LONDON METROPOLITAN UNIVERSITY

Islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CS5004NA Emerging Programming Platforms and Technologies

Assessment Weightage & Type

50% Group Coursework

Title (Where Required):

Year and Semester

2019-20 Autumn / 2019-20 Spring

Group Name:			
SN	**Student Name**	**College ID**	**University ID**
1	**Sohail Dangol**	**np01cp4a180122**	**18029901**
2	**Sai Shrestha**	**np01cp4a180261**	**18030036**
3	**Yogesh Shrestha**	**np01cp4a180265**	**18030028**

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

.

Contents

Table of Tables

Table of Figures

CS5004NA

INDIVIDUAL TASKS

Group Member	Tasks
Sohail Dongol	Binary Searching, Menu
Sai Shrestha	Linear Searching, Sorting
Yogesh Shrestha	GUI, Debugging and Testing, Report

INTRODUCTION

In this assessment we are required to propose and develop an Information System. In this assignment we have developed a java swing-based a bike servicing application, 'Husqvarna Bike Service Application'. Only the admin has the privilege to add new bike models of specific Husqvarna Manufacturer. The client visits the reception of the servicing company and provides his/her personal details (Name, Age, Address, Phone no.) and bike information (Model, Bike no., Kms drove, Chassis no., Last servicing date). Then the data values are added to the table. Also, the GUI has been as user-friendly and attractive as possible.

List of Features:

Below mentioned are the list of features of our Husqvarna Bike Service Application:

- The attributes of the bikes can be added only by the admin.
- The table will feature all the bikes available with the categories and their attributes.
- Quick search can be performed by either Bike Model using linear search or price using binary search.
- Guide to use the GUI can be found at taskbar of the GUI under Help.

List of Data:

The application consists of the following categories where the user can add or search attributes according to his/her needs.

- Name
- Age
- Address
- Phone_no.
- Bike_Model_Name
- Bike_number

- Kms drove
- Chassis number
- Last_servicing_date
- Price

Tools Used:

NetBeans

For programming side of the project, we used NetBeans for it. We used NetBeans because it is very user friendly and holds many features that us to easily write codes. We can also easily create a GUI for the project by the help of this application.

Microsoft Visio

We used Microsoft Visio for creating templates and prototypes of the GUI. It helps us to see the looks and feels of the UI which in turn helps us to create the GUI very easily and quickly.

Binary Searching

Binary search is a quick search algorithm that works on the principle of divide and conquer. However, all the data should be in sorted form. In order to carry out the searching method, the values are taken in an array, the array is the sorted. After the array is sorted, the searching can be done. The process is explained below. (Tutorailspoint, 2020)

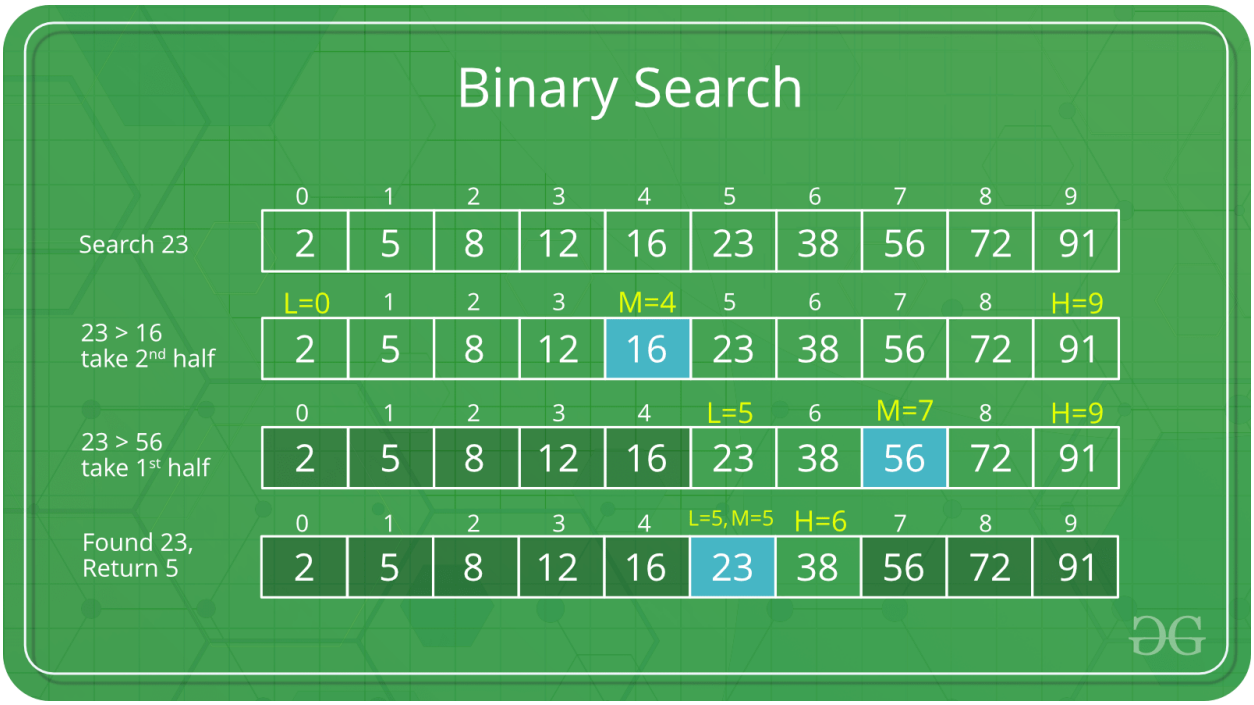

Figure 1:Example: Binary Search (geeksforgeeks, n.d.)

Sorting

All the values are added in an array. Then the values of the price are collected by using the index of the array. Now, the first value in the zeroth index is compared to the value that was collected at the last i.e. the maximum index. They are compared and if the first one is greater than they are swapped. They are swapped by temp variable.

If not, then they are not swapped. Then, the minimum index remains the same but the maximum index is decreased by one. The newly swapped value in the first position is compared to that of the value of the new maximum index. This process goes on until the maximum value is equal to the minimum value. After the completion of this loop, the minimum index value is increased by one, maximum is reset to the original value. Then, the first loop is repeated until the maximum value is equal to the minimum value. Then

the second loop is executed, and after completion the first loop is executed until the array is sorted.

Searching

The values which are to be searched are taken from the array which was created after sorting. As the values are sorted so the value at in the first position i.e. 0 index is the minimum value and the value at the last position is taken as the maximum value. Then, the value at the middle position of the array is taken. The middle value is compared with the value to be searched, if they are equal then the value is output and is terminated.

If they are not equal, then they are compared for which value is greater. If the value to be searched is greater than the middle value, then the middle value is taken as the lowest value and the vale at the last is taken as maximum value. Then, the value at the middle position is taken. However, if the value to be searched is less than the middle value, then first value is taken as the lowest value and the middle value is taken as the maximum value. Then, the value at the middle position is taken.

Then, the first process is repeated until the middle value and searched value are equal. If they are not equal, then the second process is executed. This loop continues until the middle and searched values are equal.

Methods

Sorting

All the values are added in an array. Then the values of the price are collected by using the index of the array. Now, the first value in the zeroth index is compared to the value that was collected at the last i.e. the maximum index. They are compared and if the first one is greater than they are swapped. They are swapped by temp variable.

If not, then they are not swapped. Then, the minimum index remains the same but the maximum index is decreased by one. The newly swapped value in the first position is compared to that of the value of the new maximum index. This process goes on until the maximum value is equal to the minimum value. After the completion of this loop, the minimum index value is increased by one, maximum is reset to the original value. Then, the first loop is repeated until the maximum value is equal to the minimum value. Then the second loop is executed, and after completion the first loop is executed until the array is sorted.

Add value in Table

This method checks if the column where the data is to be placed is an empty column or not. There are four local variables. Two of them are, rowCount and colCount, which stores the row count and the column count in the tables respectively. The method checks if value at the first row is null or occupied given that if the first row is empty in a column, then all the rows of that column are empty.

If the first row is not occupied then the value, then the value is appended in the tables. If the row is occupied then, the value of the nextRow is increased by one, and the process is repeated until the value of a firstRow is null.

Clear

This method sets all the value of the Text Fields to “”, clears all button groups selections, sets index to 0 of all of the combo box.

Add

After having given all the values in the text field after the add action performed button is called, the method checks the data input for valid data, if the values are not valid, then it displays an error message. If the data input is valid then, the method retrieves all the value from the GUI form and adds all the values in an array which is created in the beginning. Then, all the values of the text fields are set to “”.

Exit

When the exit action performed method is called the program is terminated.

Search Price

The search price method is based on the binary search process. When the search price action performed method is called the values which are to be searched are taken from the array which was created after sorting. As the values are sorted so the value at in the first position i.e. 0 index is the minimum value and the value at the last position is taken as the maximum value. Then, the value at the middle position of the array is taken. The middle value is compared with the value to be searched, if they are equal then the value is output and is terminated.

If they are not equal, then they are compared for which value is greater. If the value to be searched is greater than the middle value, then the middle value is taken as the lowest value and the vale at the last is taken as maximum value. Then, the value at the middle position is taken. However, if the value to be searched is less than the middle value, then first value is taken as the lowest value and the middle value is taken as the maximum value. Then, the value at the middle position is taken.

Search Model

The search model method is based on the linear search process. Linear search is a very simple sequential search algorithm. A sequential search is made one by one over all items until the result matches. Unlike the binary search, array does not have to sorted out. we have implemented linear search method in our project, to find data using model no. of a motorcycles. The model no. is stored in the array as i=4 ; i<array.size(); i+10. The model no. to be found is than compared with every i+10 location values until the model number matches.

Help

When the help method is called, the method opens a .txt file which contains information and instruction of the program and how to use the program.

Open

When the open method is called, a file chooser open ups where the user can browse for .txt files. The .txt files should contain information about the client's records in the databases. After opening the .txt file, the method appends the value in the table which is displayed in the GUI.

Testing

Test 1 – Run Program

Table 1 Run Program

Objective	To run program in netbeans
Action	The program was executed
Expected Result	Program should run
Actual Report	Program runs
Conclusion	Test Successful

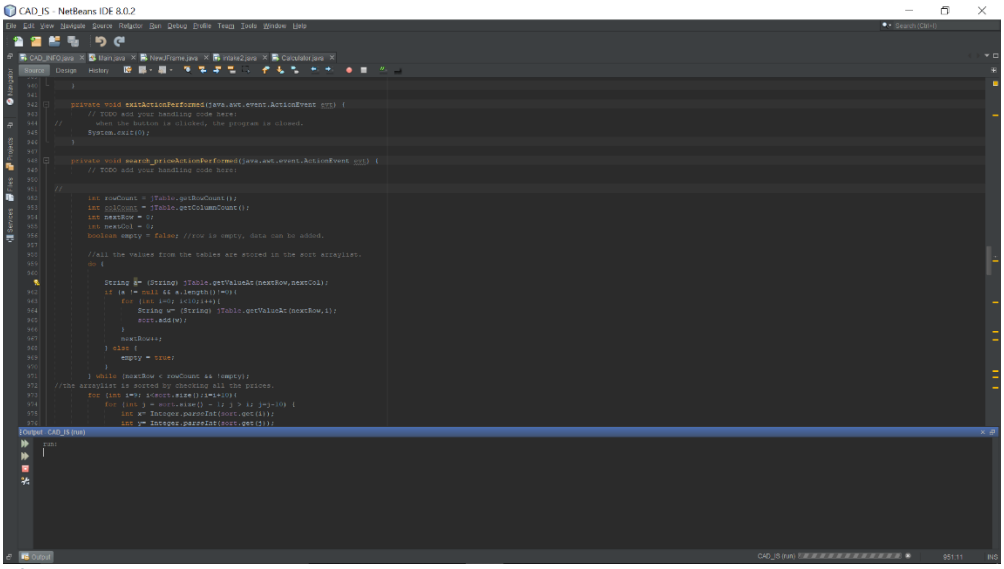

Figure 2 Run porgram

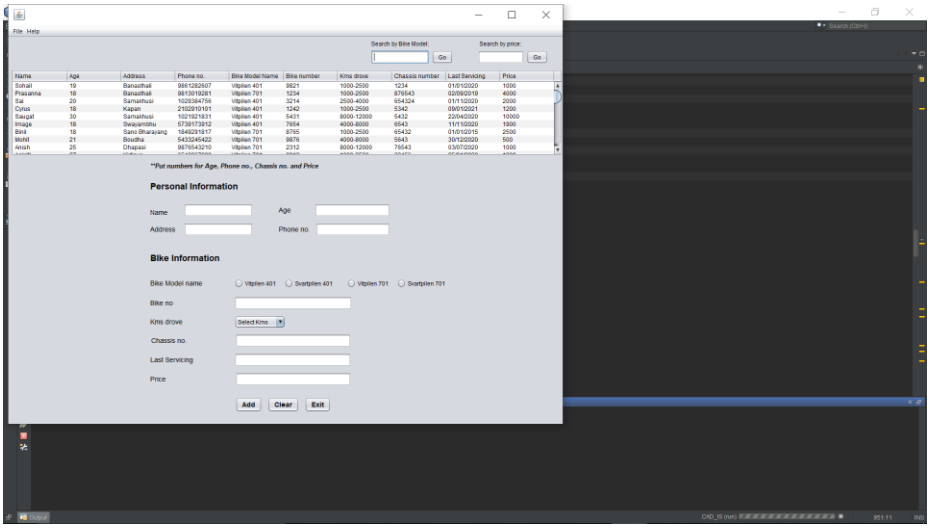

Figure 3 Compile {gram

Test 2

Test 2.1 - Evidence for adding items to table:

Table 2 adding items to table

Objective	To add items to table
Action	The program was executed
Expected Result	items should be added to table
Actual Report	Program runs
Conclusion	Test Successful

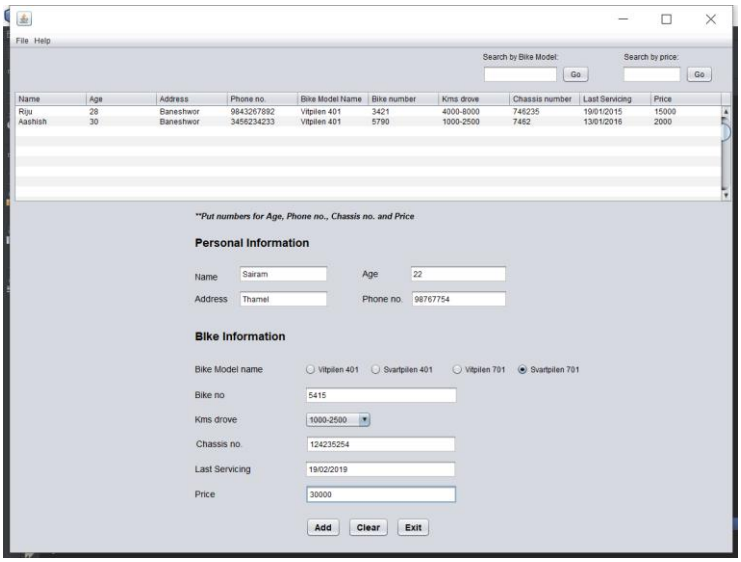

Figure 4 before adding items to table

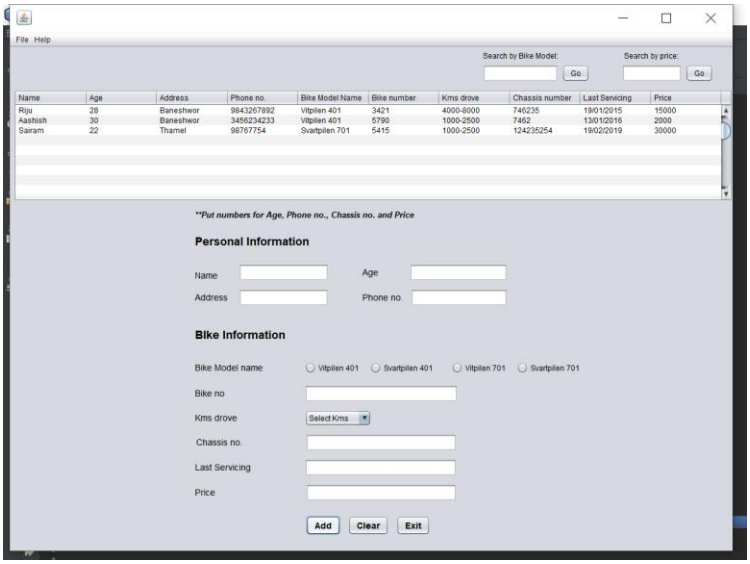

Figure 5 after adding items to table

Task2.2-Evidence of searching for items based on price and bike model

1. Searching items based on model:

Table 3 Searching items based on model

Objective	**To Search items based on model**
Action	The program was executed
Expected Result	Searched item should be displayed
Actual Report	Program runs
Conclusion	Test Successful

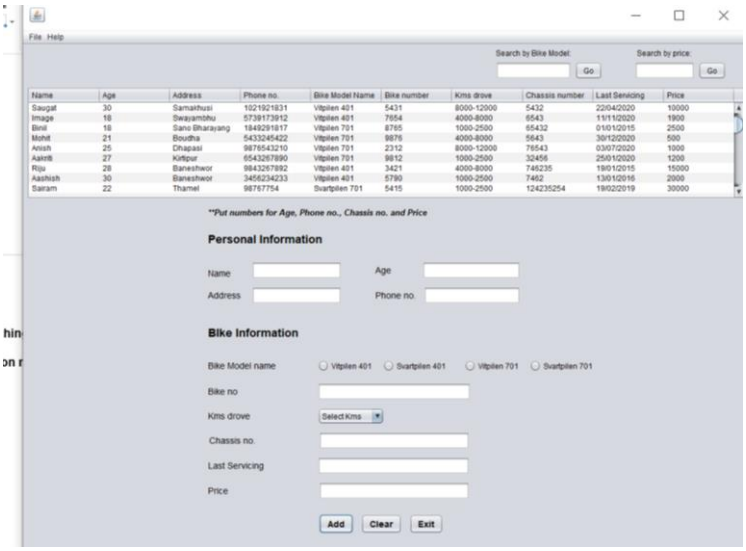

Figure 6 Before Searching items based on model

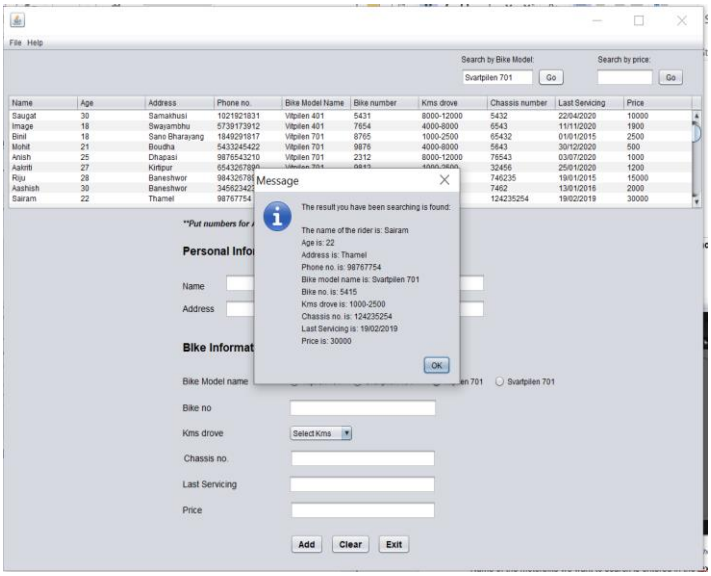

Figure 7 After Searching item on model

2. Searching items based on price:

Table 4 Searching items based on price

Objective	**To Search items based on pricel**
Action	The program was executed
Expected Result	Searched item should be displayed
Actual Report	Program runs
Conclusion	Test Successful

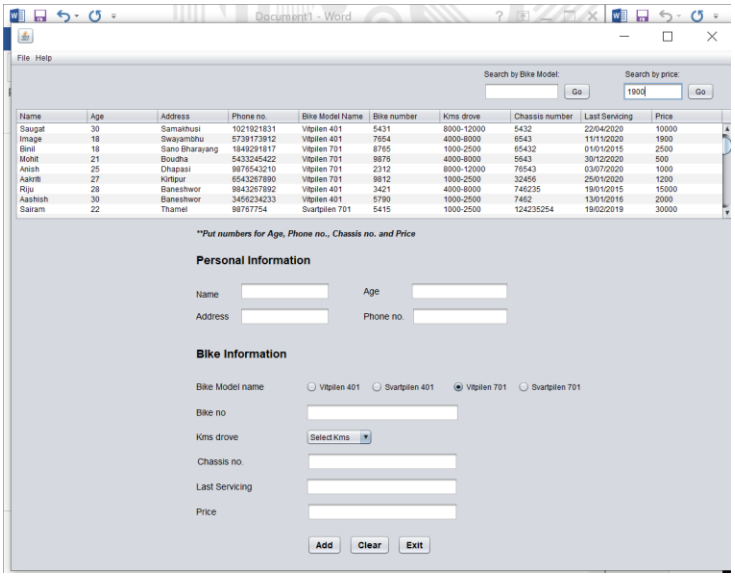

Figure 8 Before Searching items based on price

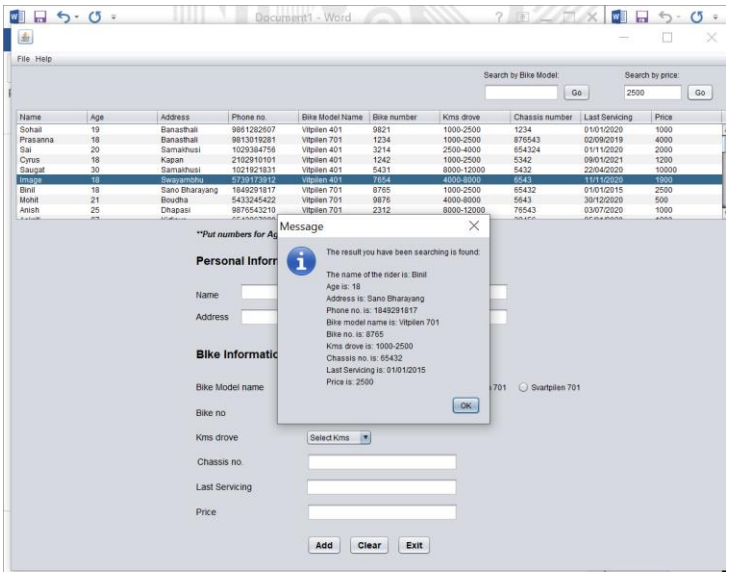

Figure 9 After Searching items based on price

Task 3 - Evidence of opening a file from menu

Table 5 Open FIle

Objective	**To opening a file from menu**
Action	The program was executed
Expected Result	File should open
Actual Report	Program runs
Conclusion	Test Successful

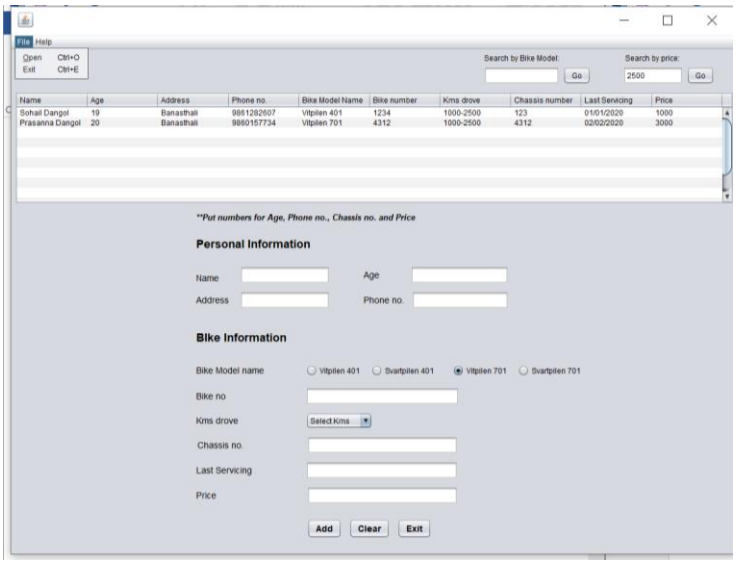

Figure 10 Evidence of opening a file from menu

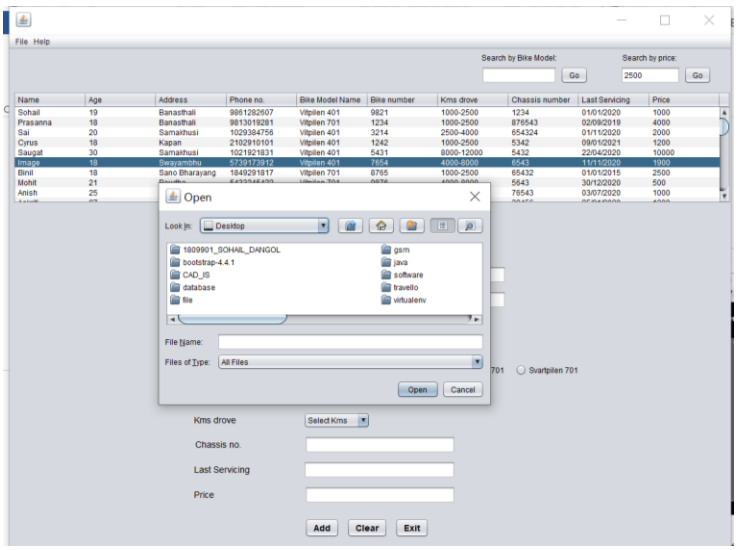

Figure 11 File Chooser

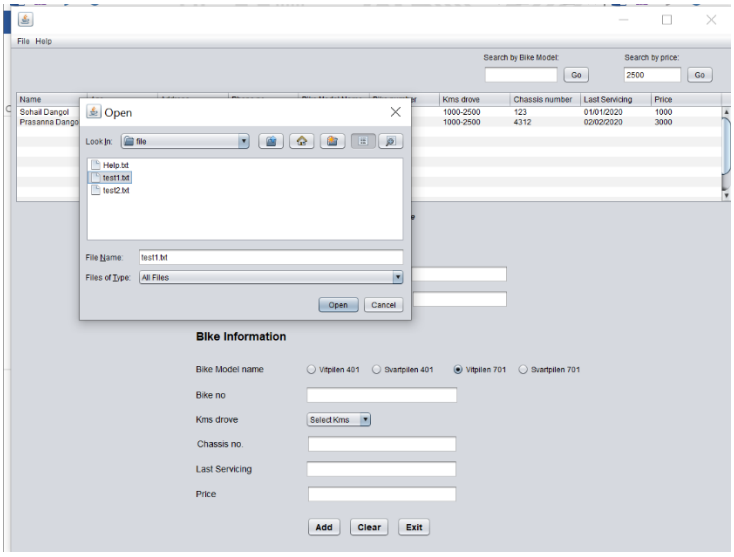

Figure 12 Locating File

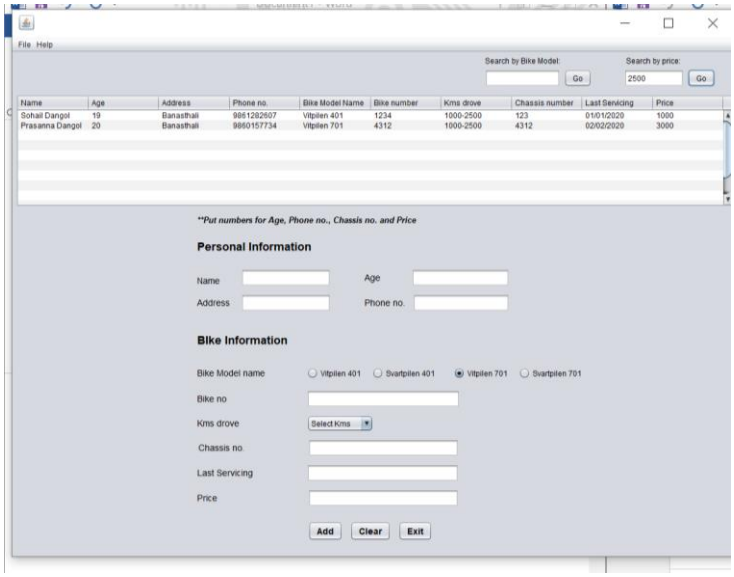

Figure 13 File Opens

Test 4 - Evidence of system validation

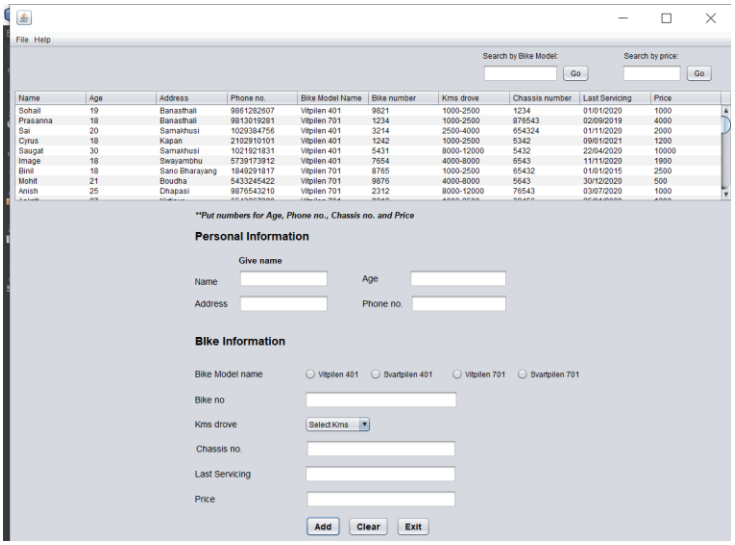

Figure 14 Name Validation

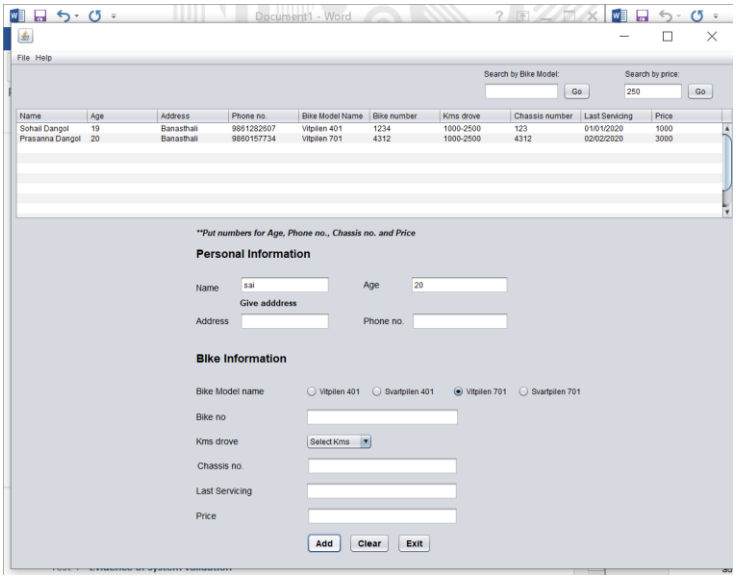

Figure 15 Address Validation

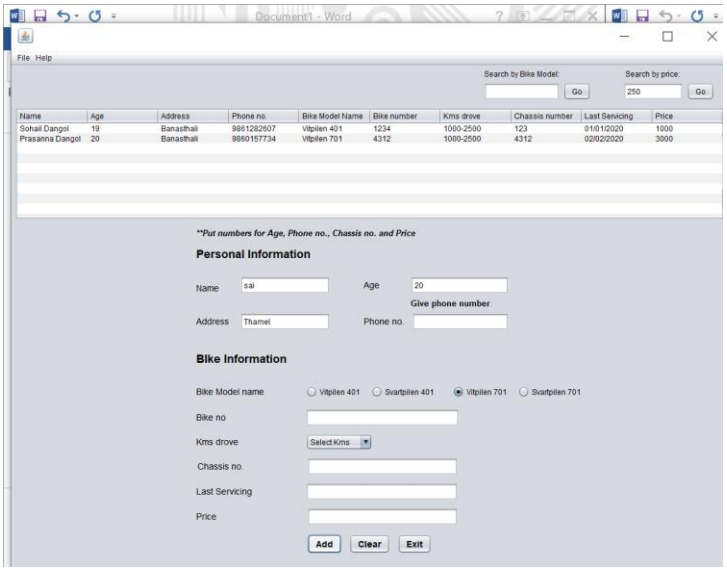

Figure 16 Phone number Validation

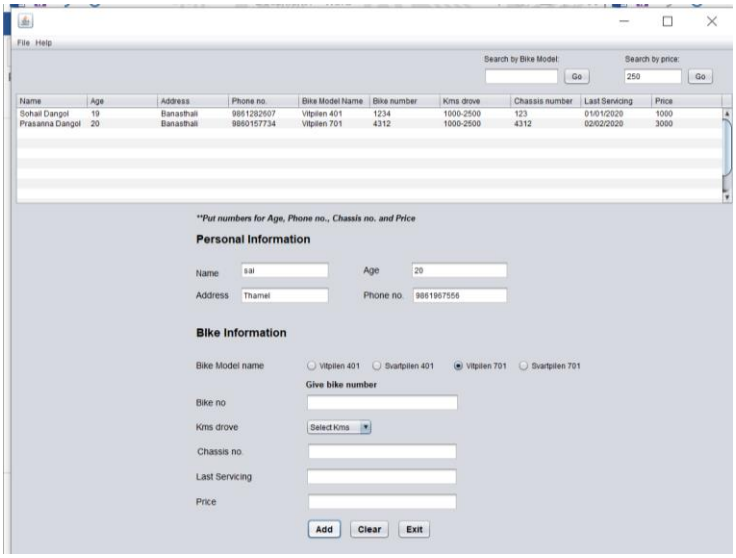

Figure 17 Bike Number 15 Validation

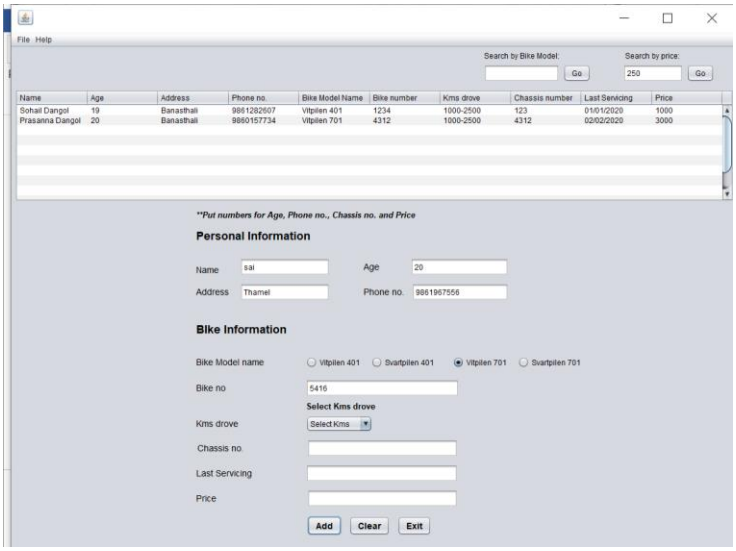

Figure 18 Kilometres Drove Validation

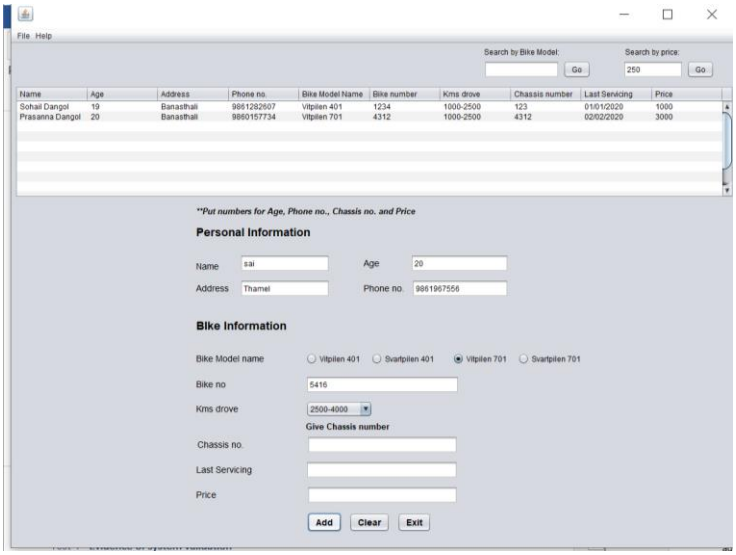

Figure 19 Chassis Number Validation

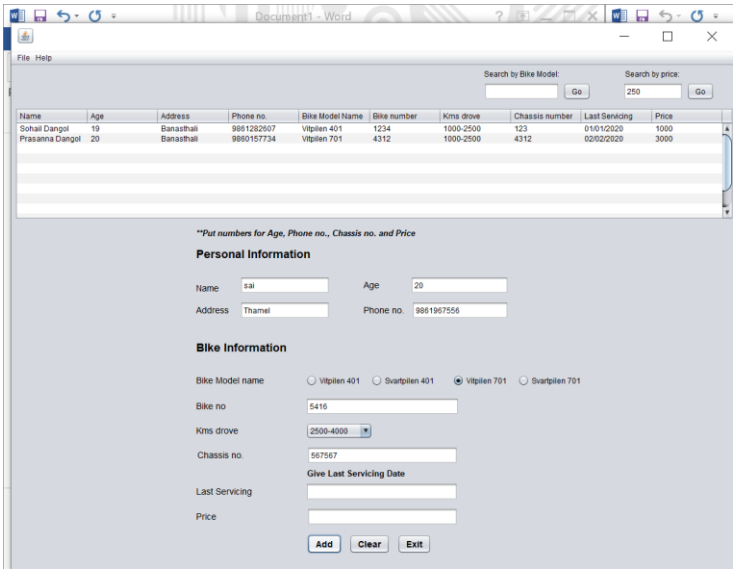

Figure 20 Last Servicing Validation

Conclusion

The purpose of this assessment was to gain knowledge on proposing and developing an Information System. In this assignment we have developed a java swing-based a bike servicing application named Husqvarna Bike Service Application. We had to build this program with special requirement (user with/out privileges, adaption of binary and linear search algorithm).

We faced different difficulties developing the information system. Some of them being validating text fields, binary search algorithm, and linear search algorithm. By going through the lecture slides, attending lab and tutorial classes and surfing the net we managed to overcome these problems.

This coursework enabled us to use our theoretical knowledge of java into a practical code. Lastly, after the completion of this assignment we have gained additional knowledge on java and feel more confident and practical in programming.

APPENDIX:

A pictorial example of Linear Search has been provided below:

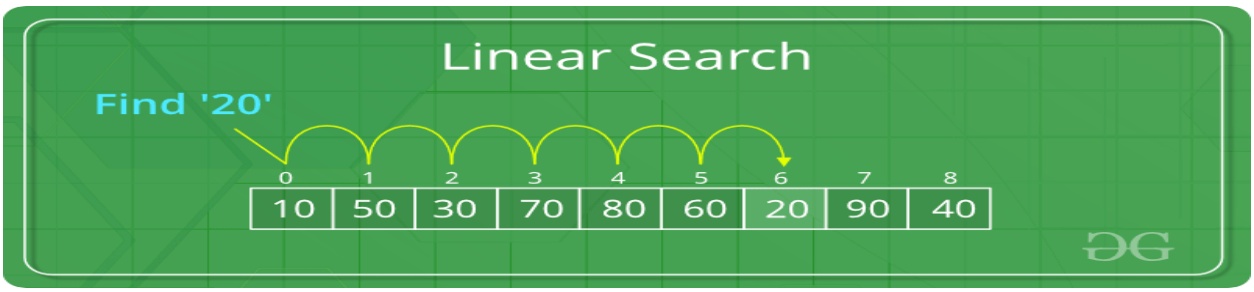

Figure 21:- Example: Linear Search (geeksforgeeks, n.d.)

Here, we compare the value stored from location 1, with value being searched which in this case is 20. We find that value at location 1 is 10, which does not match. As linear search pattern is sequential. It then compares the value with the value stored in location 2 i.e. 50. It's not a match either. The search continues with location 3 and so on until the value is matched with location 6 i.e. value to find:20 = value of location 6:20. It then returns the location index number.

Bibliography:

Tutorialspoint.com. (n.d.). *Data Structure and Algorithms Linear Search - Tutorialspoint*. [online] Available at: https://www.tutorialspoint.com/data_structures_algorithms/linear_search_algorithm.htm [Accessed 11 Jan. 2020].

geeksforgeeks (n.d.). *A simple approach is to do linear search*. [image] Available at: https://www.geeksforgeeks.org/wp-content/uploads/Linear-Search.png [Accessed 11 Jan. 2020].

Tutorialspoint.com. (n.d.). *Data Structure and Algorithms Binary Search - Tutorialspoint*. [online] Available at: https://www.tutorialspoint.com/data_structures_algorithms/binary_search_algorithm.htm [Accessed 11 Jan. 2020].