

# **JAVA PROGRAMMING**

# **PROPOSAL**

# **ON**

# **C&M MINING MACHINES RESERVATION SYSTEM**

# **BY**

**STUDENT NAME: KHALIL MUHAMMAD SANI**

**STUDENT ID: 2308-1903-0010**

**COURSE NAME: JAVA**

**COURSE CODE: IT 209**

**LECTURER’S NAME: MR. NAZIFI MUSA**

**GROUP: A**

|  |
| --- |
| **IT209 Marking Scheme** |

Official Note:

**Diploma in Information Technology**

|  |  |
| --- | --- |
| Student Id: 2308-1903-0010 | Centre Code: |
| Student Name: KHALIL MUHAMMAD SANI | Marker’s Code: |
| Class Code: | Project Due Date: |
| Date Submitted: | Country: NIGERIA |

**IT209 - Java Programming**

|  |  |  |
| --- | --- | --- |
| **Description** | **Max.**  **Marks** | **Total Marks** |
| Submit project proposal | 10 |  |
|  |  |  |
| **1st Submission Date(L6)** |  |  |
| **A. Analysis & Design** | **21** |  |
| Introduction of Current System | 3 |  |
| Introduction and objective of Proposed System | 3 |  |
| Screen Design (Input and Output Screen Design) | 5 |  |
| Program Design – Flowchart | 10 |  |
|  |  |  |
| **2nd Submission Date (L10)** |  |  |
| **B. Development and Testing** | **28** |  |
| Meaning identifier names, comments, indentation | 5 |  |
| User-defined methods, arguments, methods returning values | 5 |  |
| Workable program (error free) | 3 |  |
| Test Plan, Test Log, Conclusion for Test Cases | 5 |  |
| Test Cases and result (expected and actual results) | 10 |  |
|  |  |  |
| **3rd Submission Date(L12)** |  |  |
| **C. Implementation** | **15** |  |
| System Configuration (H/W and S/W requirements) | 5 |  |
| User Manual (with screen captures) | 5 |  |
| Conclusion (Strength, Weakness, Enhancement) | 5 |  |
|  |  |  |
| **D. Documentation** | **16** |  |
| Appendices/Reference | 4 |  |
| Page Number, Table of Contents, Cover Page | 4 |  |
| Completeness of project | 4 |  |
| Project Layout (neatness, header, footer etc.) | 4 |  |
|  |  |  |
| **E. Presentation** | **10** |  |
| Student Performance (attendance, participation in class etc.) | 5 |  |
| Presentation skills (question handling, eye contact, punctuality etc.) | 5 |  |
|  |  |  |
| **Maximum 100 marks only** |  |  |

IT209 Marking Scheme Page 1 of 1

Table of Contents

[**JAVA PROGRAMMING** 1](#_Toc66497778)

[**PROPOSAL** 1](#_Toc66497779)

[**ON** 1](#_Toc66497780)

[**C&M MINING MACHINES RESERVATION SYSTEM** 1](#_Toc66497781)

[**BY** 1](#_Toc66497782)

[INTRODUCTION 4](#_Toc66497783)

[CURRENT SYSTEM 5](#_Toc66497784)

[PROPOSE NEW SYSTEM 6](#_Toc66497785)

[AIMS AND OBJECTIVES 7](#_Toc66497786)

[FLOW CHAT 8](#_Toc66497787)

[SCREEN DESIGN 12](#_Toc66497788)

[SCREEN DESIGN 13](#_Toc66497789)

[PROGRAM LISTING 18](#_Toc66497790)

[**PROGRAM TESTING** 35](#_Toc66497791)

[TEST RESULTS 36](#_Toc66497792)

[TEST LOG 49](#_Toc66497793)

[HARDWARE AND SOFTWARE REQUIREMENT**.** 50](#_Toc66497794)

[**Hardware Requirement** 50](#_Toc66497795)

[Software Requirement 51](#_Toc66497796)

[USER MANUAL 51](#_Toc66497797)

[The first step of user manual is 51](#_Toc66497798)

[Like this C&M MINING MACHINES RESERVATION SYSTEM 51](#_Toc66497799)

[Enter 1 to add customer record 52](#_Toc66497800)

[CONCLUSION: PROGRAM STRENGHT 55](#_Toc66497801)

[PROGRAM WEAKNESS56](#_Toc66497802)

[PROGRAM ENHANCEMENT56](#_Toc66497803)

# INTRODUCTION

C&M Mining Machine which primarily produce enormous and medium-sized pounding and granulating types of gear was established in 1987. It is a cutting-edge joint-stock company with examination, assembling and deals together. The Settle is situated in Howdy TECH Industry Improvement Zone of Zhengzhou and covers 80000m². Another workshop in Shangjie Industry Park covers 67000 m². Over the over 30 years, our organization holds fast to current logical administration framework, exactness fabricating, spearheading and advancement. Presently C&M Mining Machine has become the pioneer in homegrown and oversea hardware fabricating industry.

# CURRENT SYSTEM

C&M Mining is slightly obsolete for the time being, and the administration have sat on a few events to examine about how to improve it. As of now, it utilizes handbook framework which is all the data are continued utilizing nearby document preparing framework which implies they are kept in cabinets in a protected record room in the administrator block. In this framework each staff has his own data controlled in the record, the data covers Staff ID, Staff Name, Date of Birth, Address, Date of Confirmation, Parent/Watchman data and so forth There are a few issues the eatery has been looking because of obsolete thoughts. Furthermore, a portion of the serious issues is that, the framework depends on manual activity which made the activity exceptionally moderate for example at whatever point data is needed about any staff or client, the record official need to look for the document number in the file organizers for the people subtleties which burns-through a ton of time contrasted with the most current of present day thoughts which defers the cycle. When there is need for any change in a record in any record, the past record must be eradicated totally, additionally the standard looking of the documents may prompt loss of some report, and the most concerning issue is that the documents are not all around secured, which means it can undoubtedly be harmed by both cataclysmic events, for example, flood and tremor just as man-made calamities, for example, fire flare-up which can prompt missing of fundamental data that can make issue to the cafés notoriety as a rule. So, the association have met as often as possible these days on how they will conquer the above issues by refreshing their framework.

# PROPOSE NEW SYSTEM

To take care of the issues confronting the old framework, another framework refreshed framework should be utilized. C&M Mining are at present utilize manual record preparing framework to automate information base framework. The new framework will take over with all the issues looked in the past framework. Information and data are protected, additionally preparing of the record will be substantially more simpler contrasted with the document handling framework since there are a few modules that characterize the different capacities utilized in the program, if any staffs to be added, there is a capacity that speak to the add module in the primary menu, what is required is for the client to enter his/her decision, likewise information will be put away together which will significantly decrease information terminating, getting data is more simpler contrast with the record handling framework, since at whatever point data is required, the individual's chronic number will be entered and the data will be shown once. In addition, information is more defensive in the data set framework in light of the fact that the information can't be harmed or missed set, there is constancy in this arrangement of record keeping. The new framework they need presents or execute is automating enlistment framework the program.

[1] ADD RECORD: This capacity permits the client to add request subtleties like client Id, client name, client sexual orientation, client telephone, submit of conveyance and the requests.

[2] VIEW ALL RECORD: This capacity will list all the records in the framework.

[3] SEARCH RECORDS: This capacity permits the client to look for the ideal record when the client id is called upon.

[4] DELETE RECORD: This page will let the client erase any ideal subtleties that exists in the framework.

[5] EDIT RECORD: choice 6 was chosen, the program will close and all subtleties are saved into the framework.

# AIMS AND OBJECTIVES

The reason for this task is to examine the current framework on which C&M Mining is working and see the framework and issues they confronting, with the goal that it tends to be defeats at the earliest opportunity for greatest consumer loyalty. Bringing another framework is the primary objective of the association which contains the utilization of new electrical and mechanical gadgets for the capacity of imperative data. Their subsequent objective is to back up in various outer drives to forestall loss of information and increment security. Also, propose the electronic arrangement of recording information to beat the issues of putting away the record and copying. This new framework gives conceivable arrangement of our present framework.

# 

# FLOW CHAT

**START**

Input: username

And password

Process data

Verify username/password

?

**Wrong password**

**Wow**

**Correct password**

**MENU**

1. Add record
2. View record
3. Search record
4. Delete record

5. Exit record

ENTER OPTION

e

Input: add Serial number

add record

?

Input: customer

Gender

Process choice

**yes**

Input: customer

day of delivery

Input: customer time of delivery

View All record

?

**yes**

Output: display all records

Process choice

**no**

Search record

?

Input: enter id

**yes**

Process choice

Output: display record

Found

?

**yes**

**no**

**no**

Output: record not found

Delete record

?

**yes**

Process choice

If found

Record not found

Input: enter id

To delete

**no**

**no**

**yes**

Output: record deleted

Delete record

Exit

?

**yes**

**STOP**

# SCREEN DESIGN

MENU

Add record

Exit

Delete record

Search record

View All record

“The Menu shows the Key functions of the program:

1. To Add Customer Records
2. To View Customer Records
3. To Search Customer Records
4. To Update Customer Records
5. To Delete Customer Records
6. To Calculate Customer Items
7. To Exit the Program

# SCREEN DESIGN

* **LOGIN**

--------------------Configuration: <Default>--------------------

Username: MUHAMMAD

Password: KHALIL

Logged In Successfully.

* **FUNCTIONS (RECORDS)**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*#

C&M MINING MACHINES RESERVATION SYSTEM

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. To Add Customer Records

2. To View Customer Records

3. To Search Customer Records

4. To Update Customer Records

5. To Delete Customer Records

6. To Calculate Customer Items

7. To Exit the Program

* **CUSTOMER RECORD**

Enter Your Choice [1-7]: 1

--------------------------------------

ADDING CUSTOMER RECORD

--------------------------------------

Enter the Customer ID:

* **DISPLAYING CUSTOMER RECORDS**

Enter Your Choice [1-7]: 2

------------------------------------

DISPLAYING CUSTOMER RECORDS

--------------------------------------

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*---------------------

| ID Name Item Name Item Quantity Item Price

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*---------------------

|

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*---------------------

Total number of Item(s) is 0

* **SEARCHING CUSTOMER RECORDS**

Enter Your Choice [1-7]: 3

--------------------------------------

SEARCHING CUSTOMER RECORDS

---------------------------------------

Enter the CUSTOMER ID:

* **UPDATING CUSTOMER RECORD**

Enter Your Choice [1-7]: 4

---------------------------------------

UPDATING CUSTOMER RECORD

---------------------------------------

Enter the CUSTOMER ID:

* **DELETING CUSTOMER RECORD**

Enter Your Choice [1-7]: 5

-------------------------------------

DELETING CUSTOMER RECORD

-------------------------------------

Enter the Customer ID:

* **CALCULATE CUSTOMER ITEM**

Enter Your Choice [1-7]: 6

----------------------------------------

CALCULATE CUSTOMER ITEM

----------------------------------------

Enter the Item Price:

* **EXITING THE PROGRAM**

Enter Your Choice [1-7]: 7

----------------------------------

CLOSING THE PROGRAM

----------------------------------

THANK YOU FOR PATRONIZING US!!!

Designed By:KHALIL MUHAMMAD SANI

With ID NO: 2308 - 1903 - 0010

Supervised By: MR NAZIFI MUSA

WE ARE HONOURED TO SERVE YOU!!!

Process completed.

# 

# PROGRAM LISTING

import java.io.\*;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.Paths;

import static java.nio.file.StandardOpenOption.CREATE;

import java.util.Scanner;

import java.util.StringTokenizer;

public class KHALIL {

public static void main(String[] args) throws Exception {

KHALIL project = new KHALIL(); // Create new MainProj Object.

Path file = Paths.get("B:\\2308-1903-0010\\IT209\_2308-1903-0010\\KHALIL.txt");

String s = "";

project.loginPage();

if(!Files.exists(file)){

project.createEmptyFile(file, s);

project.menu();

}

else

project.menu();

}

public void createEmptyFile(Path file, String s)

{ final int NUMRECS = 1000;

try

{

OutputStream outputStr = new BufferedOutputStream(Files.newOutputStream(file,CREATE));

BufferedWriter writer = new BufferedWriter(new OutputStreamWriter(outputStr));

for(int count = 0; count < NUMRECS; ++count)

writer.write(s, 0, s.length());

writer.close();

}

catch(Exception e)

{

System.out.println("Error message: " + e);

}

}

public void loginPage()throws Exception{

BufferedReader input = new BufferedReader(new InputStreamReader(System.in));

String username;

String password;

do {System.out.print("\nUsername: ");

username = input.readLine();

System.out.print("\nPassword: ");

password = input.readLine();

if (!(password.equalsIgnoreCase("KHALIL") && username.equalsIgnoreCase("MUHAMMAD"))){

System.out.print("\nInvalid Entry, Try Again\n");

System.out.println();

}

} while(!(password.equalsIgnoreCase("KHALIL") && username.equalsIgnoreCase("MUHAMMAD")));

System.out.println("\nLogged In Successfully.\n");

System.out.println();

System.out.println();

}

public void menu()throws Exception // declaring a menu method

{

BufferedReader input = new BufferedReader (new InputStreamReader(System.in));

int choice;

do {

System.out.println("\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*#");

System.out.println("\t\t\t C&M MINING MACHINES RESERVATION SYSTEM ");

System.out.println("\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("\n");

System.out.println("\n\t\t 1. To Add Customer Records");

System.out.println("\n\t\t 2. To View Customer Records");

System.out.println("\n\t\t 3. To Search Customer Records");

System.out.println("\n\t\t 4. To Update Customer Records");

System.out.println("\n\t\t 5. To Delete Customer Records");

System.out.println("\n\t\t 6. To Calculate Customer Items");

System.out.println("\n\t\t 7. To Exit the Program");

System.out.print("\n\n\t\t Enter Your Choice [1-7]: ");

choice=Integer.parseInt(input.readLine());

switch(choice)

{

case 1:AddRecord();

break;

case 2:ViewAllRecord();

break;

case 3:SearchRecordbyID();

break;

case 4:updateRecordbyID();

break;

case 5:DeleteRecordByID();

break;

case 6:calculateCUSTOMERPrice();

break;

case 7:Exit();

break;

default :System.out.print("\n Invalid Choice ");

}

}while(choice ==1);

}

public void AddRecord() throws IOException {

BufferedWriter fs = new BufferedWriter( new FileWriter("B:\\2308-1903-0010\\IT209\_2308-1903-0010\\CUSTOMER.txt",true) );

Scanner strInput = new Scanner(System.in);

String fstech;

String ID, name, avail, quant, prize;

final String QUIT = "BACK";

System.out.println("\n\t\t--------------------------------------");

System.out.println("\t\t\t ADDING CUSTOMER RECORD");

System.out.println("\t\t--------------------------------------");

System.out.print("\nEnter the Customer ID: ");

ID = strInput.nextLine();

while(!(ID.equals(QUIT)||!ID.matches("[0-9]+")))

{

System.out.print("\nEnter the Customer Name: ");

name = strInput.nextLine();

if (!name.matches("[a-z]+")) {

System.out.println("Customer Name is Wrong!\n");

}

else {

System.out.print("\nEnter the Item Name: ");

avail = strInput.nextLine();

if (!avail.matches("[a-z]+")) {

System.out.println("Item Name is Wrong!\n");

}

else {

System.out.print("\nEnter the Item Quantity: ");

quant = strInput.nextLine();

if (!quant.matches("[0-9]+")) {

System.out.println("Item Quantity is Wrong!\n");

}

else {

System.out.print("\nEnter the Item Price: ");

prize = strInput.nextLine();

if (!prize.matches("[0-9]+")) {

System.out.println("Item Price is Wrong!\n");

}

else {

fs.write(ID+","+name+","+avail+","+quant+","+prize);

fs.flush();

fs.newLine();

}

}

}

}

System.out.print("\nEnter Next Customer ID or " + QUIT + " to Quit >> \n");

fstech = strInput.nextLine();

ID = fstech.toUpperCase();

}

fs.close();

}

public void ViewAllRecord() throws IOException {

BufferedReader fs = new BufferedReader( new FileReader("B:\\2308-1903-0010\\IT209\_2308-1903-0010\\CUSTOMER.txt") );

String record;

int total = 0;

System.out.println("\n\t\t------------------------------------");

System.out.println("\t\t\t DISPLAYING CUSTOMER RECORDS");

System.out.println("\t\t--------------------------------------");

System.out.println(" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*---------------------");

System.out.println("| \t ID \t Name \t\t Item Name \t\t Item Quantity \t Item Price ");

System.out.println(" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*---------------------");

while( ( record = fs.readLine() ) != null ) {

StringTokenizer st = new StringTokenizer(record,",");

System.out.println("| \t "+st.nextToken()+" \t "+st.nextToken()+"\t\t "+st.nextToken()+" \t\t\t "+st.nextToken()+" \t\t "+st.nextToken()+"\t");

total += 1;

}

System.out.println("|\t\t\t\t\t\t\t\t\t\t");

System.out.println(" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*---------------------");

System.out.println("Total number of Item(s) is " + total);

fs.close();

}

public void SearchRecordbyID() throws IOException {

String ID,record;

Scanner strInput = new Scanner(System.in);

BufferedReader fs = new BufferedReader( new FileReader("B:\\2308-1903-0010\\IT209\_2308-1903-0010\\CUSTOMER.txt") );

System.out.println("\n\t\t--------------------------------------");

System.out.println("\t\t\t SEARCHING CUSTOMER RECORDS");

System.out.println("\t\t---------------------------------------");

System.out.println("Enter the CUSTOMER ID: ");

ID = strInput.nextLine();

System.out.println(" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*------------");

System.out.println("| \t ID \t Name \t Item Name \t Item Quantity Item Price ");

System.out.println(" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*------------");

while( ( record = fs.readLine() ) != null ) {

StringTokenizer st = new StringTokenizer(record,",");

if( record.contains(ID) ) {

System.out.println("| \t "+st.nextToken()+" \t "+st.nextToken()+"\t "+st.nextToken()+" \t\t "+st.nextToken()+"\t\t "+st.nextToken()+"\t");

}

}

System.out.println("|\t\t\t\t\t\t\t\t\t\t|");

System.out.println(" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-------------");

fs.close();

}

public void updateRecordbyID() throws IOException {

String newName, newAvail, newQuant, newPrize, record, ID,record2;

File db = new File("B:\\2308-1903-0010\\IT209\_2308-1903-0010\\CUSTOMER.txt");

File tempDB = new File("B:\\2308-1903-0010\\IT209\_2308-1903-0010\\CUSTOMER\_new\_temp.txt");

BufferedReader br = new BufferedReader( new FileReader(db) );

BufferedWriter fs = new BufferedWriter( new FileWriter(tempDB) );

Scanner strInput = new Scanner(System.in);

System.out.println("\n\t\t---------------------------------------");

System.out.println("\t\t\t UPDATING CUSTOMER RECORD");

System.out.println("\t\t---------------------------------------");

/\*\*\*\*/

System.out.println("Enter the CUSTOMER ID: ");

ID = strInput.nextLine();

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*--------------------");

System.out.println("| \t ID \t Name \t\t Item Name \t\t Item Quantity \t Item Price ");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*--------------------");

while( ( record = br.readLine() ) != null ) {

StringTokenizer st = new StringTokenizer(record,",");

if( record.contains(ID) ) {

System.out.println("| \t "+st.nextToken()+" \t "+st.nextToken()+"\t\t "+st.nextToken()+" \t\t "+st.nextToken()+" \t\t "+st.nextToken()+"\t");

}

}

System.out.println("|\t\t\t\t\t\t\t\t\t\t|");

System.out.println(" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-------------------");

br.close();

/\*\*\*\*/

System.out.println("\nEnter New Customer Name: ");

newName = strInput.nextLine();

if (!newName.matches("[a-z]+")) {

System.out.println("New Customer Name is Wrong!\n");

}

else {

System.out.println("\nEnter New Item Name: ");

newAvail = strInput.nextLine();

if (!newAvail.matches("[a-z]+")) {

System.out.println("New Item Name is Wrong!\n");

}

else {

System.out.println("\nEnter New Item Quantity: ");

newQuant = strInput.nextLine();

if (!newQuant.matches("[0-9]+")) {

System.out.println("New Item Quantity is Wrong!\n");

}

else {

System.out.println("\nEnter New Item Price: ");

newPrize = strInput.nextLine();

if (!newPrize.matches("[0-9]+")) {

System.out.println("New Item Price is Wrong!\n");

}

else {

BufferedReader br2 = new BufferedReader( new FileReader(db) );

while( (record2 = br2.readLine() ) != null ) {

if(record2.contains(ID)) {

fs.write(ID+","+newName+","+newAvail+","+newQuant+","+newPrize);

} else {

fs.write(record2);

}

fs.flush();

fs.newLine();

}

fs.close();

br2.close();

db.delete();

boolean success = tempDB.renameTo(db);

System.out.println(success);

}

}

}

}

}

public void DeleteRecordByID() throws IOException {

Scanner strInput = new Scanner(System.in);

String ID, record;

File tempDB = new File("B:\\2308-1903-0010\\IT209\_2308-1903-0010\\CUSTOMER\_new\_temp.txt");

File db = new File("B:\\2308-1903-0010\\IT209\_2308-1903-0010\\CUSTOMER.txt");

BufferedReader br = new BufferedReader( new FileReader( db ) );

BufferedWriter fs = new BufferedWriter( new FileWriter( tempDB ) );

System.out.println("\n\t\t-------------------------------------");

System.out.println("\t\t\t DELETING CUSTOMER RECORD");

System.out.println("\t\t-------------------------------------");

System.out.println("Enter the Customer ID: ");

ID = strInput.nextLine();

while( ( record = br.readLine() ) != null ) {

if( record.contains(ID) )

continue;

fs.write(record);

fs.flush();

fs.newLine();

}

br.close();

fs.close();

db.delete();

tempDB.renameTo(db);

System.out.print("Record removed successfully!!");

}

public void calculateCUSTOMERPrice()throws Exception { // declaring a method to perform calculation

BufferedReader input = new BufferedReader (new InputStreamReader(System.in));

double quantity;

System.out.println();

System.out.println("\n\t\t\t----------------------------------------");

System.out.println("\t\t\t\t CALCULATE CUSTOMER ITEM");

System.out.println("\t\t\t----------------------------------------");

System.out.println();

System.out.print("\nEnter the Item Price: ");

double price=Double.parseDouble(input.readLine());

System.out.print("\nEnter the Amount Of Items: ");

quantity=Double.parseDouble(input.readLine());

double total = price \* quantity;

System.out.println("\nThe Total Price is: $"+ total);

}

public void Exit()throws Exception { // declaring a method to perform calculation

System.out.println("\n\t\t\t\t\t----------------------------------");

System.out.println("\t\t\t\t\t\tCLOSING THE PROGRAM");

System.out.println("\t\t\t\t\t----------------------------------");

System.out.println("\t\t\t\t\tTHANK YOU FOR PATRONIZING US!!!\n");

System.out.print("\t\t\t\tDesigned By:KHALIL MUHAMMAD SANI\n");

System.out.print("\t\t\t\tWith ID NO: 2308 - 1903 - 0010\n");

System.out.println("\t\t\t\tSupervised By: MR NAZIFI MUSA\n");

System.out.println("\t\t\t\t\t WE ARE HONOURED TO SERVE YOU!!!\n");

}

}

# **PROGRAM TESTING**

**TEST PLAN**

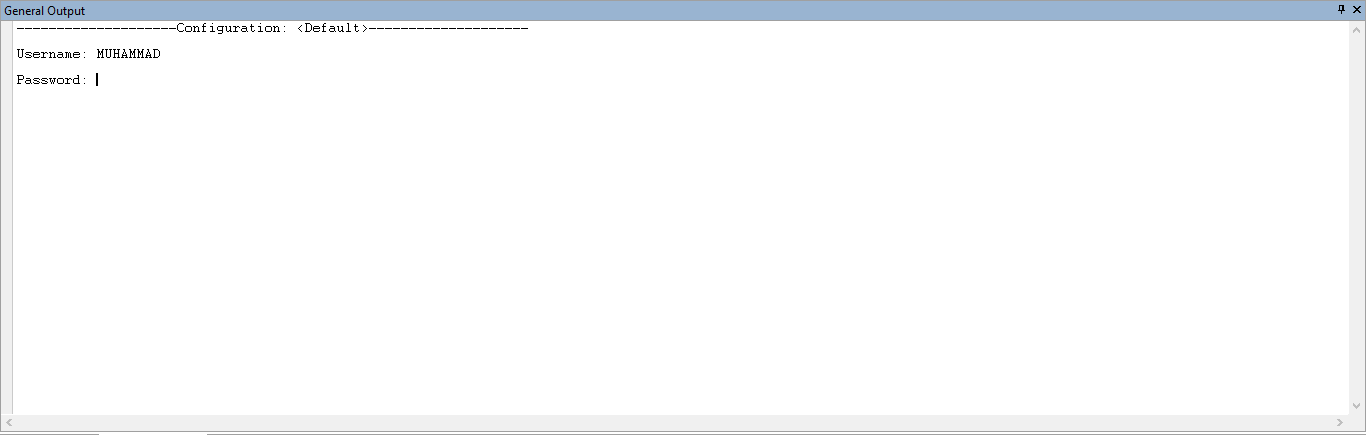
|  |  |
| --- | --- |
| **TEST** | **OBJECTIVE** |
| **Case 1** | To check if program will ask for your username |
| **Case 2** | To check if program will ask for your password |
| **Case 3** | To check if your password is correct |
| **Case 4** | To check if your username is correct |
| **Case 5** | To check if main menu page will display correct to inter number form [1-6] |
| **Case 6** | To check if choosing option number 1 will go to adding page |
| **Case 7** | To check if choosing option number 2 will go to view all patient record page |
| **Case 8** | To check if choosing option number 3 will go to searching record page |
| **Case 9** | To check if choosing option number 4 will go to update record page |
| **Case 10** | Check if choosing option number 5 will go to delete record page |
| **Case 11** | Check if choosing option number 6 will go to Calculate Customer Items |
| **Case 12** | Check if choosing option number 7 will go to Exit the Program |
| **Case 13** | To check if the program will receive number in patient name and words in patient id |
| **Case 14** | To check whether drug price will receive alphabet |
| **Case 15** | To check if view of all patient record will display |
| **Case 16** | To check if patient ward will receive numbers |
| **Case 17** | To check if drug price will receive alphabet |
| **Case 18** | To add every requirement, it will display |
| **Case 19** | To check whether the program will display all the inserted record |
| **Case 20** | To delete record, we can check weather in will be deleted |
| **Case 21** | To see if your username and password is correct |

# TEST RESULTS

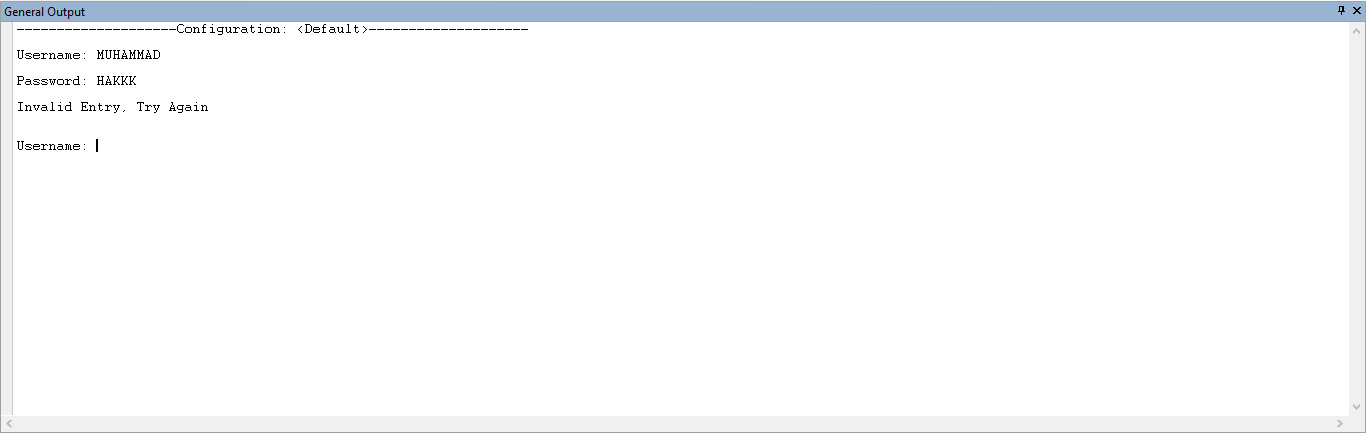
|  |  |
| --- | --- |
| **Test Case** | 1 |
| **Objective** | To check if program will ask for your username |
| **Test Data** | Run Program |
| **Expected Test Result** | Program will ask for user name |
| **Actual Test Result** | As shown below |
| **Conclusion** | Successful. |



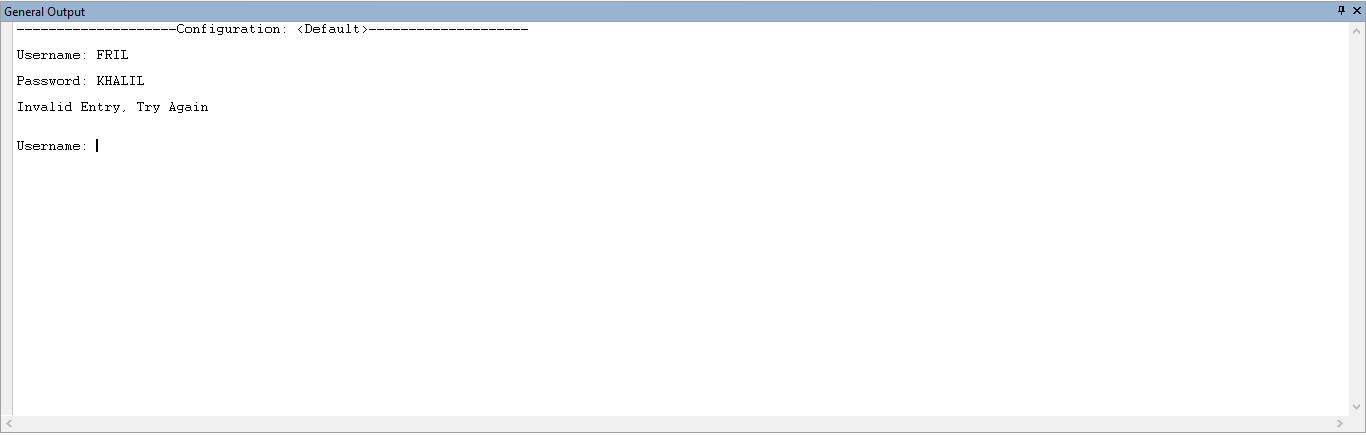
|  |  |
| --- | --- |
| **Test Case** | 2 |
| **Objective** | To check if program will ask for your password |
| **Test Data** | Key is your Username |
| **Expected Test Result** | program will ask for password |
| **Actual Test Result** | As shown below |
| **Conclusion** | Successful. |



|  |  |
| --- | --- |
| **Test Case** | 3 |
| **Objective** | To check if your password is correct |
| **Test Data** | Enter wrong |
| **Expected Test Result** | Invalid Entry, Try Again |
| **Actual Test Result** | As shown below |
| **Conclusion** | Successful. |

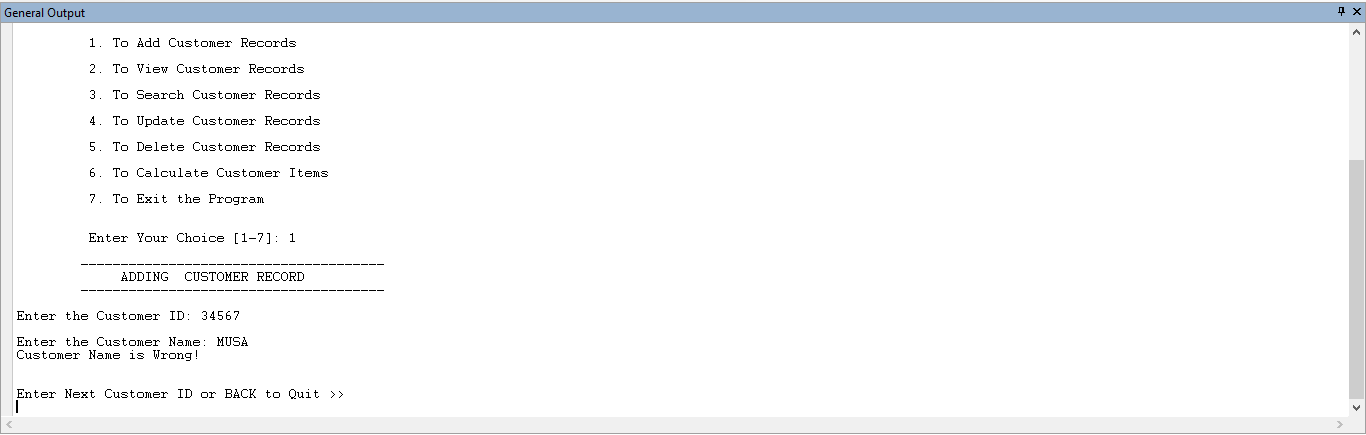


|  |  |
| --- | --- |
| **Test Case** | 4 |
| **Objective** | To check if your username is correct |
| **Test Data** | Enter wrong |
| **Expected Test Result** | It will display Invalid Entry, Try Again |
| **Actual Test Result** | As shown below |
| **Conclusion** | Successful. |

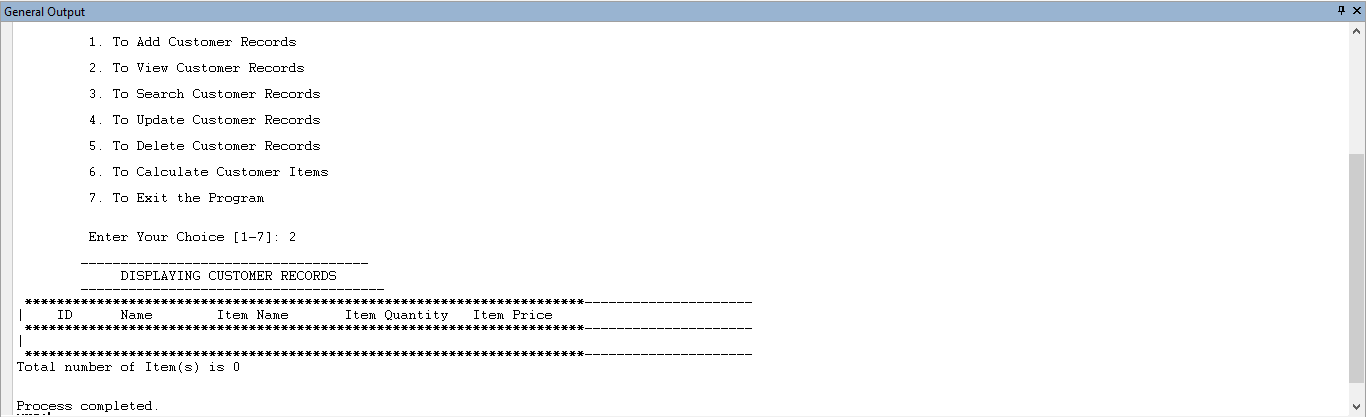


|  |  |
| --- | --- |
| **Test Case** | 5 |
| **Objective** | To check if main menu page will display correct to Enter number form [1-7] |
| **Test Data** | Key correct login details |
| **Expected Test Result** | main menu page will display correct with number from [1-7] |
| **Actual Test Result** | As shown below |
| **Conclusion** | Successful. |

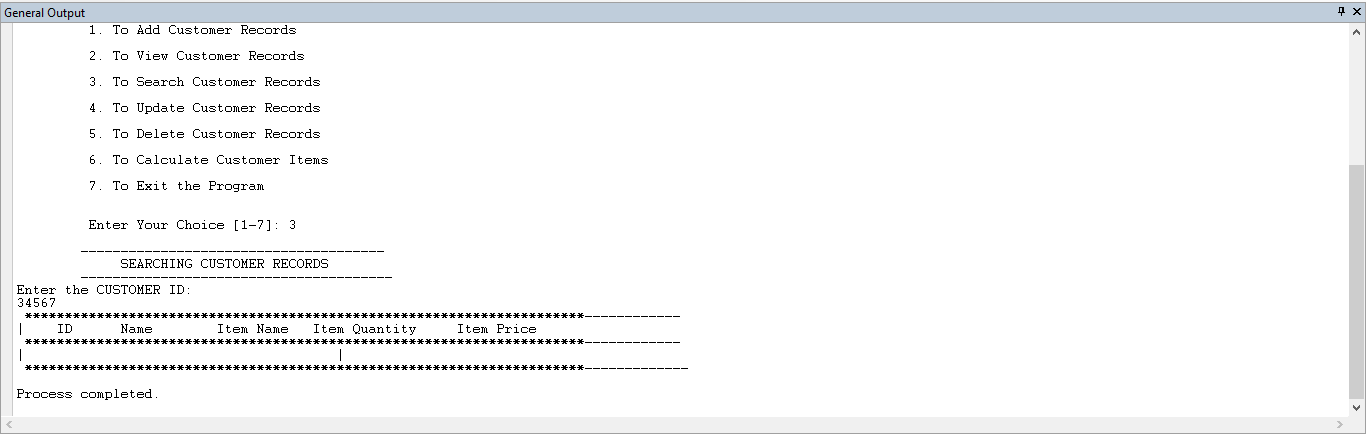
|  |  |
| --- | --- |
| **Test Case** | 6 |
| **Objective** | To check if choosing option number 1 will go to adding page |
| **Test Data** | Enter 1 |
| **Expected Test Result** | It will go to add page |
| **Actual Test Result** | As shown below |
| **Conclusion** | Successful. |



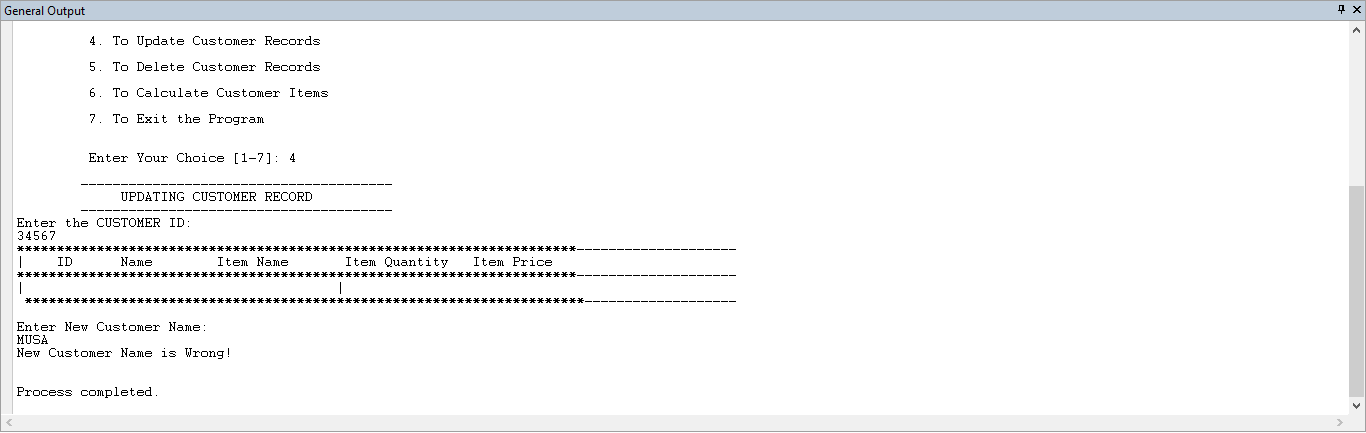
|  |  |
| --- | --- |
| **Test Case** | 7 |
| **Objective** | To check if choosing option number 2 will go to view all patient record page |
| **Test Data** | Enter 2 |
| **Expected Test Result** | It will go to view all patient record page |
| **Actual Test Result** | As shown below |
| **Conclusion** | Successful. |



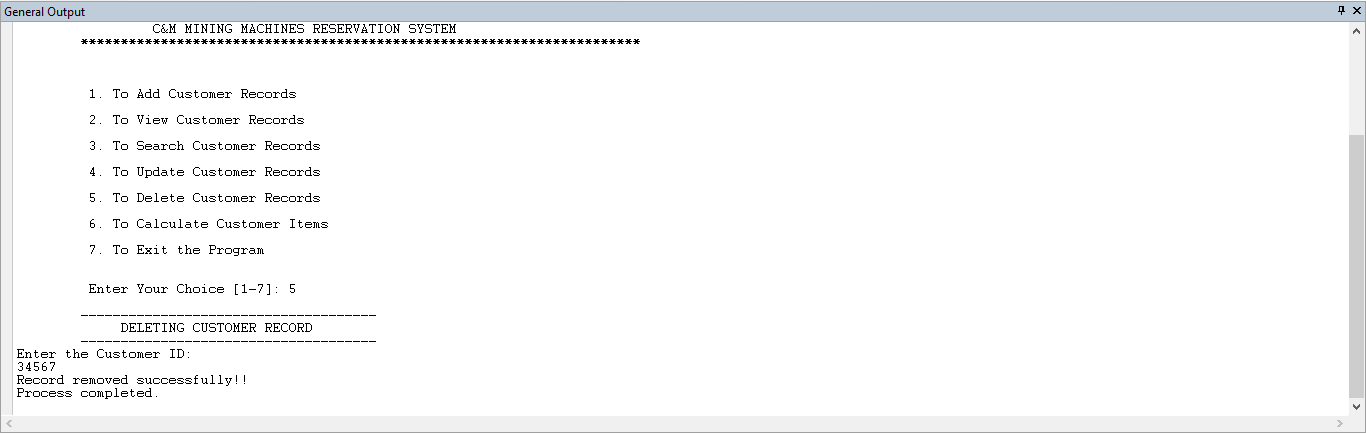
|  |  |
| --- | --- |
| **Test Case** | 8 |
| **Objective** | To check if choosing option number 3 will go to searching record page |
| **Test Data** | Enter 3 |
| **Expected Test Result** | It will go to searching record page |
| **Actual Test Result** | As shown below |
| **Conclusion** | Successful. |



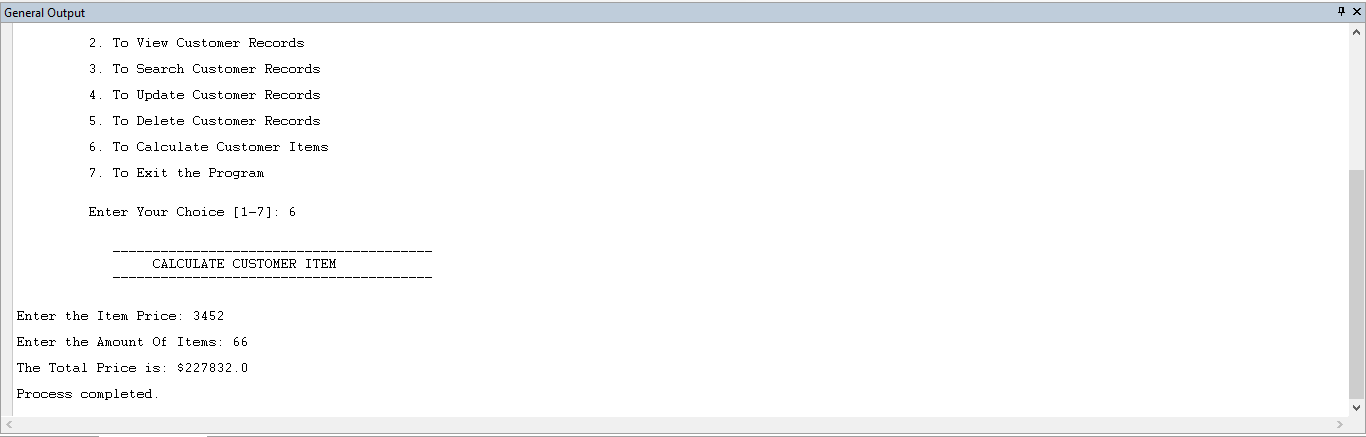
|  |  |
| --- | --- |
| **Test Case** | 9 |
| **Objective** | To check if choosing option number 4 will go to update record page |
| **Test Data** | Enter 4 |
| **Expected Test Result** | It will go to update record page |
| **Actual Test Result** | As shown below |
| **Conclusion** | Successful. |



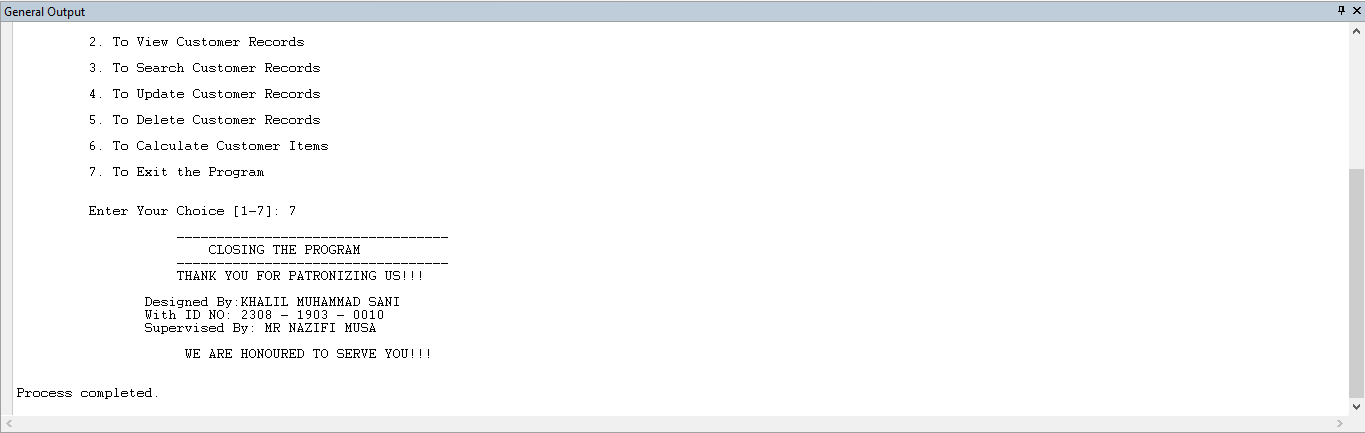
|  |  |
| --- | --- |
| **Test Case** | 10 |
| **Objective** | Check if choosing option number 5 will go to delete record page |
| **Test Data** | Enter 5 |
| **Expected Test Result** | It will go to delete record page |
| **Actual Test Result** | As shown below |
| **Conclusion** | Successful. |



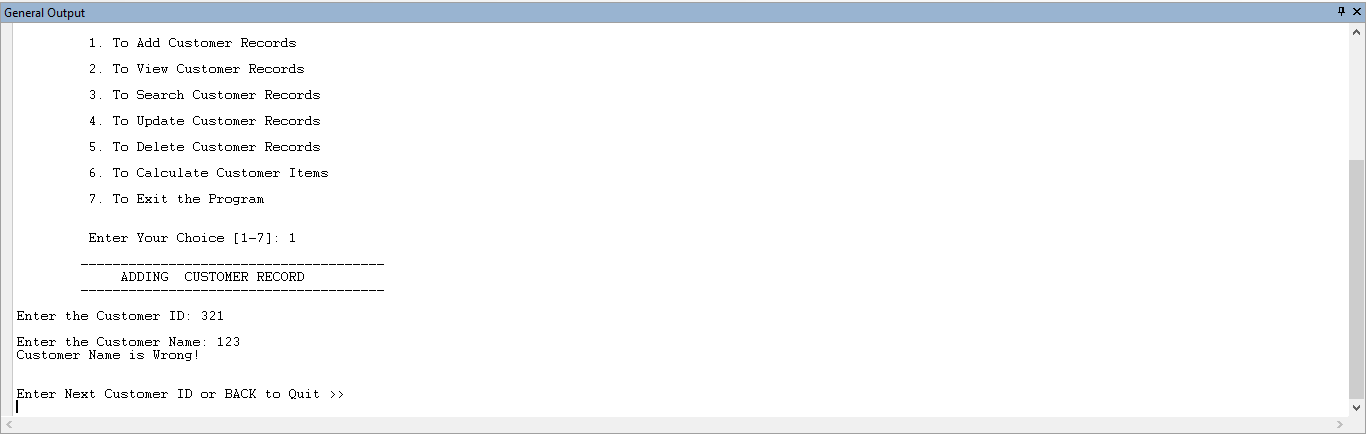
|  |  |
| --- | --- |
| **Test Case** | 11 |
| **Objective** | Check if choosing option number 6 will go to Calculate Customer Items page |
| **Test Data** | Enter 6 |
| **Expected Test Result** | It will go to exit the program page |
| **Actual Test Result** | As shown below |
| **Conclusion** | Successful. |



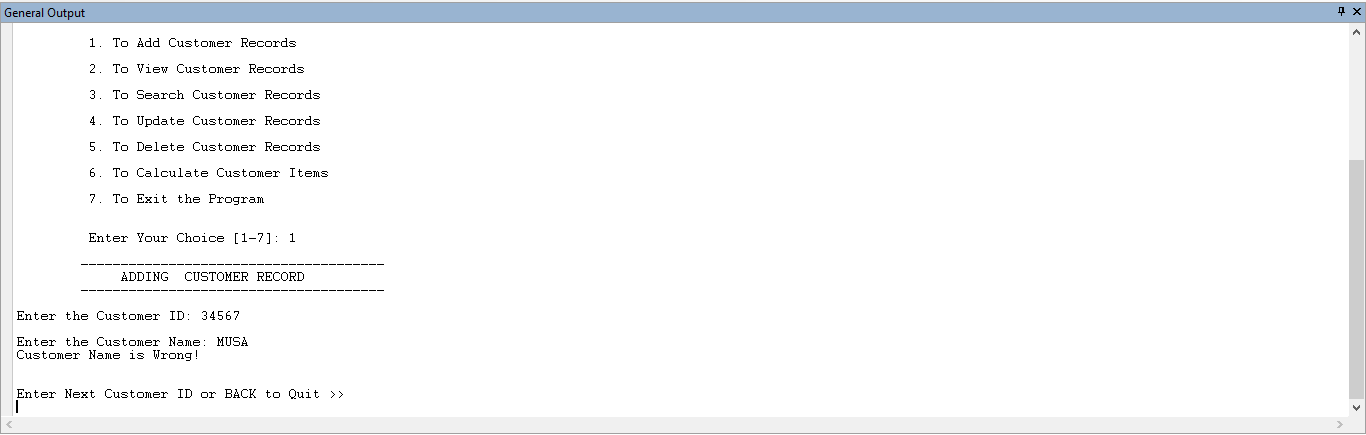
|  |  |
| --- | --- |
| **Test Case** | 12 |
| **Objective** | Check if choosing option number 7 will go to Exit the Program page |
| **Test Data** | Enter 7 |
| **Expected Test Result** | It will go to exit the program page |
| **Actual Test Result** | As shown below |
| **Conclusion** | Successful. |



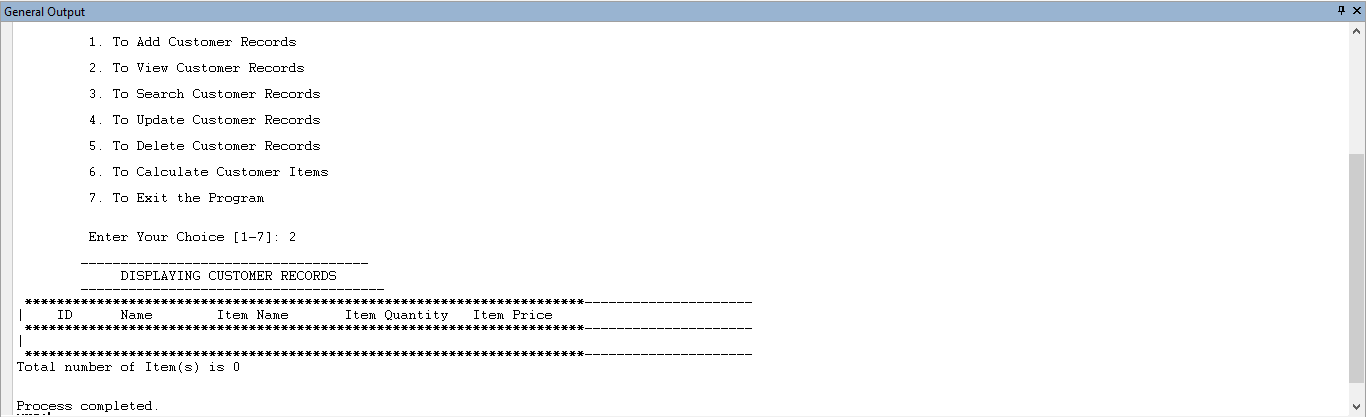
|  |  |
| --- | --- |
| **Test Case** | 13 |
| **Objective** | To check if the program will receive number in customer name and words in customer id |
| **Test Data** | Enter the customer name and customer id |
| **Expected Test Result** | It will display this |
| **Actual Test Result** | As shown below |
| **Conclusion** | Successful. |



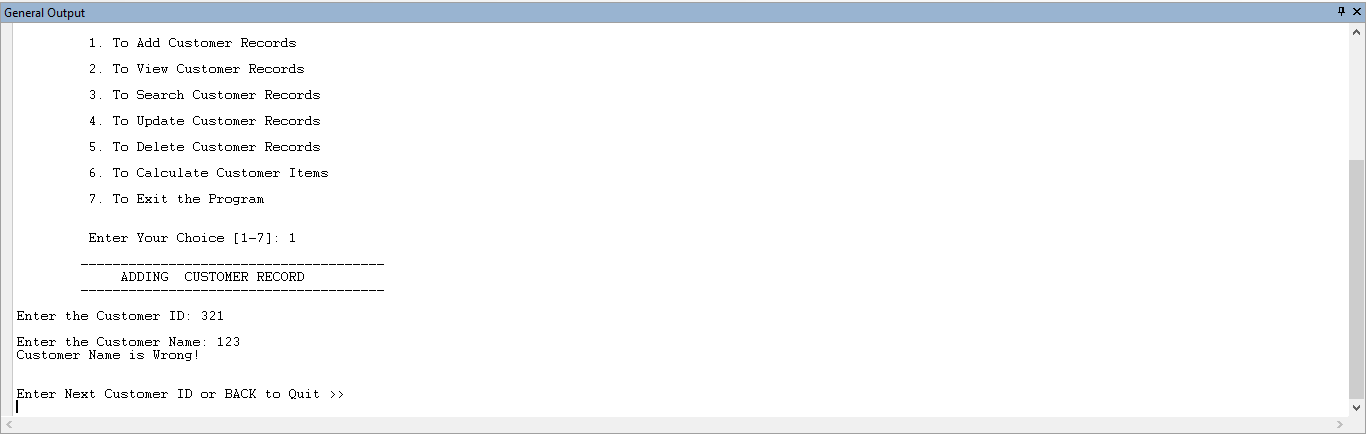
|  |  |
| --- | --- |
| **Test Case** | 14 |
| **Objective** | To check whether item price will receive alphabet |
| **Test Data** | Enter item price as alphabet |
| **Expected Test Result** | It will display |
| **Actual Test Result** | As shown below |
| **Conclusion** | Successful. |



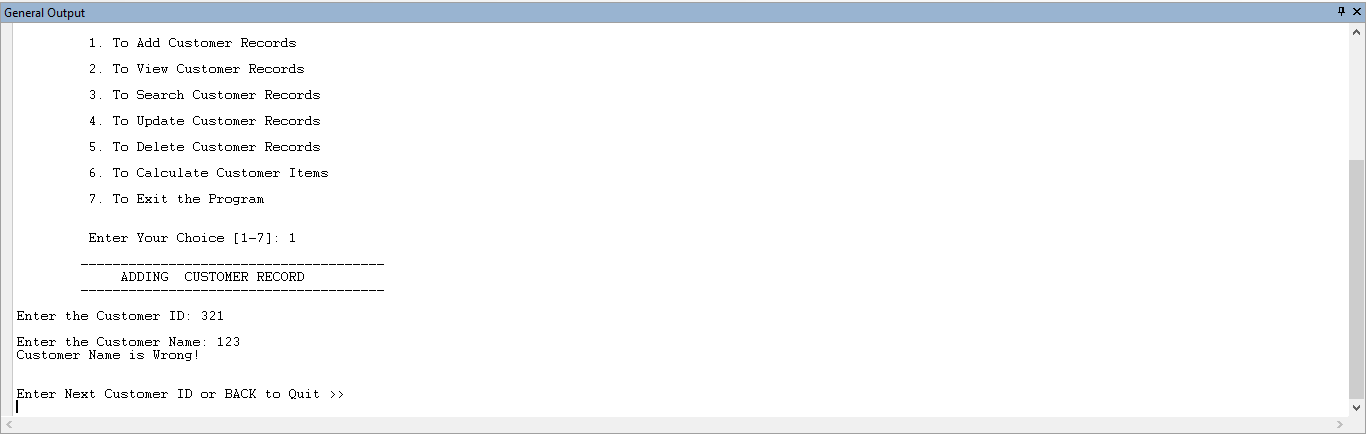
|  |  |
| --- | --- |
| **Test Case** | 15 |
| **Objective** | To check if view of all customer record will display |
| **Test Data** | Enter 2 |
| **Expected Test Result** | It will display all record but now the record is empty |
| **Actual Test Result** | As shown below |
| **Conclusion** | Successful. |



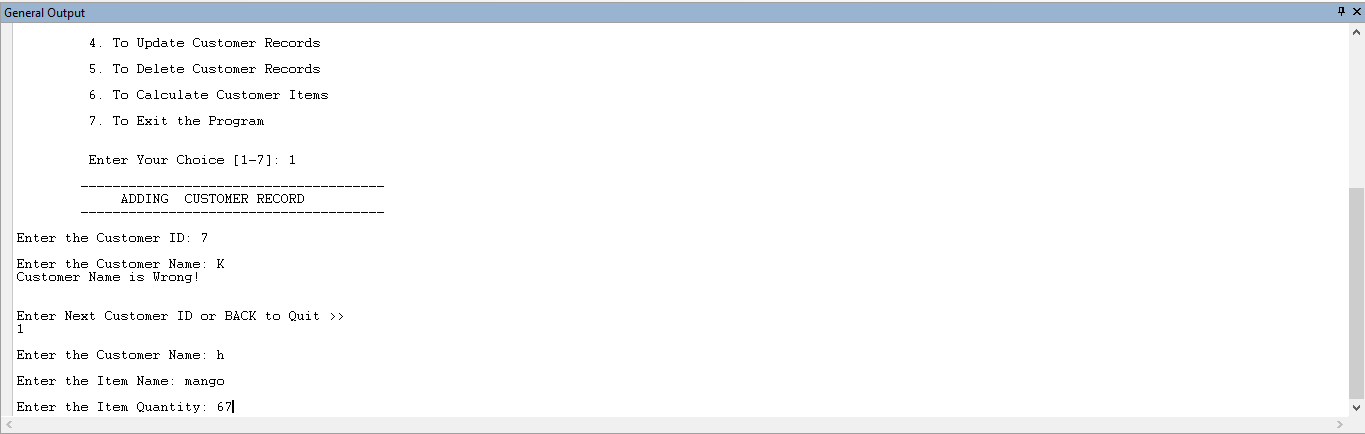
|  |  |
| --- | --- |
| **Test Case** | 16 |
| **Objective** | To check if customer page will receive numbers |
| **Test Data** | Enter numbers |
| **Expected Test Result** | It will display incorrect |
| **Actual Test Result** | As shown below |
| **Conclusion** | Successful. |



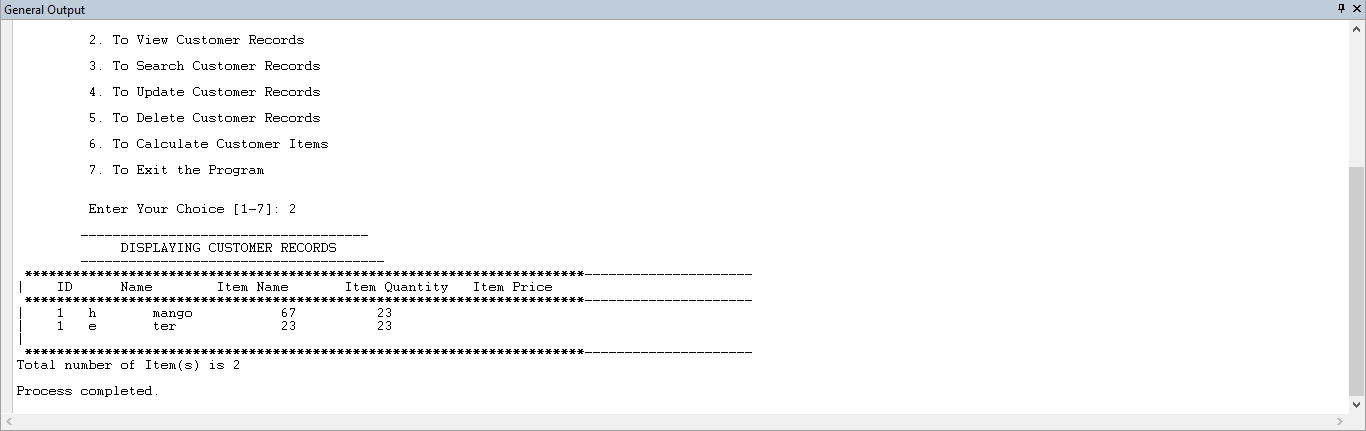
|  |  |
| --- | --- |
| **Test Case** | 17 |
| **Objective** | To check if item price will receive alphabet |
| **Test Data** | Enter alphabet |
| **Expected Test Result** | It will display incorrect |
| **Actual Test Result** | As shown below |
| **Conclusion** | Successful. |



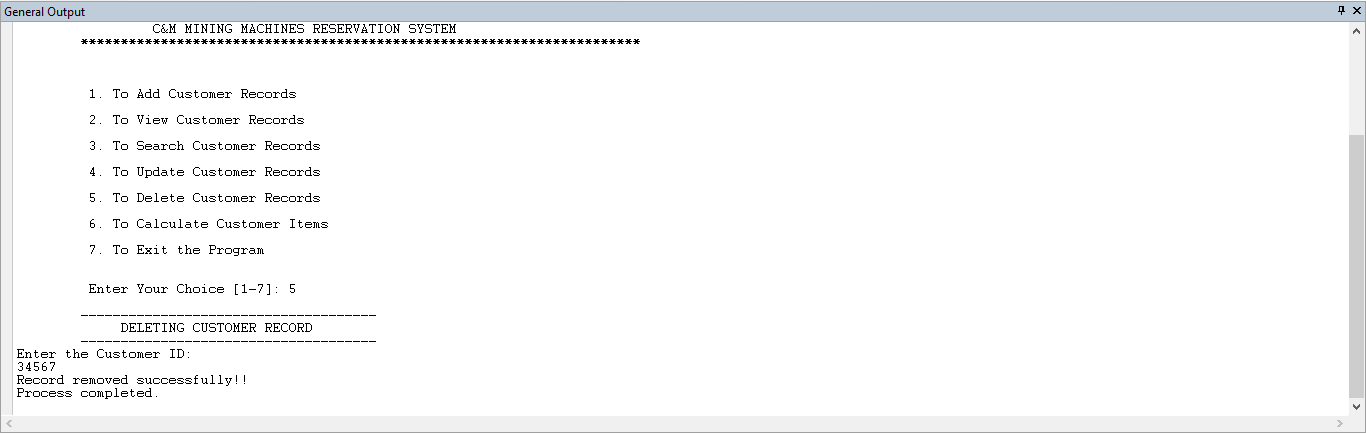
|  |  |
| --- | --- |
| **Test Case** | 18 |
| **Objective** | To add every requirement, it will display |
| **Test Data** | Enter correct requirement |
| **Expected Test Result** | It will display incorrect |
| **Actual Test Result** | As shown below |
| **Conclusion** | Successful. |



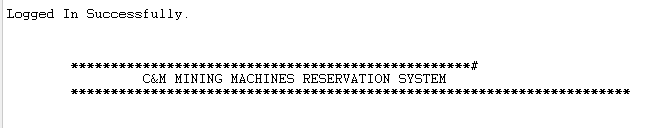
|  |  |
| --- | --- |
| **Test Case** | 19 |
| **Objective** | To check whether the program will display all the inserted record |
| **Test Data** | Enter 2 to view record |
| **Expected Test Result** | It will display all the inserted record |
| **Actual Test Result** | As shown below |
| **Conclusion** | Successful. |



|  |  |
| --- | --- |
| **Test Case** | 20 |
| **Objective** | To delete record, we can check weather in will be deleted |
| **Test Data** | Enter patient id to delete the record |
| **Expected Test Result** | It will display the record deleted successfully!! |
| **Actual Test Result** | As shown below |
| **Conclusion** | Successful. |



|  |  |
| --- | --- |
| **Test Case** | 21 |
| **Objective** | To see if your username and password is correct |
| **Test Data** | Enter your username and password |
| **Expected Test Result** | It will display some text |
| **Actual Test Result** | As shown below |
| **Conclusion** | Successful. |



# TEST LOG

|  |  |
| --- | --- |
| **TEST CASES** | **RESULT** |
| **Case 1** | Successful |
| **Case 2** | Successful |
| **Case 3** | Successful |
| **Case 4** | Successful |
| **Case 5** | Successful |
| **Case 6** | Successful |
| **Case 7** | Successful |
| **Case 8** | Successful |
| **Case 9** | Successful |
| **Case 10** | Successful |
| **Case 11** | Successful |
| **Case 12** | Successful |
| **Case 13** | Successful |
| **Case 14** | Successful |
| **Case 15** | Successful |
| **Case 16** | Successful |
| **Case 17** | Successful |
| **Case 18** | Successful |
| **Case 19** | Successful |
| **Case 20** | Successful |
| **Case 21** | Successful |

# HARDWARE AND SOFTWARE REQUIREMENT**.**

## **Hardware Requirement**

Complete Intel Pentium system (CORETMi3) or above

2 GB RAM or above

500 GB Hard Disk or Above

External drives

Laptop

Laser printer

CCTV Camera

## Software Requirement

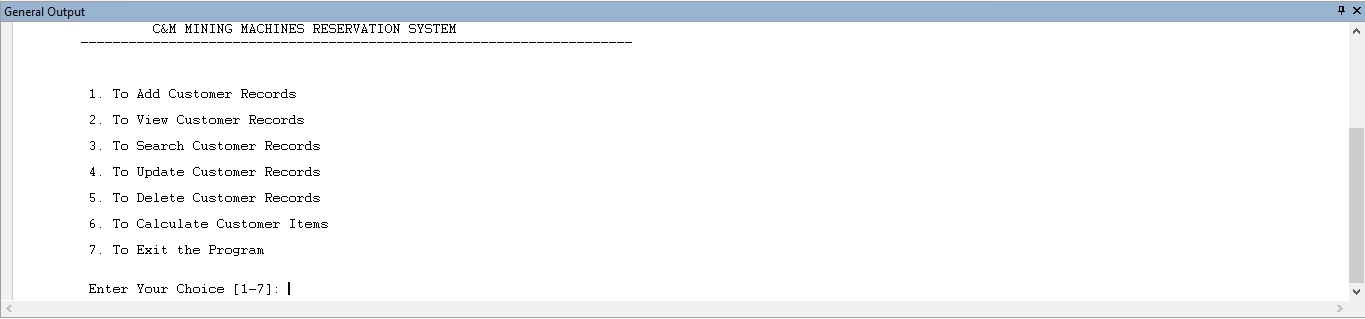
Windows OS (windows 10)

Creator

JDK 1.8.\_111

# USER MANUAL

# The first step of user manual is



If you inter your JCREATOR after entering the java file it will need for your user name and password make sure you insert your user name and password correctly after that it will display those texts shows in image

## Like this C&M MINING MACHINES RESERVATION SYSTEM

The Menu shows the Key functions of the program:

1. To Add Customer Records

2. To View Customer Records

3. To Search Customer Records

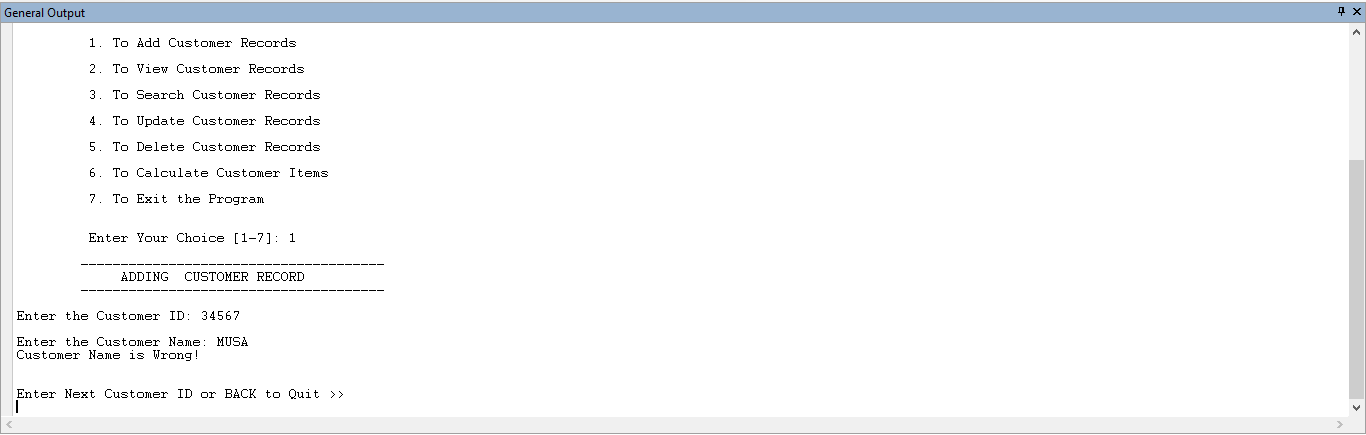
4. To Update Customer Records

5. To Delete Customer Records

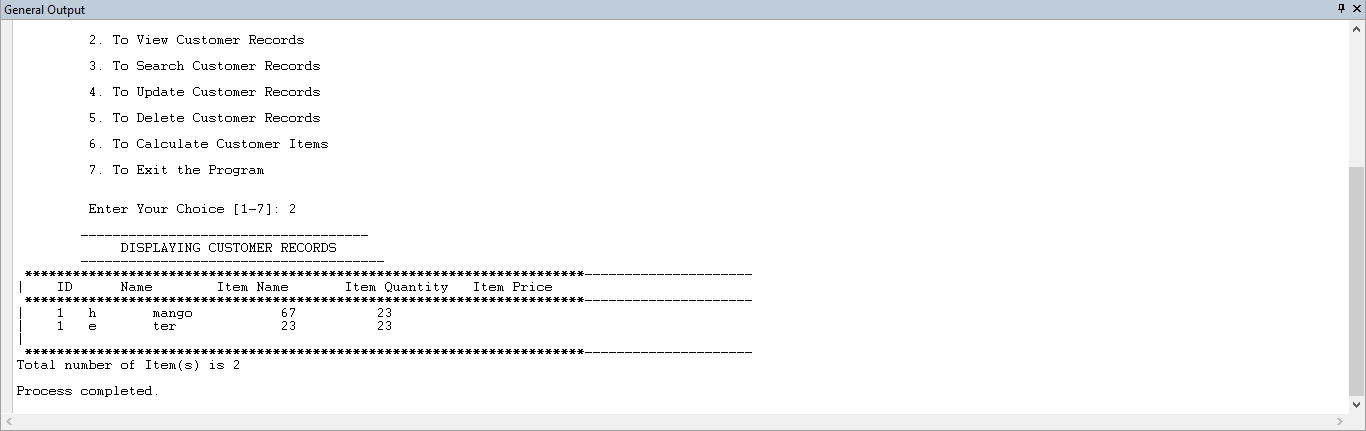
6. To Calculate Customer Items

7. To Exit the Program

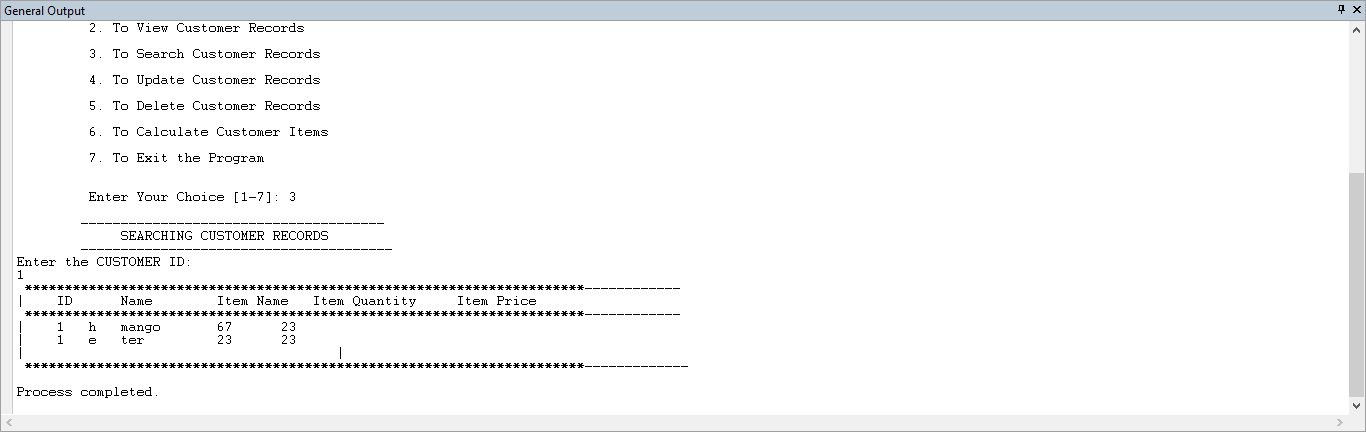
**Step two of user manual**

****

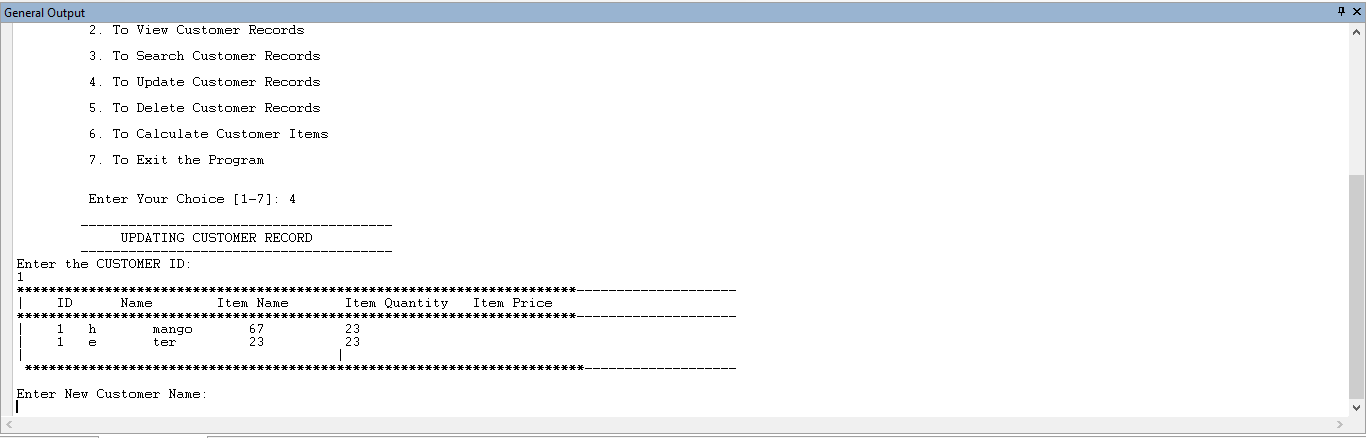
# Enter 1 to add customer record



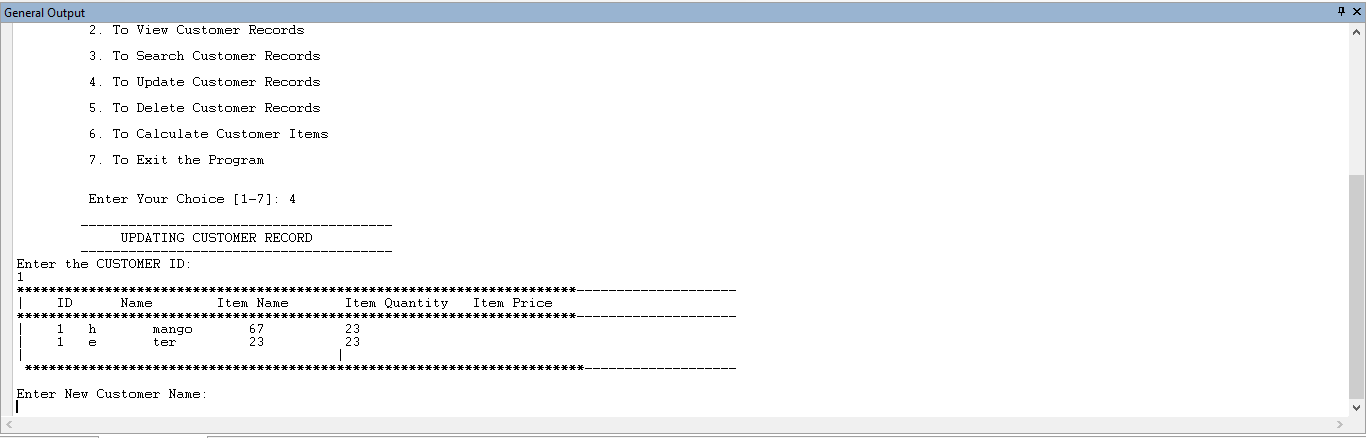
Enter 2 view all the customer record



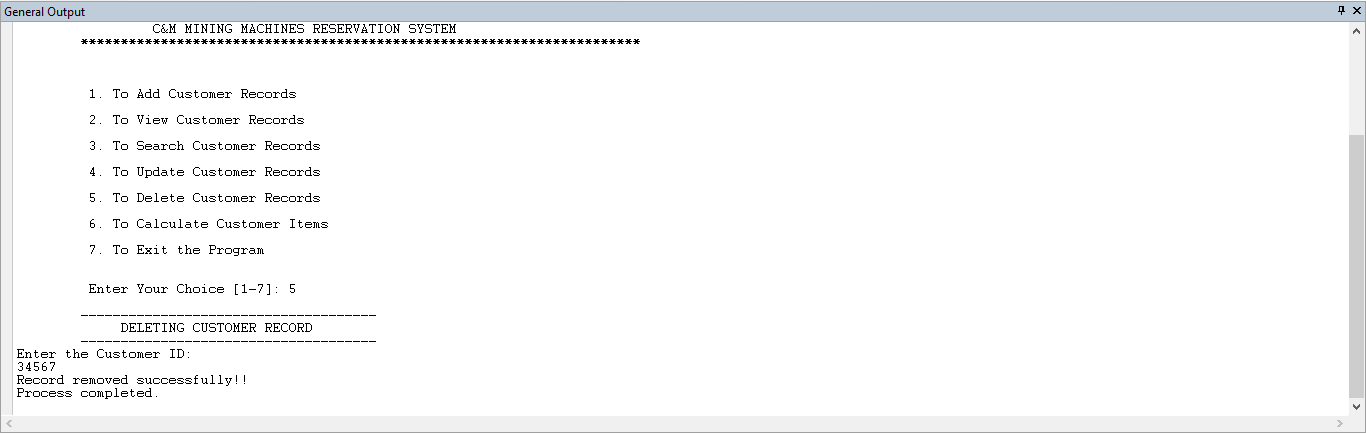
Enter 3 to search customer record



Enter 4 to update customer record

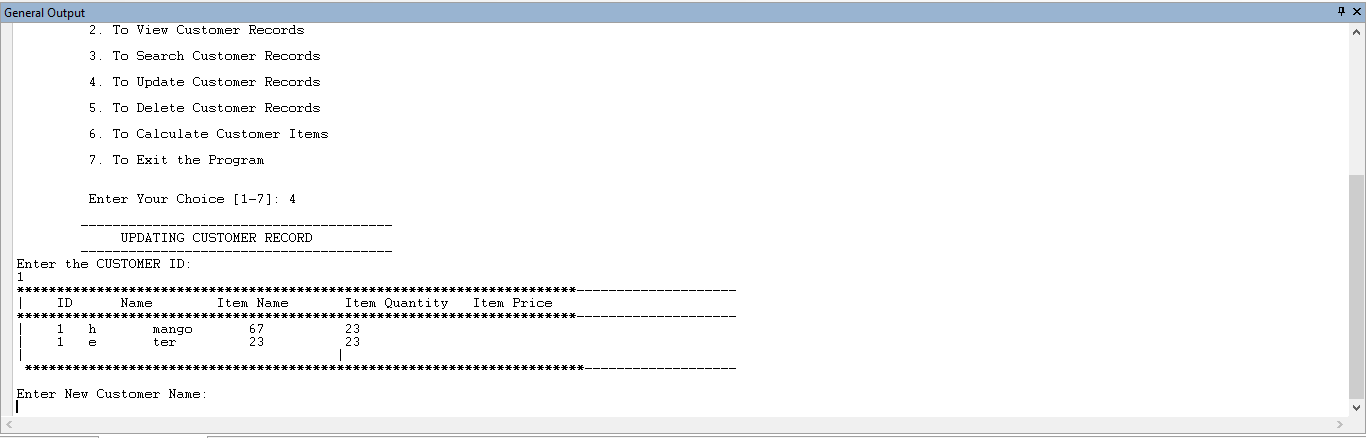


Enter 5 delete record



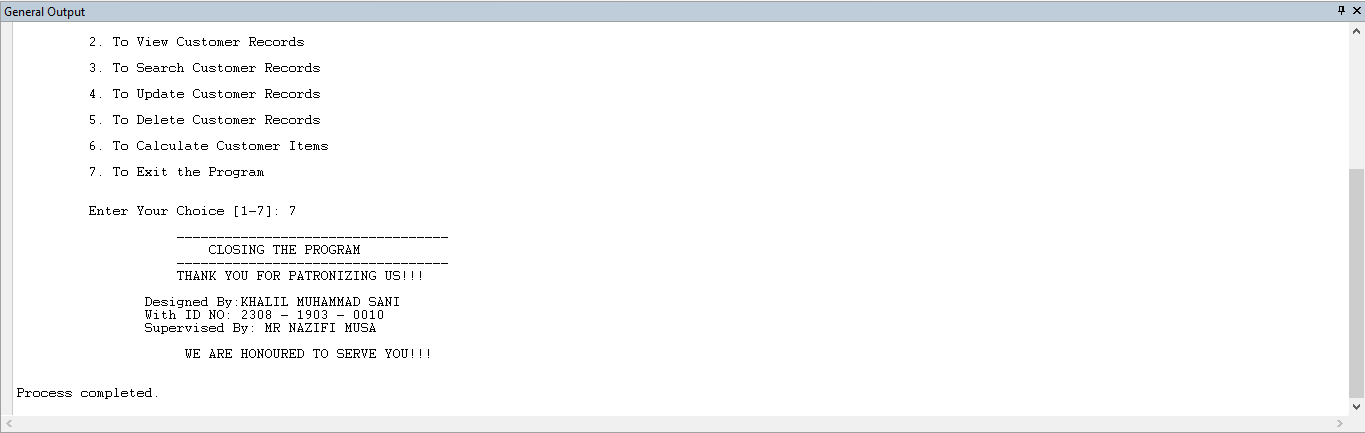
Enter 6 to Calculate Customer Items

This is the end of the program I’m sure using this user manual you me able to use the program successful.



Enter 7 to Exit the program

This is the end of the program I’m sure using this user manual you me able to use the program successful.



# CONCLUSION: PROGRAM STRENGHT

In conclusion, c&m mining machines reservation system have a very good way of solving their day to day activities, I would like to pass some comments about my program. Basically, my program is made up of 5 options to choose from. These 5 options will prove to be efficient to the user just by entering the buses code, the user can easily retrieve the record that is to be deleted or edited. This is to prevent any acceptance of ridiculous entry by the user.

to improve the efficiency of the user at work, the user can actually check the patient code, patient name, blood group, status, and age under the listing menu. below are the itemized ways that shows the strength of the program.

* This program has the capability of asking user to enter the patient’s information and reduce a space usually for keeping files.
* It has the capability of searching records more accurately and efficiently within a short period of time.
* It has the capability of adding new patient’s information.

## **PROGRAM WEAKNESS**

Due to time constraint, I could not do the source program for the chart. This chart is used in the Add module.

When the user enters the code and the name of the patient, this chart is supposed to check for the code of patient within the range of name(s) entered. Without this function, the program is unable to locate the patient record of the clinic in the chart.

### 

### **PROGRAM ENHANCEMENT**

In the near future, some enhancements can be made to the program. A print function can be added to the program so that the user can print out a particular record or print out all the information that took place.