# Chapter 3: Design

## Introduction

Design phase is the stage where the system architecture is designed based on the user requirements. The logical system design produced during the analysis phase is converted into physical design in design phase. Different tools and techniques are used to describe the design of the system.

## Structural Modeling

Structural model provides the structural perspective of the data that is processed by the system.

### Class diagram

Class diagram is a static structural diagram which shows the structure of classes of the system, its methods, attributes and relationship among objects.

**Justification**

The reasons for using class diagrams are as follows:

1. Shows static structure of the class and objects
2. Because I am using object-oriented methodology
3. Gives the overview of the system and helps to maintain time

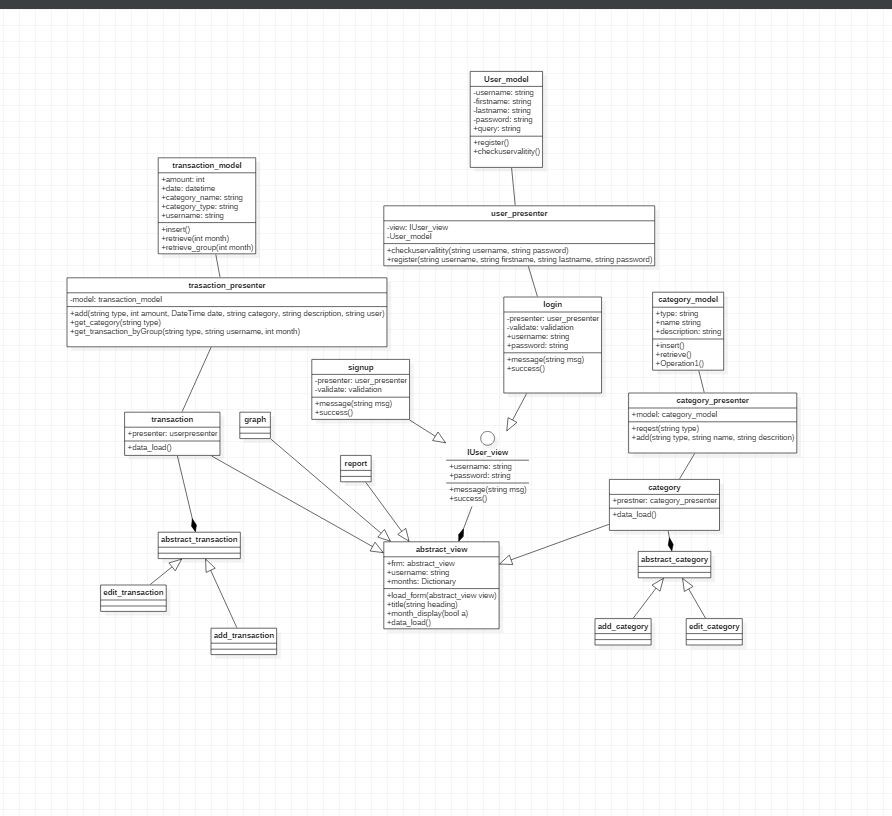


Figure 4: Class Diagram

### Flowchart diagram

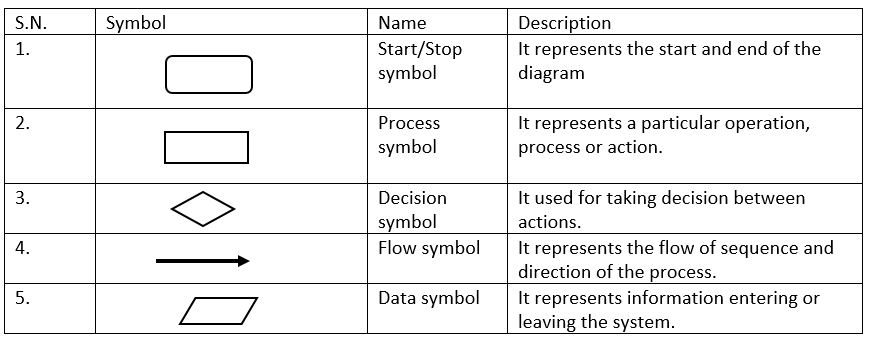
Flowchart is a representation of steps in graphical form. It shows the sequential flow of steps to represent the algorithm, workflow or processes.

**Justification**

The reason for using flowchart diagram are as follows:

1. Flowchart represents the overall logic of the system to all involved which improves communication.
2. The flow of process is shown in a systematic way.
3. All the process can be identified clearly

Notation used

 Table 6: notation used in flowchart

Diagram

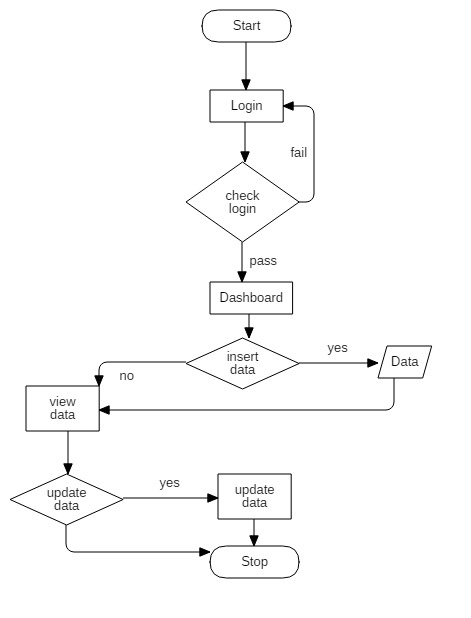


Figure 5: Flow chart diagram

The given diagram shows that user can get into the Dashboard after successful and valid login details. The user enters its data and start its transaction and can view, update their details.

## Behavioral Modeling

Behavioral model is a type of system model in which dynamic behavior of the system are shown how it responds to the system.

### Activity Diagram

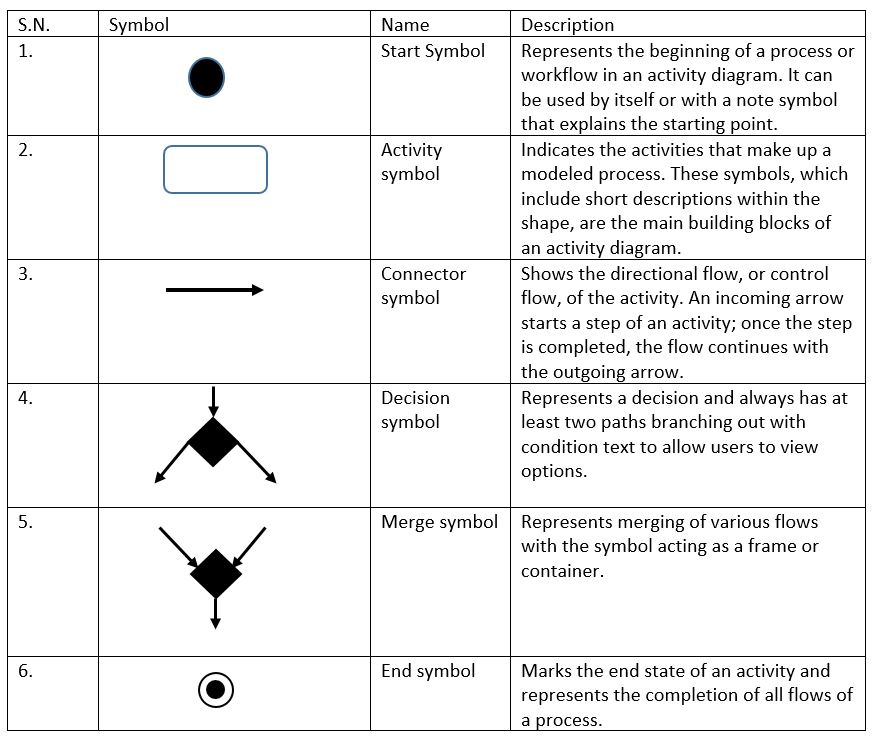
Activity diagram is a dynamic behavioral diagram which shows the activity involved in data processing or a process.

**Justification**

The reasons for using activity diagrams are as follows:

1. It displays the flow of activity between the user and the system
2. It interprets the parallel, concurrent and branched flow of the system
3. It can be used for analyzing a use case
4. It describes the sequence of an activity to another

Notation used

Table 7: Notation used in activity diagram

Diagram

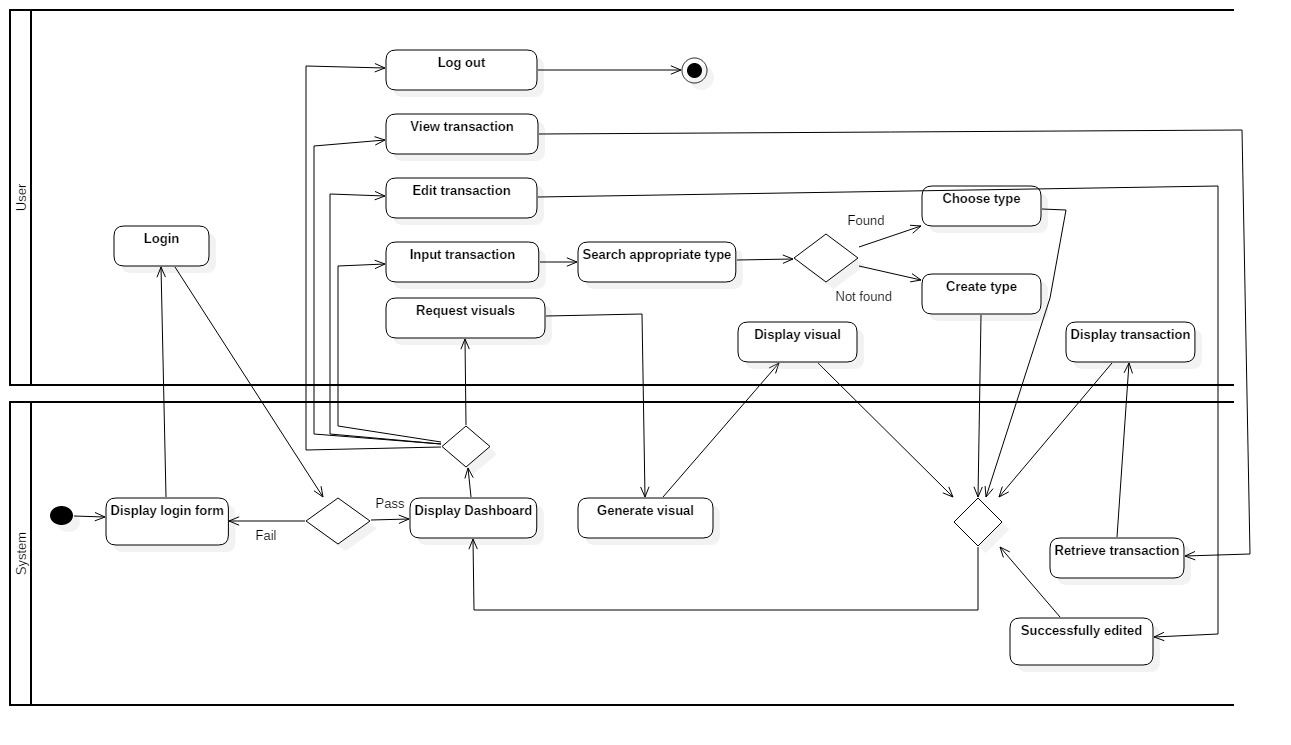


Figure 6: Activity diagram

The above diagram, at first the system displays the login form so that the user can login. If the user enters the right credentials, then the system will display else back to the login form. From the dashboard the user can either input transaction, view transaction, edit transaction, request visuals or logout. If the user wants to enter the transaction, then he\she must search for appropriate type. If found the user can choose it else, he\she can also create type. After this the system take the user back to the dashboard. If the user requests visuals, then the system first generate the visuals based on the data it has stored then display it to the user. After this again the system takes the user back to the dashboard. The user can also logout to terminate the system.

### Sequence Diagram

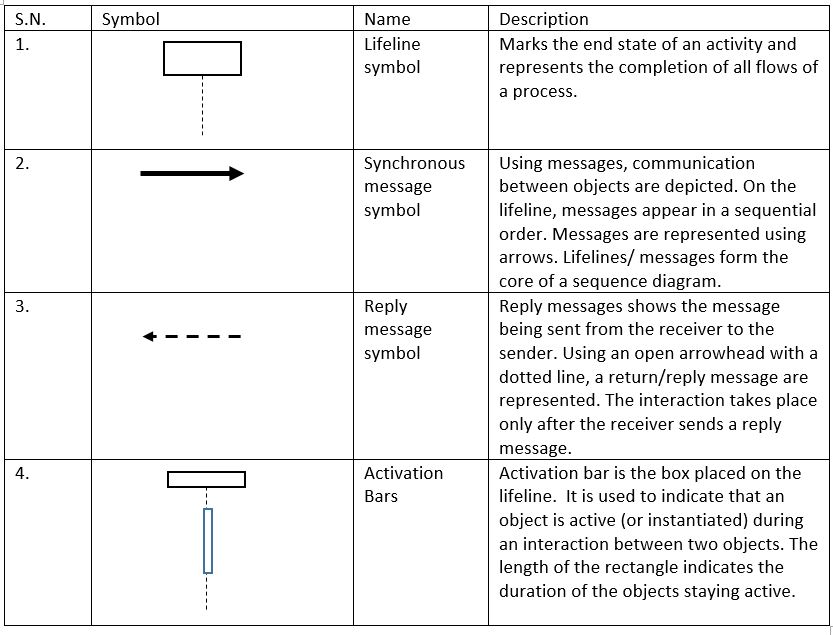
Sequence diagram is a dynamic behavioral diagram. It represents the interaction between actors, system and system components.

**Justification**

The reasons for using activity diagrams are as follows:

1. It models high-level interaction between the active objects in a system
2. Allows reverse engineering
3. Shows how objects and components interact with each other to complete a process

Notation used

 Table 8: Notation used in activity diagram

**Diagram**

**Login sequence**

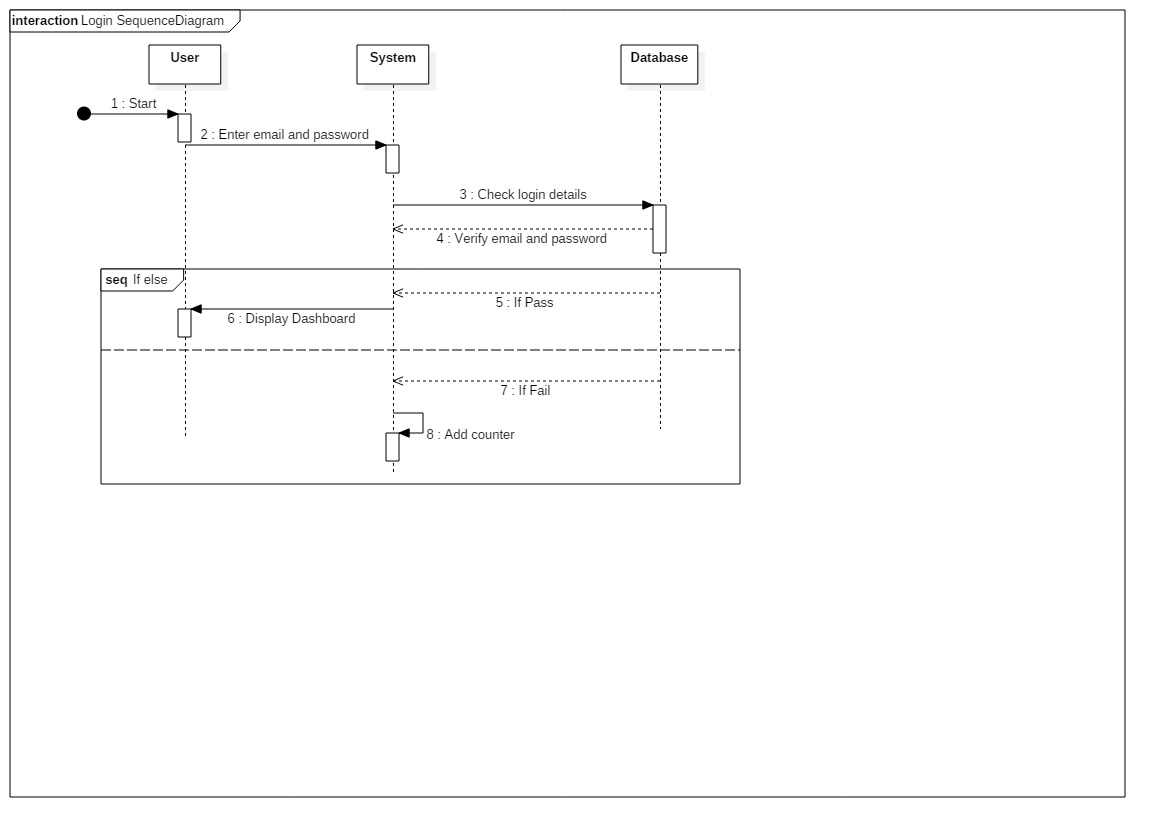


Figure 7: sequence diagram of login

The above diagram shows the login sequential flow done by user to the system. The user inserts the email and password and the login details data is check for validation in the database. If the input data is correct then the user dashboard is displayed but if the input data is invalid, then user need to enter the email and password to login.

**User sequence**

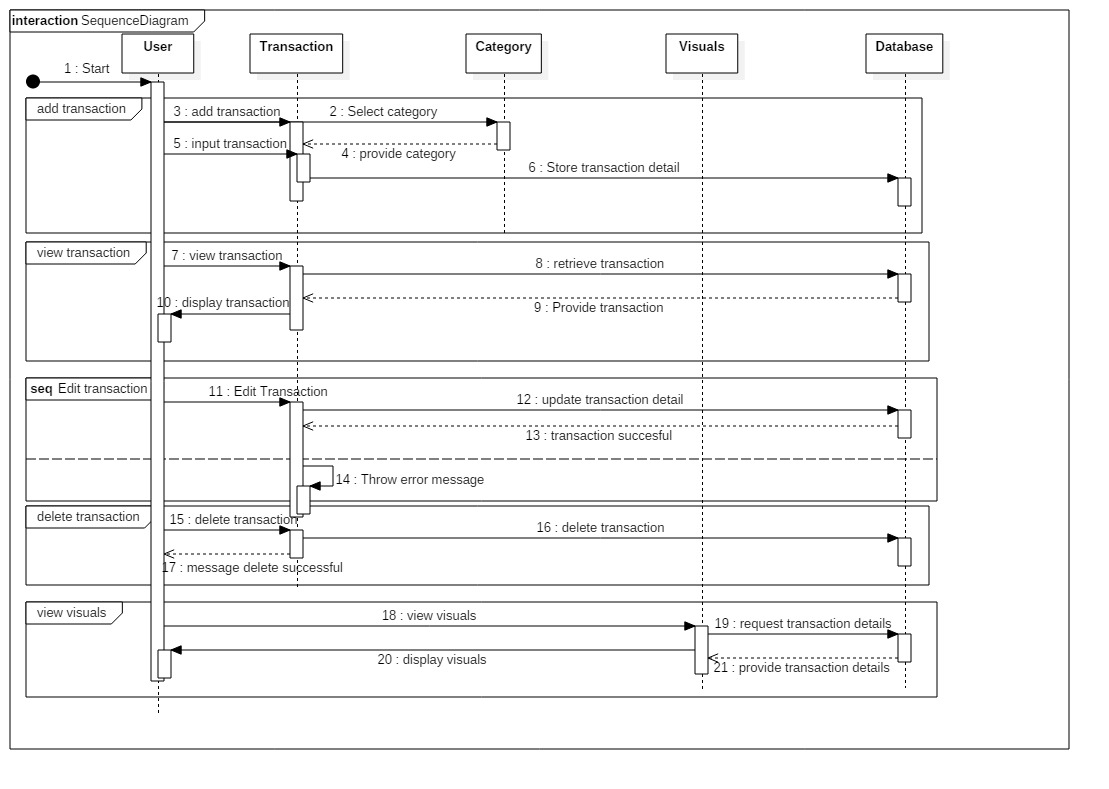


Figure 8: sequence diagram of system

The above diagram shows sequence of work done by the user. The user can add income or expense. They can also choose a category to the income or expense. Similarly, the user can also request graph or reports. The system fetches the expenses and income entered by the user then it generates a graph or report based on the data provided. User can add, update and delete transaction and store data in database. User has access to view transaction and also has authority to view visual by sending message to database after that database reply by providing transaction details.

## Database Modeling

### ER modelling

Entity Relationship (ER) modeling is graphical representation of database design. Entity is a real world object or thing that can be distinguished from the environment. The relation of these entity are represented in the ER diagram.

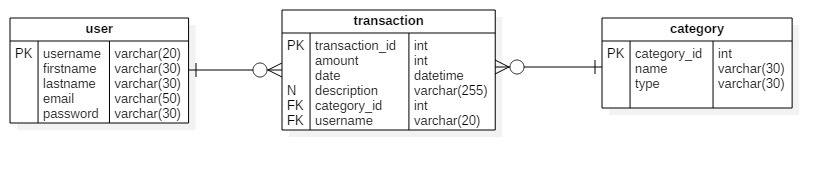
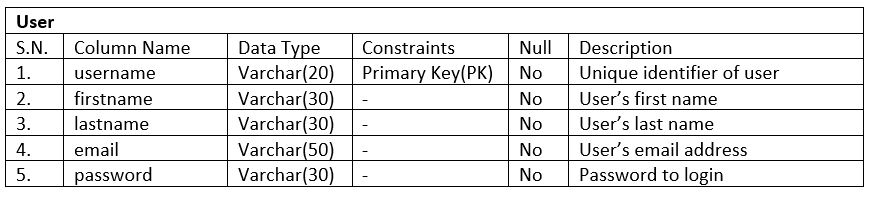
The ER diagram of this project is presented below

Figure 9: ER Diagram

### Data dictionary

Data dictionary is a file or a group of files in which metadata as well as data ownership, data relationship and other data are stored. The database user does not interact with data dictionary. It is invisible to the user but it is an important part of the relational database. Only the database administrator handles data dictionary.

The metadata of this project are listed below:

 Table 9: Data dictionary of user table

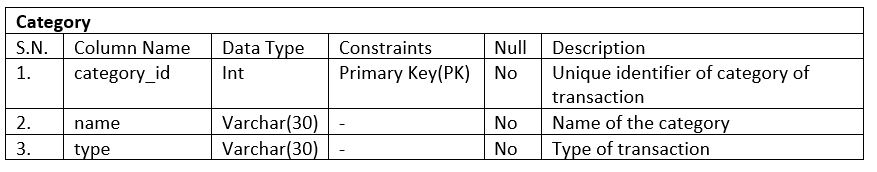


Table 10: Data dictionary of category table

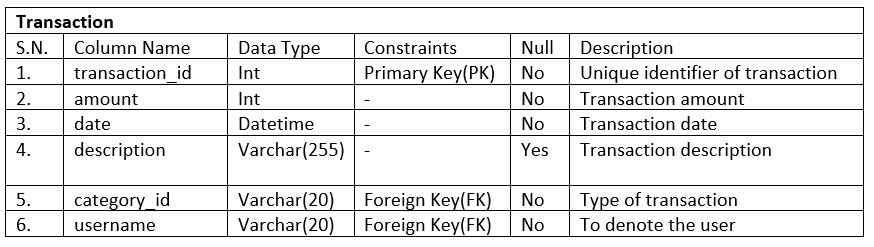


Table 11: Data dictionary of transaction table

## UI Modeling

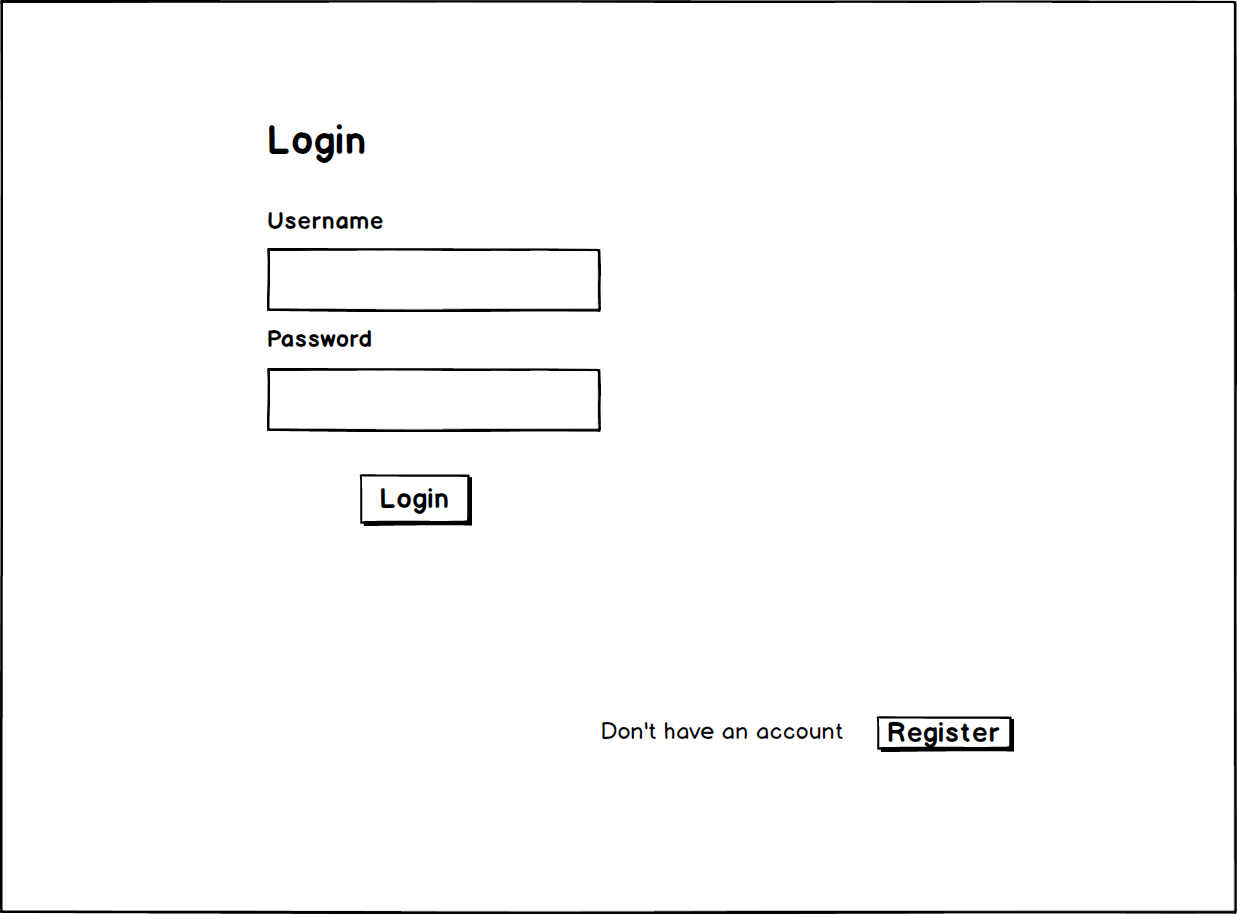


Figure 10: UI for login

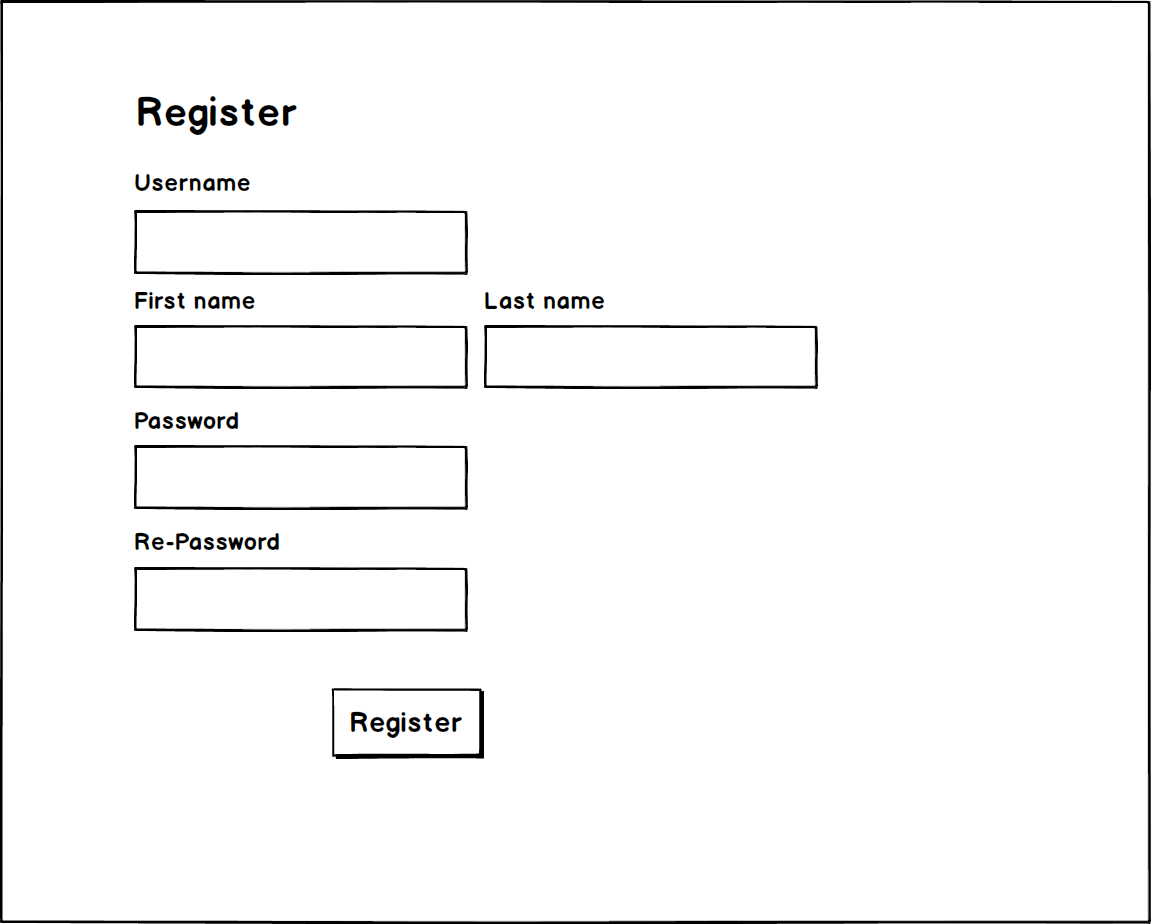


Figure 11: UI for register

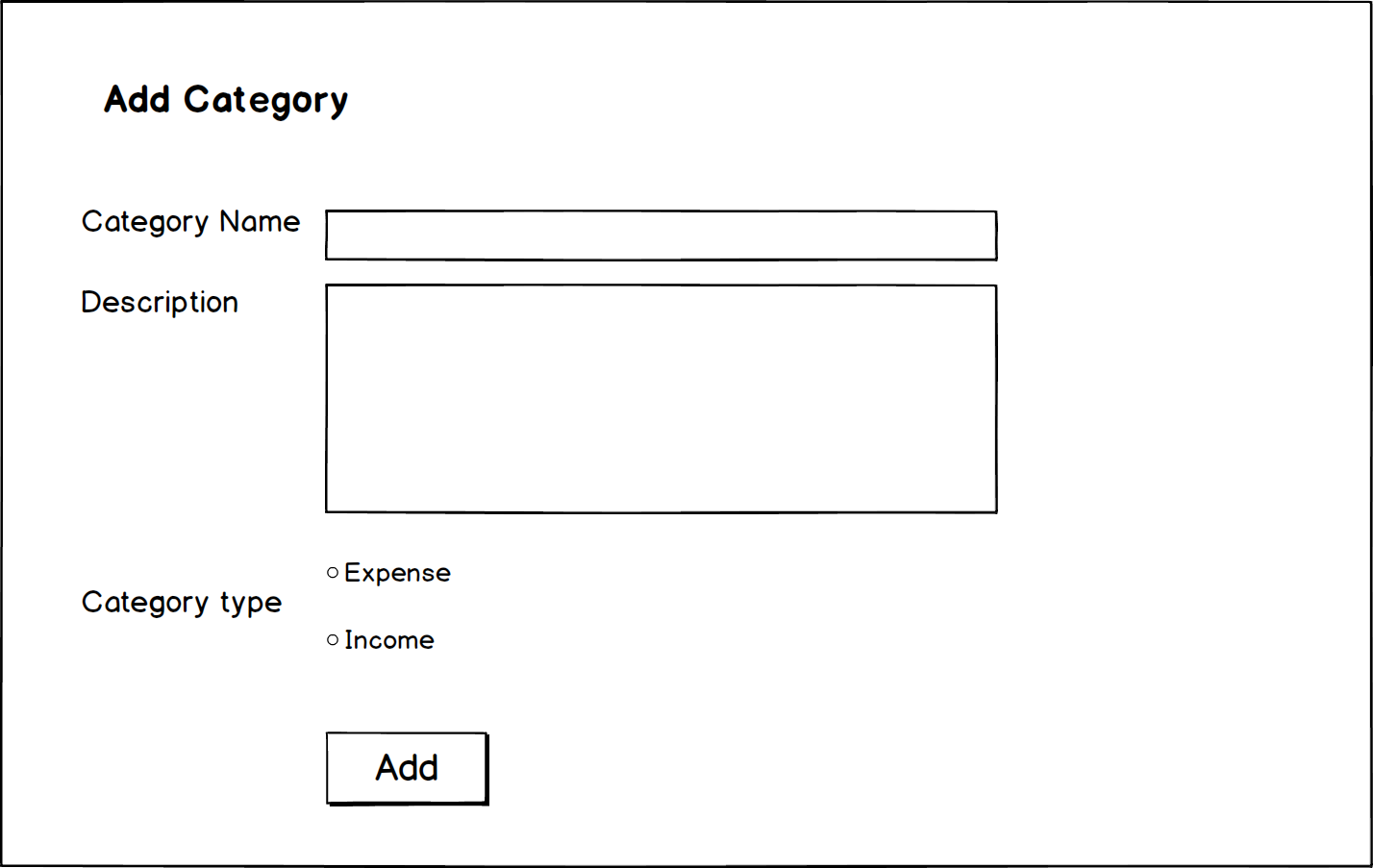


Figure 12: UI for add category

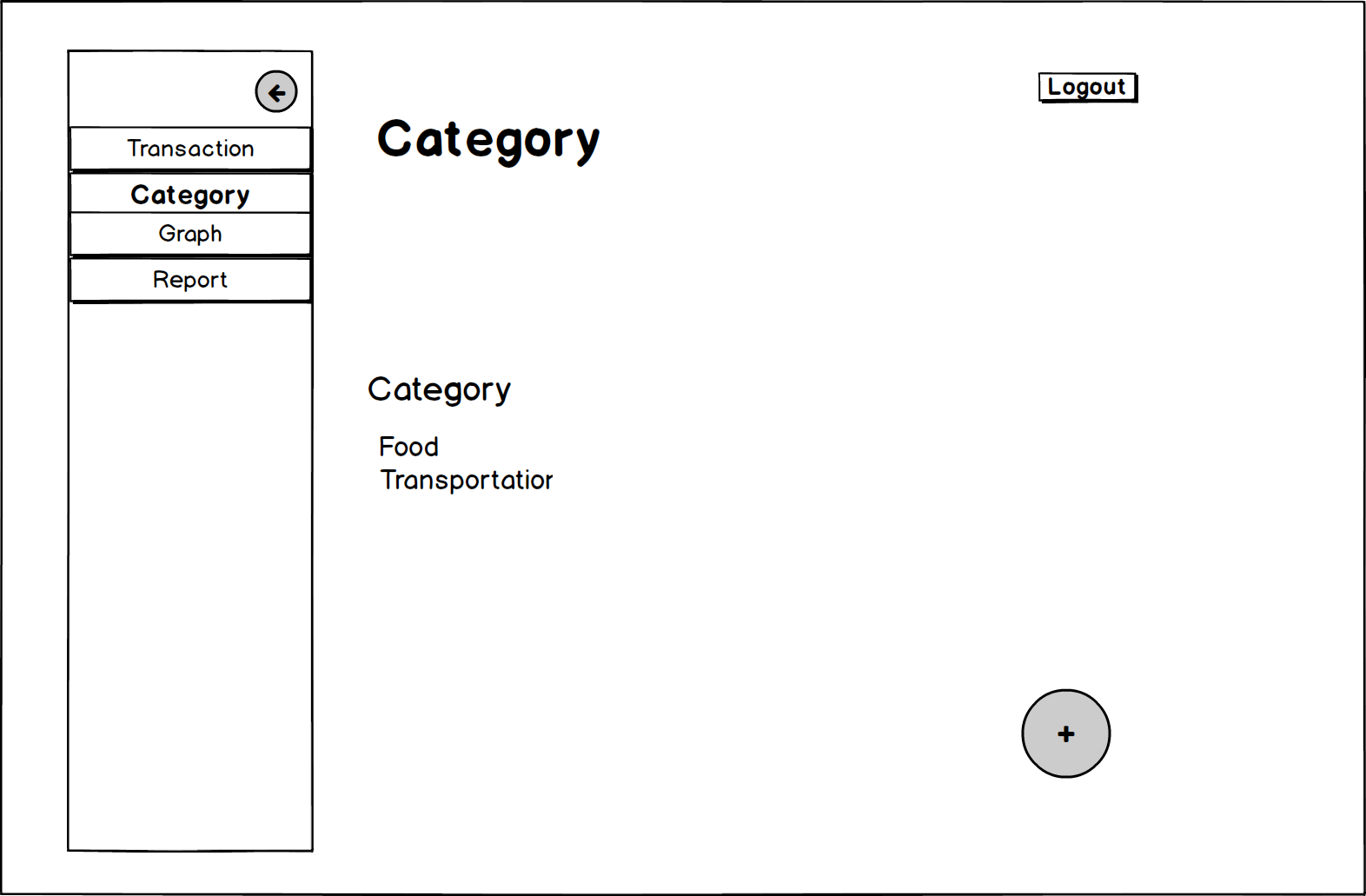


Figure 13: UI for category

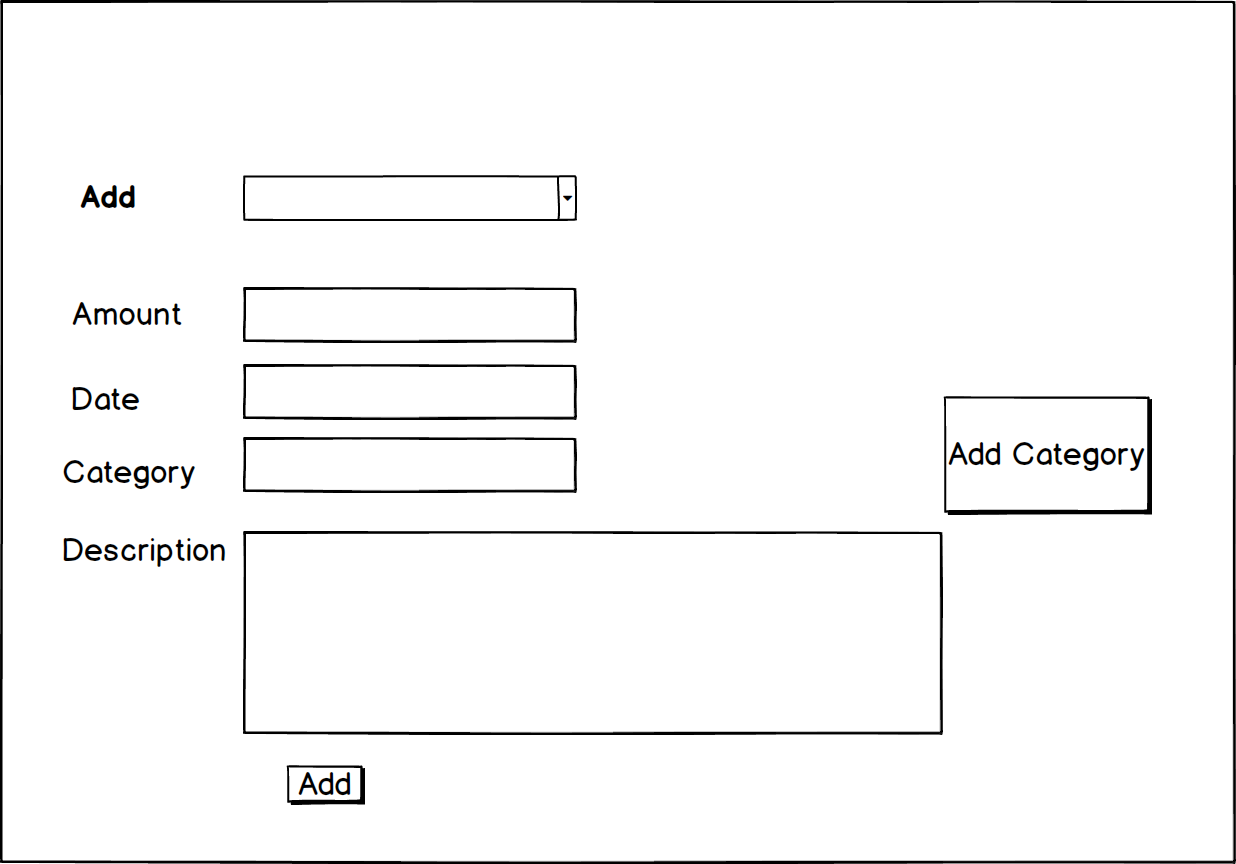


Figure 14: UI for add transaction

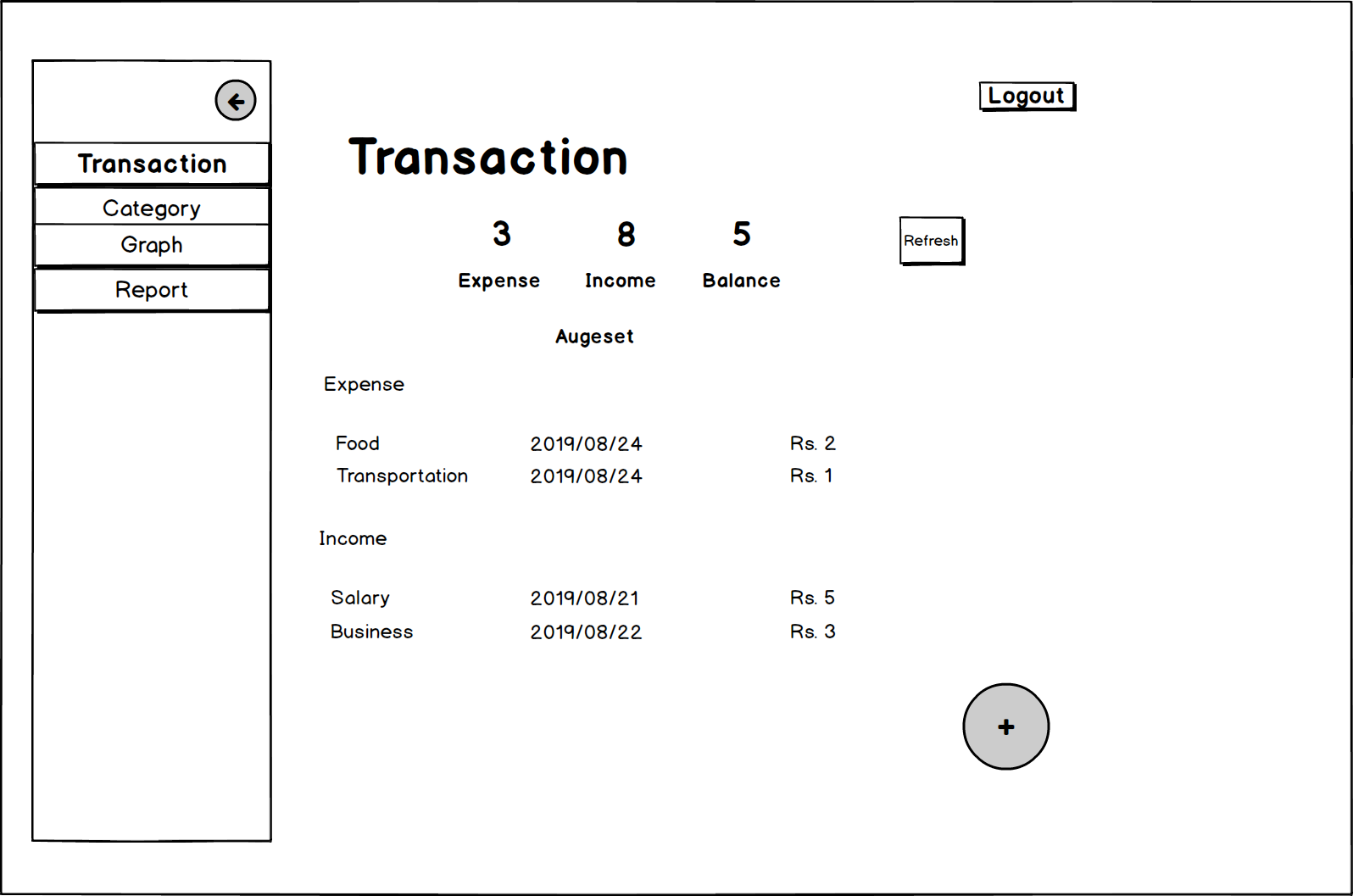


Figure 15: UI for transaction

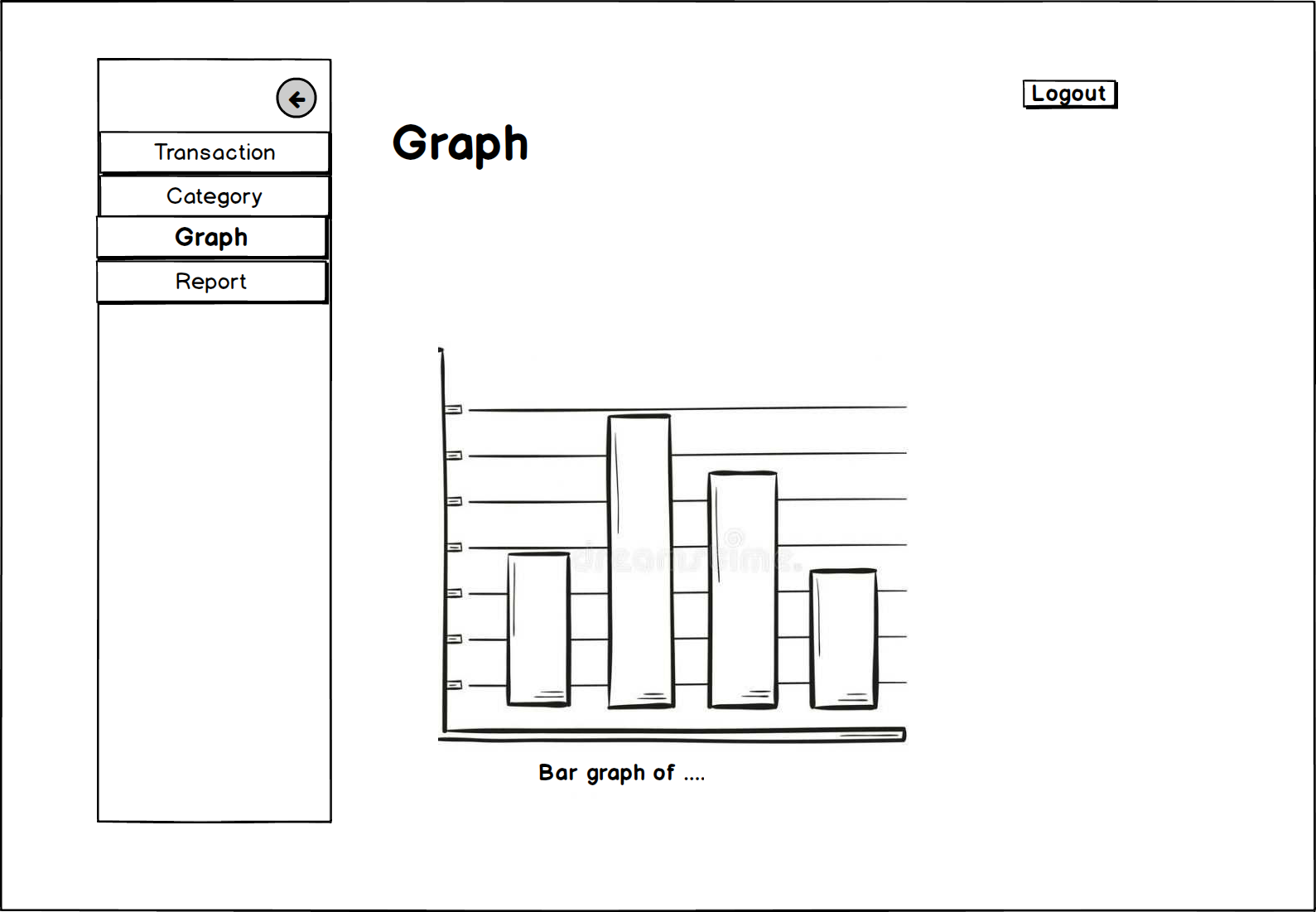


Figure 16: UI for graph

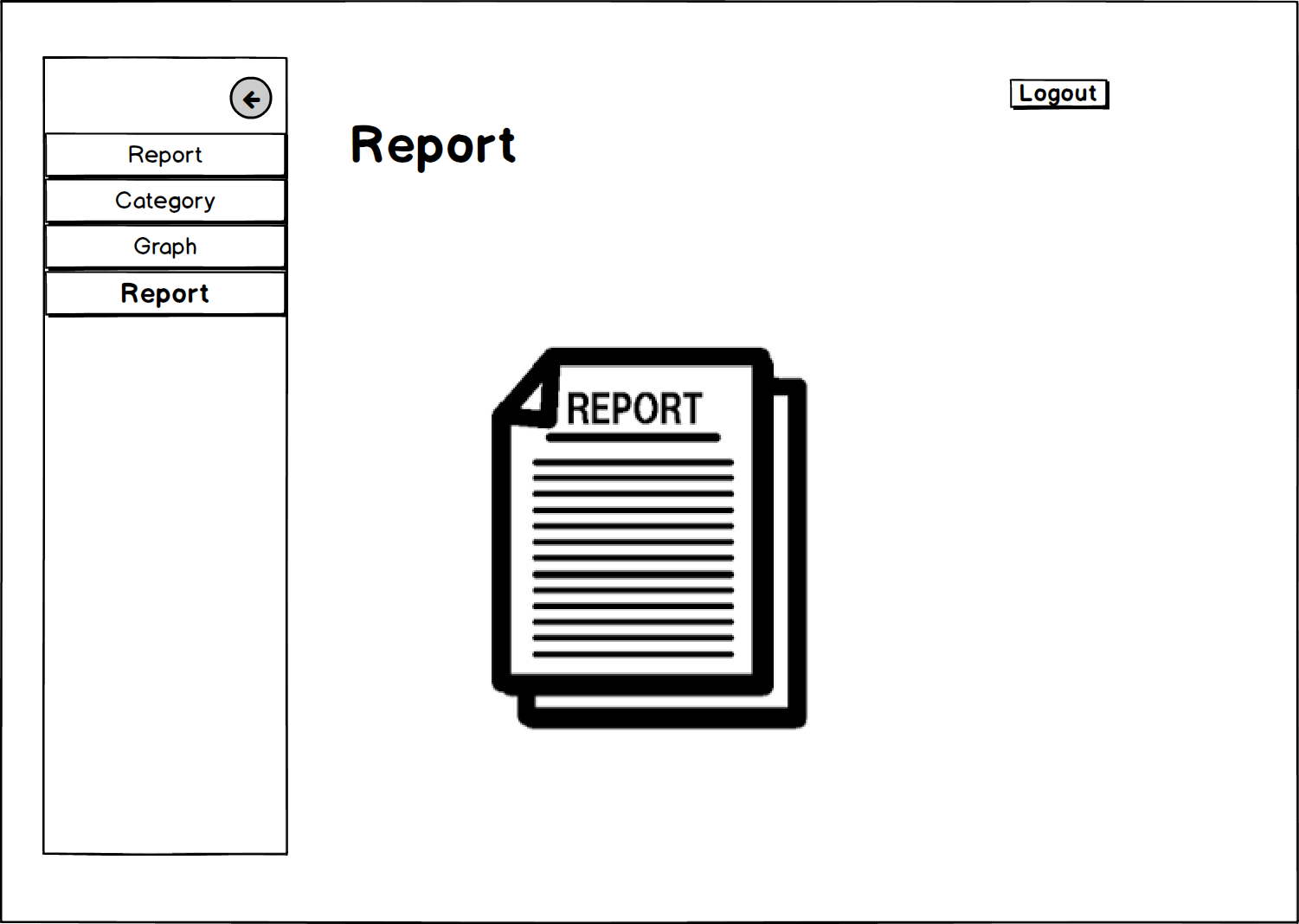


Figure 17: UI for report