



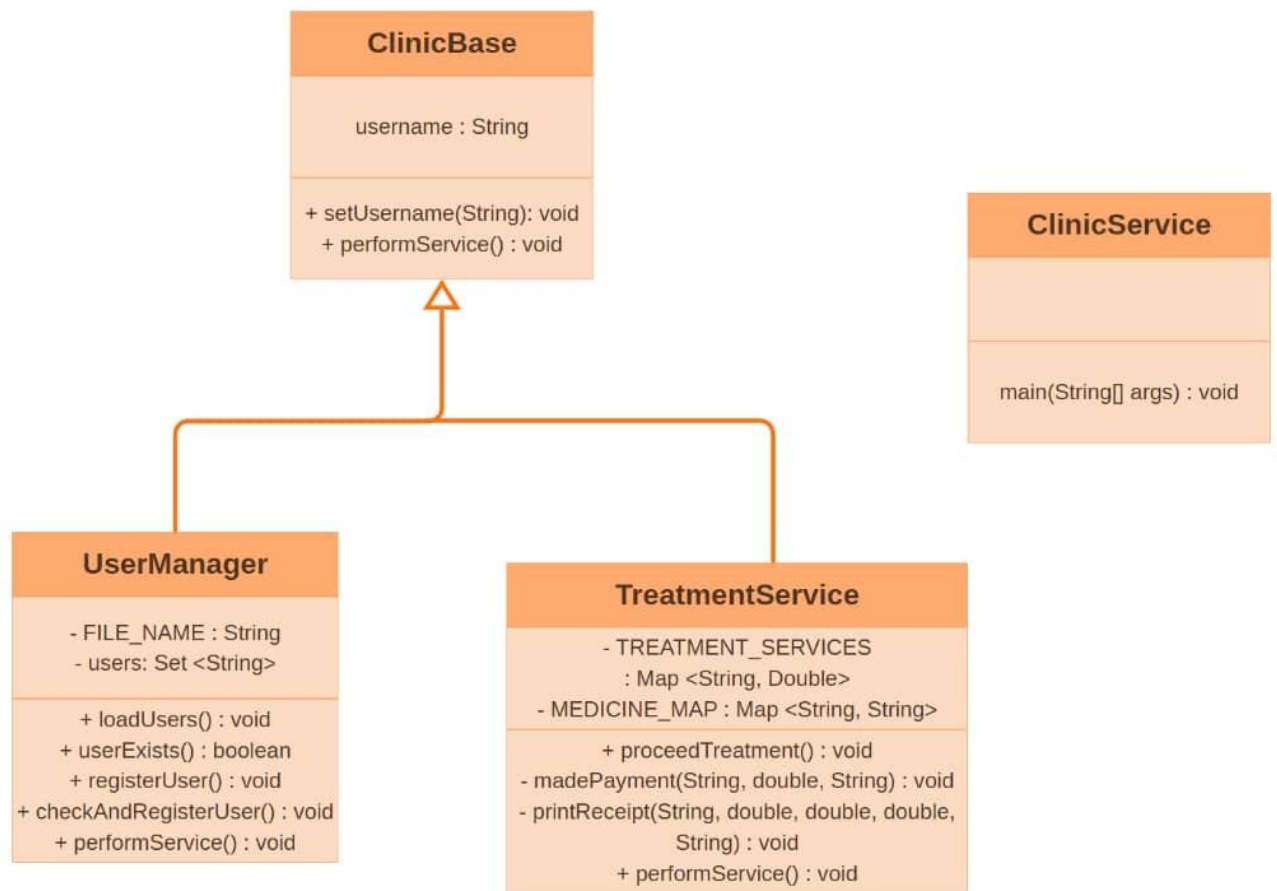
## **GROUP ASSIGNMENT**

### **DTS3093: PENGATURCARAAN BERASASKAN OBJEK**

<b>NAME</b>	<b>MATRIC NUMBER</b>
<b>WONG CHUN LIANG</b>	<b>D20231106337</b>
<b>GOH QUO TENG</b>	<b>D20231106372</b>
<b>JIMBARAN JEDOL DAYOU</b>	<b>D20231106347</b>

**PUAN ASMARA BINTI ALIAS**

## Class Diagram



## **FULL PROGRAM (COPY FROM TXT)**

```
import java.io.*;
import java.util.*;
import javax.swing.JOptionPane;

// ===== Super Class =====
class ClinicBase
{
    String username;

    public void setUsername(String username)
    {
        this.username = username;
    }

    // Polymorphic method
    public void performService() throws IOException
    {
        System.out.println("ClinicBase: Performing base service.");
    }
}

// ===== UserManager Class =====
class UserManager extends ClinicBase
{
    private static final String FILE_NAME = "users.txt";
    private Set<String> users = new HashSet<>();

    public void loadUsers() throws IOException
    {
        File file = new File(FILE_NAME);

        if (!file.exists())
        {
            file.createNewFile();
        }

        BufferedReader reader = new BufferedReader(new FileReader(file));
        String line;

        while ((line = reader.readLine()) != null)
        {
            users.add(line.trim());
        }

        reader.close();
    }
}
```

```

public boolean userExists()
{
return users.contains(username);
}

public void registerUser() throws IOException
{
users.add(username);
PrintWriter writer = new PrintWriter(new FileWriter(FILE_NAME, true));
writer.println(username);
writer.close();
System.out.println("User " + username + " registered.");
}

public void checkAndRegisterUser() throws IOException
{
if (!userExists())
{
JOptionPane.showMessageDialog(null, "User not found. Registering...");
System.out.println("User not found. Registering...");
registerUser();
}
else
{
JOptionPane.showMessageDialog(null, "Welcome back, " + username + "!");
System.out.println("User found.");
}
}

@Override
public void performService() throws IOException
{
checkAndRegisterUser();
}
}

// ===== TreatmentService Class =====
class TreatmentService extends ClinicBase
{
private static final Map<String, Double> TREATMENT_PRICES = new
HashMap<>();
private static final Map<String, String> MEDICINE_MAP = new HashMap<>();

static
{

```

```

TREATMENT_PRICES.put("Fever", 100.0);
TREATMENT_PRICES.put("Cold", 70.0);
TREATMENT_PRICES.put("Cough", 50.0);

MEDICINE_MAP.put("Fever", "Acetaminophen");
MEDICINE_MAP.put("Cold", "Ibuprofen");
MEDICINE_MAP.put("Cough", "Ambroxol");
}

public void proceedTreatment()
{
    Object[] options = {"Fever", "Cold", "Cough"};
    String treatment = (String) JOptionPane.showInputDialog(
        null,
        "Choose your treatment:",
        "Treatment Selection",
        JOptionPane.PLAIN_MESSAGE,
        null,
        options,
        "Fever");

    if (treatment == null)
    {
        JOptionPane.showMessageDialog(null, "No treatment selected. Exiting.");
        System.out.println("No treatment selected. Exiting.");
        return;
    }

    double price = TREATMENT_PRICES.get(treatment);
    String medicine = MEDICINE_MAP.get(treatment);

    // Inform user about medicine and price
    String info = "You selected: " + treatment + "\n"
        + "Recommended Medicine: " + medicine + "\n"
        + "Price: RM " + price;
    JOptionPane.showMessageDialog(null, info);
    System.out.println("Treatment selected: " + treatment + ", Medicine: " + medicine
+ ", Price: RM " + price);

    madePayment(treatment, price, medicine);
}

private void madePayment(String treatment, double price, String medicine)
{
    String input = JOptionPane.showInputDialog("Enter payment amount (RM):");

```

```

try
{
double amount = Double.parseDouble(input);
double balance = amount - price;

if (balance < 0)
{
JOptionPane.showMessageDialog(null, "Insufficient payment. Required: RM
" + price);
System.out.println("Insufficient payment. Required: RM " + price);
return;
}

JOptionPane.showMessageDialog(null, "Payment received: RM " + amount);
System.out.println("Payment received: RM " + amount);
printReceipt(treatment, price, amount, balance, medicine);

} catch (NumberFormatException e) {
JOptionPane.showMessageDialog(null, "Invalid input.");
System.out.println("Invalid payment input.");
}
}

private void printReceipt(String treatment, double price, double amount, double
balance, String medicine)
{
String receipt = "----- Receipt -----\\n"
+ "Patient: " + username + "\\n"
+ "Treatment: " + treatment + "\\n"
+ "Medicine: " + medicine + "\\n"
+ "Price: RM " + price + "\\n"
+ "Paid: RM " + amount + "\\n"
+ "Balance: RM " + balance + "\\n"
+ "Thank you. Get well soon!";

JOptionPane.showMessageDialog(null, receipt);
System.out.println(receipt);
}

@Override
public void performService()
{
proceedTreatment();
}
}

```

```
// ===== Main Class =====
public class ClinicService
{
    public static void main(String[] args) throws IOException
    {
        ClinicBase userManager = new UserManager();

        String username = JOptionPane.showInputDialog("Enter your username:");
        if (username == null || username.trim().isEmpty())
        {
            JOptionPane.showMessageDialog(null, "Username cannot be empty.");
            System.out.println("No username entered. Exiting.");
            return;
        }

        userManager.setUsername(username);
        ((UserManager) userManager).loadUsers();
        userManager.performService();

        ClinicBase treatment = new TreatmentService();
        treatment.setUsername(username);
        treatment.performService();
    }
}
```

## FULL PROGRAM SCREENSHOT

ClinicService - Notepad

File Edit Format View Help

```
import java.io.*;
import java.util.*;
import javax.swing.JOptionPane;

// ===== Super Class =====
class ClinicBase
{
    String username;

    public void setUsername(String username)
    {
        this.username = username;
    }

    // Polymorphic method
    public void performService() throws IOException
    {
        System.out.println("ClinicBase: Performing base service.");
    }
}

// ===== UserManager Class =====
class UserManager extends ClinicBase
{
    private static final String FILE_NAME = "users.txt";
    private Set<String> users = new HashSet<>();

    public void loadUsers() throws IOException
    {
        File file = new File(FILE_NAME);

        if (!file.exists())
        {
            file.createNewFile();
        }

        BufferedReader reader = new BufferedReader(new FileReader(file));
        String line;

        while ((line = reader.readLine()) != null)
        {
            users.add(line.trim());
        }
    }
}
```

---

```
    }

    reader.close();
}

public boolean userExists()
{
    return users.contains(username);
}

public void registerUser() throws IOException
{
    users.add(username);
    PrintWriter writer = new PrintWriter(new FileWriter(FILE_NAME, true));
    writer.println(username);
    writer.close();
    System.out.println("User " + username + " registered.");
}

public void checkAndRegisterUser() throws IOException
{
    if (!userExists())
    {
        JOptionPane.showMessageDialog(null, "User not found. Registering...");
        System.out.println("User not found. Registering...");
        registerUser();
    }
    else
    {
        JOptionPane.showMessageDialog(null, "Welcome back, " + username + "!");
        System.out.println("User found.");
    }
}

@Override
public void performService() throws IOException
{
    checkAndRegisterUser();
}
```

---

```
// ===== TreatmentService Class =====
class TreatmentService extends ClinicBase
{
    private static final Map<String, Double> TREATMENT_PRICES = new HashMap<>();
    private static final Map<String, String> MEDICINE_MAP = new HashMap<>();

    static
    {
        TREATMENT_PRICES.put("Fever", 100.0);
        TREATMENT_PRICES.put("Cold", 70.0);
        TREATMENT_PRICES.put("Cough", 50.0);

        MEDICINE_MAP.put("Fever", "Acetaminophen");
        MEDICINE_MAP.put("Cold", "Ibuprofen");
        MEDICINE_MAP.put("Cough", "Ambroxol");
    }

    public void proceedTreatment()
    {
        Object[] options = {"Fever", "Cold", "Cough"};
        String treatment = (String) JOptionPane.showInputDialog(
            null,
            "Choose your treatment:",
            "Treatment Selection",
            JOptionPane.PLAIN_MESSAGE,
            null,
            options,
            "Fever");

        if (treatment == null)
        {
            JOptionPane.showMessageDialog(null, "No treatment selected. Exiting.");
            System.out.println("No treatment selected. Exiting.");
            return;
        }

        double price = TREATMENT_PRICES.get(treatment);
        String medicine = MEDICINE_MAP.get(treatment);

        // Inform user about medicine and price
        String info = "You selected: " + treatment + "\n"
            + "Recommended Medicine: " + medicine + "\n"
    }
}
```

---

```
        + "Price: RM " + price;
        JOptionPane.showMessageDialog(null, info);
        System.out.println("Treatment selected: " + treatment + ", Medicine: " + medicine + ", Price: RM " + price);

        madePayment(treatment, price, medicine);
    }

    private void madePayment(String treatment, double price, String medicine)
    {
        String input = JOptionPane.showInputDialog("Enter payment amount (RM):");

        try
        {
            double amount = Double.parseDouble(input);
            double balance = amount - price;

            if (balance < 0)
            {
                JOptionPane.showMessageDialog(null, "Insufficient payment. Required: RM " + price);
                System.out.println("Insufficient payment. Required: RM " + price);
                return;
            }

            JOptionPane.showMessageDialog(null, "Payment received: RM " + amount);
            System.out.println("Payment received: RM " + amount);
            printReceipt(treatment, price, amount, balance, medicine);

        } catch (NumberFormatException e) {
            JOptionPane.showMessageDialog(null, "Invalid input.");
            System.out.println("Invalid payment input.");
        }
    }

    private void printReceipt(String treatment, double price, double amount, double balance, String medicine)
    {
        String receipt = "----- Receipt -----\n"
            + "Patient: " + username + "\n"
            + "Treatment: " + treatment + "\n"
            + "Medicine: " + medicine + "\n"
            + "Price: RM " + price + "\n"
            + "Paid: RM " + amount + "\n";
    }
}
```

```

        + "Balance: RM " + balance + "\n"
        + "Thank you. Get well soon!";

        JOptionPane.showMessageDialog(null, receipt);
        System.out.println(receipt);
    }

    @Override
    public void performService()
    {
        proceedTreatment();
    }
}

// ===== Main Class =====
public class ClinicService
{
    public static void main(String[] args) throws IOException
    {
        ClinicBase userManager = new UserManager();

        String username = JOptionPane.showInputDialog("Enter your username:");
        if (username == null || username.trim().isEmpty())
        {
            JOptionPane.showMessageDialog(null, "Username cannot be empty.");
            System.out.println("No username entered. Exiting.");
            return;
        }

        userManager.setUsername(username);
        ((UserManager) userManager).loadUsers();
        userManager.performService();

        ClinicBase treatment = new TreatmentService();
        treatment.setUsername(username);
        treatment.performService();
    }
}

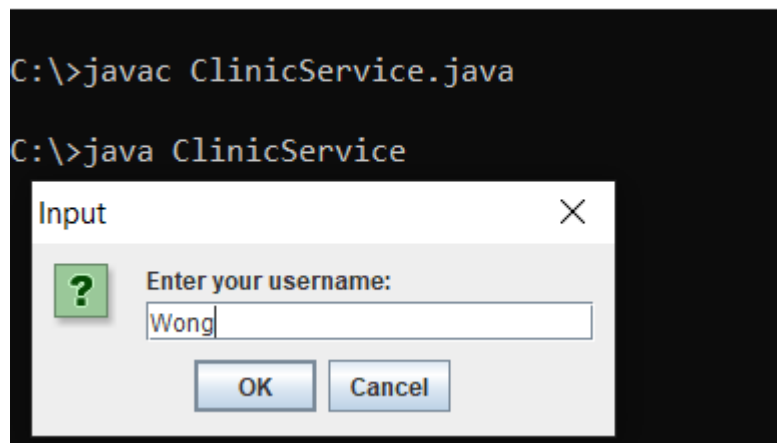
```

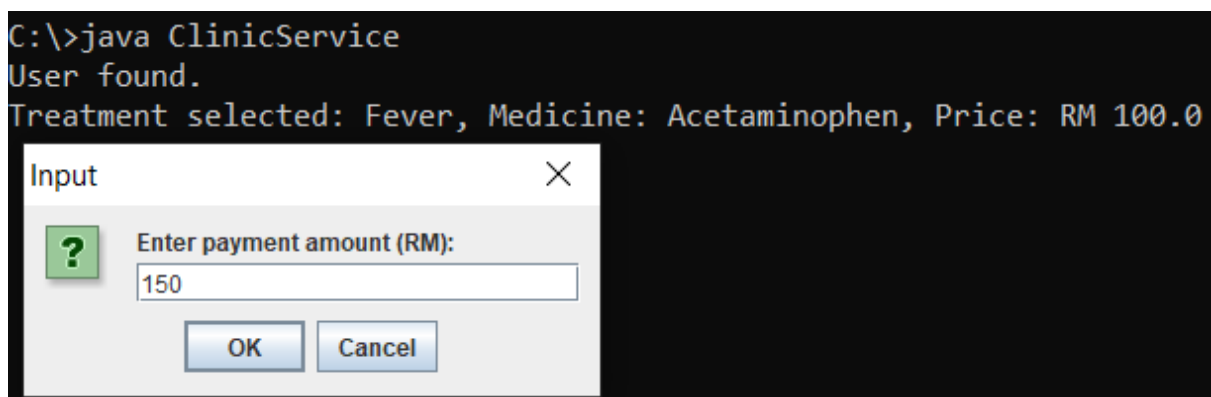
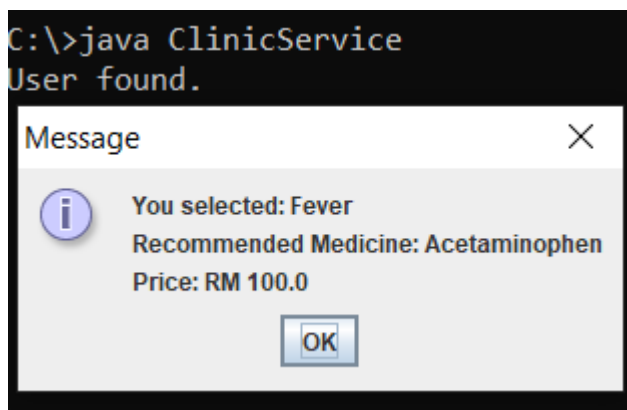
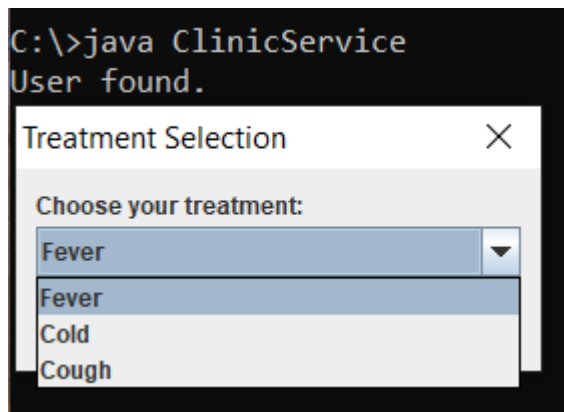
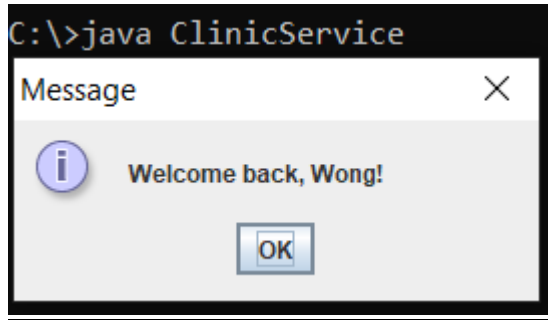
---

## OUTPUT

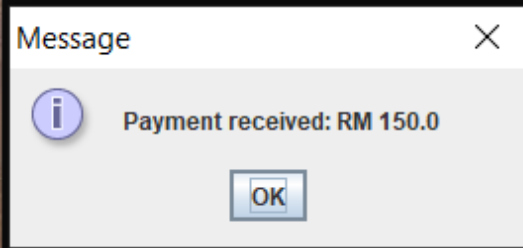
### Serve with existing user

 Administrator: Command Prompt - java ClinicService

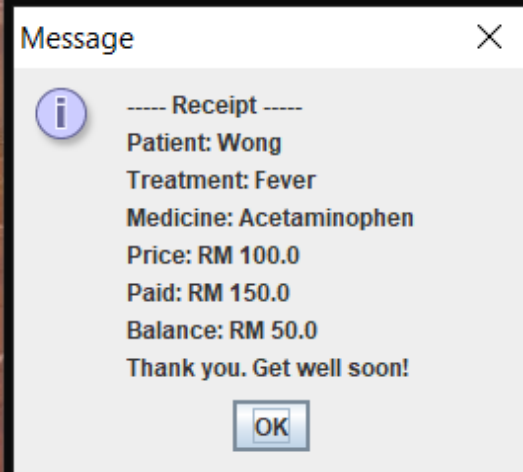




```
C:\>java ClinicService
User found.
Treatment selected: Fever, Medicine: Acetaminophen, Price: RM 100.0
```

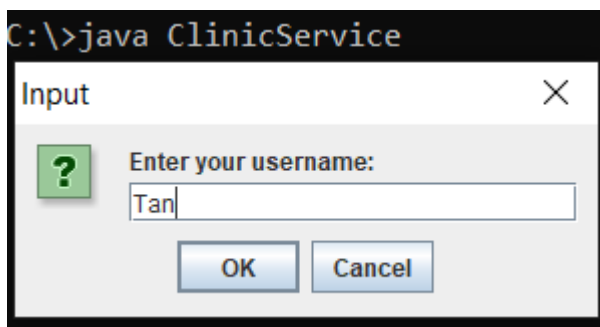


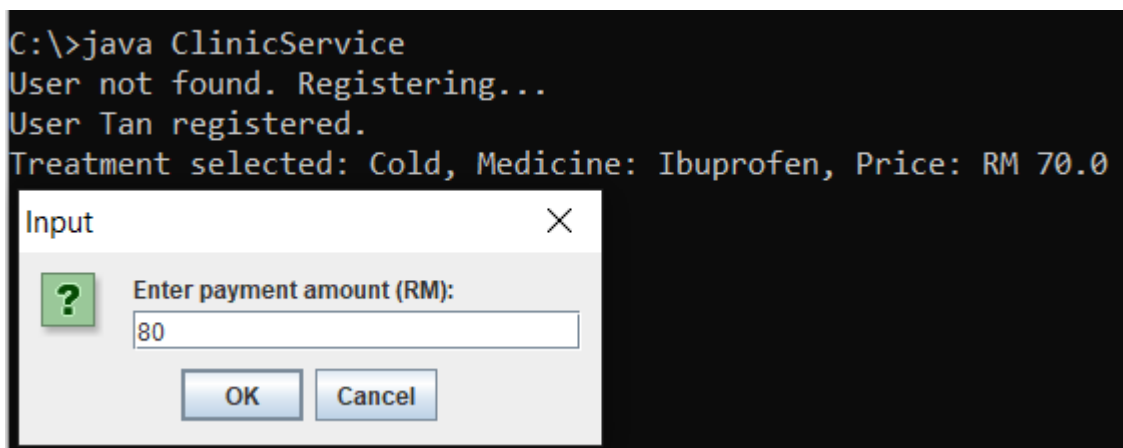
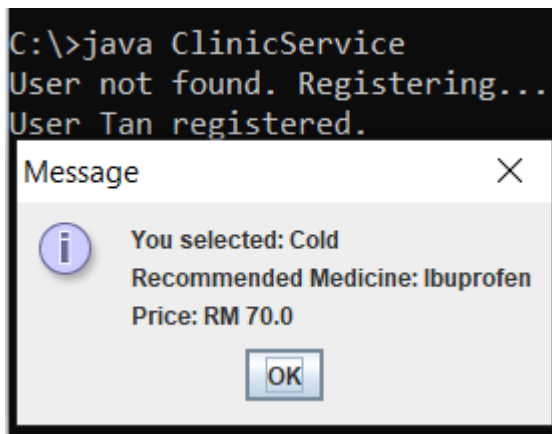
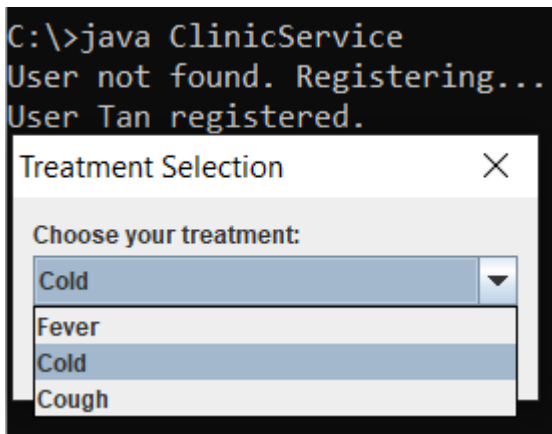
```
C:\>java ClinicService
User found.
Treatment selected: Fever, Medicine: Acetaminophen, Price: RM 100.0
Payment received: RM 150.0
```



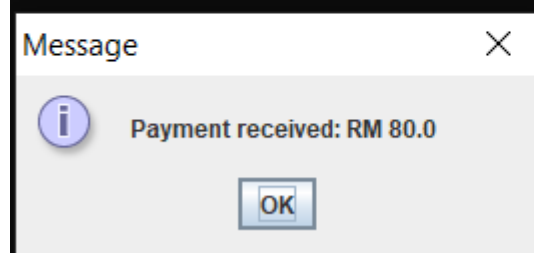
```
----- Receipt -----
Patient: Wong
Treatment: Fever
Medicine: Acetaminophen
Price: RM 100.0
Paid: RM 150.0
Balance: RM 50.0
Thank you. Get well soon!
```

### Serve with new user

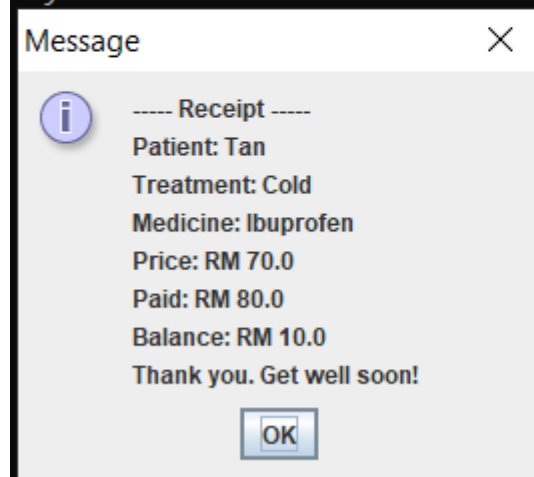




```
C:\>java ClinicService
User not found. Registering...
User Tan registered.
Treatment selected: Cold, Medicine: Ibuprofen, Price: RM 70.0
```



```
C:\>java ClinicService
User not found. Registering...
User Tan registered.
Treatment selected: Cold, Medicine: Ibuprofen, Price: RM 70.0
Payment received: RM 80.0
```



```
----- Receipt -----
Patient: Tan
Treatment: Cold
Medicine: Ibuprofen
Price: RM 70.0
Paid: RM 80.0
Balance: RM 10.0
Thank you. Get well soon!
```

### Output which is the user list in the txt

```
users - Notepad
File Edit Format View Help
Wong
Goh
Tan
```