package VendingMachine;

import java.io.BufferedReader;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.util.ArrayList;

class User {

private String username;

private float credit;

private String password;

public User(String username, float credit, String password) {

this.username = username;

this.credit = credit;

this.password = password;

}

public String getUsername() {

return username;

}

public float getCredit() {

return credit;

}

public void setCredit(float credit) {

this.credit = credit;

}

public String getPassword() {

return password;

}

}

class Admin {

private String username;

private String password;

public Admin(String username, String password) {

this.username = username;

this.password = password;

}

public String getUsername() {

return username;

}

public String getPassword() {

return password;

}

}

class Product {

private String description;

private float price;

private String location;

private int quantity;

private String currency = "€"; // added field for currency

public Product(String description, float price, String location, int quantity) {

this.description = description;

this.price = price;

this.location = location;

this.quantity = quantity;

}

public String getDescription() {

return description;

}

public float getPrice() {

return price;

}

public String getLocation() {

return location;

}

public int getQuantity() {

return quantity;

}

public void setQuantity(int quantity) {

this.quantity = quantity;

}

public String getCurrency() {

return currency;

}

// added method to display price with currency symbol

public String getPriceFormatted() {

return currency + price;

}

}

class VendingMachine {

private ArrayList<User> users = new ArrayList<User>();

private ArrayList<Product> products = new ArrayList<Product>();

private ArrayList<Admin> admins = new ArrayList<Admin>();

private User currentUser;

private Admin currentAdmin;

public void loadUsers(String filename) throws IOException {

BufferedReader br = new BufferedReader(new FileReader("src\\clients.dat.csv"));

System.out.println("Reading users...");

String line;

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

String[] cells = line.split(",");

String username = cells[0];

float credit = Float.parseFloat(cells[1]);

String password = cells[2];

users.add(new User(username, credit, password));

}

br.close();

}

public void loadAdmins(String filename) throws IOException {

BufferedReader br = new BufferedReader(new FileReader("src\\admins.dat.csv"));

System.out.println("Reading Admins...");

String line;

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

String[] cells = line.split(",");

String username = cells[0];

String password = cells[1];

admins.add(new Admin(username, password));

}

br.close();

}

public void reloadProducts(String filename) throws IOException {

BufferedReader br = new BufferedReader(new FileReader("src\\product.dat.csv"));

System.out.println("reading product...");

String line;

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

String[] cells = line.split(",");

String description = cells[0];

float price = Float.parseFloat(cells[1]);

String location = cells[2];

int quantity = Integer.parseInt(cells[3]);

products.add(new Product(description, price, location, quantity));

}

br.close();

}

public void saveProducts(String filename) throws IOException {

FileWriter fw = new FileWriter("src\\product.dat.csv");

for (Product product : products) {

fw.write(product.getDescription() + "," + product.getPrice() + "," + product.getLocation() + "," + product.getQuantity() + "\n");

}

fw.close();

}

public boolean login(String username, String password) {

for (User user : users) {

if (user.getUsername().equals(username) && user.getPassword().equals(password)) {

currentUser = user;

return true;

}

}

return false;

}

public boolean loginAdmin(String username, String password) {

for (Admin admin : admins) {

if (admin.getUsername().equals(username) && admin.getPassword().equals(password)) {

currentAdmin = admin;

return true;

}

}

return false;

}

public void logout() {

currentUser = null;

}

public boolean buyProduct(String location) {

if (currentUser == null) {

return false;

}

Product productToBuy = null;

for (Product product : products) {

if (product.getLocation().equals(location)) {

productToBuy = product;

break;

}

}

if (productToBuy == null || productToBuy.getQuantity() == 0 || currentUser.getCredit() < productToBuy.getPrice()) {

return false;

}

productToBuy.setQuantity(productToBuy.getQuantity() - 1);

currentUser.setCredit(currentUser.getCredit() - productToBuy.getPrice());

return true;

}

public void addProduct(String description, float price, String location, int quantity) {

products.add(new Product(description, price, location, quantity));

}

public void reloadProduct(String location, int quantity) {

for (Product product : products) {

if (product.getLocation().equals(location)) {

product.setQuantity(product.getQuantity() + quantity);

break;

}

}

}

public void shutdown(String filename) throws IOException {

saveProducts(filename);

}

}

class VendingMachineApp {

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

VendingMachine vendingMachine = new VendingMachine();

vendingMachine.loadUsers("src\\clients.dat.csv");

vendingMachine.loadAdmins("src\\admins.dat.csv");

vendingMachine.reloadProducts("src\\product.dat.csv");

Admin admin = new Admin("JohnT", "125648");

// login as admin

System.out.println("Logging in as admin...");

boolean adminLoggedIn = vendingMachine.loginAdmin(admin.getUsername(), admin.getPassword());

if (!adminLoggedIn) {

System.out.println("Invalid Admin username or password.");

return;

}

// add a product

System.out.println("Adding a product...");

vendingMachine.addProduct("Snickers", 2, "A1", 10);

// reload a product

System.out.println("Reloading a product...");

vendingMachine.reloadProduct("A1", 5);

// logout admin

System.out.println("Logging out admin...");

vendingMachine.logout();

// login as user

System.out.println("Logging in as user...");

boolean userLoggedIn = vendingMachine.login("Rebecca", "685439");

if (!userLoggedIn) {

System.out.println("Invalid username or password");

return;

}

// buy a product

System.out.println("Buying a product...");

boolean productBought = vendingMachine.buyProduct("A1");

if (productBought) {

System.out.println("Product bought successfully.");

} else {

System.out.println("Failed to buy product.");

}

// logout user

System.out.println("Logging out user...");

vendingMachine.logout();

// shutdown vending machine

System.out.println("Shutting down vending machine...");

vendingMachine.shutdown("src\\product.dat.csv");

}

}