Master in Artificial Intelligence
Facultat d'Informàtica de Barcelona
Universitat Politècnica de Catalunya

Money Tracker User Manual

Gilles Charlier
Jason Klimack

2021-01-14

Contents

1 Description of the System's Purpose

The Money Tracker application is designed for the use of tracking the users expenditures. This tracking of money being spent can then aid the user in reducing unnecessary purchases, in order to save money.

The user uploads images of their purchase receipts from the store, and the money tracker will automatically categorize the products purchased, as well as the price of each product. A complete list of the available categories is shown below:

- Appliances
- Electronics
- Grocery Gourmet Food
- Home Kitchen
- Musical Instruments
- Office Products
- Patio Lawn Garden
- Pet Supplies
- Sports Outdoors
- Tools Home Improvement
- Other

As described in section 3.1, the user may manually add new categories that do not appear in this list.

After grouping the products into categories, the amount of money spent on each product category is then saved to a database. The information from the database is used to display information to the user in a series of charts, described in section 3.

2 Start-up/Shutdown of the System

2.1 Installation

The Money Tracker Application does not need to be installed. It is ready to be used out-of-the-box. However, there is an external package that is required in order for the Money Tracker Application to work. This application is the tesseract-ocr module. It is recommended to use **Linux** OS, simply for the installation procedure of the tesseract-ocr package. This can be installed with the following line in the terminal on a linux OS:

```
sudo apt install tesseract-ocr
```

Furthermore, the application was implemented in python, and requires a number of non-standard python libraries. These libraries are listed below:

- PIL
- pandas
- Keras
- tensorflow
- nltk
- cv2
- pytesseract
- shap

2.2 Startup

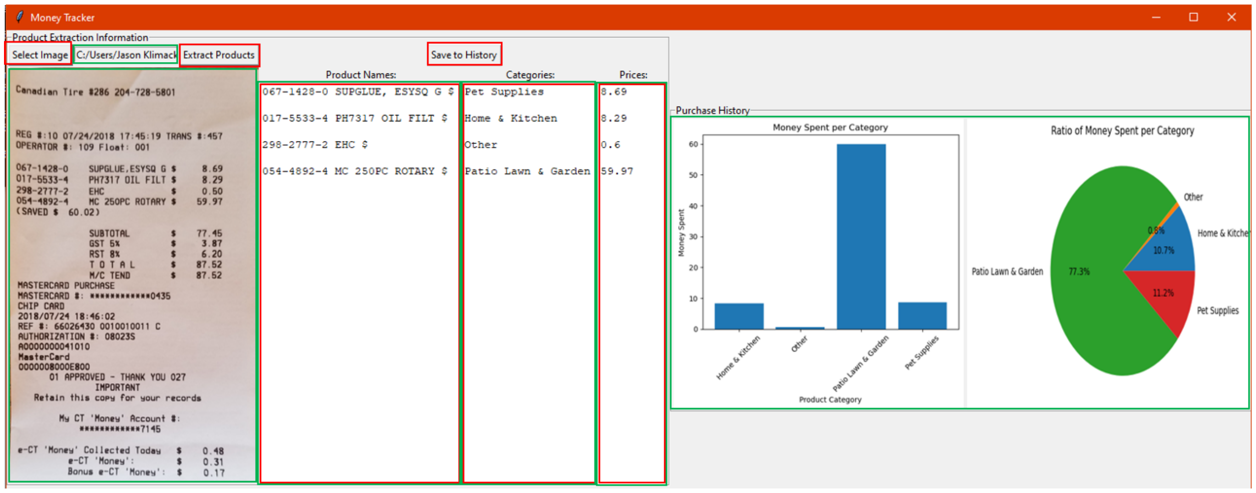
There is no executable for the system. In order to run the program, the python script file `UI.py` must be ran from command line using the following instruction:

```
python UI.py
```

This command will load and open the Money Tracker Application, as long as all system requirements described in section 2.1.

2.3 Shutdown

In order to shutdown the system, the user interface simply needs to be closed. This is done by clicking with the mouse on the red X in the top-right corner of the screen.



```
Canadian Tire #286 204-728-5801

REG #:10 07/24/2018 17:45:19 TRANS #:457
OPERATOR #: 109 Float: 001

067-1428-0    SUPGLUE,ESYSQ G $     8.69
017-5533-4    PH7317 OIL FILT $     8.29
298-2777-2    EHC             $     0.50
054-4892-4    MC 250PC ROTARY $    59.97
(SAVED $  60.02)

              SUBTOTAL        $    77.45
              GST 5%          $     3.87
              RST 8%          $     6.20
              T O T A L       $    87.52
              M/C TEND        $    87.52
MASTERCARD PURCHASE
MASTERCARD #: ************0435
CHIP CARD
2018/07/24 18:46:02
REF #: 66026430 0010010011 C
AUTHORIZATION #: 08023S
A0000000041010
MasterCard
0000008000E800
    01 APPROVED - THANK YOU 027
           IMPORTANT
  Retain this copy for your records

       My CT 'Money' Account #:
          ************7145

e-CT 'Money' Collected Today  $    0.48
          e-CT 'Money':       $    0.31
      Bonus e-CT 'Money':     $    0.17
```

Product Names:	Categories:	Prices:
067-1428-0 SUPGLUE, ESYSQ G $	Pet Supplies	8.69
017-5533-4 PH7317 OIL FILT $	Home & Kitchen	8.29
298-2777-2 EHC $	Other	0.6
054-4892-4 MC 250PC ROTARY $	Patio Lawn & Garden	59.97

Figure 1: Input/Output devices of the Money Tracker Application. Input devices consist of buttons and text-fields and are marked by red bounding boxes, and output devices consist of images and text-fields and are marked by green bounding boxes. The central text-fields have both green and red bounding-boxes because they are both input and output devices.

3 User Interactions of the System

There are six different locations on the UI for the user to input information and perform actions: three buttons that can be clicked using a cursor, and three text areas that can be filled with information from the keyboard. Section 3.1 explains the proper usage of these input mechanisms, as well as the expected output of the system after performing these actions. Section 3.2 describes possible error messages that can arise due to improper use of the system, or lack of system requirements being met.

3.1 Input / Output

The user can input information in two different methods: selecting an image and uploading it to the program, or manually adding the information on their receipt by typing on their keyboard. Input devices are highlighted in red bounding-boxes in figure 1.

The primary method of input to the system is uploading an image of the users receipt. This is done by using the `Select Image` button in the upper-left corner of the window. The button brings up a pop-up window that allows the user to browse their computer for a file. The user can then select an image file, and then click open. The resulting image will then appear in the UI on the left-hand side of the window with the path to the image displayed in a text-field above it, as shown in figure 1. An error will occur if the user selects a non-image, or does not select any file at all when browsing the files on their computer. See section 3.2 for more details on errors.

After selecting the desired image, the user can click on the `Extract Products` button, in order to process the image information. At this time, the product names, categories, and prices are extracted and predicted from the contents of the image provided. This information is then displayed in three columns: `Product Names`, `Product Categories`, and `Prices`. Each column consists of a single line representing each product, with a blank line separating the products. In the case that any name, category, or price is too long to fit within the column, the excess contents of the line wrap-around to the next line.

At this time the user can then review the contents of the extracted information by manually making changes to the columns. These changes may be necessary in the event that the product information is not correctly interpreted from the original input image. Also, as the secondary main input to the system, the user can simply add the information from the receipt directly into these columns if they desire, rather than uploading the image and automatically extracting the information. Furthermore, if the user would like one of their products put into a different category than the predefined list in section 1, the user simply just needs to modify the category displayed for the product they wish to change.

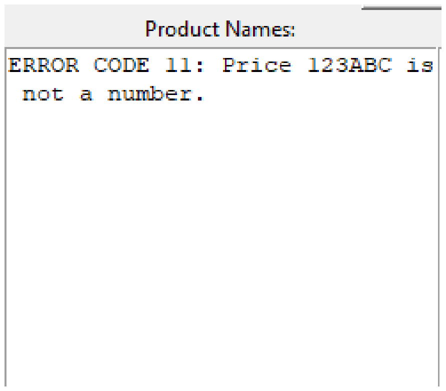
After all corrections are performed, the user clicks on the `Save to History` button, in order to save the displayed information to a database on the users computer. The new information is appended to a CSV file called `history.csv` that already exists, and contains the previous purchase history of the user.

Once the information in the database is updated, there are two charts describing the users purchase history: a bar chart and a pie chart. These charts are updated every time the user starts up the program, or presses the `Save to History` button. The bar chart shows the amount of money that the user has spent on each product category, and the pie chart shows the ratio of money spent on each of the product categories, in order to get an understanding as to where the most money is being spent.

3.2 List of Error Messages

There are a number of possible errors that can occur due to incorrect usage of the application, or corruption of the data files. This section outlines a list of all perceived errors. The errors are listed below by the error code that occurs:

1. `No file has been selected.`
2. `Could not load file <`*`filename`*`>. Please ensure the file exists, and is an image.`
3. `Could not extract text from image <`*`filename`*`>. Please ensure the file exists, and is an image. If the problem continues please make sure that tesseract-ocr is properly installed on your computer. It is recommended to use Linux OS. If you are using Linux, then tesseract-ocr can be installed by the following command in the terminal: sudo apt install tesseract-ocr`


Product Names:

```
ERROR CODE 11: Price 123ABC is
 not a number.
```

Figure 2: Example of an error code and message that appear in the UI.

4. `Reading product/price information from the extracted receipt failed.`
5. `Insufficient product and/or price information found on the receipt image.`
6. `Product classification failed.`
7. `Failed to read data file:` <*filename*>`. Please ensure the file exists, and is of type CSV.`
8. `Failed to read data file:` <*filename*>`. Ensure the contents of the CSV file contain only two columns: 'category' and 'Price'. Also check that there are only character [0-9] or '.' under the price column.`
9. `Failed to plot spending history data. Ensure the 'temp' directory still exists.`
10. `Failed to write to data file` <*filename*>`. Enusure the file exists.`
11. `Price` <*VALUE*> `is not a number.`

Whenever an error occurs, the error code along with the message displayed above will appear in the `Product Names` column of the UI, as shown in figure 2.

4 Examples of Use

This section outlines some example usage of the Money Tracker Application. These examples begin with a new data file, that has no entries inserted, and accumulate data as the examples progress. There are three examples: Product Classification without Changes, Product Classification with Changes, and Product Insertion without the Image.

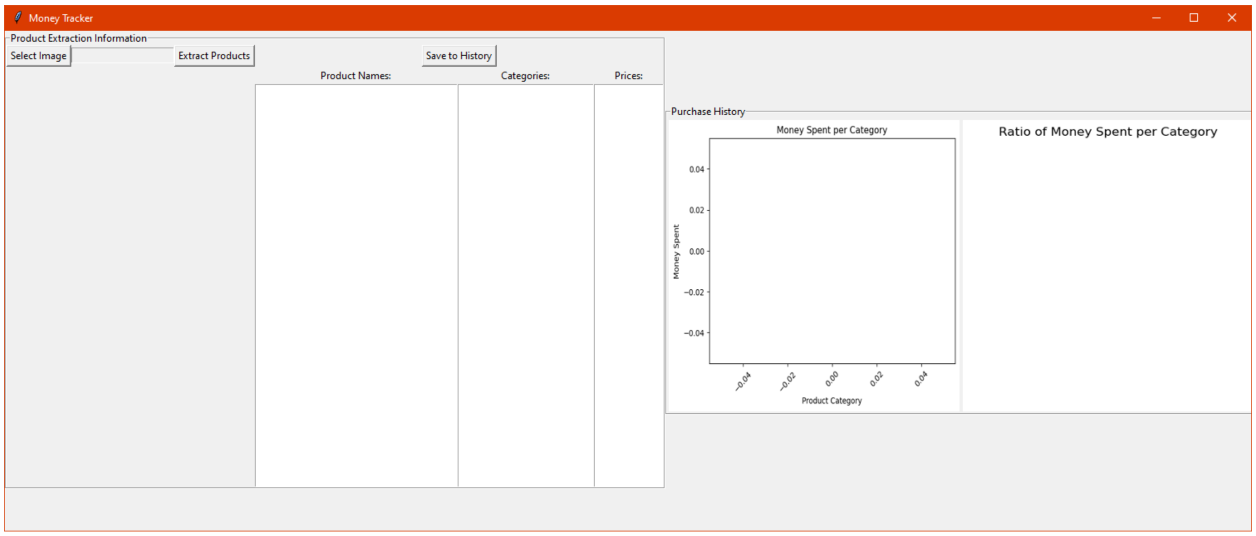


Figure 3: Initial state of the program after opening for the first time.

4.1 Product Classification without Changes

This example describes the basic usage of the system, from start to finish. Initially, upon the first time the application is opened, the data file will be empty, resulting in a blank window as shown in figure 3.

The first step is to select an image of a receipt by clicking on the `Select Image` button. Browse to the location of the image file and open it. Now there should be an image of the receipt displayed on the screen, along with the path to the image file above it, as shown in figure 4.

The `Extract Products` button is then used to extract the information from the receipt image, and display the results in the three columns: `Product Names`, `Categories`, and `Prices`. The result is shown in figure 5.

Finally, after the information is extracted from the image, the `Save to History` button can be clicked in order to save all of the category and price data to a CSV data file located on the users computer. After the information is saved to the data file, the two plots are updated with the new information. The results are displayed in figure 6.

And that is it! The amount of money spent on different product categories has now been extracted from the receipt, saved in a data file, and displayed in two charts. The first chart demonstrates the amount of money the user has spent in total on each category, while the second shows a pie chart so the user can see proportions of money spent compared amongst the different categories.

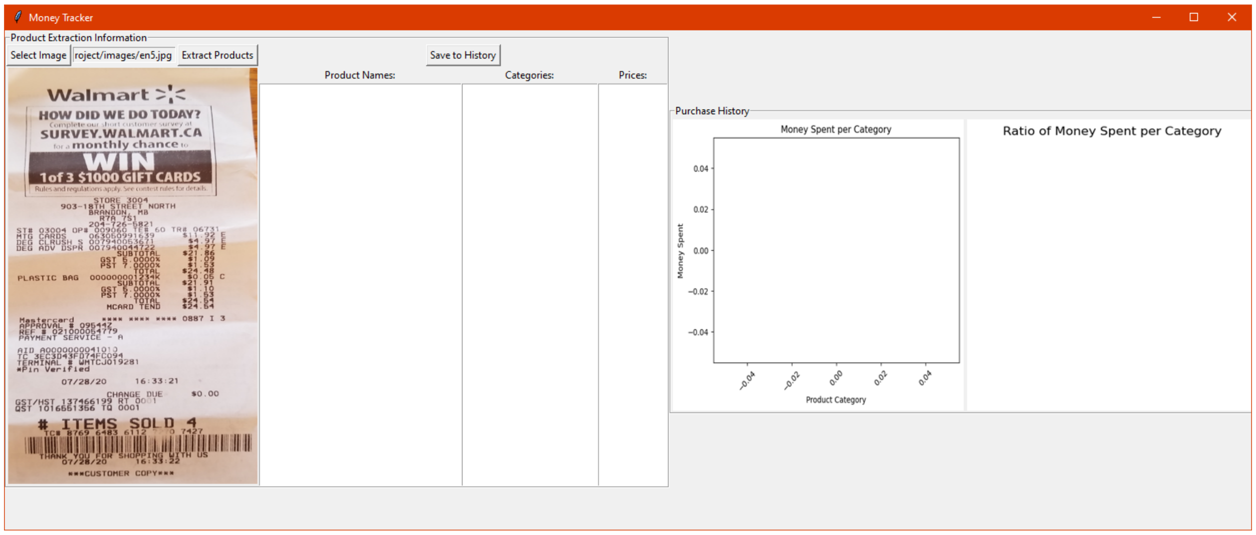
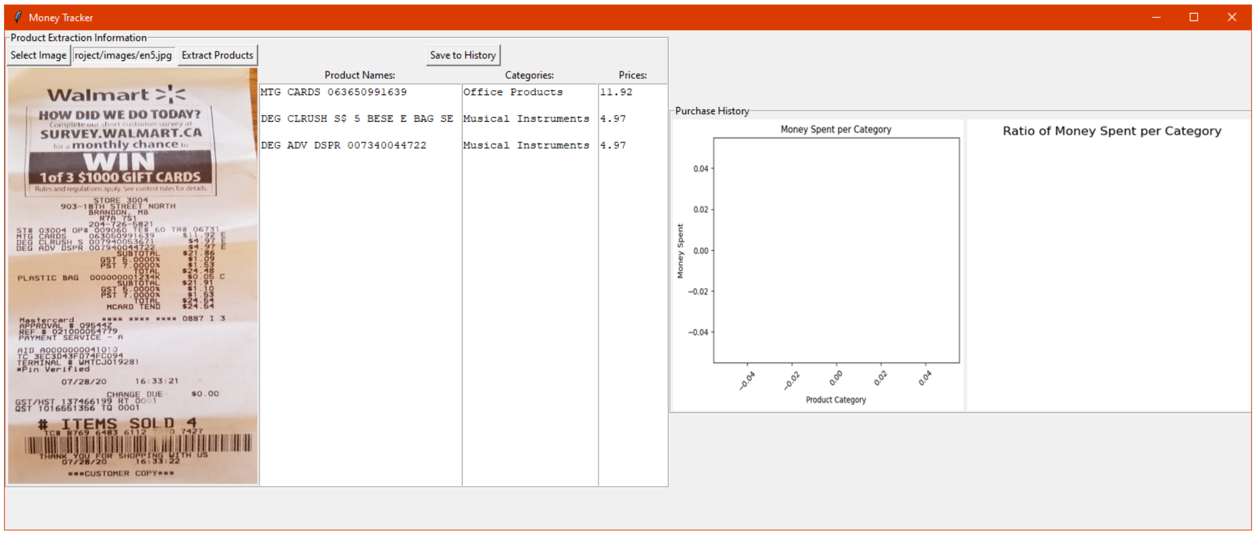


Figure 4: Resulting view after selecting the first receipt image.



Product Names:	Categories:	Prices:
MTG CARDS 063650991639	Office Products	11.92
DEG CLRUSH S$ 5 BESE E BAG SE	Musical Instruments	4.97
DEG ADV DSPR 007340044722	Musical Instruments	4.97

Figure 5: Results after the `Extract Products` button has been clicked.

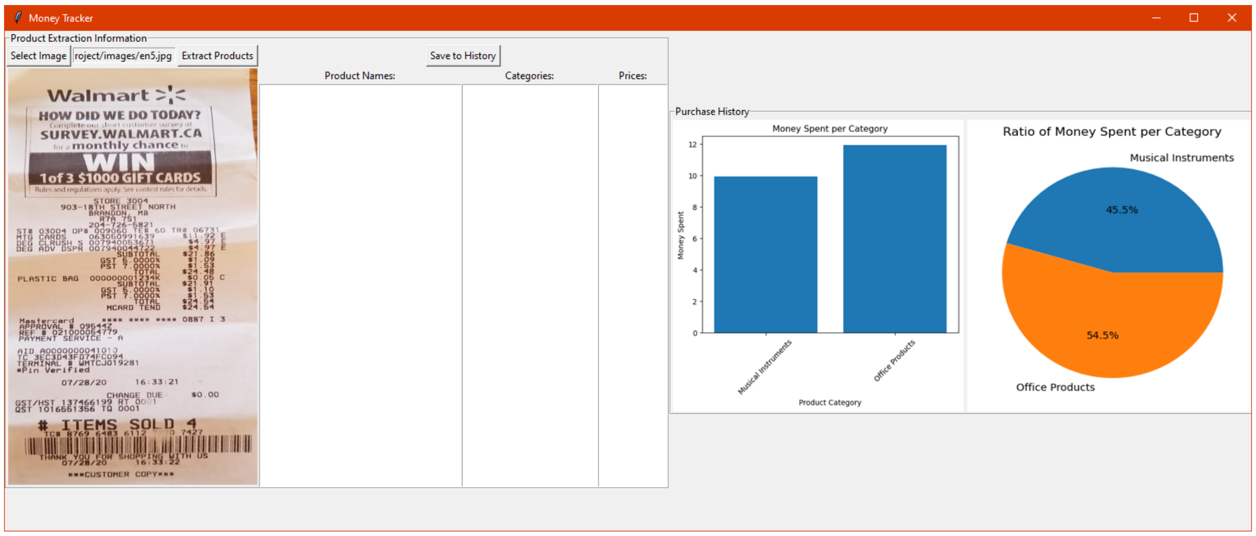
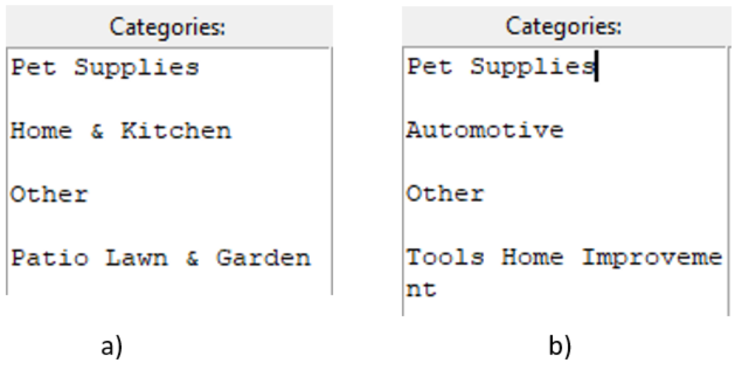


Figure 6: Results after the `Save to History` button has been clicked.



a)

Categories:
Pet Supplies
Home & Kitchen
Other
Patio Lawn & Garden

b)

Categories:
Pet Supplies
Automotive
Other
Tools Home Improveme nt

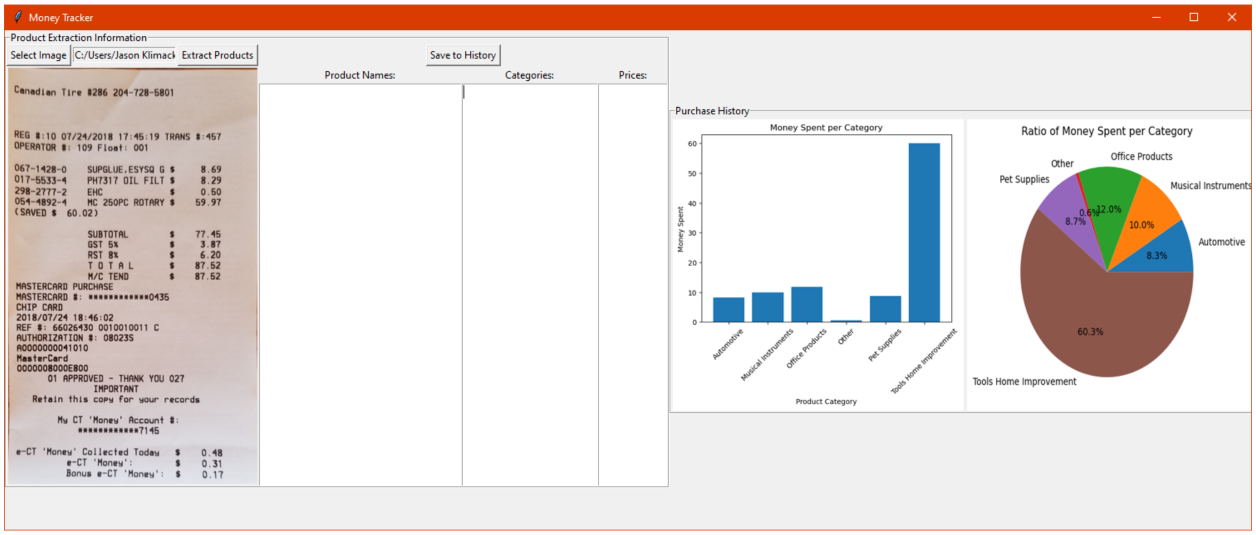
Figure 7: Demonstration showing the manual modification of the product categories: a) original categories as output from the extraction process, and b) modified product categories.

4.2 Product Classification with Changes

This example covers how to manually make changes to the product details after the information has been extracted from the receipt image.

Let us assume that a new image receipt has been selected, and the information extracted by following the beginning steps in the previous example, but the category information for one of the products is incorrect. This can occur when the product name is not descriptive enough for the system to know the correct category, or the item it completely new to the system. When this occurs, the user can manually edit the category column so that the results are correct. Figure 8 shows a demonstration.

The manual modification is not limited to the `Categories` column. If there is an error in either the



```
Canadian Tire #286 204-728-5801

REG #:10 07/24/2018 17:45:19 TRANS #:457
OPERATOR #: 109 Float: 001

067-1428-0    SUPGLUE,ESYSQ G $     8.69
017-5533-4    PH7317 OIL FILT $     8.29
298-2777-2    EHC             $     0.50
054-4892-4    MC 250PC ROTARY $    59.97
(SAVED $  60.02)

              SUBTOTAL        $    77.45
              GST 5%          $     3.87
              RST 8%          $     6.20
              T O T A L       $    87.52
              M/C TEND        $    87.52
MASTERCARD PURCHASE
MASTERCARD #: ************0435
CHIP CARD
2018/07/24 18:46:02
REF #: 66026430 0010010011 C
AUTHORIZATION #: 08023S
A0000000041010
MasterCard
0000008000E800
      01 APPROVED - THANK YOU 027
              IMPORTANT
   Retain this copy for your records

        My CT 'Money' Account #:
           ************7145

e-CT 'Money' Collected Today   $    0.48
           e-CT 'Money':       $    0.31
           Bonus e-CT 'Money': $    0.17
```

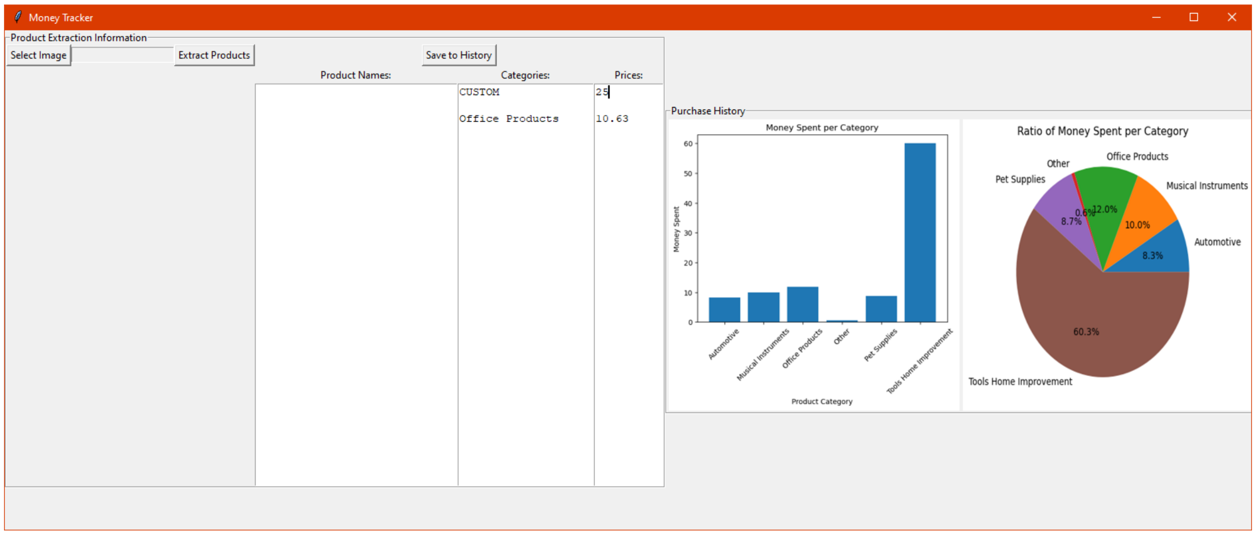
Figure 8: Resulting state of the application after the new changes are applied. Note the new categories added to the Purchase History charts.

Product Names or `Prices` columns, the user can modify these values as well. Furthermore, it is possible for the user to create a new product category simply by typing a new name in the column. For example, Automotive is not one of the items in the list of categories in section 1, but it is one of the manual changes performed by the user in this example. After the changes are applied, the user can save the new products as done before. Figure **??** shows the final results.

4.3 Product Insertion without using an Image

This final example shows how the user can insert data into the program without the need to upload an image. All that is needed is for the user to manually insert the category and price information for the products that they wish to add to the data file. Figure 9 demonstrates the necessary fields that need to be filled out. Note, it is important that the rows of the categories are aligned with the rows of the prices, or the information will be ignored.

The final result of the application is shown in figure 10.



Product Names:	Categories:	Prices:
	CUSTOM	25
	Office Products	10.63

Figure 9: Demonstration of necessary text-fields that need information in order to submit information without the use of an image.

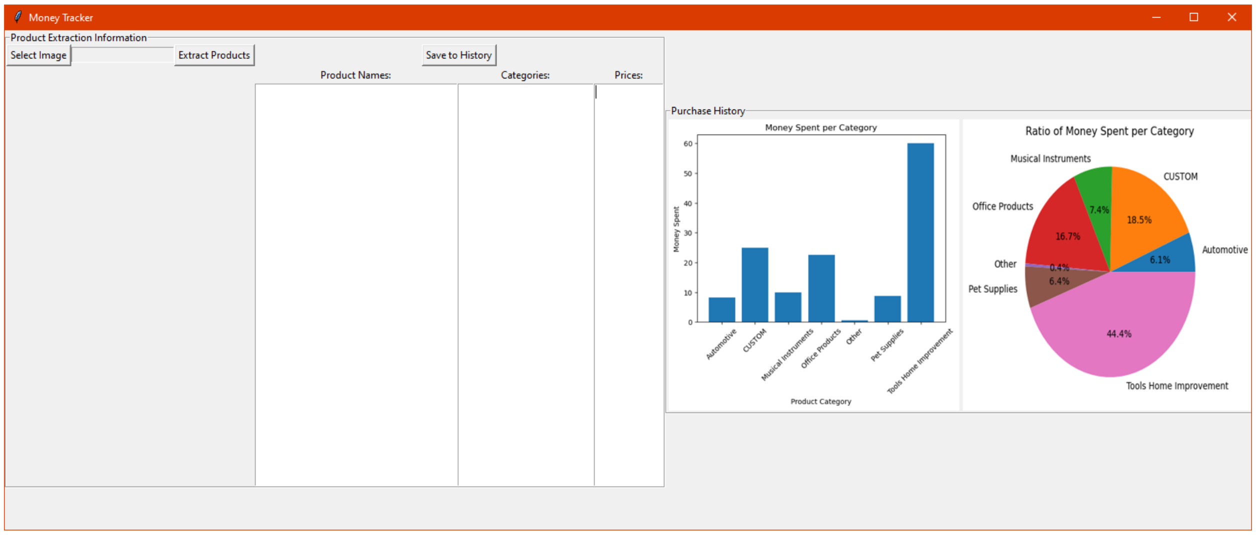


Figure 10: Final view of the application at the end of the examples. Note the increase in product categories to the plots as new categories are introduced, compared against the plots in the previous examples.