ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.Ι ΗΡΑΚΛΕΙΟΥ ΚΡΗΤΗΣ

E-SHOP ΓΙΑ ΥΠΟΛΟΓΙΣΤΕΣ & ΗΛΕΚΤΡΟΝΙΚΑ ΕΙΔΗ

**Ο ΣΥΝΟΛΙΚΟΣ ΚΩΔΙΚΑΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ**

**AddCamera.java**

package stavroulakis\_vasilis\_thesis\_2018;

import java.awt.BorderLayout;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileWriter;

import java.io.IOException;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JTextField;

public class AddCamera implements ActionListener {

@Override

public void actionPerformed(ActionEvent ae) {

JFrame frame = new JFrame("Add Camera");

frame.setSize(250, 340);

frame.setLayout(new BorderLayout());

JPanel panel = new JPanel();

frame.add(panel);

panel.setLayout(new GridLayout(17, 1));

JLabel label1 = new JLabel("Camera model");

JLabel label2 = new JLabel("Camera monitor size");

JLabel label3 = new JLabel("Camera megapixels");

JLabel label4 = new JLabel("Camera frames/second");

JLabel label5 = new JLabel("Camera recording type");

JLabel label6 = new JLabel("Camera price");

JLabel label7 = new JLabel("Store name");

JLabel label8 = new JLabel("Camera quantity");

JTextField field1 = new JTextField(25);

JTextField field2 = new JTextField(25);

JTextField field3 = new JTextField(25);

JTextField field4 = new JTextField(25);

JTextField field5 = new JTextField(25);

JTextField field6 = new JTextField(25);

JTextField field7 = new JTextField(25);

JTextField field8 = new JTextField(25);

JButton btn = new JButton("ADD");

panel.add(label1);

panel.add(field1);

panel.add(label2);

panel.add(field2);

panel.add(label3);

panel.add(field3);

panel.add(label4);

panel.add(field4);

panel.add(label5);

panel.add(field5);

panel.add(label6);

panel.add(field6);

panel.add(label7);

panel.add(field7);

panel.add(label8);

panel.add(field8);

panel.add(btn);

btn.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ae) {

String text1 = field1.getText();

String text2 = field2.getText();

String text3 = field3.getText();

String text4 = field4.getText();

String text5 = field5.getText();

String text6 = field6.getText();

String text7 = field7.getText();

String text8 = field8.getText();

//Add by using quantity given

int quantity = Integer.parseInt(text8);

for (int i = 0; i < quantity; i++) {

CameraClass new\_camera = new CameraClass(text1, text2, text3, text4, text5, text6, text7);

writeCamera(new\_camera);

Log.i("YOU ADDED A CAMERA ITEM");

}

frame.dispose();

}

});

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.setResizable(false);

}

public void writeCamera(CameraClass c) {

try {

String current = new java.io.File(".").getCanonicalPath();

File file = new File(current + "\\cameras.txt");

if (!file.exists()) {

file.createNewFile();

}

FileWriter fw = new FileWriter(file, true);

BufferedWriter bw = new BufferedWriter(fw);

bw.write("\n" + c.toString());

bw.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

**AddCompany.java**

package stavroulakis\_vasilis\_thesis\_2018;

import java.awt.BorderLayout;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileWriter;

import java.io.IOException;

import java.util.ArrayList;

import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JTextField;

public class AddCompany implements ActionListener {

@Override

public void actionPerformed(ActionEvent ae) {

JFrame frame = new JFrame("Add Company");

frame.setSize(250, 300);

frame.setLayout(new BorderLayout());

JPanel panel = new JPanel();

frame.add(panel);

panel.setLayout(new GridLayout(9, 1));

JLabel label1 = new JLabel("Company name");

JLabel label2 = new JLabel("Company owner");

JLabel label3 = new JLabel("Company ITN");

JLabel label4 = new JLabel("Company location");

JTextField field1 = new JTextField(25);

JTextField field2 = new JTextField(25);

JTextField field3 = new JTextField(25);

JTextField field4 = new JTextField(25);

JButton btn = new JButton("ADD");

panel.add(label1);

panel.add(field1);

panel.add(label2);

panel.add(field2);

panel.add(label3);

panel.add(field3);

panel.add(label4);

panel.add(field4);

panel.add(btn);

btn.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ae) {

String text1 = field1.getText();

String text2 = field2.getText();

String text3 = field3.getText();

String text4 = field4.getText();

//befor adding check if ITN belongs to our system

Scanner file = null;

ArrayList<CompanyClass> companies\_list = new ArrayList<>();

try {

file = new Scanner(new File("companies.txt"));

} catch (FileNotFoundException ex) {

Logger.getLogger(AddCompany.class.getName()).log(Level.SEVERE, null, ex);

}

//we try to find the given company using scanner function, an array and reconstruct the companies\_list

while (file.hasNext()) {

String line = file.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String name = stringArr[0];

String owner = stringArr[1];

String ITN = stringArr[2];

String location = stringArr[3];

CompanyClass companies = new CompanyClass(name, owner, ITN, location);

companies\_list.add(companies);

}

}

int j = 0;

for (int i = 0; i < companies\_list.size(); i++) {

CompanyClass cc = companies\_list.get(i);

if (cc.getCompany\_ITN().equals(text3)) {

j = i + 1;

}

}

if (j <= 0) {

CompanyClass new\_company = new CompanyClass(text1, text2, text3, text4);

writeCompany(new\_company);

}

if (j > 0) {

Log.i("THIS COMPANY IS IN OUR SYSTEM!");

}

frame.dispose();

}

});

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.setResizable(false);

}

public void writeCompany(CompanyClass c) {

try {

String current = new java.io.File(".").getCanonicalPath();

File file = new File(current + "\\companies.txt");

if (!file.exists()) {

file.createNewFile();

}

FileWriter fw = new FileWriter(file, true);

BufferedWriter bw = new BufferedWriter(fw);

bw.write("\n" + c.toString());

bw.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

**AddGamingConsole.java**

package stavroulakis\_vasilis\_thesis\_2018;

import java.awt.BorderLayout;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileWriter;

import java.io.IOException;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JTextField;

public class AddGamingConsole implements ActionListener {

@Override

public void actionPerformed(ActionEvent ae) {

JFrame frame = new JFrame("Add Gaming Console");

frame.setSize(320, 340);

frame.setLayout(new BorderLayout());

JPanel panel = new JPanel();

frame.add(panel);

panel.setLayout(new GridLayout(15, 1));

JLabel label1 = new JLabel("Console model");

JLabel label2 = new JLabel("Console HD size");

JLabel label3 = new JLabel("Console year released");

JLabel label4 = new JLabel("Console controllers");

JLabel label5 = new JLabel("Console price");

JLabel label6 = new JLabel("Store name");

JLabel label7 = new JLabel("Console quantity");

JTextField field1 = new JTextField(25);

JTextField field2 = new JTextField(25);

JTextField field3 = new JTextField(25);

JTextField field4 = new JTextField(25);

JTextField field5 = new JTextField(25);

JTextField field6 = new JTextField(25);

JTextField field7 = new JTextField(25);

JButton btn = new JButton("ADD");

panel.add(label1);

panel.add(field1);

panel.add(label2);

panel.add(field2);

panel.add(label3);

panel.add(field3);

panel.add(label4);

panel.add(field4);

panel.add(label5);

panel.add(field5);

panel.add(label6);

panel.add(field6);

panel.add(label7);

panel.add(field7);

panel.add(btn);

btn.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ae) {

String text1 = field1.getText();

String text2 = field2.getText();

String text3 = field3.getText();

String text4 = field4.getText();

String text5 = field5.getText();

String text6 = field6.getText();

String text7 = field7.getText();

//Add by using quantity given

int quantity = Integer.parseInt(text7);

for (int i = 0; i < quantity; i++) {

GamingConsoleClass new\_console = new GamingConsoleClass(text1, text2, text3, text4, text5, text6);

writeGamingConsole(new\_console);

Log.i("YOU ADDED A GAMING CONSOLE ITEM");

}

frame.dispose();

}

});

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.setResizable(false);

}

public void writeGamingConsole(GamingConsoleClass c) {

try {

String current = new java.io.File(".").getCanonicalPath();

File file = new File(current + "\\gaming\_consoles.txt");

if (!file.exists()) {

file.createNewFile();

}

FileWriter fw = new FileWriter(file, true);

BufferedWriter bw = new BufferedWriter(fw);

bw.write("\n" + c.toString());

bw.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

**AddMobile.java**

package stavroulakis\_vasilis\_thesis\_2018;

import java.awt.BorderLayout;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileWriter;

import java.io.IOException;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JTextField;

public class AddMobile implements ActionListener {

@Override

public void actionPerformed(ActionEvent ae) {

JFrame frame = new JFrame("Add Mobile");

frame.setSize(250, 360);

frame.setLayout(new BorderLayout());

JPanel panel = new JPanel();

frame.add(panel);

panel.setLayout(new GridLayout(19, 1));

JLabel label1 = new JLabel("Mobile model");

JLabel label2 = new JLabel("Mobile screen size");

JLabel label3 = new JLabel("Mobile front camera");

JLabel label4 = new JLabel("Mobile back camera");

JLabel label5 = new JLabel("Mobile cpu");

JLabel label6 = new JLabel("Mobile storage");

JLabel label7 = new JLabel("Store price");

JLabel label8 = new JLabel("Store name");

JLabel label9 = new JLabel("Mobile quantity");

JTextField field1 = new JTextField(25);

JTextField field2 = new JTextField(25);

JTextField field3 = new JTextField(25);

JTextField field4 = new JTextField(25);

JTextField field5 = new JTextField(25);

JTextField field6 = new JTextField(25);

JTextField field7 = new JTextField(25);

JTextField field8 = new JTextField(25);

JTextField field9 = new JTextField(25);

JButton btn = new JButton("ADD");

panel.add(label1);

panel.add(field1);

panel.add(label2);

panel.add(field2);

panel.add(label3);

panel.add(field3);

panel.add(label4);

panel.add(field4);

panel.add(label5);

panel.add(field5);

panel.add(label6);

panel.add(field6);

panel.add(label7);

panel.add(field7);

panel.add(label8);

panel.add(field8);

panel.add(label9);

panel.add(field9);

panel.add(btn);

btn.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ae) {

String text1 = field1.getText();

String text2 = field2.getText();

String text3 = field3.getText();

String text4 = field4.getText();

String text5 = field5.getText();

String text6 = field6.getText();

String text7 = field7.getText();

String text8 = field8.getText();

String text9 = field9.getText();

//Add by using quantity given

int quantity = Integer.parseInt(text9);

for (int i = 0; i < quantity; i++) {

MobileClass new\_camera = new MobileClass(text1, text2, text3, text4, text5, text6, text7, text8);

writeCamera(new\_camera);

Log.i("YOU ADDED A MOBILE ITEM");

}

frame.dispose();

}

});

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.setResizable(false);

}

public void writeCamera(MobileClass c) {

try {

String current = new java.io.File(".").getCanonicalPath();;

File file = new File(current + "\\mobiles.txt");

if (!file.exists()) {

file.createNewFile();

}

FileWriter fw = new FileWriter(file, true);

BufferedWriter bw = new BufferedWriter(fw);

bw.write("\n" + c.toString());

bw.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

**AddPC.java**

package stavroulakis\_vasilis\_thesis\_2018;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class AdminMainPanel {

public static void adminMainPanelMethod() {

JFrame frame = new JFrame("Admin Panel");

frame.setSize(360, 400);

frame.setLayout(new BorderLayout());

JPanel panel = new JPanel();

frame.add(panel);

panel.setLayout(new GridLayout(10, 1));

JButton btn1 = new JButton("PRINT PRODUCTS");

JButton btn2 = new JButton("ADD PRODUCTS");

JButton btn3 = new JButton("DELETE PRODUCTS");

JButton btn4 = new JButton("PRINT CUSTOMERS");

JButton btn5 = new JButton("ADD COMPANY");

JButton btn6 = new JButton("REMOVE COMPANY");

JButton btn7 = new JButton("PRINT COMPANIES");

JButton btn8 = new JButton("PRINT ORDERS");

JButton btn9 = new JButton("FINISH AN ORDER");

JButton btn10 = new JButton("LOGOUT");

panel.add(btn1);

panel.add(btn2);

panel.add(btn3);

panel.add(btn4);

panel.add(btn5);

panel.add(btn6);

panel.add(btn7);

panel.add(btn8);

panel.add(btn9);

panel.add(btn10);

//actionListeners for buttons

btn1.addActionListener(new PrintProducts());

btn2.addActionListener(new AddProducts());

btn3.addActionListener(new DeleteProducts());

btn4.addActionListener(new PrintCustomers());

btn5.addActionListener(new AddCompany());

btn6.addActionListener(new RemoveCompany());

btn7.addActionListener(new PrintCompanies());

btn8.addActionListener(new PrintFinishedOrders());

btn9.addActionListener(new FinishCustomerOrder());

btn10.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ae) {

frame.dispose();

}

});

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.setResizable(false);

}

}

**AddProducts.java**

package stavroulakis\_vasilis\_thesis\_2018;

import java.awt.BorderLayout;

import java.awt.GridLayout;

import java.awt.event.\*;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JPanel;

public class AddProducts implements ActionListener {

@Override

public void actionPerformed(ActionEvent ae) {

JFrame frame = new JFrame("Add products");

frame.setSize(250, 360);

frame.setLayout(new BorderLayout());

JPanel panel = new JPanel();

frame.add(panel);

panel.setLayout(new GridLayout(6, 1));

JButton btn1 = new JButton("ADD CAMERAS");

JButton btn2 = new JButton("ADD TELEVISIONS");

JButton btn3 = new JButton("ADD MOBILES");

JButton btn4 = new JButton("ADD PCS");

JButton btn5 = new JButton("ADD GAMING CONSOLES");

JButton btn6 = new JButton("BACK");

panel.add(btn1);

panel.add(btn2);

panel.add(btn3);

panel.add(btn4);

panel.add(btn5);

panel.add(btn6);

//actionListeners for the above buttons

btn1.addActionListener(new AddCamera());

btn2.addActionListener(new AddTV());

btn3.addActionListener(new AddMobile());

btn4.addActionListener(new AddPC());//working here

btn5.addActionListener(new AddGamingConsole());

btn6.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ae) {

frame.dispose();

}

});

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.setResizable(false);

}

}

**AddTV.java**

package stavroulakis\_vasilis\_thesis\_2018;

import java.awt.BorderLayout;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileWriter;

import java.io.IOException;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JTextField;

public class AddTV implements ActionListener {

@Override

public void actionPerformed(ActionEvent ae) {

JFrame frame = new JFrame("Add TV");

frame.setSize(250, 300);

frame.setLayout(new BorderLayout());

JPanel panel = new JPanel();

frame.add(panel);

panel.setLayout(new GridLayout(15, 1));

JLabel label1 = new JLabel("TV model");

JLabel label2 = new JLabel("TV type");

JLabel label3 = new JLabel("TV product year");

JLabel label4 = new JLabel("TV monitor size");

JLabel label5 = new JLabel("TV price");

JLabel label6 = new JLabel("Store name");

JLabel label7 = new JLabel("TV quantity");

JTextField field1 = new JTextField(25);

JTextField field2 = new JTextField(25);

JTextField field3 = new JTextField(25);

JTextField field4 = new JTextField(25);

JTextField field5 = new JTextField(25);

JTextField field6 = new JTextField(25);

JTextField field7 = new JTextField(25);

JButton btn = new JButton("ADD");

panel.add(label1);

panel.add(field1);

panel.add(label2);

panel.add(field2);

panel.add(label3);

panel.add(field3);

panel.add(label4);

panel.add(field4);

panel.add(label5);

panel.add(field5);

panel.add(label6);

panel.add(field6);

panel.add(label7);

panel.add(field7);

panel.add(btn);

btn.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ae) {

String text1 = field1.getText();

String text2 = field2.getText();

String text3 = field3.getText();

String text4 = field4.getText();

String text5 = field5.getText();

String text6 = field6.getText();

String text7 = field7.getText();

//Add by using quantity given

int quantity = Integer.parseInt(text7);

for (int i = 0; i < quantity; i++) {

TVClass new\_tv = new TVClass(text1, text2, text3, text4, text5, text6);

writeTV(new\_tv);

Log.i("YOU ADDED A TV ITEM");

}

frame.dispose();

}

});

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.setResizable(false);

}

public void writeTV(TVClass c) {

try {

String current = new java.io.File(".").getCanonicalPath();

File file = new File(current + "\\tvs.txt");

if (!file.exists()) {

file.createNewFile();

}

FileWriter fw = new FileWriter(file, true);

BufferedWriter bw = new BufferedWriter(fw);

bw.write("\n" + c.toString());

bw.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

**AdminMainPanel.java**

package stavroulakis\_vasilis\_thesis\_2018;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class AdminMainPanel {

public static void adminMainPanelMethod() {

JFrame frame = new JFrame("Admin Panel");

frame.setSize(360, 400);

frame.setLayout(new BorderLayout());

JPanel panel = new JPanel();

frame.add(panel);

panel.setLayout(new GridLayout(10, 1));

JButton btn1 = new JButton("PRINT PRODUCTS");

JButton btn2 = new JButton("ADD PRODUCTS");

JButton btn3 = new JButton("DELETE PRODUCTS");

JButton btn4 = new JButton("PRINT CUSTOMERS");

JButton btn5 = new JButton("ADD COMPANY");

JButton btn6 = new JButton("REMOVE COMPANY");

JButton btn7 = new JButton("PRINT COMPANIES");

JButton btn8 = new JButton("PRINT ORDERS");

JButton btn9 = new JButton("FINISH AN ORDER");

JButton btn10 = new JButton("LOGOUT");

panel.add(btn1);

panel.add(btn2);

panel.add(btn3);

panel.add(btn4);

panel.add(btn5);

panel.add(btn6);

panel.add(btn7);

panel.add(btn8);

panel.add(btn9);

panel.add(btn10);

//actionListeners for buttons

btn1.addActionListener(new PrintProducts());

btn2.addActionListener(new AddProducts());

btn3.addActionListener(new DeleteProducts());

btn4.addActionListener(new PrintCustomers());

btn5.addActionListener(new AddCompany());

btn6.addActionListener(new RemoveCompany());

btn7.addActionListener(new PrintCompanies());

btn8.addActionListener(new PrintFinishedOrders());

btn9.addActionListener(new FinishCustomerOrder());

btn10.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ae) {

frame.dispose();

}

});

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.setResizable(false);

}

}

**CameraClass.java**

package stavroulakis\_vasilis\_thesis\_2018;

public class CameraClass {

String camera\_model;

String monitor\_size;

String megapixel\_size;

String frames\_per\_second;

String recording\_type;

String camera\_price;

String store\_name;

String username;

public CameraClass(String camera\_model, String monitor\_size, String megapixel\_size, String frames\_per\_second, String recording\_type, String camera\_price, String store\_name) {

this.camera\_model = camera\_model;

this.monitor\_size = monitor\_size;

this.megapixel\_size = megapixel\_size;

this.frames\_per\_second = frames\_per\_second;

this.recording\_type = recording\_type;

this.camera\_price = camera\_price;

this.store\_name = store\_name;

}

public CameraClass(String username, String camera\_model, String monitor\_size, String megapixel\_size, String frames\_per\_second, String recording\_type, String camera\_price, String store\_name) {

this.username= username;

this.camera\_model = camera\_model;

this.monitor\_size = monitor\_size;

this.megapixel\_size = megapixel\_size;

this.frames\_per\_second = frames\_per\_second;

this.recording\_type = recording\_type;

this.camera\_price = camera\_price;

this.store\_name = store\_name;

this.username = username;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getStore\_name() {

return store\_name;

}

public void setStore\_name(String store\_name) {

this.store\_name = store\_name;

}

@Override

public String toString() {

return this.getCamera\_model() + "\t" + this.getMonitor\_size() + "\t" + this.getMegapixel\_size() + "\t" + this.getFrames\_per\_second() + "\t" + this.getRecording\_type() + "\t" + this.getCamera\_price() + "\t" + this.getStore\_name() + "\n";

}

public String getCamera\_model() {

return camera\_model;

}

public void setCamera\_model(String camera\_model) {

this.camera\_model = camera\_model;

}

public String getMonitor\_size() {

return monitor\_size;

}

public void setMonitor\_size(String monitor\_size) {

this.monitor\_size = monitor\_size;

}

public String getMegapixel\_size() {

return megapixel\_size;

}

public void setMegapixel\_size(String megapixel\_size) {

this.megapixel\_size = megapixel\_size;

}

public String getFrames\_per\_second() {

return frames\_per\_second;

}

public void setFrames\_per\_second(String frames\_per\_second) {

this.frames\_per\_second = frames\_per\_second;

}

public String getRecording\_type() {

return recording\_type;

}

public void setRecording\_type(String recording\_type) {

this.recording\_type = recording\_type;

}

public String getCamera\_price() {

return camera\_price;

}

public void setCamera\_price(String camera\_price) {

this.camera\_price = camera\_price;

}

}

**CompanyClass.java**

package stavroulakis\_vasilis\_thesis\_2018;

public class CompanyClass {

String company\_name;

String company\_owner;

String company\_ITN;

String company\_location;

public CompanyClass(String company\_name, String company\_owner, String company\_ITN, String company\_location) {

this.company\_name = company\_name;

this.company\_owner = company\_owner;

this.company\_ITN = company\_ITN;

this.company\_location = company\_location;

}

@Override

public String toString() {

return this.getCompany\_name()+"\t"+this.getCompany\_owner()+"\t"+this.getCompany\_ITN()+"\t"+this.getCompany\_location()+"\n";

}

public String getCompany\_name() {

return company\_name;

}

public void setCompany\_name(String company\_name) {

this.company\_name = company\_name;

}

public String getCompany\_owner() {

return company\_owner;

}

public void setCompany\_owner(String company\_owner) {

this.company\_owner = company\_owner;

}

public String getCompany\_ITN() {

return company\_ITN;

}

public void setCompany\_ITN(String company\_ITN) {

this.company\_ITN = company\_ITN;

}

public String getCompany\_location() {

return company\_location;

}

public void setCompany\_location(String company\_location) {

this.company\_location = company\_location;

}

}

**CustomerMainPanel.java**

package stavroulakis\_vasilis\_thesis\_2018;

import java.awt.BorderLayout;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JPanel;

public class CustomerMainPanel {

public static void customerMainPanelMethod(String customer\_username) {

JFrame frame = new JFrame("Customer Panel");

frame.setSize(250, 280);

frame.setLayout(new BorderLayout());

JPanel panel = new JPanel();

frame.add(panel);

panel.setLayout(new GridLayout(3, 1));

JButton btn1 = new JButton("PRINT PRODUCTS");

JButton btn2 = new JButton("SHOPPING BASKET");

JButton btn3 = new JButton("LOGOUT");

panel.add(btn1);

panel.add(btn2);

panel.add(btn3);

//actionListeners for buttons also we know customer's username which is customer\_username(String)

btn1.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ae) {

JFrame frameForPrint = new JFrame("Print Products");

frameForPrint.setSize(260, 270);

frameForPrint.setLayout(new BorderLayout());

JPanel panelForPrint = new JPanel();

frameForPrint.add(panelForPrint);

panelForPrint.setLayout(new GridLayout(3, 1));

JButton btn1 = new JButton("PRINT ALL PRODUCTS");

JButton btn2 = new JButton("PRINT BY CATEGORY");

JButton btn3 = new JButton("BACK");

panelForPrint.add(btn1);

panelForPrint.add(btn2);

panelForPrint.add(btn3);

//actionListeners for buttons

btn1.addActionListener(new PrintProducts());

btn2.addActionListener(new PrintProductsByCategory());

btn3.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ae) {

frameForPrint.dispose();

}

});

frameForPrint.setVisible(true);

frameForPrint.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frameForPrint.setResizable(false);

}

});

btn2.addActionListener(new ShoppingBasket(customer\_username) {

});

btn3.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ae) {

frame.dispose();

}

});

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.setResizable(false);

}

}

**CustomerOrderClass.java**

package stavroulakis\_vasilis\_thesis\_2018;

public class CustomerOrderClass {

String username;

String surname;

String address;

String city;

int order\_id;

String time\_zone;

int timeFinished;

double order\_cost;

public CustomerOrderClass(String username, String surname, String address, String city, int order\_id, String time\_zone, int timeFinished, double order\_cost) {

this.username = username;

this.surname = surname;

this.address = address;

this.city = city;

this.order\_id = order\_id;

this.time\_zone = time\_zone;

this.timeFinished = timeFinished;

this.order\_cost = order\_cost;

}

public String toString() {

return this.getUsername() + "\t" + this.getSurname() + "\t" + this.getAddress() + "\t" + this.getCity() + "\t" + this.getOrder\_id() + "\t" + this.getTime\_zone() + "\t" + this.getTimeFinished() + "\t" + this.getOrder\_cost() + "\n";

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getSurname() {

return surname;

}

public void setSurname(String surname) {

this.surname = surname;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

public String getCity() {

return city;

}

public void setCity(String city) {

this.city = city;

}

public int getOrder\_id() {

return order\_id;

}

public void setOrder\_id(int order\_id) {

this.order\_id = order\_id;

}

public String getTime\_zone() {

return time\_zone;

}

public void setTime\_zone(String time\_zone) {

this.time\_zone = time\_zone;

}

public int getTimeFinished() {

return timeFinished;

}

public void setTimeFinished(int timeFinished) {

this.timeFinished = timeFinished;

}

public double getOrder\_cost() {

return order\_cost;

}

public void setOrder\_cost(double order\_cost) {

this.order\_cost = order\_cost;

}

}

**DeleteProducts.java**

package stavroulakis\_vasilis\_thesis\_2018;

import java.awt.BorderLayout;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JPanel;

public class DeleteProducts implements ActionListener {

@Override

public void actionPerformed(ActionEvent ae) {

JFrame frame = new JFrame("Remove product");

frame.setSize(280, 340);

frame.setLayout(new BorderLayout());

JPanel panel = new JPanel();

frame.add(panel);

panel.setLayout(new GridLayout(6, 1));

JButton btn1 = new JButton("REMOVE CAMERAS");

JButton btn2 = new JButton("REMOVE TELEVISIONS");

JButton btn3 = new JButton("REMOVE MOBILES");

JButton btn4 = new JButton("REMOVE PERSONAL COMPUTERS"); //

JButton btn5 = new JButton("REMOVE GAMING CONSOLES");

JButton btn6 = new JButton("BACK");

panel.add(btn1);

panel.add(btn2);

panel.add(btn3);

panel.add(btn4);

panel.add(btn5);

panel.add(btn6);

//actionListeners for buttons

btn1.addActionListener(new RemoveCamera());

btn2.addActionListener(new RemoveTV());

btn3.addActionListener(new RemoveMobiles());

btn4.addActionListener(new RemovePC());

btn5.addActionListener(new RemoveGamingConsoles());

btn6.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ae) {

frame.dispose();

}

});

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.setResizable(false);

}

}

**FinishCustomerOrder.java**

package stavroulakis\_vasilis\_thesis\_2018;

import java.awt.BorderLayout;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.FileWriter;

import java.io.IOException;

import java.io.PrintWriter;

import java.util.ArrayList;

import java.util.Iterator;

import java.util.List;

import java.util.Random;

import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JTextField;

public class FinishCustomerOrder implements ActionListener {

@Override

public void actionPerformed(ActionEvent ae) {

JFrame frame = new JFrame("Finish");

frame.setSize(200, 220);

frame.setLayout(new BorderLayout());

JPanel panel = new JPanel();

frame.add(panel);

panel.setLayout(new GridLayout(3, 1));

JLabel label1 = new JLabel("Customer's username");

JTextField field1 = new JTextField(20);

JButton btn1 = new JButton("CONFIRM");

panel.add(label1);

panel.add(field1);

panel.add(btn1);

btn1.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ae) {

String get\_customer\_username = field1.getText();

double final\_order\_cost = 0;

//checking variable(just for testing the code)

try {

final\_order\_cost = reconstructTemporaryAndNoTemporaryFilesArrayListsPlusCalculateOrderCost(get\_customer\_username);

} catch (FileNotFoundException ex) {

Logger.getLogger(FinishCustomerOrder.class.getName()).log(Level.SEVERE, null, ex);

}

//TRYING TO REMOVE EVERYTHING FROM THE SHOPPING BASKET

File f1 = null;

File f2 = null;

File f3 = null;

File f4 = null;

File f5 = null;

try {

f1 = new File("cameras\_temp\_basket.txt");

f2 = new File("consoles\_temp\_basket.txt");

f3 = new File("mobiles\_temp\_file.txt");

f4 = new File("pcs\_temp\_file.txt");

f5 = new File("tvs\_temp\_file.txt");

//delete shopping basket's files

f1.delete();

f2.delete();

f3.delete();

f4.delete();

f5.delete();

Log.i("THE SHOPPING BASKET DELETED! REPLACED BY A NEW ONE!");

//create a new empty shopping basket(files)

f1.createNewFile();

f2.createNewFile();

f3.createNewFile();

f4.createNewFile();

f5.createNewFile();

} catch (Exception e) {

e.printStackTrace();

Log.e("WE CANNOT DELETE THE SHOPPING BASKET!");

}

//TRYING TO FINISH AN ORDER

createAnOrder(get\_customer\_username, final\_order\_cost);

}

}

);

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.setResizable(false);

}

//FINISH AN ORDER

public void createAnOrder(String get\_customer\_username, double final\_order\_cost) {

Scanner file = null;

Scanner file1 = null;

ArrayList<RegisterClass> registered\_customers = new ArrayList<>();

try {

file = new Scanner(new File("customers.txt"));

} catch (FileNotFoundException ex) {

Logger.getLogger(FinishCustomerOrder.class.getName()).log(Level.SEVERE, null, ex);

}

//recreate customer's file

while (file.hasNext()) {

String line = file.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String fname = stringArr[0];

String surname = stringArr[1];

String address = stringArr[2];

String city = stringArr[3];

String TK = stringArr[4];

String AFM = stringArr[5];

String username = stringArr[6];

String password = stringArr[7];

RegisterClass customer = new RegisterClass(fname, surname, address, city, TK, AFM, username, password);

registered\_customers.add(customer);

}

}

file.close();

//define some variables

String customer\_surname = null;

String customer\_address = null;

String customer\_city = null;

int order\_id = 0;

String time\_zone = null;

int timeFinished = 0;

//get customer's values

for (RegisterClass customer : registered\_customers) {

if (customer.getUsername().contains(get\_customer\_username)) {

customer\_surname = customer.getSurname();

customer\_address = customer.getAddress();

customer\_city = customer.getCity();

}

}

//generate some usefull values for the customer

Random rand = new Random();

order\_id = rand.nextInt(340) + 1;

timeFinished = rand.nextInt(45) + 1; //days

time\_zone = getTimeZone(time\_zone); //call function for getting a fake timezone

//pass all values to function and then save to file

try {

file1 = new Scanner(new File("customer\_orders.txt"));

} catch (FileNotFoundException ex) {

Logger.getLogger(FinishCustomerOrder.class.getName()).log(Level.SEVERE, null, ex);

}

//create an object and call a save to file function

CustomerOrderClass customer = new CustomerOrderClass(get\_customer\_username, customer\_surname, customer\_address, customer\_city, order\_id, time\_zone, timeFinished, final\_order\_cost);

writeCustomerOrder(customer);

file1.close();

}

//FAKE TIMEZONE FUNCTION

public String getTimeZone(String time\_zone) {

String t1 = "UTC-12:00", t2 = "UTC-11:00", t3 = "UTC-09:30", t4 = "UTC-08:00", t5 = "UTC-07:00", t6 = "UTC-06:00";

String t7 = "UTC-05:00", t8 = "UTC-04:00", t9 = "UTC-03:30", t10 = "UTC-02:00", t11 = "UTC-01:00", t12 = "UTC-00:00";

String t13 = "UTC+01:00", t14 = "UTC+02:00", t15 = "UTC+03:30", t16 = "UTC+03:30", t17 = "UTC+04:00", t18 = "UTC+04:30";

String t19 = "UTC+05:00", t20 = "UTC+05:30", t21 = "UTC+05:45", t22 = "UTC+06:00", t23 = "UTC+06:30", t24 = "UTC+07:00";

String t25 = "UTC+08:00", t26 = "UTC+08:30", t27 = "UTC+08:45", t28 = "UTC+09:00", t29 = "UTC+09:30", t30 = "UTC+10:00";

String t31 = "UTC+10:30", t32 = "UTC+11:00", t33 = "UTC+12:00", t34 = "UTC+12:45", t35 = "UTC+13:00", t36 = "UTC+14:00";

int select\_time\_zone = 0;

Random rand = new Random();

select\_time\_zone = rand.nextInt(36) + 1;

if (select\_time\_zone == 1) {

return t1;

}

if (select\_time\_zone == 2) {

return t2;

}

if (select\_time\_zone == 3) {

return t3;

}

if (select\_time\_zone == 4) {

return t4;

}

if (select\_time\_zone == 5) {

return t5;

}

if (select\_time\_zone == 6) {

return t6;

}

if (select\_time\_zone == 7) {

return t7;

}

if (select\_time\_zone == 8) {

return t8;

}

if (select\_time\_zone == 9) {

return t9;

}

if (select\_time\_zone == 10) {

return t10;

}

if (select\_time\_zone == 11) {

return t11;

}

if (select\_time\_zone == 12) {

return t12;

}

if (select\_time\_zone == 13) {

return t13;

}

if (select\_time\_zone == 14) {

return t14;

}

if (select\_time\_zone == 15) {

return t15;

}

if (select\_time\_zone == 16) {

return t16;

}

if (select\_time\_zone == 17) {

return t17;

}

if (select\_time\_zone == 18) {

return t18;

}

if (select\_time\_zone == 19) {

return t19;

}

if (select\_time\_zone == 20) {

return t20;

}

if (select\_time\_zone == 21) {

return t21;

}

if (select\_time\_zone == 22) {

return t22;

}

if (select\_time\_zone == 23) {

return t23;

}

if (select\_time\_zone == 24) {

return t24;

}

if (select\_time\_zone == 25) {

return t25;

}

if (select\_time\_zone == 26) {

return t26;

}

if (select\_time\_zone == 27) {

return t27;

}

if (select\_time\_zone == 28) {

return t28;

}

if (select\_time\_zone == 29) {

return t29;

}

if (select\_time\_zone == 30) {

return t30;

}

if (select\_time\_zone == 31) {

return t31;

}

if (select\_time\_zone == 32) {

return t32;

}

if (select\_time\_zone == 33) {

return t33;

}

if (select\_time\_zone == 34) {

return t34;

}

if (select\_time\_zone == 35) {

return t35;

}

if (select\_time\_zone == 36) {

return t36;

}

return null;

}

//WRITE CUSTOMER ORDER

public void writeCustomerOrder(CustomerOrderClass c) {

try {

String current = new java.io.File(".").getCanonicalPath();

File file = new File(current + "\\customer\_orders.txt");

if (!file.exists()) {

file.createNewFile();

}

FileWriter fw = new FileWriter(file, true);

BufferedWriter bw = new BufferedWriter(fw);

bw.write("\n" + c.toString());

bw.close();

} catch (IOException e) {

e.printStackTrace();

}

}

//DELETE FROM NON - TEMPORARY FILES

public double reconstructTemporaryAndNoTemporaryFilesArrayListsPlusCalculateOrderCost(String customer\_username) throws FileNotFoundException {

//construct the non- temporary files into arrayLists

List<CameraClass> cameras\_temp\_list = new ArrayList<>();

List<GamingConsoleClass> consoles\_temp\_list = new ArrayList<>();

List<MobileClass> mobiles\_temp\_list = new ArrayList<>();

List<PersonalComputerClass> pcs\_temp\_list = new ArrayList<>();

List<TVClass> tvs\_temp\_list = new ArrayList<>();

Scanner file1 = null;

Scanner file2 = null;

Scanner file3 = null;

Scanner file4 = null;

Scanner file5 = null;

try {

file1 = new Scanner(new File("cameras\_temp\_basket.txt"));

file2 = new Scanner(new File("consoles\_temp\_basket.txt"));

file3 = new Scanner(new File("mobiles\_temp\_file.txt"));

file4 = new Scanner(new File("pcs\_temp\_file.txt"));

file5 = new Scanner(new File("tvs\_temp\_file.txt"));

} catch (FileNotFoundException ex) {

Logger.getLogger(FinishCustomerOrder.class.getName()).log(Level.SEVERE, null, ex);

}

while (file1.hasNext()) {

String line = file1.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String username = stringArr[0];

String cameraModel = stringArr[1];

String cameraMonitorSize = stringArr[2];

String cameraMP = stringArr[3];

String cameraFramesPerSecond = stringArr[4];

String cameraRecordingType = stringArr[5];

String cameraPrice = stringArr[6];

String cameraStoreName = stringArr[7];

CameraClass camera = new CameraClass(username, cameraModel, cameraMonitorSize, cameraMP, cameraFramesPerSecond, cameraRecordingType, cameraPrice, cameraStoreName);

cameras\_temp\_list.add(camera);

}

}

while (file2.hasNext()) {

String line = file2.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String username = stringArr[0];

String consoleModel = stringArr[1];

String consoleSize = stringArr[2];

String consoleYearReleased = stringArr[3];

String consoleControlllers = stringArr[4];

String consolePrice = stringArr[5];

String consoleStoreName = stringArr[6];

GamingConsoleClass console = new GamingConsoleClass(username, consoleModel, consoleSize, consoleYearReleased, consoleControlllers, consolePrice, consoleStoreName);

consoles\_temp\_list.add(console);

}

}

while (file3.hasNext()) {

String line = file3.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String username = stringArr[0];

String mobileModel = stringArr[1];

String mobileFrontCamera = stringArr[2];

String mobileBackCamera = stringArr[3];

String mobileScreenSize = stringArr[4];

String mobileCPU = stringArr[5];

String mobileStorage = stringArr[6];

String mobilePrice = stringArr[7];

String mobileStoreName = stringArr[8];

MobileClass mobile = new MobileClass(username, mobileModel, mobileFrontCamera, mobileBackCamera, mobileScreenSize, mobileCPU, mobileStorage, mobilePrice, mobileStoreName);

mobiles\_temp\_list.add(mobile);

}

}

while (file4.hasNext()) {

String line = file4.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String username = stringArr[0];

String pcModelName = stringArr[1];

String pcCpuName = stringArr[2];

String pcClock = stringArr[3];

String pcGPUname = stringArr[4];

String pcHardDrive = stringArr[5];

String pcRamSize = stringArr[6];

String pcPrice = stringArr[7];

String pcStoreName = stringArr[8];

PersonalComputerClass pc = new PersonalComputerClass(username, pcModelName, pcCpuName, pcClock, pcGPUname, pcHardDrive, pcRamSize, pcPrice, pcStoreName);

pcs\_temp\_list.add(pc);

}

}

while (file5.hasNext()) {

String line = file5.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String username = stringArr[0];

String tvModelName = stringArr[1];

String tvType = stringArr[2];

String tvYear = stringArr[3];

String tvMonitorSize = stringArr[4];

String tvPrice = stringArr[5];

String tvStoreName = stringArr[6];

TVClass tv = new TVClass(username, tvModelName, tvType, tvYear, tvMonitorSize, tvPrice, tvStoreName);

tvs\_temp\_list.add(tv);

}

}

//calculate customer's order cost

double final\_order\_cost = 0.0;

final\_order\_cost = calculateOrderCost(cameras\_temp\_list, consoles\_temp\_list, mobiles\_temp\_list, pcs\_temp\_list, tvs\_temp\_list, customer\_username);

//reconstruct temporaryFiles

reconstructBasicProductFiles(cameras\_temp\_list, consoles\_temp\_list, mobiles\_temp\_list, pcs\_temp\_list, tvs\_temp\_list, customer\_username);

file1.close();

file2.close();

file3.close();

file4.close();

file5.close();

return final\_order\_cost;

}

public void reconstructBasicProductFiles(List cameras\_temp\_list, List consoles\_temp\_list, List mobiles\_temp\_list, List pcs\_temp\_list, List tvs\_temp\_list, String customer\_username) throws FileNotFoundException {

//construct the temporary files into arrayLists

List<CameraClass> cameras\_list = new ArrayList<>();

List<GamingConsoleClass> consoles\_list = new ArrayList<>();

List<MobileClass> mobiles\_list = new ArrayList<>();

List<PersonalComputerClass> personalComputer\_list = new ArrayList<>();

List<TVClass> TV\_list = new ArrayList<>();

Scanner file11 = null;

Scanner file22 = null;

Scanner file33 = null;

Scanner file44 = null;

Scanner file55 = null;

try {

file11 = new Scanner(new File("cameras.txt"));

file22 = new Scanner(new File("gaming\_consoles.txt"));

file33 = new Scanner(new File("mobiles.txt"));

file44 = new Scanner(new File("personalComputers.txt"));

file55 = new Scanner(new File("tvs.txt"));

} catch (FileNotFoundException ex) {

Logger.getLogger(FinishCustomerOrder.class.getName()).log(Level.SEVERE, null, ex);

}

while (file11.hasNext()) {

String line = file11.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String cameraModel = stringArr[0];

String monitorSize = stringArr[1];

String megaPixelSize = stringArr[2];

String framesPerSecond = stringArr[3];

String recordingType = stringArr[4];

String cameraPrice = stringArr[5];

String storeName = stringArr[6];

CameraClass camera = new CameraClass(cameraModel, monitorSize, megaPixelSize, framesPerSecond, recordingType, cameraPrice, storeName);

cameras\_list.add(camera);

}

}

while (file22.hasNext()) {

String line = file22.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String consoleModel = stringArr[0];

String consoleSize = stringArr[1];

String consoleYearReleased = stringArr[2];

String consoleControlllers = stringArr[3];

String consolePrice = stringArr[4];

String consoleStoreName = stringArr[5];

GamingConsoleClass console = new GamingConsoleClass(consoleModel, consoleSize, consoleYearReleased, consoleControlllers, consolePrice, consoleStoreName);

consoles\_list.add(console);

}

}

while (file33.hasNext()) {

String line = file33.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String mobileModel = stringArr[0];

String mobileFrontCamera = stringArr[1];

String mobileBackCamera = stringArr[2];

String mobileScreenSize = stringArr[3];

String mobileCPU = stringArr[4];

String mobileStorage = stringArr[5];

String mobilePrice = stringArr[6];

String mobileStoreName = stringArr[7];

MobileClass mobile = new MobileClass(mobileModel, mobileFrontCamera, mobileBackCamera, mobileScreenSize, mobileCPU, mobileStorage, mobilePrice, mobileStoreName);

mobiles\_list.add(mobile);

}

}

while (file44.hasNext()) {

String line = file44.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String pcModelName = stringArr[0];

String pcCpuName = stringArr[1];

String pcClock = stringArr[2];

String pcGPUname = stringArr[3];

String pcHardDrive = stringArr[4];

String pcRamSize = stringArr[5];

String pcPrice = stringArr[6];

String pcStoreName = stringArr[7];

PersonalComputerClass pc = new PersonalComputerClass(pcModelName, pcCpuName, pcClock, pcGPUname, pcHardDrive, pcRamSize, pcPrice, pcStoreName);

personalComputer\_list.add(pc);

}

}

while (file55.hasNext()) {

String line = file55.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String tvModelName = stringArr[0];

String tvType = stringArr[1];

String tvYear = stringArr[2];

String tvMonitorSize = stringArr[3];

String tvPrice = stringArr[4];

String tvStoreName = stringArr[5];

TVClass tv = new TVClass(tvModelName, tvType, tvYear, tvMonitorSize, tvPrice, tvStoreName);

TV\_list.add(tv);

}

}

file11.close();

file22.close();

file33.close();

file44.close();

file55.close();

//check products by username & delete chosen products from temporary and non temporary file

checkByUsernamePlusDeleteProductsFromTemporary(cameras\_temp\_list, consoles\_temp\_list, mobiles\_temp\_list, pcs\_temp\_list, tvs\_temp\_list, cameras\_list, consoles\_list, mobiles\_list, personalComputer\_list, TV\_list, customer\_username);

}

public double calculateOrderCost(List<CameraClass> cameras\_temp\_list, List<GamingConsoleClass> consoles\_temp\_list, List<MobileClass> mobiles\_temp\_list, List<PersonalComputerClass> pcs\_temp\_list, List<TVClass> tvs\_temp\_list, String customer\_username) {

double customer\_cameras\_order\_cost = 0;

double customer\_consoles\_order\_cost = 0;

double customer\_mobiles\_order\_cost = 0;

double customer\_pcs\_order\_cost = 0;

double customer\_tvs\_order\_cost = 0;

double order\_final\_cost = 0;

//calculate cameras cost

String camera\_price = null;

for (CameraClass camera : cameras\_temp\_list) {

if (camera.getUsername().contains(customer\_username)) {

camera\_price = camera.getCamera\_price();

double camera\_price\_float = Float.parseFloat(camera\_price);

customer\_cameras\_order\_cost = customer\_cameras\_order\_cost + camera\_price\_float;

}

}

//calculate consoles cost

for (GamingConsoleClass console : consoles\_temp\_list) {

if (console.getUsername().contains(customer\_username)) {

String console\_price = console.getConsole\_price();

float console\_price\_float = Float.parseFloat(console\_price);

customer\_consoles\_order\_cost = customer\_consoles\_order\_cost + console\_price\_float;

}

}

//calculate mobiles cost

for (MobileClass mobile : mobiles\_temp\_list) {

if (mobile.getUsername().contains(customer\_username)) {

String mobile\_price = mobile.getMobile\_price();

float mobile\_price\_float = Float.parseFloat(mobile\_price);

customer\_mobiles\_order\_cost = customer\_mobiles\_order\_cost + mobile\_price\_float;

}

}

//calculate pcs cost

for (PersonalComputerClass pc : pcs\_temp\_list) {

if (pc.getUsername().contains(customer\_username)) {

String pc\_price = pc.getPc\_price();

float pc\_price\_float = Float.parseFloat(pc\_price);

customer\_pcs\_order\_cost = customer\_pcs\_order\_cost + pc\_price\_float;

}

}

//calculate tvs cost

for (TVClass tv : tvs\_temp\_list) {

if (tv.getUsername().contains(customer\_username)) {

String tv\_price = tv.getTv\_price();

float tv\_price\_float = Float.parseFloat(tv\_price);

customer\_tvs\_order\_cost = customer\_tvs\_order\_cost + tv\_price\_float;

}

}

order\_final\_cost = customer\_cameras\_order\_cost + customer\_consoles\_order\_cost + customer\_mobiles\_order\_cost + customer\_pcs\_order\_cost + customer\_tvs\_order\_cost;

return order\_final\_cost;

}

public void checkByUsernamePlusDeleteProductsFromTemporary(List<CameraClass> cameras\_temp\_list, List<GamingConsoleClass> consoles\_temp\_list, List<MobileClass> mobiles\_temp\_list, List<PersonalComputerClass> pcs\_temp\_list, List<TVClass> tvs\_temp\_list, List<CameraClass> cameras\_list, List<GamingConsoleClass> consoles\_list, List<MobileClass> mobiles\_list, List<PersonalComputerClass> personalComputer\_list, List<TVClass> TV\_list, String customer\_username) throws FileNotFoundException {

String camera\_model\_name = null;

String console\_model\_name = null;

String mobile\_model\_name = null;

String pc\_model\_name = null;

String tv\_model\_name = null;

//search in temp files the appropriate model names and then call functions in order

//to delete from non-temporary files and then save

for (CameraClass camera : cameras\_temp\_list) {

if (camera.getUsername().contains(customer\_username)) {

camera\_model\_name = camera.getCamera\_model();

deleteCamerasFromNonTemporary(camera\_model\_name, (ArrayList) cameras\_list);

}

}

for (GamingConsoleClass console : consoles\_temp\_list) {

if (console.getUsername().contains(customer\_username)) {

console\_model\_name = console.getConsole\_name();

deleteConsolesFromNonTemporary(console\_model\_name, (ArrayList) consoles\_list);

}

}

for (MobileClass mobile : mobiles\_temp\_list) {

if (mobile.getUsername().contains(customer\_username)) {

mobile\_model\_name = mobile.getMobile\_model();

deleteMobilesFromNonTemporary(mobile\_model\_name, (ArrayList) mobiles\_list);

}

}

for (PersonalComputerClass pc : pcs\_temp\_list) {

if (pc.getUsername().contains(customer\_username)) {

pc\_model\_name = pc.getModel\_name();

deletePCSFromNonTemporary(pc\_model\_name, (ArrayList) personalComputer\_list);

}

}

for (TVClass tv : tvs\_temp\_list) {

if (tv.getUsername().contains(customer\_username)) {

tv\_model\_name = tv.getTv\_model();

deleteTVSFromNonTemporary(tv\_model\_name, (ArrayList) TV\_list);

}

}

}

public void deleteCamerasFromNonTemporary(String camera\_model\_name, ArrayList cameras\_list) throws FileNotFoundException {

Iterator<CameraClass> it = cameras\_list.iterator();

while (it.hasNext()) {

if (it.next().getCamera\_model().equals(camera\_model\_name)) {

it.remove();

break;

}

}

saveCamerasListToFile(cameras\_list, "cameras.txt");

}

public void deleteConsolesFromNonTemporary(String console\_model\_name, ArrayList consoles\_list) throws FileNotFoundException {

Iterator<GamingConsoleClass> it = consoles\_list.iterator();

while (it.hasNext()) {

if (it.next().getConsole\_name().equals(console\_model\_name)) {

it.remove();

break;

}

}

saveConsolesListToFile(consoles\_list, "gaming\_consoles.txt");

}

public void deleteMobilesFromNonTemporary(String mobile\_model\_name, ArrayList mobiles\_list) throws FileNotFoundException {

Iterator<MobileClass> it = mobiles\_list.iterator();

while (it.hasNext()) {

if (it.next().getMobile\_model().equals(mobile\_model\_name)) {

it.remove();

break;

}

}

saveMobilesListToFile(mobiles\_list, "mobiles.txt");

}

public void deletePCSFromNonTemporary(String pc\_model\_name, ArrayList personalComputer\_list) throws FileNotFoundException {

Iterator<PersonalComputerClass> it = personalComputer\_list.iterator();

while (it.hasNext()) {

if (it.next().getModel\_name().equals(pc\_model\_name)) {

it.remove();

break;

}

}

savePCSListToFile(personalComputer\_list, "personalComputers.txt");

}

public void deleteTVSFromNonTemporary(String tv\_model\_name, ArrayList TV\_list) throws FileNotFoundException {

Iterator<TVClass> it = TV\_list.iterator();

while (it.hasNext()) {

if (it.next().getTv\_model().equals(tv\_model\_name)) {

it.remove();

break;

}

}

saveTVSListToFile(TV\_list, "tvs.txt");

}

public void saveCamerasListToFile(ArrayList cameras\_list, String fileName) throws FileNotFoundException {

PrintWriter pw = new PrintWriter(new FileOutputStream(fileName));

for (Object camera : cameras\_list) {

pw.println(camera.toString());

}

pw.close();

}

//SAVE PRODUCTS FUNCTIONS( WE HAVE REMOVED THE PRODUCTS WERE CUSTOMER HAD BOUGHT)

public void saveConsolesListToFile(ArrayList consoles\_list, String fileName) throws FileNotFoundException {

PrintWriter pw = new PrintWriter(new FileOutputStream(fileName));

for (Object console : consoles\_list) {

pw.println(console.toString());

}

pw.close();

}

public void saveMobilesListToFile(ArrayList mobiles\_list, String fileName) throws FileNotFoundException {

PrintWriter pw = new PrintWriter(new FileOutputStream(fileName));

for (Object mobile : mobiles\_list) {

pw.println(mobile.toString());

}

pw.close();

}

public void savePCSListToFile(ArrayList personalComputer\_list, String fileName) throws FileNotFoundException {

PrintWriter pw = new PrintWriter(new FileOutputStream(fileName));

for (Object pc : personalComputer\_list) {

pw.println(pc.toString());

}

pw.close();

}

public void saveTVSListToFile(ArrayList TV\_list, String fileName) throws FileNotFoundException {

PrintWriter pw = new PrintWriter(new FileOutputStream(fileName));

for (Object tv : TV\_list) {

pw.println(tv.toString());

}

pw.close();

}

}

**GamingConsoleClass.java**

package stavroulakis\_vasilis\_thesis\_2018;

import java.awt.BorderLayout;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JPanel;

public class CustomerMainPanel {

public static void customerMainPanelMethod(String customer\_username) {

JFrame frame = new JFrame("Customer Panel");

frame.setSize(250, 280);

frame.setLayout(new BorderLayout());

JPanel panel = new JPanel();

frame.add(panel);

panel.setLayout(new GridLayout(3, 1));

JButton btn1 = new JButton("PRINT PRODUCTS");

JButton btn2 = new JButton("SHOPPING BASKET");

JButton btn3 = new JButton("LOGOUT");

panel.add(btn1);

panel.add(btn2);

panel.add(btn3);

//actionListeners for buttons also we know customer's username which is customer\_username(String)

btn1.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ae) {

JFrame frameForPrint = new JFrame("Print Products");

frameForPrint.setSize(260, 270);

frameForPrint.setLayout(new BorderLayout());

JPanel panelForPrint = new JPanel();

frameForPrint.add(panelForPrint);

panelForPrint.setLayout(new GridLayout(3, 1));

JButton btn1 = new JButton("PRINT ALL PRODUCTS");

JButton btn2 = new JButton("PRINT BY CATEGORY");

JButton btn3 = new JButton("BACK");

panelForPrint.add(btn1);

panelForPrint.add(btn2);

panelForPrint.add(btn3);

//actionListeners for buttons

btn1.addActionListener(new PrintProducts());

btn2.addActionListener(new PrintProductsByCategory());

btn3.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ae) {

frameForPrint.dispose();

}

});

frameForPrint.setVisible(true);

frameForPrint.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frameForPrint.setResizable(false);

}

});

btn2.addActionListener(new ShoppingBasket(customer\_username) {

});

btn3.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ae) {

frame.dispose();

}

});

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.setResizable(false);

}

}

**Log.java**

//JAVA CONSOLE COLORING - Ajmal Salim/StackOverFlow Date: September 2017

//MODIFIED BY VASILIS STAVROULAKIS

package stavroulakis\_vasilis\_thesis\_2018;

public class Log {

public static final String ANSI\_RESET = "\u001B[0m";

public static final String ANSI\_BLACK = "\u001B[30m";

public static final String ANSI\_RED = "\u001B[31m";

public static final String ANSI\_GREEN = "\u001B[32m";

public static final String ANSI\_YELLOW = "\u001B[33m";

public static final String ANSI\_BLUE = "\u001B[34m";

public static final String ANSI\_PURPLE = "\u001B[35m";

public static final String ANSI\_CYAN = "\u001B[36m";

public static final String ANSI\_WHITE = "\u001B[37m";

//info

public static void i(String message) {

System.out.println(ANSI\_GREEN + " : " + message + ANSI\_RESET);

}

//error

public static void e(String message) {

System.out.println(ANSI\_RED + " : " + message + ANSI\_RESET);

}

//debug

public static void d(String message) {

System.out.println(ANSI\_BLUE + " : " + message + ANSI\_RESET);

}

//warning

public static void w(String message) {

System.out.println(ANSI\_YELLOW + " : " + message + ANSI\_RESET);

}

public static void f(String message) {

System.out.println(ANSI\_CYAN + " : " + message + ANSI\_RESET);

}

}

**MobileClass.java**

package stavroulakis\_vasilis\_thesis\_2018;

public class MobileClass {

String mobile\_model;

String front\_camera\_mpixel;

String back\_camera\_mpixel;

String mobile\_screen\_size;

String mobile\_cpu;

String mobile\_storage;

String mobile\_price;

String store\_name;

String username;

public MobileClass(String mobile\_model, String front\_camera\_mpixel, String back\_camera\_mpixel, String mobile\_screen\_size, String mobile\_cpu, String mobile\_storage, String mobile\_price, String store\_name) {

this.mobile\_model = mobile\_model;

this.front\_camera\_mpixel = front\_camera\_mpixel;

this.back\_camera\_mpixel = back\_camera\_mpixel;

this.mobile\_screen\_size = mobile\_screen\_size;

this.mobile\_cpu = mobile\_cpu;

this.mobile\_storage = mobile\_storage;

this.mobile\_price = mobile\_price;

this.store\_name = store\_name;

}

public MobileClass(String username, String mobile\_model, String front\_camera\_mpixel, String back\_camera\_mpixel, String mobile\_screen\_size, String mobile\_cpu, String mobile\_storage, String mobile\_price, String store\_name) {

this.username = username;

this.mobile\_model = mobile\_model;

this.front\_camera\_mpixel = front\_camera\_mpixel;

this.back\_camera\_mpixel = back\_camera\_mpixel;

this.mobile\_screen\_size = mobile\_screen\_size;

this.mobile\_cpu = mobile\_cpu;

this.mobile\_storage = mobile\_storage;

this.mobile\_price = mobile\_price;

this.store\_name = store\_name;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getMobile\_storage() {

return mobile\_storage;

}

public void setMobile\_storage(String mobile\_storage) {

this.mobile\_storage = mobile\_storage;

}

@Override

public String toString() {

return this.getMobile\_model() + "\t" + this.getFront\_camera\_mpixel() + "\t" + this.getBack\_camera\_mpixel() + "\t" + this.getMobile\_screen\_size() + "\t" + this.getMobile\_cpu() + "\t" + this.getMobile\_storage() + "\t" + this.getMobile\_price() + "\t" + this.getStore\_name() + "\n";

}

public String getMobile\_model() {

return mobile\_model;

}

public void setMobile\_model(String mobile\_model) {

this.mobile\_model = mobile\_model;

}

public String getFront\_camera\_mpixel() {

return front\_camera\_mpixel;

}

public void setFront\_camera\_mpixel(String front\_camera\_mpixel) {

this.front\_camera\_mpixel = front\_camera\_mpixel;

}

public String getBack\_camera\_mpixel() {

return back\_camera\_mpixel;

}

public void setBack\_camera\_mpixel(String back\_camera\_mpixel) {

this.back\_camera\_mpixel = back\_camera\_mpixel;

}

public String getMobile\_screen\_size() {

return mobile\_screen\_size;

}

public void setMobile\_screen\_size(String mobile\_screen\_size) {

this.mobile\_screen\_size = mobile\_screen\_size;

}

public String getMobile\_cpu() {

return mobile\_cpu;

}

public void setMobile\_cpu(String mobile\_cpu) {

this.mobile\_cpu = mobile\_cpu;

}

public String getMobile\_price() {

return mobile\_price;

}

public void setMobile\_price(String mobile\_price) {

this.mobile\_price = mobile\_price;

}

public String getStore\_name() {

return store\_name;

}

public void setStore\_name(String store\_name) {

this.store\_name = store\_name;

}

}

**PersonalComputerClass.java**

package stavroulakis\_vasilis\_thesis\_2018;

public class PersonalComputerClass {

String model\_name;

String pc\_cpu\_name;

String cpu\_clock;

String graphics\_card\_name;

String pc\_hard\_drive;

String pc\_ram\_size;

String pc\_price;

String pc\_store\_name;

String username;

public PersonalComputerClass(String model\_name, String pc\_cpu\_name, String cpu\_clock, String graphics\_card\_name, String pc\_hard\_drive, String pc\_ram\_size, String pc\_price, String pc\_store\_name) {

this.model\_name = model\_name;

this.pc\_cpu\_name = pc\_cpu\_name;

this.cpu\_clock = cpu\_clock;

this.graphics\_card\_name = graphics\_card\_name;

this.pc\_hard\_drive = pc\_hard\_drive;

this.pc\_ram\_size = pc\_ram\_size;

this.pc\_price = pc\_price;

this.pc\_store\_name = pc\_store\_name;

}

public PersonalComputerClass(String username, String model\_name, String pc\_cpu\_name, String cpu\_clock, String graphics\_card\_name, String pc\_hard\_drive, String pc\_ram\_size, String pc\_price, String pc\_store\_name) {

this.username = username;

this.model\_name = model\_name;

this.pc\_cpu\_name = pc\_cpu\_name;

this.cpu\_clock = cpu\_clock;

this.graphics\_card\_name = graphics\_card\_name;

this.pc\_hard\_drive = pc\_hard\_drive;

this.pc\_ram\_size = pc\_ram\_size;

this.pc\_price = pc\_price;

this.pc\_store\_name = pc\_store\_name;

this.username = username;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

@Override

public String toString() {

return this.getModel\_name() + "\t" + this.getPc\_cpu\_name() + "\t" + this.getCpu\_clock() + "\t" + this.getGraphics\_card\_name() + "\t" + this.getPc\_hard\_drive() + "\t" + this.getPc\_ram\_size() + "\t" + this.getPc\_price() + "\t " + this.getPc\_store\_name() + "\n";

}

public String getModel\_name() {

return model\_name;

}

public void setModel\_name(String model\_name) {

this.model\_name = model\_name;

}

public String getPc\_cpu\_name() {

return pc\_cpu\_name;

}

public void setPc\_cpu\_name(String pc\_cpu\_name) {

this.pc\_cpu\_name = pc\_cpu\_name;

}

public String getCpu\_clock() {

return cpu\_clock;

}

public void setCpu\_clock(String cpu\_clock) {

this.cpu\_clock = cpu\_clock;

}

public String getGraphics\_card\_name() {

return graphics\_card\_name;

}

public void setGraphics\_card\_name(String graphics\_card\_name) {

this.graphics\_card\_name = graphics\_card\_name;

}

public String getPc\_hard\_drive() {

return pc\_hard\_drive;

}

public void setPc\_hard\_drive(String pc\_hard\_drive) {

this.pc\_hard\_drive = pc\_hard\_drive;

}

public String getPc\_ram\_size() {

return pc\_ram\_size;

}

public void setPc\_ram\_size(String pc\_ram\_size) {

this.pc\_ram\_size = pc\_ram\_size;

}

public String getPc\_price() {

return pc\_price;

}

public void setPc\_price(String pc\_price) {

this.pc\_price = pc\_price;

}

public String getPc\_store\_name() {

return pc\_store\_name;

}

public void setPc\_store\_name(String pc\_store\_name) {

this.pc\_store\_name = pc\_store\_name;

}

}

**PrintBasket.java**

package stavroulakis\_vasilis\_thesis\_2018;

import java.awt.BorderLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.File;

import java.io.FileNotFoundException;

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JFrame;

import javax.swing.JScrollPane;

import javax.swing.JTextArea;

public class PrintBasket implements ActionListener {

@Override

public void actionPerformed(ActionEvent ae) {

//PRINT ALL PRODUCTS FROM SHOPPING BASKET

List<CameraClass> cameras\_list = new ArrayList<>();

List<GamingConsoleClass> gamingConsoles\_\_list = new ArrayList<>();

List<MobileClass> mobiles\_list = new ArrayList<>();

List<PersonalComputerClass> personalComputers\_list = new ArrayList<>();

List<TVClass> tvs\_list = new ArrayList<>();

Scanner file1 = null;

Scanner file2 = null;

Scanner file3 = null;

Scanner file4 = null;

Scanner file5 = null;

try {

file1 = new Scanner(new File("cameras\_temp\_basket.txt"));

file2 = new Scanner(new File("consoles\_temp\_basket.txt"));

file3 = new Scanner(new File("mobiles\_temp\_file.txt"));

file4 = new Scanner(new File("pcs\_temp\_file.txt"));

file5 = new Scanner(new File("tvs\_temp\_file.txt"));

} catch (FileNotFoundException ex) {

Logger.getLogger(PrintProducts.class.getName()).log(Level.SEVERE, null, ex);

}

while (file1.hasNext()) {

String line = file1.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String cameraModel = stringArr[0];

String cameraMonitorSize = stringArr[1];

String cameraMP = stringArr[2];

String cameraFramesPerSecond = stringArr[3];

String cameraRecordingType = stringArr[4];

String cameraPrice = stringArr[5];

String cameraStoreName = stringArr[6];

CameraClass camera = new CameraClass(cameraModel, cameraMonitorSize, cameraMP, cameraFramesPerSecond, cameraRecordingType, cameraPrice, cameraStoreName);

cameras\_list.add(camera);

}

}

while (file2.hasNext()) {

String line = file2.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String consoleModel = stringArr[0];

String consoleSize = stringArr[1];

String consoleYearReleased = stringArr[2];

String consoleControlllers = stringArr[3];

String consolePrice = stringArr[4];

String consoleStoreName = stringArr[5];

GamingConsoleClass console = new GamingConsoleClass(consoleModel, consoleSize, consoleYearReleased, consoleControlllers, consolePrice, consoleStoreName);

gamingConsoles\_\_list.add(console);

}

}

while (file3.hasNext()) {

String line = file3.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String mobileModel = stringArr[0];

String mobileFrontCamera = stringArr[1];

String mobileBackCamera = stringArr[2];

String mobileScreenSize = stringArr[3];

String mobileCPU = stringArr[4];

String mobileStorage = stringArr[5];

String mobilePrice = stringArr[6];

String mobileStoreName = stringArr[7];

MobileClass mobile = new MobileClass(mobileModel, mobileFrontCamera, mobileBackCamera, mobileScreenSize, mobileCPU, mobileStorage, mobilePrice, mobileStoreName);

mobiles\_list.add(mobile);

}

}

while (file4.hasNext()) {

String line = file4.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String pcModelName = stringArr[0];

String pcCpuName = stringArr[1];

String pcClock = stringArr[2];

String pcGPUname = stringArr[3];

String pcHardDrive = stringArr[4];

String pcRamSize = stringArr[5];

String pcPrice = stringArr[4];

String pcStoreName = stringArr[5];

PersonalComputerClass pc = new PersonalComputerClass(pcModelName, pcCpuName, pcClock, pcGPUname, pcHardDrive, pcRamSize, pcPrice, pcStoreName);

personalComputers\_list.add(pc);

}

}

while (file5.hasNext()) {

String line = file5.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String tvModelName = stringArr[0];

String tvType = stringArr[1];

String tvYear = stringArr[2];

String tvMonitorSize = stringArr[3];

String tvPrice = stringArr[4];

String tvStoreName = stringArr[5];

TVClass tv = new TVClass(tvModelName, tvType, tvYear, tvMonitorSize, tvPrice, tvStoreName);

tvs\_list.add(tv);

}

}

JFrame frame = new JFrame("Print Shopping Basket");

frame.setLayout(new BorderLayout());

frame.setSize(800, 500);

JTextArea textArea = new JTextArea();

frame.add(textArea);

JScrollPane sp = new JScrollPane(textArea);

textArea.setEditable(false);

JScrollPane scroll = new JScrollPane(textArea,

JScrollPane.VERTICAL\_SCROLLBAR\_ALWAYS, JScrollPane.HORIZONTAL\_SCROLLBAR\_ALWAYS);

frame.add(scroll);

textArea.append("-----------------------------------CAMERA PRODUCTS-----------------------------------");

textArea.append("\n");

for (CameraClass c : cameras\_list) {

textArea.append("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" + "\n");

textArea.append("Camera Model : " + c.getCamera\_model() + "\n");

textArea.append("Camera Monitor Size : " + c.getMonitor\_size() + "\n");

textArea.append("Camera Megapixels : " + c.getMegapixel\_size() + "\n");

textArea.append("Camera Frames/Sec : " + c.getFrames\_per\_second() + "\n");

textArea.append("Camera Recording Type : " + c.getRecording\_type() + "\n");

textArea.append("Camera Price : " + c.getCamera\_price() + "\n");

textArea.append("Camera Store Name : " + c.getStore\_name() + "\n");

textArea.append("\n");

}

textArea.append("-----------------------------------GAMING CONSOLES-----------------------------------");

textArea.append("\n");

for (GamingConsoleClass gm : gamingConsoles\_\_list) {

textArea.append("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" + "\n");

textArea.append("Console Model : " + gm.getConsole\_name() + "\n");

textArea.append("Console HD Size : " + gm.getConsole\_size() + "\n");

textArea.append("Console Controllers : " + gm.getConsole\_controllers\_count() + "\n");

textArea.append("Console Price : " + gm.getConsole\_price() + "\n");

textArea.append("Console Store Name : " + gm.getStore\_name() + "\n");

textArea.append("\n");

}

textArea.append("-----------------------------------MOBILES-----------------------------------");

textArea.append("\n");

for (MobileClass m : mobiles\_list) {

textArea.append("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" + "\n");

textArea.append("Mobile Model : " + m.getMobile\_model() + "\n");

textArea.append("Mobile Front Camera Size : " + m.getFront\_camera\_mpixel() + "\n");

textArea.append("Mobile Back Camera Size : " + m.getBack\_camera\_mpixel() + "\n");

textArea.append("Mobile Screen Size : " + m.getMobile\_screen\_size() + "\n");

textArea.append("Mobile CPU Clock : " + m.getMobile\_cpu() + "\n");

textArea.append("Mobile Storage : " + m.getMobile\_storage() + "\n");

textArea.append("Mobile Price : " + m.getMobile\_price() + "\n");

textArea.append("Mobile Store Name : " + m.getStore\_name() + "\n");

textArea.append("\n");

}

textArea.append("-----------------------------------PERSONAL COMPUTERS-----------------------------------");

textArea.append("\n");

for (PersonalComputerClass pc : personalComputers\_list) {

textArea.append("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" + "\n");

textArea.append("PC Model : " + pc.getModel\_name() + "\n");

textArea.append("PC CPU Name : " + pc.getPc\_cpu\_name() + "\n");

textArea.append("PC CPU Clock : " + pc.getCpu\_clock() + "\n");

textArea.append("PC GPU Name : " + pc.getGraphics\_card\_name() + "\n");

textArea.append("PC Hard Drive Size : " + pc.getPc\_hard\_drive() + "\n");

textArea.append("PC RAM Size : " + pc.getPc\_ram\_size() + "\n");

textArea.append("PC Price : " + pc.getPc\_price() + "\n");

textArea.append("PC Store Name : " + pc.getPc\_store\_name() + "\n");

textArea.append("\n");

}

textArea.append("-----------------------------------TELEVISIONS-----------------------------------");

textArea.append("\n");

for (TVClass tv : tvs\_list) {

textArea.append("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" + "\n");

textArea.append("TV Model : " + tv.getTv\_model() + "\n");

textArea.append("TV Type : " + tv.getTv\_type() + "\n");

textArea.append("TV Year of Release: " + tv.getTv\_year() + "\n");

textArea.append("TV Monitor Size : " + tv.getTv\_monitor\_size() + "\n");

textArea.append("TV Price : " + tv.getTv\_price() + "\n");

textArea.append("PC Store Name : " + tv.getStore\_name() + "\n");

textArea.append("\n");

}

file1.close();

file2.close();

file3.close();

file4.close();

file5.close();

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.setResizable(false);

}

}

**PrintCompanies.java**

package stavroulakis\_vasilis\_thesis\_2018;

import java.awt.BorderLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.File;

import java.io.FileNotFoundException;

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JFrame;

import javax.swing.JTextArea;

public class PrintCompanies implements ActionListener {

@Override

public void actionPerformed(ActionEvent ae) {

List<CompanyClass> companies\_list = new ArrayList<>();

Scanner file = null;

try {

file = new Scanner(new File("companies.txt"));

} catch (FileNotFoundException ex) {

Logger.getLogger(PrintCompanies.class.getName()).log(Level.SEVERE, null, ex);

}

while (file.hasNext()) {

String line = file.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String name = stringArr[0];

String owner = stringArr[1];

String ITN = stringArr[2];;

String location = stringArr[3];

CompanyClass companies = new CompanyClass(name, owner, ITN, location);

companies\_list.add(companies);

}

}

//print companies

JFrame frame = new JFrame("Print Companies");

frame.setLayout(new BorderLayout());

frame.setSize(800, 600);

JTextArea textArea = new JTextArea();

frame.add(textArea);

textArea.setEditable(false);

frame.getContentPane().add(textArea);

for (CompanyClass s : companies\_list) {

textArea.append("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"+"\n");

textArea.append("Name : "+ s.getCompany\_name()+"\n");

textArea.append("Owner : "+ s.getCompany\_owner()+"\n");

textArea.append("ITN : "+ s.getCompany\_ITN()+"\n");

textArea.append("Location : "+ s.getCompany\_location()+"\n");

textArea.append("\n");

}

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.setResizable(false);

}

}

**PrintCustomers.java**

package stavroulakis\_vasilis\_thesis\_2018;

import java.awt.BorderLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.File;

import java.io.FileNotFoundException;

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JFrame;

import javax.swing.JTextArea;

public class PrintCustomers implements ActionListener {

@Override

public void actionPerformed(ActionEvent ae) {

List<RegisterClass> customers\_list = new ArrayList<>();

Scanner file = null;

try {

file = new Scanner(new File("customers.txt"));

} catch (FileNotFoundException ex) {

Logger.getLogger(PrintCompanies.class.getName()).log(Level.SEVERE, null, ex);

}

while (file.hasNext()) {

String line = file.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String fname = stringArr[0];

String surname = stringArr[1];

String address = stringArr[2];;

String city = stringArr[3];

String TK = stringArr[4];

String AFM = stringArr[5];

String username = stringArr[6];

String password = stringArr[7];

RegisterClass customers = new RegisterClass(fname, surname, address, city, TK, AFM, username, password);

customers\_list.add(customers);

}

}

JFrame frame = new JFrame("Print Customers");

frame.setLayout(new BorderLayout());

frame.setSize(800, 600);

JTextArea textArea = new JTextArea();

frame.add(textArea);

textArea.setEditable(false);

frame.getContentPane().add(textArea);

for (RegisterClass s : customers\_list) {

textArea.append("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" + "\n");

textArea.append("Firstname : " + s.getFname() + "\n");

textArea.append("Lastname : " + s.getSurname() + "\n");

textArea.append("Address : " + s.getAddress() + "\n");

textArea.append("City : " + s.getCity() + "\n");

textArea.append("TK : " + s.getTK() + "\n");

textArea.append("AFM : " + s.getAFM() + "\n");

textArea.append("Username : " + s.getUsername() + "\n");

textArea.append("Password : " + s.getPassword() + "\n");

textArea.append("\n");

}

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.setResizable(false);

}

}

**PrintFinishedOrder.java**

package stavroulakis\_vasilis\_thesis\_2018;

import java.awt.BorderLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.File;

import java.io.FileNotFoundException;

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JFrame;

import javax.swing.JScrollPane;

import javax.swing.JTextArea;

public class PrintFinishedOrders implements ActionListener {

@Override

public void actionPerformed(ActionEvent ae) {

List<CustomerOrderClass> finished\_orders = new ArrayList<>();

Scanner file = null;

try {

file = new Scanner(new File("customer\_orders.txt"));

} catch (FileNotFoundException ex) {

Logger.getLogger(PrintFinishedOrders.class.getName()).log(Level.SEVERE, null, ex);

}

//reconstruct customer\_orders file

while (file.hasNext()) {

String line = file.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String username = stringArr[0];

String surname = stringArr[1];

String address = stringArr[2];

String city = stringArr[3];

String order\_id = stringArr[4];

String time\_zone = stringArr[5];

String timeFinished = stringArr[6];

String order\_cost = stringArr[7]; //STOPPED HERE 17-04-218 BOUNDARIES ERROR 7

int order\_id\_int = Integer.parseInt(order\_id);

int timeFinished\_int = Integer.parseInt(timeFinished);

double order\_cost\_double = Double.parseDouble(order\_cost);

CustomerOrderClass customer = new CustomerOrderClass(username, surname, address, city, order\_id\_int, time\_zone, timeFinished\_int, order\_cost\_double);

finished\_orders.add(customer);

}

}

//PRINTING

JFrame frame = new JFrame("Print Orders");

frame.setLayout(new BorderLayout());

frame.setSize(800, 500);

JTextArea textArea = new JTextArea();

frame.add(textArea);

JScrollPane sp = new JScrollPane(textArea);

textArea.setEditable(false);

JScrollPane scroll = new JScrollPane(textArea,

JScrollPane.VERTICAL\_SCROLLBAR\_ALWAYS, JScrollPane.HORIZONTAL\_SCROLLBAR\_ALWAYS);

frame.add(scroll);

textArea.append("-----------------------------------ORDERS-----------------------------------");

textArea.append("\n");

for (CustomerOrderClass c : finished\_orders) {

textArea.append("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" + "\n");

textArea.append("Customer Username : " + c.getUsername() + "\n");

textArea.append("Customer : " + c.getSurname() + "\n");

textArea.append("Customer Address : " + c.getAddress() + "\n");

textArea.append("Customer City : " + c.getCity() + "\n");

textArea.append("Customer Order ID : " + c.getOrder\_id() + "\n");

textArea.append("Customer TimeZone : " + c.getTime\_zone() + "\n");

textArea.append("Customer Finished(Days) : " + c.getTimeFinished() + "\n");

textArea.append("Customer Order Cost : " + c.getOrder\_cost() + "\n");

textArea.append("\n");

}

file.close();

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.setResizable(false);

}

}

**PrintProducts.java**

package stavroulakis\_vasilis\_thesis\_2018;

import java.awt.BorderLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.File;

import java.io.FileNotFoundException;

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JFrame;

import javax.swing.JScrollPane;

import javax.swing.JTextArea;

public class PrintProducts implements ActionListener {

@Override

public void actionPerformed(ActionEvent ae) {

List<CameraClass> cameras\_list = new ArrayList<>();

List<GamingConsoleClass> gamingConsoles\_\_list = new ArrayList<>();

List<MobileClass> mobiles\_list = new ArrayList<>();

List<PersonalComputerClass> personalComputers\_list = new ArrayList<>();

List<TVClass> tvs\_list = new ArrayList<>();

Scanner file1 = null;

Scanner file2 = null;

Scanner file3 = null;

Scanner file4 = null;

Scanner file5 = null;

try {

file1 = new Scanner(new File("cameras.txt"));

file2 = new Scanner(new File("gaming\_consoles.txt"));

file3 = new Scanner(new File("mobiles.txt"));

file4 = new Scanner(new File("personalComputers.txt"));

file5 = new Scanner(new File("tvs.txt"));

} catch (FileNotFoundException ex) {

Logger.getLogger(PrintProducts.class.getName()).log(Level.SEVERE, null, ex);

}

while (file1.hasNext()) {

String line = file1.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String cameraModel = stringArr[0];

String cameraMonitorSize = stringArr[1];

String cameraMP = stringArr[2];

String cameraFramesPerSecond = stringArr[3];

String cameraRecordingType = stringArr[4];

String cameraPrice = stringArr[5];

String cameraStoreName = stringArr[6];

CameraClass camera = new CameraClass(cameraModel, cameraMonitorSize, cameraMP, cameraFramesPerSecond, cameraRecordingType, cameraPrice, cameraStoreName);

cameras\_list.add(camera);

}

}

while (file2.hasNext()) {

String line = file2.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String consoleModel = stringArr[0];

String consoleSize = stringArr[1];

String consoleYearReleased = stringArr[2];

String consoleControlllers = stringArr[3];

String consolePrice = stringArr[4];

String consoleStoreName = stringArr[5];

GamingConsoleClass console = new GamingConsoleClass(consoleModel, consoleSize, consoleYearReleased, consoleControlllers, consolePrice, consoleStoreName);

gamingConsoles\_\_list.add(console);

}

}

while (file3.hasNext()) {

String line = file3.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String mobileModel = stringArr[0];

String mobileFrontCamera = stringArr[1];

String mobileBackCamera = stringArr[2];

String mobileScreenSize = stringArr[3];

String mobileCPU = stringArr[4];

String mobileStorage = stringArr[5];

String mobilePrice = stringArr[6];

String mobileStoreName = stringArr[7];

MobileClass mobile = new MobileClass(mobileModel, mobileFrontCamera, mobileBackCamera, mobileScreenSize, mobileCPU, mobileStorage, mobilePrice, mobileStoreName);

mobiles\_list.add(mobile);

}

}

while (file4.hasNext()) {

String line = file4.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String pcModelName = stringArr[0];

String pcCpuName = stringArr[1];

String pcClock = stringArr[2];

String pcGPUname = stringArr[3];

String pcHardDrive = stringArr[4];

String pcRamSize = stringArr[5];

String pcPrice = stringArr[4];

String pcStoreName = stringArr[5];

PersonalComputerClass pc = new PersonalComputerClass(pcModelName, pcCpuName, pcClock, pcGPUname, pcHardDrive, pcRamSize, pcPrice, pcStoreName);

personalComputers\_list.add(pc);

}

}

while (file5.hasNext()) {

String line = file5.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String tvModelName = stringArr[0];

String tvType = stringArr[1];

String tvYear = stringArr[2];

String tvMonitorSize = stringArr[3];

String tvPrice = stringArr[4];

String tvStoreName = stringArr[5];

TVClass tv = new TVClass(tvModelName, tvType, tvYear, tvMonitorSize, tvPrice, tvStoreName);

tvs\_list.add(tv);

}

}

JFrame frame = new JFrame("Print Products");

frame.setLayout(new BorderLayout());

frame.setSize(800, 500);

JTextArea textArea = new JTextArea();

frame.add(textArea);

JScrollPane sp = new JScrollPane(textArea);

textArea.setEditable(false);

JScrollPane scroll = new JScrollPane(textArea,

JScrollPane.VERTICAL\_SCROLLBAR\_ALWAYS, JScrollPane.HORIZONTAL\_SCROLLBAR\_ALWAYS);

frame.add(scroll);

textArea.append("-----------------------------------CAMERA PRODUCTS-----------------------------------");

textArea.append("\n");

for (CameraClass c : cameras\_list) {

textArea.append("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" + "\n");

textArea.append("Camera Model : " + c.getCamera\_model() + "\n");

textArea.append("Camera Monitor Size : " + c.getMonitor\_size() + "\n");

textArea.append("Camera Megapixels : " + c.getMegapixel\_size() + "\n");

textArea.append("Camera Frames/Sec : " + c.getFrames\_per\_second() + "\n");

textArea.append("Camera Recording Type : " + c.getRecording\_type() + "\n");

textArea.append("Camera Price : " + c.getCamera\_price() + "\n");

textArea.append("Camera Store Name : " + c.getStore\_name() + "\n");

textArea.append("\n");

}

textArea.append("-----------------------------------GAMING CONSOLES-----------------------------------");

textArea.append("\n");

for (GamingConsoleClass gm : gamingConsoles\_\_list) {

textArea.append("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" + "\n");

textArea.append("Console Model : " + gm.getConsole\_name() + "\n");

textArea.append("Console HD Size : " + gm.getConsole\_size() + "\n");

textArea.append("Console Controllers : " + gm.getConsole\_controllers\_count() + "\n");

textArea.append("Console Price : " + gm.getConsole\_price() + "\n");

textArea.append("Console Store Name : " + gm.getStore\_name() + "\n");

textArea.append("\n");

}

textArea.append("-----------------------------------MOBILES-----------------------------------");

textArea.append("\n");

for (MobileClass m : mobiles\_list) {

textArea.append("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" + "\n");

textArea.append("Mobile Model : " + m.getMobile\_model() + "\n");

textArea.append("Mobile Front Camera Size : " + m.getFront\_camera\_mpixel() + "\n");

textArea.append("Mobile Back Camera Size : " + m.getBack\_camera\_mpixel() + "\n");

textArea.append("Mobile Screen Size : " + m.getMobile\_screen\_size() + "\n");

textArea.append("Mobile CPU Clock : " + m.getMobile\_cpu() + "\n");

textArea.append("Mobile Storage : " + m.getMobile\_storage() + "\n");

textArea.append("Mobile Price : " + m.getMobile\_price() + "\n");

textArea.append("Mobile Store Name : " + m.getStore\_name() + "\n");

textArea.append("\n");

}

textArea.append("-----------------------------------PERSONAL COMPUTERS-----------------------------------");

textArea.append("\n");

for (PersonalComputerClass pc : personalComputers\_list) {

textArea.append("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" + "\n");

textArea.append("PC Model : " + pc.getModel\_name() + "\n");

textArea.append("PC CPU Name : " + pc.getPc\_cpu\_name() + "\n");

textArea.append("PC CPU Clock : " + pc.getCpu\_clock() + "\n");

textArea.append("PC GPU Name : " + pc.getGraphics\_card\_name() + "\n");

textArea.append("PC Hard Drive Size : " + pc.getPc\_hard\_drive() + "\n");

textArea.append("PC RAM Size : " + pc.getPc\_ram\_size() + "\n");

textArea.append("PC Price : " + pc.getPc\_price() + "\n");

textArea.append("PC Store Name : " + pc.getPc\_store\_name() + "\n");

textArea.append("\n");

}

textArea.append("-----------------------------------TELEVISIONS-----------------------------------");

textArea.append("\n");

for (TVClass tv : tvs\_list) {

textArea.append("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" + "\n");

textArea.append("TV Model : " + tv.getTv\_model() + "\n");

textArea.append("TV Type : " + tv.getTv\_type() + "\n");

textArea.append("TV Year of Release: " + tv.getTv\_year() + "\n");

textArea.append("TV Monitor Size : " + tv.getTv\_monitor\_size() + "\n");

textArea.append("TV Price : " + tv.getTv\_price() + "\n");

textArea.append("PC Store Name : " + tv.getStore\_name() + "\n");

textArea.append("\n");

}

file1.close();

file2.close();

file3.close();

file4.close();

file5.close();

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.setResizable(false);

}

}

**PrintBasket.java**

package stavroulakis\_vasilis\_thesis\_2018;

import java.awt.BorderLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.File;

import java.io.FileNotFoundException;

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JFrame;

import javax.swing.JScrollPane;

import javax.swing.JTextArea;

public class PrintBasket implements ActionListener {

@Override

public void actionPerformed(ActionEvent ae) {

//PRINT ALL PRODUCTS FROM SHOPPING BASKET

List<CameraClass> cameras\_list = new ArrayList<>();

List<GamingConsoleClass> gamingConsoles\_\_list = new ArrayList<>();

List<MobileClass> mobiles\_list = new ArrayList<>();

List<PersonalComputerClass> personalComputers\_list = new ArrayList<>();

List<TVClass> tvs\_list = new ArrayList<>();

Scanner file1 = null;

Scanner file2 = null;

Scanner file3 = null;

Scanner file4 = null;

Scanner file5 = null;

try {

file1 = new Scanner(new File("cameras\_temp\_basket.txt"));

file2 = new Scanner(new File("consoles\_temp\_basket.txt"));

file3 = new Scanner(new File("mobiles\_temp\_file.txt"));

file4 = new Scanner(new File("pcs\_temp\_file.txt"));

file5 = new Scanner(new File("tvs\_temp\_file.txt"));

} catch (FileNotFoundException ex) {

Logger.getLogger(PrintProducts.class.getName()).log(Level.SEVERE, null, ex);

}

while (file1.hasNext()) {

String line = file1.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String cameraModel = stringArr[0];

String cameraMonitorSize = stringArr[1];

String cameraMP = stringArr[2];

String cameraFramesPerSecond = stringArr[3];

String cameraRecordingType = stringArr[4];

String cameraPrice = stringArr[5];

String cameraStoreName = stringArr[6];

CameraClass camera = new CameraClass(cameraModel, cameraMonitorSize, cameraMP, cameraFramesPerSecond, cameraRecordingType, cameraPrice, cameraStoreName);

cameras\_list.add(camera);

}

}

while (file2.hasNext()) {

String line = file2.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String consoleModel = stringArr[0];

String consoleSize = stringArr[1];

String consoleYearReleased = stringArr[2];

String consoleControlllers = stringArr[3];

String consolePrice = stringArr[4];

String consoleStoreName = stringArr[5];

GamingConsoleClass console = new GamingConsoleClass(consoleModel, consoleSize, consoleYearReleased, consoleControlllers, consolePrice, consoleStoreName);

gamingConsoles\_\_list.add(console);

}

}

while (file3.hasNext()) {

String line = file3.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String mobileModel = stringArr[0];

String mobileFrontCamera = stringArr[1];

String mobileBackCamera = stringArr[2];

String mobileScreenSize = stringArr[3];

String mobileCPU = stringArr[4];

String mobileStorage = stringArr[5];

String mobilePrice = stringArr[6];

String mobileStoreName = stringArr[7];

MobileClass mobile = new MobileClass(mobileModel, mobileFrontCamera, mobileBackCamera, mobileScreenSize, mobileCPU, mobileStorage, mobilePrice, mobileStoreName);

mobiles\_list.add(mobile);

}

}

while (file4.hasNext()) {

String line = file4.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String pcModelName = stringArr[0];

String pcCpuName = stringArr[1];

String pcClock = stringArr[2];

String pcGPUname = stringArr[3];

String pcHardDrive = stringArr[4];

String pcRamSize = stringArr[5];

String pcPrice = stringArr[4];

String pcStoreName = stringArr[5];

PersonalComputerClass pc = new PersonalComputerClass(pcModelName, pcCpuName, pcClock, pcGPUname, pcHardDrive, pcRamSize, pcPrice, pcStoreName);

personalComputers\_list.add(pc);

}

}

while (file5.hasNext()) {

String line = file5.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String tvModelName = stringArr[0];

String tvType = stringArr[1];

String tvYear = stringArr[2];

String tvMonitorSize = stringArr[3];

String tvPrice = stringArr[4];

String tvStoreName = stringArr[5];

TVClass tv = new TVClass(tvModelName, tvType, tvYear, tvMonitorSize, tvPrice, tvStoreName);

tvs\_list.add(tv);

}

}

JFrame frame = new JFrame("Print Shopping Basket");

frame.setLayout(new BorderLayout());

frame.setSize(800, 500);

JTextArea textArea = new JTextArea();

frame.add(textArea);

JScrollPane sp = new JScrollPane(textArea);

textArea.setEditable(false);

JScrollPane scroll = new JScrollPane(textArea,

JScrollPane.VERTICAL\_SCROLLBAR\_ALWAYS, JScrollPane.HORIZONTAL\_SCROLLBAR\_ALWAYS);

frame.add(scroll);

textArea.append("-----------------------------------CAMERA PRODUCTS-----------------------------------");

textArea.append("\n");

for (CameraClass c : cameras\_list) {

textArea.append("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" + "\n");

textArea.append("Camera Model : " + c.getCamera\_model() + "\n");

textArea.append("Camera Monitor Size : " + c.getMonitor\_size() + "\n");

textArea.append("Camera Megapixels : " + c.getMegapixel\_size() + "\n");

textArea.append("Camera Frames/Sec : " + c.getFrames\_per\_second() + "\n");

textArea.append("Camera Recording Type : " + c.getRecording\_type() + "\n");

textArea.append("Camera Price : " + c.getCamera\_price() + "\n");

textArea.append("Camera Store Name : " + c.getStore\_name() + "\n");

textArea.append("\n");

}

textArea.append("-----------------------------------GAMING CONSOLES-----------------------------------");

textArea.append("\n");

for (GamingConsoleClass gm : gamingConsoles\_\_list) {

textArea.append("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" + "\n");

textArea.append("Console Model : " + gm.getConsole\_name() + "\n");

textArea.append("Console HD Size : " + gm.getConsole\_size() + "\n");

textArea.append("Console Controllers : " + gm.getConsole\_controllers\_count() + "\n");

textArea.append("Console Price : " + gm.getConsole\_price() + "\n");

textArea.append("Console Store Name : " + gm.getStore\_name() + "\n");

textArea.append("\n");

}

textArea.append("-----------------------------------MOBILES-----------------------------------");

textArea.append("\n");

for (MobileClass m : mobiles\_list) {

textArea.append("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" + "\n");

textArea.append("Mobile Model : " + m.getMobile\_model() + "\n");

textArea.append("Mobile Front Camera Size : " + m.getFront\_camera\_mpixel() + "\n");

textArea.append("Mobile Back Camera Size : " + m.getBack\_camera\_mpixel() + "\n");

textArea.append("Mobile Screen Size : " + m.getMobile\_screen\_size() + "\n");

textArea.append("Mobile CPU Clock : " + m.getMobile\_cpu() + "\n");

textArea.append("Mobile Storage : " + m.getMobile\_storage() + "\n");

textArea.append("Mobile Price : " + m.getMobile\_price() + "\n");

textArea.append("Mobile Store Name : " + m.getStore\_name() + "\n");

textArea.append("\n");

}

textArea.append("-----------------------------------PERSONAL COMPUTERS-----------------------------------");

textArea.append("\n");

for (PersonalComputerClass pc : personalComputers\_list) {

textArea.append("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" + "\n");

textArea.append("PC Model : " + pc.getModel\_name() + "\n");

textArea.append("PC CPU Name : " + pc.getPc\_cpu\_name() + "\n");

textArea.append("PC CPU Clock : " + pc.getCpu\_clock() + "\n");

textArea.append("PC GPU Name : " + pc.getGraphics\_card\_name() + "\n");

textArea.append("PC Hard Drive Size : " + pc.getPc\_hard\_drive() + "\n");

textArea.append("PC RAM Size : " + pc.getPc\_ram\_size() + "\n");

textArea.append("PC Price : " + pc.getPc\_price() + "\n");

textArea.append("PC Store Name : " + pc.getPc\_store\_name() + "\n");

textArea.append("\n");

}

textArea.append("-----------------------------------TELEVISIONS-----------------------------------");

textArea.append("\n");

for (TVClass tv : tvs\_list) {

textArea.append("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" + "\n");

textArea.append("TV Model : " + tv.getTv\_model() + "\n");

textArea.append("TV Type : " + tv.getTv\_type() + "\n");

textArea.append("TV Year of Release: " + tv.getTv\_year() + "\n");

textArea.append("TV Monitor Size : " + tv.getTv\_monitor\_size() + "\n");

textArea.append("TV Price : " + tv.getTv\_price() + "\n");

textArea.append("PC Store Name : " + tv.getStore\_name() + "\n");

textArea.append("\n");

}

file1.close();

file2.close();

file3.close();

file4.close();

file5.close();

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.setResizable(false);

}

}

**PrintProductsByCategory.java**

package stavroulakis\_vasilis\_thesis\_2018;

import java.awt.BorderLayout;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JTextField;

public class PrintProductsByCategory implements ActionListener {

@Override

public void actionPerformed(ActionEvent ae) {

JFrame frame = new JFrame("Print by Type");

frame.setSize(360, 300);

frame.setLayout(new BorderLayout());

JPanel panel = new JPanel();

frame.add(panel);

panel.setLayout(new GridLayout(6, 1));

JButton btn1 = new JButton("CAMERAS");

JButton btn2 = new JButton("GAMING CONSOLES");

JButton btn3 = new JButton("MOBILES");

JButton btn4 = new JButton("PERSONAL COMPUTERS");

JButton btn5 = new JButton("TVS");

JButton btn6 = new JButton("BACK");

panel.add(btn1);

panel.add(btn2);

panel.add(btn3);

panel.add(btn4);

panel.add(btn5);

panel.add(btn6);

//actionListeners for buttons

btn1.addActionListener(new printCameras());

btn2.addActionListener(new printGamingConsoles());

btn3.addActionListener(new printMobiles());

btn4.addActionListener(new printPersonalComputers());

btn5.addActionListener(new printTVS());

btn6.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ae) {

frame.dispose();

}

});

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.setResizable(false);

}

}

**RegisterClass.java**

package stavroulakis\_vasilis\_thesis\_2018;

import java.io.Serializable;

public class RegisterClass {

String fname;

String surname;

String address;

String city;

String TK;

String AFM;

String username;

String password;

public RegisterClass(String fname, String surname, String address, String city, String TK, String AFM, String username, String password) {

this.fname = fname;

this.surname = surname;

this.address = address;

this.city = city;

this.TK = TK;

this.AFM = AFM;

this.username = username;

this.password = password;

}

public String getCity() {

return city;

}

public void setCity(String city) {

this.city = city;

}

public String getAFM() {

return AFM;

}

public void setAFM(String AFM) {

this.AFM = AFM;

}

@Override

public String toString() {

return this.getFname()+"\t"+this.getSurname()+"\t"+this.getAddress()+"\t"+this.getCity()+"\t"+this.getTK()+"\t"+this.getAFM()+"\t"+this.getUsername()+"\t"+this.getPassword()+"\n";

}

public String getFname() {

return fname;

}

public void setFname(String fname) {

this.fname = fname;

}

public String getSurname() {

return surname;

}

public void setSurname(String surname) {

this.surname = surname;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

public String getTK() {

return TK;

}

public void setTK(String TK) {

this.TK = TK;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

}

**RegisterCustomer.java**

package stavroulakis\_vasilis\_thesis\_2018;

import java.awt.BorderLayout;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileWriter;

import java.io.IOException;

import java.util.ArrayList;

import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JPasswordField;

import javax.swing.JTextField;

public class RegisterCustomer implements ActionListener {

@Override

public void actionPerformed(ActionEvent ae) {

JFrame frame = new JFrame("Register");

frame.setSize(360, 500);

frame.setLayout(new BorderLayout());

JPanel panel = new JPanel();

frame.add(panel);

panel.setLayout(new GridLayout(17, 1));

JLabel label1 = new JLabel("Firstname");

JLabel label2 = new JLabel("Surname");

JLabel label3 = new JLabel("Address");

JLabel label4 = new JLabel("City");

JLabel label5 = new JLabel("TK");

JLabel label6 = new JLabel("AFM");

JLabel label7 = new JLabel("Username");

JLabel label8 = new JLabel("Password");

JTextField text1 = new JTextField(20);

JTextField text2 = new JTextField(20);

JTextField text3 = new JTextField(20);

JTextField text4 = new JTextField(20);

JTextField text5 = new JTextField(20);

JTextField text6 = new JTextField(20);

JTextField text7 = new JTextField(20);

JPasswordField text8 = new JPasswordField(16);

JButton register = new JButton("REGISTER");

panel.add(label1);

panel.add(text1);

panel.add(label2);

panel.add(text2);

panel.add(label3);

panel.add(text3);

panel.add(label4);

panel.add(text4);

panel.add(label5);

panel.add(text5);

panel.add(label6);

panel.add(text6);

panel.add(label7);

panel.add(text7);

panel.add(label8);

panel.add(text8);

panel.add(register);

//register button writes in file

register.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ae) {

String a1 = text1.getText();

String a2 = text2.getText();

String a3 = text3.getText();

String a4 = text4.getText();

String a5 = text5.getText();

String a6 = text6.getText();

String a7 = text7.getText();

String a8 = text8.getText();

Scanner file = null;

ArrayList<RegisterClass> registered\_customers = new ArrayList<>();

try {

file = new Scanner(new File("customers.txt"));

} catch (FileNotFoundException ex) {

Logger.getLogger(RegisterCustomer.class.getName()).log(Level.SEVERE, null, ex);

}

while (file.hasNext()) {

String line = file.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String fname = stringArr[0];

String surname = stringArr[1];

String address = stringArr[2];

String city = stringArr[3];

String TK = stringArr[4];

String AFM = stringArr[5];

String username = stringArr[6];

String password = stringArr[7];

RegisterClass customer = new RegisterClass(fname, surname, address, city, TK, AFM, username, password);

registered\_customers.add(customer);

}

}

int j = 0;

for (int i = 0; i < registered\_customers.size(); i++) {

RegisterClass cc = registered\_customers.get(i);

if (cc.getAFM().equals(a6) || cc.getUsername().equals(a7)) {

j = i + 1;

}

}

if (j <= 0) {

RegisterClass customer = new RegisterClass(a1, a2, a3, a4, a5, a6, a7, a8);

writeCustomers(customer);

}

if (j > 0) {

System.out.println("THIS CUSTOMER IS IN OUR SYSTEM!");

}

frame.dispose();

}

});

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.setResizable(false);

}

public void writeCustomers(RegisterClass c) {

try {

String current = new java.io.File(".").getCanonicalPath();

File file = new File(current + "\\customers.txt");

if (!file.exists()) {

file.createNewFile();

}

FileWriter fw = new FileWriter(file, true);

BufferedWriter bw = new BufferedWriter(fw);

bw.write("\n" + c.toString());

bw.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

**RemoveCamera.java**

package stavroulakis\_vasilis\_thesis\_2018;

import java.awt.BorderLayout;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.FileWriter;

import java.io.PrintWriter;

import java.util.ArrayList;

import java.util.Iterator;

import java.util.ListIterator;

import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JTextField;

public class RemoveCamera implements ActionListener {

@Override

public void actionPerformed(ActionEvent ae) {

JFrame frame = new JFrame("Remove Cameras");

frame.setSize(200, 220);

frame.setLayout(new BorderLayout());

JPanel panel = new JPanel();

frame.add(panel);

panel.setLayout(new GridLayout(5, 1));

JLabel label = new JLabel("Camera model");

JTextField field = new JTextField(60);

JLabel label1 = new JLabel("Camera quantity");

JTextField field1 = new JTextField(60);

JButton btn = new JButton("CONFIRM");

panel.add(label);

panel.add(field);

panel.add(label1);

panel.add(field1);

panel.add(btn);

btn.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ae) {

String cameraModelName = field.getText();

String cameraQuantity = field1.getText();

int cameraQuantity\_integer = Integer.parseInt(cameraQuantity);

try {

reconstructArrayList(cameraModelName, cameraQuantity\_integer);

} catch (FileNotFoundException ex) {

Logger.getLogger(RemoveCamera.class.getName()).log(Level.SEVERE, null, ex);

}

frame.dispose();

}

});

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.setResizable(false);

}

public void reconstructArrayList(String cameraModelName, int cameraQuantity\_integer) throws FileNotFoundException {

Scanner file = null;

ArrayList<CameraClass> cameras\_list = new ArrayList<>();

try {

file = new Scanner(new File("cameras.txt"));

} catch (FileNotFoundException ex) {

Logger.getLogger(RemoveCamera.class.getName()).log(Level.SEVERE, null, ex);

}

while (file.hasNext()) {

String line = file.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String cameraModel = stringArr[0];

String monitorSize = stringArr[1];

String megaPixelSize = stringArr[2];

String framesPerSecond = stringArr[3];

String recordingType = stringArr[4];

String cameraPrice = stringArr[5];

String storeName = stringArr[6];

CameraClass camera = new CameraClass(cameraModel, monitorSize, megaPixelSize, framesPerSecond, recordingType, cameraPrice, storeName);

cameras\_list.add(camera);

}

}

int count\_times = cameraQuantity\_integer, product\_count = 0;

product\_count = removeUsingIterator(count\_times, cameraQuantity\_integer, cameras\_list, cameraModelName);

System.out.println(" YOU SUCCESSFULLY REMOVED:" + product\_count + " ITEMS!");

if (product\_count == 0) {

Log.e(" THIS NUMBER/TYPE OF PRODUCTS ARE NOT EXIST IN OUR SYSTEM! ");

//System.out.println(" THIS NUMBER/TYPE OF PRODUCTS ARE NOT EXIST IN OUR SYSTEM! ");

}

saveListToFile(cameras\_list, "cameras.txt");

}

public int removeUsingIterator(int count\_times, int cameraQuantity\_integer, ArrayList cameras\_list, String cameraModelName) throws FileNotFoundException {

Iterator<CameraClass> it = cameras\_list.iterator();

int product\_count = 0;

while (it.hasNext()) {

if (it.next().getCamera\_model().equals(cameraModelName)) {

count\_times--;

it.remove();

product\_count++;

}

if (count\_times == 0) {

break;

}

}

return product\_count;

}

public void saveListToFile(ArrayList cameras\_list, String fileName) throws FileNotFoundException {

PrintWriter pw = new PrintWriter(new FileOutputStream(fileName));

for (Object camera : cameras\_list) {

pw.println(camera.toString());

}

pw.close();

}

}

**RemoveCompany.java**

package stavroulakis\_vasilis\_thesis\_2018;

import java.awt.BorderLayout;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.util.ArrayList;

import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JTextField;

public class RemoveCompany implements ActionListener {

@Override

public void actionPerformed(ActionEvent ae) {

JFrame frame = new JFrame("Remove Company");

frame.setSize(200, 220);

frame.setLayout(new BorderLayout());

JPanel panel = new JPanel();

frame.add(panel);

panel.setLayout(new GridLayout(3, 1));

JLabel label = new JLabel("Company's ITN");

JTextField field = new JTextField(30);

JButton btn = new JButton("CONFIRM");

panel.add(label);

panel.add(field);

panel.add(btn);

btn.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ae) {

String company\_ITN = field.getText();

removeCompany(company\_ITN);

frame.dispose();

}

});

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.setResizable(false);

}

public void removeCompany(String company\_ITN) {

Scanner file = null;

ArrayList<CompanyClass> companies\_list = new ArrayList<>();

try {

file = new Scanner(new File("companies.txt"));

} catch (FileNotFoundException ex) {

Logger.getLogger(RemoveCompany.class.getName()).log(Level.SEVERE, null, ex);

}

//we try to find the given company using scanner function, an array and reconstruct the companies\_list

while (file.hasNext()) {

String line = file.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String name = stringArr[0];

String owner = stringArr[1];

String ITN = stringArr[2];

String location = stringArr[3];

CompanyClass companies = new CompanyClass(name, owner, ITN, location);

companies\_list.add(companies);

}

}

String lineToRemove = null;

for (int i = 0; i < companies\_list.size(); i++) {

CompanyClass cc = companies\_list.get(i);

if (cc.getCompany\_ITN().equals(company\_ITN)) {

lineToRemove = cc.getCompany\_name();

break;

}

}

//do not forget to close the file. If you dont, then you cannot use delete or rename functions!

file.close();

String currentt;

File inputFile = null, tempFile = null;

try {

currentt = new java.io.File(".").getCanonicalPath();

inputFile = new File(currentt + "\\companies.txt");

tempFile = new File(currentt + "\\myTempFile.txt");

} catch (IOException ex) {

Logger.getLogger(RemoveCompany.class.getName()).log(Level.SEVERE, null, ex);

}

try {

BufferedReader reader = new BufferedReader(new FileReader(inputFile));

BufferedWriter writer = new BufferedWriter(new FileWriter(tempFile));

String currentLine;

while ((currentLine = reader.readLine()) != null) {

String trimmedLine = currentLine.trim();

if (trimmedLine.contains(lineToRemove) == false) {

writer.write(currentLine + System.getProperty("line.separator"));

}

}

writer.close();

reader.close();

} catch (FileNotFoundException ex) {

Logger.getLogger(RemoveCompany.class.getName()).log(Level.SEVERE, null, ex);

} catch (IOException ex) {

Logger.getLogger(RemoveCompany.class.getName()).log(Level.SEVERE, null, ex);

}

//delete and rename a file using fileDeleteAndRename function

fileDeleteAndRename(inputFile, tempFile);

}

public void fileDeleteAndRename(File inputFile, File tempFile) {

if (inputFile.exists() == true && tempFile.exists() == true) {

inputFile.setWritable(true);

inputFile.delete();

tempFile.renameTo(new File("companies.txt"));

Log.d("SUCCESSFULL UPDATE!");

//System.out.println((""));

}

}

}

**RemoveGamingConsoles.java**

package stavroulakis\_vasilis\_thesis\_2018;

import java.awt.BorderLayout;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.PrintWriter;

import java.util.ArrayList;

import java.util.Iterator;

import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JTextField;

public class RemoveGamingConsoles implements ActionListener {

@Override

public void actionPerformed(ActionEvent ae) {

JFrame frame = new JFrame("Remove Consoles");

frame.setSize(250, 270);

frame.setLayout(new BorderLayout());

JPanel panel = new JPanel();

frame.add(panel);

panel.setLayout(new GridLayout(5, 1));

JLabel label = new JLabel("Console model");

JTextField field = new JTextField(60);

JLabel label1 = new JLabel("Console quantity");

JTextField field1 = new JTextField(60);

JButton btn = new JButton("CONFIRM");

panel.add(label);

panel.add(field);

panel.add(label1);

panel.add(field1);

panel.add(btn);

btn.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ae) {

String consoleModelName = field.getText();

String consoleQuantity = field1.getText();

int consoleQuantity\_integer = Integer.parseInt(consoleQuantity);

try {

reconstructArrayList(consoleModelName, consoleQuantity\_integer);

} catch (FileNotFoundException ex) {

Logger.getLogger(RemoveGamingConsoles.class.getName()).log(Level.SEVERE, null, ex);

}

frame.dispose();

}

});

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.setResizable(false);

}

public void reconstructArrayList(String consoleModelName, int consoleQuantity) throws FileNotFoundException {

Scanner file = null;

ArrayList<GamingConsoleClass> consoles\_list = new ArrayList<>();

try {

file = new Scanner(new File("gaming\_consoles.txt"));

} catch (FileNotFoundException ex) {

Logger.getLogger(RemoveTV.class.getName()).log(Level.SEVERE, null, ex);

}

while (file.hasNext()) {

String line = file.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String consoleName = stringArr[0];

String consoleSize = stringArr[1];

String consoleYearReleased = stringArr[2];

String consoleControllers = stringArr[3];

String consolePrice = stringArr[4];

String consoleStoreName = stringArr[5];

GamingConsoleClass console = new GamingConsoleClass(consoleName, consoleSize, consoleYearReleased, consoleControllers, consolePrice, consoleStoreName);

consoles\_list.add(console);

}

}

int count\_times = consoleQuantity, product\_count = 0;

product\_count = removeUsingIterator(count\_times, consoleQuantity, consoles\_list, consoleModelName);

System.out.println(" YOU SUCCESSFULLY REMOVED:" + product\_count + " ITEMS!");

if(product\_count==0){

Log.w(" THIS NUMBER/TYPE OF PRODUCTS ARE NOT EXIST IN OUR SYSTEM! ");

//System.out.println(" THIS NUMBER/TYPE OF PRODUCTS ARE NOT EXIST IN OUR SYSTEM! ");

}

saveListToFile(consoles\_list, "gaming\_consoles.txt");

}

public int removeUsingIterator(int count\_times, int consoleQuantity, ArrayList consoles\_list, String consoleModelName) throws FileNotFoundException {

Iterator<GamingConsoleClass> it = consoles\_list.iterator();

int product\_count = 0;

while (it.hasNext()) {

if (it.next().getConsole\_name().equals(consoleModelName)) {

count\_times--;

it.remove();

product\_count++;

}

if (count\_times == 0) {

break;

}

}

return product\_count;

}

public void saveListToFile(ArrayList consoles\_list, String fileName) throws FileNotFoundException {

PrintWriter pw = new PrintWriter(new FileOutputStream(fileName));

for (Object console : consoles\_list) {

pw.println(console.toString());

}

pw.close();

}

}

**RemoveMobiles.java**

package stavroulakis\_vasilis\_thesis\_2018;

import java.awt.BorderLayout;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.PrintWriter;

import java.util.ArrayList;

import java.util.Iterator;

import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JTextField;

public class RemoveMobiles implements ActionListener {

@Override

public void actionPerformed(ActionEvent ae) {

JFrame frame = new JFrame("Remove Mobiles");

frame.setSize(200, 220);

frame.setLayout(new BorderLayout());

JPanel panel = new JPanel();

frame.add(panel);

panel.setLayout(new GridLayout(5, 1));

JLabel label = new JLabel("Mobile model");

JTextField field = new JTextField(60);

JLabel label1 = new JLabel("Mobile quantity");

JTextField field1 = new JTextField(60);

JButton btn = new JButton("CONFIRM");

panel.add(label);

panel.add(field);

panel.add(label1);

panel.add(field1);

panel.add(btn);

btn.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ae) {

String mobileModelName = field.getText();

String mobileQuantity = field1.getText();

int mobileQuantity\_integer = Integer.parseInt(mobileQuantity);

try {

reconstructArrayList(mobileModelName, mobileQuantity\_integer);

} catch (FileNotFoundException ex) {

Logger.getLogger(RemoveMobiles.class.getName()).log(Level.SEVERE, null, ex);

}

frame.dispose();

}

});

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.setResizable(false);

}

public void reconstructArrayList(String mobileModelName, int mobileQuantity\_integer) throws FileNotFoundException {

Scanner file = null;

ArrayList<MobileClass> mobiles\_list = new ArrayList<>();

try {

file = new Scanner(new File("mobiles.txt"));

} catch (FileNotFoundException ex) {

Logger.getLogger(RemovePC.class.getName()).log(Level.SEVERE, null, ex);

}

while (file.hasNext()) {

String line = file.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String mobileName = stringArr[0];

String frontCameraMP = stringArr[1];

String backCameraMP = stringArr[2];

String mobileScreenSize = stringArr[3];

String mobileCPU = stringArr[4];

String mobileStorage = stringArr[5];

String mobilePrice = stringArr[6];

String mobileStoreName = stringArr[7];

MobileClass mobile = new MobileClass(mobileName, frontCameraMP, backCameraMP, mobileScreenSize, mobileCPU, mobileStorage, mobilePrice, mobileStoreName);

mobiles\_list.add(mobile);

}

}

int count\_times = mobileQuantity\_integer, product\_count = 0;

product\_count = removeUsingIterator(count\_times, mobileQuantity\_integer, mobiles\_list, mobileModelName);

System.out.println(" YOU SUCCESSFULLY REMOVED:" + product\_count + " ITEMS!");

if (product\_count == 0) {

Log.w(" THIS NUMBER/TYPE OF PRODUCTS ARE NOT EXIST IN OUR SYSTEM! ");

//System.out.println(" THIS NUMBER/TYPE OF PRODUCTS ARE NOT EXIST IN OUR SYSTEM! ");

}

saveListToFile(mobiles\_list, "mobiles.txt");

}

public int removeUsingIterator(int count\_times, int mobileQuantity\_integer, ArrayList mobiles\_list, String mobileModelName) throws FileNotFoundException {

Iterator<MobileClass> it = mobiles\_list.iterator();

int product\_count = 0;

while (it.hasNext()) {

if (it.next().getMobile\_model().equals(mobileModelName)) {

count\_times--;

it.remove();

product\_count++;

}

if (count\_times == 0) {

break;

}

}

return product\_count;

}

public void saveListToFile(ArrayList mobiles\_list, String fileName) throws FileNotFoundException {

PrintWriter pw = new PrintWriter(new FileOutputStream(fileName));

for (Object mobile : mobiles\_list) {

pw.println(mobile.toString());

}

pw.close();

}

}

**RemovePC.java**

package stavroulakis\_vasilis\_thesis\_2018;

import java.awt.BorderLayout;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.PrintWriter;

import java.util.ArrayList;

import java.util.Iterator;

import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JTextField;

public class RemovePC implements ActionListener {

@Override

public void actionPerformed(ActionEvent ae) {

JFrame frame = new JFrame("Remove Computers");

frame.setSize(200, 220);

frame.setLayout(new BorderLayout());

JPanel panel = new JPanel();

frame.add(panel);

panel.setLayout(new GridLayout(5, 1));

JLabel label = new JLabel("PC model");

JTextField field = new JTextField(60);

JLabel label1 = new JLabel("PC quantity");

JTextField field1 = new JTextField(60);

JButton btn = new JButton("CONFIRM");

panel.add(label);

panel.add(field);

panel.add(label1);

panel.add(field1);

panel.add(btn);

btn.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ae) {

String pcModelName = field.getText();

String pcQuantity = field1.getText();

int pcQuantity\_integer = Integer.parseInt(pcQuantity);

try {

reconstructArrayList(pcModelName, pcQuantity\_integer);

} catch (FileNotFoundException ex) {

Logger.getLogger(RemovePC.class.getName()).log(Level.SEVERE, null, ex);

}

frame.dispose();

}

});

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.setResizable(false);

}

public void reconstructArrayList(String pcModelName, int pcQuantity\_integer) throws FileNotFoundException {

Scanner file = null;

ArrayList<PersonalComputerClass> pc\_list = new ArrayList<>();

try {

file = new Scanner(new File("personalComputers.txt"));

} catch (FileNotFoundException ex) {

Logger.getLogger(RemovePC.class.getName()).log(Level.SEVERE, null, ex);

}

while (file.hasNext()) {

String line = file.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String pcName = stringArr[0];

String pcCpuName = stringArr[1];

String cpuClock = stringArr[2];

String gpuName = stringArr[3];

String pcHardDrive = stringArr[4];

String pcRamSize = stringArr[5];

String pcPrice = stringArr[6];

String pcStoreName = stringArr[7];

PersonalComputerClass pc = new PersonalComputerClass(pcName, pcCpuName, cpuClock, gpuName, pcHardDrive, pcRamSize, pcPrice, pcStoreName);

pc\_list.add(pc);

}

}

int count\_times = pcQuantity\_integer, product\_count = 0;

product\_count = removeUsingIterator(count\_times, pcQuantity\_integer, pc\_list, pcModelName);

System.out.println(" YOU SUCCESSFULLY REMOVED:" + product\_count + " ITEMS!");

if (product\_count == 0) {

Log.w(" THIS NUMBER/TYPE OF PRODUCTS ARE NOT EXIST IN OUR SYSTEM! ");

//System.out.println(" THIS NUMBER/TYPE OF PRODUCTS ARE NOT EXIST IN OUR SYSTEM! ");

}

saveListToFile(pc\_list, "personalComputers.txt");

}

public int removeUsingIterator(int count\_times, int pcQuantity\_integer, ArrayList pc\_list, String pcModelName) throws FileNotFoundException {

Iterator<PersonalComputerClass> it = pc\_list.iterator();

int product\_count = 0;

while (it.hasNext()) {

if (it.next().getModel\_name().equals(pcModelName)) {

count\_times--;

it.remove();

product\_count++;

}

if (count\_times == 0) {

break;

}

}

return product\_count;

}

public void saveListToFile(ArrayList pc\_list, String fileName) throws FileNotFoundException {

PrintWriter pw = new PrintWriter(new FileOutputStream(fileName));

for (Object pc : pc\_list) {

pw.println(pc.toString());

}

pw.close();

}

}

**RemoveTV.java**

package stavroulakis\_vasilis\_thesis\_2018;

import java.awt.BorderLayout;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.PrintWriter;

import java.util.ArrayList;

import java.util.Iterator;

import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JTextField;

public class RemoveTV implements ActionListener {

@Override

public void actionPerformed(ActionEvent ae) {

JFrame frame = new JFrame("Remove tvs");

frame.setSize(240, 260);

frame.setLayout(new BorderLayout());

JPanel panel = new JPanel();

frame.add(panel);

panel.setLayout(new GridLayout(5, 1));

JLabel label = new JLabel("TV model");

JTextField field = new JTextField(60);

JLabel label1 = new JLabel("TV quantity");

JTextField field1 = new JTextField(60);

JButton btn = new JButton("CONFIRM");

panel.add(label);

panel.add(field);

panel.add(label1);

panel.add(field1);

panel.add(btn);

btn.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ae) {

String tvModelName = field.getText();

String tvQuantity = field1.getText();

int tvQuantity\_integer = Integer.parseInt(tvQuantity);

try {

reconstructArrayList(tvModelName, tvQuantity\_integer);

} catch (FileNotFoundException ex) {

Logger.getLogger(RemoveMobiles.class.getName()).log(Level.SEVERE, null, ex);

}

frame.dispose();

}

});

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.setResizable(false);

}

public void reconstructArrayList(String tvModelName, int tvQuantity\_integer) throws FileNotFoundException {

Scanner file = null;

ArrayList<TVClass> tv\_list = new ArrayList<>();

try {

file = new Scanner(new File("tvs.txt"));

} catch (FileNotFoundException ex) {

Logger.getLogger(RemoveTV.class.getName()).log(Level.SEVERE, null, ex);

}

while (file.hasNext()) {

String line = file.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String tvName = stringArr[0];

String tvType = stringArr[1];

String tvYearMade = stringArr[2];

String tvMonitorSize = stringArr[3];

String tvPrice = stringArr[4];

String tvStoreName = stringArr[5];

TVClass tv = new TVClass(tvName, tvType, tvYearMade, tvMonitorSize, tvPrice, tvStoreName);

tv\_list.add(tv);

}

}

int count\_times = tvQuantity\_integer, product\_count = 0;

product\_count = removeUsingIterator(count\_times, tvQuantity\_integer, tv\_list, tvModelName);

System.out.println(" YOU SUCCESSFULLY REMOVED:" + product\_count + " ITEMS!");

if (product\_count == 0) {

Log.w(" THIS NUMBER/TYPE OF PRODUCTS ARE NOT EXIST IN OUR SYSTEM! ");

//System.out.println(" THIS NUMBER/TYPE OF PRODUCTS ARE NOT EXIST IN OUR SYSTEM! ");

}

saveListToFile(tv\_list, "tvs.txt");

}

public int removeUsingIterator(int count\_times, int tvQuantity\_integer, ArrayList tv\_list, String tvModelName) throws FileNotFoundException {

Iterator<TVClass> it = tv\_list.iterator();

int product\_count = 0;

while (it.hasNext()) {

if (it.next().getTv\_model().equals(tvModelName)) {

count\_times--;

it.remove();

product\_count++;

}

if (count\_times == 0) {

break;

}

}

return product\_count;

}

public void saveListToFile(ArrayList tv\_list, String fileName) throws FileNotFoundException {

PrintWriter pw = new PrintWriter(new FileOutputStream(fileName));

for (Object tv : tv\_list) {

pw.println(tv.toString());

}

pw.close();

}

}

**ShoppingBasket.java**

package stavroulakis\_vasilis\_thesis\_2018;

import java.awt.BorderLayout;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileWriter;

import java.io.IOException;

import java.util.ArrayList;

import java.util.Collections;

import java.util.Iterator;

import java.util.List;

import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JButton;

import javax.swing.\*;

public class ShoppingBasket implements ActionListener {

//Now we have the customer's username using the username variable

String username = null;

//Then follow the logic for filling the shopping basket

public ShoppingBasket(String customer\_username) {

this.username = customer\_username;

}

@Override

public void actionPerformed(ActionEvent ae) {

JFrame frame = new JFrame("Shopping Basket");

frame.setSize(270, 310);

frame.setLayout(new BorderLayout());

JPanel panel = new JPanel();

frame.add(panel);

panel.setLayout(new GridLayout(3, 1));

JButton btn1 = new JButton("ADD TO BASKET");

//JButton btn2 = new JButton("REMOVE FROM BASKET");

JButton btn3 = new JButton("PRINT BASKET");

JButton btn4 = new JButton("BACK");

panel.add(btn1);

//panel.add(btn2);

panel.add(btn3);

panel.add(btn4);

//ADD TO SHOPPING BASKET

btn1.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ae) {

JFrame frame1 = new JFrame("Add product");

frame1.setSize(270, 310);

frame1.setLayout(new BorderLayout());

JPanel panel1 = new JPanel();

frame1.add(panel1);

panel1.setLayout(new GridLayout(7, 1));

JLabel label1 = new JLabel("PRODUCT NAME");

JTextField field1 = new JTextField(50);

JLabel label2 = new JLabel("PRODUCT TYPE");

JTextField field2 = new JTextField(25);

JLabel label3 = new JLabel("QUANTITY");

JTextField field3 = new JTextField(25);

JButton button1 = new JButton("CONFIRM");

panel1.add(label1);

panel1.add(field1);

panel1.add(label2);

panel1.add(field2);

panel1.add(label3);

panel1.add(field3);

panel1.add(button1);

//actionListener for the button1

button1.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ae) {

//get form's values

String product\_name = field1.getText();

String product\_type = field2.getText();

String product\_quantity = field3.getText();

//call function for writing to temporary Shopping Basket's files

if (product\_type.equals("Cameras")) {

//try to find those products into customer's catalogue.

//If they exist, then we can add this kind of products into temporary file

shoppingBasketProccessingCamerasProducts(product\_name, product\_type, product\_quantity, username);

}

if (product\_type.equals("Gaming Consoles")) {

//try to find those products into customer's catalogue.

//If they exist, then we can add this kind of products into temporary file

shoppingBasketProccessingConsolesProducts(product\_name, product\_type, product\_quantity, username);

}

if (product\_type.equals("Mobiles")) {

//try to find those products into customer's catalogue.

//If they exist, then we can add this kind of products into temporary file

shoppingBasketProccessingMobilesProducts(product\_name, product\_type, product\_quantity, username);

}

if (product\_type.equals("Personal Computers")) {

//try to find those products into customer's catalogue.

//If they exist, then we can add this kind of products into temporary file

shoppingBasketProccessingPCSProducts(product\_name, product\_type, product\_quantity, username);

}

if (product\_type.equals("TV")) {

//try to find those products into customer's catalogue.

//If they exist, then we can add this kind of products into temporary file

shoppingBasketProccessingTVSProducts(product\_name, product\_type, product\_quantity, username);

}

}

});

frame1.setVisible(true);

frame1.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame1.setResizable(false);

}

});

//REMOVE FROM BASKET

//PRINT BASKET

btn3.addActionListener(new PrintBasket());

//DISPOSE THE FRAME

btn4.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ae) {

frame.dispose();

}

});

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.setResizable(false);

}

public void shoppingBasketProccessingCamerasProducts(String product\_name, String product\_type, String product\_quantity, String username) {

int count\_products = 0;

Scanner file = null;

//create the arrayLists

List<CameraClass> cameras\_list = new ArrayList<>();

//open the original file

try {

file = new Scanner(new File("cameras.txt"));

} catch (FileNotFoundException ex) {

Logger.getLogger(ShoppingBasket.class.getName()).log(Level.SEVERE, null, ex);

}

//reconstruct the arraylist

while (file.hasNext()) {

String line = file.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String cameraModel = stringArr[0];

String cameraMonitorSize = stringArr[1];

String cameraMP = stringArr[2];

String cameraFramesPerSecond = stringArr[3];

String cameraRecordingType = stringArr[4];

String cameraPrice = stringArr[5];

String cameraStoreName = stringArr[6];

CameraClass camera = new CameraClass(cameraModel, cameraMonitorSize, cameraMP, cameraFramesPerSecond, cameraRecordingType, cameraPrice, cameraStoreName);

cameras\_list.add(camera);

}

}

//close the file

file.close();

//count the products by product model using customer's choice

for (CameraClass camera : cameras\_list) {

if (camera.getCamera\_model().contains(product\_name)) {

count\_products++;

}

}

//products stock decisions

int product\_quantity\_integer = Integer.parseInt(product\_quantity);

//the number of products we asked for are in stock

if (product\_quantity\_integer <= count\_products) {

System.out.println("WE ADDED THESE PRODUCTS TO SHOPPING BASKET!");

int product\_count = product\_quantity\_integer;

//consruct a new arrayList which contains the chosen items by the customer

for (CameraClass camera : cameras\_list) {

if (camera.getCamera\_model().contains(product\_name)) {

product\_count--;

CameraClass camera\_to\_add\_basket = new CameraClass(camera.getCamera\_model(), camera.getMonitor\_size(), camera.getMegapixel\_size(), camera.getFrames\_per\_second(), camera.getRecording\_type(), camera.getCamera\_price(), camera.getStore\_name());

//call function for adding the chosen products to file plus using the customer's username as an order id

camera\_to\_add\_basket.setUsername(username);

writeCamerasToFileUsingCustomerUsername(camera\_to\_add\_basket, username);

}

if (product\_count == 0) {

break;

}

}

//the number of products we asked for are too many. The stock has fewer number of products

if (product\_quantity\_integer > count\_products) {

System.out.println("WE DID NOT ADDED THESE PRODUCTS TO SHOPPING BASKET");

}

}

}

public void shoppingBasketProccessingConsolesProducts(String product\_name, String product\_type, String product\_quantity, String username) {

int count\_products = 0;

Scanner file = null;

//create the arrayLists

List<GamingConsoleClass> consoles\_list = new ArrayList<>();

//open the original file

try {

file = new Scanner(new File("gaming\_consoles.txt"));

} catch (FileNotFoundException ex) {

Logger.getLogger(ShoppingBasket.class.getName()).log(Level.SEVERE, null, ex);

}

//reconstruct the arraylist

while (file.hasNext()) {

String line = file.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String consoleName = stringArr[0];

String consoleSize = stringArr[1];

String consoleYearReleased = stringArr[2];

String consoleControllers = stringArr[3];

String consolePrice = stringArr[4];

String consoleStoreName = stringArr[5];

GamingConsoleClass console = new GamingConsoleClass(consoleName, consoleSize, consoleYearReleased, consoleControllers, consolePrice, consoleStoreName);

consoles\_list.add(console);

}

}

//count the products by product model using customer's choice

for (GamingConsoleClass console : consoles\_list) {

if (console.getConsole\_name().contains(product\_name)) {

count\_products++;

}

}

//products stock decisions

int product\_quantity\_integer = Integer.parseInt(product\_quantity);

//close the file

file.close();

//the number of products we asked for are in stock

if (product\_quantity\_integer <= count\_products) {

System.out.println("WE ADDED THESE PRODUCTS TO SHOPPING BASKET!");

int product\_count = product\_quantity\_integer;

//consruct a new arrayList which contains the chosen items by the customer

for (GamingConsoleClass console : consoles\_list) {

if (console.getConsole\_name().contains(product\_name)) {

product\_count--;

GamingConsoleClass console\_to\_add\_basket = new GamingConsoleClass(console.getConsole\_name(), console.getConsole\_size(), console.getConsole\_year\_released(), console.getConsole\_controllers\_count(), console.getConsole\_price(), console.getStore\_name());

//call function for adding the chosen products to file plus using the customer's username as an order id

console\_to\_add\_basket.setUsername(username);

writeConsolesToFileUsingCustomerUsername(console\_to\_add\_basket, username);

}

if (product\_count == 0) {

break;

}

}

//the number of products we asked for are too many. The stock has fewer number of products

if (product\_quantity\_integer > count\_products) {

System.out.println("WE DID NOT ADDED THESE PRODUCTS TO SHOPPING BASKET");

}

}

}

public void shoppingBasketProccessingMobilesProducts(String product\_name, String product\_type, String product\_quantity, String username) {

int count\_products = 0;

Scanner file = null;

//create the arrayLists

List<MobileClass> mobiles\_list = new ArrayList<>();

//open the original file

try {

file = new Scanner(new File("mobiles.txt"));

} catch (FileNotFoundException ex) {

Logger.getLogger(ShoppingBasket.class.getName()).log(Level.SEVERE, null, ex);

}

//reconstruct the arraylist

while (file.hasNext()) {

String line = file.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String mobileModel = stringArr[0];

String mobileFrontCamera = stringArr[1];

String mobileBackCamera = stringArr[2];

String mobileScreenSize = stringArr[3];

String mobileCPU = stringArr[4];

String mobileStorage = stringArr[5];

String mobilePrice = stringArr[6];

String mobileStoreName = stringArr[7];

MobileClass mobile = new MobileClass(mobileModel, mobileFrontCamera, mobileBackCamera, mobileScreenSize, mobileCPU, mobileStorage, mobilePrice, mobileStoreName);

mobiles\_list.add(mobile);

}

}

//count the products by product model using customer's choice

for (MobileClass mobile : mobiles\_list) {

if (mobile.getMobile\_model().contains(product\_name)) {

count\_products++;

}

}

//products stock decisions

int product\_quantity\_integer = Integer.parseInt(product\_quantity);

//close the file

file.close();

//the number of products we asked for are in stock

if (product\_quantity\_integer <= count\_products) {

System.out.println("WE ADDED THESE PRODUCTS TO SHOPPING BASKET!");

int product\_count = product\_quantity\_integer;

//consruct a new arrayList which contains the chosen items by the customer

for (MobileClass mobile : mobiles\_list) {

if (mobile.getMobile\_model().contains(product\_name)) {

product\_count--;

MobileClass mobile\_to\_add\_basket = new MobileClass(mobile.getMobile\_model(), mobile.getFront\_camera\_mpixel(), mobile.getBack\_camera\_mpixel(), mobile.getMobile\_screen\_size(), mobile.getMobile\_cpu(), mobile.getMobile\_storage(), mobile.getMobile\_price(), mobile.getStore\_name());

//call function for adding the chosen products to file plus using the customer's username as an order id

mobile\_to\_add\_basket.setUsername(username);

writeMobilesToFileUsingCustomerUsername(mobile\_to\_add\_basket, username);

}

if (product\_count == 0) {

break;

}

}

//the number of products we asked for are too many. The stock has fewer number of products

if (product\_quantity\_integer > count\_products) {

System.out.println("WE DID NOT ADDED THESE PRODUCTS TO SHOPPING BASKET");

}

}

}

public void shoppingBasketProccessingPCSProducts(String product\_name, String product\_type, String product\_quantity, String username) {

int count\_products = 0;

Scanner file = null;

//create the arrayLists

List<PersonalComputerClass> pc\_list = new ArrayList<>();

//open the original file

try {

file = new Scanner(new File("personalComputers.txt"));

} catch (FileNotFoundException ex) {

Logger.getLogger(ShoppingBasket.class.getName()).log(Level.SEVERE, null, ex);

}

//reconstruct the arraylist

while (file.hasNext()) {

String line = file.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String pcName = stringArr[0];

String pcCpuName = stringArr[1];

String cpuClock = stringArr[2];

String gpuName = stringArr[3];

String pcHardDrive = stringArr[4];

String pcRamSize = stringArr[5];

String pcPrice = stringArr[6];

String pcStoreName = stringArr[7];

PersonalComputerClass pc = new PersonalComputerClass(pcName, pcCpuName, cpuClock, gpuName, pcHardDrive, pcRamSize, pcPrice, pcStoreName);

pc\_list.add(pc);

}

}

//count the products by product model using customer's choice

for (PersonalComputerClass computer : pc\_list) {

if (computer.getModel\_name().contains(product\_name)) {

count\_products++;

}

}

//products stock decisions

int product\_quantity\_integer = Integer.parseInt(product\_quantity);

//close the file

file.close();

//the number of products we asked for are in stock

if (product\_quantity\_integer <= count\_products) {

System.out.println("WE ADDED THESE PRODUCTS TO SHOPPING BASKET!");

int product\_count = product\_quantity\_integer;

//consruct a new arrayList which contains the chosen items by the customer

for (PersonalComputerClass computer : pc\_list) {

if (computer.getModel\_name().contains(product\_name)) {

product\_count--;

PersonalComputerClass pc\_to\_add\_basket = new PersonalComputerClass(computer.getModel\_name(), computer.getPc\_cpu\_name(), computer.getCpu\_clock(), computer.getGraphics\_card\_name(), computer.getPc\_hard\_drive(), computer.getPc\_ram\_size(), computer.getPc\_price(), computer.getPc\_store\_name());

//call function for adding the chosen products to file plus using the customer's username as an order id

pc\_to\_add\_basket.setUsername(username);

writePCSToFileUsingCustomerUsername(pc\_to\_add\_basket, username);

}

if (product\_count == 0) {

break;

}

}

//the number of products we asked for are too many. The stock has fewer number of products

if (product\_quantity\_integer > count\_products) {

System.out.println("WE DID NOT ADDED THESE PRODUCTS TO SHOPPING BASKET");

}

}

}

public void shoppingBasketProccessingTVSProducts(String product\_name, String product\_type, String product\_quantity, String username) {

int count\_products = 0;

Scanner file = null;

//create the arrayLists

List<TVClass> tv\_list = new ArrayList<>();

//open the original file

try {

file = new Scanner(new File("tvs.txt"));

} catch (FileNotFoundException ex) {

Logger.getLogger(ShoppingBasket.class.getName()).log(Level.SEVERE, null, ex);

}

//reconstruct the arraylist

while (file.hasNext()) {

String line = file.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String tvName = stringArr[0];

String tvType = stringArr[1];

String tvYearMade = stringArr[2];

String tvMonitorSize = stringArr[3];

String tvPrice = stringArr[4];

String tvStoreName = stringArr[5];

TVClass tv = new TVClass(tvName, tvType, tvYearMade, tvMonitorSize, tvPrice, tvStoreName);

tv\_list.add(tv);

}

}

//count the products by product model using customer's choice

for (TVClass tv : tv\_list) {

if (tv.getTv\_model().contains(product\_name)) {

count\_products++;

}

}

//products stock decisions

int product\_quantity\_integer = Integer.parseInt(product\_quantity);

//close the file

file.close();

//the number of products we asked for are in stock

if (product\_quantity\_integer <= count\_products) {

System.out.println("WE ADDED THESE PRODUCTS TO SHOPPING BASKET!");

int product\_count = product\_quantity\_integer;

//consruct a new arrayList which contains the chosen items by the customer

for (TVClass tv : tv\_list) {

if (tv.getTv\_model().contains(product\_name)) {

product\_count--;

TVClass tv\_to\_add\_basket = new TVClass(tv.getTv\_model(), tv.getTv\_type(), tv.getTv\_year(), tv.getTv\_monitor\_size(), tv.getTv\_price(), tv.getStore\_name());

//call function for adding the chosen products to file plus using the customer's username as an order id

tv\_to\_add\_basket.setUsername(username);

writeTVSToFileUsingCustomerUsername(tv\_to\_add\_basket, username);

}

if (product\_count == 0) {

break;

}

}

//the number of products we asked for are too many. The stock has fewer number of products

if (product\_quantity\_integer > count\_products) {

System.out.println("WE DID NOT ADDED THESE PRODUCTS TO SHOPPING BASKET");

}

}

}

public void writeCamerasToFileUsingCustomerUsername(CameraClass cameras\_temp\_basket, String username) {

//write to temporary shopping basket.

//customer & admin must agree in order to finish an order

try {

String current = new java.io.File(".").getCanonicalPath();

File file = new File(current + "\\cameras\_temp\_basket.txt");

if (!file.exists()) {

file.createNewFile();

}

FileWriter fw = new FileWriter(file, true);

BufferedWriter bw = new BufferedWriter(fw);

bw.write("\n" + cameras\_temp\_basket.getUsername() + "\t" + cameras\_temp\_basket.toString());

bw.close();

} catch (IOException e) {

e.printStackTrace();

}

}

public void writeConsolesToFileUsingCustomerUsername(GamingConsoleClass consoles\_temp\_basket, String username) {

//write to temporary shopping basket.

//customer & admin must agree in order to finish an order

try {

String current = new java.io.File(".").getCanonicalPath();

File file = new File(current + "\\consoles\_temp\_basket.txt");

if (!file.exists()) {

file.createNewFile();

}

FileWriter fw = new FileWriter(file, true);

BufferedWriter bw = new BufferedWriter(fw);

bw.write("\n" + consoles\_temp\_basket.getUsername() + "\t" + consoles\_temp\_basket.toString());

bw.close();

} catch (IOException e) {

e.printStackTrace();

}

}

public void writeMobilesToFileUsingCustomerUsername(MobileClass mobiles\_temp\_basket, String username) {

//write to temporary shopping basket.

//customer & admin must agree in order to finish an order

try {

String current = new java.io.File(".").getCanonicalPath();

File file = new File(current + "\\mobiles\_temp\_file.txt");

if (!file.exists()) {

file.createNewFile();

}

FileWriter fw = new FileWriter(file, true);

BufferedWriter bw = new BufferedWriter(fw);

bw.write("\n" + mobiles\_temp\_basket.getUsername() + "\t" + mobiles\_temp\_basket.toString());

bw.close();

} catch (IOException e) {

e.printStackTrace();

}

}

public void writePCSToFileUsingCustomerUsername(PersonalComputerClass pcs\_temp\_basket, String username) {

//write to temporary shopping basket.

//customer & admin must agree in order to finish an order

try {

String current = new java.io.File(".").getCanonicalPath();

File file = new File(current + "\\pcs\_temp\_file.txt");

if (!file.exists()) {

file.createNewFile();

}

FileWriter fw = new FileWriter(file, true);

BufferedWriter bw = new BufferedWriter(fw);

bw.write("\n" + pcs\_temp\_basket.getUsername() + "\t" + pcs\_temp\_basket.toString());

bw.close();

} catch (IOException e) {

e.printStackTrace();

}

}

public void writeTVSToFileUsingCustomerUsername(TVClass tvs\_temp\_basket, String username) {

//write to temporary shopping basket.

//customer & admin must agree in order to finish an order

try {

String current = new java.io.File(".").getCanonicalPath();

File file = new File(current + "\\tvs\_temp\_file.txt");

if (!file.exists()) {

file.createNewFile();

}

FileWriter fw = new FileWriter(file, true);

BufferedWriter bw = new BufferedWriter(fw);

bw.write("\n" + tvs\_temp\_basket.getUsername() + "\t" + tvs\_temp\_basket.toString());

bw.close();

} catch (IOException e) {

e.printStackTrace();

}

}

}

**Stavroulakis\_Vasilis\_Thesis\_2018.java**

package stavroulakis\_vasilis\_thesis\_2018;

import java.awt.BorderLayout;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.\*;

import java.util.\*;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JPasswordField;

import javax.swing.JTextField;

public class Stavroulakis\_Vasilis\_Thesis\_2018 {

public static void main(String[] args) {

JFrame frame = new JFrame("EshopPanel");

frame.setSize(250, 300);

frame.setLayout(new BorderLayout());

JPanel panel = new JPanel();

frame.add(panel);

panel.setLayout(new GridLayout(6, 1));

JLabel label1 = new JLabel("Username");

JLabel label2 = new JLabel("Password");

JTextField text1 = new JTextField(20);

JPasswordField text2 = new JPasswordField(16);

JButton login = new JButton("LOGIN");

JButton register = new JButton("REGISTER");

panel.add(label1);

panel.add(text1);

panel.add(label2);

panel.add(text2);

panel.add(login);

panel.add(register);

register.addActionListener(new RegisterCustomer());

//login actionListener by getting login values

login.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent ae) {

String a1 = text1.getText();

String a2 = text2.getText();

try {

checkUser(a1, a2);

} catch (FileNotFoundException ex) {

Logger.getLogger(Stavroulakis\_Vasilis\_Thesis\_2018.class.getName()).log(Level.SEVERE, null, ex);

} catch (IOException ex) {

Logger.getLogger(Stavroulakis\_Vasilis\_Thesis\_2018.class.getName()).log(Level.SEVERE, null, ex);

}

}

});

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setResizable(false);

}

public static void checkUser(String a1, String a2) throws FileNotFoundException, IOException {

//admin's credentials username: vasilis, password: 12345

String admin\_username = "vasilis";

String admin\_password = "12345";

List<RegisterClass> customers\_list = new ArrayList<>();

Scanner file;

file = new Scanner(new File("customers.txt"));

while (file.hasNext()) {

String line = file.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String fname = stringArr[0];

String surname = stringArr[1];

String address = stringArr[2];;

String city = stringArr[3];

String tk = stringArr[4];

String afm = stringArr[5];

String username = stringArr[6];

String password = stringArr[7];

//toys prosthetoyme sthn lista me oloys tous users toy eshop

RegisterClass customer = new RegisterClass(fname, surname, address, city, tk, afm, username, password);

customers\_list.add(customer);

}

}

//checking user type

Log.f("STAVROULAKIS VASILIS THESIS 2018");

for (RegisterClass num : customers\_list) {

if (a1.equals(num.getUsername()) & a2.equals(num.getPassword())) {

Log.i("YOU LOGGED AS CUSTOMER");

//System.out.println("YOU LOGGED AS CUSTOMER");

//For shopping basket's purposes we must know and pass customer's username

CustomerMainPanel.customerMainPanelMethod(a1);

}

}

if (a1.equals(admin\_username) && a2.equals(admin\_password)) {

//There is an only one admin, so we dont need its username

Log.i("YOU LOGGED AS ADMIN");

//System.out.println("YOU LOGGED AS ADMIN");

AdminMainPanel.adminMainPanelMethod();

}

file.close();

}

}

**TVClass.java**

package stavroulakis\_vasilis\_thesis\_2018;

public class TVClass {

String tv\_model;

String tv\_type;

String tv\_year;

String tv\_monitor\_size;

String tv\_price;

String store\_name;

String username;

public TVClass(String tv\_model, String tv\_type, String tv\_year, String tv\_monitor\_size, String tv\_price, String store\_name) {

this.tv\_model = tv\_model;

this.tv\_type = tv\_type;

this.tv\_year = tv\_year;

this.tv\_monitor\_size = tv\_monitor\_size;

this.tv\_price = tv\_price;

this.store\_name = store\_name;

}

public TVClass(String username, String tv\_model, String tv\_type, String tv\_year, String tv\_monitor\_size, String tv\_price, String store\_name) {

this.username = username;

this.tv\_model = tv\_model;

this.tv\_type = tv\_type;

this.tv\_year = tv\_year;

this.tv\_monitor\_size = tv\_monitor\_size;

this.tv\_price = tv\_price;

this.store\_name = store\_name;

this.username = username;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getStore\_name() {

return store\_name;

}

public void setStore\_name(String store\_name) {

this.store\_name = store\_name;

}

@Override

public String toString() {

return this.getTv\_model() + "\t" + this.getTv\_type() + "\t" + this.getTv\_year() + "\t" + this.getTv\_monitor\_size() + "\t" + this.getTv\_price() + "\t" + this.getStore\_name() + "\n";

}

public String getTv\_model() {

return tv\_model;

}

public void setTv\_model(String tv\_model) {

this.tv\_model = tv\_model;

}

public String getTv\_type() {

return tv\_type;

}

public void setTv\_type(String tv\_type) {

this.tv\_type = tv\_type;

}

public String getTv\_year() {

return tv\_year;

}

public void setTv\_year(String tv\_year) {

this.tv\_year = tv\_year;

}

public String getTv\_monitor\_size() {

return tv\_monitor\_size;

}

public void setTv\_monitor\_size(String tv\_monitor\_size) {

this.tv\_monitor\_size = tv\_monitor\_size;

}

public String getTv\_price() {

return tv\_price;

}

public void setTv\_price(String tv\_price) {

this.tv\_price = tv\_price;

}

}

**printCameras.java**

package stavroulakis\_vasilis\_thesis\_2018;

import java.awt.BorderLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.File;

import java.io.FileNotFoundException;

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JFrame;

import javax.swing.JScrollPane;

import javax.swing.JTextArea;

public class printCameras implements ActionListener {

@Override

public void actionPerformed(ActionEvent ae) {

List<CameraClass> cameras\_list = new ArrayList<>();

Scanner file1 = null;

try {

file1 = new Scanner(new File("cameras.txt"));

} catch (FileNotFoundException ex) {

Logger.getLogger(printCameras.class.getName()).log(Level.SEVERE, null, ex);

}

while (file1.hasNext()) {

String line = file1.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String cameraModel = stringArr[0];

String cameraMonitorSize = stringArr[1];

String cameraMP = stringArr[2];

String cameraFramesPerSecond = stringArr[3];

String cameraRecordingType = stringArr[4];

String cameraPrice = stringArr[5];

String cameraStoreName = stringArr[6];

CameraClass camera = new CameraClass(cameraModel, cameraMonitorSize, cameraMP, cameraFramesPerSecond, cameraRecordingType, cameraPrice, cameraStoreName);

cameras\_list.add(camera);

}

}

JFrame frame = new JFrame("Print Cameras");

frame.setLayout(new BorderLayout());

frame.setSize(800, 500);

JTextArea textArea = new JTextArea();

frame.add(textArea);

JScrollPane sp = new JScrollPane(textArea);

textArea.setEditable(false);

JScrollPane scroll = new JScrollPane(textArea,

JScrollPane.VERTICAL\_SCROLLBAR\_ALWAYS, JScrollPane.HORIZONTAL\_SCROLLBAR\_ALWAYS);

frame.add(scroll);

textArea.append("-----------------------------------CAMERA PRODUCTS-----------------------------------");

textArea.append("\n");

for (CameraClass c : cameras\_list) {

textArea.append("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" + "\n");

textArea.append("Camera Model : " + c.getCamera\_model() + "\n");

textArea.append("Camera Monitor Size : " + c.getMonitor\_size() + "\n");

textArea.append("Camera Megapixels : " + c.getMegapixel\_size() + "\n");

textArea.append("Camera Frames/Sec : " + c.getFrames\_per\_second() + "\n");

textArea.append("Camera Recording Type : " + c.getRecording\_type() + "\n");

textArea.append("Camera Price : " + c.getCamera\_price() + "\n");

textArea.append("Camera Store Name : " + c.getStore\_name() + "\n");

textArea.append("\n");

}

file1.close();

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.setResizable(false);

}

}

**printGamingConsoles.java**

package stavroulakis\_vasilis\_thesis\_2018;

import java.awt.BorderLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.File;

import java.io.FileNotFoundException;

import java.util.ArrayList;

import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JFrame;

import javax.swing.JScrollPane;

import javax.swing.JTextArea;

public class printGamingConsoles implements ActionListener {

@Override

public void actionPerformed(ActionEvent ae) {

ArrayList<GamingConsoleClass> gamingConsoles\_\_list = new ArrayList<>();

Scanner file1 = null;

try {

file1 = new Scanner(new File("gaming\_consoles.txt"));

} catch (FileNotFoundException ex) {

Logger.getLogger(printGamingConsoles.class.getName()).log(Level.SEVERE, null, ex);

}

while (file1.hasNext()) {

String line = file1.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String consoleModel = stringArr[0];

String consoleSize = stringArr[1];

String consoleYearReleased = stringArr[2];

String consoleControlllers = stringArr[3];

String consolePrice = stringArr[4];

String consoleStoreName = stringArr[5];

GamingConsoleClass console = new GamingConsoleClass(consoleModel, consoleSize, consoleYearReleased, consoleControlllers, consolePrice, consoleStoreName);

gamingConsoles\_\_list.add(console);

}

}

JFrame frame = new JFrame("Print Consoles");

frame.setLayout(new BorderLayout());

frame.setSize(800, 500);

JTextArea textArea = new JTextArea();

frame.add(textArea);

JScrollPane sp = new JScrollPane(textArea);

textArea.setEditable(false);

JScrollPane scroll = new JScrollPane(textArea,

JScrollPane.VERTICAL\_SCROLLBAR\_ALWAYS, JScrollPane.HORIZONTAL\_SCROLLBAR\_ALWAYS);

frame.add(scroll);

textArea.append("-----------------------------------CONSOLE PRODUCTS-----------------------------------");

textArea.append("\n");

for (GamingConsoleClass gm : gamingConsoles\_\_list) {

textArea.append("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" + "\n");

textArea.append("Console Model : " + gm.getConsole\_name() + "\n");

textArea.append("Console HD Size : " + gm.getConsole\_size() + "\n");

textArea.append("Console Controllers : " + gm.getConsole\_controllers\_count() + "\n");

textArea.append("Console Price : " + gm.getConsole\_price() + "\n");

textArea.append("Console Store Name : " + gm.getStore\_name() + "\n");

textArea.append("\n");

}

file1.close();

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.setResizable(false);

}

}

**printMobiles.java**

package stavroulakis\_vasilis\_thesis\_2018;

import java.awt.BorderLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.File;

import java.io.FileNotFoundException;

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JFrame;

import javax.swing.JScrollPane;

import javax.swing.JTextArea;

public class printMobiles implements ActionListener {

@Override

public void actionPerformed(ActionEvent ae) {

List<MobileClass> mobiles\_list = new ArrayList<>();

Scanner file1 = null;

try {

file1 = new Scanner(new File("mobiles.txt"));

} catch (FileNotFoundException ex) {

Logger.getLogger(printMobiles.class.getName()).log(Level.SEVERE, null, ex);

}

while (file1.hasNext()) {

String line = file1.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String mobileModel = stringArr[0];

String mobileFrontCamera = stringArr[1];

String mobileBackCamera = stringArr[2];

String mobileScreenSize = stringArr[3];

String mobileCPU = stringArr[4];

String mobileStorage = stringArr[5];

String mobilePrice = stringArr[6];

String mobileStoreName = stringArr[7];

MobileClass mobile = new MobileClass(mobileModel, mobileFrontCamera, mobileBackCamera, mobileScreenSize, mobileCPU, mobileStorage, mobilePrice, mobileStoreName);

mobiles\_list.add(mobile);

}

}

JFrame frame = new JFrame("Print Mobiles");

frame.setLayout(new BorderLayout());

frame.setSize(800, 500);

JTextArea textArea = new JTextArea();

frame.add(textArea);

JScrollPane sp = new JScrollPane(textArea);

textArea.setEditable(false);

JScrollPane scroll = new JScrollPane(textArea,

JScrollPane.VERTICAL\_SCROLLBAR\_ALWAYS, JScrollPane.HORIZONTAL\_SCROLLBAR\_ALWAYS);

frame.add(scroll);

textArea.append("-----------------------------------MOBILES-----------------------------------");

textArea.append("\n");

for (MobileClass m : mobiles\_list) {

textArea.append("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" + "\n");

textArea.append("Mobile Model : " + m.getMobile\_model() + "\n");

textArea.append("Mobile Front Camera Size : " + m.getFront\_camera\_mpixel() + "\n");

textArea.append("Mobile Back Camera Size : " + m.getBack\_camera\_mpixel() + "\n");

textArea.append("Mobile Screen Size : " + m.getMobile\_screen\_size() + "\n");

textArea.append("Mobile CPU Clock : " + m.getMobile\_cpu() + "\n");

textArea.append("Mobile Storage : " + m.getMobile\_storage() + "\n");

textArea.append("Mobile Price : " + m.getMobile\_price() + "\n");

textArea.append("Mobile Store Name : " + m.getStore\_name() + "\n");

textArea.append("\n");

}

file1.close();

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.setResizable(false);

}

}

**printPersonalComputers.java**

package stavroulakis\_vasilis\_thesis\_2018;

import java.awt.BorderLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.File;

import java.io.FileNotFoundException;

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JFrame;

import javax.swing.JScrollPane;

import javax.swing.JTextArea;

public class printPersonalComputers implements ActionListener {

@Override

public void actionPerformed(ActionEvent ae) {

List<PersonalComputerClass> personalComputers\_list = new ArrayList<>();

Scanner file1 = null;

try {

file1 = new Scanner(new File("personalComputers.txt"));

} catch (FileNotFoundException ex) {

Logger.getLogger(printPersonalComputers.class.getName()).log(Level.SEVERE, null, ex);

}

while (file1.hasNext()) {

String line = file1.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String pcModelName = stringArr[0];

String pcCpuName = stringArr[1];

String pcClock = stringArr[2];

String pcGPUname = stringArr[3];

String pcHardDrive = stringArr[4];

String pcRamSize = stringArr[5];

String pcPrice = stringArr[4];

String pcStoreName = stringArr[5];

PersonalComputerClass pc = new PersonalComputerClass(pcModelName, pcCpuName, pcClock, pcGPUname, pcHardDrive, pcRamSize, pcPrice, pcStoreName);

personalComputers\_list.add(pc);

}

}

JFrame frame = new JFrame("Print PCS");

frame.setLayout(new BorderLayout());

frame.setSize(800, 500);

JTextArea textArea = new JTextArea();

frame.add(textArea);

JScrollPane sp = new JScrollPane(textArea);

textArea.setEditable(false);

JScrollPane scroll = new JScrollPane(textArea,

JScrollPane.VERTICAL\_SCROLLBAR\_ALWAYS, JScrollPane.HORIZONTAL\_SCROLLBAR\_ALWAYS);

frame.add(scroll);

textArea.append("-----------------------------------PERSONAL COMPUTERS-----------------------------------");

textArea.append("\n");

for (PersonalComputerClass pc : personalComputers\_list) {

textArea.append("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" + "\n");

textArea.append("PC Model : " + pc.getModel\_name() + "\n");

textArea.append("PC CPU Name : " + pc.getPc\_cpu\_name() + "\n");

textArea.append("PC CPU Clock : " + pc.getCpu\_clock() + "\n");

textArea.append("PC GPU Name : " + pc.getGraphics\_card\_name() + "\n");

textArea.append("PC Hard Drive Size : " + pc.getPc\_hard\_drive() + "\n");

textArea.append("PC RAM Size : " + pc.getPc\_ram\_size() + "\n");

textArea.append("PC Price : " + pc.getPc\_price() + "\n");

textArea.append("PC Store Name : " + pc.getPc\_store\_name() + "\n");

textArea.append("\n");

}

file1.close();

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.setResizable(false);

}

}

**printTVS.java**

package stavroulakis\_vasilis\_thesis\_2018;

import java.awt.BorderLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.File;

import java.io.FileNotFoundException;

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JFrame;

import javax.swing.JScrollPane;

import javax.swing.JTextArea;

public class printTVS implements ActionListener {

@Override

public void actionPerformed(ActionEvent ae) {

List<TVClass> tvs\_list = new ArrayList<>();

Scanner file1 = null;

try {

file1 = new Scanner(new File("tvs.txt"));

} catch (FileNotFoundException ex) {

Logger.getLogger(printTVS.class.getName()).log(Level.SEVERE, null, ex);

}

while (file1.hasNext()) {

String line = file1.nextLine();

if (!line.isEmpty()) {

String[] stringArr = line.split("\t");

String tvModelName = stringArr[0];

String tvType = stringArr[1];

String tvYear = stringArr[2];

String tvMonitorSize = stringArr[3];

String tvPrice = stringArr[4];

String tvStoreName = stringArr[5];

TVClass tv = new TVClass(tvModelName, tvType, tvYear, tvMonitorSize, tvPrice, tvStoreName);

tvs\_list.add(tv);

}

}

JFrame frame = new JFrame("Print TVS");

frame.setLayout(new BorderLayout());

frame.setSize(800, 500);

JTextArea textArea = new JTextArea();

frame.add(textArea);

JScrollPane sp = new JScrollPane(textArea);

textArea.setEditable(false);

JScrollPane scroll = new JScrollPane(textArea,

JScrollPane.VERTICAL\_SCROLLBAR\_ALWAYS, JScrollPane.HORIZONTAL\_SCROLLBAR\_ALWAYS);

frame.add(scroll);

textArea.append("-----------------------------------TELEVISIONS-----------------------------------");

textArea.append("\n");

for (TVClass tv : tvs\_list) {

textArea.append("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" + "\n");

textArea.append("TV Model : " + tv.getTv\_model() + "\n");

textArea.append("TV Type : " + tv.getTv\_type() + "\n");

textArea.append("TV Year of Release: " + tv.getTv\_year() + "\n");

textArea.append("TV Monitor Size : " + tv.getTv\_monitor\_size() + "\n");

textArea.append("TV Price : " + tv.getTv\_price() + "\n");

textArea.append("PC Store Name : " + tv.getStore\_name() + "\n");

textArea.append("\n");

}

file1.close();

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.setResizable(false);

}

}