UNYT Logo

**University of New York Tirana**

**Project: Customer Relationship Management System**

Project Notes

**There are two main classes in the project.** The first is the class **Run** which serves to test the Graphical User Interface (GUI), while the second is **CRMTest** which serves for testing the Console version of the program.

At the beginning, it will require you to log in. On the console version you need only the username and password. The credentials are:

Username: Danja

Password: 1234

On the GUI version it will require an additional field. The manager PIN. Therefore the credentials needed to login are:

Username: Danja

Password: 1234

PIN: 123

Contents

[Project Notes 2](#_Toc536047600)

[1 System Design review 3](#_Toc536047601)

[2 UML Class Diagram 4](#_Toc536047602)

[3 Include the code for each class 5](#_Toc536047603)

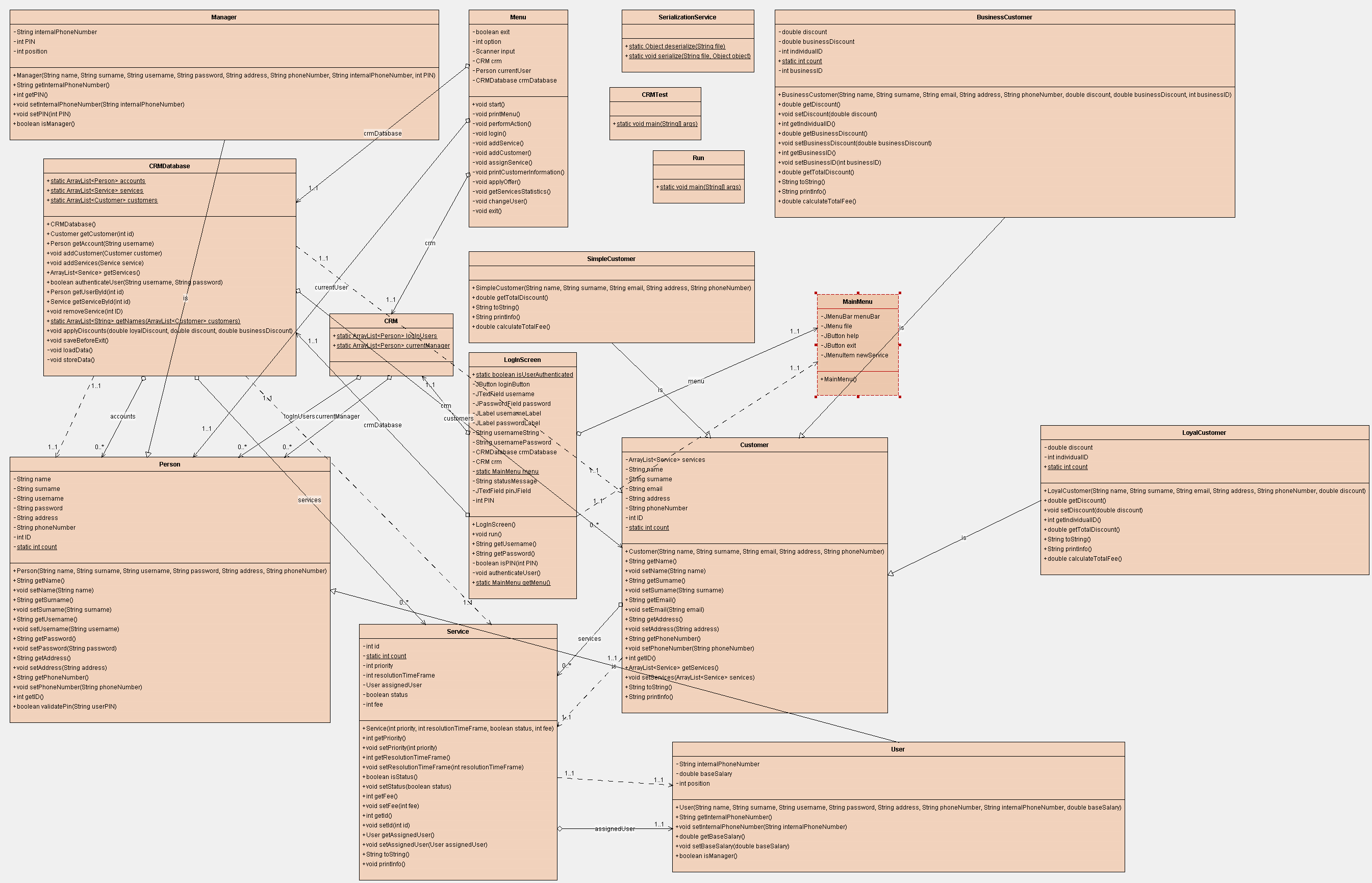
[4 Conclusion 32](#_Toc536047604)

# System Design review

Polymorphism is used on the program. Classes SimpleCustomer, LoyalCustomer and BusinessCustomer all extend the class Customer. Also the Classes User and Manager extend from the same class, the class Person and have additional properties and methods. There are two main classes, one for the console version, and the other for the GUI version. The two version works separately, and even the Serialize service for each is different. For the console version it uses the SerializeService class, while for the GUI version, the process of saving and loading data happens inside the MainMenu class.

Also, the menu for the console and GUI version are separate. The class Menu corresponds to the menu of the Console version, while the class MainMenu corresponds to the GUI version of the program. Classes are divided into 3 packages. Entities, CRM and GUI to make it easier to differ them. Classes CRM and CRMDatabase act as a pseudo Database to hold information. CRM holds information about the logged in user, while CRMDatabase holds all the services and customers created.

# UML Class Diagram



# Include the code for each class

**CLASS: BusinessCustomer**

package entities;

public class BusinessCustomer extends Customer{

private double discount;

private double businessDiscount;

private int individualID;

public static int count = 0;

private int businessID;

public BusinessCustomer (String name, String surname, String email, String address, String phoneNumber, double discount, double businessDiscount, int businessID) {

super(name, surname, email, address, phoneNumber);

this.discount = discount;

this.businessDiscount = businessDiscount;

this.businessID = businessID;

individualID = ++count;

}

public double getDiscount() {

return discount;

}

public void setDiscount(double discount) {

this.discount = discount;

}

public int getIndividualID() {

return individualID;

}

public double getBusinessDiscount() {

return businessDiscount;

}

public void setBusinessDiscount(double businessDiscount) {

this.businessDiscount = businessDiscount;

}

public int getBusinessID() {

return businessID;

}

public void setBusinessID(int businessID) {

this.businessID = businessID;

}

public double getTotalDiscount(){

return (discount + businessDiscount);

}

@Override

public String toString(){

return super.toString() + " / Individual ID: " + individualID

+ " / Business ID: " + businessID

+ " / Total Discount: " + getTotalDiscount()

+ " / Type: Business Costumer";

}

@Override

public String printInfo(){

return toString();

}

public double calculateTotalFee(){

double totalFee = 0;

for(Service serv : super.getServices()){

totalFee += serv.getFee() \* getTotalDiscount();

}

return totalFee;

}

}

**CLASS: Customer**

package entities;

import java.util.\*;

public class Customer {

private ArrayList<Service> services = new ArrayList();

private String name;

private String surname;

private String email;

private String address;

private String phoneNumber;

private int ID;

private static int count = 0;

public Customer (String name, String surname, String email, String address, String phoneNumber) {

this.name = name;

this.surname = surname;

this.email = email;

this.address = address;

this.phoneNumber = phoneNumber;

ID = ++count;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getSurname() {

return surname;

}

public void setSurname(String surname) {

this.surname = surname;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

public String getPhoneNumber() {

return phoneNumber;

}

public void setPhoneNumber(String phoneNumber) {

this.phoneNumber = phoneNumber;

}

public int getID() {

return ID;

}

public ArrayList<Service> getServices() {

return services;

}

public void setServices(ArrayList<Service> services) {

this.services = services;

}

@Override

public String toString(){

return ("Customer (" + ID + "): " + name + " " + surname

+ " / email " + email

+ " / address " + address

+ " / phone number " + phoneNumber);

}

public String printInfo(){

return toString();

}

}

**CLASS: LoyalCustomer**

package entities;

public class LoyalCustomer extends Customer{

private double discount;

private int individualID;

public static int count = 0;

public LoyalCustomer (String name, String surname, String email, String address, String phoneNumber, double discount) {

super(name, surname, email, address, phoneNumber);

this.discount = discount;

individualID = ++count;

}

public double getDiscount() {

return discount;

}

public void setDiscount(double discount) {

this.discount = discount;

}

public int getIndividualID() {

return individualID;

}

public double getTotalDiscount(){

return this.getDiscount();

}

@Override

public String toString(){

return super.toString() + " / Individual ID: " + individualID

+ " / Discount: " + discount

+ " / Type: Loyal Costumer";

}

@Override

public String printInfo(){

return toString();

}

public double calculateTotalFee(){

double totalFee = 0;

for(Service serv : super.getServices()){

totalFee += serv.getFee() \* getTotalDiscount();

}

return totalFee;

}

}

**CLASS: Manager**

package entities;

public class Manager extends Person {

private String internalPhoneNumber;

private int PIN;

private final int position = 0;

public Manager(String name, String surname, String username, String password, String address, String phoneNumber, String internalPhoneNumber, int PIN) {

super(name, surname, username, password, address, phoneNumber);

this.internalPhoneNumber = internalPhoneNumber;

this.PIN = PIN;

}

public String getInternalPhoneNumber(){

return internalPhoneNumber;

}

public int getPIN(){

return PIN;

}

public void setInternalPhoneNumber(String internalPhoneNumber){

this.internalPhoneNumber = internalPhoneNumber;

}

public void setPIN(int PIN){

this.PIN = PIN;

}

public boolean isManager(){

if(this.position == 0){

return true;

}

return false;

}

}

**CLASS: Manager**

package entities;

public class Manager extends Person {

private String internalPhoneNumber;

private int PIN;

private final int position = 0;

public Manager(String name, String surname, String username, String password, String address, String phoneNumber, String internalPhoneNumber, int PIN) {

super(name, surname, username, password, address, phoneNumber);

this.internalPhoneNumber = internalPhoneNumber;

this.PIN = PIN;

}

public String getInternalPhoneNumber(){

return internalPhoneNumber;

}

public int getPIN(){

return PIN;

}

public void setInternalPhoneNumber(String internalPhoneNumber){

this.internalPhoneNumber = internalPhoneNumber;

}

public void setPIN(int PIN){

this.PIN = PIN;

}

public boolean isManager(){

if(this.position == 0){

return true;

}

return false;

}

}

**CLASS: Person**

package entities;

public class Person {

private String name;

private String surname;

private String username;

private String password;

private String address;

private String phoneNumber;

private int ID;

private static int count = 0;

public Person(String name, String surname, String username, String password, String address, String phoneNumber) {

this.name = name;

this.surname = surname;

this.username = username;

this.password = password;

this.address = address;

this.phoneNumber = phoneNumber;

ID = ++count;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getSurname() {

return surname;

}

public void setSurname(String surname) {

this.surname = surname;

}

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;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

public String getPhoneNumber() {

return phoneNumber;

}

public void setPhoneNumber(String phoneNumber) {

this.phoneNumber = phoneNumber;

}

public int getID() {

return ID;

}

public boolean validatePin(String userPIN) {

if(userPIN.equals(password)) {

return true;

}else{

return false;

}

}

}

**CLASS: Service**

package entities;

public class Service {

private int id;

private static int count = 0;

private int priority;

private int resolutionTimeFrame;

private User assignedUser;

private boolean status;

private int fee;

public Service(int priority, int resolutionTimeFrame, boolean status, int fee) {

this.priority = priority;

this.resolutionTimeFrame = resolutionTimeFrame;

this.status = status;

this.fee = fee;

id = ++count;

}

public int getPriority() {

return priority;

}

public void setPriority(int priority) {

this.priority = priority;

}

public int getResolutionTimeFrame() {

return resolutionTimeFrame;

}

public void setResolutionTimeFrame(int resolutionTimeFrame) {

this.resolutionTimeFrame = resolutionTimeFrame;

}

public boolean isStatus() {

return status;

}

public void setStatus(boolean status) {

this.status = status;

}

public int getFee() {

return fee;

}

public void setFee(int fee) {

this.fee = fee;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public User getAssignedUser() {

return assignedUser;

}

public void setAssignedUser(User assignedUser) {

this.assignedUser = assignedUser;

}

@Override

public String toString() {

return "Service{" +

"Priority=" + priority +

", resolutionTimeFrame='" + resolutionTimeFrame + '\'' +

", fee=" + fee +

", status=" + status;

}

public void printInfo(){

System.out.println(toString());

}

}

**CLASS: SimpleCustomer**

package entities;

public class SimpleCustomer extends Customer {

public SimpleCustomer (String name, String surname, String email, String address, String phoneNumber) {

super(name, surname, email, address, phoneNumber);

}

public double getTotalDiscount(){

return 0;

}

@Override

public String toString(){

return super.toString() + " / Type: Simple Costumer";

}

@Override

public String printInfo(){

return toString();

}

public double calculateTotalFee(){

double totalFee = 0;

for(Service serv : super.getServices()){

totalFee += serv.getFee() \* getTotalDiscount();

}

return totalFee;

}

}

**CLASS: User**

package entities;

public class User extends Person{

private String internalPhoneNumber;

private double baseSalary;

private final int position = 1;

public User(String name, String surname, String username, String password, String address, String phoneNumber, String internalPhoneNumber, double baseSalary) {

super(name, surname, username, password, address, phoneNumber);

this.internalPhoneNumber = internalPhoneNumber;

this.baseSalary = baseSalary;

}

public String getInternalPhoneNumber() {

return internalPhoneNumber;

}

public void setInternalPhoneNumber(String internalPhoneNumber) {

this.internalPhoneNumber = internalPhoneNumber;

}

public double getBaseSalary() {

return baseSalary;

}

public void setBaseSalary(double baseSalary) {

this.baseSalary = baseSalary;

}

public boolean isManager(){

return false;

}

}

**CLASS: CRM**

package crm;

import java.util.\*;

import entities.\*;

public class CRM {

public static ArrayList<Person> logInUsers = new ArrayList<>();

public static ArrayList<Person> currentManager = new ArrayList<>();

}

**CLASS: CRMTest**

package crm;

public class CRMTest {

public static void main(String[] args) {

Menu menu = new Menu();

menu.start();

}

}

**CLASS: CRMDatabase**

package crm;

import java.util.\*;

import entities.\*;

import java.io.IOException;

import java.io.Serializable;

public class CRMDatabase implements Serializable {

public static ArrayList<Person> accounts = new ArrayList<>();

public static ArrayList<Service> services = new ArrayList<>();

public static ArrayList<Customer> customers = new ArrayList<>();

public CRMDatabase(){

storeData();

loadData();

if (getAccount("Rea") == null) {

Person Rea = new Manager("Rea", "Harja", "rea", "1234", "Tirane", "123", "123", 123);

accounts.add(Rea);

}

}

public Customer getCustomer(int id) {

try {

for(Customer cust : customers) {

if(cust.getID() == id){

return cust;

}

}

} catch (Exception e) {

e.printStackTrace();

}

return null;

}

public Person getAccount(String username) {

for(Person acc : accounts) {

if(acc.getUsername().equals(username)) {

return acc;

}

}

return null;

}

public void addCustomer(Customer customer) {

customers.add(customer);

}

public void addServices(Service service) {

services.add(service);

}

public ArrayList<Service> getServices() {

return services;

}

public boolean authenticateUser(String username, String password) {

Person userAccount = getAccount(username);

if (userAccount != null) {

return userAccount.validatePin(password);

} else {

return false;

}

}

public Person getUserById(int id){

for(Person user : accounts){

if(user.getID() == id){

return user;

}

}

return null;

}

public Service getServiceById(int id){

for(Service serv : services){

if(serv.getId() == id){

return serv;

}

}

return null;

}

public void removeService(int ID){

for(Iterator<Service> iterator = services.iterator(); iterator.hasNext();){

Service serv = iterator.next();

if(serv.getId() == ID){

iterator.remove();

}

}

}

public static ArrayList<String> getNames(ArrayList<Customer> customers){

ArrayList<String> nameList = new ArrayList<>();

for(Customer cust : customers){

nameList.add(cust.getName());

}

return nameList;

}

public void applyDiscounts(double loyalDiscount, double discount, double businessDiscount) {

for (Customer customer : customers) {

if (customer instanceof LoyalCustomer) {

((LoyalCustomer) customer).setDiscount(loyalDiscount);

} else if (customer instanceof BusinessCustomer) {

((BusinessCustomer) customer).setDiscount(discount);

((BusinessCustomer) customer).setBusinessDiscount(businessDiscount);

}

}

}

public void saveBeforeExit() {

storeData();

}

private void loadData() {

try {

accounts = (ArrayList<Person>) SerializationService.deserialize("accounts.ser");

services = (ArrayList<Service>) SerializationService.deserialize("services.ser");

customers = (ArrayList<Customer>) SerializationService.deserialize("customers.ser");

} catch (IOException | ClassNotFoundException ex) {

ex.printStackTrace();

}

}

private void storeData() {

try {

SerializationService.serialize("accounts.ser", accounts);

SerializationService.serialize("services.ser", services);

SerializationService.serialize("customers.ser", customers);

} catch (IOException ex) {

ex.printStackTrace();

}

}

}

**CLASS: Menu**

package crm;

import entities.Customer;

import entities.Manager;

import entities.Person;

import entities.Service;

import entities.User;

import java.util.Scanner;

public class Menu {

private boolean exit = false;

private int option;

Scanner input = new Scanner(System.in);

private CRM crm = new CRM();

private Person currentUser;

private CRMDatabase crmDatabase = new CRMDatabase();

public void start() {

login();

do {

printMenu();

performAction();

System.out.println();

} while (!exit);

}

private void printMenu() {

System.out.print("CRM Menu\n"

+ "Choose an option:\n"

+ "\t1 - Create a new Service\n"

+ "\t2 - Create a new Customer\n"

+ "\t3 - Assign Service to User\n"

+ "\t4 - Print Customer information\n"

+ "\t5 - Apply offer to Customer\n"

+ "\t6 - Show number of Services by User's Customer\n"

+ "\t7 - Change user\n"

+ "\t8 - Exit\n"

+ "Select an option: ");

option = input.nextInt();

}

private void performAction() {

switch (option) {

case 1:

addService();

break;

case 2:

addCustomer();

break;

case 3:

assignService();

break;

case 4:

printCustomerInformation();

break;

case 5:

applyOffer();

break;

case 6:

getServicesStatistics();

break;

case 7:

changeUser();

break;

case 8:

exit();

break;

default:

System.out.println("Wrong Input. Enter a number between 1 and 8.");

break;

}

}

private void login() {

boolean isAuthenticated;

System.out.println("LOG IN");

do {

System.out.print("Username: ");

String username = input.nextLine();

System.out.print("Password: ");

String password = input.nextLine();

isAuthenticated = crmDatabase.authenticateUser(username, password);

if (isAuthenticated) {

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

currentUser = crmDatabase.getAccount(username);

} else {

System.out.println("Invalid Username or Password\n");

}

} while (!isAuthenticated);

}

private void addService() {

System.out.println("Creating a new Service.");

System.out.print("Enter the priority: ");

int priority = input.nextInt();

System.out.print("Enter the resolution time frame: ");

int resolutionTimeFrame = input.nextInt();

// System.out.print("Enter the status: ");

// boolean status = input.nextBoolean();

System.out.print("Enter the fee: ");

int fee = input.nextInt();

Service service = new Service(priority, resolutionTimeFrame, false, fee);

System.out.println("Service created: " + service);

crmDatabase.addServices(service);

}

private void addCustomer() {

System.out.println("Creating a new Customer.");

System.out.print("Enter customer name: ");

String name = input.nextLine();

System.out.print("Enter customer surname: ");

String surname = input.nextLine();

System.out.print("Enter customer email: ");

String email = input.nextLine();

System.out.print("Enter customer address: ");

String address = input.nextLine();

System.out.print("Enter customer phone number: ");

String phoneNumber = input.nextLine();

Customer customer = new Customer(name, surname, email, address, phoneNumber);

System.out.println("Customer created: " + customer);

crmDatabase.addCustomer(customer);

}

private void assignService() {

int serviceID, userId;

System.out.println("Select an option: \n"

+ "\t1) Assign Service\n"

+ "\t2) Remove Service\n"

+ "\t3) Resolve Service");

int choice = input.nextInt();

System.out.println("Enter service ID: ");

serviceID = input.nextInt();

switch (choice) {

case 1:

System.out.println("Enter customer ID: ");

userId = input.nextInt();

crmDatabase.getServiceById(serviceID).setAssignedUser((User) crmDatabase.getUserById(userId));

break;

case 2:

crmDatabase.removeService(serviceID);

break;

case 3:

crmDatabase.getServiceById(serviceID).setStatus(true);

break;

default:

System.out.println("Wrong input. Enter a number between 1 and 3.");

break;

}

}

private void printCustomerInformation() {

System.out.println("Printing information about a Customer.");

System.out.println("Enter customer ID: ");

int id = input.nextInt();

crmDatabase.getCustomer(id).printInfo();

}

private void applyOffer() {

System.out.println("Applying 10% discount for loyal customer");

System.out.print("Enter normal discount rate for BusinessCustomer: ");

double discount = input.nextDouble();

System.out.print("Enter business discount rate for BusinessCustomer: ");

double businessDiscount = input.nextDouble();

crmDatabase.applyDiscounts(10, discount, businessDiscount);

}

private void getServicesStatistics() {

if (currentUser instanceof Manager) {

System.out.print("Enter Manager password: ");

String password = input.nextLine();

if (!password.equals(currentUser.getPassword())) {

return;

}

} else {

System.out.println("You must log in as Manager");

return;

}

System.out.print("Enter User's username: ");

String username = input.nextLine();

double userFees = 0, totalFees = 0;

User user = (User) crmDatabase.getAccount(username);

for (Service service : crmDatabase.getServices()) {

if (user == service.getAssignedUser() && service.isStatus()) {

userFees += service.getFee();

totalFees += service.getFee();

} else if (service.isStatus() && service.getResolutionTimeFrame() == 1) {

totalFees += service.getFee();

}

}

System.out.println("Total Fees of Services by user " + user.getUsername() + ": " + userFees);

System.out.println("Total Fees of Services for today: " + totalFees);

}

private void changeUser() {

System.out.println("Changing User.");

login();

}

private void exit() {

System.out.println("Closing the Program.");

crmDatabase.saveBeforeExit();

exit = true;

}

}

**CLASS: SerializationService**

package crm;

import java.io.BufferedInputStream;

import java.io.BufferedOutputStream;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

public class SerializationService {

public static Object deserialize(String file) throws IOException, ClassNotFoundException {

FileInputStream fileInputStream = new FileInputStream(file);

BufferedInputStream bufferedInputStream = new BufferedInputStream(fileInputStream);

ObjectInputStream objectInputStream = new ObjectInputStream(bufferedInputStream);

Object object = objectInputStream.readObject();

objectInputStream.close();

return object;

}

public static void serialize(String file, Object object) throws IOException {

FileOutputStream fileOutputStream = new FileOutputStream(file);

BufferedOutputStream bufferedOutputStream = new BufferedOutputStream(fileOutputStream);

ObjectOutputStream objectOutputStream = new ObjectOutputStream(bufferedOutputStream);

objectOutputStream.writeObject(object);

objectOutputStream.close();

}

}

**CLASS: LogInScreen**

package GUI;

import crm.CRM;

import java.awt.\*;

import javax.swing.\*;

import java.awt.event.\*;

import entities.\*;

import crm.\*;

public class LogInScreen extends JFrame {

public static boolean isUserAuthenticated;

private JButton loginButton;

private JTextField username;

private JPasswordField password;

private JLabel usernameLabel;

private JLabel passwordLabel;

private String usernameString;

private String usernamePassword;

private CRMDatabase crmDatabase = new CRMDatabase();

private CRM crm = new CRM();

private static MainMenu menu;

private String statusString;

private JTextField pinJField;

private int PIN;

public LogInScreen(){

super("CRM");

setLayout(new FlowLayout());

isUserAuthenticated = false;

usernameString = "";

loginButton = new JButton("Login");

username = new JTextField(10);

password = new JPasswordField(13);

pinJField = new JTextField(4);

usernameLabel = new JLabel("Username");

passwordLabel = new JLabel("Password");

loginButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

usernameString = username.getText();

usernamePassword = password.getText();

PIN = Integer.parseInt(pinJField.getText());

}

});

add(usernameLabel);

add(username);

add(passwordLabel);

add(password);

add(loginButton);

add(new JLabel("PIN"));

add(pinJField);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

setVisible(true);

setSize(600, 350);

}

public void run() {

while(!isUserAuthenticated) {

authenticateUser();

}

Person user = crmDatabase.getAccount(usernameString);

crm.logInUsers.add(user);

menu = new MainMenu();

menu.setVisible(true);

this.setVisible(false);

}

public String getUsername(){

return usernameString;

}

public String getPassword(){

return usernamePassword;

}

private boolean isPIN(int PIN){

for(Person p : CRMDatabase.accounts){

if(p instanceof Manager && ((Manager) p).getPIN() == PIN){

return true;

}

}

return false;

}

private void authenticateUser() {

this.setVisible(true);

String username = this.getUsername();

String password = this.getPassword();

isUserAuthenticated = crmDatabase.authenticateUser(username, password) && isPIN(PIN);

if(isUserAuthenticated) {

this.usernameString = username;

statusString = "Correct";

} else {

statusString = "The info you entered is not correct. Please try again.";

}

}

public static MainMenu getMenu() {

return menu;

}

}

**CLASS: MainMenu**

package GUI;

import javax.swing.\*;

import java.awt.event.\*;

import java.awt.\*;

import java.io.\*;

import java.util.\*;

import crm.\*;

import entities.\*;

public class MainMenu extends JFrame {

private JMenuBar menuBar;

private JMenu file;

private JButton help;

private JButton exit;

private JMenuItem newService, overallServices, allServices, loadData, saveData;

public MainMenu(){

super("Main Menu");

setLayout(new FlowLayout());

menuBar = new JMenuBar();

file = new JMenu("File");

help = new JButton("Help");

loadData = new JMenuItem("Load Data");

loadData.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

try{

//ArrayList<Invoice> deser = null;

ArrayList<Service> deser = null;

ObjectInputStream in = new ObjectInputStream(new FileInputStream("service.ser"));

deser = (ArrayList<Service>) in.readObject();

CRMDatabase.services = deser;

in.close();

JOptionPane.showMessageDialog(null,"Data Loaded Successfully!");

}catch(Exception exc){

exc.printStackTrace();

}

}

});

help.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

JOptionPane.showMessageDialog(null,"Instructions on using the program: \n" +

"On the file submenu, user can create new Customers, new Services, and can view statistics. \n " +

"If Exit is pressed, the program will terminate and the user will be logged out.");

}

});

exit = new JButton("Exit");

exit.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

System.exit(0);

}

});

newService = new JMenuItem("New Customer");

overallServices = new JMenuItem("Overall Customer");

overallServices.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

int count = 0;

for(Customer cust : CRMDatabase.customers){

ArrayList<Service> temp = cust.getServices();

for(int i = 0; i < cust.getServices().size(); i++){

if(CRM.logInUsers.get(0).getName().equals(temp.get(i).getAssignedUser().getName())){

count++;

}

}

}

JOptionPane.showMessageDialog(null, "The number of Overall Current Customers for some user " +

CRM.logInUsers.get(0).getName() + ": " + count);

}

});

allServices = new JMenuItem("All Customers");

allServices.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

int count = 0;

for(Customer cust : CRMDatabase.customers){

ArrayList<Service> temp = cust.getServices();

for(int i = 0; i < cust.getServices().size(); i++){

if(CRM.logInUsers.get(0).getName().equals(temp.get(i).getAssignedUser().getName())){

count++;

}

}

}

JOptionPane.showMessageDialog(null, "The total number of customers for current user " +

CRM.logInUsers.get(0).getName() + ": " + count);

}

});

saveData = new JMenuItem("Save Data");

saveData.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

try{

FileOutputStream fileOut =

new FileOutputStream("services.ser");

ObjectOutputStream out = new ObjectOutputStream(fileOut);

out.writeObject(CRMDatabase.services);

out.close();

fileOut.close();

JOptionPane.showMessageDialog(null,"Data are saved");

}catch(IOException i){

i.printStackTrace();

}

}

});

add(menuBar);

menuBar.add(file);

menuBar.add(help);

menuBar.add(exit);

file.add(newService);

if(CRM.logInUsers.get(0) instanceof Manager){

file.add(overallServices);

file.add(allServices);

}

file.add(loadData);

file.add(saveData);

setVisible(false);

setSize(600, 350);

}

}

**CLASS: Run**

package GUI;

public class Run {

public static void main(String[] args) {

//main for running GUI

LogInScreen login = new LogInScreen();

login.run();

}

}

# Conclusion

  This is a basic CRM system designed for two categories: User and Manager. Each has its subcategories and features and both inherit the Person class, meaning every feature of the Person class, may be used on the User and Manager class. The users of the system have a unique ID which increments for every user registered. The program presents a menu for the simple user in which he can login by writing the username and password. In the menu there are features like: new service (remove resolve), new costumer, different offers and some of them are accessible only by the managers like: Overall number of Services requested by customers and Total amount of all fees of resolved Services are found.

Each user is able to change from manager to user and vice versa. With the usage of inheritance in the classes it is able to access more attributes easily. For example there are three Classes inheriting from Customer class: Simple Customer, Loyal Customer, and Business Customer.

The program has two main classes: One for the console version and the second main class for executing the application with the GUI. Before closing the system, the program will save the data by using serialization, so they can be loaded later.