```
import java.io.*;
import java.util.*;

class User {
    private String username;
    private String password;

    public User(String username, String password) {
        this.username = username;
        this.password = password;
    }

    public String getUsername() {
        return username;
    }

    public String getPassword() {
        return password;
    }
}

class Expense {
    private String date;
    private String category;
    private double amount;

    public Expense(String date, String category, double amount) {
        this.date = date;
        this.category = category;
        this.amount = amount;
    }
```

```
    public String getDate() {
        return date;
    }

    public String getCategory() {
        return category;
    }

    public double getAmount() {
        return amount;
    }
}

class ExpenseTracker {
    private List<User> users;
    private List<Expense> expenses;
    private User currentUser;

    public ExpenseTracker() {
        this.users = new ArrayList<>();
        this.expenses = new ArrayList<>();
        this.currentUser = null;
    }

    public boolean register(String username, String password) {
        for (User user : users) {
            if (user.getUsername().equals(username)) {
                return false; // Username already exists
            }
        }
```

```
    users.add(new User(username, password));
    return true;
}

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; // Invalid username or password
}

public void logout() {
    currentUser = null;
}

public void addExpense(String date, String category, double amount) {
    expenses.add(new Expense(date, category, amount));
}

public List<Expense> getExpenses() {
    return expenses;
}

public Map<String, Double> getCategoryWiseTotal() {
    Map<String, Double> categoryWiseTotal = new HashMap<>();
    for (Expense expense : expenses) {
        categoryWiseTotal.put(expense.getCategory(),
                categoryWiseTotal.getOrDefault(expense.getCategory(), 0.0) + expense.getAmount());
```

```
        }
        return categoryWiseTotal;
    }

    public void saveExpensesToFile(String fileName) {
        try (PrintWriter writer = new PrintWriter(new FileWriter(fileName))) {
            for (Expense expense : expenses) {
                writer.println(expense.getDate() + "," + expense.getCategory() + "," +
expense.getAmount());
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public void loadExpensesFromFile(String fileName) {
        expenses.clear();
        try (BufferedReader reader = new BufferedReader(new FileReader(fileName))) {
            String line;
            while ((line = reader.readLine()) != null) {
                String[] parts = line.split(",");
                if (parts.length == 3) {
                    expenses.add(new Expense(parts[0], parts[1], Double.parseDouble(parts[2])));
                }
            }
        } catch (IOException | NumberFormatException e) {
            e.printStackTrace();
        }
    }
}
```

```
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        ExpenseTracker expenseTracker = new ExpenseTracker();

        while (true) {
            System.out.println("1. Register");
            System.out.println("2. Login");
            System.out.println("3. Exit");
            System.out.print("Enter your choice: ");
            int choice = scanner.nextInt();
            scanner.nextLine(); // Consume newline character

            switch (choice) {
                case 1:
                    System.out.print("Enter username: ");
                    String username = scanner.nextLine();
                    System.out.print("Enter password: ");
                    String password = scanner.nextLine();
                    if (expenseTracker.register(username, password)) {
                        System.out.println("Registration successful!");
                    } else {
                        System.out.println("Username already exists. Please choose a different username.");
                    }
                    break;
                case 2:
                    System.out.print("Enter username: ");
                    username = scanner.nextLine();
                    System.out.print("Enter password: ");
                    password = scanner.nextLine();
                    if (expenseTracker.login(username, password)) {
```

```
                System.out.println("Login successful!");
                expenseMenu(scanner, expenseTracker);
            } else {
                System.out.println("Invalid username or password.");
            }
            break;
        case 3:
            System.out.println("Exiting...");
            System.exit(0);
            break;
        default:
            System.out.println("Invalid choice. Please try again.");
    }
  }
}

public static void expenseMenu(Scanner scanner, ExpenseTracker expenseTracker) {
  while (true) {
    System.out.println("\nExpense Menu");
    System.out.println("1. Add Expense");
    System.out.println("2. View Expenses");
    System.out.println("3. View Category-wise Total");
    System.out.println("4. Save Expenses to File");
    System.out.println("5. Load Expenses from File");
    System.out.println("6. Logout");
    System.out.print("Enter your choice: ");
    int choice = scanner.nextInt();
    scanner.nextLine(); // Consume newline character

    switch (choice) {
        case 1:
```

```
            System.out.print("Enter date (YYYY-MM-DD): ");
            String date = scanner.nextLine();
            System.out.print("Enter category: ");
            String category = scanner.nextLine();
            System.out.print("Enter amount: ");
            double amount = scanner.nextDouble();
            expenseTracker.addExpense(date, category, amount);
            System.out.println("Expense added successfully.");
            break;
        case 2:
            List<Expense> expenses = expenseTracker.getExpenses();
            if (expenses.isEmpty()) {
                System.out.println("No expenses to display.");
            } else {
                for (Expense expense : expenses) {
                    System.out.println(expense.getDate() + " - " + expense.getCategory() + " - $" +
expense.getAmount());
                }
            }
            break;
        case 3:
            Map<String, Double> categoryWiseTotal = expenseTracker.getCategoryWiseTotal();
            if (categoryWiseTotal.isEmpty()) {
                System.out.println("No expenses to display.");
            } else {
                for (Map.Entry<String, Double> entry : categoryWiseTotal.entrySet()) {
                    System.out.println(entry.getKey() + " - $" + entry.getValue());
                }
            }
            break;
        case 4:
```

```
                System.out.print("Enter file name to save: ");
                String saveFileName = scanner.nextLine();
                expenseTracker.saveExpensesToFile(saveFileName);
                System.out.println("Expenses saved to file.");
                break;
            case 5:
                System.out.print("Enter file name to load: ");
                String loadFileName = scanner.nextLine();
                expenseTracker.loadExpensesFromFile(loadFileName);
                System.out.println("Expenses loaded from file.");
                break;
            case 6:
                expenseTracker.logout();
                System.out.println("Logged out.");
                return;
            default:
                System.out.println("Invalid choice. Please try again.");
        }
    }
  }
}
```