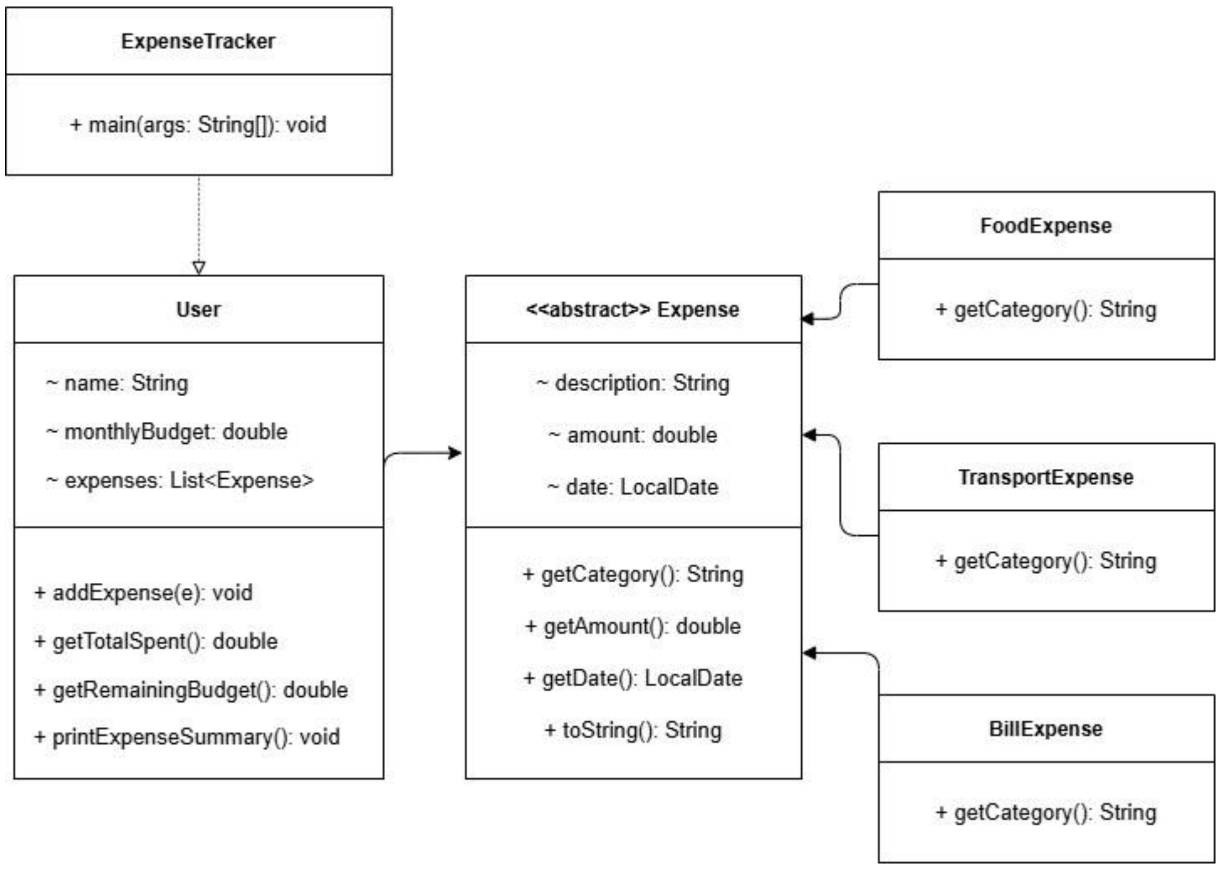
EXPENSE TRACKER PROJECT REPORT

Tables Of Contents

1. BRIEF EXPLANATION OF THE PROBLEM

This project aims to implement a terminal-based expense tracker using Java and Object-Oriented Programming (OOP) principles. The program allows users to enter a budget, log categorized expenses, and view a summary showing total spending and remaining balance. It applies OOP concepts like inheritance and polymorphism to manage expense types.

2. UML DIAGRAM



3. DATA FIELD DEFINITION LIST

#	Modifier	Type	Name	Comment
1	default	String	name	Stores the user's name.
2	default	Double	monthlyBudget	Stores the user's monthly budget.
3	default	List<Expense>	expenses	Holds a list of all expenses logged by the user.
4	default	String	description	Describes the purpose or details of the expense. (in Expense class)
5	default	Double	amount	Represent the monetary value of an expense
6	default	LocalDate	date	The date on which the expense occurred.

4. OBJECT DEFINITION LIST

#	Name	Args	Comment
1	User	String name, double monthlyBudget	Initializes a user with a name and monthly budget. Also creates an empty list of expenses.
2	Expense	String description, double amount, LocalDate date	Abstract constructor used to initialize common fields of all expense types. Called by subclasses.

3	FoodExpense / TransportExpense / BillExpense	String description, double amount, LocalDate date	Each subclass constructor calls the parent Expense constructor to initialize its fields.

5. METHOD DEFINITION LIST

#	Modifier	Return Type	Name	Args	Comment
1	public	Void	addExpense	(Expense e)	Adds a new expense to the user's expense list.
2	public	Double	getTotalSpent	()	Calculates and returns the total amount spent.
3	public	Double	getRemainingBudget	()	Returns the remaining budget (monthlyBudget - totalSpent).
4	public	Void	printExpenseSummary	()	Displays a summary of all expenses and budget information.
5	public	String	getCategory	()	Abstract method in Expense; implemented by subclasses to return category.
6	public	String	toString	()	Returns a formatted string with category, description, amount, and date.

7	public	LocalDate	getDate	()	Returns the date of the expense.
8	public	Double	getAmount	()	Returns the amount of the expense.

6. CLASS DEFINITION LIST

#	Modifier	Name	Abstract ?	Extends what ?	Comment
1	public	User	No	None	Represents a user with a name, monthly budget, and a list of expenses.
2	public	Expense	Yes	None	Abstract base class for all expense types, defines shared fields and methods.
3	public	FoodExpense	No	Expense	A subclass representing a food-related expense.
4	public	TransportExpense	No	Expense	A subclass representing a transport-related expense.
5	public	BillExpense	No	Expense	A subclass representing a bill-related expense.

7. OOP CONCEPTS

1. Inheritance

The FoodExpense, TransportExpense, and BillExpense classes inherit from the abstract Expense class.

Why: This allows all expense types to share common fields (description, amount, date) and methods, reducing code duplication.

2. Abstraction

The Expense class is declared as abstract and includes an abstract method getCategory().

Why: This enforces that each subclass must provide its own category, allowing a clean and extendable structure for new expense types.

3. Encapsulation

Fields like name, monthlyBudget, and expenses in the User class, and fields in Expense, are accessed and modified via methods (e.g., addExpense(), getAmount()).

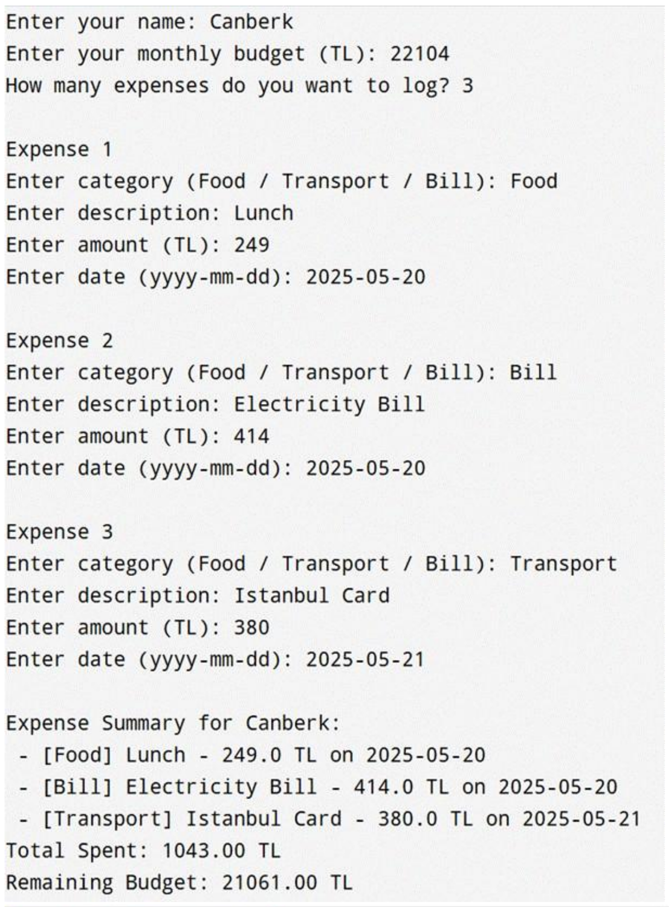
Why: This protects internal data and controls how data is accessed or modified.

4. Polymorphism

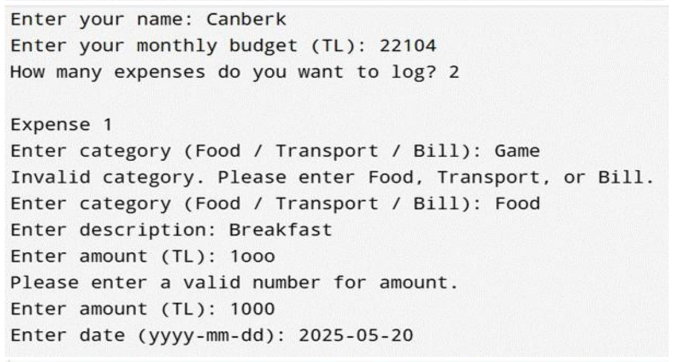
The program stores all expenses as Expense type in a list (List<Expense>), but at runtime, the correct subclass methods like getCategory() or toString() are invoked.

Why: This allows different behaviors (food, transport, bill) to be handled uniformly.

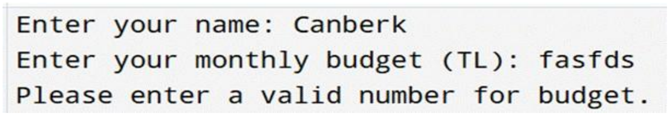
8. SCREENSHOTS OF THE SEQUENCE



```
Enter your name: Canberk
Enter your monthly budget (TL): 22104
How many expenses do you want to log? 3

Expense 1
Enter category (Food / Transport / Bill): Food
Enter description: Lunch
Enter amount (TL): 249
Enter date (yyyy-mm-dd): 2025-05-20

Expense 2
Enter category (Food / Transport / Bill): Bill
Enter description: Electricity Bill
Enter amount (TL): 414
Enter date (yyyy-mm-dd): 2025-05-20

Expense 3
Enter category (Food / Transport / Bill): Transport
Enter description: Istanbul Card
Enter amount (TL): 380
Enter date (yyyy-mm-dd): 2025-05-21

Expense Summary for Canberk:
 - [Food] Lunch - 249.0 TL on 2025-05-20
 - [Bill] Electricity Bill - 414.0 TL on 2025-05-20
 - [Transport] Istanbul Card - 380.0 TL on 2025-05-21
Total Spent: 1043.00 TL
Remaining Budget: 21061.00 TL
```



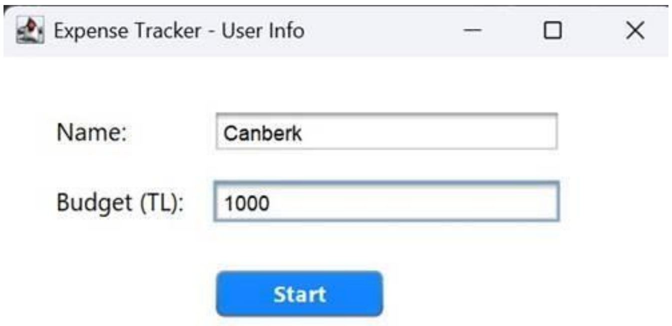
```
Enter your name: Canberk
Enter your monthly budget (TL): 22104
How many expenses do you want to log? 2

Expense 1
Enter category (Food / Transport / Bill): Game
Invalid category. Please enter Food, Transport, or Bill.
Enter category (Food / Transport / Bill): Food
Enter description: Breakfast
Enter amount (TL): 1ooo
Please enter a valid number for amount.
Enter amount (TL): 1000
Enter date (yyyy-mm-dd): 2025-05-20
```



```
Enter your name: Canberk
Enter your monthly budget (TL): fasfds
Please enter a valid number for budget.
```

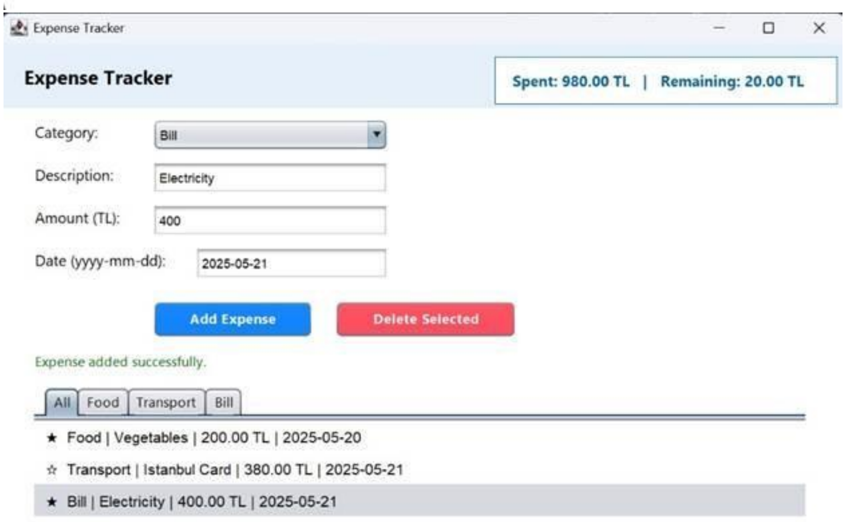
9. EXTRA FEATURES

- GUI Design



Name: Canberk

Budget (TL): 1000



Spent: 980.00 TL | Remaining: 20.00 TL

Category: Bill

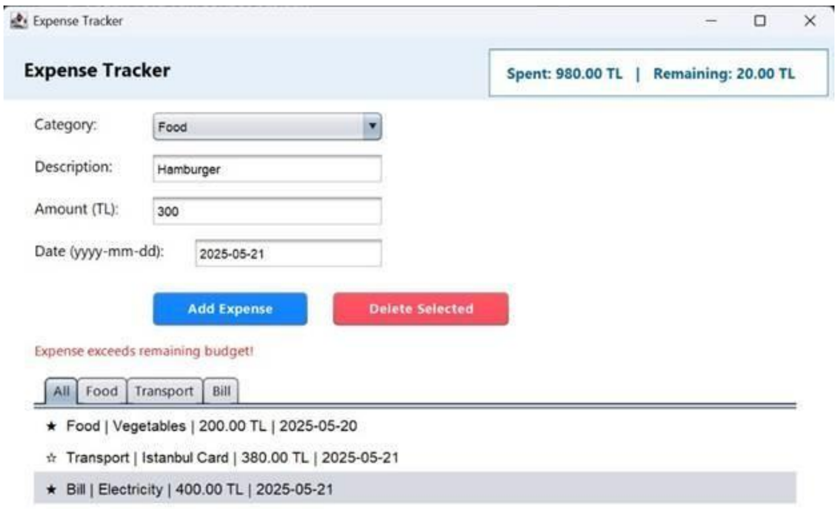
Description: Electricity

Amount (TL): 400

Date (yyyy-mm-dd): 2025-05-21

Expense added successfully.

★ Food | Vegetables | 200.00 TL | 2025-05-20
☆ Transport | Istanbul Card | 380.00 TL | 2025-05-21
★ Bill | Electricity | 400.00 TL | 2025-05-21



Spent: 980.00 TL | Remaining: 20.00 TL

Category: Food

Description: Hamburger

Amount (TL): 300

Date (yyyy-mm-dd): 2025-05-21

Expense exceeds remaining budget!

★ Food | Vegetables | 200.00 TL | 2025-05-20
☆ Transport | Istanbul Card | 380.00 TL | 2025-05-21
★ Bill | Electricity | 400.00 TL | 2025-05-21