Finance Tracker

Requirements and Test Document

Pablo Bandera Lopez

21 April 2025

CS 225, Spring 25

Embry-Riddle Aeronautical University

Daytona Beach campus

1 Aerospace Boulevard

Daytona Beach, FL 32114

INTRODUCTION:

This document will detail the various requirements and test cases needed to have a fully functional application. This program will be a personal monthly finance tracker where the user will be able to create entries, edit entries throughout the month and view all entries that have been entered. Further details can be found in the Software Design Document.

The requirements and test cases that follow break down the application into testable units. Some requirements and test cases are designed to test program reliability and adequate function of the back-end code. Other requirements and test cases are designed to test the user interface and its ability to interact with the back-end program. Both of these types of test cases requirements are necessary for the application to function.

BACKGROUND INFORMATION:

As previously stated, further information about the specific categories, sub-categories, accounts, types, etc. Is available in the software design document. Refer to tables 1-6 to see the list of available selections and when they should become available.

REQUIREMENTS:

Table 1: Requirement Specifications

ID	Requirement Specification
1	As a tester I want to create new instance of Entry.
	1.1: A new instance can only be created when a valid entryID, entryID>0, is passed in the constructor. 1.2: A new instance of Entry shall be initialized by setting all its values to –1 and amount to 0.0. 1.3: The dollar amount, type, category, subcategory1-4, account, attributes shall only be settable to a value greater than 0 after being initialized. An InvalidEntryException shall be thrown if attempted.
2	As a tester I want to group entries together in collection class Entries.
	2.1: A new Entry may only be added if the entry is valid. 2.2: An Entry shall be able to be removed based on and entryNumber. 2.3: An Entry shall be able to be found by entryNumber.
3	As a tester I want to create new instance of EntryFrame.
	3.1: The program shall only display selection menus for dollarAmount, type, account, an area for a comment, and the submit or cancel buttons if instantiated with a null entry.

	3.2: The program shall load the Entry data if instantiated and display all the previously selected and entered values for the user to change. 3.3: The program shall add the Entry when the submit button is pressed only if the Entry contains valid information.
4	As a tester I want to be able to conduct file operations after instantiating a FileManager.
	4.1: The program shall name a new file Month.txt and it should be located in the directory ./Files/YYYY/Month.txt where YYYY is the current numerical year, and Month is the current month's name. 4.2: The program shall be able to print a series of entries onto the file. 4.3: The program shall be able to read the entries in a file, parse them into Entry objects and return an Entries object with all the file Entry objects in it.
5	As a tester I want to be able to translate integers into readable strings using an instance of Decoder.
	5.1: The program can take a String read from a file and decode it into a textual representation. 5.2: The program will determine what account was charged and represent it in a text version. 5.3: The program will only take the relevant information from an Entry.
6	As a tester I want to be able to instantiate a DisplayFrame.
	6.1: The program shall display all, non-null, entries stored in a file with a neat table. 6.2: The program shall translate the raw entries using the Decoder class. 6.3: The program shall display the total income, total spent, and remaining for each available account.
7	As a user I want to be able to create a new Entry by instantiating EntryFrame and store it locally.
	7.1: The program will display an error message when attempting to create a New Entry and the month file has not been created. 7.2: The program will create a new instance of New Entry. 7.3: The program will change the if the New Entry frame to match the entry number if the Entries list is not empty.
8	As a user I want to be able to edit an instance of Entry stored in Entries by instantiating EntryFrame and update it in the storage.
	8.1: The program will ask for an entry number when Edit Entry is clicked. If the entry number does not exist a pop-up message will notify the user. 8.2: The program will ask for an entry number when Edit Entry is clicked. If the entry number exists a new instance of Entry Frame will be created and load the entry. 8.3: The program will notify the user using a pop-up message if there is no file.
9	As a user I want to be able to delete an instance of Entry from all storage.
	9.1: The program will ask for an entry number to delete. 9.2: The program will search for an Entry by number and if found remove it from the Entries collection class and update the file.

	9.3: The program will search for an Entry by number and if not found notify the user using a pop-up.
10	As a user I want to be able to view all instances of Entry stored in the current month's File by instantiating Display Frame.
	10.1: The program will notify the user using a pop-up when no current file exists. 10.2: The program will load and display each Entry stored in the current file. 10.3: The program will calculate the total income, total spent, and remaining for each account, for the month.
11	As a user I want to be able to view all instances of Entry stored in a selected month's File by instantiating Display Frame.
	11.1: The program will prompt the user to select the desired year and month using a series pf pop-up dropdown. 11.2: The program will load the corresponding file if it exists and display it in a DisplayFrame. 11.3 The program will total the income, expense, and remaining using the data in the file.
12	As a user I want to be able to delete all data for the current month.
	12.1: The program will remove the current month file if it exists. 12.2: The program will notify the user using a pop-up if the file does not exist. 12.3: The program will clear the local Entries array so it does not store the local data to the file.
13	As a user I want to be able to create a File for the new month.
	13.1: The program will create a new file for the current month if one does not exist. 13.2: The program will prompt the user to override the current file if one exists. 13.3: The program will create the directory if it does not exist.

TEST CASES:

Table 1: Test Case Summary

User Story ID	Requirement ID	Test Case ID	Date	Status Pass/Fail/Pending
1	1.1	1.1	03.28.25	PASS
1	1.2	1.2	03.28.25	PASS
1	1.3	1.3	03.28.25	PASS
2	2.1	2.1	03.28.25	PASS
2	2.2	2.2	03.28.25	PASS
2	2.3	2.3	03.28.25	PASS
3	3.1	3.1	03.28.25	PASS
3	3.2	3.2	03.28.25	PASS
3	3.3	3.3	03.28.25	PASS
4	4.1	4.1	03.31.25	PASS

4	4.2	4.2	03.31.25	PASS
4	4.3	4.3	03.31.25	PASS
5	5.1	5.1	03.31.25	PASS
5	5.2	5.2	03.31.25	PASS
5	5.3	5.3	03.31.25	PASS
6	6.1	6.1	04.07.25	PASS
6	6.2	6.2	04.07.25	PASS
6	6.3	6.3	04.07.25	PASS
7	7.1	7.1	04.01.25	PASS
7	7.2	7.2	04.01.25	PASS
7	7.3	7.3	04.01.25	PASS
8	8.1	8.1	04.01.25	PASS
8	8.2	8.2	04.01.25	PASS
8	8.3	8.3	04.01.25	PASS
9	9.1	9.1	04.01.25	PASS
9	9.2	9.2	04.01.25	PASS
9	9.3	9.3	04.01.25	PASS
10	10.1	10.1	04.07.25	PASS
10	10.2	10.2	04.07.25	PASS
10	10.3	10.3	04.07.25	PASS
11	11.1	11.1	04.07.25	PASS
11	11.2	11.2	04.07.25	PASS
11	11.3	11.3	04.07.25	PASS
12	12.1	12.1	04.01.25	PASS
12	12.2	12.2	04.01.25	PASS
12	12.3	12.3	04.01.25	PASS
13	13.1	13.1	04.01.25	PASS
13	13.2	13.2	04.01.25	PASS
13	13.3	13.3	04.01.25	PASS

Table 2: Test Case Template and Results

Test Case ID: 1.1	**Current Status: PASS**	**Date:** 03.28.25
Req. ID: 1.1 A new instance can only be created when a valid entryID, entryID>0, is passed in the constructor.		

Step #	**Operator Action**	**Expected Results**	**Comments**
1	In Entry Class /*Test Main*/ instantiate a new Entry object and pass it int 0.	Program should throw an InvalidEntryException with message: Invalid Entry Number.	
2	In Entry Class /*Test Main*/ instantiate a new Entry object and pass it int 1.	Program should print the new entry instance in the format: id,0.0,-1,-1,-1,-1,-1,-1,-1,-1, This format represents an empty entry.	

Screenshots:

Test Case ID: 1.2	**Current Status: pending**	**Date:** 03.28.25
Req. ID: A new instance of Entry shall be initialized by setting all its values to –1 and amount to 0.0.		

Step #	Operator Action	Expected Results	Comments
1	In Entry Class /*Test Main*/ instantiate a new Entry object and pass it int > 0.	Program should print the new entry instance in the format: id,0.0,-1,-1,-1,-1,-1,-1,-1, This format represents an empty entry.	

Screenshots:

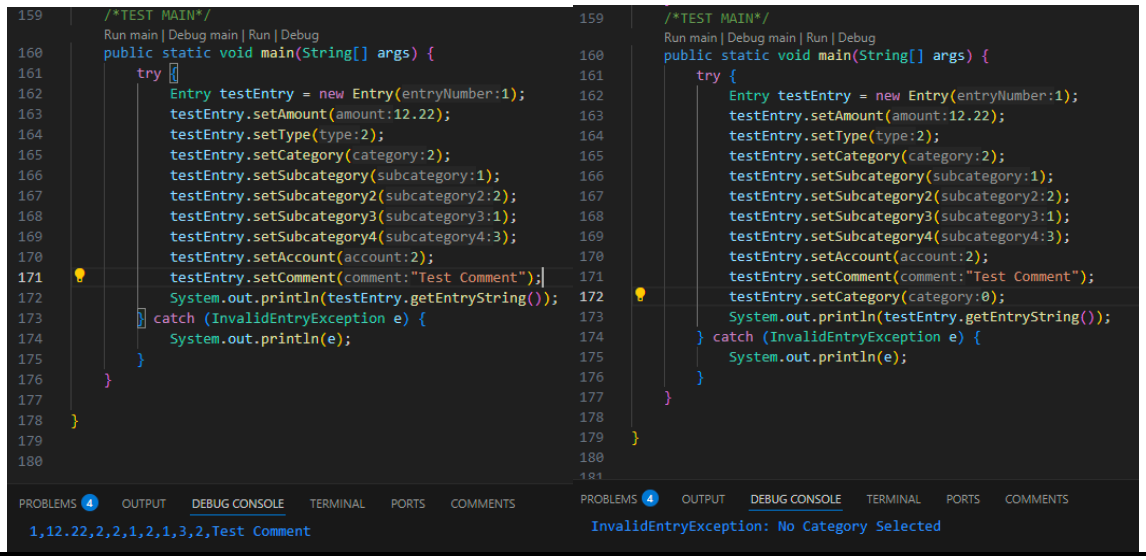
Test Case ID: 1.3	**Current Status: PASS**	**Date:** 03.28.25

Req. ID: 1.3 The dollar amount, type, category, subcategory1-4, account, attributes shall only be settable to a value greater than 0 after being initialized. An InvalidEntryException shall be thrown if attempted.

Step #	**Operator Action**	**Expected Results**	**Comments**
1	In Entry Class /*Test Main*/ instantiate a new Entry object and pass it int 1. Use the setter methods to set all parameters to a value greater than 0 or non-empty string.	Program should print the new entry instance in the format: id,dollarAmount,type,category,subcategory,subcategory2,subcategory3,subcategory4,accout,comment	
2	In Entry Class /*Test Main*/ instantiate a new Entry object and pass it int 1. Use the setter methods to set all parameters to a value greater than 0 or non-empty string. Using one of the setter	Program should throw and InvalidEntryException and print the corresponding error message. Example: InvalidEntryException: No Category Selected	

	methods set a value of 0 for any of the elements. Example: testEntry.setCategory(0);		

Screenshots:



```
/*TEST MAIN*/
Run main | Debug main | Run | Debug
public static void main(String[] args) {
    try {
        Entry testEntry = new Entry(entryNumber:1);
        testEntry.setAmount(amount:12.22);
        testEntry.setType(type:2);
        testEntry.setCategory(category:2);
        testEntry.setSubcategory(subcategory:1);
        testEntry.setSubcategory2(subcategory2:2);
        testEntry.setSubcategory3(subcategory3:1);
        testEntry.setSubcategory4(subcategory4:3);
        testEntry.setAccount(account:2);
        testEntry.setComment(comment:"Test Comment");
        System.out.println(testEntry.getEntryString());
    } catch (InvalidEntryException e) {
        System.out.println(e);
    }
}
}
```

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

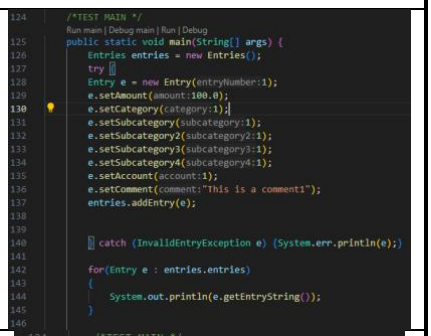
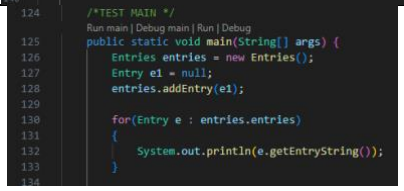
```
1,12.22,2,2,1,2,1,3,2,Test Comment
```

```
/*TEST MAIN*/
Run main | Debug main | Run | Debug
public static void main(String[] args) {
    try {
        Entry testEntry = new Entry(entryNumber:1);
        testEntry.setAmount(amount:12.22);
        testEntry.setType(type:2);
        testEntry.setCategory(category:2);
        testEntry.setSubcategory(subcategory:1);
        testEntry.setSubcategory2(subcategory2:2);
        testEntry.setSubcategory3(subcategory3:1);
        testEntry.setSubcategory4(subcategory4:3);
        testEntry.setAccount(account:2);
        testEntry.setComment(comment:"Test Comment");
        testEntry.setCategory(category:0);
        System.out.println(testEntry.getEntryString());
    } catch (InvalidEntryException e) {
        System.out.println(e);
    }
}
}
```

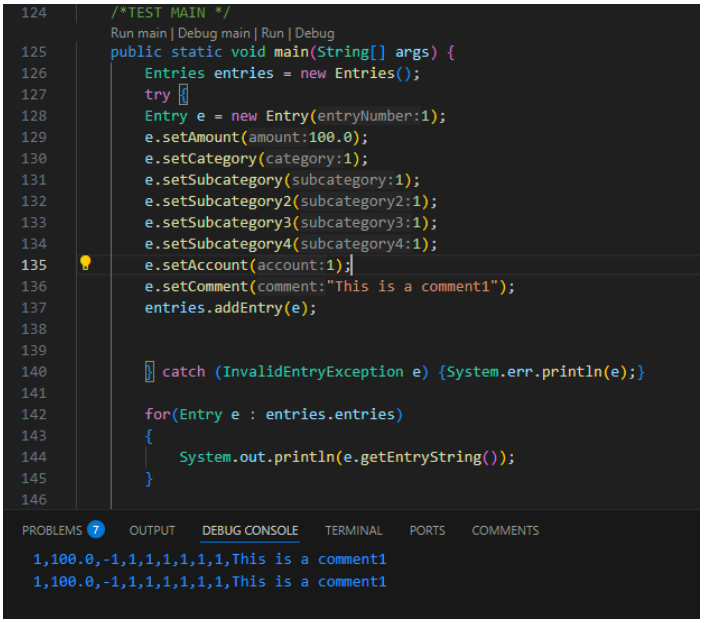
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
InvalidEntryException: No Category Selected
```

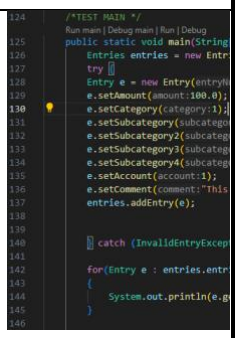
Test Case ID: 2.1	**Current Status: PASS**	**Date:** 03.28.25
Req. ID: 2.1 A new Entry may only be added if the entry is valid. Valid is not null.		

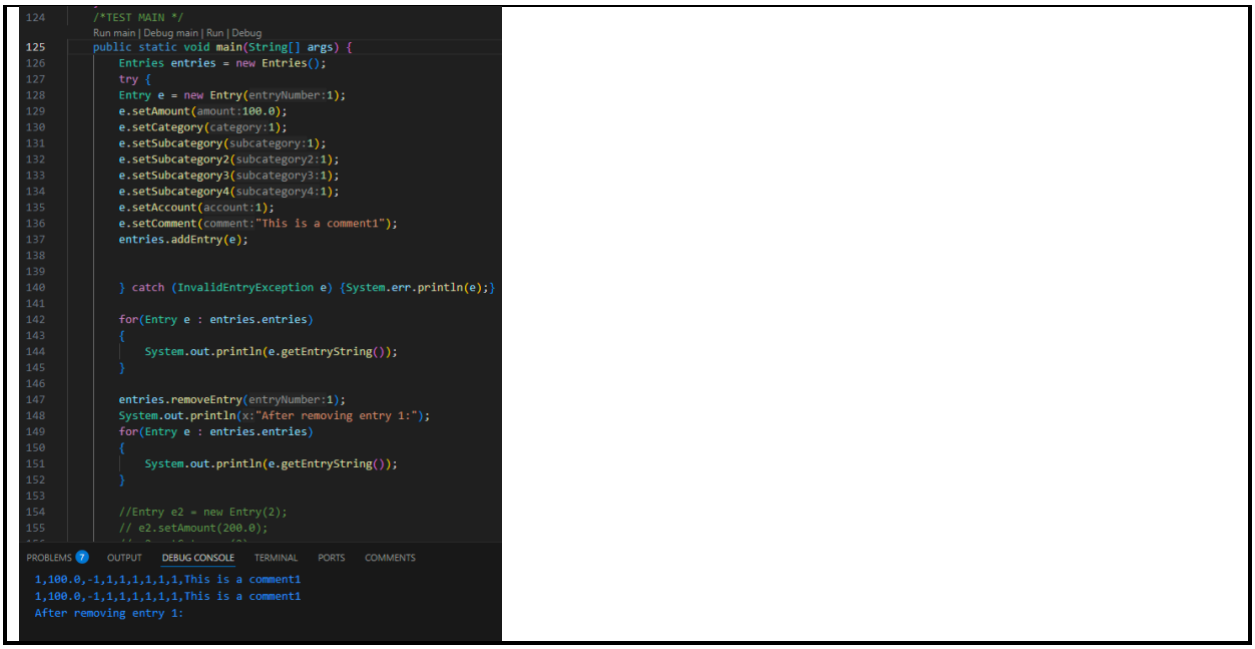
Step #	Operator Action	Expected Results	Comments
1	In Entries Class /*TEST MAIN*/ instantiate a new Entries object. Instantiate a new Entry object: e1 and set its data. Add e1 to entries.	The Entry object e1 should be added to the entries attribute.	
2	In Entries Class /*TEST MAIN*/ instantiate a new Entries object. Instantiate a new Entry object: e1 and set it equal to null.	The Entry object e1 should be rejected from the entries.	

Screenshots:



Test Case ID: 2.2	**Current Status: pending**	**Date:** 03.28.25
Req. ID: 2.2 An Entry shall be able to be removed based on and entryNumber.		

Step #	**Operator Action**	**Expected Results**	**Comments**
1	In Entries Class /*TEST MAIN*/ instantiate a new Entries object. Instantiate a new Entry object: e1 and set its data. Add e1 to entries.	Program should print the new entry instance in the format: id,dollarAmount,type,category,subcategory,subcategory2,subcategory3,subcategory4,accout,comment	
2	Use the removeEntryByNumber() function to remove the entry by ID.	Program should print the rest of the entries in the array in this case nothing should be printed.	
Screenshots:			

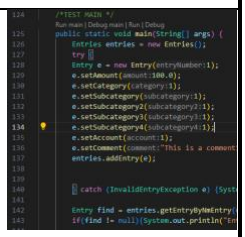


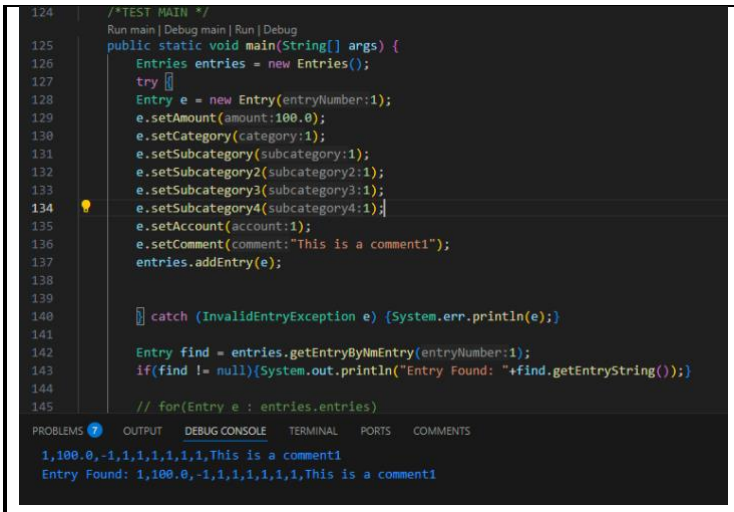
```
/*TEST MAIN */
Run main | Debug main | Run | Debug
public static void main(String[] args) {
    Entries entries = new Entries();
    try {
    Entry e = new Entry(entryNumber:1);
    e.setAmount(amount:100.0);
    e.setCategory(category:1);
    e.setSubcategory(subcategory:1);
    e.setSubcategory2(subcategory2:1);
    e.setSubcategory3(subcategory3:1);
    e.setSubcategory4(subcategory4:1);
    e.setAccount(account:1);
    e.setComment(comment:"This is a comment1");
    entries.addEntry(e);


    } catch (InvalidEntryException e) {System.err.println(e);}

    for(Entry e : entries.entries)
    {
        System.out.println(e.getEntryString());
    }

    entries.removeEntry(entryNumber:1);
    System.out.println(x:"After removing entry 1:");
    for(Entry e : entries.entries)
    {
        System.out.println(e.getEntryString());
    }

    //Entry e2 = new Entry(2);
    // e2.setAmount(200.0);
```

PROBLEMS 7 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
1,100.0,-1,1,1,1,1,1,1,1,This is a comment1
1,100.0,-1,1,1,1,1,1,1,1,This is a comment1
After removing entry 1:
```

Test Case ID: 2.3	**Current Status: PASS**	**Date:** 03.28.25
Req. ID: An Entry shall be able to be found by entryNumber.		

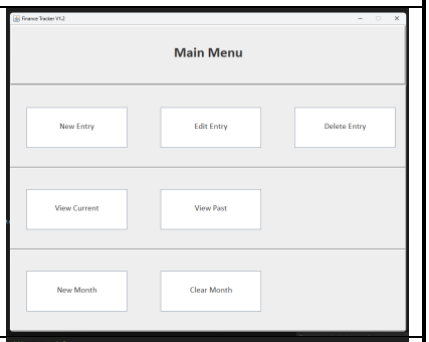
Step #	**Operator Action**	**Expected Results**	**Comments**
1	In Entries Class /*TEST MAIN*/ instantiate a new Entries object. Instantiate a new Entry object: e1 and set its data. Add e1 to entries.	Program should print the new entry instance in the format: id,dollarAmount,type,category,subcategory,subcategory2,subcategory3,subcategory4,accout,comment	
2	Use the getEntryByNmEntry function with parameter 1.	Program should return the entry and print it.	
Screenshots:			



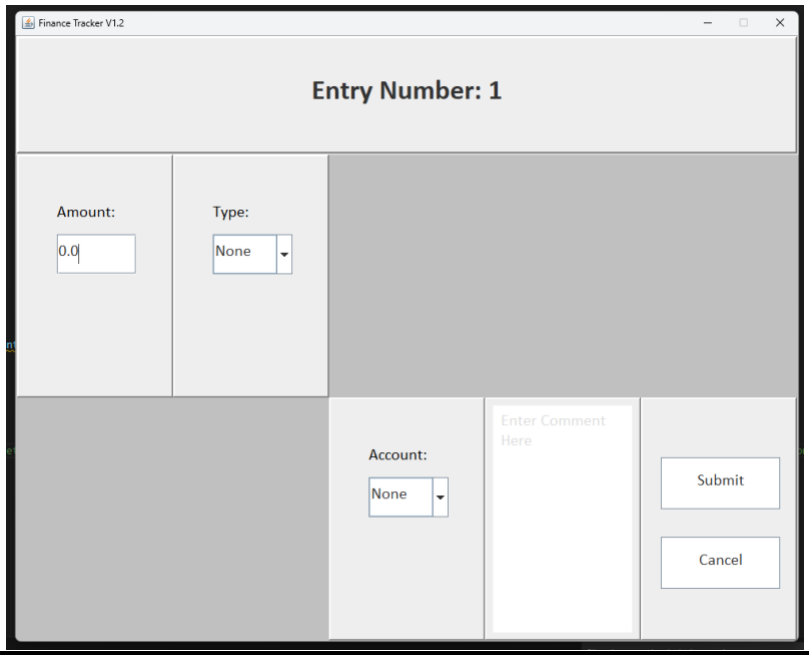
```
124     /*TEST MAIN */
        Run main | Debug main | Run | Debug
125     public static void main(String[] args) {
126         Entries entries = new Entries();
127         try {
128         Entry e = new Entry(entryNumber:1);
129         e.setAmount(amount:100.0);
130         e.setCategory(category:1);
131         e.setSubcategory(subcategory:1);
132         e.setSubcategory2(subcategory2:1);
133         e.setSubcategory3(subcategory3:1);
134         e.setSubcategory4(subcategory4:1);
135         e.setAccount(account:1);
136         e.setComment(comment:"This is a comment1");
137         entries.addEntry(e);
138
139
140         } catch (InvalidEntryException e) {System.err.println(e);}
141
142         Entry find = entries.getEntryByNmEntry(entryNumber:1);
143         if(find != null){System.out.println("Entry Found: "+find.getEntryString());}
144
145         // for(Entry e : entries.entries)
```

PROBLEMS 7 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
1,100.0,-1,1,1,1,1,1,1,1,This is a comment1
Entry Found: 1,100.0,-1,1,1,1,1,1,1,1,This is a comment1
```

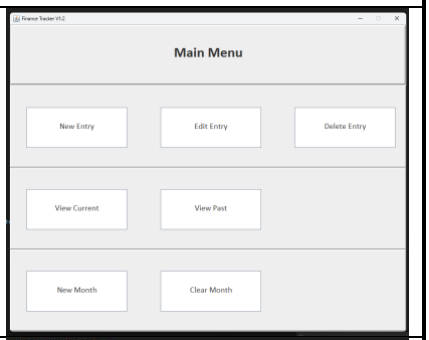
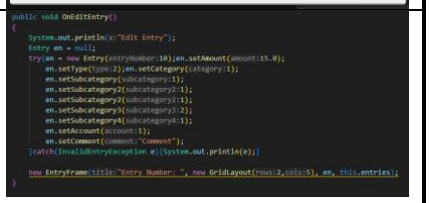
Test Case ID: 3.1	**Current Status: PASS**	**Date:** 03.28.25
Req. ID: 3.1 The program shall only display selection menus for dollarAmount, type, account, an area for a comment, and the submit or cancel buttons if instantiated with a null entry.		
Step #	**Operator Action**	**Expected Results**	**Comments**
1	Run the main method of FinanceTracker.	The Main Menu Should Display.	
2	Select New Entry.	A new instance of Entry Frame should be created with a null entry passed in the constructor.	

Screenshots:

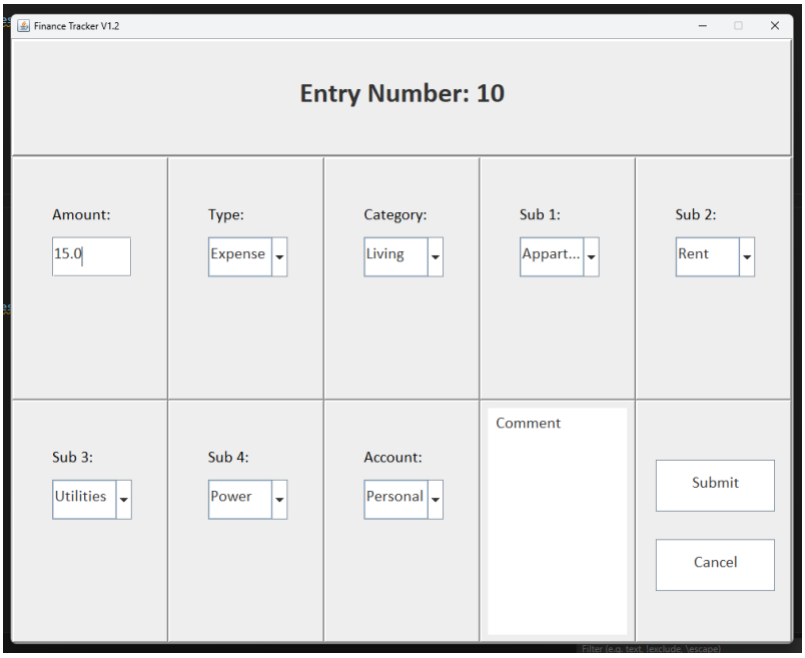


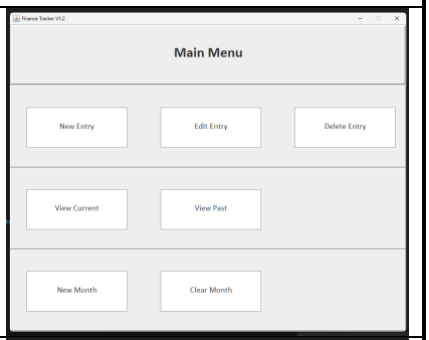
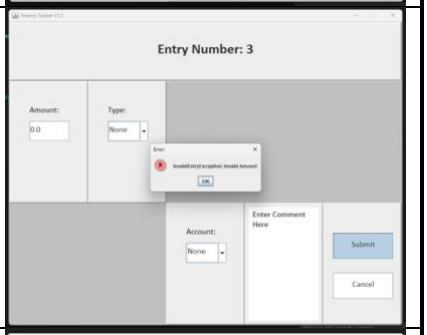
Test Case ID: 3.2	Current Status: Pass	Date: 03.28.25

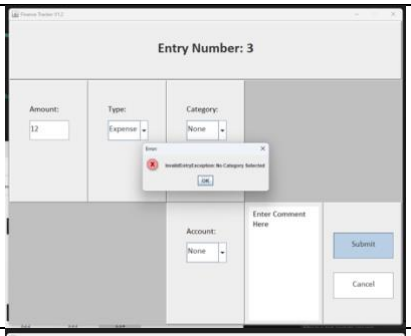
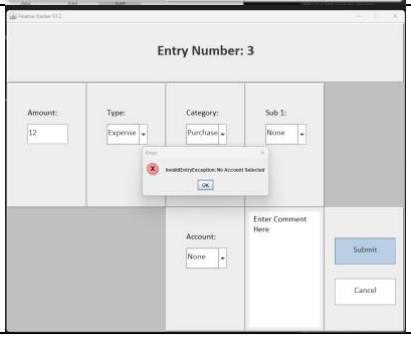
Req. ID: 3.2 The program shall load the Entry data if instantiated and display all the previously selected and entered values for the user to change.

Step #	Operator Action	Expected Results	Comments
1	Run the main method of FinanceTracker.	The Main Menu Should Display.	
2	Select Edit Entry.	A new instance of Entry Frame should be created with a fabricated entry passed in the constructor.	

Screenshots:



Test Case ID: 3.3	**Current Status: pending**	**Date:** 03.28.25
Req. ID: 3.3 The program shall add the Entry when the submit button is pressed only if the Entry contains valid information.		
Step #	**Operator Action**	**Expected Results**	**Comments**
1	Run the main method of FinanceTracker.	The Main Menu Should Display.	
2	Select New Entry.	A new instance of Entry Frame should be created with a null entry passed in the constructor.	
3	Click Submit	An Error message should appear, invalid amount.	
4	Enter Amount > 0 and click Submit.	An Error message should appear, invalid type.	

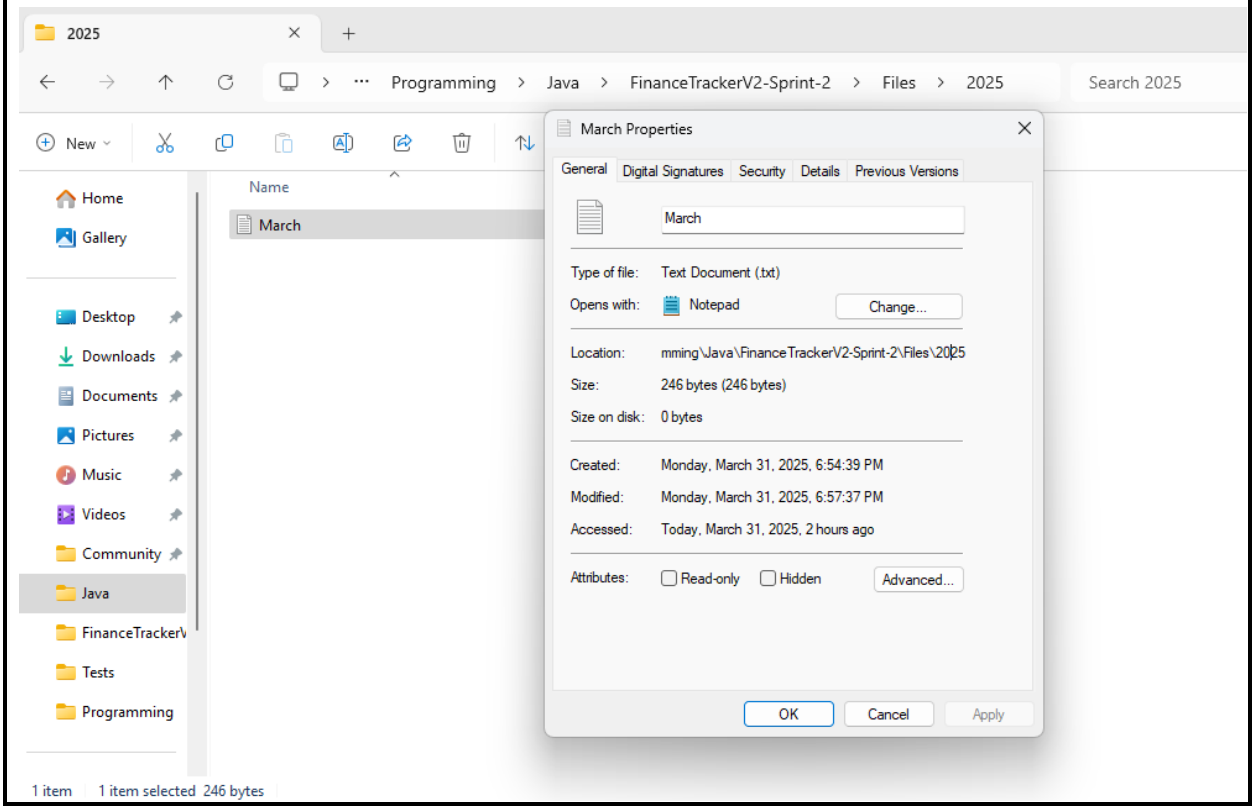
5	Select type expense and click submit.	An Error message should appear, invalid category.	
6	Select category personal and click submit.	An Error message should appear, no account selected.	
7	Select Account personal and click submit.	The frame should close and return to the main menu.	
Screenshots:			

Test Case ID: 4.1	Current Status: PASS	Date: 03.31.25

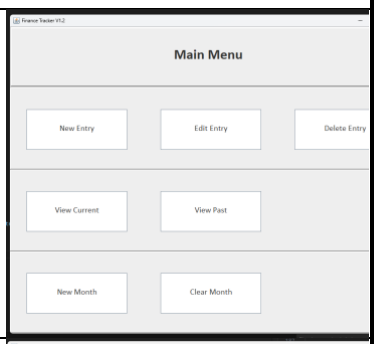
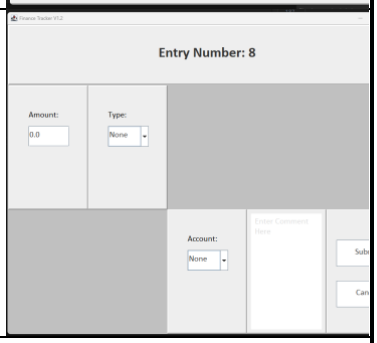
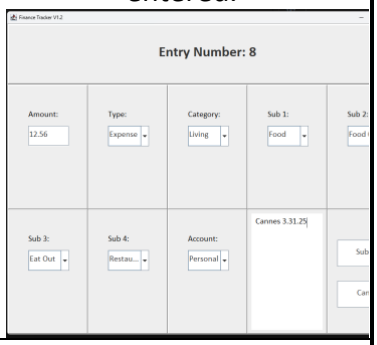
Req. ID: 4.1 The program shall name a new file Month.txt and it should be located in the directory ./Files/YYYY/Month.txt where YYYY is the current numerical year, and Month is the current month's name.

Step #	Operator Action	Expected Results	Comments
1	In the FileManager class's /*TEST MAIN*/ create a new instance of FileManager.	A new instance shall be created.	FileManager fm = new FileManager();
2	Using the makeDir and makeFile functions create a directory and file with the required name and location.	A new Directory shall be created named after the numerical year and inside of it contains a text file named after the current month.	LOCATION: ./Files/YYYY/Month.txt Example: ./Files/2025.March.txt

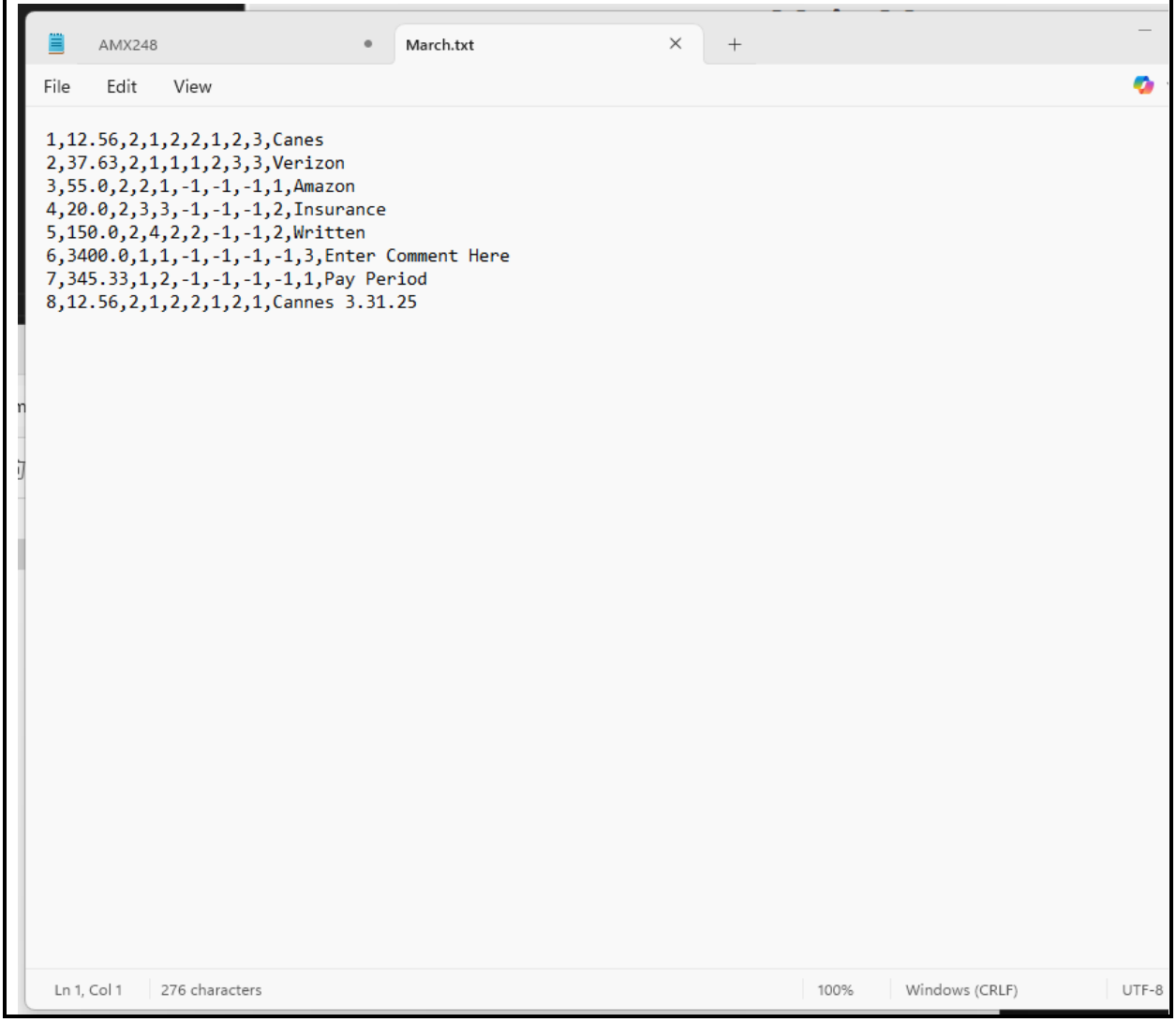
Screenshots:



Test Case ID: 4.2	Current Status: PASS	Date: 03.31.25
Req. ID: 4.2 The program shall be able to print a series of entries onto the file.		

Step #	Operator Action	Expected Results	Comments
1	Run the main funciton in FinanceTracker.java.	A new main menu frame should be visible.	
2	Select New Entry.	A new empty entry Frame should be visible.	
3	Fill out a new Entry with valid data and Click Submit.	A new Entry object should be created and added to the Entries ArrayList.	Valid data includes: amount > 0, type selected, category selected, a selection made for any subcategories that are visible, an account selected, a comment entered. 
4	Using File Explorer or equivalent, navigate to ./Files/2025/March.txt	Any Entry objects stored in the Entries object ArrayList should be printed in a semi-encoded string inside the file.	The numbers represent the selected index from the respective JComboBox.

Screenshots:

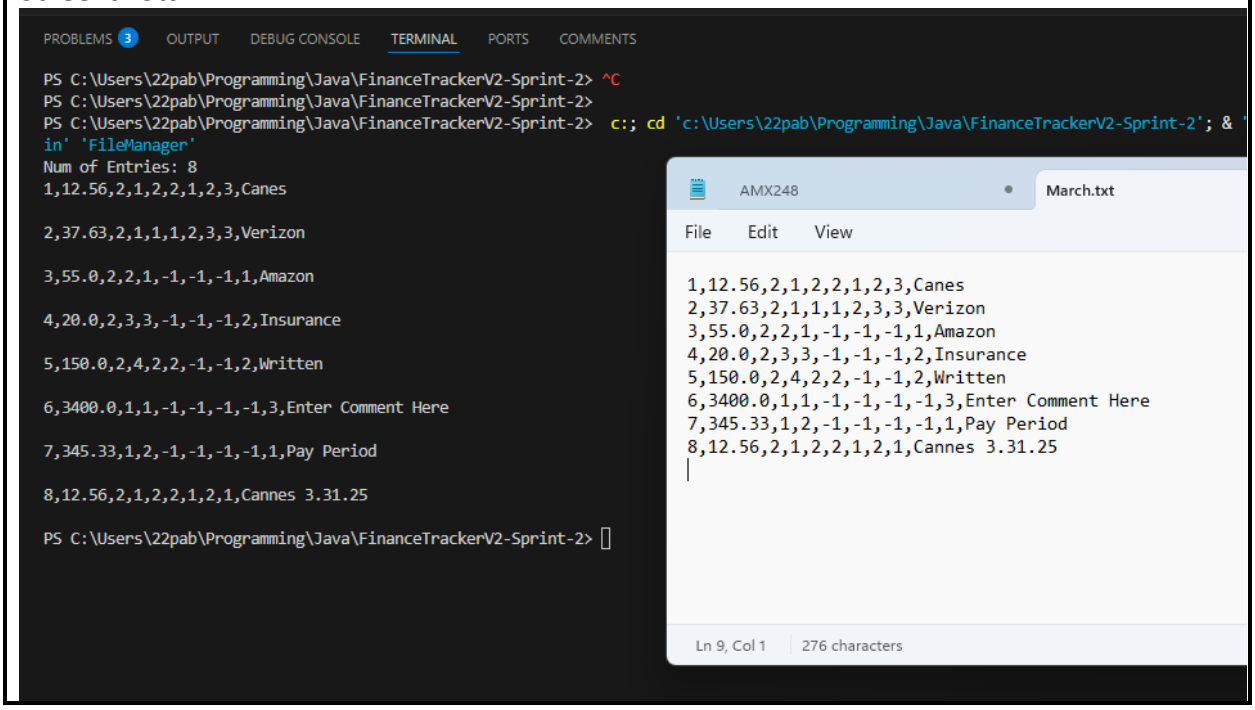


Test Case ID: 4.3	Current Status: PENDING	Date: 03.21.25

Req. ID: 4.3 The program shall be able to read the entries in a file, parse them into Entry objects and return an Entries object with all the file Entry objects in it.

Step #	Operator Action	Expected Results	Comments
1	In the FileManager /*TEST MAIN*/ function: Create a new instance of File Manager.	A new instance of FileManager shoud be created.	
2	Create an new instance of Entries and utilize the readEntries function in combination with the FILE PATH functions to read from the file created int the previous test case 4.2.	All the entries stored in the file should be read into the the Entries object.	
3	Create a new instance of ArraList<Entry> and initialize it with the value if the Entries object array list using the getEntries method. Iterate through the entries and print them to the console.	All the entries stored in the file should be printed onto the console.	

Screenshots:



Test Case ID: 5.1	Current Status: PASS	Date: 03.31.25

Req. ID: 5.1 The program can take a String read from a file and decode it into a textual representation.

Step #	Operator Action	Expected Results	Comments
1	Using the /*TEST MAIN*/ found in FileManager.java, create a new instance of Decoder.	New null instance of decoder is created.	
2	Inside the for loop: initialize the new Decoder with the string of the current entry being read.	The current entry being read from the file, will be passed into Decoder.	
3	Us the getDecoded Method to print out the decoded equivalent of the data.	The console should display the decoded user selected data from the entry.	

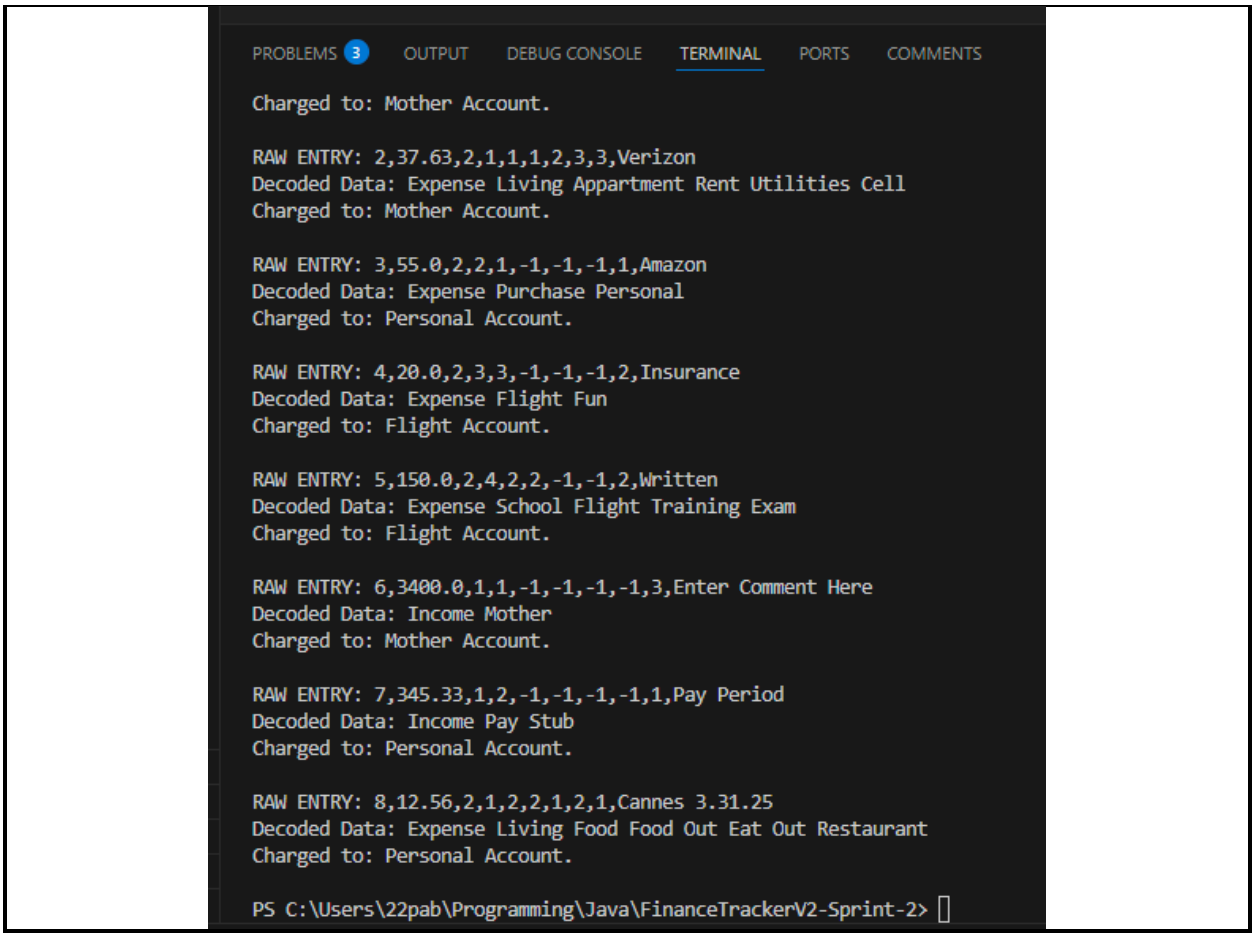
Screenshots:

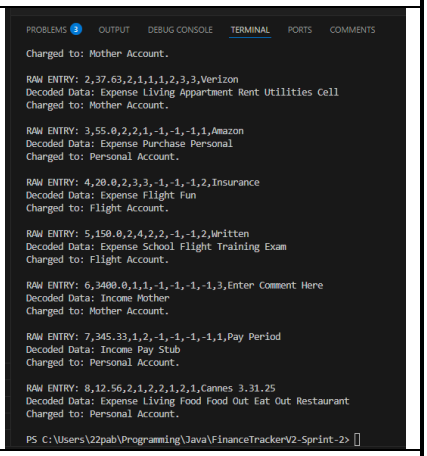
```
PROBLEMS 3   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   COMMENTS

PS C:\Users\22pab\Programming\Java\FinanceTrackerV2-Sprint-2> ^C
PS C:\Users\22pab\Programming\Java\FinanceTrackerV2-Sprint-2>
PS C:\Users\22pab\Programming\Java\FinanceTrackerV2-Sprint-2>  c:; cd 'c:\Users\22pab
in' 'FileManager'
Num of Entries: 8

RAW ENTRY: 1,12.56,2,1,2,2,1,2,3,Canes
Decoded Data: Expense Living Food Food Out Eat Out Restaurant

RAW ENTRY: 2,37.63,2,1,1,1,2,3,3,Verizon
Decoded Data: Expense Living Appartment Rent Utilities Cell

RAW ENTRY: 3,55.0,2,2,1,-1,-1,-1,1,Amazon
Decoded Data: Expense Purchase Personal

RAW ENTRY: 4,20.0,2,3,3,-1,-1,-1,2,Insurance
Decoded Data: Expense Flight Fun

RAW ENTRY: 5,150.0,2,4,2,2,-1,-1,2,Written
Decoded Data: Expense School Flight Training Exam

RAW ENTRY: 6,3400.0,1,1,-1,-1,-1,-1,3,Enter Comment Here
Decoded Data: Income Mother

RAW ENTRY: 7,345.33,1,2,-1,-1,-1,-1,1,Pay Period
Decoded Data: Income Pay Stub

RAW ENTRY: 8,12.56,2,1,2,2,1,2,1,Cannes 3.31.25
Decoded Data: Expense Living Food Food Out Eat Out Restaurant

PS C:\Users\22pab\Programming\Java\FinanceTrackerV2-Sprint-2>
```

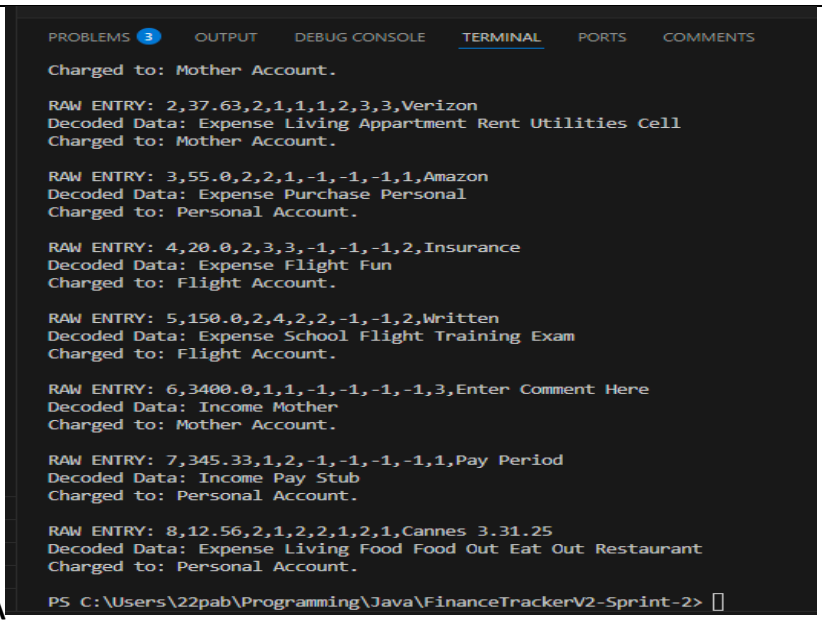
Test Case ID: 5.2	**Current Status: PASS**	**Date: 03.31.25**
Req. ID: 5.2 The program will determine what account was charged and represent it in a text version.		

Step #	**Operator Action**	**Expected Results**	**Comments**
1	Using the /*TEST MAIN*/ found in FileManager.java, create a new instance of Decoder.	New null instance of decoder is created.	
2	Inside the for loop: initialize the new Decoder with the string of the current entry being read.	The current entry being read from the file, will be passed into Decoder.	
3	Use the decodeAccount method in the while loop to print out what account the entry was charged to.	The console should print what account the entries were charged to.	

Screenshots:

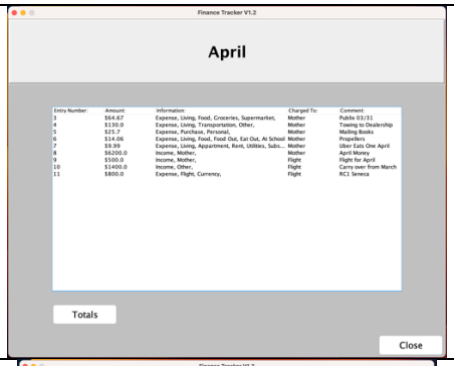
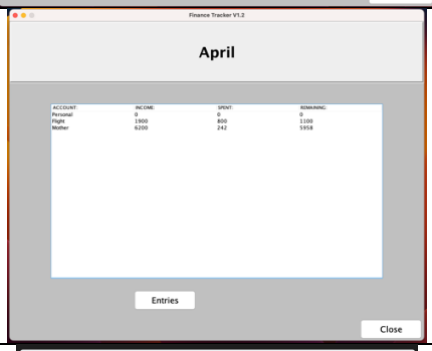
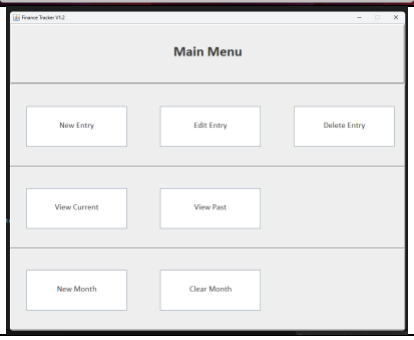

PROBLEMS 3
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
COMMENTS
Charged to: Mother Account.
RAW ENTRY: 2,37.63,2,1,1,1,2,3,3,Verizon
Decoded Data: Expense Living Appartment Rent Utilities Cell
Charged to: Mother Account.
RAW ENTRY: 3,55.0,2,2,1,-1,-1,-1,1,Amazon
Decoded Data: Expense Purchase Personal
Charged to: Personal Account.
RAW ENTRY: 4,20.0,2,3,3,-1,-1,-1,2,Insurance
Decoded Data: Expense Flight Fun
Charged to: Flight Account.
RAW ENTRY: 5,150.0,2,4,2,2,-1,-1,2,Written
Decoded Data: Expense School Flight Training Exam
Charged to: Flight Account.
RAW ENTRY: 6,3400.0,1,1,-1,-1,-1,-1,3,Enter Comment Here
Decoded Data: Income Mother
Charged to: Mother Account.
RAW ENTRY: 7,345.33,1,2,-1,-1,-1,-1,1,Pay Period
Decoded Data: Income Pay Stub
Charged to: Personal Account.
RAW ENTRY: 8,12.56,2,1,2,2,1,2,1,Cannes 3.31.25
Decoded Data: Expense Living Food Food Out Eat Out Restaurant
Charged to: Personal Account.
PS C:\Users\22pab\Programming\Java\FinanceTrackerV2-Sprint-2>

Test Case ID: 5.3	**Current Status: PASS**	**Date: 03.31.25**
Req. ID: 5.3 The program will only take the relevant information from an Entry.		
Step #	**Operator Action**	**Expected Results**	**Comments**
1	Run the /* Test Main*/ funciton in FileManager.java	A series of entries should be printed to the console.	
2	Verify printed values are those that are relevant.	When the relevant information is printed, the data from Type to SubCat4 should be converted into text form if not a –1. If a –1 is encountered nothing should be printed for that value.	The format of a Raw Entry is as follows: Entry Number, Ammount, Type, Category, SubCat1, SubCat2, SubCat3, SubCat4, Account, Comment If a piece of data is not relevant, it should be marked by a –1.
Screenshots:			


PROBLEMS 3
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
COMMENTS
Charged to: Mother Account.
RAW ENTRY: 2,37.63,2,1,1,1,2,3,3,Verizon
Decoded Data: Expense Living Appartment Rent Utilities Cell
Charged to: Mother Account.
RAW ENTRY: 3,55.0,2,2,1,-1,-1,-1,1,Amazon
Decoded Data: Expense Purchase Personal
Charged to: Personal Account.
RAW ENTRY: 4,20.0,2,3,3,-1,-1,-1,2,Insurance
Decoded Data: Expense Flight Fun
Charged to: Flight Account.
RAW ENTRY: 5,150.0,2,4,2,2,-1,-1,2,Written
Decoded Data: Expense School Flight Training Exam
Charged to: Flight Account.
RAW ENTRY: 6,3400.0,1,1,-1,-1,-1,-1,3,Enter Comment Here
Decoded Data: Income Mother
Charged to: Mother Account.
RAW ENTRY: 7,345.33,1,2,-1,-1,-1,-1,1,Pay Period
Decoded Data: Income Pay Stub
Charged to: Personal Account.
RAW ENTRY: 8,12.56,2,1,2,2,1,2,1,Cannes 3.31.25
Decoded Data: Expense Living Food Food Out Eat Out Restaurant
Charged to: Personal Account.
PS C:\Users\22pab\Programming\Java\FinanceTrackerV2-Sprint-2>

Test Case ID: 6.1	Current Status: PASS	Date: 04.07.2025

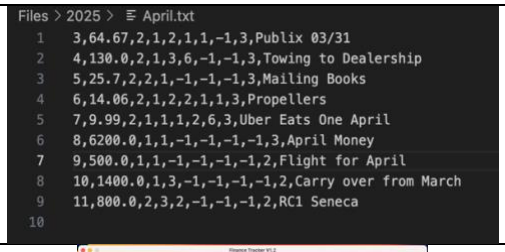
Req. ID: 6.1: The program shall display all, non-null, entries stored in a file with a neat table.

Step #	Operator Action	Expected Results	Comments
1	Run the application and click View Current	A new instance of Display Frame is created and the data from the corresponding file is loaded in.	
2	Click on the totals button.	The table data should change to show totals.	
3	Click the close button.	The frame is disposed and you are returned to the main menu.	

Screenshots:

Test Case ID: 6.2	Current Status: PASS	Date: 04.07.2025

Req. ID: 6.2: The program shall translate the raw entries using the Decoder class.

Step #	Operator Action	Expected Results	Comments
1	Verify the Current Month File is not empty.	At least one entry should be visible in the raw format.	
2	Run the application and click View Current.	A new instance of Display Frame is created and the data from the corresponding file is loaded in. The data in the table should be the text equivalent of the raw entry.	

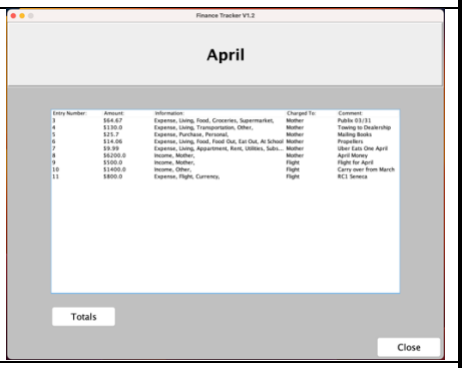
Screenshots:

Entry Number:	Amount:	Information:	Charged To:	Comment:
3	$64.67	Expense, Living, Food, Groceries, Supermarket,	Mother	Publix 03/31

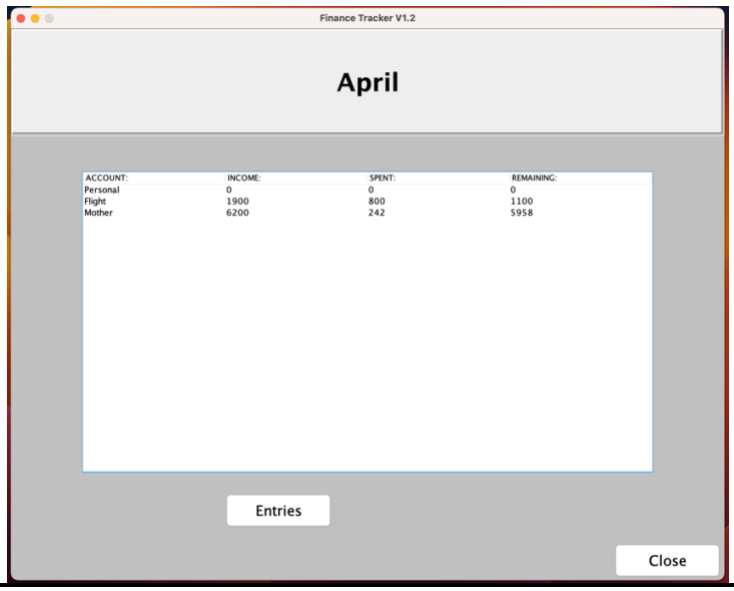
```
1    3,64.67,2,1,2,1,1,-1,3,Publix 03/31
```

Test Case ID: 6.3	Current Status: PASS	Date: 04.07.2025

Req. ID: 6.3: The program shall display the total income, total spent, and remaining for each available account.

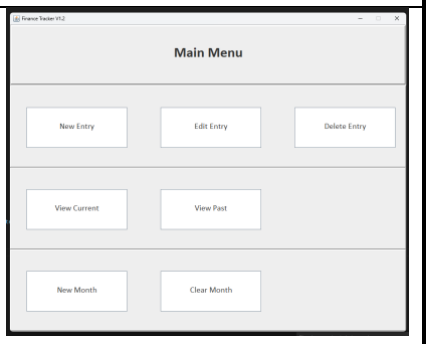
Step #	Operator Action	Expected Results	Comments
1	Run the application and click View Current	A new instance of Display Frame is created and the data from the corresponding file is loaded in.	
2	Click on the totals button.	The table data should change to show totals.	

Screenshots:

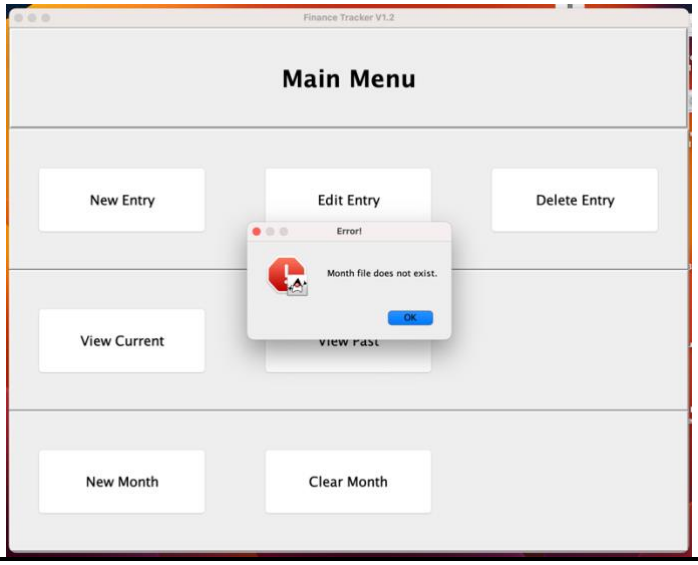


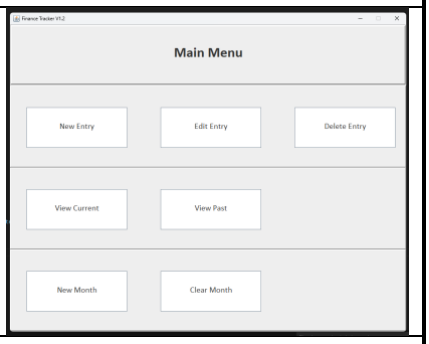
Test Case ID: 7.1	Current Status: PASS	Date: 04.01.25

Req. ID: 7.1 The program will display an error message when attempting to create a New Entry and the month file has not been created.

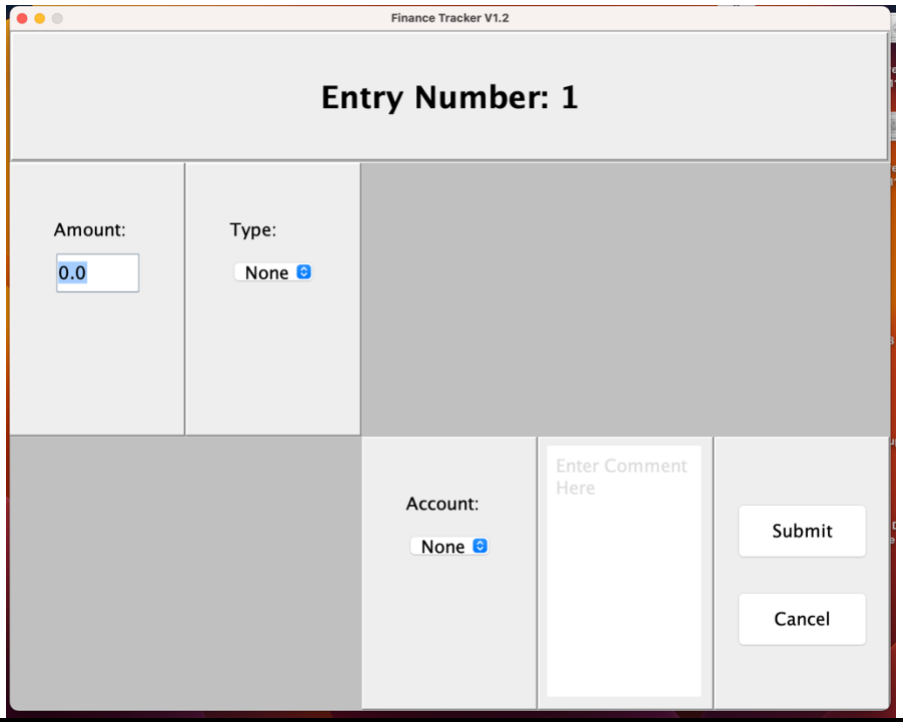
Step #	Operator Action	Expected Results	Comments
1	Run the Main function in FinanceTracker.java.	A new instance of Main Menu should appear on screen.	
2	Click the Clear Month button.	If a current month file exists, it will be removed from the computer.	
3	Click the New Entry button.	The application should create a pop-up window stating that there is no current month file.	

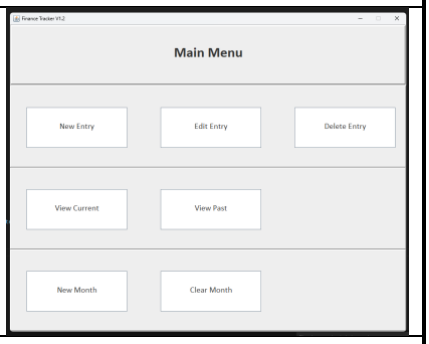
Screenshots:

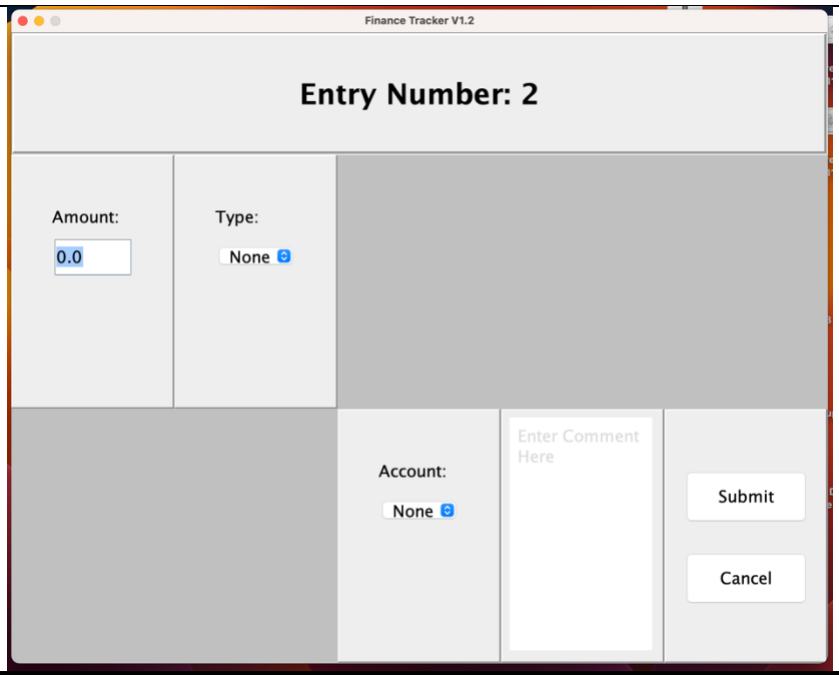


Test Case ID: 7.2	Current Status: PASS	Date: 04.01.25	
Req. ID: 7.2 The program will create a new instance of New Entry.			
Step #	**Operator Action**	**Expected Results**	**Comments**
1	Run the Main function in FinanceTracker.java.	A new instance of Main Menu should appear on screen.	
2	Click the New Month Button.	If no current month file exists, one will be created. If one exists it will ask to override: select ok.	
3	Click New Entry	A new instance of Entry Frame shall be loaded with a null Entry. The entry number should be 1.	

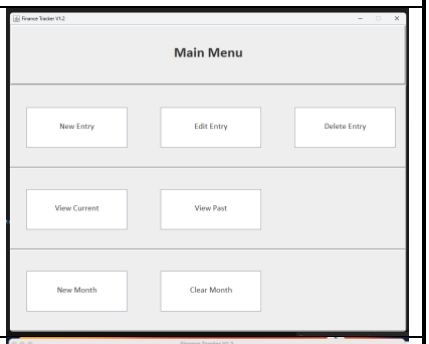
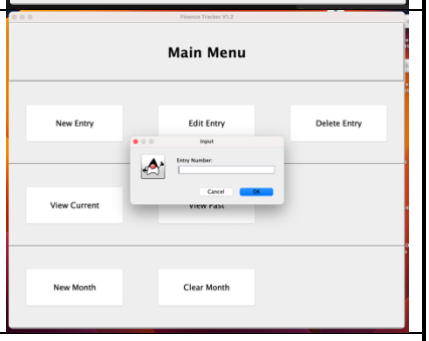
Screenshots:



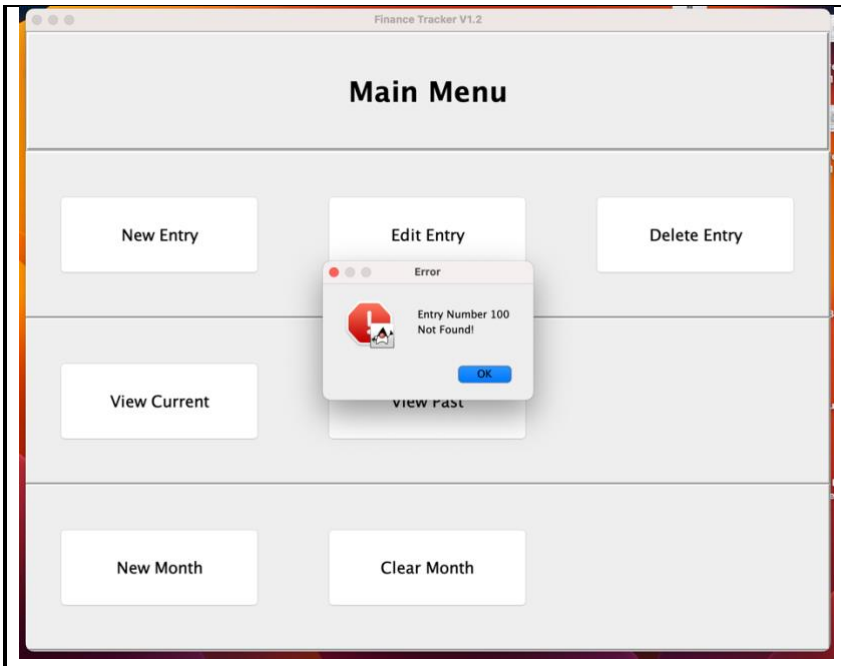
Test Case ID: 7.3	Current Status: PENDING	Date: 04.01.25	
Req. ID: 7.3			
Step #	**Operator Action**	**Expected Results**	**Comments**
1	Run the Main function in FinanceTracker.java.	A new instance of Main Menu should appear on screen.	
2	Click the New Month Button.	If no current month file exists, one will be created. If one exists it will ask to override: select ok.	
3	Click New Entry	A new instance of Entry Frame shall be loaded with a null Entry. The entry number should be 1.	
4	Enter a valid entry.	The entry should be added to the Entries collection class and printed to the current month file.	If the entry number reads something other than 1, it is likely there are already entries in the file. Skip this step if that is true.
5	Click New Entry	A new instance of Entry Frame shall be created with a null Entry. The entry number should be sequential to the previous entry.	
Screenshots:			

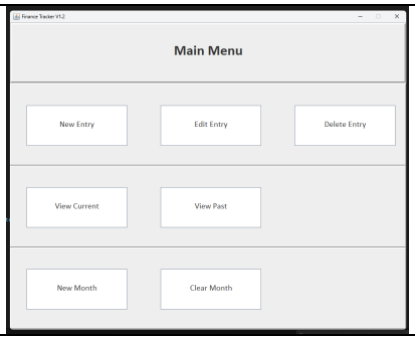
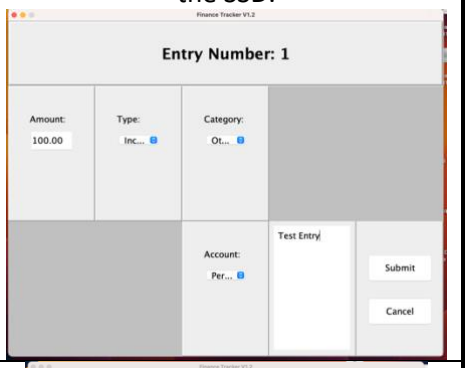
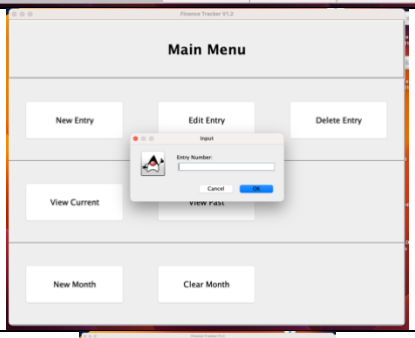
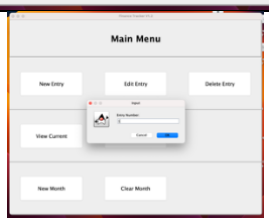

Finance Tracker V1.2
Entry Number: 2
Amount:
0.0
Type:
None
Account:
None
Enter Comment Here
Submit
Cancel

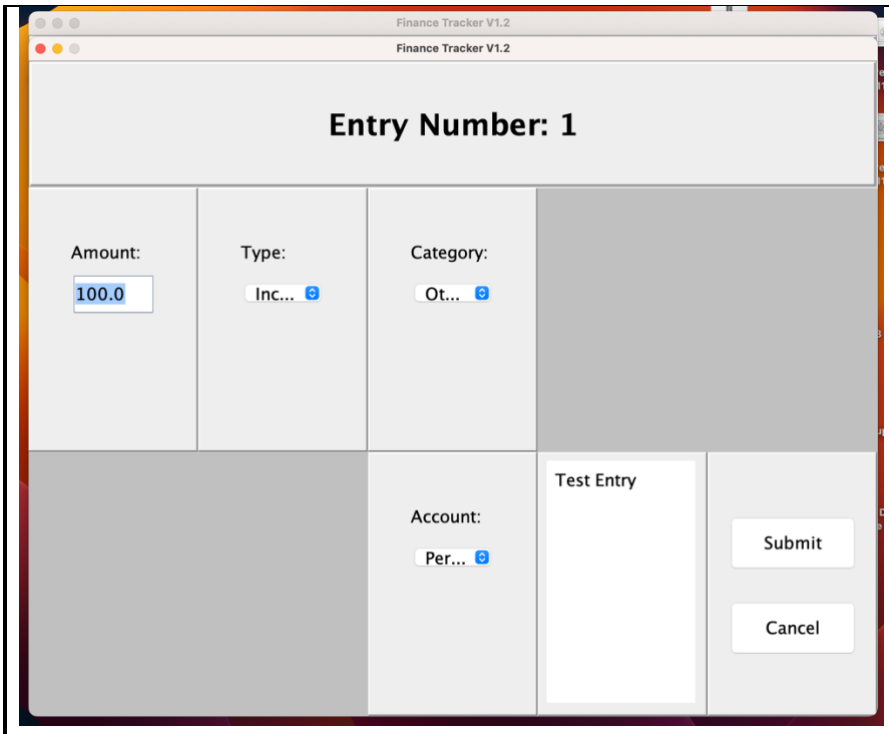
Test Case ID: 8.1	Current Status: PASS	Date: 04.01.25
Req. ID: 8.1 The program will ask for an entry number when Edit Entry is clicked. If the entry number does not exist a pop-up message will notify the user.		

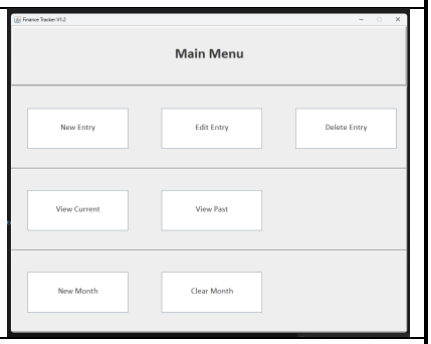
Step #	Operator Action	Expected Results	Comments
1	Run the Main function in FinanceTracker.java.	A new instance of Main Menu should open.	
2	Select Edit Entry.	A pop-up should appear asking for an entry number.	
3	Enter 100	A pop-up should appear stating entry number 100 was not found.	This assumes there is less than 100 entries in the current file.

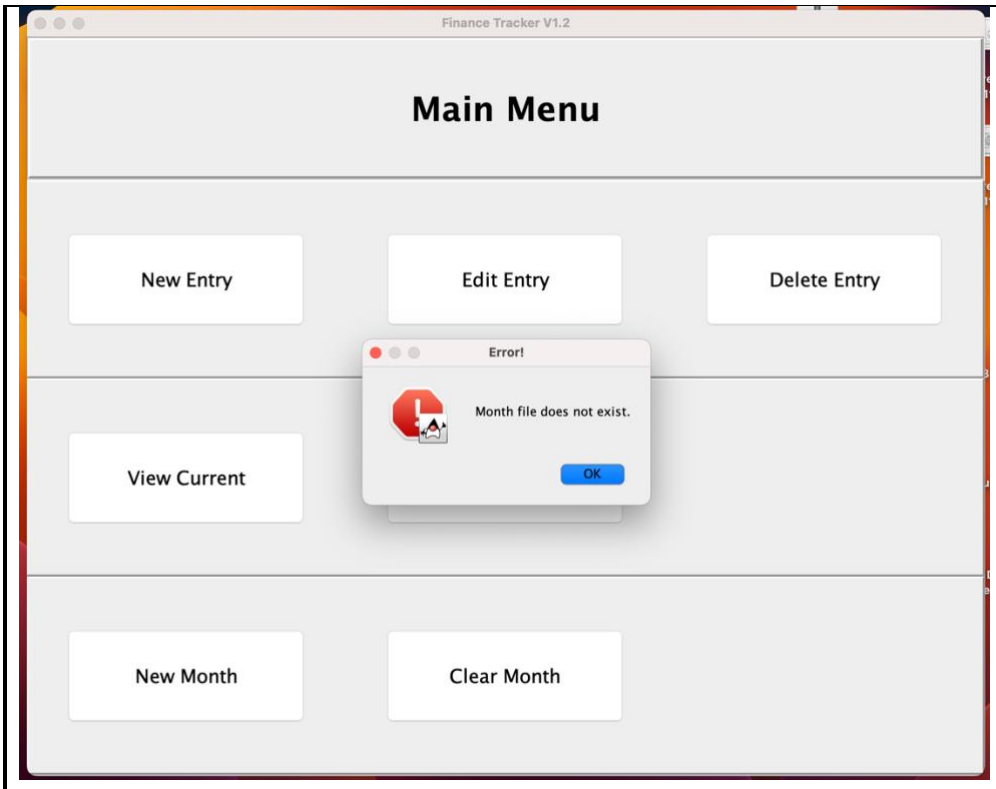
Screenshots:

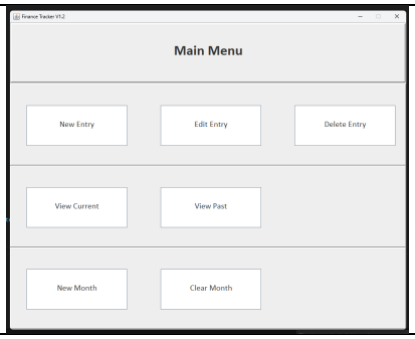
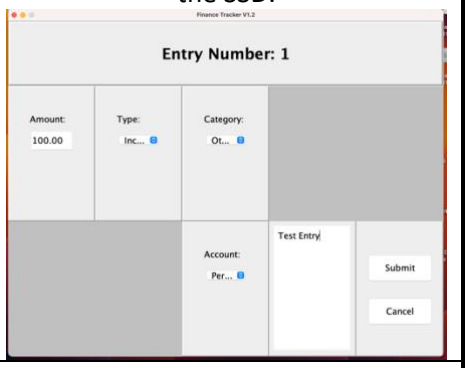

Finance Tracker V1.2
Main Menu
New Entry
Edit Entry
Delete Entry
Error
Entry Number 100
Not Found!
OK
View Current
View Past
New Month
Clear Month

Test Case ID: 8.2	Current Status: PASS	Date: 04.01.25	
Req. ID: 8.2 The program will ask for an entry number when Edit Entry is clicked. If the entry number exists a new instance of Entry Frame will be created and load the entry.			
Step #	**Operator Action**	**Expected Results**	**Comments**
1	Run the Main function in FinanceTracker.java.	A new instance of Main Menu should open.	
2	Select New Entry and create a new valid entry. Submit.	A new entry should be added to the file and entries collection class.	This assumes valid data is the same as other cases and as described in the SSD. 
3	Select Edit Entry.	A pop-up should appear asking for an entry number.	
4	Enter the number of the entry created in step 2.	A new instance of Entry Frame should be created and the selected Entry should be loaded into the selections.	
Screenshots:			

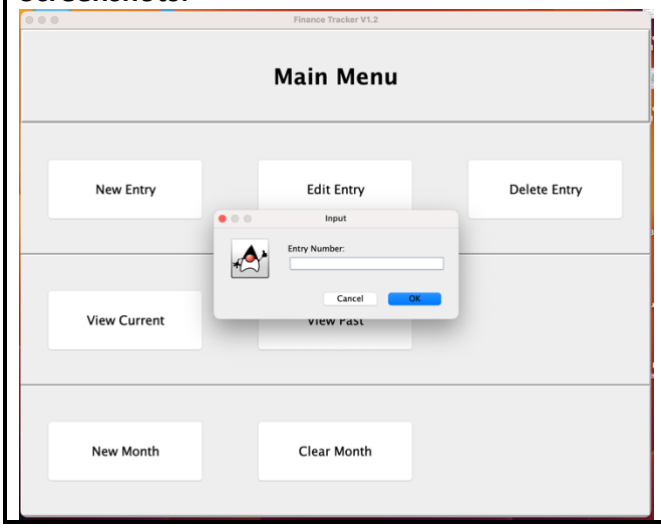


Test Case ID: 8.3	Current Status: PASS		Date: 04.01.25
Req. ID: 8.3			
Step #	**Operator Action**	**Expected Results**	**Comments**
1	Run the Main function in FinanceTracker.java.	A new instance of Main Menu should open.	
2	Select Clear Month and confirm.	The current file should be removed from the users computer.	
3	Select Edit Entry	A pop-up should appear stating there is no current month file.	
Screenshots:			


Finance Tracker V1.2
Main Menu
New Entry
Edit Entry
Delete Entry
Error!
Month file does not exist.
OK
View Current
New Month
Clear Month

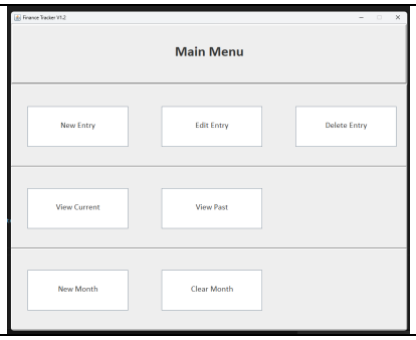
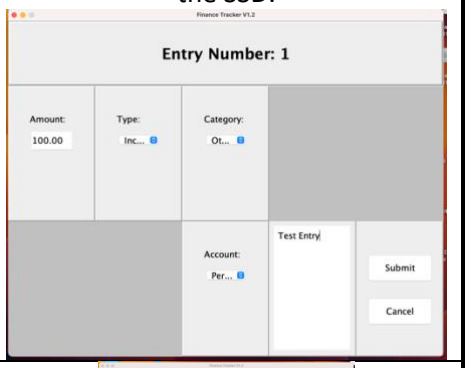
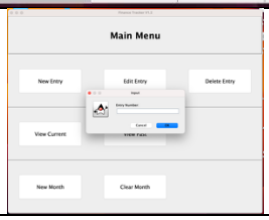
Test Case ID: 9.1	Current Status: PASS	Date: 04.01.25	
Req. ID: 9.1 The program will ask for an entry number to delete.			
Step #	**Operator Action**	**Expected Results**	**Comments**
1	Run the Main function in FinanceTracker.java.	A new instance of Main Menu should open.	
2	Select New Entry and create a new valid entry. Submit.	A new entry should be added to the file and entries collection class.	This assumes valid data is the same as other cases and as described in the SSD. 
3	Select Delete Entry.	A pop-up should appear asking for an entry number.	

Screenshots:

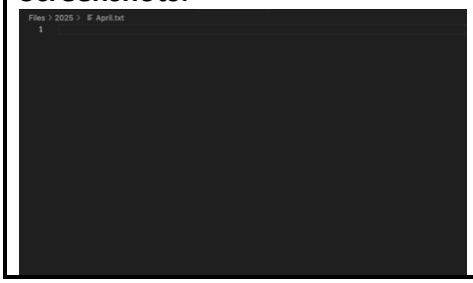


Test Case ID: 9.2	Current Status: PASS	Date: 04.01.25

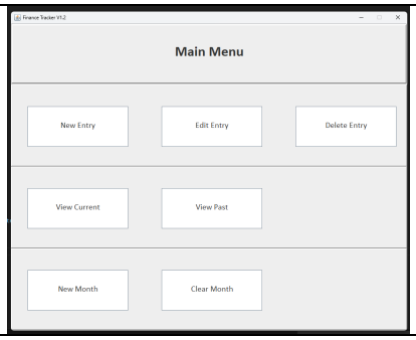
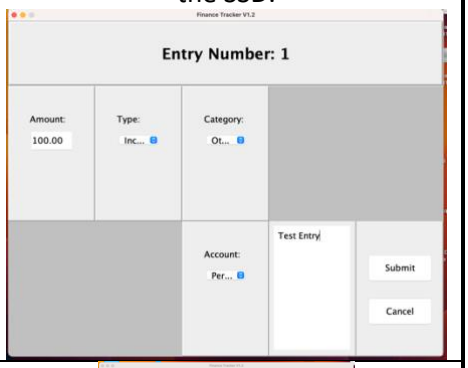
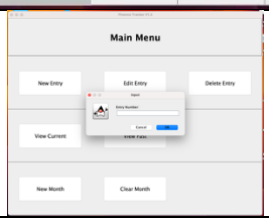
Req. ID: 9.2 The program will search for an Entry by number and if found remove it from the Entries collection class and update the file.

Step #	Operator Action	Expected Results	Comments
1	Run the Main function in FinanceTracker.java.	A new instance of Main Menu should open.	
2	Select New Entry and create a new valid entry. Submit.	A new entry should be added to the file and entries collection class.	This assumes valid data is the same as other cases and as described in the SSD. 
3	Select Delete Entry.	A pop-up should appear asking for an entry number.	
4	Enter 1.	The Entry should be removed from the text file and collection class.	

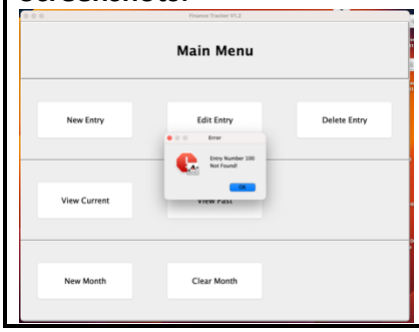
Screenshots:



Test Case ID: 9.3	Current Status: PASS	Date: 04.01.25
Req. ID: 9.3 The program will search for an Entry by number and if not found notify the user using a pop-up.		

Step #	Operator Action	Expected Results	Comments
1	Run the Main function in FinanceTracker.java.	A new instance of Main Menu should open.	
2	Select New Entry and create a new valid entry. Submit.	A new entry should be added to the file and entries collection class.	This assumes valid data is the same as other cases and as described in the SSD. 
3	Select Delete Entry.	A pop-up should appear asking for an entry number.	
4	Enter 100.	A pop-up should appear stating entry number 100 was not found.	This assumes the current file has less than 100 entries.

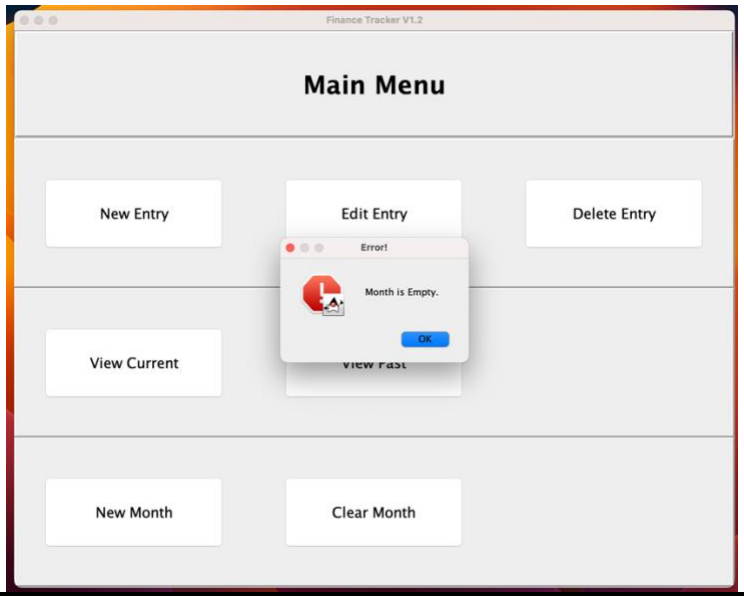
Screenshots:



Test Case ID: 10.1	Current Status: PASS	Date: 04.07.2025
Req. ID: 10.1: The program will notify the user using a pop-up when no current file exists.		

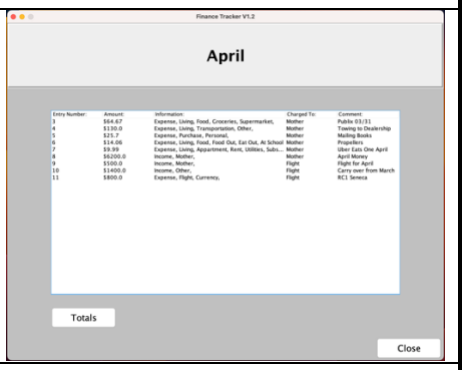
Step #	Operator Action	Expected Results	Comments
1	Run the application and click the Clear Month and confirm.	The current month file should be deleted.	
2	Click the View Current button.	The application should notify you the month does not exist.	

Screenshots:



Test Case ID: 10.2	Current Status: PASS	Date: 04.07.2025

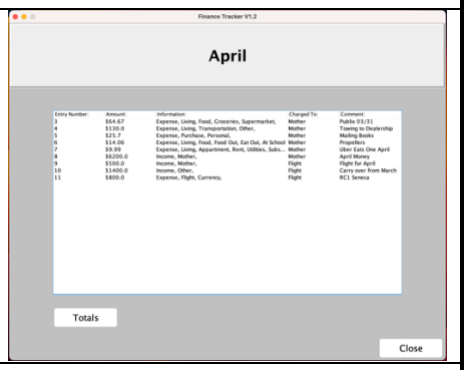
Req. ID: 10.2: The program will load and display each Entry stored in the current file.

Step #	Operator Action	Expected Results	Comments
1	Run the application and click View Current	A new instance of Display Frame is created and the data from the corresponding file is loaded in.	

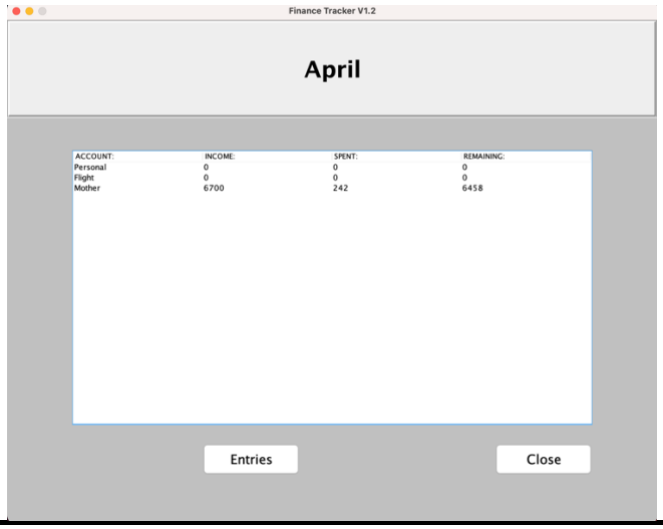
Screenshots:

Test Case ID: 10.3	Current Status: PASS	Date: 04.07.2025

Req. ID: 10.3: The program will calculate the total income, total spent, and remaining for each account, for the month.

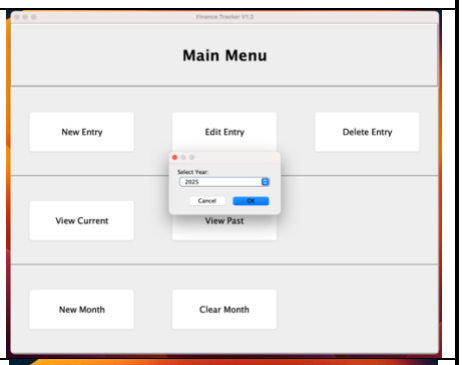
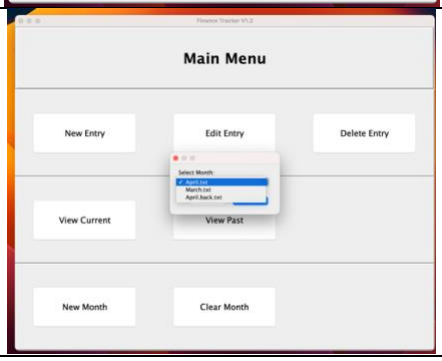
Step #	Operator Action	Expected Results	Comments
1	Run the application and click View Current	A new instance of Display Frame is created and the data from the corresponding file is loaded in.	
2	Click the Totals Button.	The Application should display a new table with the total income, expense, and remaining for each account.	

Screenshots:



Test Case ID: 11.1	Current Status: PASS	Date: 04.07.2025

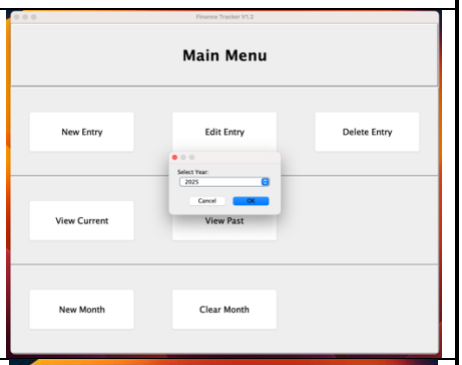
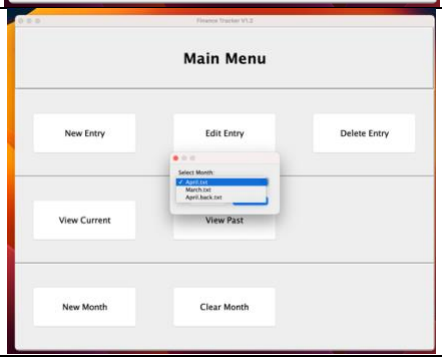
11.1: The program will prompt the user to select the desired year and month using a series pf pop-up dropdown.

Step #	Operator Action	Expected Results	Comments
1	Run the program and select View Past.	The program will prompt you to select a year.	
2	Select a year and click ok.	The program will prompt you to select a month.	
3	Select a month and click ok.	The program will create a new instance of Display Frame and load the corresponding data.	

Screenshots:

Test Case ID: 11.2	Current Status: PASS	Date: 04.07.2025

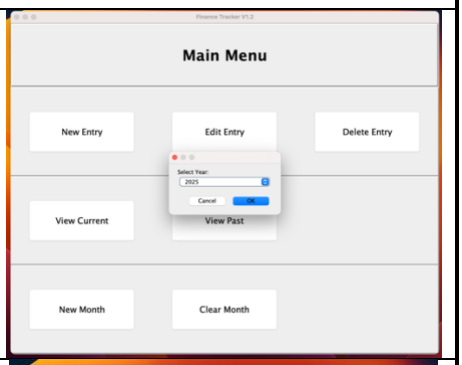
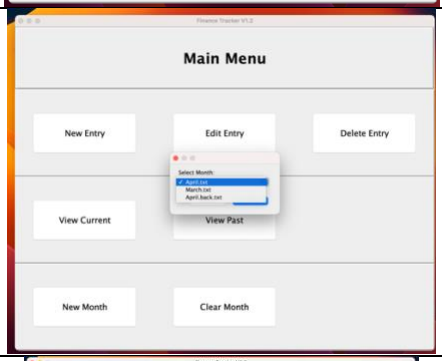
Req. ID: 11.2: The program will load the corresponding file if it exists and display it in a DisplayFrame.

Step #	Operator Action	Expected Results	Comments
1	Run the program and select View Past.	The program will prompt you to select a year.	
2	Select a 2025 and click ok.	The program will prompt you to select a month.	
3	Select March and click ok.	The program will create a new instance of Display Frame and load the corresponding data.	

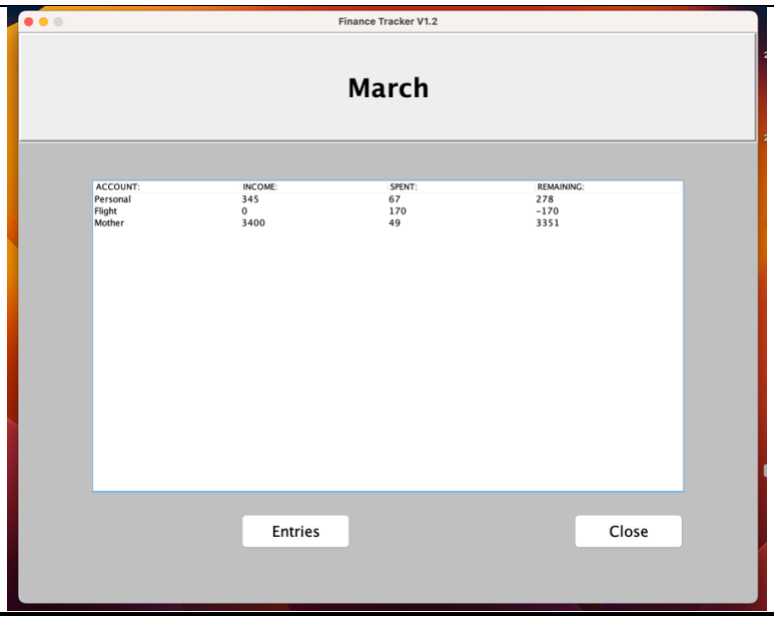
Screenshots:

Test Case ID: 11.3	Current Status: PASS	Date: 04.07.2025

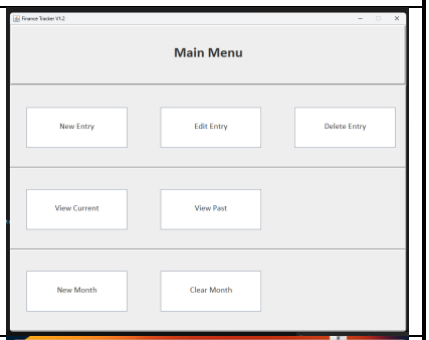
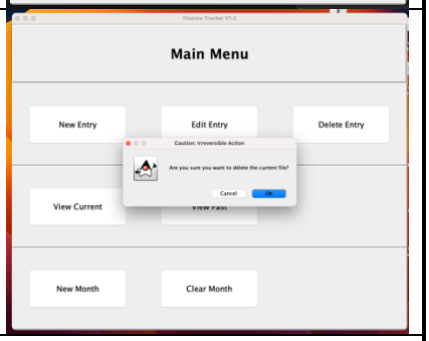
Req. ID: 11.3 The program will total the income, expense, and remaining using the data in the file.

Step #	Operator Action	Expected Results	Comments
1	Run the program and select View Past.	The program will prompt you to select a year.	
2	Select a year and click ok.	The program will prompt you to select a month.	
3	Select a month and click ok.	The program will create a new instance of Display Frame and load the corresponding data.	
4	Click totals.	The program will display a new table with the correct totals for all accounts	

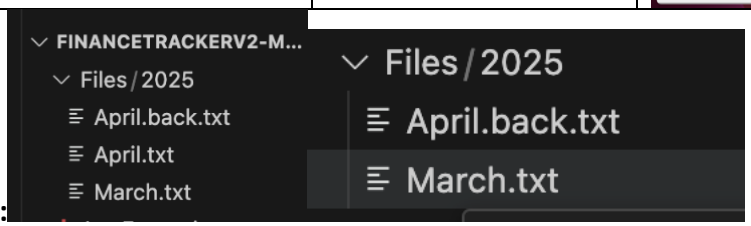
Screenshots:


Finance Tracker V1.2
March
ACCOUNT:
INCOME:
SPENT:
REMAINING:
Personal
345
67
278
Flight
0
170
-170
Mother
3400
49
3351
Entries
Close

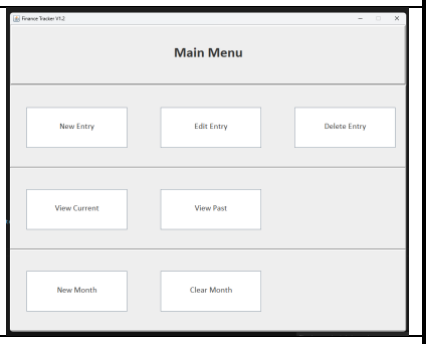
Test Case ID: 12.1	Current Status: PASS	Date: 04.01.25
Req. ID: 12.1 The program will remove the current month file if it exists.		

Step #	Operator Action	Expected Results	Comments
1	Run the Main function in FinanceTracker.java.	A new instance of Main Menu should open.	
2	Click Clear Month and select Ok.	The current month file will be removed from the device.	

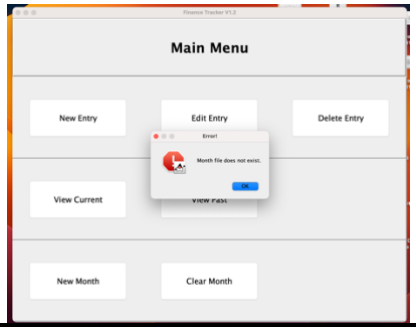
Screenshots:



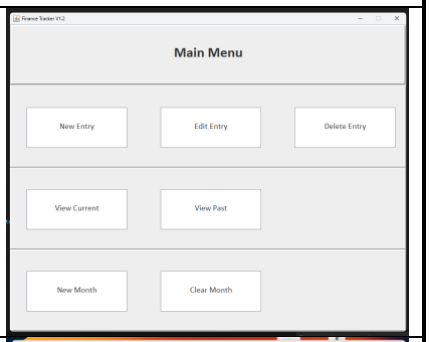
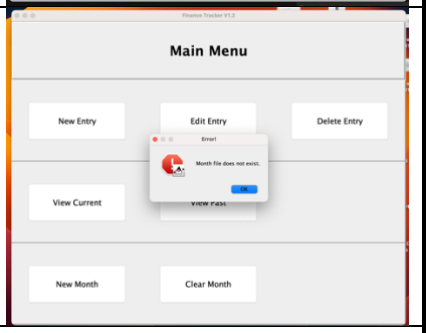
Test Case ID: 12.2	**Current Status: PASS**	**Date:** 04.01.25
Req. ID: 12.2 The program will notify the user using a pop-up if the file does not exist.		

Step #	**Operator Action**	**Expected Results**	**Comments**
1	Run the Main function in FinanceTracker.java.	A new instance of Main Menu should open.	
2	Click Clear Month	A pop-up will appear notifying you the file does not exist.	

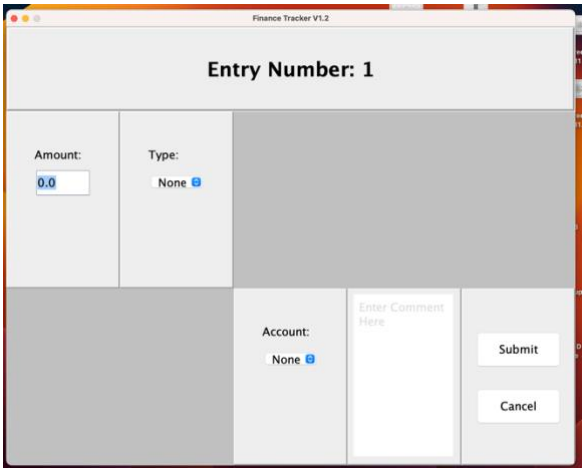
Screenshots:

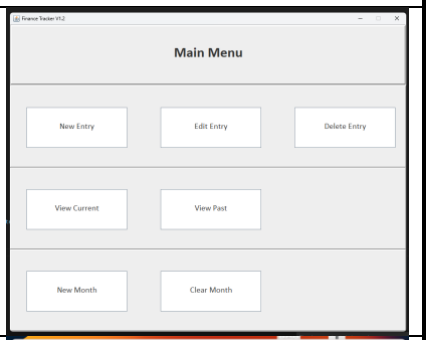
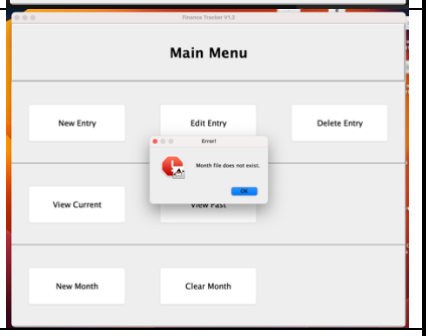


Test Case ID: 12.3	**Current Status: PASS**	**Date:** 04.01.25
Req. ID: 12.3 The program will clear the local Entries array so it does not store the local data to the file.		

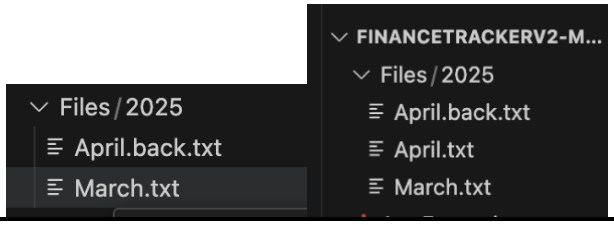
Step #	**Operator Action**	**Expected Results**	**Comments**
1	Run the Main function in FinanceTracker.java.	A new instance of Main Menu should open.	
2	Click clear month.	If the file exists it will be removed. If it doesn't exist a pop up will appear, click ok.	
3	Click New Month	A new month file will be created.	
4	Click New Entry	A new instance of Entry Frame should be created with a null entry and entry number 1.	If the entry number is 1 it means the array list has been properly cleared.

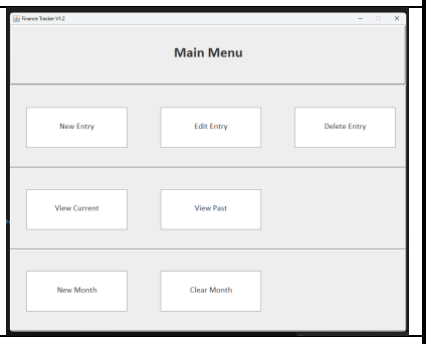
Screenshots:



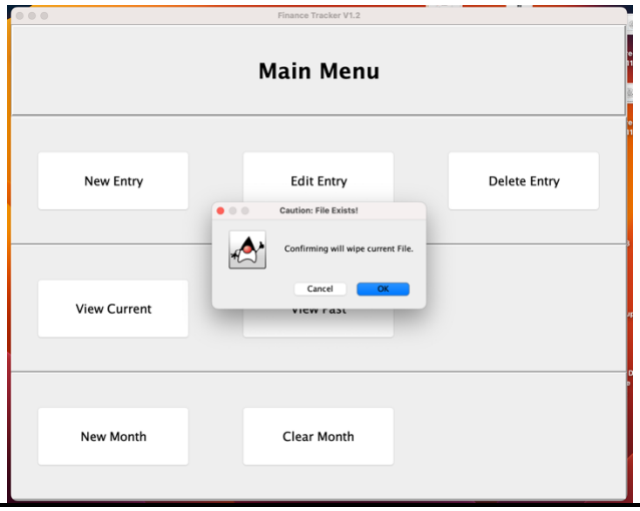
Test Case ID: 13.1	**Current Status: PASS**	**Date:** 04.01.25
Req. ID: 13.1 The program will create a new file for the current month if one does not exist.		
Step #	**Operator Action**	**Expected Results**	**Comments**
1	Run the Main function in FinanceTracker.java.	A new instance of Main Menu should open.	
2	Click Clear Month	If the file exists it will be removed. If it doesn't exist a pop up will appear, click ok.	
3	Click New Month	A new month file will be created.	

Screenshots:



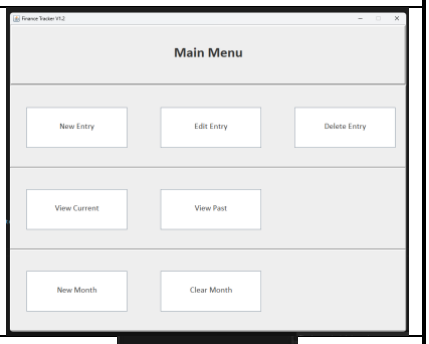
Test Case ID: 13.2	**Current Status: PASS**	**Date:** 04.01.25
Req. ID: 13.2 The program will prompt the user to override the current file if one exists.		
Step #	**Operator Action**	**Expected Results**	**Comments**
1	Run the Main function in FinanceTracker.java.	A new instance of Main Menu should open.	
2	Click New Month	A pop-up will appear asking if you would like to override the current file.	
3	Select Ok	The month file will be overriden.	

Screenshots:



Test Case ID: 13.3	Current Status: PASS	Date: 04.01.25

Req. ID: 13.3 The program will create the directory if it does not exist.

Step #	Operator Action	Expected Results	Comments
1	Run the Main function in FinanceTracker.java.	A new instance of Main Menu should open.	
2	Delete the Files Directory in the application folder.	No directory will exist.	
3	Click New Month.	The Files/20XX/ directory will be created depending on the current year.	

Screenshots:

REFERENCES:

Oracle. (n.d.). *Trail: Creating a GUI with swing*. Trail: Creating a GUI With Swing (The JavaTM Tutorials). https://docs.oracle.com/javase/tutorial/uiswing/index.html

APPENDICES: