Peyton Wolf

4/28/2025

Cst-250 Programming in C# ||

Mark Smithers

Grand Canyon University

Activity 6

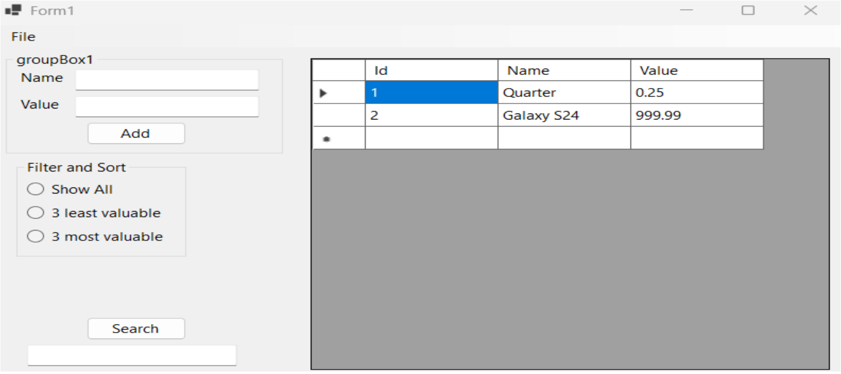
Loom Videos

1. ***https://www.loom.com/share/53640dfafb314365bc09faa127df347c?sid=9c596c0a-39d0-4b02-afb2-ed32f4371204***
2. ***https://www.loom.com/share/05d753766c8d47429678eeac4c4e4be8?sid=05dbf079-136d-4a77-933a-d2d296fa8f0e***

GitHub Link

https://github.com/KnoxHighStax/CST250/tree/main/Activity%206

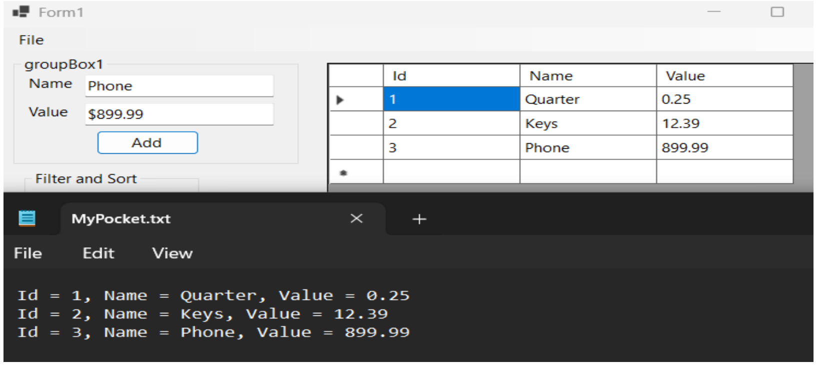
Adding Some Items to my Pocket



Id	Name	Value
1	Quarter	0.25
2	Galaxy S24	999.99

In this screenshot we are demonstrating how we have created a form that imitates the "thingsInMyPocket" for any individual to be able to track. The user will be able to input the Name and Value of the items that are in their pocket and then click on the "Add" button. From here the system will take the values inputting by the user in "txtName" and "txtValue" and display them in the DataGrid "gridInventory" and then give each of the items a unique identifying number.

Saving Items to a List



Name: Phone

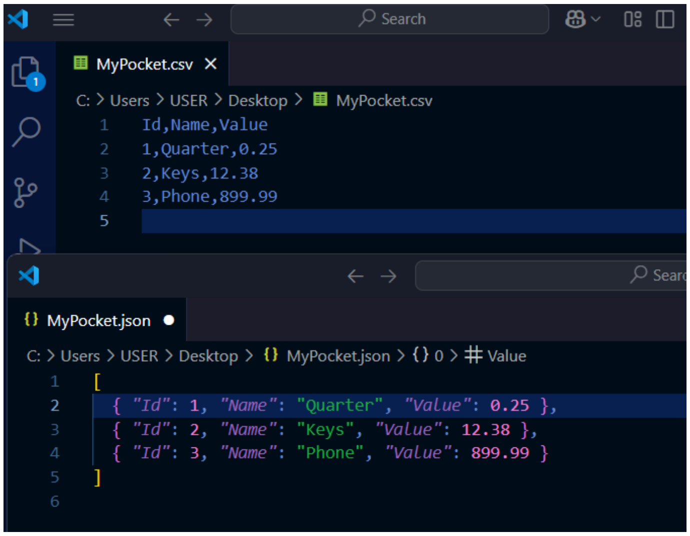
Value: $899.99

Id	Name	Value
1	Quarter	0.25
2	Keys	12.39
3	Phone	899.99

```
Id = 1, Name = Quarter, Value = 0.25
Id = 2, Name = Keys, Value = 12.39
Id = 3, Name = Phone, Value = 899.99
```

In this screenshot we are demonstrating how we have added the ability for the user to save files, specifically the user will have the ability to save ".txt" files. When the user loads up the program like normal, but this time they will see the newly populated menu strip labeled with “File”. The user will be able to click on “File” and from there they can click on “Save” and then continue along the normal saving process, they do just need to make sure that they select the file type to save it as to be “.txt” or it will not work. To test if their file has been saved, they search for it on the computer or if the user chooses to save it to desktop can just find it there and should have all the information the user inputted. The user also can load up any “.txt” files and display them in the DataGrid “gridInventory” to review different “.txt” lists in the program.

CSV and JSON Saving and Formatting

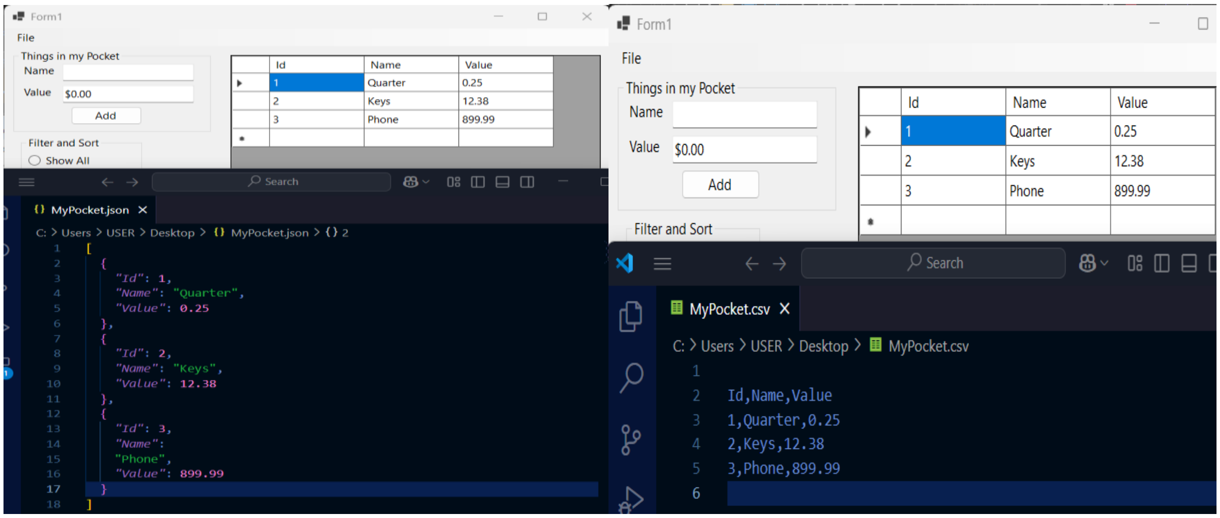

Search
MyPocket.csv
C: > Users > USER > Desktop > MyPocket.csv
Search
MyPocket.json
C: > Users > USER > Desktop > MyPocket.json > {} 0 > # Value

```
Id,Name,Value
1,Quarter,0.25
2,Keys,12.38
3,Phone,899.99
```

```
[
  { "Id": 1, "Name": "Quarter", "Value": 0.25 },
  { "Id": 2, "Name": "Keys", "Value": 12.38 },
  { "Id": 3, "Name": "Phone", "Value": 899.99 }
]
```

In this screenshot we are demonstrating how we have added the ability for the user to save files, specifically the user now also can save “.json” and “.csv” files. When the user loads up the program like normal, but this time they will see the newly populated menu strip labeled with “File”. The user will be able to click on “File” and from there they can click on “Save” and then continue along the normal saving process, they do just need to make sure that they select the file type to save it as to be “.json” and “.csv” or it will not work. To test if their file has been saved, they search for it on the computer or if the user chooses to save it to desktop can just find it there and should have all the information the user inputted.

CSV and JSON Loading Saved Docs



Things in my Pocket

Name

Value $0.00

Filter and Sort

○ Show All

	Id	Name	Value
▶	1	Quarter	0.25
	2	Keys	12.38
	3	Phone	899.99
*			

MyPocket.json

C: > Users > USER > Desktop > MyPocket.json > {} 2

```
[
  {
    "Id": 1,
    "Name": "Quarter",
    "Value": 0.25
  },
  {
    "Id": 2,
    "Name": "Keys",
    "Value": 12.38
  },
  {
    "Id": 3,
    "Name":
    "Phone",
    "Value": 899.99
  }
]
```

Things in my Pocket

Name

Value $0.00

Filter and Sort

	Id	Name	Value
▶	1	Quarter	0.25
	2	Keys	12.38
	3	Phone	899.99
*			

MyPocket.csv

C: > Users > USER > Desktop > MyPocket.csv

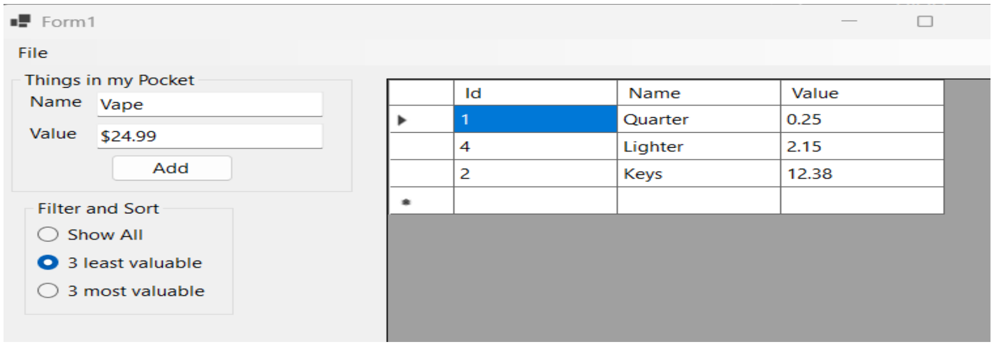
```

Id,Name,Value
1,Quarter,0.25
2,Keys,12.38
3,Phone,899.99

```

In this screenshot we are demonstrating how we have added the ability for the user to save files, specifically the user now also can save ".json" and ".csv" files. When the user loads up the program like normal, but this time they will see the newly populated menu strip labeled with "File". The user will be able to click on "File" and from there they can click on "Save" and then continue along the normal saving process, they do just need to make sure that they select the file type to save it as to be ".json" and ".csv" or it will not work. To test if their file has been saved, they search for it on the computer or if the user chooses to save it to desktop can just find it there and should have all the information the user inputted.

Sorting by Least Valuable



Things in my Pocket

Name Vape

Value $24.99

Filter and Sort

○ Show All

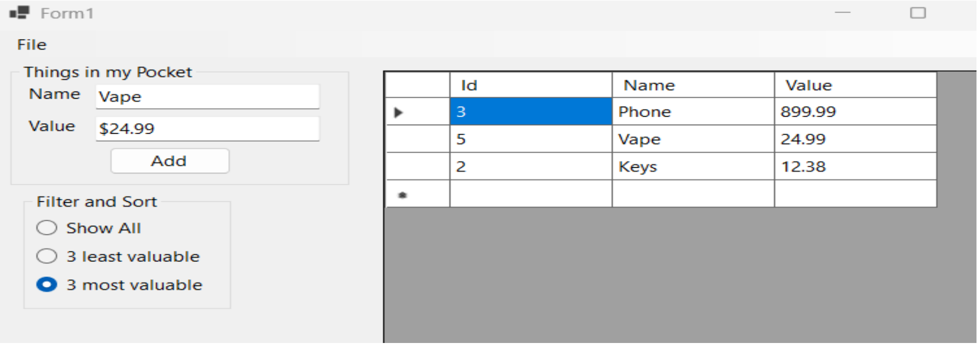
● 3 least valuable

○ 3 most valuable

	Id	Name	Value
▶	1	Quarter	0.25
	4	Lighter	2.15
	2	Keys	12.38
*			

In this screenshot we are demonstrating how we have added the ability for the user to be able to sort the things in their pocket. Once the user has a list of thingInMyPocket loaded up in the DataGrid "gridInventory" they will now be able to toggle the radio button called "3 least valuable" to be able to sort the value of the 3 cheapest items that the user may have and just displays them in the Data Grid view. This will help to make things easy for the user to sort and find specific items based on their value.

Sorting by Most Valuable



Things in my Pocket

Name: Vape

Value: $24.99

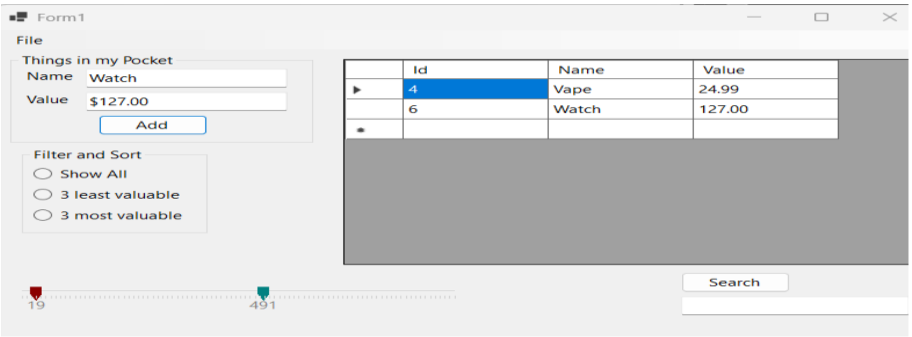
Filter and Sort

○ Show All
○ 3 least valuable
● 3 most valuable

	Id	Name	Value
▶	3	Phone	899.99
	5	Vape	24.99
	2	Keys	12.38
*			

In this screenshot we are demonstrating how we have added the ability for the user to be able to sort the things in their pocket, this time based on the most valuable items. Once the user has a list of thingInMyPocket loaded up in the DataGrid "gridInventory" they will now be able to toggle the radio button called "3 most valuable" to be able to sort the value of the 3 most expensive items that the user may have and just displays them in the Data Grid view. This will help to make things easy for the user to sort and find specific items based on their value.

Track Bar



Things in my Pocket

Name: Watch

Value: $127.00

Filter and Sort

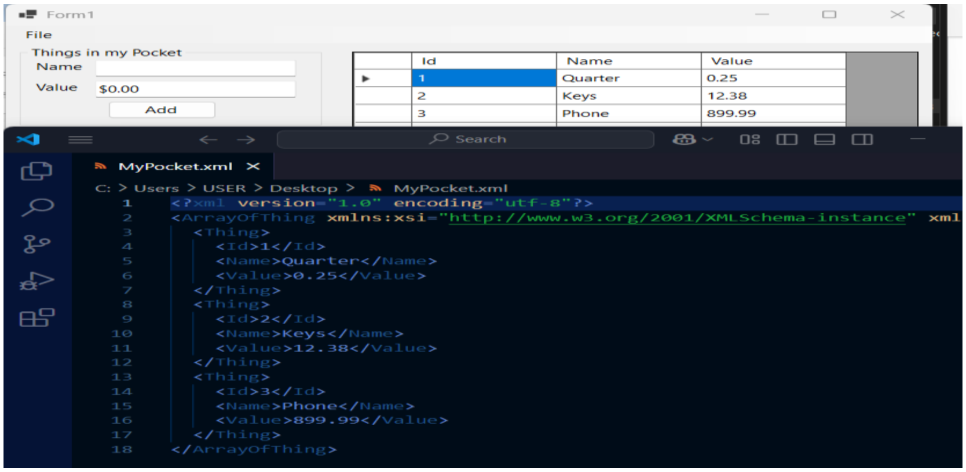
○ Show All
○ 3 least valuable
○ 3 most valuable

	Id	Name	Value
▶	4	Vape	24.99
	6	Watch	127.00
*			

19 491

In this screenshot we are demonstrating for we have added the ability for the user to be able to use a Track bar to control what items will show up in girdInventory based on the value range that is selected by the user. The Track bar will have two little arrows controlling the ranges, one being red which is responsible for the lowest value that will be looked for in the range. Then we have the green arrow which will be responsible for controlling what the make value in the range will be. Once the user slides those to their desired spots, then the data grid will display the items that have a value within the range of numbers seen in the track bar. In our screenshot an example of the lowest value being 19 and the largest value being 491 and only items with a value between these two ranges will be displayed for the user.

Adding XML Saving and Loading Functionality



Name

Value $0.00

	Id	Name	Value
▶	1	Quarter	0.25
	2	Keys	12.38
	3	Phone	899.99

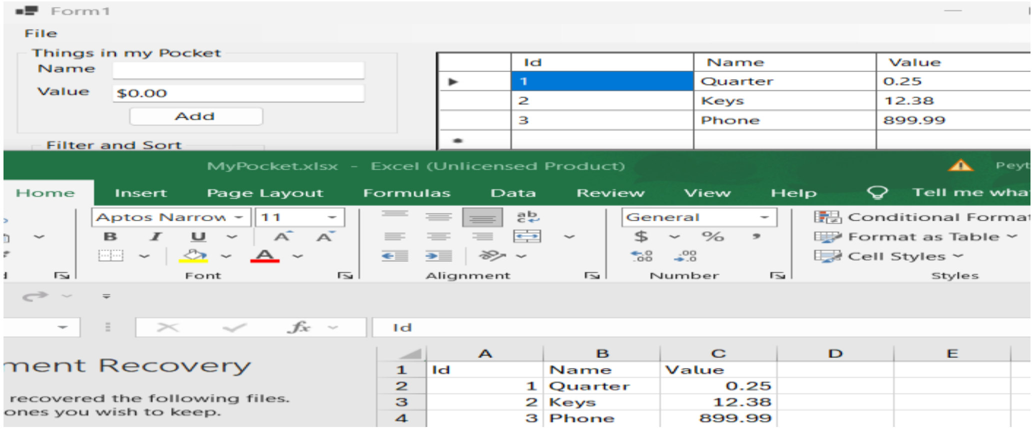
MyPocket.xml

C: > Users > USER > Desktop > MyPocket.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ArrayOfThing xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xml
  <Thing>
    <Id>1</Id>
    <Name>Quarter</Name>
    <Value>0.25</Value>
  </Thing>
  <Thing>
    <Id>2</Id>
    <Name>Keys</Name>
    <Value>12.38</Value>
  </Thing>
  <Thing>
    <Id>3</Id>
    <Name>Phone</Name>
    <Value>899.99</Value>
  </Thing>
</ArrayOfThing>
```

In this screenshot we are demonstrating how we have added the ability for the user to be able to save and load the information in the Data Grid to and from an xml file. When the user has all the information input that they want saved they can next click on the menu strip in the upper left-hand corner of the program and click on “File”. Next, they will click on “Save”, from here the user will be prompted to save the file to their computer or desktop just like any other program that allows for saving to files. Then if teh user is looking to load information from a xml file they would click on “File” in the menu strip. From there they will click on load and then select the xml file they would like to grab information from to be displayed in the data grid for easy readability.

Adding Excel Functionality



Name

Value $0.00

	Id	Name	Value
▶	1	Quarter	0.25
	2	Keys	12.38
	3	Phone	899.99

Id

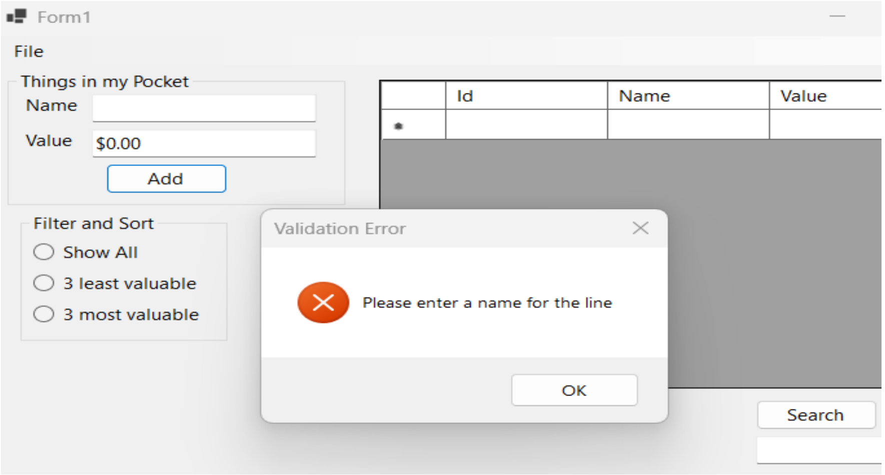
ment Recovery

recovered the following files.
ones you wish to keep.

	A	B	C	D	E
1	Id	Name	Value		
2	1	Quarter	0.25		
3	2	Keys	12.38		
4	3	Phone	899.99		

In this screenshot we are demonstrating how we have added the ability for the user to be able to save to and load the information in the Data Grid for Excel files. When the user has all the information input that they want saved they can next click on the menu strip in the upper left-hand corner of the program and click on “File”. Next, they will click on “Save”, from here the user will be prompted to save the file to their computer or desktop just like any other program that allows for saving to files. Then if the user is looking to load information from an Excel file they would click on “File” in the menu strip. From there they will click on load and then select the Excel file they would like to grab information from to be displayed in the data grid for easy readability.

Enhanced Data Validation



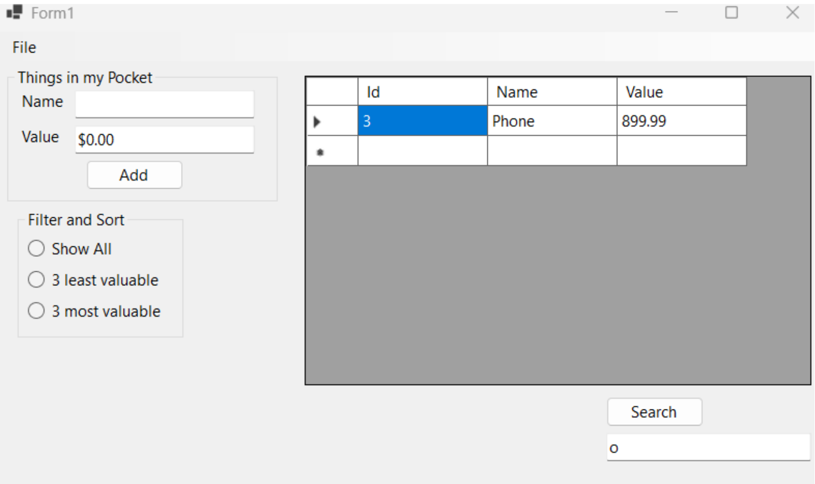
Value $0.00

Validation Error

Please enter a name for the line

In this screenshot we are just demonstrating how we have added some validation errors handling to help ensure the user does not try to submit any empty fields. If the user tries to leave the “Name” field blank and/or does not enter in a value, then they will receive the following error message.

Search Functionality



Name:

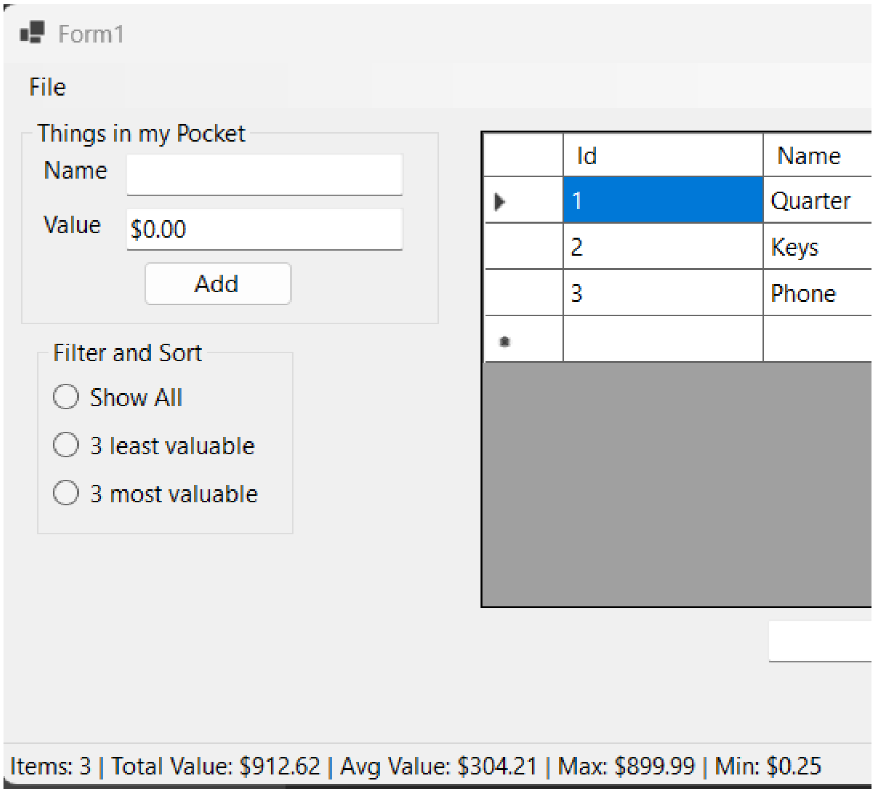
Value: $0.00

Id	Name	Value
3	Phone	899.99

Search: o

In this screenshot we are demonstrating how we have added the ability for the user to be able to search for specific items in their pocket. If the user has to many things in their pocket and they need to find a specific item, they can click on the textbox below the data grid and try in the first letters of the item they are trying to find then click on the "Search" button and it will clear all results from the data grid beside the letters inputted that match item names/partial letter match and display only the results in the grid for the user to review.

Summary Stats



Name:

Value: $0.00

Id	Name
1	Quarter
2	Keys
3	Phone

Items: 3 | Total Value: $912.62 | Avg Value: $304.21 | Max: $899.99 | Min: $0.25

In this screenshot we are demonstrating how we have added the ability for the user to be able to trach different metrics about the data in their grid. From whatever information that the user has in their data grid for the things in their pocket, then at the bottom of the program's page they will get a total item count, a total value of the items in the users pcoket, the average value per item, the highest value item and the lowest value item.

1. What was challenging?
   I would have to say that the most challenging aspect of this project would have been the handling of multiple file formats (like JSON, CSV, EXCEL, TXT and XLSX) with consistent error handling and validation. Ensuring that the data remained correct when saving, loading, and editing required careful attention to serialization and deserialization logic. Additionally, when implementing the real-time filtering, sorting and statistics updates while maintaining performance and keeping clean code was very tricky.
2. What did you learn?
   Throughout this project I would have to say that I have gained a deeper understanding of data bidning in WinForms, including how BindingSource works with collections and grids. I have also improved my skills in file I/O operations, working with external libraries like EPPlus for Excel and ServiceStack for JSON/CSV serialization.
3. How would you improve on the project?
   I would say that to enhance this project I would refractor the code to follow the "Repository Pattern" separating file operations from the UI logic for better maintainability. Adding in unit test would help to ensure reliability, especially for file parsing and validation logic. The UI could be improved and with "async/await" for file operations to prevent freezing during large files loads.
4. How can you use what you learned on the job?

I would have to say that the skills in this project are highly transferable to real-world applications, particularly in business software where data import/export, validation, and reporting are common requirements. Understanding file handling is very useful for integration with other systems, while the data binding and LINQ experience applies to database operations for building robust applications.