



slington college
(इस्लिंग्टन 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

INTRODUCTION	1
LIST OF FEATURES:	1
LIST OF DATA:.....	1
TOOLS USED:.....	2
<i>NetBeans</i>	2
<i>Microsoft Visio</i>	2
BINARY SEARCHING	3
<i>Sorting</i>	3
<i>Searching</i>	4
METHODS	5
<i>Sorting</i>	5
<i>Add value in Table</i>	5
<i>Clear</i>	5
<i>Add</i>	6
<i>Exit</i>	6
<i>Search Price</i>	6
<i>Search Model</i>	7
<i>Help</i>	7
<i>Open</i>	7
TESTING	8
TEST 1 – RUN PROGRAM	8
TASK2.2-EVIDENCE OF SEARCHING FOR ITEMS BASED ON PRICE AND BIKE MODEL	11
1. <i>Searching items based on model</i> :.....	11
2. <i>Searching items based on price</i> :	13
<i>Task 3 - Evidence of opening a file from menu</i>	14
<i>Test 4 - Evidence of system validation</i>	16
CONCLUSION	20
APPENDIX:.....	21
BIBLIOGRAPHY:	22

Table of Tables

Table 1 Run Program	8
Table 2 adding items to table	9
Table 3 Searching items based on model	11
Table 4 Searching items based on price	13
Table 5 Open File	14

Table of Figures

Figure 1:Example: Binary Search (geeksforgeeks, n.d.)	3
Figure 1 Run porgram	8
Figure 2 Compile {gram.....	8
Figure 3 before adding items to table	9
Figure 4 after adding items to table	10
Figure 5 Before Searching items based on model	11
Figure 6 After Searching item on model	12
Figure 7 Before Searching items based on price	13
Figure 8 After Searching items based on price	13
Figure 10 Evidence of opening a file from menu	14
Figure 11 File Chooser.....	14
Figure 12 Locating File.....	15
Figure 13 File Opens.....	15
Figure 14 Name Validation.....	16
Figure 15 Address Validation	16
Figure 16 Phone number Validation	17
Figure 17 Bike Number 15 Validation.....	17
Figure 18 Kilometres Drove Validation	18
Figure 19 Chassis Number Validation	18
Figure 20 Last Servicing Validation	19
Figure 2:- Example: Linear Search (geeksforgeeks, n.d.)	21

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

CS5004NA

- 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)

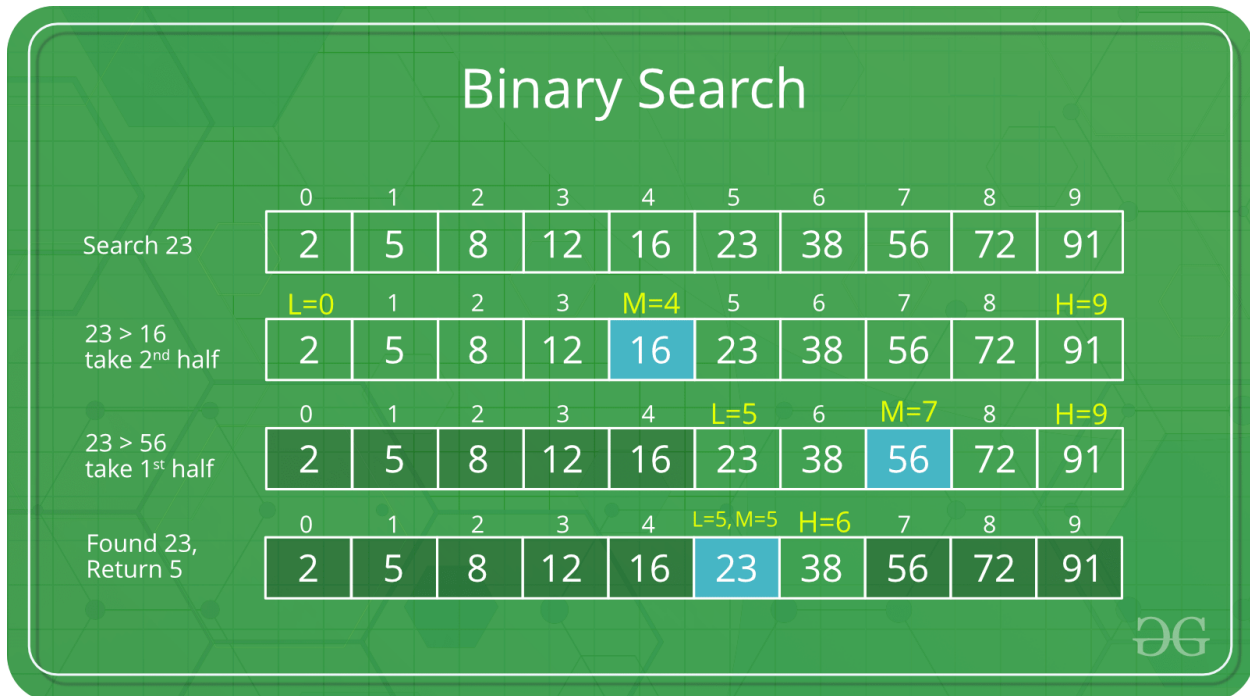


Figure 1: Example: Binary Search (geeksforgeeks, n.d.)

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 value 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 value 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 be sorted out. We have implemented linear search method in our project, to find data using model no. of a motorcycle. The model no. is stored in the array as `i=4 ; i<array.size(); i+=10`. The model no. to be found is then 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 opens up 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

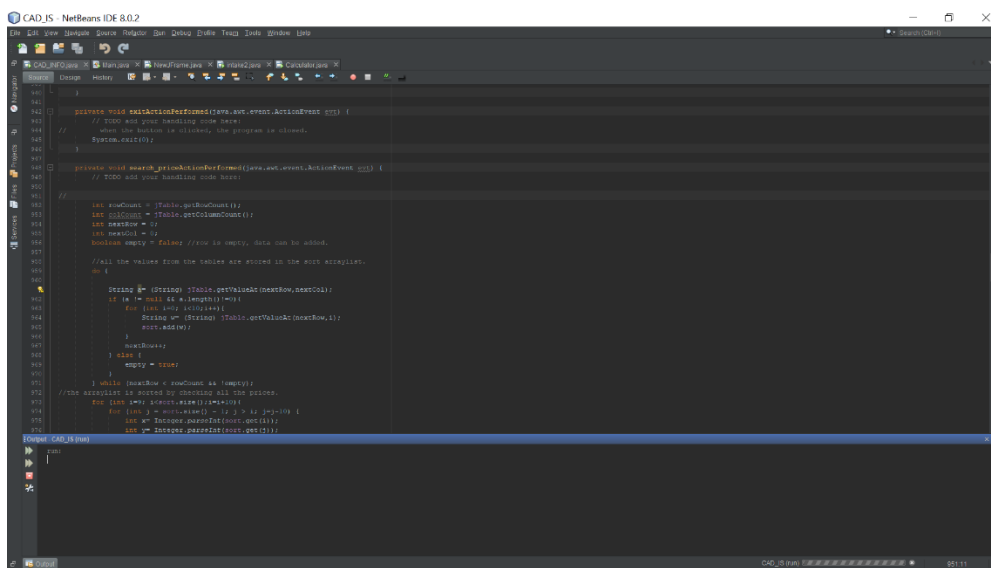


Figure 2 Run program

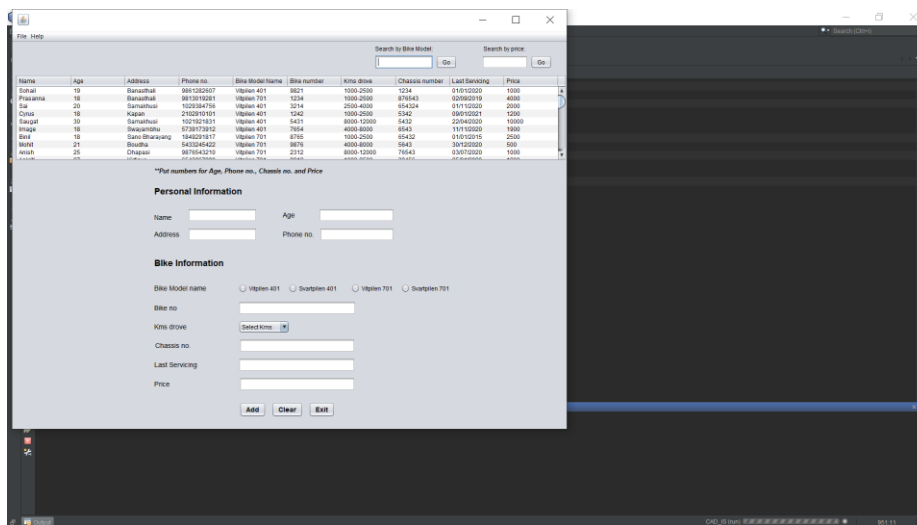


Figure 3 Compile Igram

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

Search by Bike Model: Go

Search by price: Go

Name	Age	Address	Phone no.	Bike Model Name	Bike number	Kms drove	Chassis number	Last Servicing	Price
Riju	26	Banesthor	9843267892	Vitpilen 401	3421	4000-8000	746235	19/01/2015	15000
Aashish	30	Banesthor	3456234233	Vitpilen 401	5790	1000-2500	7462	13/01/2016	2000

Put numbers for Age, Phone no., Chassis no. and Price

Personal Information

Name: Age:

Address: Phone no.:

Bike Information

Bike Model name: ☐ Vitpilen 401 ☐ Svarpilen 401 ☐ Vitpilen 751 ☒ Svarpilen 751

Bike no:

Kms drove:

Chassis no:

Last Servicing:

Price:

Figure 4 before adding items to table

File Help

Search by Bike Model: Go

Search by price: Go

Name	Age	Address	Phone no.	Bike Model Name	Bike number	Kms drove	Chassis number	Last Servicing	Price
Riju	29	Baneshwor	9843207892	Vitpilen 401	3421	4000-8000	746235	19/01/2015	15000
Aashish	30	Baneshwor	3456234233	Vitpilen 401	5799	1000-2500	7462	13/01/2016	2000
Sarrah	22	Thamel	98767754	Svartpilen 701	5415	1000-2500	124235254	19/02/2019	30000

Put numbers for Age, Phone no., Chassis no. and Price

Personal Information

Name Age

Address Phone no.

Bike Information

Bike Model name: ☐ Vitpilen 401 ☐ Svartpilen 401 ☐ Vitpilen 701 ☐ Svartpilen 701

Bike no

Kms drove

Chassis no

Last Servicing

Price

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

The screenshot shows a Java Swing application window with a title bar. Inside, there are two search buttons: 'Search by Bike Model' and 'Search by price', each with a 'Go' button next to it. Below these is a table with the following data:

Name	Age	Address	Phone no.	Bike Model Name	Bike number	Kms drove	Chassis number	Last Servicing	Price
Saugat	30	Samathusi	1921921931	Vipiten 401	5431	8000-12000	5432	2204/2020	10000
Image	18	Saragomhu	5739173912	Vipiten 401	7654	4000-8000	6543	1111/2020	1900
Bholi	18	Sano Bharayang	1849291817	Vipiten 701	8765	1000-2500	65432	0101/2015	2500
Mohit	21	Boudha	5433245422	Vipiten 701	9876	4000-8000	5643	3012/2020	500
Avish	25	Chapali	9876543210	Vipiten 701	2312	8000-12000	76543	0307/2020	1000
Aash	27	Kirtpur	6543267890	Vipiten 701	9812	1000-2500	32456	2501/2020	1200
Raja	28	Baneshwor	9843267892	Vipiten 401	3421	4000-8000	745235	1901/2015	15000
Aashish	30	Baneshwor	3456234533	Vipiten 401	5789	1000-2500	7462	1301/2016	2000
Saram	22	Thamel	98767754	Scaripiten 701	5415	1000-2500	124235254	1802/2019	30000

Below the table, there is a note: "**Put numbers for Age, Phone no., Chassis no. and Price".

The form is divided into two sections:

- Personal Information:** Fields for Name, Age, Address, and Phone no.
- Bike Information:** Fields for Bike Model name (with radio buttons for Vipiten 401, Scaripiten 401, Vipiten 701, and Scaripiten 701), Bike no., Kms drove (with a dropdown menu), Chassis no., Last Servicing, and Price.

At the bottom of the form are three buttons: 'Add', 'Clear', and 'Exit'.

Figure 6 Before Searching items based on model

The screenshot shows a software application window titled "File Help". It features a search bar at the top with "Sarpilen 701" entered. Below the search bar is a table listing bike records. A modal dialog box is open in the center, displaying the search results for the entered model.

Name	Age	Address	Phone no.	Bike Model Name	Bike number	Kms drove	Chassis number	Last Servicing	Price
Saugat	30	Samathusi	1021921831	Vipilen 401	5431	8000-12000	5432	22/04/2020	10000
Image	18	Sewamthi	5739173912	Vipilen 401	7654	4000-8000	5543	11/11/2020	1900
Bini	18	Sano Bharayang	1849291917	Vipilen 701	8765	1000-2500	55432	01/01/2015	2500
Mohit	21	Boudha	5433245422	Vipilen 701	9876	4000-8000	5543	30/12/2020	500
Kish	25	Chupasi	9876543210	Vipilen 701	2312	8000-12000	75543	03/07/2020	1000
Ashmi	27	Kiripur	6543207894	Vipilen 354	4567	4000-8000	32456	25/01/2020	1200
Rju	28	Baneshwor	984325781				745235	19/01/2015	15000
Aashish	30	Baneshwor	34562342				7452	13/01/2015	2000
Saram	22	Thamel	98767754				124235254	19/02/2019	30000

The modal dialog box displays the following information:

The result you have been searching is found:

- The name of the rider is: Saram
- Age is: 22
- Address is: Thamel
- Phone no. is: 98767754
- Bike model name is: Sarpilen 701
- Bike no. is: 5415
- Kms drove is: 1000-2500
- Chassis no. is: 124235254
- Last Servicing is: 19/02/2019
- Price is: 30000

The dialog box has an "OK" button. Below the dialog box, the application form shows the "Bike Model name" field with "Sarpilen 701" selected and a radio button next to it.

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 price
Action	The program was executed
Expected Result	Searched item should be displayed
Actual Report	Program runs
Conclusion	Test Successful

Search by Bike Model: Go

Search by price: 1900 Go

Name	Age	Address	Phone no.	Bike Model Name	Bike number	Kms drove	Chassis number	Last Servicing	Price
Saugat	30	Samathusi	1021821831	Vipiten 401	5431	8000-12000	5432	22/04/2020	10000
Image	18	Swamthusi	5739173912	Vipiten 401	7654	4000-8000	6543	11/11/2020	1900
Binil	19	Sano Bharayang	1849291817	Vipiten 701	8765	1000-2500	65432	01/11/2020	2500
Mohit	21	Boudha	543245422	Vipiten 701	9876	4000-8000	5643	30/12/2020	500
Anish	25	Chappasi	9876543210	Vipiten 701	2312	8000-12000	76543	03/07/2020	1000
AakrB	27	Kirtipur	6543287890	Vipiten 701	9812	1000-2500	32456	25/01/2020	1200
Riju	29	Baneshwor	0943207892	Vipiten 401	3421	4000-8000	746235	19/01/2015	15000
Aashish	30	Baneshwor	3456234233	Vipiten 401	5790	1000-2500	7462	13/01/2016	2000
Saram	22	Thamel	98767754	Svarptilen 701	5415	1000-2500	124235254	19/02/2019	30000

Put numbers for Age, Phone no., Chassis no. and Price

Personal Information

Name: Age:

Address: Phone no.:

Bike Information

Bike Model name: ☐ Vipiten 401 ☐ Svarptilen 401 ☒ Vipiten 701 ☐ Svarptilen 701

Bike no:

Kms drove: Select Kms

Chassis no.:

Last Servicing:

Price:

Add Clear Exit

Figure 8 Before Searching items based on price

Search by Bike Model: Go

Search by price: 2500 Go

Name	Age	Address	Phone no.	Bike Model Name	Bike number	Kms drove	Chassis number	Last Servicing	Price
Sohail	19	Banasthali	9981282007	Vipiten 401	9821	1000-2500	1234	01/01/2020	1000
Prasanna	18	Banasthali	9813816281	Vipiten 701	1234	1000-2500	876543	02/09/2019	4000
Sai	20	Samathusi	1029384756	Vipiten 401	3214	2500-4000	654324	01/11/2020	2000
Cyrus	18	Kapan	2102916101	Vipiten 401	1242	1000-2500	3342	09/01/2021	1200
Saugat	30	Samathusi	1021821831	Vipiten 401	5431	8000-12000	5432	22/04/2020	10000
Image	18	Swamthusi	5739173912	Vipiten 401	7654	4000-8000	6543	11/11/2020	1900
Binil	19	Sano Bharayang	1849291817	Vipiten 701	8765	1000-2500	65432	01/11/2020	2500
Mohit	21	Boudha	543245422	Vipiten 701	9876	4000-8000	5643	30/12/2020	500
Anish	25	Chappasi	9876543210	Vipiten 701	2312	8000-12000	76543	03/07/2020	1000

Put numbers for Age, Phone no., Chassis no. and Price

Personal Information

Name: Age:

Address: Phone no.:

Bike Information

Bike Model name: ☐ Vipiten 401 ☐ Svarptilen 401 ☒ Vipiten 701 ☐ Svarptilen 701

Bike no:

Kms drove: Select Kms

Chassis no.:

Last Servicing:

Price:

Add Clear Exit

Message

The result you have been searching is found:

The name of the rider is: Binil
 Age is: 19
 Address is: Sano Bharayang
 Phone no. is: 1849291817
 Bike model name is: Vipiten 701
 Bike no. is: 8765
 Kms drove is: 1000-2500
 Chassis no. is: 65432
 Last Servicing is: 01/11/2020
 Price is: 2500

OK

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

File Help

Open Ctrl+O
Exit Ctrl+E

Search by Bike Model: Go Search by price: 2500 Go

Name	Age	Address	Phone no.	Bike Model Name	Bike number	Kms drove	Chassis number	Last Servicing	Price
Sohail Dangol	19	Banasthali	9881282607	Vitpilen 401	1234	1000-2500	123	01/01/2020	1000
Prasanna Dangol	20	Banasthali	9880157734	Vitpilen 701	4312	1000-2500	4312	02/02/2020	3000

Put numbers for Age, Phone no., Chassis no. and Price

Personal Information

Name Age
Address Phone no.

Bike Information

Bike Model name: ☐ Vitpilen 401 ☐ Svarptilen 401 ☒ Vitpilen 701 ☐ Svarptilen 701
Bike no.
Kms drove:
Chassis no.
Last Servicing
Price

Add Clear Exit

Figure 10 Evidence of opening a file from menu

File Help

Open Ctrl+O
Exit Ctrl+E

Search by Bike Model: Go Search by price: 2500 Go

Name	Age	Address	Phone no.	Bike Model Name	Bike number	Kms drove	Chassis number	Last Servicing	Price
Sohail	19	Banasthali	9881282607	Vitpilen 401	9821	1000-2500	1234	01/01/2020	1000
Prasanna	18	Banasthali	9813019281	Vitpilen 701	1234	1000-2500	878543	02/09/2019	4000
Sai	20	Samasthali	1020384756	Vitpilen 401	3214	2500-4000	654324	01/11/2020	2000
Onus	18	Kapan	2102919101	Vitpilen 401	1242	1000-2500	5342	09/01/2021	1200
Gaugal	20	Samasthali	1021921831	Vitpilen 401	5431	8000-12000	5432	22/04/2020	10000
Ompr	19	Sarvashahi	5233121312	Vitpilen 401	1124	4000-12000	6543	11/01/2020	1000
Bini	18	Sano Bharavang	1849291817	Vitpilen 701	8765	1000-2500	65432	01/01/2015	2500
Mohit	21						5543	30/12/2020	500
Anish	25						76543	03/07/2020	1000

Open

Look in: Desktop

- 1809901_SOHAIL_DANGOL
- bootstrap-4.4.1
- CWD_IS
- database
- file
- gam
- java
- software
- travello
- virtualenv

File name:

Files of type: All Files

Open Cancel

Kms drove:
Chassis no.
Last Servicing
Price

Add Clear Exit

Figure 11 File Chooser

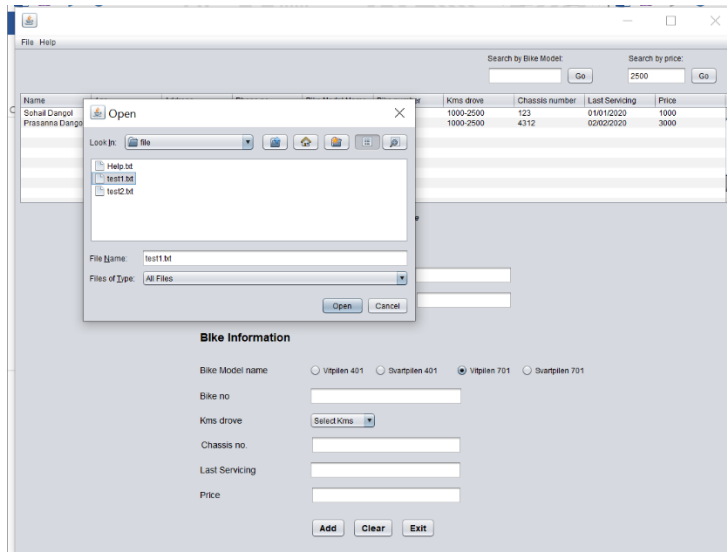


Figure 12 Locating File

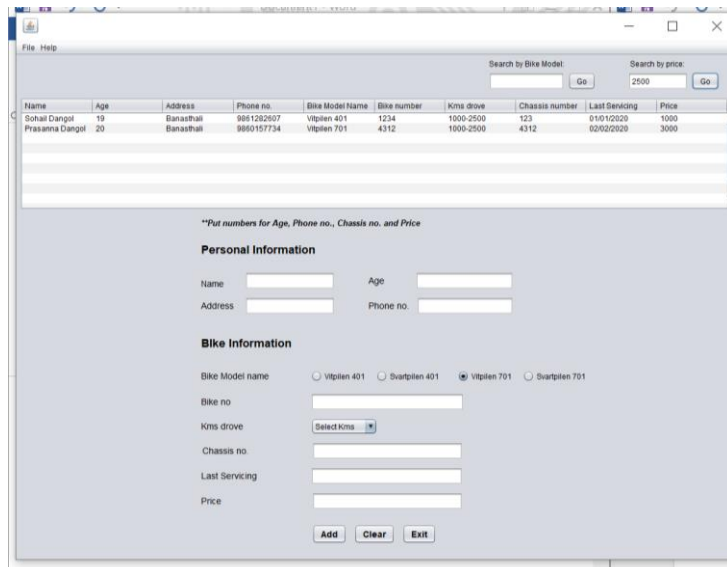


Figure 13 File Opens

Test 4 - Evidence of system validation

Search by Bike Model: Go

Search by price: Go

Name	Age	Address	Phone no.	Bike Model Name	Bike number	Kms drove	Chassis number	Last Servicing	Price
Sohail	19	Banasthali	9861282607	Vitpilen 401	9821	1000-2500	1234	01/01/2020	1000
Praiana	18	Banasthali	9813016281	Vitpilen 701	1234	1000-2500	876543	02/09/2019	4000
Sai	20	Samasthali	1020384756	Vitpilen 401	3214	2500-4000	654324	01/11/2020	2000
Cyrus	18	Kapan	2102910101	Vitpilen 401	1242	1000-2500	5342	09/01/2021	1200
Saugat	30	Samasthali	1021921831	Vitpilen 401	5431	8000-12000	5432	22/04/2020	10000
Image	18	Swayambhu	5739173912	Vitpilen 401	7854	4000-8000	6543	11/11/2020	1900
Bini	18	Sano Bharayang	1849291817	Vitpilen 701	8765	1000-2500	65432	01/01/2015	2500
Mohit	21	Boudha	5433245422	Vitpilen 701	9876	4000-8000	6543	30/12/2020	500
Anish	25	Chapasi	9876543210	Vitpilen 701	2312	8000-12000	76543	03/07/2020	1000

Put numbers for Age, Phone no., Chassis no. and Price

Personal Information

Give name
 Name Age
 Address Phone no.

Bike Information

Bike Model name ☐ Vitpilen 401 ☐ Svarpilen 401 ☐ Vitpilen 701 ☐ Svarpilen 701

Bike no

Kms drove

Chassis no

Last Servicing

Price

Figure 14 Name Validation

Search by Bike Model: Go

Search by price: Go

Name	Age	Address	Phone no.	Bike Model Name	Bike number	Kms drove	Chassis number	Last Servicing	Price
Sohail Dangol	19	Banasthali	9861282607	Vitpilen 401	1234	1000-2500	123	01/01/2020	1000
Praiana Dangol	20	Banasthali	9860157734	Vitpilen 701	4312	1000-2500	4312	02/02/2020	3000

Put numbers for Age, Phone no., Chassis no. and Price

Personal Information

Name Age
 Give address
 Address Phone no.

Bike Information

Bike Model name ☐ Vitpilen 401 ☐ Svarpilen 401 ☒ Vitpilen 701 ☐ Svarpilen 701

Bike no

Kms drove

Chassis no

Last Servicing

Price

Figure 15 Address Validation

Search by Bike Model: Go

Search by price: 250 Go

Name	Age	Address	Phone no.	Bike Model Name	Bike number	Kms drove	Chassis number	Last Servicing	Price
Sohai Dangol	19	Banasthali	9861262607	Vitpilen 401	1234	1000-2500	123	01/01/2020	1000
Prasanna Dangol	20	Banasthali	9860157734	Vitpilen 701	4312	1000-2500	4312	02/02/2020	3000

****Put numbers for Age, Phone no., Chassis no. and Price**

Personal Information

Name: sai Age: 20

Address: Thamel Phone no.:

Bike Information

Bike Model name: ☐ Vitpilen 401 ☐ Svarpilen 401 ☒ Vitpilen 701 ☐ Svarpilen 701

Bike no:

Kms drove: Select Kms

Chassis no:

Last Servicing:

Price:

Figure 16 Phone number Validation

Search by Bike Model: Go

Search by price: 250 Go

Name	Age	Address	Phone no.	Bike Model Name	Bike number	Kms drove	Chassis number	Last Servicing	Price
Sohai Dangol	19	Banasthali	9861262607	Vitpilen 401	1234	1000-2500	123	01/01/2020	1000
Prasanna Dangol	20	Banasthali	9860157734	Vitpilen 701	4312	1000-2500	4312	02/02/2020	3000

****Put numbers for Age, Phone no., Chassis no. and Price**

Personal Information

Name: sai Age: 20

Address: Thamel Phone no.: 9861967556

Bike Information

Bike Model name: ☐ Vitpilen 401 ☐ Svarpilen 401 ☒ Vitpilen 701 ☐ Svarpilen 701

Bike no:

Kms drove: Select Kms

Chassis no:

Last Servicing:

Price:

Figure 17 Bike Number 15 Validation

File Help

Search by Bike Model: Go

Search by price: 250 Go

Name	Age	Address	Phone no.	Bike Model Name	Bike number	Kms drove	Chassis number	Last Servicing	Price
Sohal Dangol	19	Banasthali	9881282607	Vitpilen 401	1234	1000-2500	123	01/01/2020	1000
Prasanna Dangol	20	Banasthali	9880157734	Vitpilen 701	4312	1000-2500	4312	02/02/2020	3000

Put numbers for Age, Phone no., Chassis no. and Price

Personal Information

Name: sai Age: 20

Address: Thamel Phone no.: 9881967556

Bike Information

Bike Model name: ☐ Vitpilen 401 ☐ Svarpilen 401 ☒ Vitpilen 701 ☐ Svarpilen 701

Bike no: 5416

Select Kms drove: Select Kms

Kms drove:

Chassis no:

Last Servicing:

Price:

Figure 18 Kilometres Drove Validation

File Help

Search by Bike Model: Go

Search by price: 250 Go

Name	Age	Address	Phone no.	Bike Model Name	Bike number	Kms drove	Chassis number	Last Servicing	Price
Sohal Dangol	19	Banasthali	9881282607	Vitpilen 401	1234	1000-2500	123	01/01/2020	1000
Prasanna Dangol	20	Banasthali	9880157734	Vitpilen 701	4312	1000-2500	4312	02/02/2020	3000

Put numbers for Age, Phone no., Chassis no. and Price

Personal Information

Name: sai Age: 20

Address: Thamel Phone no.: 9881967556

Bike Information

Bike Model name: ☐ Vitpilen 401 ☐ Svarpilen 401 ☒ Vitpilen 701 ☐ Svarpilen 701

Bike no: 5416

Kms drove: 2500-4000

Give Chassis number:

Chassis no:

Last Servicing:

Price:

Figure 19 Chassis Number Validation

Search by Bike Model: Go

Search by price: 250 Go

Name	Age	Address	Phone no.	Bike Model Name	Bike number	Kms drove	Chassis number	Last Servicing	Price
Sohai Dangol	19	Banasthali	9861262607	Vitpilen 401	1224	1000-2500	123	01/01/2020	1000
Prasanna Dangol	20	Banasthali	9860157734	Vitpilen 701	4312	1000-2500	4312	02/02/2020	3000

****Put numbers for Age, Phone no., Chassis no. and Price**

Personal Information

Name: sai Age: 20

Address: Thamel Phone no.: 9861967556

Bike Information

Bike Model name: ☐ Vitpilen 401 ☐ Svarpilen 401 ☒ Vitpilen 701 ☐ Svarpilen 701

Bike no: 5416

Kms drove: 2500-4000

Chassis no.: 567567

Last Servicing:

Price:

Give Last Servicing Date

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:

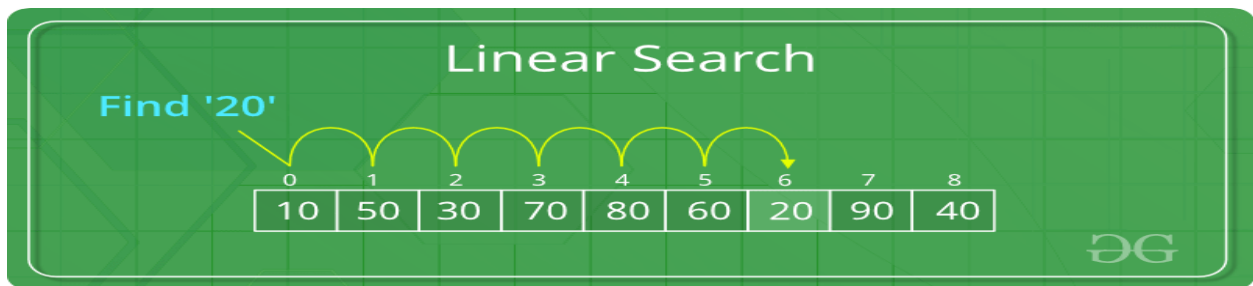


Figure 21:- Example: Linear Search (geeksforgeeks, n.d.)

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].