UECS1104/1144 Object-Oriented Application Development

Group Assignment

Name & Programme : Group 20

Eivonne Goh Yee Fang (Leader)	SE
Chin Kee Jeay	SE
Ker Ding Wei	SE
Max Vun Sheng Khai	SE

	Component	**Missing**	**Poor**	**Average**	**Good**	**Excellent**	**Max Marks**	**Marks Obtained**
2	Introduction						**10**	
3	Class Diagram (Notation used, relationship, multiplicities)						**20**	
4-6	Implementation/ Functionalities						**40**	
	File Handling						**5**	
	Exception Handling						**5**	
	Sample of Input data						**5**	
	Sample Output (Screenshot)						**5**	
	Presentation / General Effort						**10**	
Total							**100**	

Remarks (for Examiner only):

Table Of Content

1. Introduction……………………………………………………………………………..…..1

a. Functional Requirement ……………………………………………………………1

b. Use Case Diagram ….………………………………………………………………2

c. Assumptions…………………………………………………………………………3

d. Workload Allocation…………………………………………………………………4

2. Class Diagram…………………………………………………..……………………..5

3. Source Code…………………………………………………………………………...6-51

4. Sample Input…………………………………………………………………...…51-70

3. Sample Output……………………………………………………..…………..70-88

Introduction

UX Gallery focuses on selling modern fine art, particularly lithographs and photography. They need a system to keep track of the customers and their art purchases, information of artists and their artworks. They also need to keep track of the current inventory. Hence, the system developed by us is a console-based management system. We are using Visual Studio Code to develop this system. Administrator of the UX Gallery will be the only user that uses this system. He/She needs to log in to the system with his/her own password. The administrator may view, add, remove and modify the information of the customers, artists and the artworks. He/She may check the inventory by viewing the list of the artworks.

Functional Requirements

FR01. The system shall allow the user to create an account.

FR02. The system shall allow the user to login to the account.

FR03. The system shall allow the user to logout from the account.

FR04. The system shall allow the user to change his/her password for his/her account.

FR05. The system shall allow the user to upload the information of customers, artists and artworks which are newly registered.

FR06. The system shall allow the user to remove the details of customers, artists, and artworks.

FR07. The system shall allow the user to modify the detail of customers, artists, and artworks.

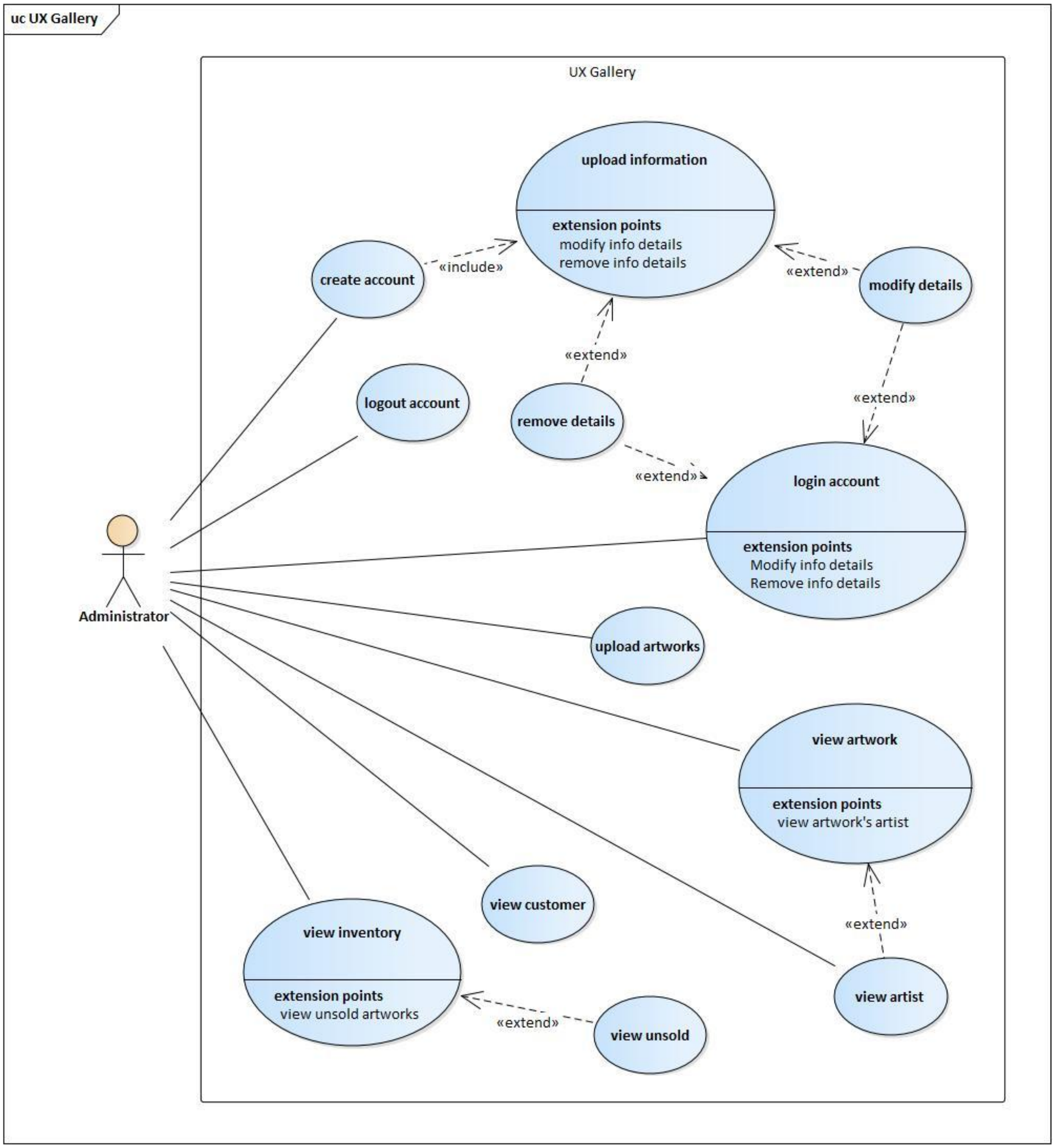
FR08. The system shall allow the user to view the information of customers and artists.

FR09. The system shall allow the user to view the detail of artwork according to different categories.

FR10. The system shall allow the user to view the details of inventory including the purchase price of the artwork and the selling price when sold.

FR11. The System shall allow the user to view the list of unsold artworks.

Use case diagram



Assumptions

1. Administrator is the only user of this management system.

2. User must use his/her own password to login to his/her account.

3. The username and the password of the user is case sensitive.

4. Customers need to select their artist preference when they register with the UX gallery to be a customer or else their artist preferences will be recorded as no artist preference.

5. Each customer can have only one artist preference.

6. The system will only show the purchase price for the unsold artworks when the user searches the list of artworks.

7. The system will force the user to create an account before entering the management system if there is no admin account record file.

8. The artwork is categorized by its artist or the status of being sold.

9. The user may check the inventory through the list of artwork.

10. The user will only add unsold artwork to the management system.

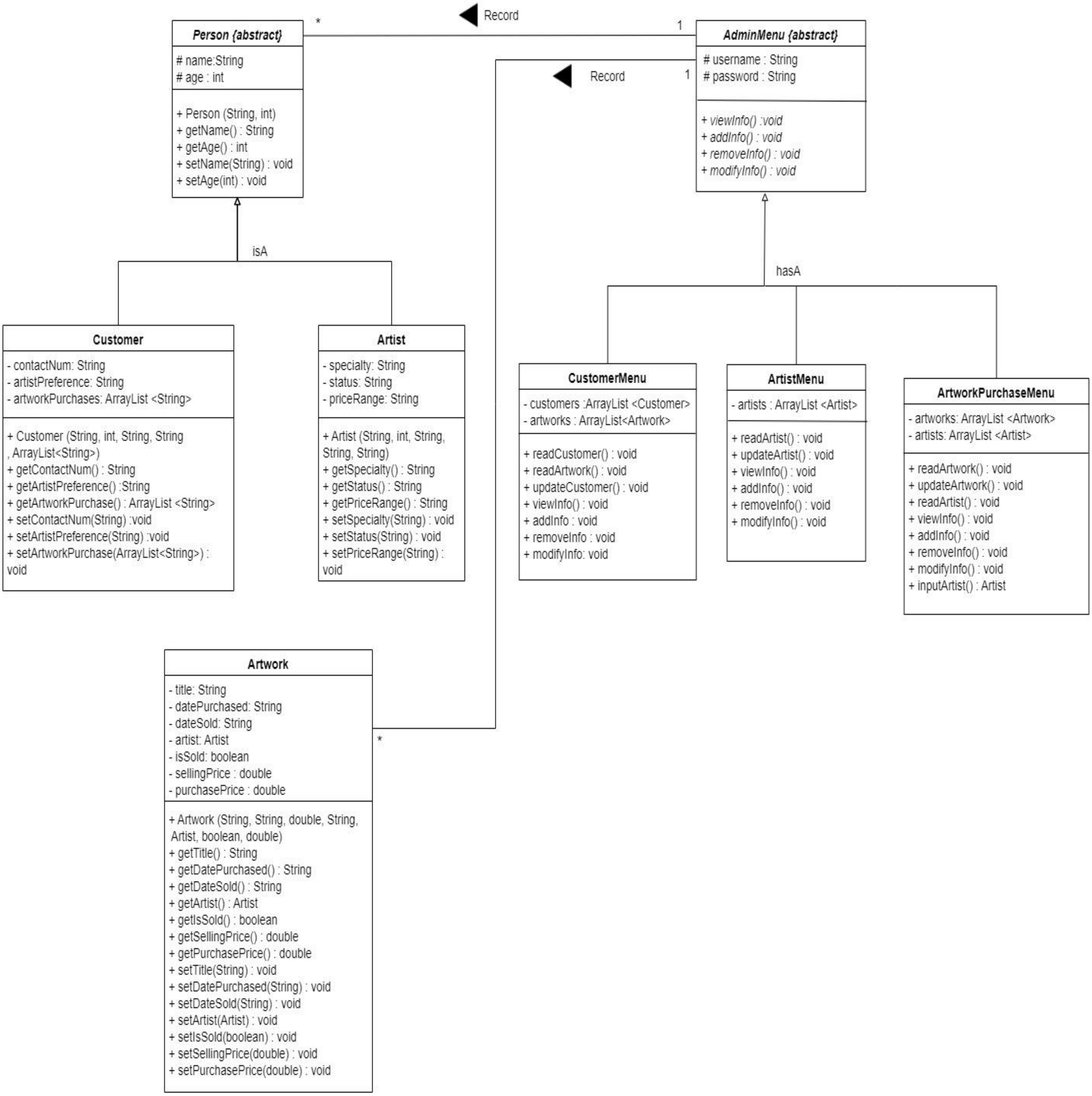
11. The user is not permitted to change the status of the artwork from SOLD to UNSOLD.

12. Right after the user enters a new artwork purchase for a customer, the artwork status is changed manually and the artwork selling price is entered manually by the user.

Workload Allocation

Name of Student	Workload
Eivonne Goh Yee Fang	Allocate workload, set deadline, write report (Introduction), draw the class diagram, develop the application, compile all the work done
Chin Kee Jeay	Develop the application and do all the test cases to assure the quality of the system.
Ker Ding Wei	Write report (Functional requirement, assumption), develop the application, draw the class diagram.
Max Vun Sheng Khai	Write report (Functional requirement, assumption, draw use case diagram) , create sample input and sample output

Class Diagram



Source Code

Admin Menu Class

```
public abstract class AdminMenu {

   protected String username, password;

   public abstract void viewInfo();
   public abstract void addInfo() throws Exception;
   public abstract void removeInfo();
   public abstract void modifyInfo();
}
```

App Class

```
import java.util.*;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.PrintWriter;

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

      Scanner input = new Scanner(System.in);

      System.out.println("\t\t___________________________________________");
      System.out.println("\t\t|                                         |");
      System.out.println("\t\t|                                         |");
      System.out.println("\t\t|     UX Gallery Management System        |");
      System.out.println("\t\t|                                         |");
      System.out.println("\t\t|_________________________________________|");

      loginAccount();
      while (true) {
         System.out.println("\t\tHome");
         System.out.println("______________________________________________");
             System.out.println("1. Customer Menu\n2. Artist Menu\n3. Artwork Menu\n4. Reset
Account\n5. Logout");
         System.out.print("Enter option: ");
         String option = input.nextLine();
         if (option.equals("1")) {
            CustomerMenu customerMenu = new CustomerMenu();
            customerMenu.readCustomer();
            System.out.println("\n\t\tCustomer Menu");
            System.out.println("______________________________________________");
```

```
                     System.out.println("1. View Customer\n2. Add Customer\n3. Modify Customer\n4.
Remove Customer\n5. Back to home");
                openMenu(customerMenu);
                customerMenu.updateCustomer();
            }
            else if (option.equals("2")){
                ArtistMenu artistMenu = new ArtistMenu();
                artistMenu.readArtist();
                System.out.println("\n\t\tArtist Menu");
                System.out.println("_____________________________________________");
                        System.out.println("1. View Artist\n2. Add Artist\n3. Modify Artist\n4. Remove
Artist\n5. Back to home");
                openMenu(artistMenu);
                artistMenu.updateArtist();
            }
            else if (option.equals("3")){
                ArtworkPurchaseMenu artworkMenu = new ArtworkPurchaseMenu();
                artworkMenu.readArtwork();
                artworkMenu.readArtist();
                System.out.println("\n\t\tArtwork Purchase Menu");
                System.out.println("_____________________________________________");
                        System.out.println("1. View Artwork\n2. Add Artwork\n3. Modify Artwork\n4.
Remove Artwork\n5. Back to home");
                openMenu(artworkMenu);
                artworkMenu.updateArtwork();
            }
            else if (option.equals("4")){
                        System.out.println("\n\t\tReset Account");
                System.out.println("_____________________________________________");
                resetAccount();
            }
            else if (option.equals("5")){
                break;
            }
            else {
                System.out.println("Invalid Option.\n");
            }
        }
    }



    public static void loginAccount()
    {
        File file = new File("OwnerAccount.txt");
        try {
```

```
        Scanner infile = new Scanner(file);
        String name = infile.nextLine();
        String password = infile.nextLine();
        Scanner input = new Scanner(System.in);
        while (true) {
            System.out.print("Enter username: ");
            String username = input.nextLine();
            System.out.print("Enter Password: ");
            String loginPassword = input.nextLine();
            if (username.equals(name) && loginPassword.equals(password)) {
                System.out.println("\nLogin successfully\n");
                break;
            } else
                System.out.println("Incorrect username or password! Please try again...\n");
        }
    }
    catch (Exception e) {
        System.out.println("You have not register account.");
        registerAccount();
    }
}

public static void registerAccount()
{
        File file = new File("OwnerAccount.txt");
    try {
        PrintWriter outfile = new PrintWriter(file);
            Scanner input = new Scanner(System.in);
        System.out.print("Enter new username: ");
        String username = input.nextLine();
        System.out.print("Enter new password: ");
        String loginPassword = input.nextLine();
            outfile.println(username + "\n" + loginPassword);
        outfile.close();
                    }
    catch (FileNotFoundException e) {
        System.out.println("File not found!");
        ;
    }
    catch (Exception e) {
                                System.out.println("An error has occured!");;
                    }
    System.out.println("\nAccount created successfully\n");
}
public static void resetAccount()
{
    File file = new File("OwnerAccount.txt");
    try {
        Scanner sc = new Scanner(file);
        String oldUserName = sc.nextLine();
        String oldPassword = sc.nextLine();
        Scanner input = new Scanner(System.in);
        while (true) {
            System.out.print("Enter old password: ");
            String oldPasswordTest = input.nextLine();
            if(oldPasswordTest.equals(oldPassword))
            {
                PrintWriter outfile = new PrintWriter(file);
                System.out.print("Enter new username: ");
                String username = input.nextLine();
                System.out.print("Enter new password: ");
                String loginPassword = input.nextLine();
                outfile.println(username + "\n" + loginPassword);
                outfile.close();
                System.out.println("\nAccount reset successfully\n");
                break;
            }
            else
            {
                System.out.println("Old password is incorrect! Please try again...\n");
            }
        }
    }
    catch (FileNotFoundException e) {
            System.out.println("File not found!");
    }
    catch (Exception ex) {
        System.out.println("An error has occured");
    }
}

public static void openMenu(AdminMenu menu) throws Exception
{
    Scanner input = new Scanner(System.in);
    System.out.print("Enter choice: ");
    String choice = input.nextLine();
    if (choice.equals("1"))
        menu.viewInfo();
    else if (choice.equals("2"))
        menu.addInfo();
    else if (choice.equals("3"))
        menu.modifyInfo();
        else if (choice.equals("4"))
            menu.removeInfo();
        else if (choice.equals("5"))
            System.out.println("Exit successfully\n");
        else
            System.out.println("Invalid Choice! Please try again...\n");

    }
}
```

Artist Class

```
public class Artist extends Person{

    private String specialty, status, priceRange;

    public String getSpecialty()
    {
        return specialty;
    }

    public void setSpecialty(String specialty)
    {
        this.specialty = specialty;
    }

    public String getStatus()
    {
        return status;
    }

    public void setStatus(String status)
    {
        this.status = status;
    }

    public String getPriceRange()
    {
        return priceRange;
    }

    public void setPriceRange(String priceRange)
    {
        this.priceRange = priceRange;
    }

    public Artist(String name, int age, String specialty, String status, String priceRange)
    {
        super(name, age);
        this.specialty = specialty;
        this.status = status;
        this.priceRange = priceRange;
    }
}
```

Artist Menu Class

```
import java.util.*;
import java.io.File;
import java.io.PrintWriter;

public class ArtistMenu extends AdminMenu {

    private ArrayList<Artist> artists = new ArrayList<Artist>();

    public void readArtist() {
        try {
            Scanner infile = new Scanner(new File("Artist.txt"));

            while (infile.hasNextLine()) {
                String name = infile.nextLine();
                int age = Integer.parseInt(infile.nextLine());
                String specialty = infile.nextLine();
                String status = infile.nextLine();
                String priceRange = infile.nextLine();
                artists.add(new Artist(name, age, specialty, status, priceRange));
            }
            infile.close();
        } catch (Exception ex) {
            System.out.println(ex);
        }
    }

    public void updateArtist() {
        try {
            PrintWriter outfile = new PrintWriter("Artist.txt");

            for (Artist artist : artists) {
                outfile.println(artist.getName().toUpperCase() + "\n" + artist.getAge() + "\n"
                            + artist.getSpecialty().toUpperCase() + "\n" + artist.getStatus().toUpperCase() +
"\n"
                        + artist.getPriceRange());
            }
            outfile.close();
        } catch (Exception ex) {
            System.out.println(ex);
        }
    }

    @Override
    public void viewInfo() {
        System.out.println("\t\tArtist List");
        System.out.println("____________________________________________\n");
        for (int i = 0; i < artists.size(); i++) {
            System.out.println("\nArtist " + (i + 1));
            System.out.println("____________________");
            System.out.println("Name: " + artists.get(i).getName());
            System.out.println("Age: " + artists.get(i).getAge());
            System.out.println("Specialty: " + artists.get(i).getSpecialty());
            System.out.println("Status: " + artists.get(i).getStatus());
            System.out.println("Price Range of Artwork: " + artists.get(i).getPriceRange());
        }
        System.out.println();
    }

    @Override
    public void addInfo() {
        System.out.println("\t\tArtist Registration");
        System.out.println("____________________________________________");
        Scanner input = new Scanner(System.in);

        boolean exist = false;

        System.out.println("Enter '0' to back to the HOME PAGE");
        System.out.print("Enter Artist Name: ");
        String name = input.nextLine();

        if(name.contentEquals("0"))
            exist = true;
        for (int i = 0; i < artists.size(); i++) {
            if (name.equalsIgnoreCase(artists.get(i).getName())) {
                exist = true;
            }
        }

        if (!exist) {
            int age = 0;
            do {
                try {
                    System.out.print("Enter Artist Age (OR age at deceased): ");
                    age = Integer.parseInt(input.nextLine());
                } catch (Exception ex) {
                    System.out.println("Artist age must larger than 0.");
                }
            } while (age <= 0);

            System.out.print("Enter Artist Specialty: ");
            String specialty = input.nextLine();

            String status;
            do {
                System.out.print("Enter Status (Alive/Deceased): ");
                status = input.nextLine();

                if (!status.equalsIgnoreCase("ALIVE") && !status.equalsIgnoreCase("Deceased")) {
                                System.out.println("\nArtist Status should be <ALIVE> or <DECEASED>
only.\n");
                }
            } while (!status.equalsIgnoreCase("ALIVE") && !status.equalsIgnoreCase("Deceased"));

            boolean validRange = false;
            String priceRange = "0";
            double min = -1;
            double max = -1;
            do {
                do {
                    try {
                        System.out.print("Enter Artwork Minimum Price: ");
                        min = Double.parseDouble(input.nextLine());
                        System.out.print("Enter Artwork Maximum Price: ");
                        max = Double.parseDouble(input.nextLine());
                        priceRange = min + "-" + max;
                        validRange = true;
                    } catch (Exception ex) {
                        System.out.println("INVALID INPUT.");
                    }
                } while (!validRange);
            } while (min < 0 || max < 0 || max < min);

            artists.add(new Artist(name, age, specialty, status, priceRange));
            System.out.println("\nArtist registered successfully.\n");
        }
        else if (name.contentEquals("0"))
            System.out.println("\nYou have back to the HOME PAGE\n");
        else
```

```
    System.out.println("\nArtist already registered.\n");
}

@Override
public void removeInfo() {
  System.out.println("\t\tArtist Deletion");
  System.out.println("__________________________________________________");
  Scanner input = new Scanner(System.in);

  System.out.println("Enter '0' to back to the HOME PAGE");
  System.out.print("Enter Artist name to be remove: ");
  String name = input.nextLine();
  if(name.equals("0"))
    System.out.println("\nYou have back to the HOME PAGE\n");
  else {
  int oriSize = artists.size();
  for (int i = 0; i < artists.size(); i++) {
    if (name.equalsIgnoreCase(artists.get(i).getName())) {
      artists.remove(i);
      System.out.println("Artist removed from record successfully.");
      i = artists.size();
    }
  }
  if(oriSize == artists.size())
  {
    System.out.println("\nArtist not Found! Please try again...\n");
  }}
}

@Override
public void modifyInfo() {
    int numArtist = artists.size();
    boolean exit =false; //exit turn true when user wants to back to HOME PAGE
  Scanner input = new Scanner(System.in);
  do {


  System.out.println("\t\tArtist Information Update");
  System.out.println("__________________________________________________");



  System.out.println("Enter '0' to back to the HOME PAGE");
  System.out.print("Enter Artist name to be modified: ");
  String name = input.nextLine();
```

```
   if(name.equals("0"))
        {
                exit = true;
                System.out.println("\nYou are back to the HOME PAGE\n");
                break;

        }


for (int i = 0; i < artists.size(); i++) // find the artist info that the user wants to modify
{
   if (name.equalsIgnoreCase(artists.get(i).getName()))
   {
        numArtist = i;
   }
}
if(numArtist == artists.size()) //if the artist not found
        {
                System.out.println("Artist Not Found !\n");
        }
}while(numArtist == artists.size());

String answer;
while(!exit){
        System.out.println("\nWhat information do you want to modify?");
        System.out.println("1. Name");
        System.out.println("2. Age");
        System.out.println("3. Specialty");
        System.out.println("4. Status (Alive/Deceased)");
        System.out.println("5. Price Range");

        int choice = input.nextInt();
        switch (choice)
{
        case 1 :
                String confirm;
                String changedName;

                input.nextLine();//avoid bug
                do {
                boolean exist = false;
                System.out.println("Please enter the name:");
                changedName = input.nextLine();
                for (int i = 0; i < artists.size(); i++) {
                if (changedName.equalsIgnoreCase(artists.get(i).getName())) {
                   exist = true;
```

```
            }
           }
            if(!exist)
            {
            System.out.println("Please confirm that the name entered is :" + changedName.toUpperCase());//let the user to re-confirm his/her input to reduce the chance of making mistake
            System.out.println("Press any key other than '0' to continue...");
            System.out.println("Press 0 to re-enter the name...");
            confirm = input.next();
            if(confirm.equals("0"))
                input.nextLine();
            }
            else
            {
                System.out.println("Artist is already exist! Please try again...\n");
                confirm = "0";
            }
            }while(confirm.equals("0"));
            artists.get(numArtist).setName(changedName);
            break;
        case 2:
            boolean loop;
            int changedAge;
            String confirm1;
            do {
                loop = false;
            input.nextLine();//avoid bug
        try {
            do {
            System.out.println("Please enter the age:");
            changedAge = input.nextInt();
            System.out.println("Please confirm that the age entered is :" + changedAge);//let the user to re-confirm his/her input to reduce the chance of making mistake
            System.out.println("Press any key other than '0' to continue...");
            System.out.println("Press 0 to re-enter the age...");
            confirm1 = input.next();
            if(confirm1.equals("0"))
                input.nextLine();
            else if(changedAge < 0)
            {
                loop = true;
                System.out.println("Age should be greater than 0.\n");
            }
            }while(confirm1.contentEquals("0"));
                artists.get(numArtist).setAge(changedAge);
            }catch(InputMismatchException ex)
            {
                System.out.println("Invalid input! Please try again...\n");
                loop = true;
            }
            }while(loop);


            break;
        case 3:
            String changedSpecialty;
            String confirm2;
            do {
            input.nextLine();//avoid bug
            System.out.println("Please enter the new specialty:");
            changedSpecialty = input.nextLine();
            System.out.println("Please confirm that the new specialty entered is :" +
changedSpecialty); //let the user to re-confirm his/her input to reduce the chance of making
mistake
            System.out.println("Press any key other than '0' to continue...");
            System.out.println("Press 0 to re-enter the specialty...");
            confirm2 = input.next();
            }while(confirm2.contentEquals("0"));
            artists.get(numArtist).setSpecialty(changedSpecialty);
            break;
        case 4:
            String changedStatus;
            String confirm3;
            do {
            input.nextLine();//avoid bug
            do {
            System.out.println("Please enter the status (Alive/Deceased):");
            changedStatus = input.nextLine();
            if(!changedStatus.equalsIgnoreCase("Alive")&&
!changedStatus.equalsIgnoreCase("Deceased"))
                System.out.println("\nArtist Status should be <ALIVE> or
<DECEASED> only.\n");
            }while(!changedStatus.equalsIgnoreCase("Alive")&&
!changedStatus.equalsIgnoreCase("Deceased"));
            System.out.println("Please confirm that the new status entered is :" +
changedStatus); //let the user to re-confirm his/her input to reduce the chance of making mistake
            System.out.println("Press any key other than '0' to continue...");
            System.out.println("Press 0 to re-enter the status...");
            confirm3 = input.next();
            }while(confirm3.contentEquals("0"));
            artists.get(numArtist).setStatus(changedStatus);
```

```
                  break;
            case 5:
                  boolean validRange = false;
      String priceRange = "0";
      double min = -1;
      double max = -1;

      input.nextLine();//avoid bug
      do {
        try {
          System.out.print("Enter Artwork Minimum Price: ");
          min = Double.parseDouble(input.nextLine());
          System.out.print("Enter Artwork Maximum Price: ");
          max = Double.parseDouble(input.nextLine());
          priceRange = min + "-" + max;
          if(min < 0 || max < 0 || max < min)
          {
            System.out.println("Price should be greater than '0' and maximum price should be
greater than minimum price.\n");
          }
          else
            validRange = true;
          } catch (Exception ex) {
            System.out.println("INVALID INPUT.");
          }
      } while (!validRange);

      artists.get(numArtist).setPriceRange(priceRange);
    default :
      System.out.println("Invalid input! Please try again...");
            }
            boolean yesNoExit = false;
    System.out.println("Modified successfully !");
    while (!yesNoExit) {

       System.out.println("Do you wish to perform any action? (y/n)"); //ask the user whether to
continue or back to the HOME PAGE
      answer = input.next();
      if(answer.equalsIgnoreCase("n"))
      {
        yesNoExit = true;
        exit = true;
        System.out.println("\nYou are back to the HOME\n");
      }
      else if(answer.equalsIgnoreCase("y"))
      {
        yesNoExit = true;
      }
      else {
        System.out.println("Invalid input! Please try again...");
      }
    }
      }
 }
}
```

Artwork Class

```
public class Artwork {

  private String title, datePurchased, dateSold;
  private double sellingPrice, purchasePrice;
  private Artist artist;
  private boolean isSold;


  public double getSellingPrice()
  {
      return sellingPrice;
  }

  public void setSellingPrice(double sellingPrice)
  {
      this.sellingPrice = sellingPrice;
  }

  public void setPurchasePrice(double purchasePrice)
  {
      this.purchasePrice = purchasePrice;
  }
  public double getPurchasePrice()
  {
      return purchasePrice;
  }
  public String getTitle()
  {
    return title;
  }

  public void setTitle(String title)
  {
    this.title = title;
  }
```

```
    public String getDatePurchased()
    {
        return datePurchased;
    }

    public void setDatePurchased(String datePurchased)
    {
        this.datePurchased = datePurchased;
    }

    public String getDateSold()
    {
        return dateSold;
    }

    public void setDateSold(String dateSold)
    {
        this.dateSold = dateSold;
    }

    public Artist getArtist()
    {
        return artist;
    }

    public void setArtist(Artist artist)
    {
        this.artist = artist;
    }

    public boolean getIsSold()
    {
        return isSold;
    }

    public void setIsSold(boolean isSold)
    {
        this.isSold = isSold;
    }

    public Artwork(String title, String datePurchased,double purchasePrice, String dateSold, Artist
artist, boolean isSold, double sellingPrice)
    {
        this.title = title;
        this.datePurchased = datePurchased;
        this.purchasePrice =purchasePrice;
        this.dateSold = dateSold;
        this.artist = artist;
        this.isSold = isSold;
        this.sellingPrice = sellingPrice;
    }

}
```

Artwork Purchase Menu Class

```
import java.util.*;
import java.util.zip.Inflater;
import java.io.File;
import java.io.PrintWriter;

public class ArtworkPurchaseMenu extends AdminMenu {

    private ArrayList<Artwork> artworks = new ArrayList<Artwork>();
    private ArrayList<Artist> artists = new ArrayList<Artist>();

    public void readArtwork() {
        try {
            Scanner infile = new Scanner(new File("Artwork.txt"));

            while (infile.hasNextLine()) {
                String title = infile.nextLine();
                String datePurchased = infile.nextLine();
                String dateSold = infile.nextLine();
                double purchasePrice =Double.parseDouble(infile.nextLine());

                String name = infile.nextLine();
                int age = Integer.parseInt(infile.nextLine());
                String specialty = infile.nextLine();
                String status = infile.nextLine();
                String priceRange = infile.nextLine();
                Artist artist = new Artist(name, age, specialty, status, priceRange);

                boolean isSold = Boolean.parseBoolean(infile.nextLine());
                double sellingPrice = Double.parseDouble(infile.nextLine());

                            artworks.add(new Artwork(title, datePurchased,purchasePrice, dateSold, artist,
isSold,sellingPrice));
            }
            infile.close();
        } catch (Exception ex) {
            System.out.println(ex);
        }
    }

    public void updateArtwork() {
        try {
            PrintWriter outfile = new PrintWriter("Artwork.txt");

            for (Artwork artwork : artworks) {
                outfile.println(artwork.getTitle().toUpperCase() + "\n" +
                artwork.getDatePurchased().toUpperCase() + "\n" +
                                                                              artwork.getDateSold().toUpperCase()    +    "\n"
+Double.toString(artwork.getPurchasePrice())+ "\n"+
                    artwork.getArtist().getName().toUpperCase() + "\n" + artwork.getArtist().getAge() +
"\n"
                                                        +  artwork.getArtist().getSpecialty().toUpperCase()  +  "\n"  +
artwork.getArtist().getStatus().toUpperCase() + "\n"
                                   + artwork.getArtist().getPriceRange() + "\n" + artwork.getIsSold()+ "\n" +
Double.toString(artwork.getPurchasePrice()));
            }
            outfile.close();

        } catch (Exception ex) {
            System.out.println(ex);
        }
    }

    public void readArtist() {
        try {
            Scanner infile = new Scanner(new File("Artist.txt"));

            while (infile.hasNextLine()) {
                String name = infile.nextLine();
                int age = Integer.parseInt(infile.nextLine());
                String specialty = infile.nextLine();
                String status = infile.nextLine();
                String priceRange = infile.nextLine();
                artists.add(new Artist(name, age, specialty, status, priceRange));
            }
            infile.close();
        } catch (Exception ex) {
            System.out.println(ex);
        }
    }

    @Override
    public void viewInfo() {
```

```
   boolean exit = false;
   do {
   try {
   Scanner input = new Scanner(System.in);
System.out.println("\t\tArtwork List");
System.out.println("__________________________________________________\n");
System.out.println("How you wants to view the artwork list?");
System.out.println("1. View All Artwork List");
System.out.println("2. View Artwork List By Categories");
System.out.println("(Enter '0' to return to the HOME PAGE)");
int choice = input.nextInt();
String artist;
if(choice == 1)
{
    for (int i = 0; i < artworks.size(); i++) {
       System.out.println("\nArtwork " + (i + 1));
       System.out.println("________________________");
       System.out.println("Title: " + artworks.get(i).getTitle());
       System.out.println("Artist: " + artworks.get(i).getArtist().getName());
       System.out.println("Purchasing Date: " + artworks.get(i).getDatePurchased());
       System.out.println("Purchase Price: "+ artworks.get(i).getPurchasePrice());
       if (artworks.get(i).getIsSold()) {
          System.out.println("Status: Sold");
          System.out.println("Sold Date: " + artworks.get(i).getDateSold());
          System.out.println("Selling Price: " + artworks.get(i).getSellingPrice());
       }
       else
          System.out.println("Status: Unsold");
    }
    System.out.println();
    exit = true;
}
else if (choice == 2)
{
   boolean exit2 = false;
   do {
            try {

   System.out.println("\nWhat is the category you wish to search for?");
   System.out.println("1. Artist");
   System.out.println("2. Status of being sold");
   int category = input.nextInt();
   if(category == 1)
   {
            input.nextLine();
            System.out.println("Enter the artist name :");
```

```
        artist = input.nextLine();
        int count =0;
        for(int i =0; i < artworks.size();i ++)
        {
                if(artist.equalsIgnoreCase(artworks.get(i).getArtist().getName()))
                                { System.out.println("\nArtwork " + (i + 1));
          System.out.println("_____________________");
          System.out.println("Title: " + artworks.get(i).getTitle());
          System.out.println("Artist: " + artworks.get(i).getArtist().getName());
          System.out.println("Purchasing Date: " + artworks.get(i).getDatePurchased());
          System.out.println("Purchase Price: "+ artworks.get(i).getPurchasePrice());
          if (artworks.get(i).getIsSold()) {
             System.out.println("Status: Sold");
             System.out.println("Sold Date: " + artworks.get(i).getDateSold());
             System.out.println("Selling Price: "+ artworks.get(i).getSellingPrice());
             exit2 = true;
                        exit = true;
          }
          else
          {
                System.out.println("Status: Unsold");
                exit2 = true;
                        exit = true;
          }

         System.out.println();
                                }
                else
                        count++;

        }
        if(count == artworks.size())
                System.out.println("Artist is not found! Please try again...\n");
        else
                exit = true;

}
else if (category == 2)
{
        boolean exit1 = false;
        do{
                System.out.println("Sold / Unsold");
        String isSold = input.next();
        if (isSold.equalsIgnoreCase("Sold"))
                {
                for(int i =0; i < artworks.size(); i ++)
        {
            if(artworks.get(i).getIsSold())
            {
                System.out.println("\nArtwork " + (i + 1));
     System.out.println("____________________");
     System.out.println("Title: " + artworks.get(i).getTitle());
     System.out.println("Artist: " + artworks.get(i).getArtist().getName());
     System.out.println("Purchasing Date: " + artworks.get(i).getDatePurchased());
     System.out.println("Purchase Price: " + artworks.get(i).getPurchasePrice());
     System.out.println("Status: Sold");
     System.out.println("Sold Date: " + artworks.get(i).getDateSold());
     System.out.println("Selling Price: "+ artworks.get(i).getSellingPrice());
     System.out.println();
            }

        }
        exit1 = true;
        exit2 = true;
        exit = true;
        }
    else if (isSold.equalsIgnoreCase("Unsold"))
    {
        for(int i =0; i < artworks.size(); i ++)
        {
            if(!artworks.get(i).getIsSold())
            {
                System.out.println("\nArtwork " + (i + 1));
     System.out.println("____________________");
     System.out.println("Title: " + artworks.get(i).getTitle());
     System.out.println("Artist: " + artworks.get(i).getArtist().getName());
     System.out.println("Purchasing Date: " + artworks.get(i).getDatePurchased());
     System.out.println("Purchase Price: "+ artworks.get(i).getPurchasePrice());
     System.out.println("Status: Unsold");
     System.out.println();
            }
        }
        exit1 = true;
        exit2 = true;
        exit = true;
    }
    else
        System.out.println("Please just enter 'sold' or 'unsold'...\n");
    }while(!exit1);

}
else
```

```
               System.out.println("Only '1' and '2' are allowed to input! Please try again...");
      }catch(InputMismatchException ex)
               {
                        System.out.println("Invalid input!\n");
                        exit2 = false;
                        input.nextLine(); //avoid bug
      }
      }while(!exit2);


      }
   else if (choice == 0)
      exit = true;
   else
   {
      System.out.println("Only '1' and '2' are allowed to input! Please try again...\n");
   }
      }catch(InputMismatchException ex)
      {
               System.out.println("Invalid input!\n");
               exit = false;
      }
      }while(!exit);


}

@Override
public void addInfo() {
   System.out.println("\t\tArtwork Registration");
   System.out.println("_____________________________________________");
   Scanner input = new Scanner(System.in);

   boolean exist = false;

   System.out.println("Enter '0' to back to the HOME PAGE");
   System.out.print("Enter Artwork Title: ");
   String title = input.nextLine();

   if(title.contentEquals("0"))
      exist = true;

   for (int i = 0; i < artworks.size(); i++) {
      if (title.equalsIgnoreCase(artworks.get(i).getTitle())) {
         exist = true;
      }
```

```
}

if (!exist) {
    String status;
    do {
        System.out.print("Enter Artwork Status (Sold/Unsold): ");
        status = input.nextLine();
        if (!status.equalsIgnoreCase("SOLD") && !status.equalsIgnoreCase("UNSOLD"))
            System.out.println("Invalid Data Entered.");
    } while (!status.equalsIgnoreCase("SOLD") && !status.equalsIgnoreCase("UNSOLD"));
    boolean isSold;
    double sellingPrice = 0.0;
    if (status.equalsIgnoreCase("SOLD"))
    {
                isSold = true;
                System.out.println("What is the selling price?: ");
                sellingPrice = input.nextDouble();
    }

    else
    {
                isSold = false;
                System.out.println("Press enter to continue...");
    }


    boolean validDate = false;
    String datePurchased = "INVALID";
    String dateSold = "INVALID";
    int pYear = 0;
    int pMonth = 0;
    int pDay = 0;
    int sYear = 0;
    int sMonth = 0;
    int sDay = 0;
    double purchasePrice = 0.0;
    do {
        try {

                input.nextLine();//avoid bug
            System.out.print("Enter Purchasing Year: ");
            pYear = Integer.parseInt(input.nextLine());

            System.out.print("Enter Purchasing Month: ");
            pMonth = Integer.parseInt(input.nextLine());
```

```
System.out.print("Enter Purchasing Day: ");
pDay = Integer.parseInt(input.nextLine());
validDate = true;

System.out.println("Enter Purchase Price: ");
purchasePrice = input.nextDouble();

if (isSold) {
   validDate = false;

   input.nextLine(); //avoid bug
   System.out.print("Enter Sold Year: ");
   sYear = Integer.parseInt(input.nextLine());

   System.out.print("Enter Sold Month: ");
   sMonth = Integer.parseInt(input.nextLine());

   System.out.print("Enter Sold Day: ");
   sDay = Integer.parseInt(input.nextLine());

   if(pYear != sYear)
   {
            if(pYear > sYear)
                     System.out.println("Invalid Date Entered.");
            else
            {
                     validDate = true;
         datePurchased = pDay + "/" + pMonth + "/" + pYear;
         dateSold = sDay + "/" + sMonth + "/" + sYear;
            }
   }
   else
   {
            if(pMonth != sMonth)
            {
                     if(pMonth > sMonth)
                              System.out.println("Invalid Date Entered.");
                     else
                     {
                              validDate = true;
            datePurchased = pDay + "/" + pMonth + "/" + pYear;
            dateSold = sDay + "/" + sMonth + "/" + sYear;
                     }

            }
            if(pMonth == sMonth)
```

```
                  {
                        if(pDay > sDay)
                              System.out.println("Invalid Date Entered.");
                        else
                        {
                              validDate = true;
                  datePurchased = pDay + "/" + pMonth + "/" + pYear;
                  dateSold = sDay + "/" + sMonth + "/" + sYear;
                        }

                  }
            }
          }
          else
          {
            validDate = true;
            datePurchased = pDay + "/" + pMonth + "/" + pYear;
          }
        }catch (Exception ex) {
          System.out.println("Invalid Date Entered");
        }
      } while (!validDate);

      Artist artist = inputArtist();


        artworks.add(new Artwork(title, datePurchased, purchasePrice, dateSold, artist, isSold,
sellingPrice));
      System.out.println("\nArtwork registered successfully.\n");
    }
    else if (title.contentEquals("0"))
      System.out.println("\nYou have back to the HOME PAGE\n");
    else
      System.out.println("Artwork already registered.\n");
  }

  @Override
  public void removeInfo() {
    System.out.println("\t\tArtwork Deletion");
    System.out.println("_____________________________________________");
    Scanner input = new Scanner(System.in);

    System.out.println("Enter '0' to back to the HOME PAGE");
    System.out.print("Enter artwork name to be remove: ");
    String title = input.nextLine();
```

```
   if(title.equals("0"))
      System.out.println("\nYou have back to the HOME PAGE\n");
   else {
   int oriSize = artworks.size();
   for (int i = 0; i < artworks.size(); i++) {
      if (title.equalsIgnoreCase(artworks.get(i).getTitle())) {
         artworks.remove(i);
         System.out.println("Artwork removed from record successfully.");
         i = artworks.size();
      }
   }
   if(oriSize == artworks.size())
   {
      System.out.println("\nArtwork not Found! Please try again...\n");
   }
   }
}

@Override
public void modifyInfo() {
      int numArtwork = artworks.size();
      boolean exit = false; //exit turn true when user wants to back to HOME PAGE
       Scanner input = new Scanner(System.in);
       do {
   System.out.println("\t\tArtwork Information Update");
   System.out.println("_____________________________________________");



   System.out.println("Enter '0' to back to the HOME PAGE");
   System.out.print("Enter Artwork Title: ");
   String title = input.nextLine();

   if(title.equals("0"))
            {
                  exit = true;
                  System.out.println("\nYou are back to the HOME PAGE\n");
                  break;

            }

   for (int i = 0; i < artworks.size(); i++) // find the artwork info that the user wants to modify
   {
      if (title.equalsIgnoreCase(artworks.get(i).getTitle()))
      {
            numArtwork = i;
```

```
    }
   }
    if(numArtwork == artworks.size()) //if the artwork not found
        {
            System.out.println("Artwork Not Found !\n");
        }
        }while(numArtwork == artworks.size());


        String answer;
        while(!exit)
            {

                boolean loop = false; //loop turn true after catch
InputMismatchException or invalid input for the choice
                do {
                    try {

                        System.out.println("\nWhat information do you
want to modify?");
                        System.out.println("1. Title");
                        System.out.println("2. Status");
                        System.out.println("3. Purchase Price");
                        System.out.println("4. Purchase Date");
                        System.out.println("5. Artist");

                        int choice = input.nextInt();
                        switch (choice) {
                            case 1:
                                String confirm;
                                String changedTitle;
                                input.nextLine();//avoid bug
                                do {
                                    boolean exist = false;
                                    System.out.println("Please
enter the title:");
                                    changedTitle =
input.nextLine();
                                    for (int i = 0; i <
artworks.size(); i++) {
                                        if
(changedTitle.equalsIgnoreCase(artworks.get(i).getTitle())) {
                                            exist = true;
                                        }
                                    }
                                    if (!exist) {
```

```
System.out.println("Please confirm that the title entered is :"
                                                +
changedTitle.toUpperCase());

System.out.println("Press any key other than '0' to continue...");

System.out.println("Press 0 to re-enter the title...");
                                        confirm    =
input.next();
                                        if
(confirm.equals("0"))

input.nextLine();
                                    } else {

System.out.println("Artwork is already exist! Please try again...\n");
                                        confirm = "0";
                                    }
                                } while (confirm.equals("0"));

artworks.get(numArtwork).setTitle(changedTitle);

                                break;
                            case 2:
                                String dateSold = null;
                                String status;

                                input.nextLine();//avoid bug
                                if
(!artworks.get(numArtwork).getIsSold()) {

                                    do {

System.out.print("Enter Artwork Status: ");
                                        status    =
input.nextLine();
                                        if
(!status.equalsIgnoreCase("SOLD"))

System.out.println("Artwork status can only change from UNSOLD to SOLD");
                                    }         while
(!status.equalsIgnoreCase("SOLD"));
                                    double sellingPrice = 0.0;
                                    boolean  exit1  =  false;  //
determine whether the input (date of sold) valid or not
```

```
                                                    boolean   exit2;  //  determine
whether the input (selling price) valid or not

                                                    do {
                                                         try {
                                                              do {
                                                                   exit2 =
true;

System.out.println("What is the selling price?: ");


sellingPrice = input.nextDouble();
                                                                   if

(sellingPrice <= 0) {

System.out.println("Price should be greater than zero!");

exit2 = false;
                                                                   }

                                                              }          while
(!exit2);


input.nextLine();//avoid bug

System.out.print("Enter Sold Year: ");
                                                              int    sYear    =
Integer.parseInt(input.nextLine());


System.out.print("Enter Sold Month: ");
                                                              int   sMonth   =
Integer.parseInt(input.nextLine());


System.out.print("Enter Sold Day: ");
                                                              int    sDay     =
Integer.parseInt(input.nextLine());

                                                              if (sDay < 0 ||
sMonth < 0 || sYear < 0) {
                                                                   exit1  =
false;
System.out.println(

        "\nInvalid input\nPress Enter to re-enter the information for the sold status...");
                                                                                          } else {
                                                                                                  exit1  =
true;

dateSold = sDay + "/" + sMonth + "/" + sYear;

System.out.println("Press enter to continue...");
                                                                                          }

                                                                                  }                   catch
(InputMismatchException ex) {

System.out.println("Invalid Input");

                                                                                  }                   catch
(NumberFormatException ex) {

System.out.println(

"Invalid input\nPress Enter to re-enter the information for the sold status...");

                                                                                  }
                                                                                  input.nextLine();//
avoid bug
                                                                          } while (!exit1);

artworks.get(numArtwork).setDateSold(dateSold);

artworks.get(numArtwork).setSellingPrice(sellingPrice);

artworks.get(numArtwork).setIsSold(true);

                                                                  } else {
                                                                          System.out.println("Artwork
status can only change from UNSOLD to SOLD");
                                                                  }

                                                                  break;

                                                          case 3:
                                                                  double changedPPrice = 0;
                                                                  String confirm1;
```

```
                                        input.nextLine();//avoid bug
                                        do {

                                            try {


System.out.println("Please enter the purchase price:");
                                                changedPPrice          =
input.nextDouble();

                                            }                       catch
(InputMismatchException ex) {

System.out.println("Invalid input! System can't record the price");
                                                input.nextLine();
                                            }

                                            System.out.println(
                                                    "Please
confirm that the purchase price entered is :" + changedPPrice);
                                            System.out.println("Press any
key other than '0' to continue...");
                                            System.out.println("Press    0
to re-enter the purchase price...");
                                            confirm1 = input.next();
                                            if (confirm1.equals("0"))
                                                input.nextLine();
                                        }                          while
(confirm1.contentEquals("0"));


artworks.get(numArtwork).setPurchasePrice(changedPPrice);


                                        break;
                                    case 4:
                                        String changedPurchaseDate = null;
                                        int day, month, year;
                                        boolean  loop1  =  false;  //determine
whether the input (date of purchased) valid or not
                                        String confirm2;
                                        do {
                                            input.nextLine();//avoid bug
                                            try {
                                                do {
```

```
                                                                  loop1 = false;
System.out.print("Enter Purchasing Year: ");
                                                                  year           =
Integer.parseInt(input.nextLine());

System.out.print("Enter Purchasing Month: ");
                                                                  month          =
Integer.parseInt(input.nextLine());

System.out.print("Enter Purchasing Day: ");
                                                                  day            =
Integer.parseInt(input.nextLine());
                                                                  if  (year  <  0  ||
month < 0 || day < 0) {
confirm2 = "0";
System.out.println("\nInvalid input. Please try again...\n");
                                                                  }
                                                                  else {
changedPurchaseDate = day + "/" + month + "/" + year;
System.out.println("Please confirm that the contact number entered is :"
     + changedPurchaseDate);
System.out.println("Press any key other than '0' to continue...");
System.out.println("Press 0 to re-enter the contact...");
confirm2 = input.next();
                                                                       if
(confirm2.equals("0"))
input.nextLine();
                                                                  }
                                                             }              while
(confirm2.contentEquals("0"));
artworks.get(numArtwork).setDatePurchased(changedPurchaseDate);
```

```
                                                                                                    }                              catch
(NumberFormatException ex) {

System.out.println("Invalid input! Press enter to try again...\n");
                                                                                                              loop1 = true;

                                                                                                    }
                                                                                          } while (loop1);


                                                                                          break;

                                                                                case 5:
                                                                                          Artist artist = inputArtist();

artworks.get(numArtwork).setArtist(artist);

                                                                                          break;
                                                                                default:
                                                                                          System.out.println("Invalid      input!
Please try again...\n");
                                                                                          loop = true;
                                                                        }
                                                              } catch (InputMismatchException ex) {
                                                                        System.out.println("Invalid    input!    Please    try
again...\n");
                                                                        loop = true;
                                                                        input.nextLine();//avoid bug
                                                              }
                                                    } while (loop);

                                                    boolean yesNoExit = false;
                                                    System.out.println("Modified successfully !");
                                                    while (!yesNoExit) {

                                                              System.out.println("Do  you  wish  to  perform  any  action?
(y/n)"); //ask the user whether to continue or back to the HOME PAGE
                                                              answer = input.next();
                                                              if (answer.equalsIgnoreCase("n")) {
                                                                        yesNoExit = true;
                                                                        exit = true;
                                                                        System.out.println("\nYou     are     back     to     the
HOME\n");
                                                              } else if (answer.equalsIgnoreCase("y")) {
                                                                        yesNoExit = true;
                                                              } else {
```

```
                                                System.out.println("Invalid input! Please try
again...");
                                        }
                                }
                        }
  }


   public Artist inputArtist()
   {
        Scanner input = new Scanner(System.in);
        int count;
        String name;
        boolean loop;
        do {
        count = 0;
        loop = false;

        System.out.print("Enter Artist Name: ");
        while (!input.hasNext("[A-Za-z]+"))  // to check whether the name is alphabetical letter
or not
        {
             System.out.println("Invalid input! Please try again...\n");
             System.out.print("Enter Artist Name: ");
             input.next();
        }
        name = input.nextLine();


        for (int i = 0; i < artists.size(); i++) {
           if (name.equalsIgnoreCase(artists.get(i).getName())) {
             break;
           } else {
             count++;
           }
        }

        if (count == artists.size()) {
           System.out.println("Artist not found! Please try again...\n");
           System.out.println("Press enter to continue...\n");
           loop = true;
           input.nextLine(); //avoid bug
        }

        }while(loop);
        return artists.get(count);
  }
}
```

Customer Class

```
import java.util.ArrayList;

public class Customer extends Person{

  private String contactNum, artistPreference;
  private ArrayList <String> artworkPurchases;

  public String getContactNum()
  {
    return contactNum;
  }

  public void setContactNum(String contactNum)
  {
    this.contactNum = contactNum;
  }

  public String getArtistPreference()
  {
    return artistPreference;
  }

  public void setArtistPreference(String artistPreference)
  {
    this.artistPreference = artistPreference;
  }

  public ArrayList <String> getArtworkPurchase()
  {
    return artworkPurchases;
  }

  public void setArtworkPurchase(ArrayList<String> artworkPurchases)
  {
    this.artworkPurchases = artworkPurchases;
  }

      public Customer(String name, int age, String contactNum, String artistPreference,
ArrayList<String> artworkPurchases)
  {
    super(name, age);
    this.contactNum = contactNum;
```

```
     this.artistPreference = artistPreference;
    this.artworkPurchases = artworkPurchases;
  }

}
```

Customer Menu Class

```
import java.util.*;
import java.io.File;
import java.io.PrintWriter;
import java.nio.charset.StandardCharsets;

public class CustomerMenu extends AdminMenu {

    private ArrayList<Customer> customers = new ArrayList<Customer>();
    private ArrayList<Artwork> artworks = new ArrayList<Artwork>();

    public void readCustomer() {
        try {
            Scanner infile = new Scanner(new File("Customer.txt"));

            while (infile.hasNextLine()) {
                String name = infile.nextLine();
                int age = Integer.parseInt(infile.nextLine());
                String contactNum = infile.nextLine();
                String artistPreference = infile.nextLine();
                String[] artworkPurchases = infile.nextLine().split(",");
                ArrayList <String> artworkPurchasesArray = new ArrayList
<String> ();
                for (int i =0; i < artworkPurchases.length; i++)
                    artworkPurchasesArray.add(artworkPurchases[i]);
                customers.add(new Customer(name, age, contactNum,
artistPreference, artworkPurchasesArray));
            }
            infile.close();
        } catch (Exception ex) {
            System.out.println(ex);
        }
    }

    public void readArtwork() {
        try {

            Scanner infile = new Scanner(new File("Artwork.txt"));
            while (infile.hasNextLine()) {
                String title = infile.nextLine();
```

```
                        String datePurchased = infile.nextLine();
                        String dateSold = infile.nextLine();
                        double purchasePrice =Double.parseDouble(infile.nextLine());

                        String name = infile.nextLine();
                        int age = Integer.parseInt(infile.nextLine());
                        String specialty = infile.nextLine();
                        String status = infile.nextLine();
                        String priceRange = infile.nextLine();
                        Artist artist = new Artist(name, age, specialty, status, priceRange);

                        boolean isSold = Boolean.parseBoolean(infile.nextLine());
                        double sellingPrice = Double.parseDouble(infile.nextLine());

                        artworks.add(new Artwork(title, datePurchased,purchasePrice,
dateSold, artist, isSold,sellingPrice));
    }
    infile.close();
  } catch (Exception ex) {
    System.out.println(ex);
  }}


      public void updateCustomer() {
            try {
                  PrintWriter outfile = new PrintWriter("Customer.txt");

                  for (Customer customer : customers) {
                        outfile.println(customer.getName().toUpperCase() + "\n" +
customer.getAge() + "\n"
                                    + customer.getContactNum() + "\n" +
customer.getArtistPreference().toUpperCase() + "\n"
                                    + String.join(",",
customer.getArtworkPurchase()).toUpperCase());
                  }
                  outfile.close();

            } catch (Exception ex) {
                  System.out.println(ex);
            }
      }

      @Override
      public void viewInfo() {
            System.out.println("\t\tCustomer List");
            System.out.println("____________________________________________\n");
```

```
        for (int i = 0; i < customers.size(); i++) {
            System.out.println("\nCustomer " + (i + 1));
            System.out.println("_____________________");
            System.out.println("Name: " + customers.get(i).getName());
            System.out.println("Age: " + customers.get(i).getAge());
            System.out.println("Contact Number: " +
customers.get(i).getContactNum());
            System.out.println("Artist Preference: " +
customers.get(i).getArtistPreference());
            System.out.print("Artwork Purchases: ");
            for (int j = 0; j < customers.get(i).getArtworkPurchase().size(); j++) {
                if (j != customers.get(i).getArtworkPurchase().size() - 1)

System.out.print(customers.get(i).getArtworkPurchase().get(j) + ", ");
                else

System.out.println(customers.get(i).getArtworkPurchase().get(j));

            }
            System.out.println();
        }
    }

    @Override
    public void addInfo() {
        System.out.println("\t\tCustomer Registration");
        System.out.println("______________________________________________");
        Scanner input = new Scanner(System.in);
        boolean exit = false;
        boolean exist = false;

        System.out.println("Enter '0' to back to the HOME PAGE");
        System.out.print("Enter Customer Name: ");
        String name = input.nextLine();

         if(name.equals("0"))
         {
             exit = true;
             System.out.println("\nYou are back to the HOME PAGE\n");
         }


        for (int i = 0; i < customers.size(); i++) {
            if (name.equalsIgnoreCase(customers.get(i).getName())) {
                exist = true;
            }
        }

        if (!exist && !exit) {
            int age = 0;
            do {
                try {
                    System.out.print("Enter Customer age: ");
                    age = Integer.parseInt(input.nextLine());
                } catch (Exception ex) {
                    System.out.println("Customer age must larger than 0 and
must be an integer.");
                }
            } while (age <= 0);

            System.out.print("Enter Customer Contact Number: ");
            String contactNum = input.nextLine();

            System.out.print("Enter Customer Preferred Artist(Enter <NO> if no
prefered artist): ");
            String artistPreference = input.nextLine();
            if (artistPreference.equalsIgnoreCase("NO"))
                artistPreference = "NO PREFERRED ARTIST";

            int num = 0;
            do {
                System.out.print("Enter number of artwork purchases: ");
                try{
                    num = Integer.parseInt(input.nextLine());
                }
                catch(NumberFormatException ex)
                {
                    System.out.println("Invalid input! Please try again...\n");
                    num = -1;
                }
                if (num <= 0)
                    System.out.print("Number of artwork purchases should not
be less than 1.\nPlease re-enter the number.\n");
            } while (num <= 0);

            String[] artworkPurchases = new String[num];

            boolean loop;
            for (int j = 0; j < num; j++)
            {
                do{
                    int count = 0;
                    loop = false;
                    System.out.printf("Enter Artwork Purchase %d: ", j + 1);
                    artworkPurchases[j] = input.nextLine();

                for(int i =0;i < artworks.size();i ++)// to check whether the artwork
exist or not and whether being sold
                {
if(artworkPurchases[j].equalsIgnoreCase(artworks.get(i).getTitle()))
                    {
                        if(artworks.get(i).getIsSold())
                        {
                            System.out.println("This artwork is sold!
Please try again...\n");
                            loop = true;
                            break;
                        }
                        else
                        {

                            break;
                        }


                    }
                    else
                    {
                        count ++;
                    }
                }



                }while(loop);
            }
            ArrayList <String> tempArray = new ArrayList <String> ();
            for (int a =0; a< artworkPurchases.length;a++)
                tempArray.add(artworkPurchases[a]);

            customers.add(new Customer(name, age, contactNum, artistPreference,
tempArray));
            System.out.println("\nCustomer registered successfully.\n");

        }
        else if (exist)
            System.out.println("\nCustomer already registered.\n");
```

```
    }

    @Override
    public void removeInfo() {
        System.out.println("\t\tCustomer Deletion");
        System.out.println("____________________________________________________________");
        Scanner input = new Scanner(System.in);

        System.out.println("Enter '0' to back to the HOME PAGE");
        System.out.print("Enter customer name to be remove: ");
        String name = input.nextLine();
        if(name.equals("0"))
    System.out.println("\nYou have back to the HOME PAGE\n");
        else {
        int oriSize = customers.size();
        for (int i = 0; i < customers.size(); i++) {
            if (name.equalsIgnoreCase(customers.get(i).getName())) {
                customers.remove(i);
                System.out.println("\nCustomer removed from record
successfully.\n");
            }
        }
        if(oriSize == customers.size())
   {
    System.out.println("\nCustomer not Found! Please try again...\n");
   }
        }
    }


    @Override
    public void modifyInfo() {
        boolean loop = false; //loop after catch InputMismatchException
        boolean loop1 = false; //loop after catch NumberFormatException
        int numCustomer = customers.size();
        boolean exit =false; //exit turn true when user wants to back to HOME PAGE
        Scanner input = new Scanner(System.in);
            do {

        System.out.println("\t\tCustomer Information Update");
        System.out.println("____________________________________________________________");



        System.out.println("Enter '0' to back to the HOME PAGE");
        System.out.print("Enter customer name to be modified: ");
```

```
String name = input.nextLine();

if(name.equals("0"))
{
        exit = true;
        System.out.println("\nYou have back to the HOME PAGE\n");
        break;

}

for (int i = 0; i < customers.size(); i++) // find the customer info that the user wants to modify
{
        if (name.equalsIgnoreCase(customers.get(i).getName()))
        {
                numCustomer = i;
        }
}
if(numCustomer == customers.size()) //if the customer not found
{
        System.out.println("Customer Not Found !\n");
}
}while(numCustomer == customers.size());

String answer;
while(!exit){
        do {
System.out.println("\nWhat information do you want to modify?");
System.out.println("1. Name");
System.out.println("2. Age");
System.out.println("3. Contact");
System.out.println("4. Artwork Purchases");
System.out.println("5. Artist Preference");
try {
int choice = input.nextInt();

switch (choice)
{
case 1 :
        String confirm;
        String changedName;
        input.nextLine();//avoid bug
        do {
                boolean exist = false;
                System.out.println("Please enter the name:");
                changedName = input.nextLine();
                for (int i = 0; i < customers.size(); i++) {
                    if
(changedName.equalsIgnoreCase(customers.get(i).getName())) {
                        exist = true;
                    }
                }
                if(!exist)
                {
                System.out.println("Please confirm that the name entered is :" +
changedName.toUpperCase()); //let the user to re-confirm his/her input to reduce the chance of
making mistake
                System.out.println("Press any key other than '0' to continue...");
                System.out.println("Press 0 to re-enter the name...");
                confirm = input.next();
                if(confirm.equals("0"))
                    input.nextLine();
                }
                else
                {
                    input.nextLine();
                    System.out.println("Customer is already exist! Please try
again...\n");
                    confirm = "0";
                }
            }while(confirm.equals("0"));
            customers.get(numCustomer).setName(changedName);
            break;
        case 2:
            int age;
            String confirm3;
            boolean loop2 = false;
            input.nextLine();//avoid bug
            do {
            try {
            do {
            loop2 = false;
            System.out.println("Please enter the new age:");
            age = input.nextInt();
            System.out.println("Please confirm that the age entered is :" + age); //let
the user to re-confirm his/her input to reduce the chance of making mistake
            System.out.println("Press any key other than '0' to continue...");
            System.out.println("Press 0 to re-enter the age...");
            confirm3 = input.next();
            if(confirm3.equals("0"))
                input.nextLine();
            else if (age < 0)
```

```
    {
        loop2 = true;
        System.out.println("Age should be greater than 0.\n");
    }
    }while(confirm3.contentEquals("0"));
    customers.get(numCustomer).setAge(age);

    }catch(InputMismatchException ex)
    {
        System.out.println("Invalid input! Please try again...\n");
        loop2 = true;
        input.nextLine(); //avoid bug

    }
    }while(loop2);

    break;
case 3:
    String changedContact;
    String confirm1;
    input.nextLine();//avoid bug
    do {
    System.out.println("Please enter the new contact:");
    changedContact = input.nextLine();
    System.out.println("Please confirm that the contact number entered is :" +
changedContact); //let the user to re-confirm his/her input to reduce the chance of making
mistake
    System.out.println("Press any key other than '0' to continue...");
    System.out.println("Press 0 to re-enter the contact...");
    confirm1 = input.next();
    if(confirm1.equals("0"))
        input.nextLine();
    }while(confirm1.contentEquals("0"));
    customers.get(numCustomer).setContactNum(changedContact);
    break;
case 4:
    input.nextLine();//avoid bug
    String artworkTitle;
    boolean exist;

    ArrayList <String> tempArray = new ArrayList <String>();
    tempArray = customers.get(numCustomer).getArtworkPurchase();

    do {
        int count =0;
        exist = true;
```

```
            System.out.println("Please enter the title of new artwork purchase:");
            artworkTitle = input.nextLine();
            for(int i =0;i < artworks.size();i ++) // to check whether the artwork exist or not and whether being sold
            {
                if(artworkTitle.equalsIgnoreCase(artworks.get(i).getTitle()))
                {
                    if(artworks.get(i).getIsSold())
                    {
                        System.out.println("This artwork is sold! Please try again...\n");
                        exist = false;
                        break;
                    }
                    else
                    {

                        break;
                    }


                }
                else
                {
                    count ++;
                }
            }
            }while(!exist);
            tempArray.add(artworkTitle);
            customers.get(numCustomer).setArtworkPurchase(tempArray);
            break;
        case 5:
            String changedArtist;
            String confirm2;
            do {
            input.nextLine();//avoid bug
            System.out.println("Please enter the new artist preference:");
            changedArtist = input.nextLine();
            System.out.println("Please confirm that the new artist preference entered is :" + changedArtist);//let the user to re-confirm his/her input to reduce the chance of making mistake
            System.out.println("Press any key other than '0' to continue...");
            System.out.println("Press 0 to re-enter the artist preference...");
            confirm2 = input.next();
            }while(confirm2.contentEquals("0"));
            customers.get(numCustomer).setArtistPreference(changedArtist);
            break;
        default :
            System.out.println("Invalid input! Please try again...");
            loop = true;

    }
        }catch(InputMismatchException ex)
        {
            System.out.println("Invalid input! Please try again...\n");
            loop = true;
            input.nextLine();//avoid bug
        }

    } while (loop);

    boolean yesNoExit = false;
    System.out.println("Modified successfully !");
    while (!yesNoExit) {

        System.out.println("Do you wish to perform any action? (y/n)"); //ask the user whether to continue or back to the HOME PAGE
        answer = input.next();
        if(answer.equalsIgnoreCase("n"))
        {
            yesNoExit = true;
            exit = true;
            System.out.println("\nYou are back to the HOME\n");
        }
        else if(answer.equalsIgnoreCase("y"))
        {
            yesNoExit = true;
        }
        else {
            System.out.println("Invalid input! Please try again...");
        }
    }


    }
}
}
```

Person Class

```
public abstract class Person {
    protected String name;
    protected int age;

    public String getName()
    {
        return name;
    }

    public void setName(String name)
    {
        this.name = name;
    }

    public int getAge()
    {
        return age;
    }

    public void setAge(int age)
    {
        this.age = age;
    }
    public Person(String name, int age)
    {
        this.name = name;
        this.age = age;
    }
}
```

<u>Sample Input</u>

Owner File

```
Jayden Wong
jaydenwongabcd1234
```

Customer File

```
EIVONNE GOH
20
011-12345678
LEONARDO DA VINCI
MONA LISA,ST. PETER'S BASILICA IN ROME
ISABELLE TAN
21
011-10291829
```

```
MICHELANGELO
THE LAST SUPPER
SHAREN LEE
25
012-9128723
NO PREFERRED ARTIST
THE LADY WITH AN ERMINE
LAI JYE HONG
46
012-7878978
RAPHAEL
THE BIRTH OF VENUS,GUERNICA
ALAN FONG
58
016-5656456
PABLO PICASSO
THE CREATION OF ADAM
SUSAN TIANG
35
011-2345789
JOHANNES VERMEER
GIRL WITH A PEARL EARRING
JASMINE FOH
41
016-7272455
EDWARD HOPPER
SPRING IN TOWN,YOUNG CORN
JOHNATHAN LIM
55
013-9050120
VINCENT VAN GOGH
STARRY NIGHT,SUNFLOWERS,WATER LILIES
CHELSEA WONG
25
018-9090450
NO PREFERRED ARTIST
AMERICAN GOTHIC
PATRICK KOH
34
018-6505615
EDVARD MUNCH
THE SCREAM,THE KISS
GERMAINE LEE
27
016-7667550
NO PREFERRED ARTIST
```

```
WATER LILY POND
QUAN WEI
32
017-6324898
NO PREFERRED ARTIST
THE PERSISTENCE OF MEMORY
MOHAMMED JABBAR
46
018-5750425
RAPHAEL
THE SCHOOL OF ATHENS
ALEXANDER ANBUSELVAN
33
017-5450120
NO PREFERRED ARTIST
NAPOLEON CROSSING THE ALPS
GANESH SUMANTHAN
29
018-5670987
JACQUES LOUIS DAVID
SALVATOR MUNDI
SHIKHA THAKUR
45
019-3456234
LEONARDO DA VINCI
THE VIRGIN OF THE ROCKS,HEAD OF A WOMAN
SALLEH SULAIMAN
38
011-3245878
NO PREFERRED ARTIST
GINEVRA DE' BENCI
SITI NORHALIZAH
47
017-5656321
MICHELANGELO
THE DANCE,LAS MENINAS
JONAS YONG
51
018-6780991
MARK ROTHKO
OLYMPIA,KANAGAWA
RACHEL POON
26
019-4567990
NO PREFERRED ARTIST
THE NIGHT WATCH
```

Artist File

```
LEONARDO DA VINCI
67
DRAMATIC
DECEASED
66000.0-100000.0
MICHELANGELO
89
HUMAN
DECEASED
60000.0-100000.0
RAPHAEL
37
SOLEMN
DECEASED
50000.0-100000.0
PABLO PICASSO
91
COLORFUL
DECEASED
65000.0-100000.0
JOHANNES VERMEER
43
VIBRANT
DECEASED
40000.0-80000.0
EDWARD HOPPER
84
BOLD
DECEASED
50000.0-90000.0
VINCENT VAN GOGH
37
TWISTED
DECEASED
60000.0-100000.0
EDVARD MUNCH
80
HYPNOTIC
DECEASED
57000.0-95000.0
JACQUES LOUIS DAVID
77
NEOCLASSIC
```

```
DECEASED
50000.0-70000.0
MARK ROTHKO
66
SIMPLE
DECEASED
40000.0-70000.0
SANDRO BOTTICELLI
65
RENAISSANCE
DECEASED
70000.0-100000.0
GRANT WOOD
50
NATURE
DECEASED
30000.0-60000.0
CLAUDE MONET
86
FAUNA
DECEASED
40000.0-70000.0
GUSTAV KLIMT
55
TEXTURE
DECEASED
56000.0-96000.0
SALVADOR DALI
84
CROOKED
DECEASED
60000.0-97000.0
HENRI MATISSE
84
MODERNISM
DECEASED
50000.0-65000.0
DIEGO VELAZQUEZ
61
CONTRAST
DECEASED
55000.0-70000.0
EDOUARD MANET
51
LIVELY
DECEASED
```

```
45000.0-60000.0
REMBRANDT VAN RIJN
63
DEPRESSED
DECEASED
55000.0-70000.0
HOKUSAI
88
NATURE
DECEASED
60000.0-80000.0
```

Artwork File

```
MONA LISA
12/1/2020
16/1/2022
80000.0
LEONARDO DA VINCI
67
DRAMATIC
DECEASED
66000.0-100000.0
true
80000.0
ST. PETER'S BASILICA IN ROME
13/2/2019
11/3/2022
95000.0
MICHELANGELO
89
HUMAN
DECEASED
60000.0-100000.0
true
95000.0
THE LAST SUPPER
13/2/2020
3/1/2023
99000.0
LEONARDO DA VINCI
67
DRAMATIC
DECEASED
66000.0-100000.0
true
```

```
99000.0
THE LADY WITH AN ERMINE
6/6/2020
28/12/2022
88000.0
LEONARDO DA VINCI
67
DRAMATIC
DECEASED
66000.0-100000.0
true
88000.0
THE BIRTH OF VENUS
12/11/2019
22/1/2023
98000.0
SANDRO BOTTICELLI
65
RENAISSANCE
DECEASED
70000.0-100000.0
true
98000.0
GUERNICA
25/7/2021
31/12/2022
97000.0
PABLO PICASSO
91
COLORFUL
DECEASED
65000.0-100000.0
true
97000.0
THE CREATION OF ADAM
5/5/2020
19/1/2023
96000.0
MICHELANGELO
89
HUMAN
DECEASED
60000.0-100000.0
true
96000.0
GIRL WITH A PEARL EARRING
```

```
21/8/2020
19/9/2022
66000.0
JOHANNES VERMEER
43
VIBRANT
DECEASED
40000.0-80000.0
true
66000.0
SPRING IN TOWN
7/7/2020
20/2/2023
50000.0
GRANT WOOD
50
NATURE
DECEASED
30000.0-60000.0
true
50000.0
YOUNG CORN
17/9/2020
1/3/2023
55000.0
GRANT WOOD
50
NATURE
DECEASED
30000.0-60000.0
true
55000.0
STARRY NIGHT
12/12/2021
1/1/2023
67000.0
VINCENT VAN GOGH
37
TWISTED
DECEASED
60000.0-100000.0
true
67000.0
SUNFLOWERS
25/10/2019
31/12/2021
```

```
88000.0
VINCENT VAN GOGH
37
TWISTED
DECEASED
60000.0-100000.0
true
88000.0
WATER LILIES
12/8/2020
5/2/2022
60000.0
CLAUDE MONET
86
FAUNA
DECEASED
40000.0-70000.0
true
60000.0
AMERICAN GOTHIC
4/4/2021
14/7/2022
60000.0
GRANT WOOD
50
NATURE
DECEASED
30000.0-60000.0
true
60000.0
THE SCREAM
6/6/2021
2/2/2023
90000.0
EDVARD MUNCH
80
HYPNOTIC
DECEASED
57000.0-95000.0
true
90000.0
THE KISS
11/11/2019
25/5/2022
96000.0
GUSTAV KLIMT
```

```
55
TEXTURE
DECEASED
56000.0-96000.0
true
96000.0
WATER LILY POND
1/2/2020
5/2/2023
69000.0
CLAUDE MONET
86
FAUNA
DECEASED
40000.0-70000.0
true
69000.0
THE PERSISTENCE OF MEMORY
27/3/2021
28/1/2023
97000.0
SALVADOR DALI
84
CROOKED
DECEASED
60000.0-97000.0
true
97000.0
THE SCHOOL OF ATHENS
30/12/2020
1/1/2023
99000.0
RAPHAEL
37
SOLEMN
DECEASED
50000.0-100000.0
true
99000.0
NAPOLEON CROSSING THE ALPS
2/1/2021
2/1/2023
70000.0
JACQUES LOUIS DAVID
77
NEOCLASSIC
```

```
DECEASED
50000.0-70000.0
true
70000.0
LAS MENINAS
18/2/2021
4/3/2023
70000.0
DIEGO VELAZQUEZ
61
CONTRAST
DECEASED
55000.0-70000.0
true
70000.0
OLYMPIA
30/10/2019
1/2/2023
60000.0
EDOUARD MANET
51
LIVELY
DECEASED
45000.0-60000.0
true
60000.0
KANAGAWA
13/8/2020
18/9/2022
80000.0
HOKUSAI
88
NATURE
DECEASED
60000.0-80000.0
true
80000.0
THE NIGHT WATCH
4/4/2021
10/10/2022
55000.0
REMBRANDT VAN RIJN
63
DEPRESSED
DECEASED
55000.0-70000.0
```

```
true
55000.0
UNTITLED
3/3/2019
INVALID
40000.0
MARK ROTHKO
66
SIMPLE
DECEASED
40000.0-70000.0
false
40000.0
ENTRANCE TO SUBWAY
12/9/2019
INVALID
50000.0
MARK ROTHKO
66
SIMPLE
DECEASED
40000.0-70000.0
false
50000.0
ORANGE AND YELLOW
5/5/2021
INVALID
55000.0
MARK ROTHKO
66
SIMPLE
DECEASED
40000.0-70000.0
false
55000.0
VITRUVIAN MAN
21/4/2020
INVALID
70000.0
LEONARDO DA VINCI
67
DRAMATIC
DECEASED
66000.0-100000.0
false
70000.0
```

```
DONI TONDO
23/3/2020
INVALID
70000.0
MICHELANGELO
89
HUMAN
DECEASED
60000.0-100000.0
false
70000.0
THE CONVERSION OF SAUL
16/5/2020
INVALID
72000.0
MICHELANGELO
89
HUMAN
DECEASED
60000.0-100000.0
false
72000.0
THE SISTINE MADONNA
21/10/2021
INVALID
68000.0
RAPHAEL
37
SOLEMN
DECEASED
50000.0-100000.0
false
68000.0
LA FORNARINA
11/11/2019
INVALID
55000.0
RAPHAEL
37
SOLEMN
DECEASED
50000.0-100000.0
false
55000.0
CHILD WITH DOVE
22/5/2021
```

```
INVALID
65000.0
PABLO PICASSO
91
COLORFUL
DECEASED
65000.0-100000.0
false
65000.0
THE OLD BLIND GUITARIST
18/4/2020
INVALID
70000.0
PABLO PICASSO
91
COLORFUL
DECEASED
65000.0-100000.0
false
70000.0
THE GLASS OF WINE
30/6/2021
INVALID
45000.0
JOHANNES VERMEER
43
VIBRANT
DECEASED
40000.0-80000.0
false
45000.0
THE MILKMAID
21/7/2020
INVALID
55000.0
JOHANNES VERMEER
43
VIBRANT
DECEASED
40000.0-80000.0
false
55000.0
THE LOVE LETTER
10/10/2019
INVALID
40000.0
```

```
JOHANNES VERMEER
43
VIBRANT
DECEASED
40000.0-80000.0
false
40000.0
NIGHTHAWKS
12/2/2021
INVALID
60000.0
EDWARD HOPPER
77
BOLD
DECEASED
50000.0-90000.0
false
60000.0
CHOP SUEY
20/4/2020
INVALID
60000.0
EDWARD HOPPER
84
BOLD
DECEASED
50000.0-90000.0
false
60000.0
AUTOMAT
25/10/2021
INVALID
61000.0
EDWARD HOPPER
84
BOLD
DECEASED
50000.0-90000.0
false
61000.0
EARLY SUNDAY MORNING
1/11/2020
INVALID
50000.0
EDWARD HOPPER
84
```

```
BOLD
DECEASED
50000.0-90000.0
false
50000.0
IRISES
2/3/2020
INVALID
66000.0
VINCENT VAN GOGH
37
TWISTED
DECEASED
60000.0-100000.0
false
66000.0
PAUL GAUGUIN'S ARMCHAIR
12/12/2019
INVALID
70000.0
VINCENT VAN GOGH
37
TWISTED
DECEASED
60000.0-100000.0
false
70000.0
RED VINEYARDS AT ARLES
12/1/2020
INVALID
60000.0
VINCENT VAN GOGH
37
TWISTED
DECEASED
60000.0-100000.0
false
60000.0
ANXIETY
13/4/2021
INVALID
70000.0
EDVARD MUNCH
80
HYPNOTIC
DECEASED
```

```
57000.0-95000.0
false
70000.0
ASHES
12/7/2020
INVALID
69000.0
EDVARD MUNCH
80
HYPNOTIC
DECEASED
57000.0-95000.0
false
69000.0
THE DEATH OF BARA
21/12/2021
INVALID
50000.0
JACQUES LOUIS DAVID
77
NEOCLASSIC
DECEASED
50000.0-70000.0
false
50000.0
PATROCLUS
15/1/2020
INVALID
55000.0
JACQUES LOUIS DAVID
77
NEOCLASSIC
DECEASED
50000.0-70000.0
false
55000.0
PORTRAIT OF DANTE
7/4/2020
INVALID
75000.0
SANDRO BOTTICELLI
65
RENAISSANCE
DECEASED
70000.0-100000.0
false
```

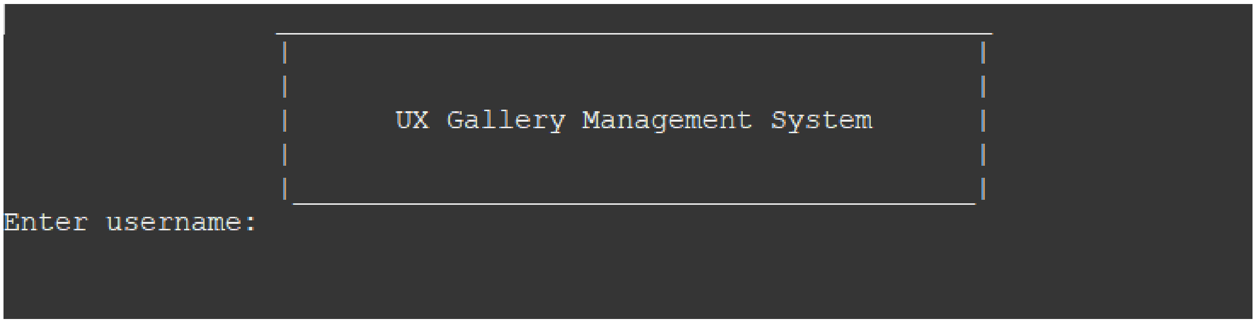
```
75000.0
VENUS AND MARS
12/12/2019
INVALID
77000.0
SANDRO BOTTICELLI
65
RENAISSANCE
DECEASED
70000.0-100000.0
false
77000.0
GOLDFISH
12/5/2021
INVALID
60000.0
GUSTAV KLIMT
55
TEXTURE
DECEASED
56000.0-96000.0
false
60000.0
DANAE
15/1/2020
INVALID
66000.0
GUSTAV KLIMT
55
TEXTURE
DECEASED
56000.0-96000.0
false
66000.0
UN CHIEN ANDALOU
14/12/2019
INVALID
60000.0
SALVADOR DALI
84
CROOKED
DECEASED
60000.0-97000.0
false
60000.0
THE JOY OF LIFE
```

```
19/11/2020
INVALID
50000.0
HENRI MATISSE
84
MODERNISM
DECEASED
50000.0-65000.0
false
50000.0
WOMAN WITH A HAT
14/5/2021
INVALID
55000.0
HENRI MATISSE
84
MODERNISM
DECEASED
50000.0-65000.0
false
55000.0
THE ROKEBY VENUS
1/12/2020
INVALID
55000.0
DIEGO VELAZQUEZ
61
CONTRAST
DECEASED
55000.0-70000.0
false
55000.0
THE FIFER
12/3/2019
INVALID
55000.0
EDOUARD MANET
51
LIVELY
DECEASED
45000.0-60000.0
false
55000.0
LOLA DE VALENCE
30/1/2020
INVALID
```

```
60000.0
EDOUARD MANET
51
LIVELY
DECEASED
45000.0-60000.0
false
60000.0
FLORA
1/1/2021
INVALID
55000.0
REMBRANDT VAN RIJN
63
DEPRESSED
DECEASED
55000.0-70000.0
false
55000.0
THE STORM OF THE SEA OF GALILEE
21/2/2020
INVALID
60000.0
REMBRANDT VAN RIJN
63
DEPRESSED
DECEASED
55000.0-70000.0
false
60000.0
```

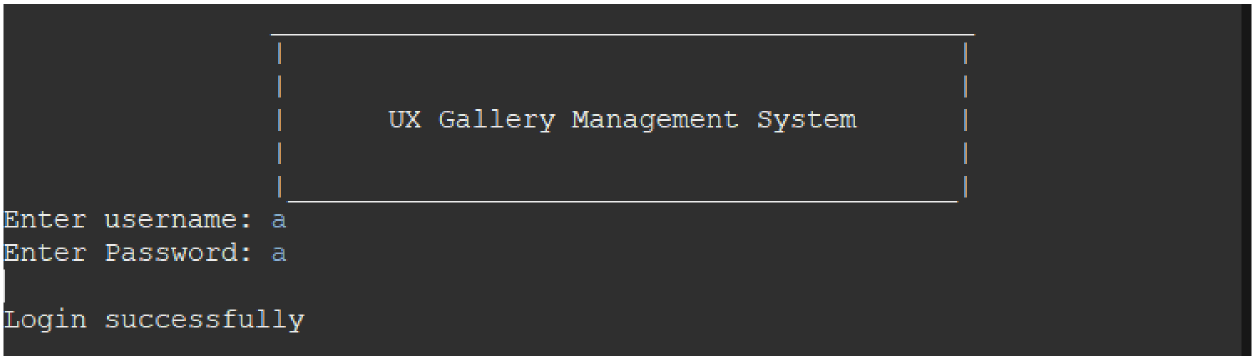
__Sample Output__

START OF PROGRAM



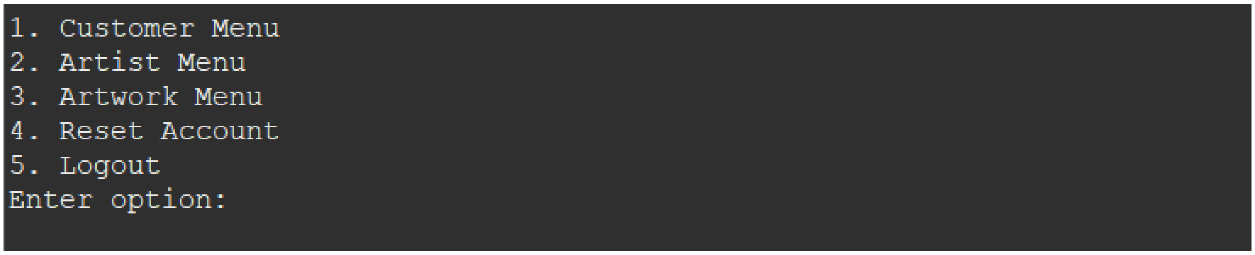
```
                    _________________________________________
                    |                                        |
                    |                                        |
                    |       UX Gallery Management System     |
                    |                                        |
                    |________________________________________|
Enter username:
```

ENTER USERNAME AND PASSWORD



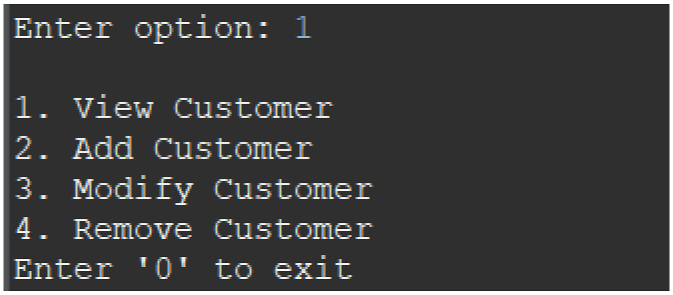
```
                   ________________________________________________
                  |                                              |
                  |                                              |
                  |       UX Gallery Management System           |
                  |                                              |
                  |______________________________________________|
Enter username: a
Enter Password: a

Login successfully
```

MENU



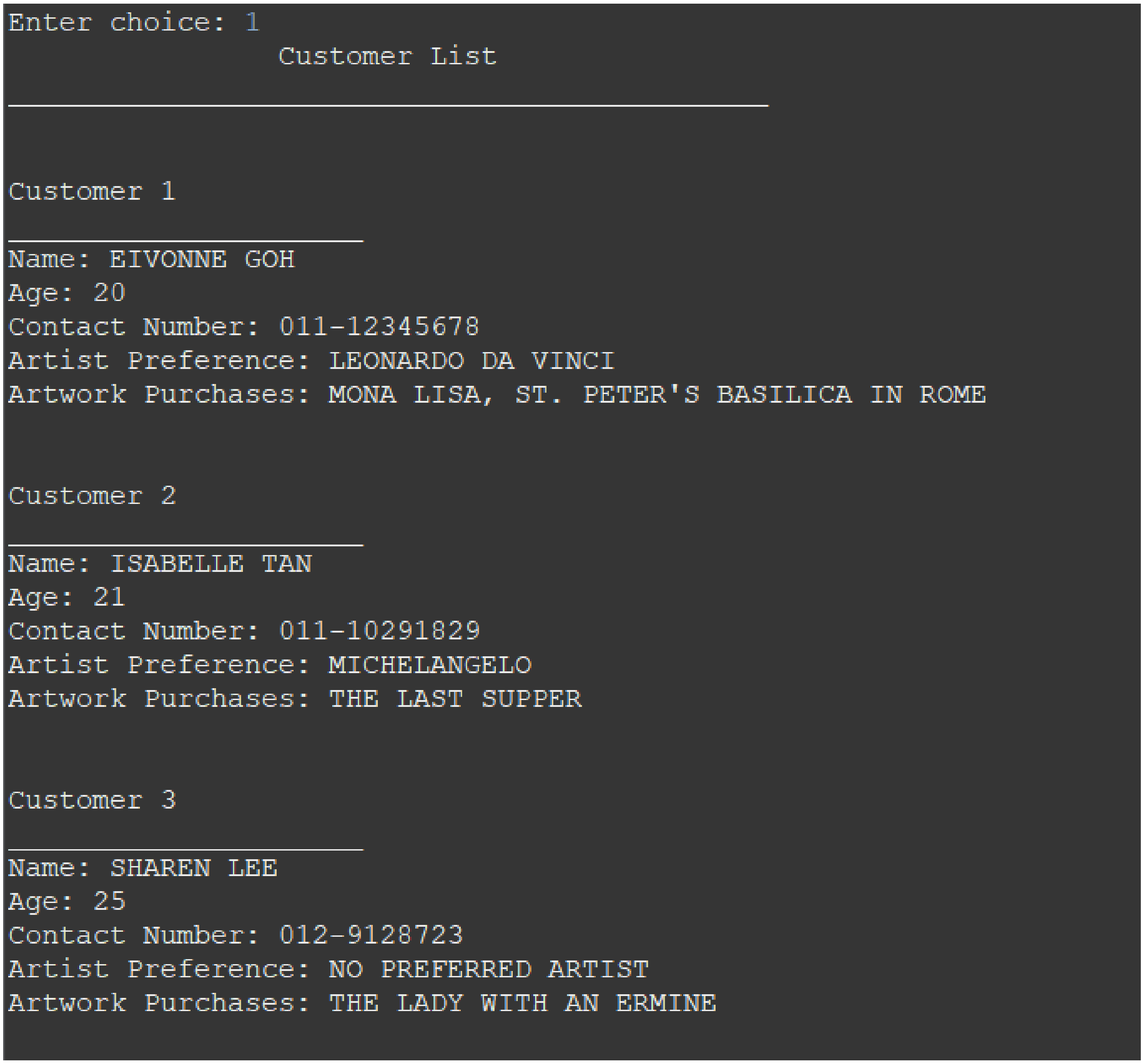
```
1. Customer Menu
2. Artist Menu
3. Artwork Menu
4. Reset Account
5. Logout
Enter option:
```

OPTION 1: CUSTOMER MENU



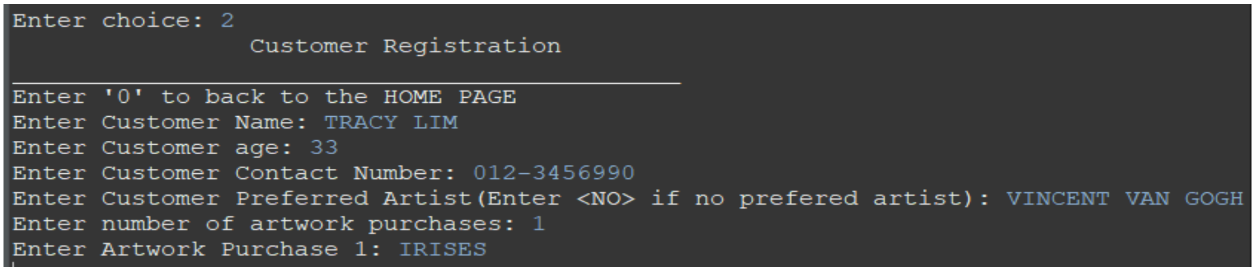
```
Enter option: 1

1. View Customer
2. Add Customer
3. Modify Customer
4. Remove Customer
Enter '0' to exit
```

OPTION 1: CHOICE 1.VIEW CUSTOMER FUNCTION



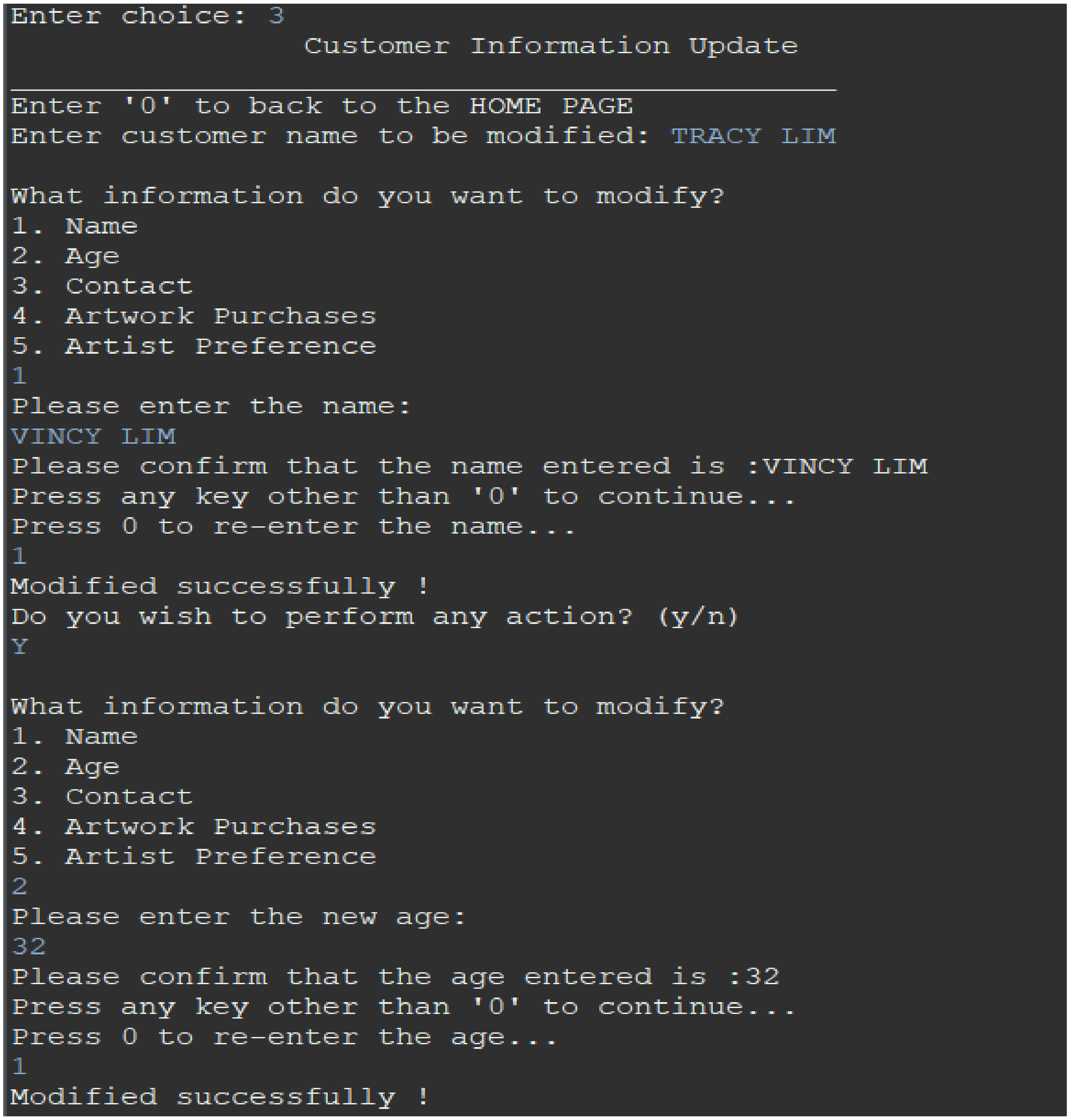
```
Enter choice: 1
                Customer List
_____________________________________________


Customer 1
_____________________
Name: EIVONNE GOH
Age: 20
Contact Number: 011-12345678
Artist Preference: LEONARDO DA VINCI
Artwork Purchases: MONA LISA, ST. PETER'S BASILICA IN ROME


Customer 2
_____________________
Name: ISABELLE TAN
Age: 21
Contact Number: 011-10291829
Artist Preference: MICHELANGELO
Artwork Purchases: THE LAST SUPPER


Customer 3
_____________________
Name: SHAREN LEE
Age: 25
Contact Number: 012-9128723
Artist Preference: NO PREFERRED ARTIST
Artwork Purchases: THE LADY WITH AN ERMINE
```

OPTION 1: CHOICE 2.ADD CUSTOMER FUNCTION

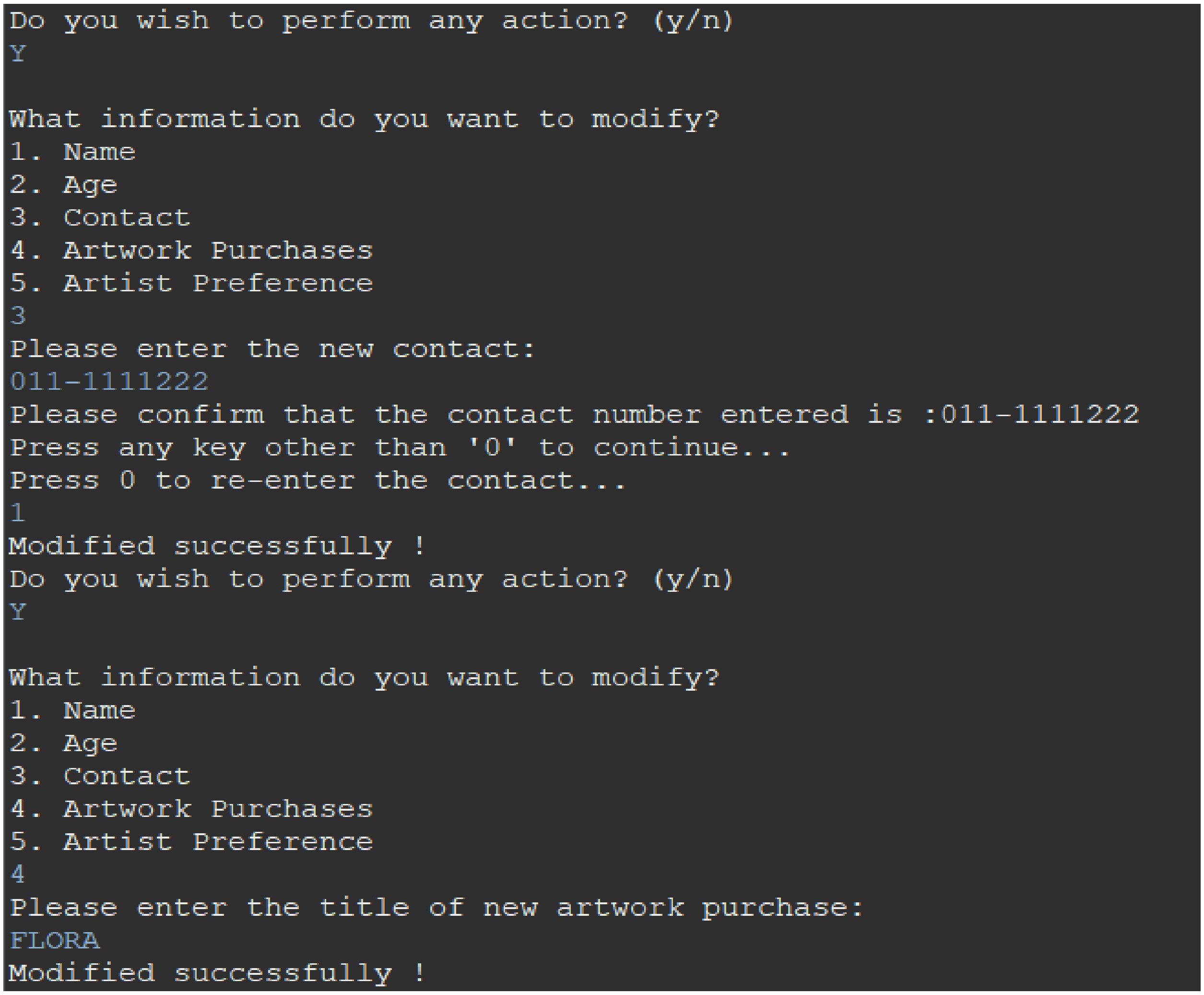


```
Enter choice: 2
                Customer Registration
_____________________________________________
Enter '0' to back to the HOME PAGE
Enter Customer Name: TRACY LIM
Enter Customer age: 33
Enter Customer Contact Number: 012-3456990
Enter Customer Preferred Artist(Enter <NO> if no prefered artist): VINCENT VAN GOGH
Enter number of artwork purchases: 1
Enter Artwork Purchase 1: IRISES
```

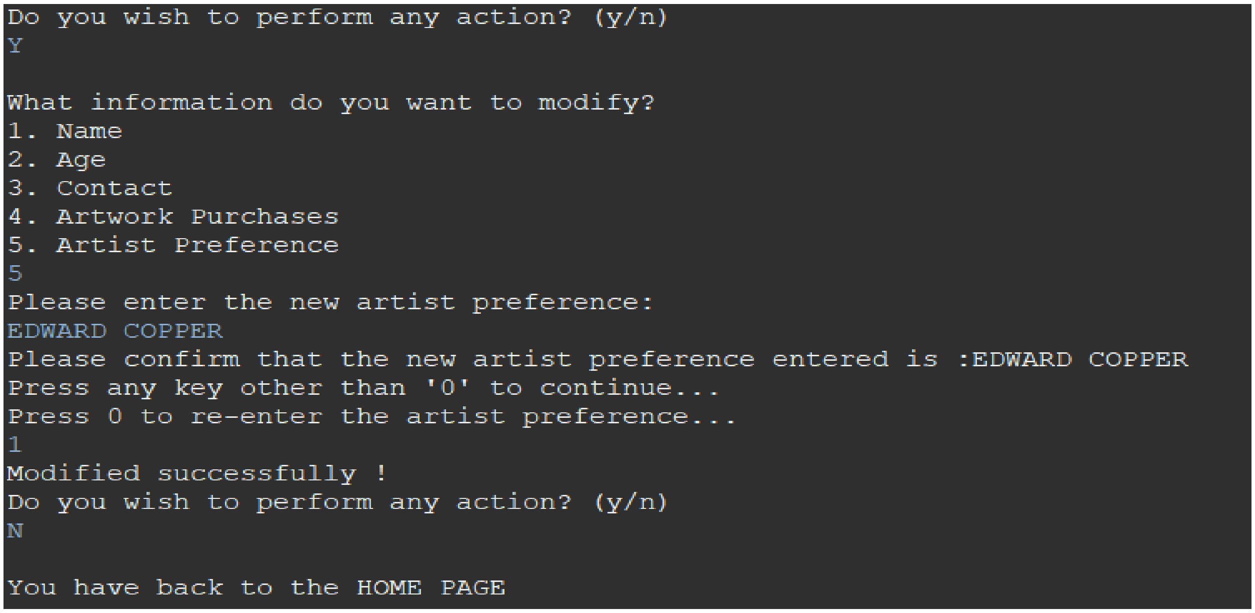
OPTION 1: CHOICE 3. MODIFY CUSTOMER FUNCTION



```
Enter choice: 3
                Customer Information Update
_____________________________________________
Enter '0' to back to the HOME PAGE
Enter customer name to be modified: TRACY LIM

What information do you want to modify?
1. Name
2. Age
3. Contact
4. Artwork Purchases
5. Artist Preference
1
Please enter the name:
VINCY LIM
Please confirm that the name entered is :VINCY LIM
Press any key other than '0' to continue...
Press 0 to re-enter the name...
1
Modified successfully !
Do you wish to perform any action? (y/n)
Y

What information do you want to modify?
1. Name
2. Age
3. Contact
4. Artwork Purchases
5. Artist Preference
2
Please enter the new age:
32
Please confirm that the age entered is :32
Press any key other than '0' to continue...
Press 0 to re-enter the age...
1
Modified successfully !
```

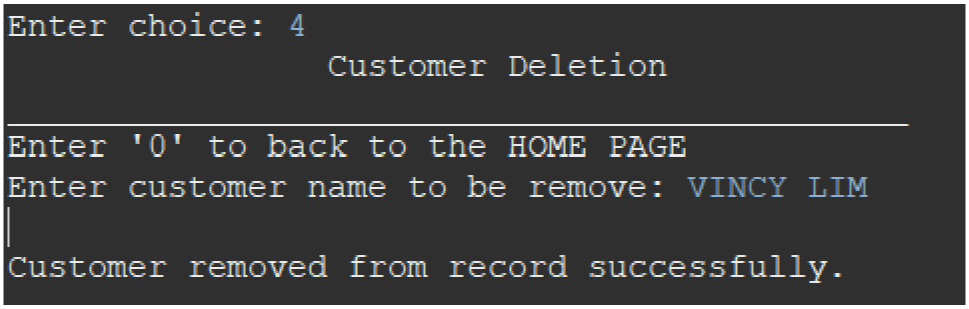


```
Do you wish to perform any action? (y/n)
Y

What information do you want to modify?
1. Name
2. Age
3. Contact
4. Artwork Purchases
5. Artist Preference
3
Please enter the new contact:
011-1111222
Please confirm that the contact number entered is :011-1111222
Press any key other than '0' to continue...
Press 0 to re-enter the contact...
1
Modified successfully !
Do you wish to perform any action? (y/n)
Y

What information do you want to modify?
1. Name
2. Age
3. Contact
4. Artwork Purchases
5. Artist Preference
4
Please enter the title of new artwork purchase:
FLORA
Modified successfully !
```



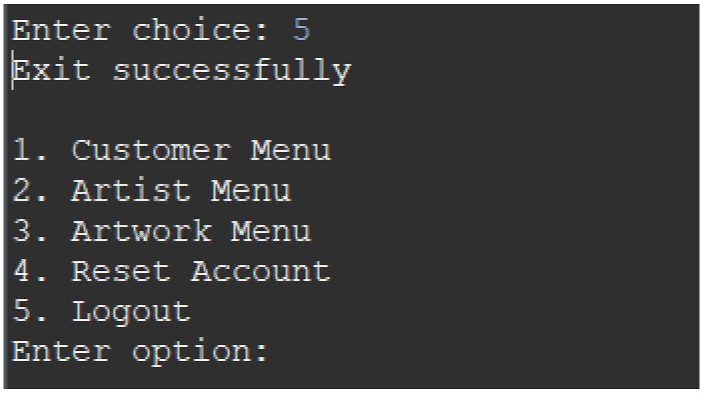
```
Do you wish to perform any action? (y/n)
Y

What information do you want to modify?
1. Name
2. Age
3. Contact
4. Artwork Purchases
5. Artist Preference
5
Please enter the new artist preference:
EDWARD COPPER
Please confirm that the new artist preference entered is :EDWARD COPPER
Press any key other than '0' to continue...
Press 0 to re-enter the artist preference...
1
Modified successfully !
Do you wish to perform any action? (y/n)
N

You have back to the HOME PAGE
```

OPTION 1: CHOICE 4.REMOVE CUSTOMER FUNCTION



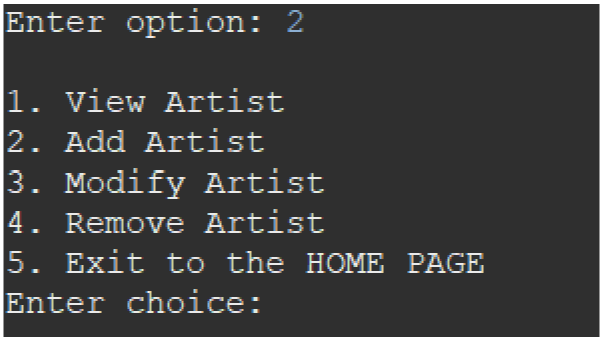
```
Enter choice: 4
                Customer Deletion
_____________________________________________
Enter '0' to back to the HOME PAGE
Enter customer name to be remove: VINCY LIM

Customer removed from record successfully.
```

OPTION 1: CHOICE 5. EXIT TO HOME PAGE FUNCTION



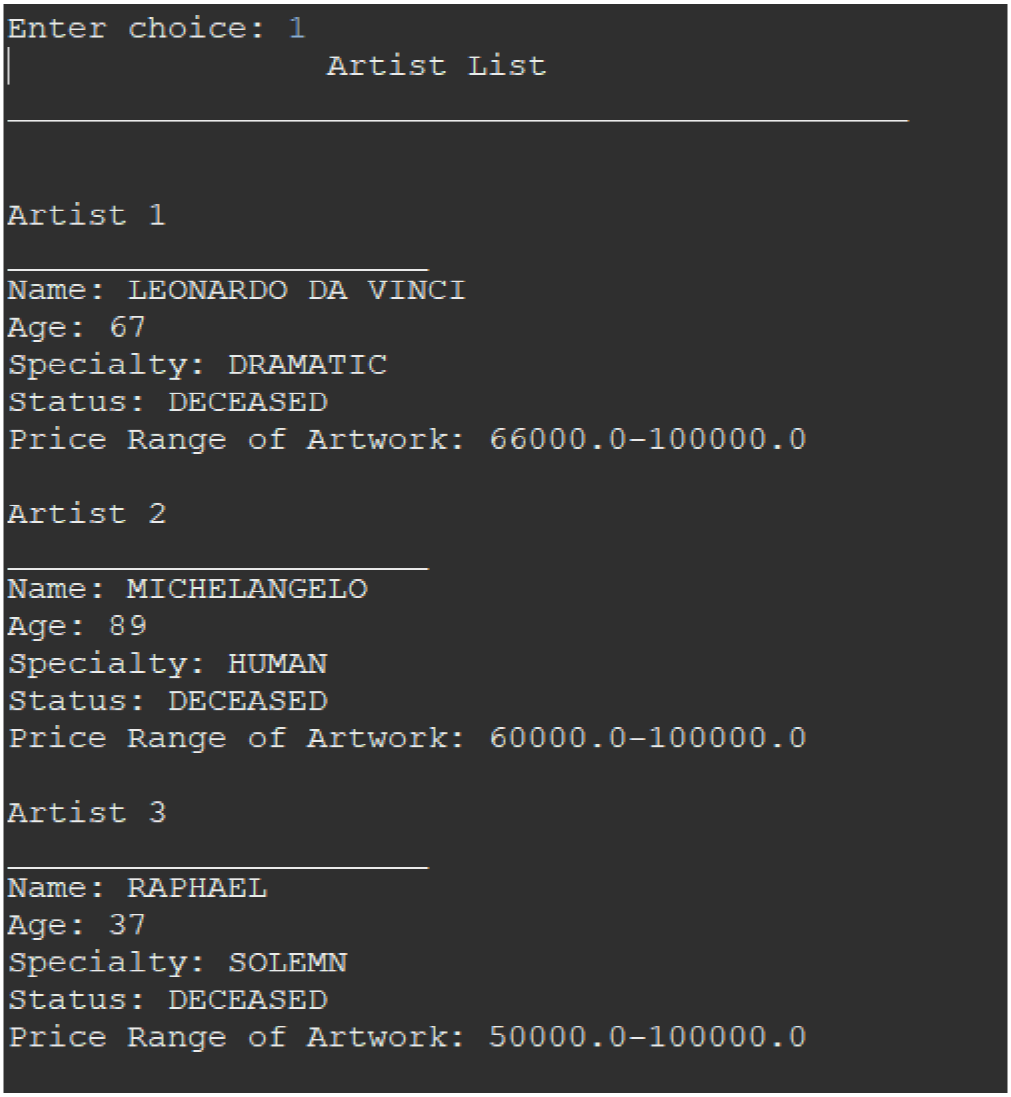
```
Enter choice: 5
Exit successfully

1. Customer Menu
2. Artist Menu
3. Artwork Menu
4. Reset Account
5. Logout
Enter option:
```

OPTION 2: ARTIST MENU



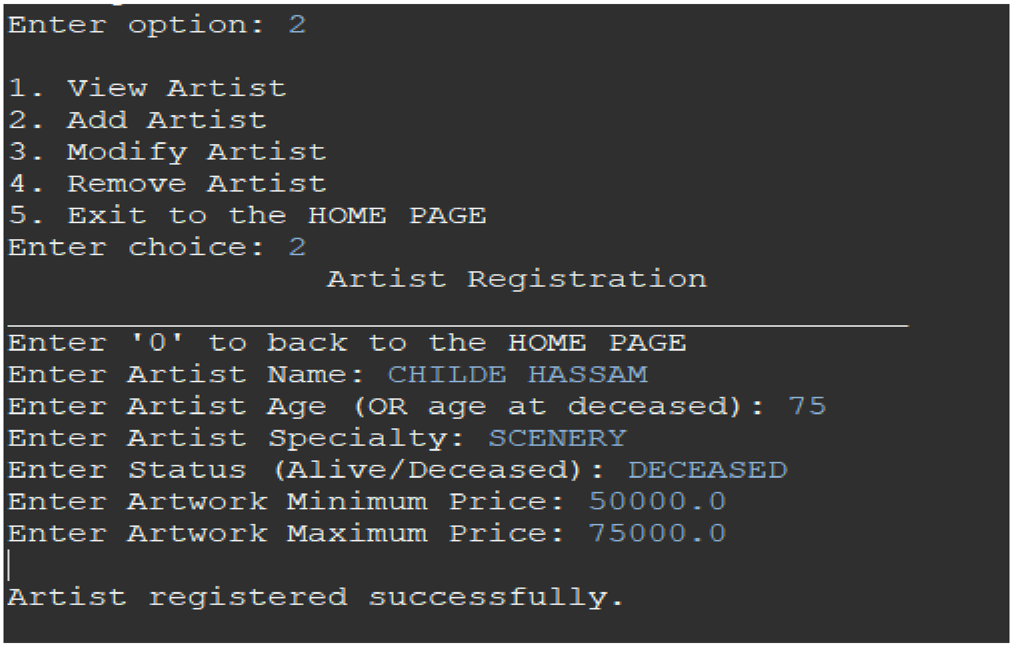
```
Enter option: 2

1. View Artist
2. Add Artist
3. Modify Artist
4. Remove Artist
5. Exit to the HOME PAGE
Enter choice:
```

OPTION 2: CHOICE 1.VIEW ARTIST FUNCTION



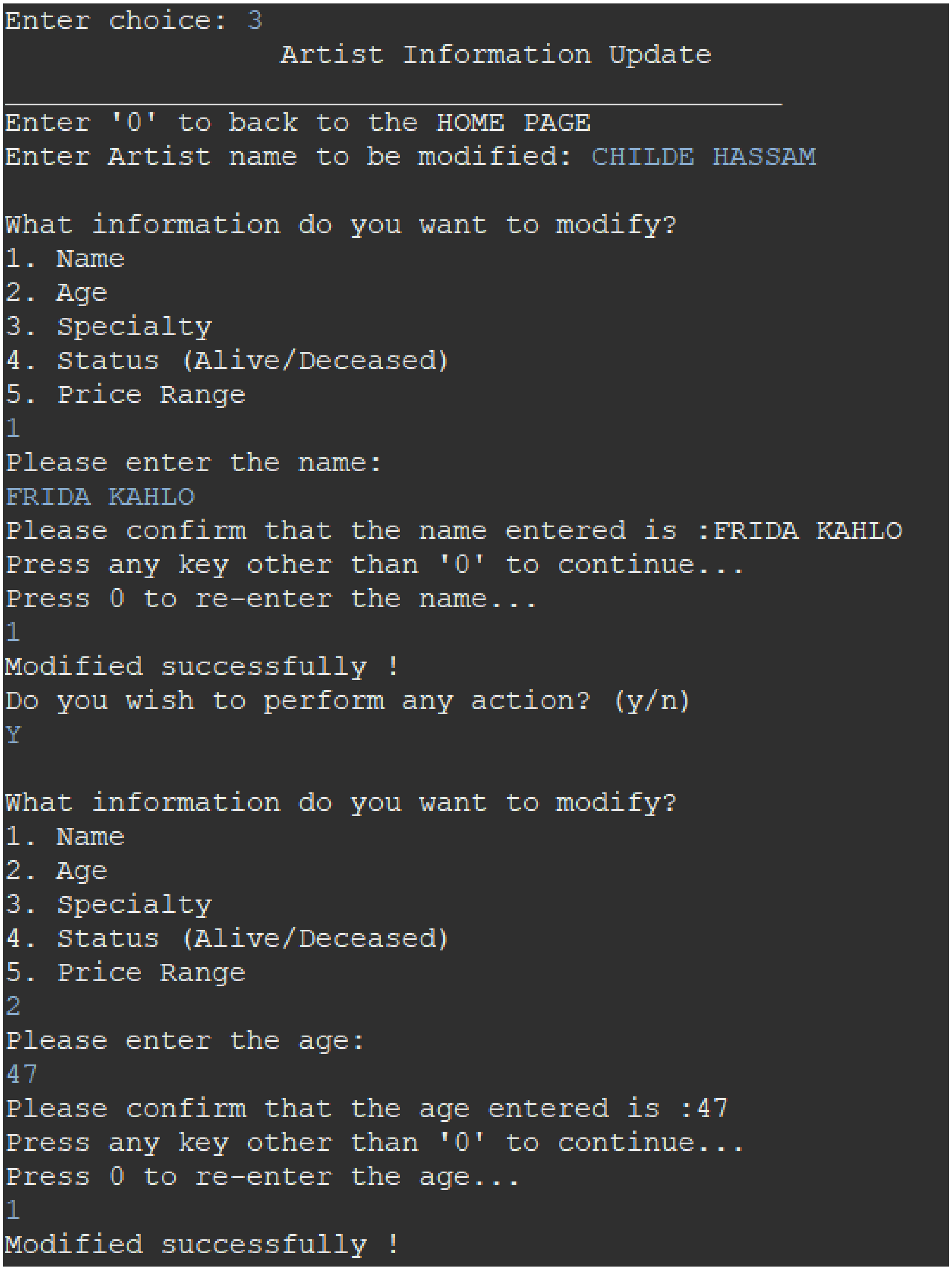
```
Enter choice: 1
                Artist List
_____________________________________________


Artist 1
_____________________
Name: LEONARDO DA VINCI
Age: 67
Specialty: DRAMATIC
Status: DECEASED
Price Range of Artwork: 66000.0-100000.0

Artist 2
_____________________
Name: MICHELANGELO
Age: 89
Specialty: HUMAN
Status: DECEASED
Price Range of Artwork: 60000.0-100000.0

Artist 3
_____________________
Name: RAPHAEL
Age: 37
Specialty: SOLEMN
Status: DECEASED
Price Range of Artwork: 50000.0-100000.0
```

OPTION 2: CHOICE 2.ADD ARTIST FUNCTION

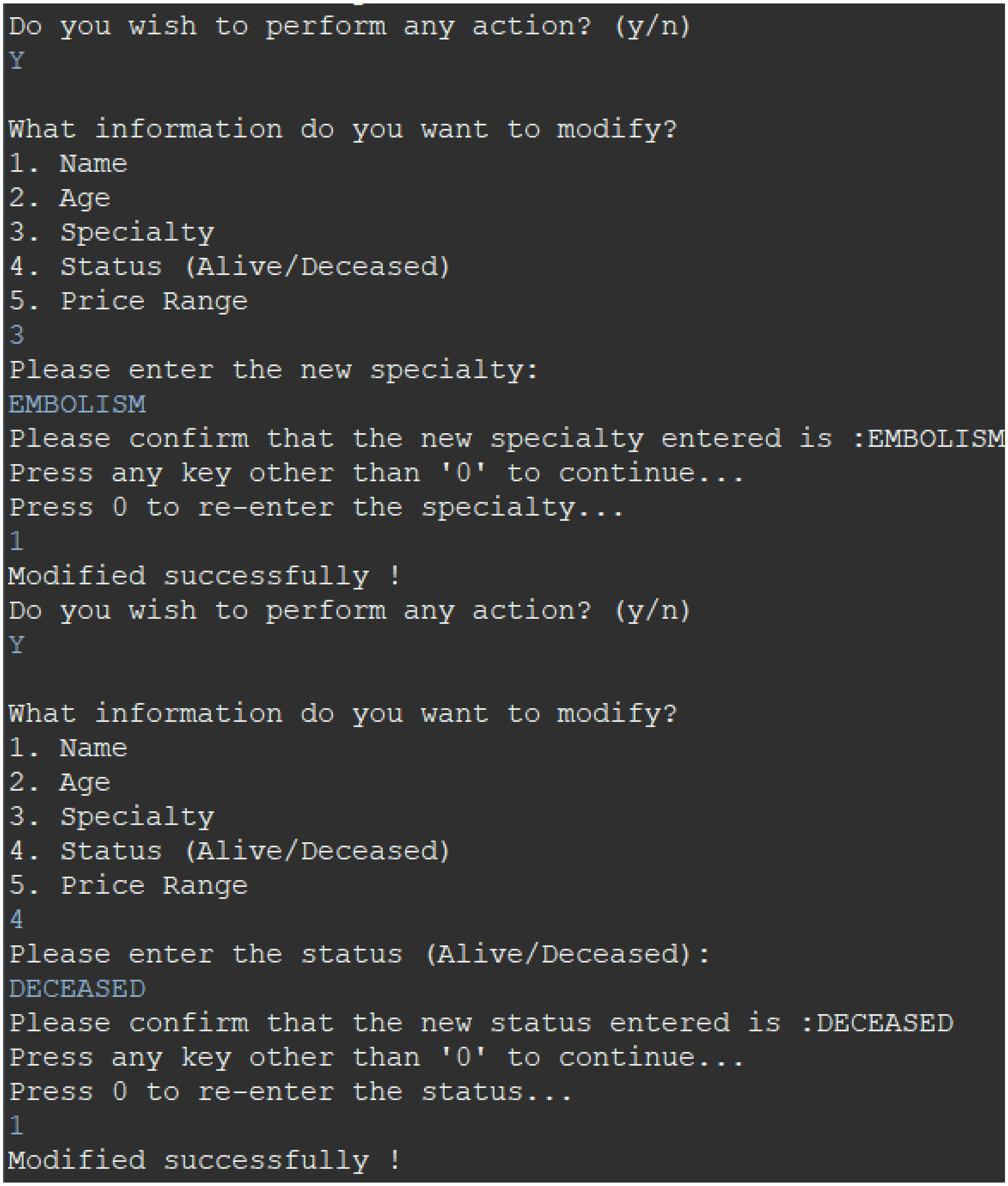


```
Enter option: 2

1. View Artist
2. Add Artist
3. Modify Artist
4. Remove Artist
5. Exit to the HOME PAGE
Enter choice: 2
                Artist Registration
_____________________________________________
Enter '0' to back to the HOME PAGE
Enter Artist Name: CHILDE HASSAM
Enter Artist Age (OR age at deceased): 75
Enter Artist Specialty: SCENERY
Enter Status (Alive/Deceased): DECEASED
Enter Artwork Minimum Price: 50000.0
Enter Artwork Maximum Price: 75000.0

Artist registered successfully.
```

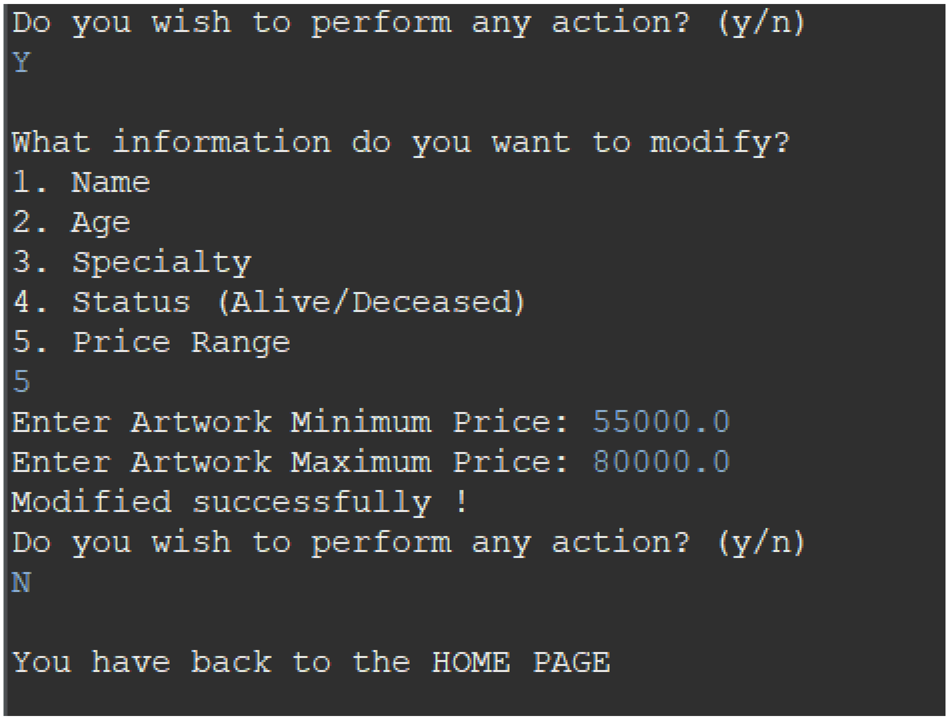
OPTION 2: CHOICE 3.MODIFY ARTIST FUNCTION



```
Enter choice: 3
                Artist Information Update
_____________________________________________
Enter '0' to back to the HOME PAGE
Enter Artist name to be modified: CHILDE HASSAM

What information do you want to modify?
1. Name
2. Age
3. Specialty
4. Status (Alive/Deceased)
5. Price Range
1
Please enter the name:
FRIDA KAHLO
Please confirm that the name entered is :FRIDA KAHLO
Press any key other than '0' to continue...
Press 0 to re-enter the name...
1
Modified successfully !
Do you wish to perform any action? (y/n)
Y

What information do you want to modify?
1. Name
2. Age
3. Specialty
4. Status (Alive/Deceased)
5. Price Range
2
Please enter the age:
47
Please confirm that the age entered is :47
Press any key other than '0' to continue...
Press 0 to re-enter the age...
1
Modified successfully !
```

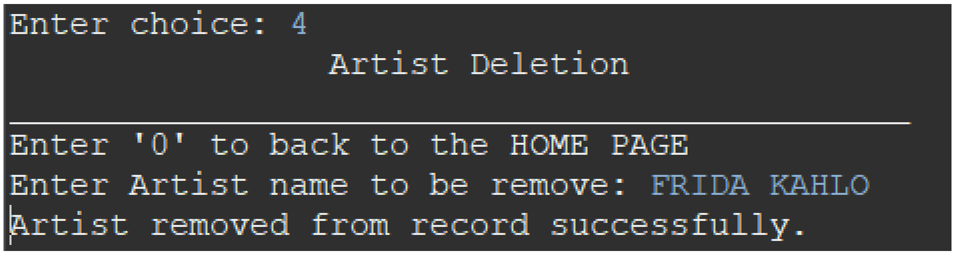


```
Do you wish to perform any action? (y/n)
Y

What information do you want to modify?
1. Name
2. Age
3. Specialty
4. Status (Alive/Deceased)
5. Price Range
3
Please enter the new specialty:
EMBOLISM
Please confirm that the new specialty entered is :EMBOLISM
Press any key other than '0' to continue...
Press 0 to re-enter the specialty...
1
Modified successfully !
Do you wish to perform any action? (y/n)
Y

What information do you want to modify?
1. Name
2. Age
3. Specialty
4. Status (Alive/Deceased)
5. Price Range
4
Please enter the status (Alive/Deceased):
DECEASED
Please confirm that the new status entered is :DECEASED
Press any key other than '0' to continue...
Press 0 to re-enter the status...
1
Modified successfully !
```



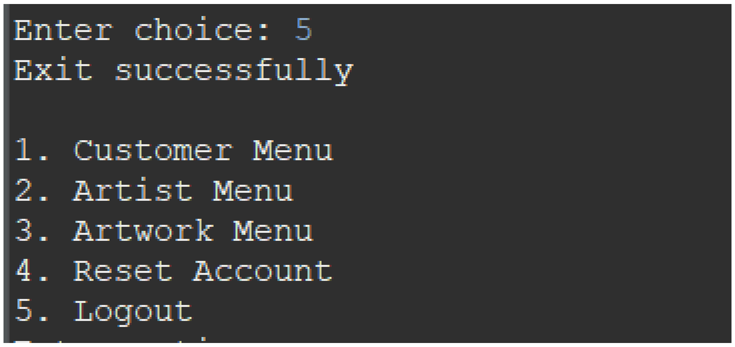
```
Do you wish to perform any action? (y/n)
Y

What information do you want to modify?
1. Name
2. Age
3. Specialty
4. Status (Alive/Deceased)
5. Price Range
5
Enter Artwork Minimum Price: 55000.0
Enter Artwork Maximum Price: 80000.0
Modified successfully !
Do you wish to perform any action? (y/n)
N

You have back to the HOME PAGE
```

OPTION 2: CHOICE 4.REMOVE ARTIST FUNCTION



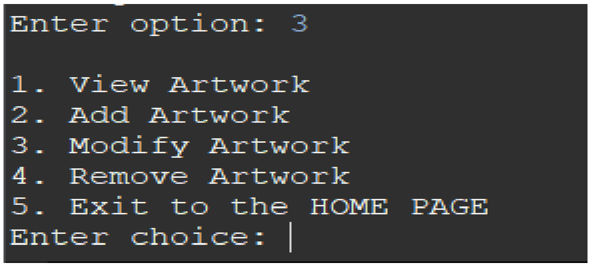
```
Enter choice: 4
                Artist Deletion
_____________________________________________
Enter '0' to back to the HOME PAGE
Enter Artist name to be remove: FRIDA KAHLO
Artist removed from record successfully.
```

OPTION 2: CHOICE 5. EXIT TO HOME PAGE FUNCTION



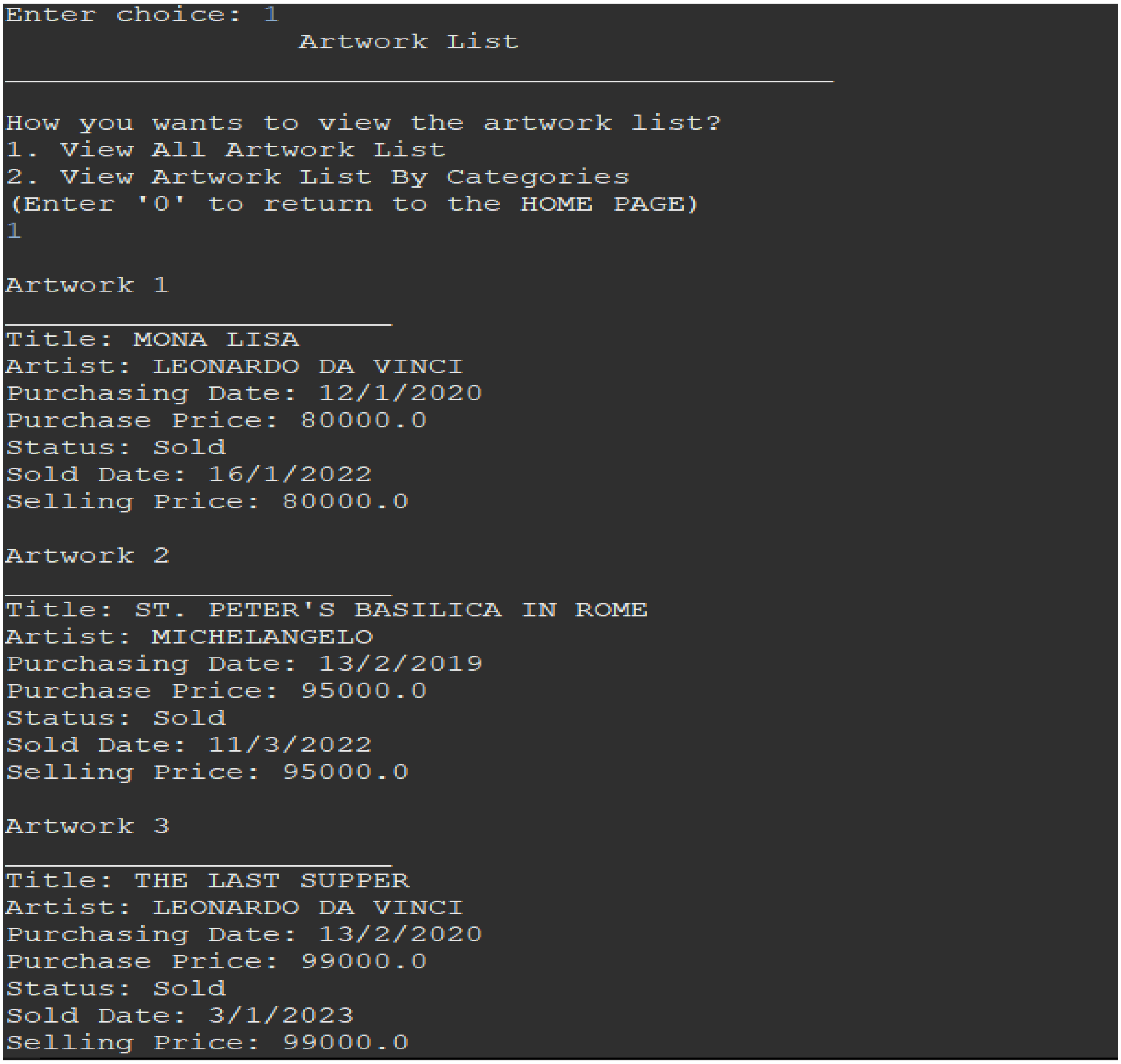
```
Enter choice: 5
Exit successfully

1. Customer Menu
2. Artist Menu
3. Artwork Menu
4. Reset Account
5. Logout
```

OPTION 3: ARTWORK MENU



```
Enter option: 3

1. View Artwork
2. Add Artwork
3. Modify Artwork
4. Remove Artwork
5. Exit to the HOME PAGE
Enter choice: |
```

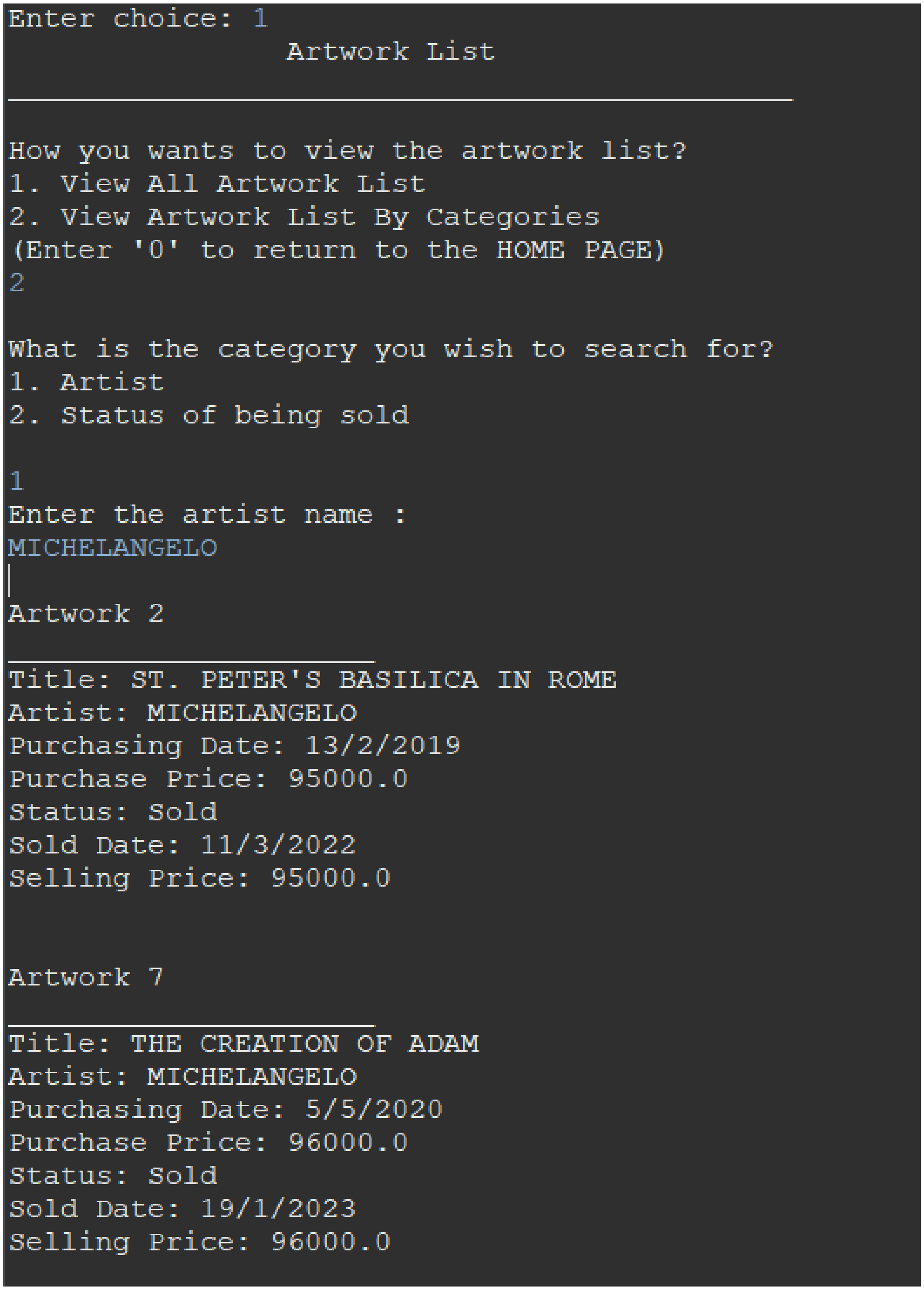
OPTION 3: CHOICE 1.1.VIEW ALL ARTWORK LIST



```
Enter choice: 1
                Artwork List
_____________________________________________

How you wants to view the artwork list?
1. View All Artwork List
2. View Artwork List By Categories
(Enter '0' to return to the HOME PAGE)
1

Artwork 1
_____________________
Title: MONA LISA
Artist: LEONARDO DA VINCI
Purchasing Date: 12/1/2020
Purchase Price: 80000.0
Status: Sold
Sold Date: 16/1/2022
Selling Price: 80000.0

Artwork 2
_____________________
Title: ST. PETER'S BASILICA IN ROME
Artist: MICHELANGELO
Purchasing Date: 13/2/2019
Purchase Price: 95000.0
Status: Sold
Sold Date: 11/3/2022
Selling Price: 95000.0

Artwork 3
_____________________
Title: THE LAST SUPPER
Artist: LEONARDO DA VINCI
Purchasing Date: 13/2/2020
Purchase Price: 99000.0
Status: Sold
Sold Date: 3/1/2023
Selling Price: 99000.0
```

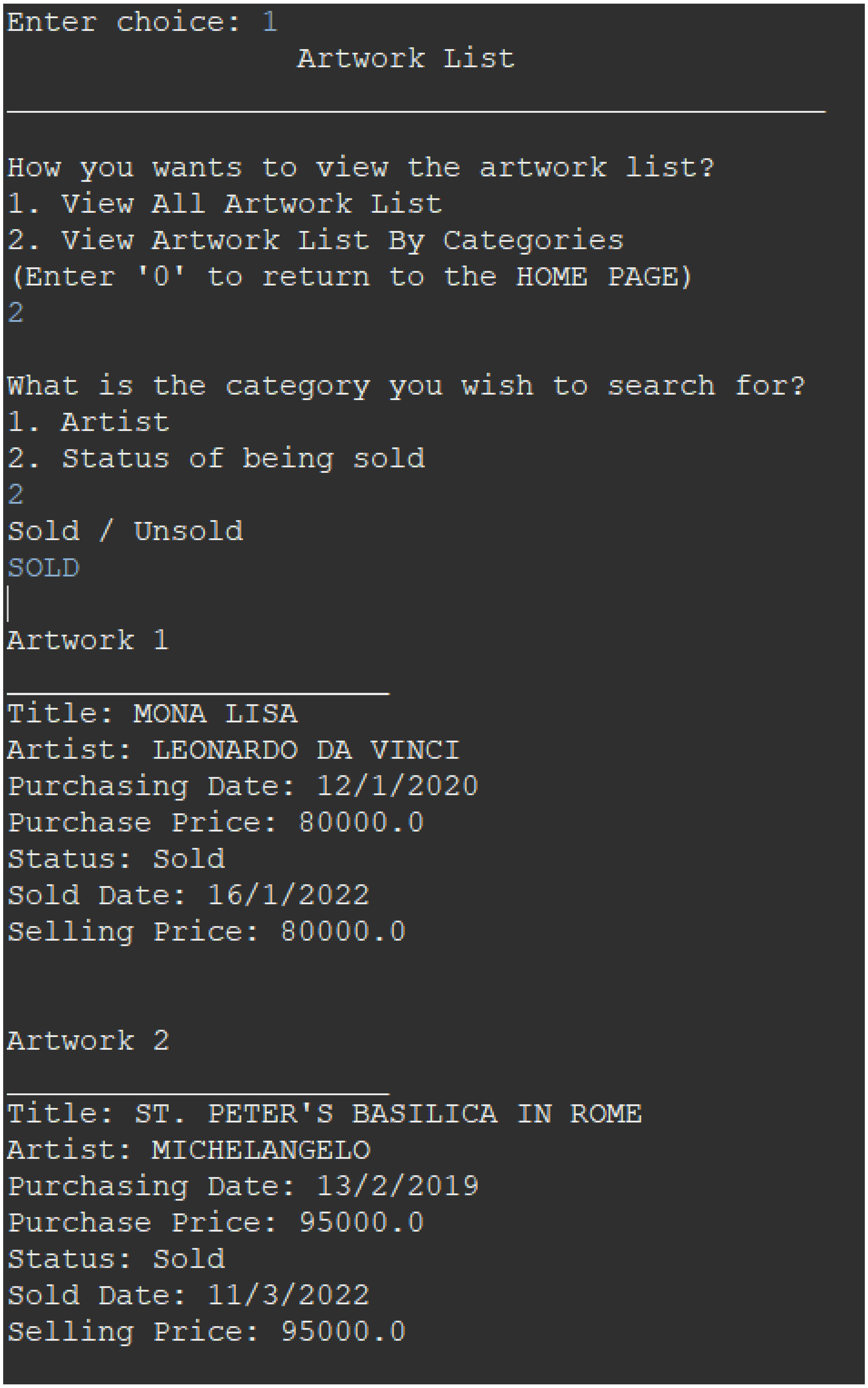
OPTION 3: CHOICE 1.2VIEW ARTWORK LIST BY CATEGORIES FUNCTION

CATEGORY TO SEARCH: ARTIST NAME

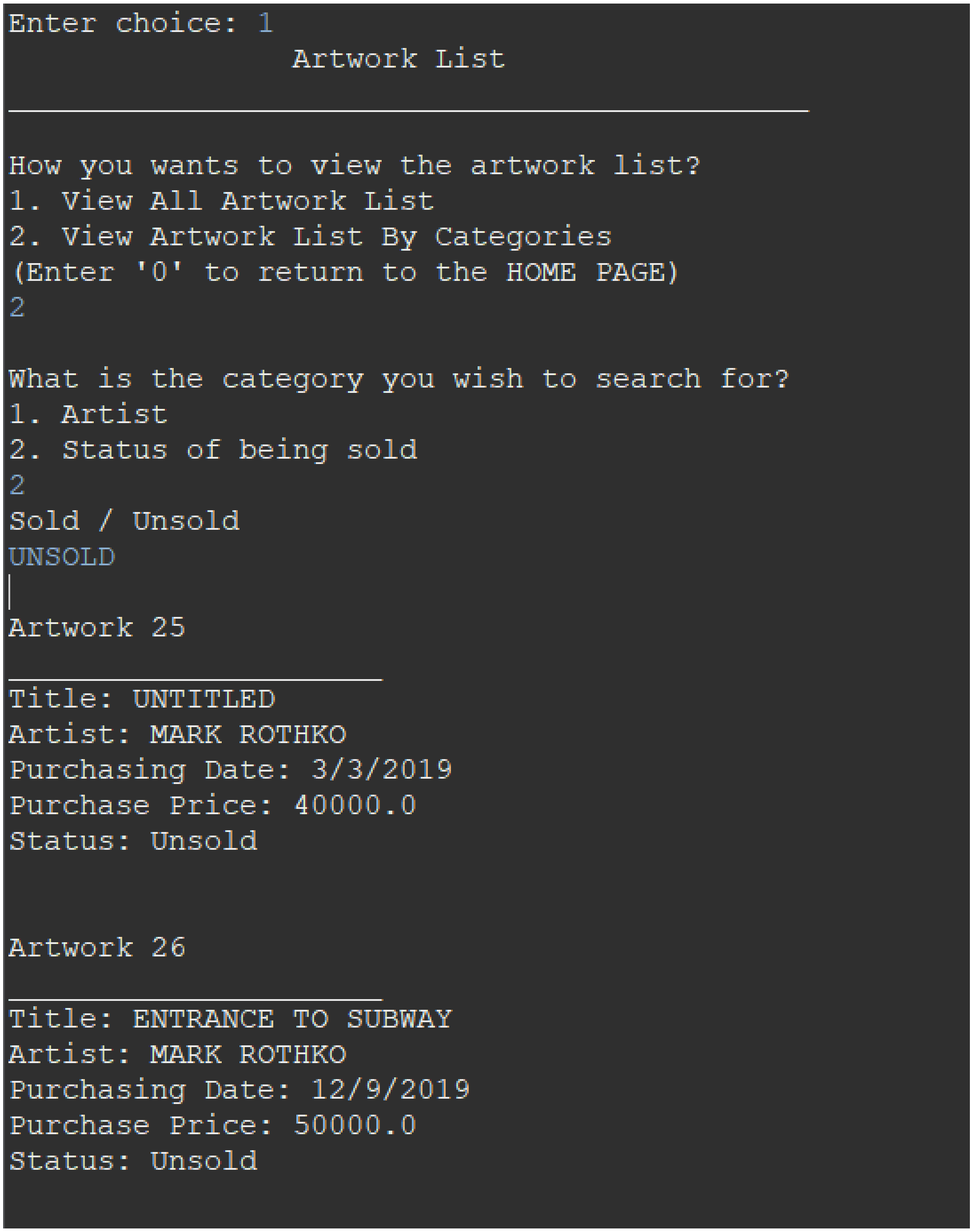


```
Enter choice: 1
                Artwork List
_____________________________________________

How you wants to view the artwork list?
1. View All Artwork List
2. View Artwork List By Categories
(Enter '0' to return to the HOME PAGE)
2

What is the category you wish to search for?
1. Artist
2. Status of being sold

1
Enter the artist name :
MICHELANGELO

Artwork 2
_____________________
Title: ST. PETER'S BASILICA IN ROME
Artist: MICHELANGELO
Purchasing Date: 13/2/2019
Purchase Price: 95000.0
Status: Sold
Sold Date: 11/3/2022
Selling Price: 95000.0


Artwork 7
_____________________
Title: THE CREATION OF ADAM
Artist: MICHELANGELO
Purchasing Date: 5/5/2020
Purchase Price: 96000.0
Status: Sold
Sold Date: 19/1/2023
Selling Price: 96000.0
```

CATEGORY TO SEARCH: STATUS OF BEING SOLD OR UNSOLD

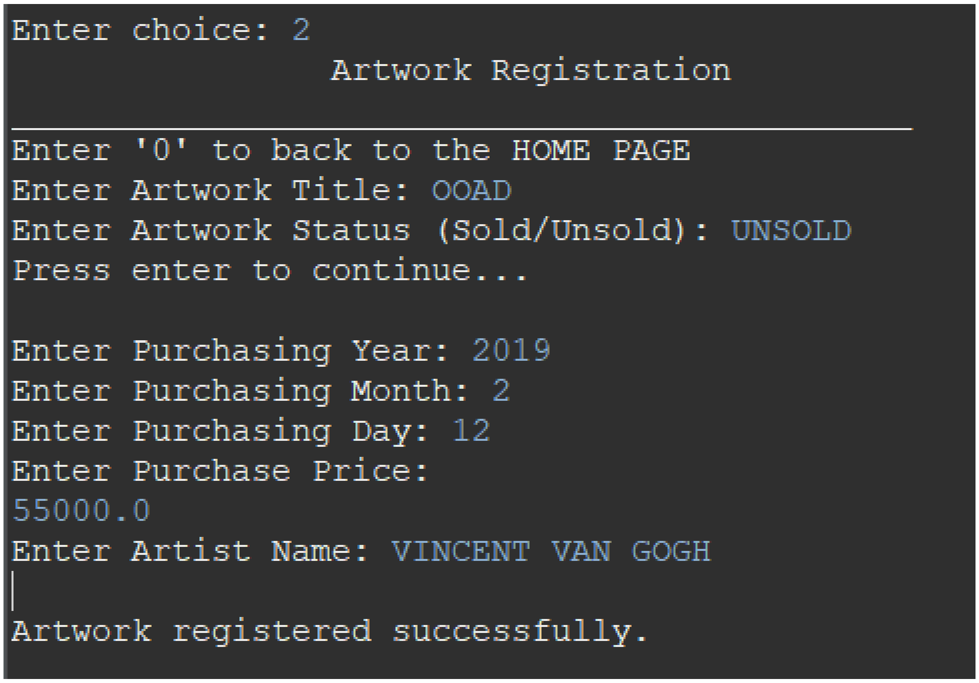


```
Enter choice: 1
                Artwork List

_____________________________________________

How you wants to view the artwork list?
1. View All Artwork List
2. View Artwork List By Categories
(Enter '0' to return to the HOME PAGE)
2

What is the category you wish to search for?
1. Artist
2. Status of being sold
2
Sold / Unsold
SOLD

Artwork 1

_____________________
Title: MONA LISA
Artist: LEONARDO DA VINCI
Purchasing Date: 12/1/2020
Purchase Price: 80000.0
Status: Sold
Sold Date: 16/1/2022
Selling Price: 80000.0


Artwork 2

_____________________
Title: ST. PETER'S BASILICA IN ROME
Artist: MICHELANGELO
Purchasing Date: 13/2/2019
Purchase Price: 95000.0
Status: Sold
Sold Date: 11/3/2022
Selling Price: 95000.0
```



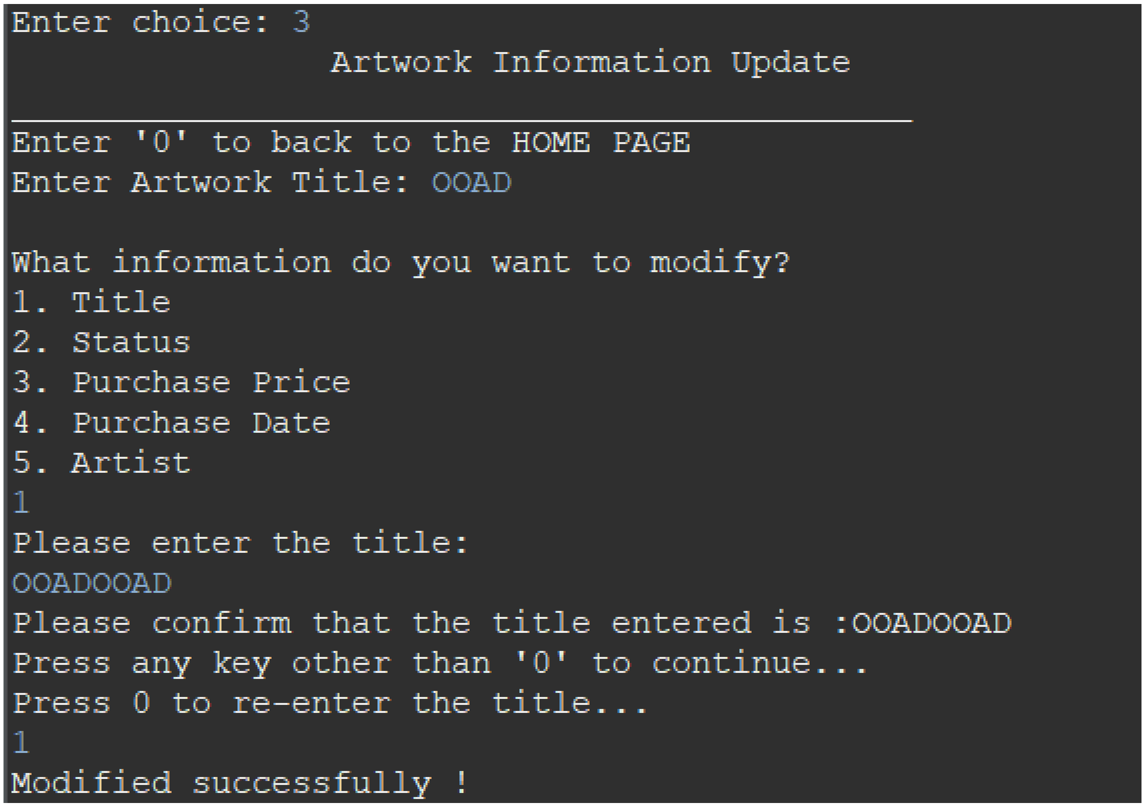
```
Enter choice: 1
                Artwork List
_____________________________________________

How you wants to view the artwork list?
1. View All Artwork List
2. View Artwork List By Categories
(Enter '0' to return to the HOME PAGE)
2

What is the category you wish to search for?
1. Artist
2. Status of being sold
2
Sold / Unsold
UNSOLD

Artwork 25
_____________________
Title: UNTITLED
Artist: MARK ROTHKO
Purchasing Date: 3/3/2019
Purchase Price: 40000.0
Status: Unsold


Artwork 26
_____________________
Title: ENTRANCE TO SUBWAY
Artist: MARK ROTHKO
Purchasing Date: 12/9/2019
Purchase Price: 50000.0
Status: Unsold
```

OPTION 3: CHOICE 2.ADD ARTWORK FUNCTION

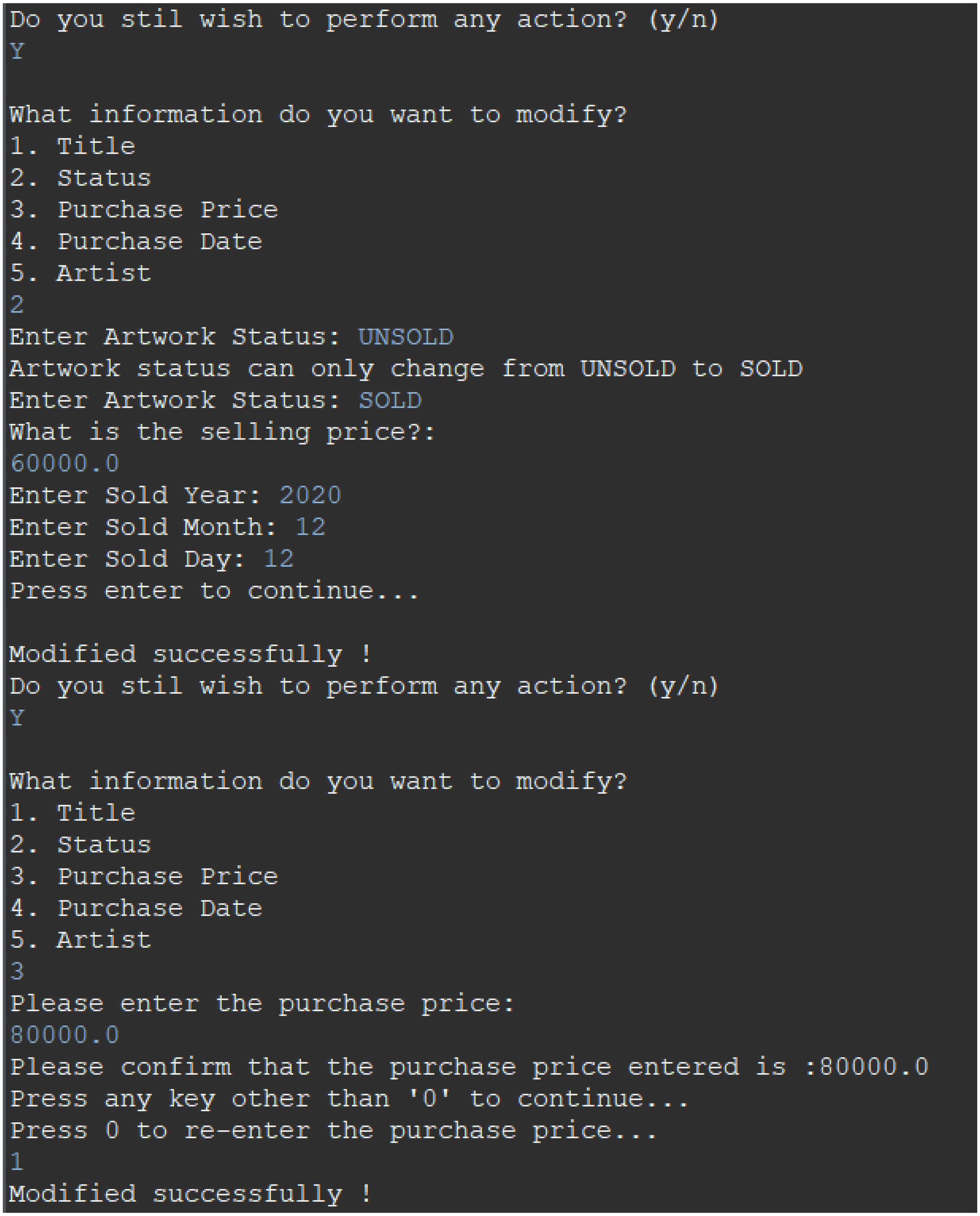


```
Enter choice: 2
                Artwork Registration
_____________________________________________
Enter '0' to back to the HOME PAGE
Enter Artwork Title: OOAD
Enter Artwork Status (Sold/Unsold): UNSOLD
Press enter to continue...

Enter Purchasing Year: 2019
Enter Purchasing Month: 2
Enter Purchasing Day: 12
Enter Purchase Price:
55000.0
Enter Artist Name: VINCENT VAN GOGH

Artwork registered successfully.
```

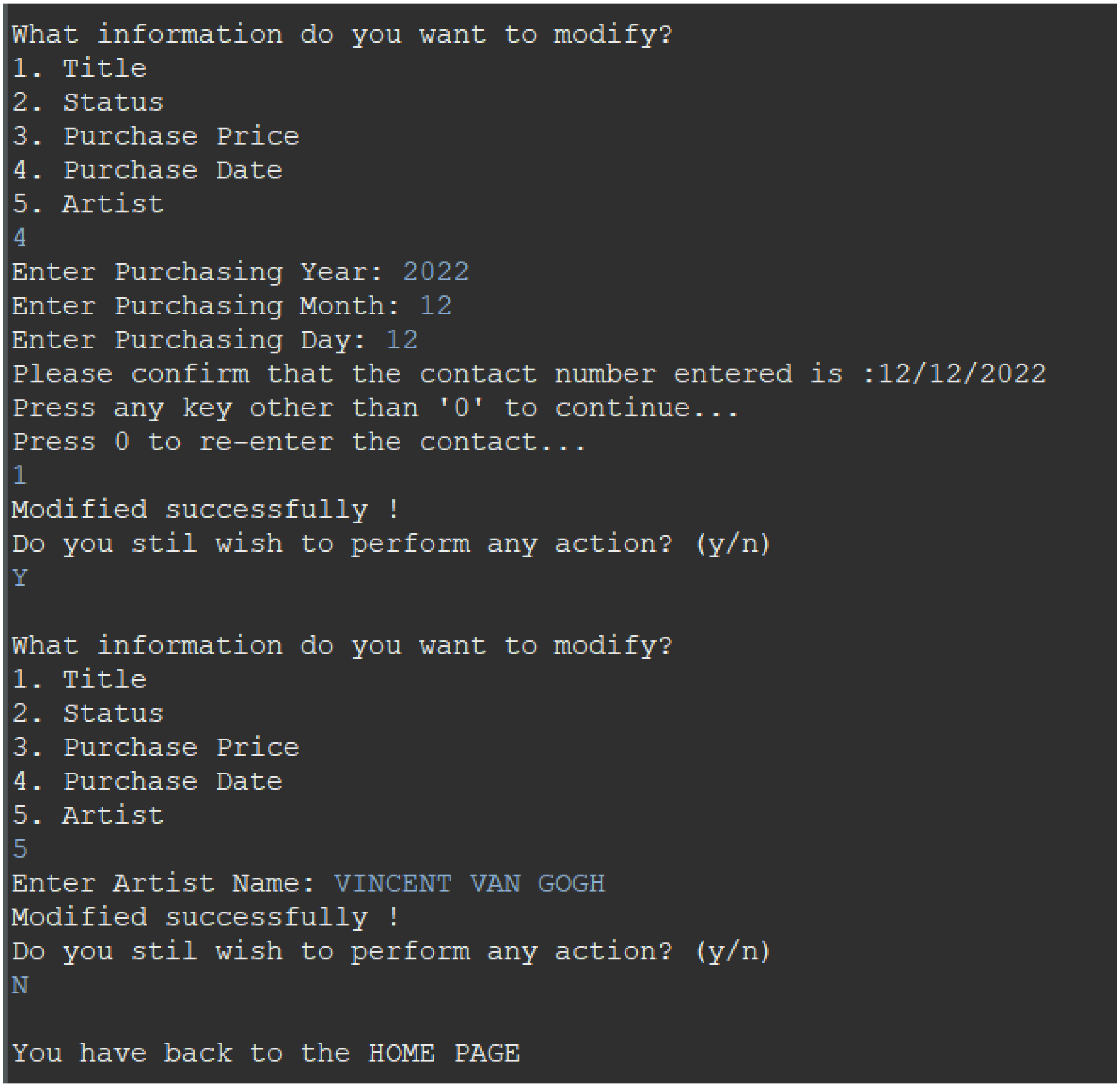
OPTION 3: CHOICE 3.MODIFY ARTWORK FUNCTION



```
Enter choice: 3
                Artwork Information Update
_____________________________________________
Enter '0' to back to the HOME PAGE
Enter Artwork Title: OOAD

What information do you want to modify?
1. Title
2. Status
3. Purchase Price
4. Purchase Date
5. Artist
1
Please enter the title:
OOADOOAD
Please confirm that the title entered is :OOADOOAD
Press any key other than '0' to continue...
Press 0 to re-enter the title...
1
Modified successfully !
```

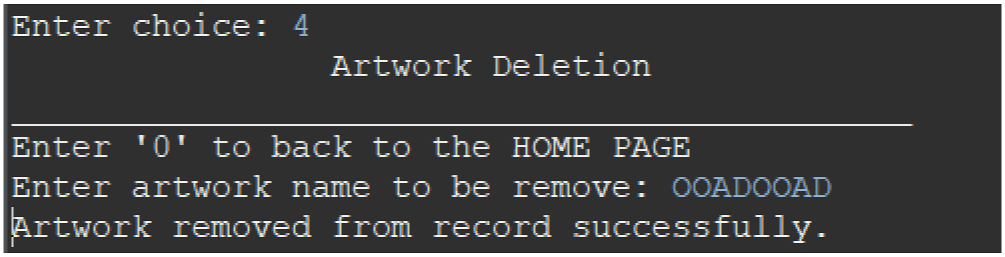


```
Do you stil wish to perform any action? (y/n)
Y

What information do you want to modify?
1. Title
2. Status
3. Purchase Price
4. Purchase Date
5. Artist
2
Enter Artwork Status: UNSOLD
Artwork status can only change from UNSOLD to SOLD
Enter Artwork Status: SOLD
What is the selling price?:
60000.0
Enter Sold Year: 2020
Enter Sold Month: 12
Enter Sold Day: 12
Press enter to continue...

Modified successfully !
Do you stil wish to perform any action? (y/n)
Y

What information do you want to modify?
1. Title
2. Status
3. Purchase Price
4. Purchase Date
5. Artist
3
Please enter the purchase price:
80000.0
Please confirm that the purchase price entered is :80000.0
Press any key other than '0' to continue...
Press 0 to re-enter the purchase price...
1
Modified successfully !
```



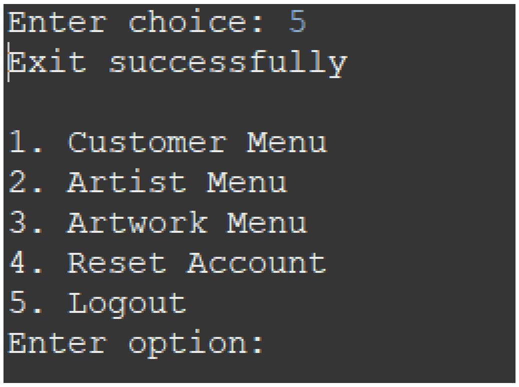
```
What information do you want to modify?
1. Title
2. Status
3. Purchase Price
4. Purchase Date
5. Artist
4
Enter Purchasing Year: 2022
Enter Purchasing Month: 12
Enter Purchasing Day: 12
Please confirm that the contact number entered is :12/12/2022
Press any key other than '0' to continue...
Press 0 to re-enter the contact...
1
Modified successfully !
Do you stil wish to perform any action? (y/n)
Y

What information do you want to modify?
1. Title
2. Status
3. Purchase Price
4. Purchase Date
5. Artist
5
Enter Artist Name: VINCENT VAN GOGH
Modified successfully !
Do you stil wish to perform any action? (y/n)
N

You have back to the HOME PAGE
```

OPTION 3: CHOICE 4. REMOVE ARTWORK FUNCTION



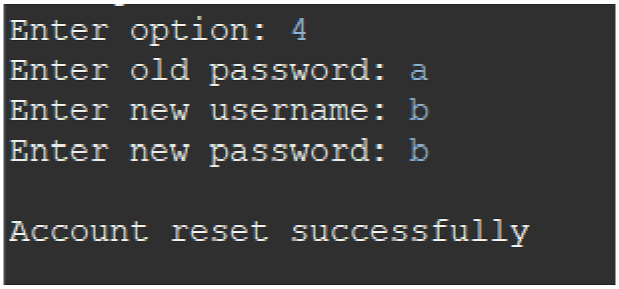
```
Enter choice: 4
                Artwork Deletion

_____________________________________________
Enter '0' to back to the HOME PAGE
Enter artwork name to be remove: OOADOOAD
Artwork removed from record successfully.
```

OPTION 3: CHOICE 5. EXIT TO HOME PAGE FUNCTION



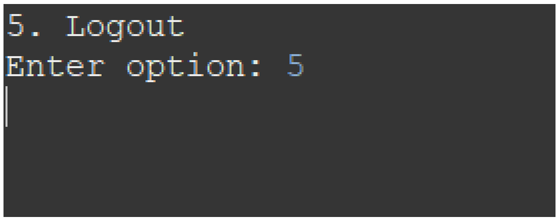
```
Enter choice: 5
Exit successfully

1. Customer Menu
2. Artist Menu
3. Artwork Menu
4. Reset Account
5. Logout
Enter option:
```

OPTION 4: RESET ACCOUNT MENU



```
Enter option: 4
Enter old password: a
Enter new username: b
Enter new password: b

Account reset successfully
```

OPTION 5: LOGOUT MENU



```
5. Logout
Enter option: 5
```