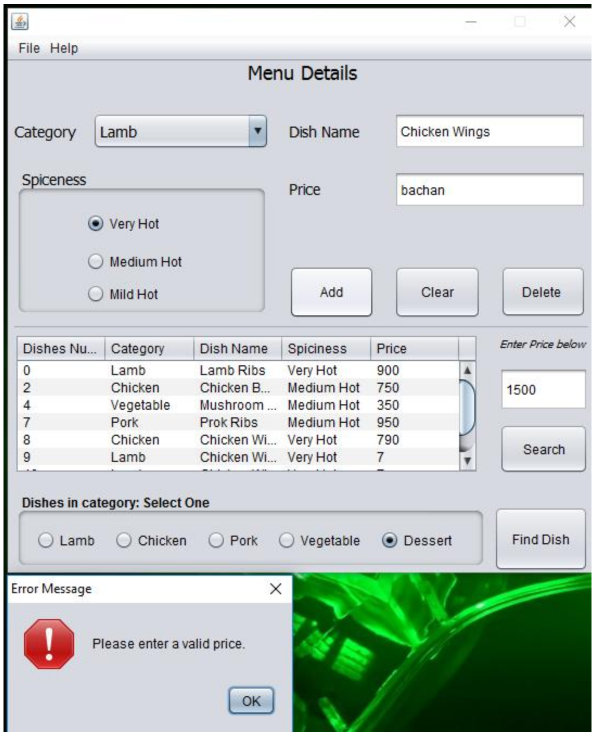
Enter Price below: 750

Dishes in category: Select One
○ Lamb ● Chicken ○ Pork ○ Vegetable ○ Dessert

Error Message

Every fields are required.

Figure 10: All fields required validation



Menu Details

Category: Lamb

Dish Name: Chicken Wings

Spiceness

● Very Hot
○ Medium Hot
○ Mild Hot

Price: bachan

Dishes Nu...	Category	Dish Name	Spiciness	Price
0	Lamb	Lamb Ribs	Very Hot	900
2	Chicken	Chicken B...	Medium Hot	750
4	Vegetable	Mushroom ...	Medium Hot	350
7	Pork	Prok Ribs	Medium Hot	950
8	Chicken	Chicken Wi...	Very Hot	790
9	Lamb	Chicken Wi...	Very Hot	7

Enter Price below

1500

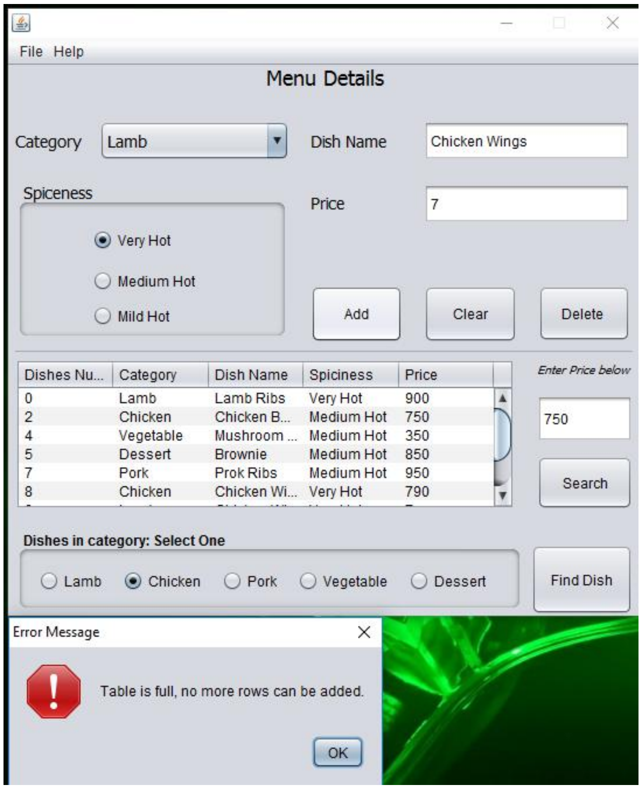
Dishes in category: Select One

○ Lamb ○ Chicken ○ Pork ○ Vegetable ● Dessert

Error Message

Please enter a valid price.

Figure 11: Correct price validation



Menu Details

Category: Lamb

Dish Name: Chicken Wings

Spiceness

● Very Hot
○ Medium Hot
○ Mild Hot

Price: 7

Dishes Nu...	Category	Dish Name	Spiciness	Price
0	Lamb	Lamb Ribs	Very Hot	900
2	Chicken	Chicken B...	Medium Hot	750
4	Vegetable	Mushroom ...	Medium Hot	350
5	Dessert	Brownie	Medium Hot	850
7	Pork	Prok Ribs	Medium Hot	950
8	Chicken	Chicken Wi...	Very Hot	790

Enter Price below

750

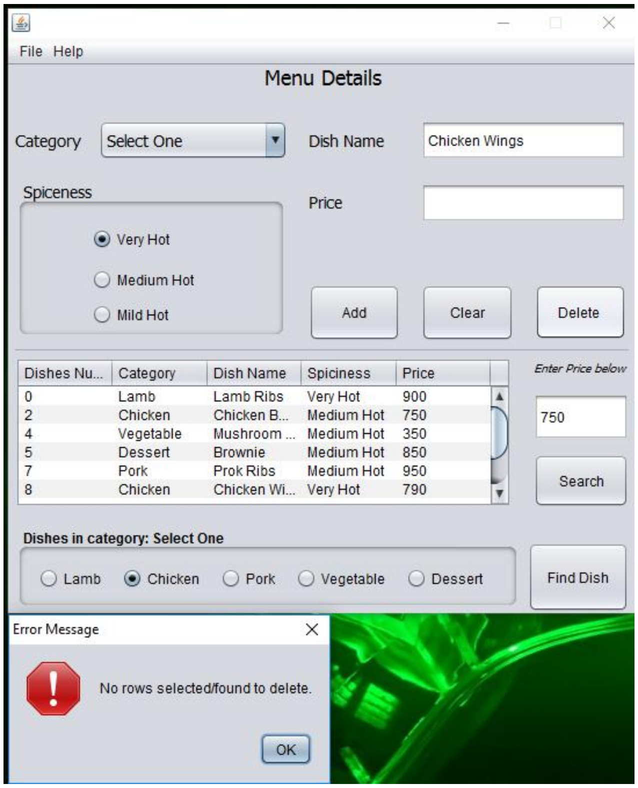
Dishes in category: Select One

○ Lamb ● Chicken ○ Pork ○ Vegetable ○ Dessert

Error Message

Table is full, no more rows can be added.

Figure 12: Table length validation



Menu Details

Category: Select One

Dish Name: Chicken Wings

Spiceness: ● Very Hot ○ Medium Hot ○ Mild Hot

Price:

Dishes Nu...	Category	Dish Name	Spiciness	Price
0	Lamb	Lamb Ribs	Very Hot	900
2	Chicken	Chicken B...	Medium Hot	750
4	Vegetable	Mushroom ...	Medium Hot	350
5	Dessert	Brownie	Medium Hot	850
7	Pork	Prok Ribs	Medium Hot	950
8	Chicken	Chicken Wi...	Very Hot	790

Enter Price below: 750

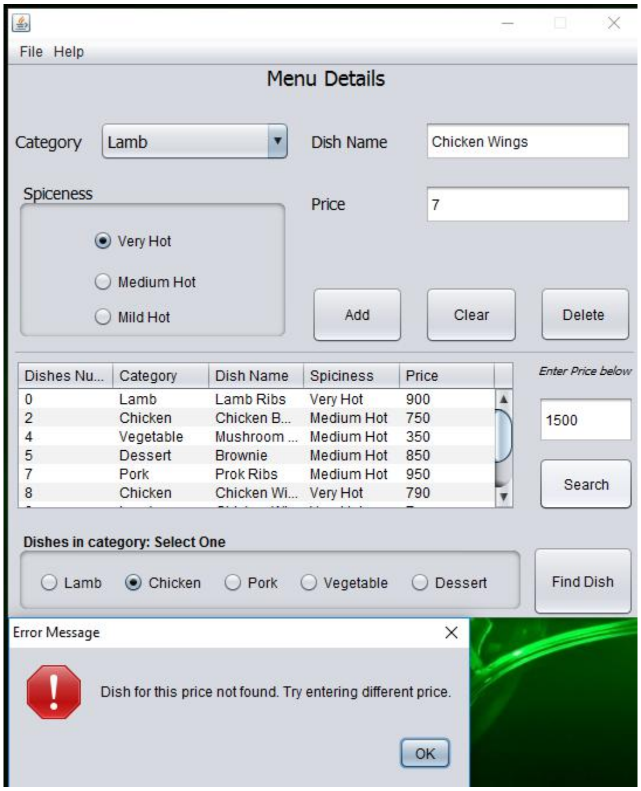
Dishes in category: Select One

○ Lamb ● Chicken ○ Pork ○ Vegetable ○ Dessert

Error Message

No rows selected/found to delete.

Figure 13: Row selection validation for deleting



Menu Details

Category: Lamb

Dish Name: Chicken Wings

Spiceness: ● Very Hot ○ Medium Hot ○ Mild Hot

Price: 7

Dishes Nu...	Category	Dish Name	Spiciness	Price
0	Lamb	Lamb Ribs	Very Hot	900
2	Chicken	Chicken B...	Medium Hot	750
4	Vegetable	Mushroom ...	Medium Hot	350
5	Dessert	Brownie	Medium Hot	850
7	Pork	Prok Ribs	Medium Hot	950
8	Chicken	Chicken Wi...	Very Hot	790

Enter Price below: 1500

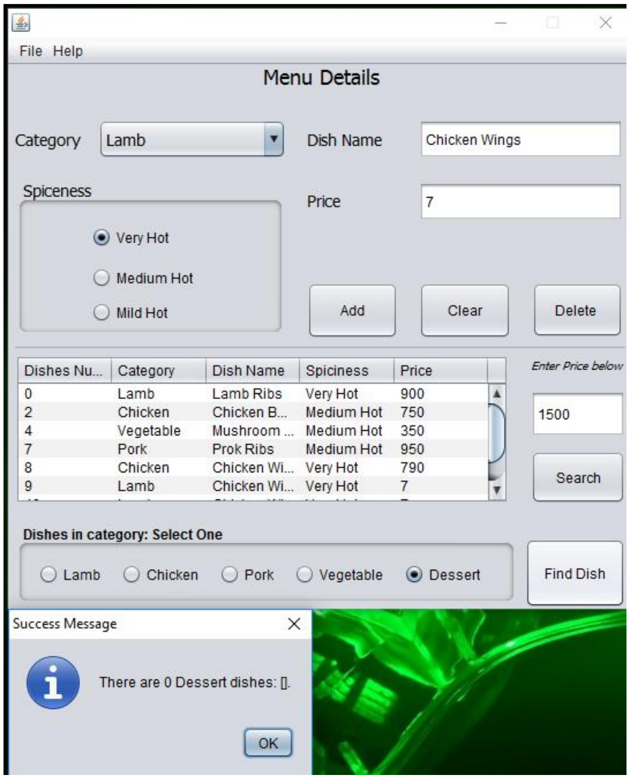
Dishes in category: Select One

○ Lamb ● Chicken ○ Pork ○ Vegetable ○ Dessert

Error Message

Dish for this price not found. Try entering different price.

Figure 14: Dish for entered price validation



Menu Details

Category: Lamb

Dish Name: Chicken Wings

Spiceness: ● Very Hot ○ Medium Hot ○ Mild Hot

Price: 7

Dishes Nu...	Category	Dish Name	Spiciness	Price
0	Lamb	Lamb Ribs	Very Hot	900
2	Chicken	Chicken B...	Medium Hot	750
4	Vegetable	Mushroom ...	Medium Hot	350
7	Pork	Prok Ribs	Medium Hot	950
8	Chicken	Chicken Wi...	Very Hot	790
9	Lamb	Chicken Wi...	Very Hot	7

Enter Price below

1500

Dishes in category: Select One

○ Lamb ○ Chicken ○ Pork ○ Vegetable ● Dessert

There are 0 Dessert dishes: [].

Figure 15: Dish for entered category validation

CONCLUSION

Throughout this project, we learned a new way to design and code to create Graphical User Interface (GUI) and use ArrayList. We also learned additional things while creating a GUI, such as, binary search algorithm and bubble sort. Likewise, we got the opportunity to enhance our creativity via this coursework as it made us think logically. Primarily, we were able to understand the concept of GUI, binary search algorithm as well as bubble sort method.

Furthermore, we did go through some difficulties using binary search algorithm, as well as, making the Add button work when it came to assembling the dish index number. However, we did not let our confusions overrule us. We did some research online and even consulted with our friends on the issues that were troubling us. This way, we were able to compile and execute our program.

Additionally, this is one of the biggest GUI program we have developed so far, which has allowed each of us to grasp better knowledge of programming and GUI. We also experienced various errors like compilation errors and run time errors, and learned how to deal with them. This is how we understood how research based work is like. We are content application despite the fact we may still be incompetent with our development. All in all, this project helped us to boost our creativity and programming skills.

REFERENCES

DZone, n.d.. *Different Approaches To Sorting Elements of an ArrayList in Java - DZone Java.* [Online]
Available at: https://dzone.com/articles/sorting-java-arraylist
[Accessed 9 December 2016].

Ericson, B., 2014. *Getting the Number of Rows and Columns — Java Review.* [Online]
Available at: http://ice-web.cc.gatech.edu/ce21/1/static/JavaReview-RU/Array2dBasics/a2dLoop.html
[Accessed 20 December 2016].

java2s, n.d.. *Fixed data vs dynamic data Table : Table Model « Swing JFC « Java.* [Online]
Available at: http://www.java2s.com/Code/Java/Swing-JFC/FixeddatavsdynamicdataTable.htm
[Accessed 16 December 2016].

NetBeans, 2016. *Designing a Swing GUI in NetBeans IDE - Tutorial.* [Online]
Available at: https://netbeans.org/kb/docs/java/quickstart-gui.html#multiple
[Accessed 2 December 2016].

Oracle, 2016. *NullPointerException (Java Platform SE 7).* [Online]
Available at: https://docs.oracle.com/javase/7/docs/api/java/lang/NullPointerException.html
[Accessed 15 December 2016].

Sanfoundry, 2016. *Java Program to Find the Median of two Sorted Arrays using Binary Search Approach - Sanfoundry.* [Online]
Available at: http://www.sanfoundry.com/java-program-find-median-array-using-binary-search-approach/
[Accessed 7 December 2016].

topcoder, 2016. *Binary Search – topcoder.* [Online]
Available at: https://www.topcoder.com/community/data-science/data-science-tutorials/binary-search/
[Accessed 8 December 2016].