|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Number** | **Test Case Type** | **Description** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Status** |
| **1** | Functionality | Moving a transaction to another category, so creating two new categories, both in the same month, and assigning a budget that exceeds their total combined and then creation of a transaction and assigning it to one of newly created categories and then the attempt to move it to the other category created. | Navigate to “Categories”, creating two new categories.  Then navigating to transaction and creating a new transaction.  Moving the transaction to another category.  Viewing all ongoing transactions. | Category label: “Essentials”.  Category budget: “400”.  Category date (MM/YYYY): “03/2024”  Category label: “Luxury”.  Category budget: “200”.  Category date (MM/YYYY): “03/2024”  Budget amount: “1000”  Transaction used: “Netflix Subscription”.  Transaction amount: “25”.  Transaction date (DD/MM/YYYY): “15/03/2024” | The creation of the categories should occur without fail and then the assigning of a budget to their retrospective month, creation of transaction and the changing of its category it belongs to should also pose no issues. | The program allowed for correct creation of categories and budget, then for the transaction, its creation proceeded as intended and changing its category worked as intended too, successful test. | Pass |
| **2** | Functionality | Display all ongoing transactions, initially, so creating transactions, assigning a category budget and a monthly budget, and checking the feature that displays all ongoing transactions. | Navigate to “Budget” and assign a budget, then navigate to “Category” and set a budget for a category and then create two transactions and then view them using display all ongoing transactions feature. | Budget for the month: “500”  Budget date (DD/MM/YYYY): “03/2024”  Category used: “Food”.  Category budget: “200”  Transaction used: “Water Bottles”.  Transaction amount: “8”.  Transaction date (DD/MM/YYYY): “01/03/2024”  Transaction used: “Restaurant”.  Transaction amount: “25”.  Transaction date (DD/MM/YYYY): “06/03/2024” | The program should let you create a budget for a month, let you change a category’s budget and then create more than a single transaction. Then finally seeing the transactions from the displaying of all ongoing transactions. | This actual result was a success, creation of budgets and assignment of category’s budget worked, so did the creation of the transactions and then the viewing of them too from the view all ongoing transactions feature. | Pass |
| **3** | Functionality | Displaying all transaction a certain category has and so creating a budget for a month and changing a preset category’s budget and then finally creating a transaction for the category we want to view transactions for and then using the “Category-wise Transactions” feature to test it. | Navigating to “Budget”, assigning a budget, then navigating to “Category” and allocating “Food” a budget.    Navigating back to categories menu and clicking on view “Category-wise transaction”. | Budget for the month: “500”  Budget date (DD/MM/YYYY): “03/2024”  Category used: “Food”.  Category budget: “200”  Transaction used: “Water Bottles”.  Transaction amount: “8”.  Transaction date (DD/MM/YYYY): “01/03/2024” | The program should let you create a budget and assign a budget to a preset category and then allow the creation of a transaction. Then the main point of showing all transactions a certain category has, and the category we used in the test data should contain the transaction we assigned to it and this should be display in the “Category-wise Transaction” viewing feature. | The test went exactly as planned, allowing correct creation and modifications of budget, category and transaction and then finally the viewing of a category’s transaction in which we created a transaction for, and it displayed exactly that. | Pass |
| **4** | Functionality | Allocation of budget to a category, so the point of this test is to make sure that the updating of any category’s budget can be updated, this is a needed test as the preset categories are set to “0” budget and so this test will ensure at a base level the program can run without needing to create a category. | Navigate to “Categories” and change the amount of preset category “Food” to “200”.  Then navigate to the view feature to see the change implemented. | Category budget amount: “200”. | The test should allow for changing of a preset category’s budget and then allow correct budget allocation for the current month, then viewing the categories list to see the change being made. | The test went as expected, allowing the changing of a preset category’s budget. | Pass |
| **5** | Functionality | Creation of a category that already exists should result in an error message, so same labelled names shouldn’t be allowed, so this test will involve creating a category that already exists (preset category). | Navigate to “Category” and create a category.  Then use the view categories feature. | Category label:  “Food”.  Category date (DD/MM/YY):  “03/2024”. | The test should come up with some form of error message saying, “duplicate value being used” or something like indicate name change must occur. | The test failed and did not go as expected, the program allows for creation of category that share the same label. | Fail |
| **6** | Functionality | viewing of budget feature, so checking the budget for a month, starting with assigning a budget for a month and then verifying it from the viewing feature. | Navigate to “Budget” and assign a budget for a date.  Then use the viewing feature to see budget for a month. | Budget amount: “350”.  Budget date (MM/YYYY): “01/2022”. | This test should allow you to set a budget for a certain date and then check to see what a certain date’s budget is, so confirming the budget initially set does work as intended. | The test went exactly as expected with the budget feature working accordingly to the test. | Pass |
| **7** | Functionality | Removal of a budget for a month, so assigning a budget for the current month, checking it was created and then removing that budget and checking it was removed, the point of this test is to verify the budget feature works as needed, as it is a core aspect of the program. | Navigate to “Budget” and assign a budget for a date.  Then use the viewing feature to see budget for a month.  Then using the remove feature to now remove it.  Then use the viewing feature to see its removal. | Budget amount: “350”.  Budget date (MM/YYYY): “01/2022”. | The program should allow easy and correct creation of a budget and the viewing of it and then also easy removal of a month’s budget and then the viewing of that month’s budget to see its removal having occurred. | The test went as planned, was creation and removal of the budget working and seeing verifying the creation and removal using the viewing feature also working as expected. | Pass |

Test Notes:

When wanting to view transactions etc. I think it is a good idea to only ask for month and year, instead of specific dates as you are going to display the entire month and not information on a specific day.

Test Screenshots

**Test number 1:**

Creation of category “Essentials”:

A screenshot of a computer program

Description automatically generated

Creation of category “Luxury”:

A screenshot of a computer program

Description automatically generated

Creation of budget:

A screenshot of a computer program

Description automatically generated

Creation of transaction, assigning to “Essentials” category:

A screenshot of a computer screen

Description automatically generated

Changing of category for created transaction from “Essentials” category to “Luxury” category:

A screenshot of a computer screen

Description automatically generated

Viewing all on going transactions to see the change:

A screenshot of a computer

Description automatically generated

**Test number 2:**

Assigning a budget for the month to be used:

A screenshot of a computer program

Description automatically generated

Assigning a budget to the category to be used:

A screen shot of a computer

Description automatically generated

Creating transactions:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Displaying all on going:

A screenshot of a computer

Description automatically generated

**Test number 3:**

Creating a budget for the date of “03/2024”:

A screenshot of a computer program

Description automatically generated

Changing the preset category “Food”’s budget:

A screen shot of a computer

Description automatically generated

Creating transaction:

A screenshot of a computer

Description automatically generated

Viewing “Category-wise Transactions”:

A screenshot of a computer

Description automatically generated

**Test number 4:**

Viewing preset categories:

A screenshot of a computer program

Description automatically generated

Changing the amount for a category’s budget. Assigning “Food” a budget of “200”:

A screen shot of a computer

Description automatically generated

Viewing that change in categories list:

A computer screen shot of a black screen

Description automatically generated

**Test number 5:**

Creating a category:

A screenshot of a computer program

Description automatically generated

Viewing of categories list:

A computer screen shot of a black screen

Description automatically generated

**Test number 6:**

Creating a budget:

A screen shot of a computer

Description automatically generated

Viewing that budget created:

A screenshot of a computer program

Description automatically generated

**Test number 7:**

Creating a budget:

A screen shot of a computer

Description automatically generated

Viewing that budget created:

A screenshot of a computer program

Description automatically generated

Removal of that budget:

A screenshot of a computer program

Description automatically generated

Viewing budget for the initial budget assigned date to see if it exists or not:

A screenshot of a computer program

Description automatically generated

Notes