International Journal of Computer Trends and Technology
ISSN: 2231 – 2803 / https://doi.org/10.14445/22312803/IJCTT-V69I3P109

Volume 69 Issue 3, 46-51, March, 2021
© 2021 Seventh Sense Research Group®

Review Article

Analysis of Optimization Technique of Same Program Written in Two Different Interfaces i.e. CUI and GUI Using Java and Calculate Their Differences

Arpana Chaturvedi[1], Deepti Khanna[2], Deepit Aggarwal [3], Ayush Prasad [4]

[1] *Associate Professor, Jagannath International Management School, New Delhi, India*
[2]*Associate Professor, JIMS, New Delhi, India*
[3]*Student, JIMS New Delhi, India*
[4]*Student, JIMS New Delhi, India*

Received Date: 02 February 2021
Revised Date: 23 March 2021
Accepted Date: 25 March 2021

Abstract *- Java is one of the most stable programming languages and form time to time Oracle Corporation frequently updates the language , it is also platform independent and supports common programming paradigms has got rich set of APIs , loads of frameworks, Libraries, IDEs and development tools ,simplify Development of real-time software , facilitates embedded computing and is robust and secure ,so a vast majority of applications use JAVA Programming Language .Here in our research work we have developed a program named inventory control management system. It's a CUI program which is created in core java. A GUI application is also created of the same program with the help of JFrame using event driven programming. in our research work we have used different optimization techniques such as CPU utilization, Heap Count and Threads, etc. to study performance on CUI and GUI, based upon the result we tried to find out which interface is better in terms of memory utilization and CPU utilization.*

Keywords *- Java program, Optimization techniques, GUI, CUI, JFrame.*

.

I. INTRODUCTION

Java is a simple object-oriented, robust, secure, architecture-neutral, portable, high performance, interpreted, threaded, platform independent programming language. it is used to develop applications for the various fields such as banking , retail , information technology , android etc. here for our research paper we have developed a program which is an inventory management system of a company to watch the products requirements of the company and which item needs to be re-ordered accordingly from time to time .this will help the companies in managing the warehouses and keeping a proper track of the items available that can be further used in the calculations of the gross production cost evaluated we have developed the program in Command Line Interface as well as in Graphical User Interface and using both the interfaces we studied the utilization and performance of the program and with the help of the results obtained we have further tried to optimize the program for better performance for the industrial purpose . we have prepared graph to study the performance and memory utilization by both the interfaces and also conducted statistical test based upon the data collected and further studied the test results for the better optimization of the program.

.

II. OBJECTIVE

A. To create same program in Core Java and JFrame using event driven programming, the program is created in both CUI and GUI

To objective of the research paper is to create a program in Core Java and JFrame using event driven programming, the program is created in both CUI and GUI.

B. To find the optimization techniques which include Memory optimization, CPU performance, Heap memory utilization and Disk utilization

To study the optimization techniques which include Memory optimization, CPU performance, Heap memory utilization and Disk utilization to study the performance of the program in CUI and GUI interface for the purpose of finding out which interface is better suitable for the industrial use.

C. To find out which interface is better suitable for the performance of the program

Based upon the study conducted, compare the results of the optimization techniques and study its performance and coming to a conclusion.

III. TOOLS USED

We used following software and hardware for our research: -

Table 1. Operating System Details

Operating System:-	
Edition	Windows 10 Home Single Language
Version	1909

Table 2. Hardware Requirement Details

Hardware Requirements: -	
Processor	Intel® Core™ i5-8250U CPU @ 1.60GHz
Installed RAM	8.00 GB (7.87 GB usable)
Hard disk	1TB
System type	64-bit OS *64-based processor

Table 3. Software Requirement Details

Software Requirements: -	
IDE	Eclipse
Web serer	Xampp
Connector	mysql-connector-java-5.0.8-bin
Analysis Tool	Your Kit-JAVA Profiler

Table 4. Additional Requirement Details

Additional Software: -	
Additional IDE Software	Window Builder
Connectors: -	rs2xml
	jgoodies-forms-1.8.0-sources
	javax.mail-1.6.2

IV. INTRODUCTION TO THE PROGRAM TO PERFORM RESEARCH

For our research paper we have a written a program in core java, core java is a collection of libraries rather than just the simple programming language. We developed a code for inventory management system which is made for the purpose of keeping the track of the available stock in the warehouses; it is a multiple user system which will help in saving the time of the organization while keeping a record. The program has got a login system which is of multiple user login and is robust and secure, the program contains mainly three functions i.e. item , supplier and reorder each function have sub functions such as insert, update, display and delete .The program is easy to use and is secure as well.

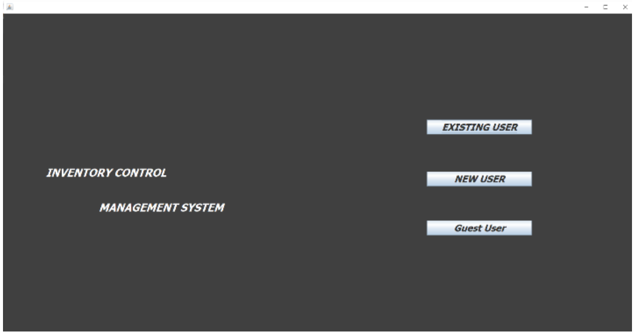


Fig. 1 Main Frame of the GUI Application

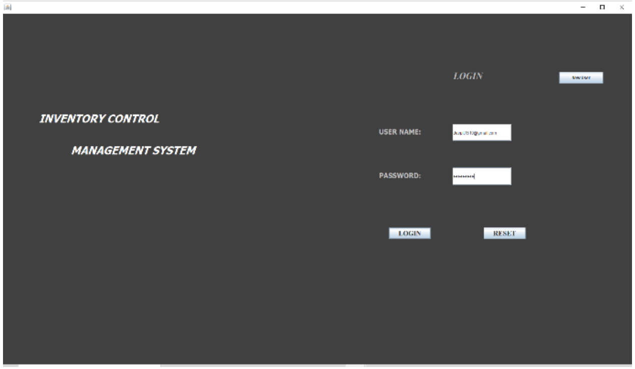


Fig. 2 Login Frame of the GUI Application

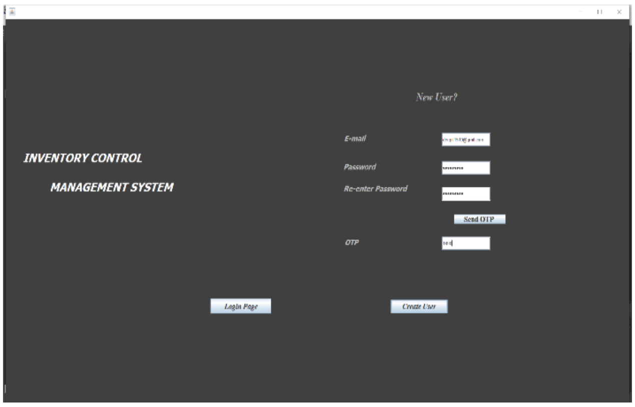


Fig. 3 New User Login Creation Frame of the GUI Application

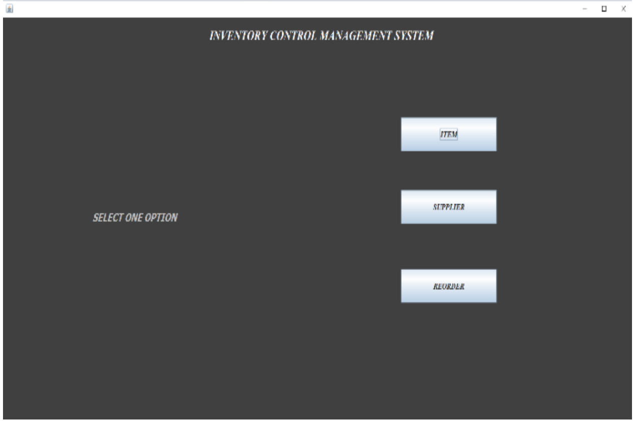
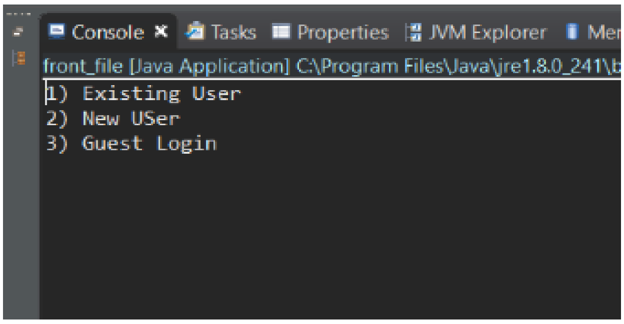
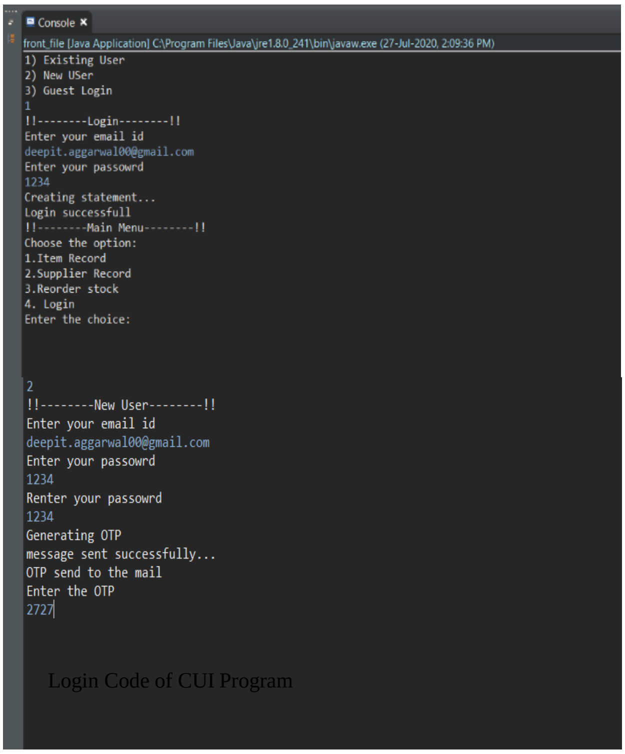


Fig. 4 First Frame After Login Frame of the GUI Application



```
front_file [Java Application] C:\Program Files\Java\jre1.8.0_241\b
1) Existing User
2) New USer
3) Guest Login
```

Fig. 5 Login and the main menu after Login of the CUI Program



```
front_file [Java Application] C:\Program Files\Java\jre1.8.0_241\bin\javaw.exe (27-Jul-2020, 2:09:36 PM)
1) Existing User
2) New USer
3) Guest Login
1
!!--------Login--------!!
Enter your email id
deepit.aggarwal00@gmail.com
Enter your passowrd
1234
Creating statement...
Login successfull
!!--------Main Menu--------!!
Choose the option:
1.Item Record
2.Supplier Record
3.Reorder stock
4. Login
Enter the choice:

2
!!--------New User--------!!
Enter your email id
deepit.aggarwal00@gmail.com
Enter your passowrd
1234
Renter your passowrd
1234
Generating OTP
message sent successfully...
OTP send to the mail
Enter the OTP
2727
```

Login Code of CUI Program

Fig. 6 Login and the main menu after Login of the CUI Program

V. CHARACTER USER INTERFACE

Character user interface or command –line user interface (CUI) is a method in which the user interacts with computer programs. The interface allows the users to issue commands in one or more lines to a program. It is difficult in navigation has got high precision, computing speed is high, is difficult to operate and require expertise, requires low memory, is less flexible and the appearance cannot be changed. One of the best examples of the CUI interface is MS-DOS and Windows command Prompt.

The CUI interface of the inventory management system is easy to understand and use.

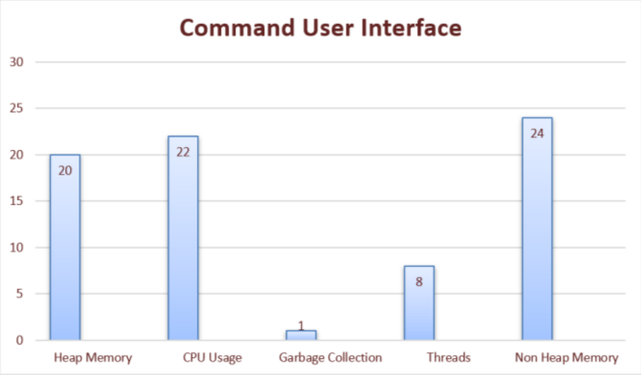
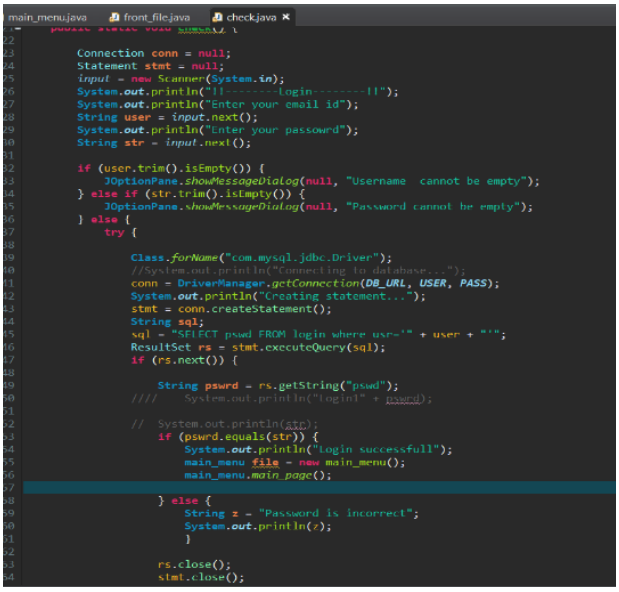
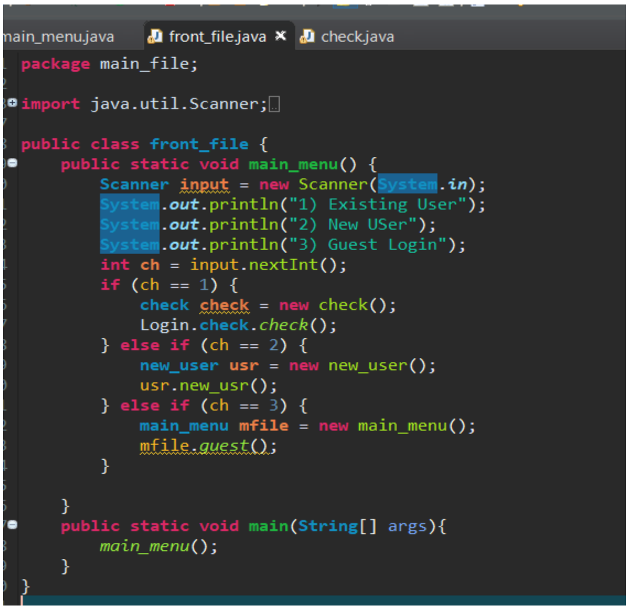


Fig. 7 The Memory, CPU, Threads Utilization and Displaying the Garbage Collection in CUI



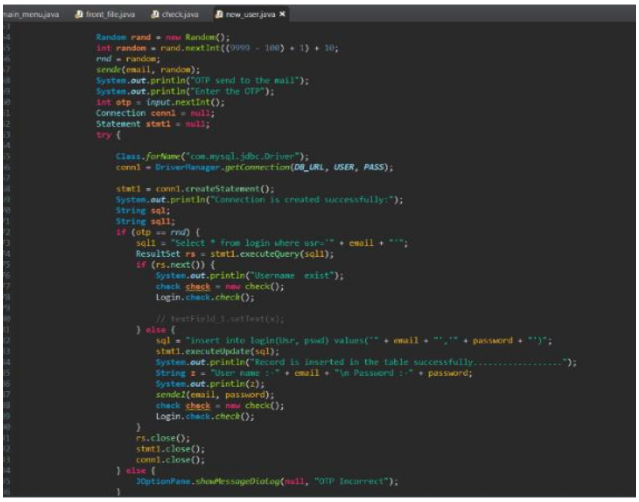
```
    public static void check() {

        Connection conn = null;
        Statement stmt = null;
        input = new Scanner(System.in);
        System.out.println("!!--------Login--------!!");
        System.out.println("Enter your email id");
        String user = input.next();
        System.out.println("Enter your passowrd");
        String str = input.next();

        if (user.trim().isEmpty()) {
            JOptionPane.showMessageDialog(null, "Username  cannot be empty");
        } else if (str.trim().isEmpty()) {
            JOptionPane.showMessageDialog(null, "Password cannot be empty");
        } else {
            try {

                Class.forName("com.mysql.jdbc.Driver");
                //System.out.println("Connecting to database...");
                conn = DriverManager.getConnection(DB_URL, USER, PASS);
                System.out.println("Creating statement...");
                stmt = conn.createStatement();
                String sql;
                sql = "SELECT pswd FROM login where usr='" + user + "'";
                ResultSet rs = stmt.executeQuery(sql);
                if (rs.next()) {

                    String pswrd = rs.getString("pswd");
                ////    System.out.println("Login1" + pswrd);

                //  System.out.println(str);
                    if (pswrd.equals(str)) {
                        System.out.println("Login successfull");
                        main_menu file = new main_menu();
                        main_menu.main_page();

                    } else {
                        String z = "Password is incorrect";
                        System.out.println(z);
                        }

                    rs.close();
                    stmt.close();
```

Fig. 8 Main Menu Code of the CUI Program



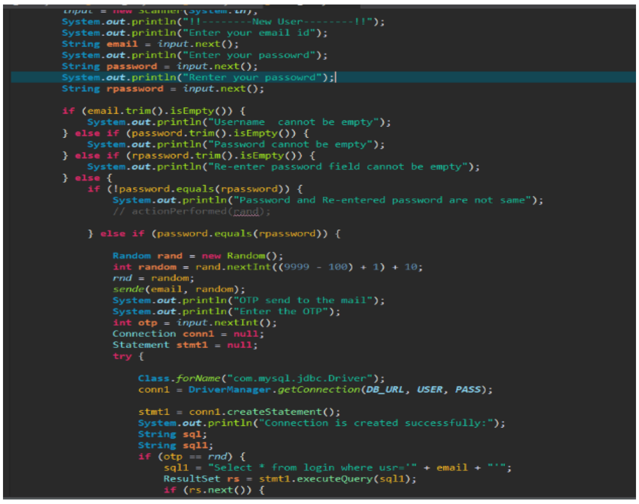
```
package main_file;

import java.util.Scanner;

public class front_file {
    public static void main_menu() {
        Scanner input = new Scanner(System.in);
        System.out.println("1) Existing User");
        System.out.println("2) New USer");
        System.out.println("3) Guest Login");
        int ch = input.nextInt();
        if (ch == 1) {
            check check = new check();
            Login.check.check();
        } else if (ch == 2) {
            new_user usr = new new_user();
            usr.new_usr();
        } else if (ch == 3) {
            main_menu mfile = new main_menu();
            mfile.guest();
        }

    }
    public static void main(String[] args){
        main_menu();
    }
}
```

Fig. 9 New User Code of CUI Program



```java
Random rand = new Random();
int random = rand.nextInt((9999 - 100) + 1) + 10;
rnd = random;
sende(email, random);
System.out.println("OTP send to the mail");
System.out.println("Enter the OTP");
int otp = input.nextInt();
Connection conn1 = null;
Statement stmt1 = null;
try {

    Class.forName("com.mysql.jdbc.Driver");
    conn1 = DriverManager.getConnection(DB_URL, USER, PASS);

    stmt1 = conn1.createStatement();
    System.out.println("Connection is created successfully:");
    String sql;
    String sql1;
    if (otp == rnd) {
        sql1 = "Select * from login where usr='" + email + "'";
        ResultSet rs = stmt1.executeQuery(sql1);
        if (rs.next()) {
            System.out.println("Username  exist");
            check check = new check();
            Login.check.check();

            // textField_1.setText(x);
        } else {
            sql = "insert into login(Usr, pswd) values('" + email + "','" + password + "')";
            stmt1.executeUpdate(sql);
            System.out.println("Record is inserted in the table successfully..................");
            String z = "User name :-" + email + "\n Password :-" + password;
            System.out.println(z);
            sende1(email, password);
            check check = new check();
            Login.check.check();
        }
        rs.close();
        stmt1.close();
        conn1.close();
    } else {
        JOptionPane.showMessageDialog(null, "OTP Incorrect");
    }
```

Fig. 10 Main Menu Code of the CUI Program



```java
input = new Scanner(System.in);
System.out.println("!!--------New User--------!!");
System.out.println("Enter your email id");
String email = input.next();
System.out.println("Enter your passowrd");
String password = input.next();
System.out.println("Renter your passowrd");
String rpassword = input.next();

if (email.trim().isEmpty()) {
    System.out.println("Username  cannot be empty");
} else if (password.trim().isEmpty()) {
    System.out.println("Password cannot be empty");
} else if (rpassword.trim().isEmpty()) {
    System.out.println("Re-enter password field cannot be empty");
} else {
    if (!password.equals(rpassword)) {
        System.out.println("Password and Re-entered password are not same");
        // actionPerformed(rand);

    } else if (password.equals(rpassword)) {

        Random rand = new Random();
        int random = rand.nextInt((9999 - 100) + 1) + 10;
        rnd = random;
        sende(email, random);
        System.out.println("OTP send to the mail");
        System.out.println("Enter the OTP");
        int otp = input.nextInt();
        Connection conn1 = null;
        Statement stmt1 = null;
        try {

            Class.forName("com.mysql.jdbc.Driver");
            conn1 = DriverManager.getConnection(DB_URL, USER, PASS);

            stmt1 = conn1.createStatement();
            System.out.println("Connection is created successfully:");
            String sql;
            String sql1;
            if (otp == rnd) {
                sql1 = "Select * from login where usr='" + email + "'";
                ResultSet rs = stmt1.executeQuery(sql1);
                if (rs.next()) {
```

Fig. 11 Main Menu Code of the CUI Program

V. GRAPHICAL USER INTERFACE

Graphical user interface is a method which allows the users to interact with electronic devices through graphical icons and primary notation. It is easy to use has got high low precision, computing speed is low, is easy to operate, requires high memory is more flexible and the appearance can be customized.

The GUI interface of the program is more user friendly as compared to the CUI interface as it has got labels, text field and buttons which helps in better understanding of the program.

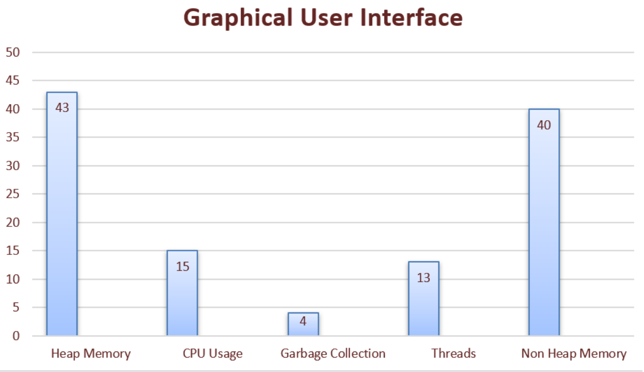
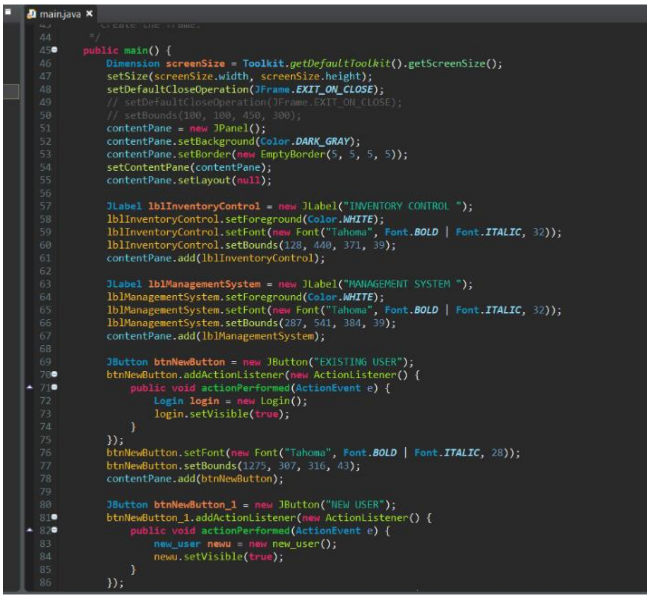
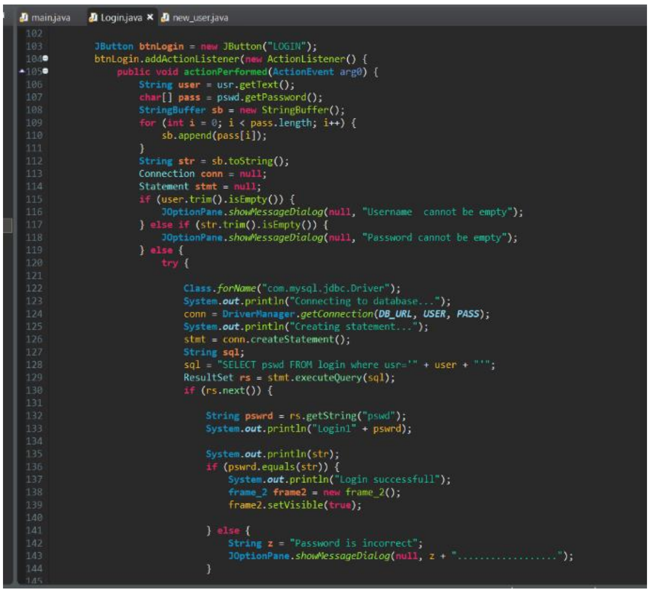


Fig. 12 The Memory, CPU, Threads Utilization and Displaying the Garbage Collection in GUI



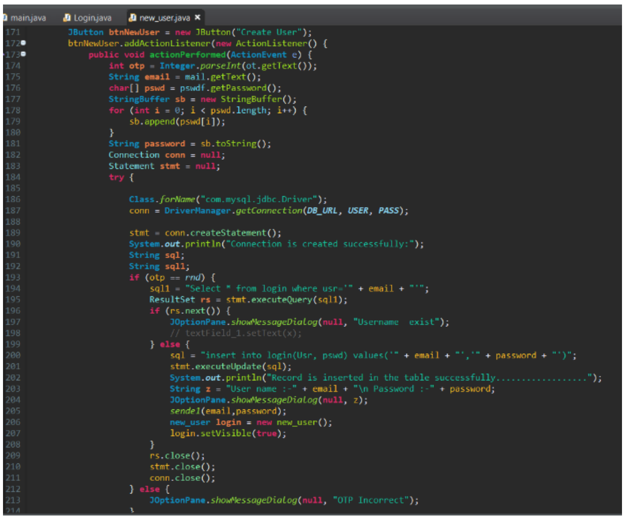
```java
    */
public main() {
    Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
    setSize(screenSize.width, screenSize.height);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    // setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    // setBounds(100, 100, 450, 300);
    contentPane = new JPanel();
    contentPane.setBackground(Color.DARK_GRAY);
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JLabel lblInventoryControl = new JLabel("INVENTORY CONTROL ");
    lblInventoryControl.setForeground(Color.WHITE);
    lblInventoryControl.setFont(new Font("Tahoma", Font.BOLD | Font.ITALIC, 32));
    lblInventoryControl.setBounds(128, 440, 371, 39);
    contentPane.add(lblInventoryControl);

    JLabel lblManagementSystem = new JLabel("MANAGEMENT SYSTEM ");
    lblManagementSystem.setForeground(Color.WHITE);
    lblManagementSystem.setFont(new Font("Tahoma", Font.BOLD | Font.ITALIC, 32));
    lblManagementSystem.setBounds(287, 541, 384, 39);
    contentPane.add(lblManagementSystem);

    JButton btnNewButton = new JButton("EXISTING USER");
    btnNewButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            Login login = new Login();
            login.setVisible(true);
        }
    });
    btnNewButton.setFont(new Font("Tahoma", Font.BOLD | Font.ITALIC, 28));
    btnNewButton.setBounds(1275, 307, 316, 43);
    contentPane.add(btnNewButton);

    JButton btnNewButton_1 = new JButton("NEW USER");
    btnNewButton_1.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            new_user newu = new new_user();
            newu.setVisible(true);
        }
    });
```

Fig. 13 New User Code of CUI Program



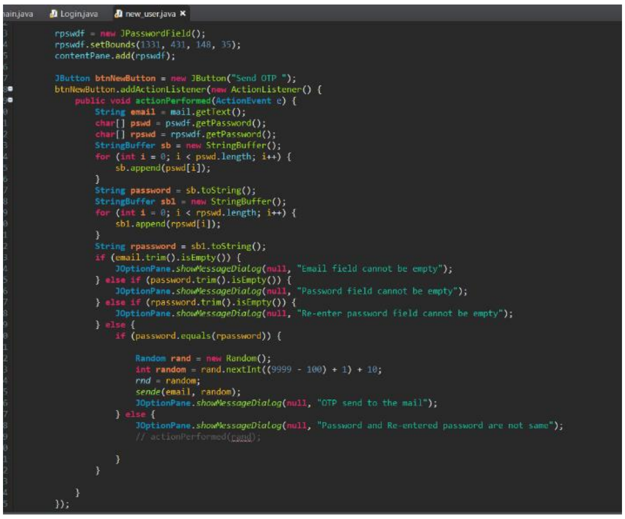
```java
JButton btnLogin = new JButton("LOGIN");
btnLogin.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        String user = usr.getText();
        char[] pass = pswd.getPassword();
        StringBuffer sb = new StringBuffer();
        for (int i = 0; i < pass.length; i++) {
            sb.append(pass[i]);
        }
        String str = sb.toString();
        Connection conn = null;
        Statement stmt = null;
        if (user.trim().isEmpty()) {
            JOptionPane.showMessageDialog(null, "Username  cannot be empty");
        } else if (str.trim().isEmpty()) {
            JOptionPane.showMessageDialog(null, "Password cannot be empty");
        } else {
            try {

                Class.forName("com.mysql.jdbc.Driver");
                System.out.println("Connecting to database...");
                conn = DriverManager.getConnection(DB_URL, USER, PASS);
                System.out.println("Creating statement...");
                stmt = conn.createStatement();
                String sql;
                sql = "SELECT pswd FROM login where usr='" + user + "'";
                ResultSet rs = stmt.executeQuery(sql);
                if (rs.next()) {

                    String pswrd = rs.getString("pswd");
                    System.out.println("Login1" + pswrd);

                    System.out.println(str);
                    if (pswrd.equals(str)) {
                        System.out.println("Login successfull");
                        frame_2 frame2 = new frame_2();
                        frame2.setVisible(true);

                    } else {
                        String z = "Password is incorrect";
                        JOptionPane.showMessageDialog(null, z + ".................");
                    }
```

Fig. 14 Main Menu Code of the CUI Program



```
main.java    Login.java    new_user.java ×
        JButton btnNewUser = new JButton("Create User");
        btnNewUser.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                int otp = Integer.parseInt(ot.getText());
                String email = mail.getText();
                char[] pswd = pswdf.getPassword();
                StringBuffer sb = new StringBuffer();
                for (int i = 0; i < pswd.length; i++) {
                    sb.append(pswd[i]);
                }
                String password = sb.toString();
                Connection conn = null;
                Statement stmt = null;
                try {

                    Class.forName("com.mysql.jdbc.Driver");
                    conn = DriverManager.getConnection(DB_URL, USER, PASS);

                    stmt = conn.createStatement();
                    System.out.println("Connection is created successfully:");
                    String sql;
                    String sql1;
                    if (otp == rnd) {
                        sql1 = "Select * from login where usr='" + email + "'";
                        ResultSet rs = stmt.executeQuery(sql1);
                        if (rs.next()) {
                            JOptionPane.showMessageDialog(null, "Username  exist");
                            // textField_1.setText(x);
                        } else {
                            sql = "insert into login(Usr, pswd) values('" + email + "','" + password + "')";
                            stmt.executeUpdate(sql);
                            System.out.println("Record is inserted in the table successfully..................");
                            String z = "User name :-" + email + "\n Password :-" + password;
                            JOptionPane.showMessageDialog(null, z);
                            sende1(email,password);
                            new_user login = new new_user();
                            login.setVisible(true);
                        }
                        rs.close();
                        stmt.close();
                        conn.close();
                    } else {
                        JOptionPane.showMessageDialog(null, "OTP Incorrect");
                    }
```

Fig. 15 Main Menu Code of the CUI Program



```
ain.java    Login.java    new_user.java ×
        rpswdf = new JPasswordField();
        rpswdf.setBounds(1331, 431, 148, 35);
        contentPane.add(rpswdf);

        JButton btnNewButton = new JButton("Send OTP ");
        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                String email = mail.getText();
                char[] pswd = pswdf.getPassword();
                char[] rpswd = rpswdf.getPassword();
                StringBuffer sb = new StringBuffer();
                for (int i = 0; i < pswd.length; i++) {
                    sb.append(pswd[i]);
                }
                String password = sb.toString();
                StringBuffer sb1 = new StringBuffer();
                for (int i = 0; i < rpswd.length; i++) {
                    sb1.append(rpswd[i]);
                }
                String rpassword = sb1.toString();
                if (email.trim().isEmpty()) {
                    JOptionPane.showMessageDialog(null, "Email field cannot be empty");
                } else if (password.trim().isEmpty()) {
                    JOptionPane.showMessageDialog(null, "Password field cannot be empty");
                } else if (rpassword.trim().isEmpty()) {
                    JOptionPane.showMessageDialog(null, "Re-enter password field cannot be empty");
                } else {
                    if (password.equals(rpassword)) {

                        Random rand = new Random();
                        int random = rand.nextInt((9999 - 100) + 1) + 10;
                        rnd = random;
                        sende(email, random);
                        JOptionPane.showMessageDialog(null, "OTP send to the mail");
                    } else {
                        JOptionPane.showMessageDialog(null, "Password and Re-entered password are not same");
                        // actionPerformed(rand);

                    }
                }

            }
        });
```

Fig. 16 Main Menu Code of the CUI Program

VI. RESEARCH STUDY WITH OUTPUTS

To find out the more appropriate result of the performance of both the CUI and GUI interfaces we conducted the t-test based upon the data which has been collected during the performance analysis of the program with the help of different software.

t-test: -is a statistical hypothesis test which follows a distribution in null hypothesis.

Formula

$$Two - sampled\ test\ t = \frac{\bar{x}_1 + \bar{x}_2}{s x_1 x_2 \sqrt{{}^{1}/_{n_1} + {}^{1}/_{n_2}}}$$

$$df = n_1 + n_2 - 2$$

t-Test: Two-Sample Assuming Equal Variances		
	20	*43*
Mean	13.75	18
Variance	122.9167	238
Observations	4	4
Pooled Variance	180.4583	
Hypothesized Mean Difference	0	
df	6	
t Stat	-0.44742	
P(T<=t) one-tail	0.335136	
t Critical one-tail	1.94318	
P(T<=t) two-tail	0.670271	
t Critical two-tail	2.446912	

Table 5. Statistical Result of t-test

Based upon the test we got the p value > 0.5 which shows that the CUI interface is 67% better performing and faster and more reliable than the GUI based interface.

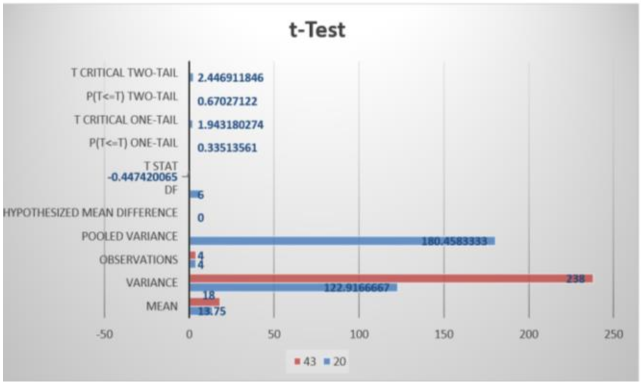


Fig. 17 New User Code of CUI Program(extended)

Graphical User Interface (GUI)

Fig. 18 Performance Utilization of GUI

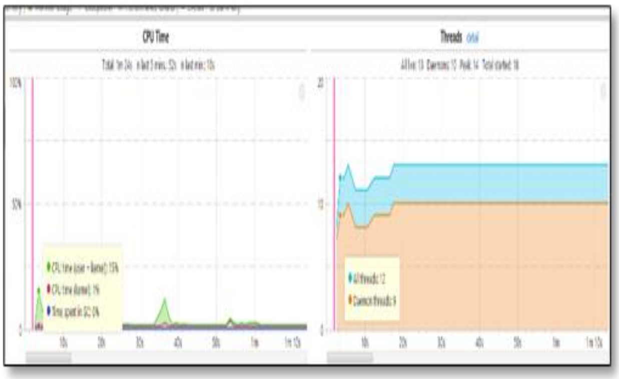


Fig. 19 CPU Utilization of GUI

Character User Interface CUI

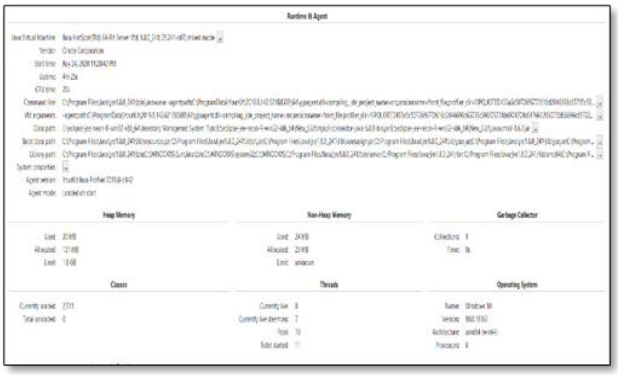


Fig. 20 CPU Utilization of GUI

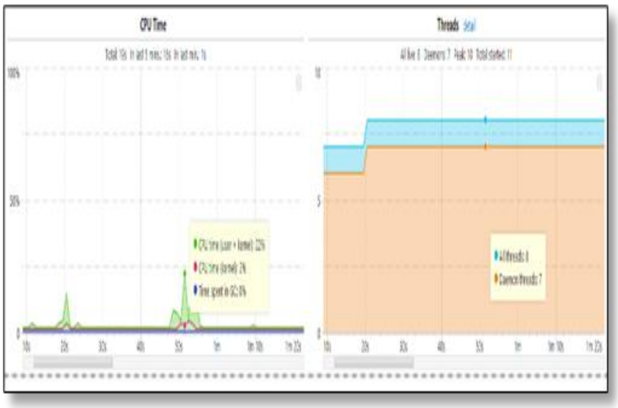


Fig. 21 CPU Utilization of GUI

VII. CONCLUSION

Based upon the entire research work and checking the performance of the program using the Your Kit Java profiler in which we compared the CPU usage of CUI and GUI in which we found out that the program was better performing in the CUI interface , using the same software we also checked the heap memory of the program in which we found out that CUI was using much less heap memory than the GUI interface of the program on further checking the different parameters of the program such as the garbage collection we found out that CUI had the lowest garbage collection over the GUI interface , the CUI interface of the program used much less no of threads than the GUI interface of the program and also the run time of both the CUI and GUI interface when compared we found out that CUI just took 4minutes.25seccond as compared to GUI which took 7minutes.25seconds for the entire program to run .Looking at the results we got from comparing both the CUI and GUI interface we can say that although the GUI interface of the also be changed if needed but still the CUI interface of the program is faster and better performing and will be much more reliable to use .

VIII. ACKNOWLEDGMENT

We have taken the support of various research work done by researchers and published papers. We would like to thank all of them as their research work has provided the insight and expertise to us and their work has helped us to complete my research. I would like to thank to my children and family members who cooperated with me during this pandemic and given us well sufficient time to perform research. We would like to thank various REST videos, as the guidance given by them helped me during the course of writing this research paper. I am immensely grateful to the reviewers for their comments on an earlier version of the manuscript, although any errors are my own and should not tarnish the reputations of these esteemed

REFERENCE

[1] Performance Monitoring Of JAVA Application

[2] Analyzing the Complexity of Java Programs using Object Oriented Software Metrics.

[3] Dynamic analysis of Java program concepts for visualization and profiling

[4] Apache Software Foundation, Log4J

[5] S. Browne, J. Dongarra, N. Garner, G. Ho, P. Mucci, A Portable Programming Interface for Performance Evaluation on Modern Processors, The International Journal of High-Performance Computing Applications 14 (3) (2000)189-204 .

[6] M. Dahm, The Byte Code Engineering Library, 2001.

[7] E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, (1995).

[8] J. Gosling, B. Joy, G. Steele, The Java Language Specification, Addison-Wesley, (1996).

[9] C.A.R. Hoare, Monitors: An Operating System Structuring Concept, Comm. ACM 17(10) (1974) 549-557

[10] M. Henning, S. Vinoski, Advanced CORBA Programming with C++, ISBN 0201379279, Addison-Wesley, (1999).

[11] IBM Research, Jinsight project,(2001).

[12] Intel, VTune Performance Analyzer,(2001).

[13] Intel Corporation, Intel Architecture Software Developer’s Manual Volume 3: System Programming Guide, (1997).

[14] IONA Technologies, Object Oriented Concepts Inc., ORBacus 4 for Java, (2000).

[15] R, Jain, The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling, John Wiley & Sons, (1991).