**PROJECT NAME : Expense Tracker**

**SOURCE CODE:**

import java.io.\*;

import java.util.\*;

class Expense {

String date;

String category;

double amount;

public Expense(String date, String category, double amount) {

this.date = date;

this.category = category;

this.amount = amount;

}

}

class User {

String username;

String password;

List<Expense> expenses;

public User(String username, String password) {

this.username = username;

this.password = password;

this.expenses = new ArrayList<>();

}

}

class ExpenseTracker {

List<User> users;

public ExpenseTracker() {

this.users = new ArrayList<>();

}

public void addUser(User user) {

users.add(user);

}

public User getUser(String username) {

for (User user : users) {

if (user.username.equals(username)) {

return user;

}

}

return null;

}

public void addExpense(User user, String date, String category, double amount) {

Expense expense = new Expense(date, category, amount);

user.expenses.add(expense);

}

public void saveData() {

try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream("expense\_data.dat"))) {

oos.writeObject(users);

} catch (IOException e) {

e.printStackTrace();

}

}

@SuppressWarnings("unchecked")

public void loadData() {

try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream("expense\_data.dat"))) {

users = (List<User>) ois.readObject();

} catch (IOException | ClassNotFoundException e) {

e.printStackTrace();

}

}

}

public class Main {

public static void main(String[] args) {

ExpenseTracker expenseTracker = new ExpenseTracker();

// Sample user registration

User user1 = new User("john\_doe", "password123");

expenseTracker.addUser(user1);

// Sample expense entry

expenseTracker.addExpense(user1, "2024-01-12", "Groceries", 50.0);

expenseTracker.addExpense(user1, "2024-01-13", "Utilities", 100.0);

// Save and load data (for demonstration purposes)

expenseTracker.saveData();

expenseTracker.loadData();

// Example: Retrieve user and display expenses

User retrievedUser = expenseTracker.getUser("john\_doe");

if (retrievedUser != null) {

System.out.println("User: " + retrievedUser.username);

System.out.println("Expenses:");

for (Expense expense : retrievedUser.expenses) {

System.out.println("Date: " + expense.date + ", Category: " + expense.category + ", Amount: " + expense.amount);

}

}

}

}

**OUTPUT:**

java.io.NotSerializableException: User

at java.base/java.io.ObjectOutputStream.writeObject0(ObjectOutputStream.java:1185)

at java.base/java.io.ObjectOutputStream.writeObject(ObjectOutputStream.java:349)

at java.base/java.util.ArrayList.writeObject(ArrayList.java:896)

at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)

at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.base/java.lang.reflect.Method.invoke(Method.java:566)

at java.base/java.io.ObjectStreamClass.invokeWriteObject(ObjectStreamClass.java:1130)

at java.base/java.io.ObjectOutputStream.writeSerialData(ObjectOutputStream.java:1497)

at java.base/java.io.ObjectOutputStream.writeOrdinaryObject(ObjectOutputStream.java:1433)

at java.base/java.io.ObjectOutputStream.writeObject0(ObjectOutputStream.java:1179)

at java.base/java.io.ObjectOutputStream.writeObject(ObjectOutputStream.java:349)

at ExpenseTracker.saveData(Main.java:55)

at Main.main(Main.java:84)

java.io.WriteAbortedException: writing aborted; java.io.NotSerializableException: User

at java.base/java.io.ObjectInputStream.readObject0(ObjectInputStream.java:1598)

at java.base/java.io.ObjectInputStream.readObject(ObjectInputStream.java:430)

at java.base/java.util.ArrayList.readObject(ArrayList.java:928)

at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)

at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.base/java.lang.reflect.Method.invoke(Method.java:566)

at java.base/java.io.ObjectStreamClass.invokeReadObject(ObjectStreamClass.java:1160)

at java.base/java.io.ObjectInputStream.readSerialData(ObjectInputStream.java:2216)

at java.base/java.io.ObjectInputStream.readOrdinaryObject(ObjectInputStream.java:2087)

at java.base/java.io.ObjectInputStream.readObject0(ObjectInputStream.java:1594)

at java.base/java.io.ObjectInputStream.readObject(ObjectInputStream.java:430)

at ExpenseTracker.loadData(Main.java:64)

at Main.main(Main.java:85)

Caused by: java.io.NotSerializableException: User

at java.base/java.io.ObjectOutputStream.writeObject0(ObjectOutputStream.java:1185)

at java.base/java.io.ObjectOutputStream.writeObject(ObjectOutputStream.java:349)

at java.base/java.util.ArrayList.writeObject(ArrayList.java:896)

at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method)

at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)

at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)

at java.base/java.lang.reflect.Method.invoke(Method.java:566)

at java.base/java.io.ObjectStreamClass.invokeWriteObject(ObjectStreamClass.java:1130)

at java.base/java.io.ObjectOutputStream.writeSerialData(ObjectOutputStream.java:1497)

at java.base/java.io.ObjectOutputStream.writeOrdinaryObject(ObjectOutputStream.java:1433)

at java.base/java.io.ObjectOutputStream.writeObject0(ObjectOutputStream.java:1179)

at java.base/java.io.ObjectOutputStream.writeObject(ObjectOutputStream.java:349)

at ExpenseTracker.saveData(Main.java:55)

at Main.main(Main.java:84)

User: john\_doe

Expenses:

Date: 2024-01-12, Category: Groceries, Amount: 50.0

Date: 2024-01-13, Category: Utilities, Amount: 100.0