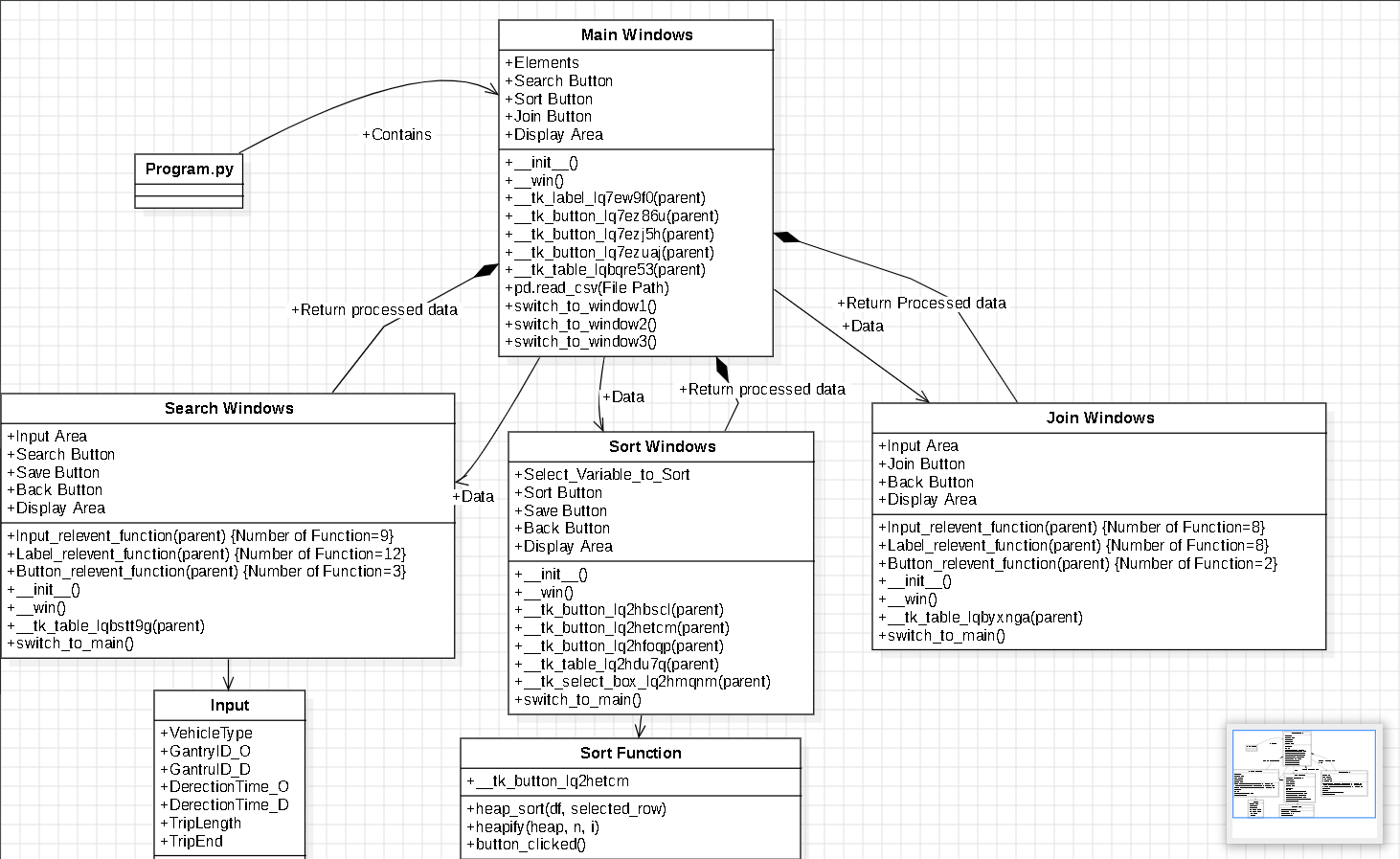
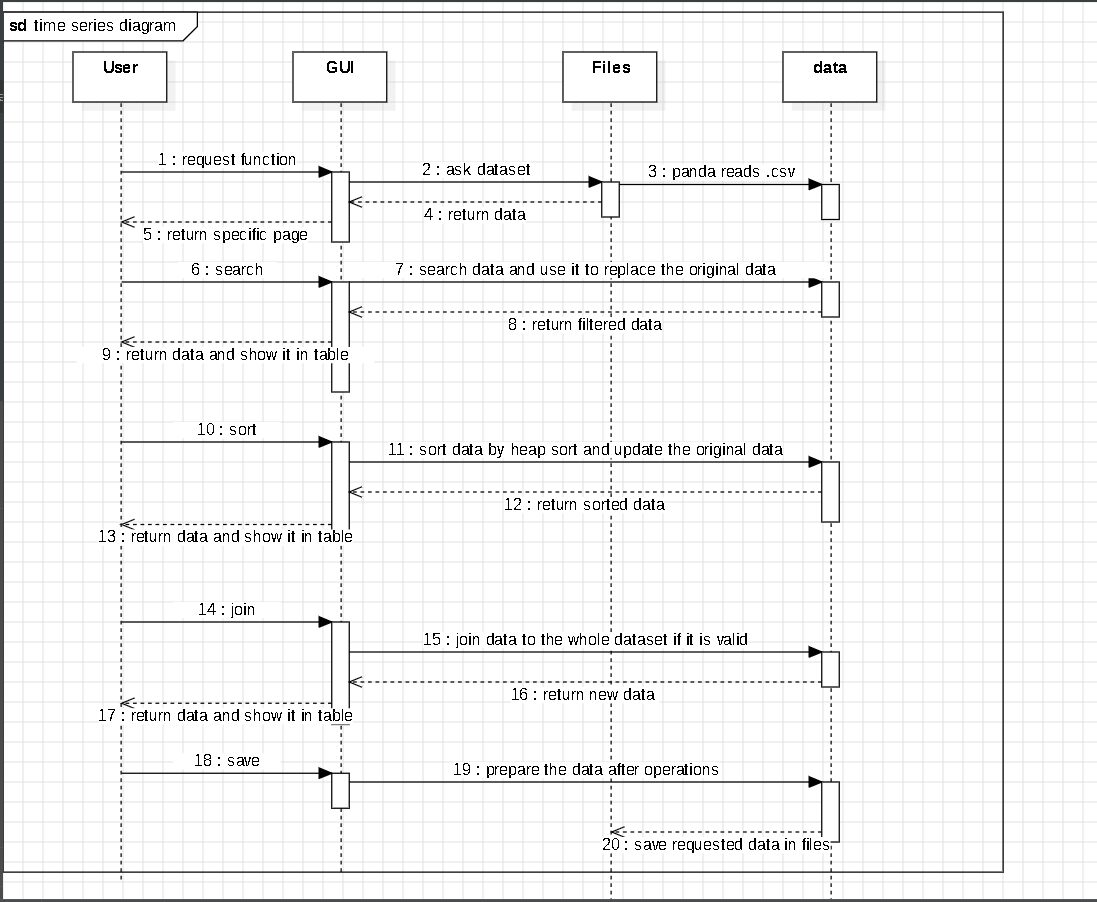
**MSDM 5051 Final project report**

LI Ruizhe, ZHENG Zeyu, OUYANG Chulei

1. **UML design and Flowchat**

The UML diagram here shows the overall structure of our code, in which we include a main window and three sub-windows. The upper part of the each UML class shows the areas owned by each window and their corresponding functions, and the lower part shows the functions and related parameters of the code we used to build this window. In the Search sub-window, we give the data that can be input. In the Sort sub-window, we give the function name for heap sort.

****

The time series diagram illustrates the whole data flow and interaction between users and inqury system. When users first enter the main page, they will find three buttons to choose the service they want, including ‘search’,’sort’ and ‘join’. They can find the table below the main window displaying the original data. After clicking the buttons, users will get specific page to check and operate the data. After operation, the ‘save’ button in every page will make sure the new dataset can be recorded by files.****

* 1. **Traffic Data**

We use the Taiwan transportation dataset TDCS\_M06A\_20190830\_080000.csv. In our project, we read the data through the pd.read\_csv() function of pandas and convert it to the dataframe format. This dataset has 8 columns of data:

VehicleType: 31, 32, 41, 42, 5;

DerectionTime\_O:The time when the vehicle passes the first station;

Gantry\_O:the number of the first station the vehicle passed;

DerectionTime\_D: the time the vehicle passed the last station;

Gantry\_D:the number of the last station the vehicle passed;

TripLength: the distance traveled by the vehicle;

TripEnd:trip marker (Y normal end, N abnormal).

TripInformation:the time and number of each station the vehicle passed.

* 1. **Data Transmission**

When main\_window switches to other window, we assign the data of main\_window to other window by switch\_to\_window1() function to realize the data interoperability. At the same time, when the operation in the window is completed, the user clicks the **back** button to return to the main window, we call the swtich\_to\_main() function to assign the data of the other window to the main\_window, and at the same time, we call the tk\_button\_update() function to update the table of the data samples displayed in the main\_window.

1. **Interface design**
   1. **Label and button**

We set the labels in the interface via tk.Label and the buttons in the interface via tk.Button.



* 1. **Data Preview**

In each window, we have a section that shows the user a portion of the data along with their column names. This makes it easy for the user to understand the structure of the data.

* 1. **Search part**

In the search window, when the user clicks the search button, we will call the button\_clicked\_search() function to get all the keywords entered by the user and search for the eligible data based on these keywords, and then display these data to the table in the search interface.

* 1. **Sort part**

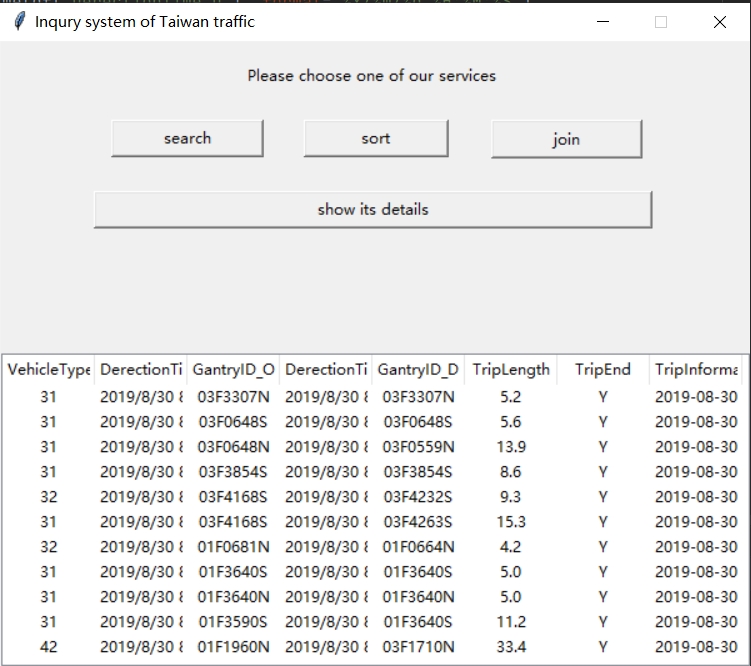
In the sort window, the user can choose to arrange the data by a certain column. The button\_clicked() function is called after clicking the sort button. This function uses the built-in heap sorting algorithm to sort the existing data by the column selected by the user and outputs a sample sort to the form.

* 1. **Join part**

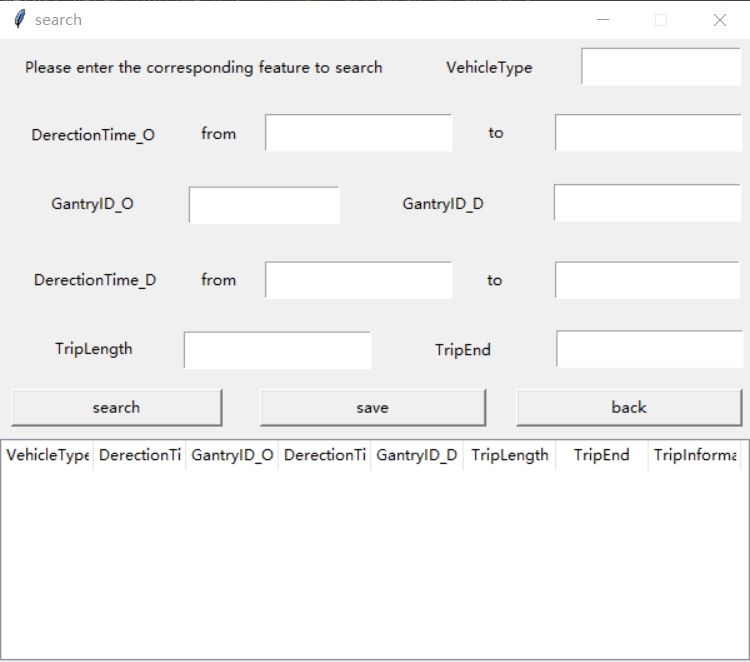
In the join window, the user can insert custom data into the dataframe. The user just needs to enter the value of each column and click the join button to insert the data. It is worth noting that the data type of "VehicleType" is int, the data type of "TripLength" is float, and the data type of "DerectionTime\_O" and "DerectionTime\_D" are time. If the entered data type does not match the actual data type, the system will display invalid and will not add this column to the dataset.

1. **Example of execution**

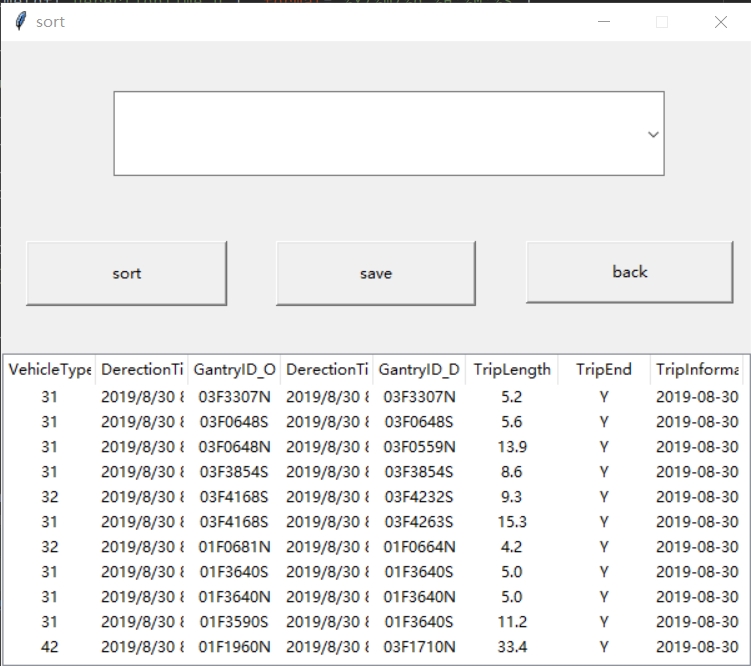
**Main window:**



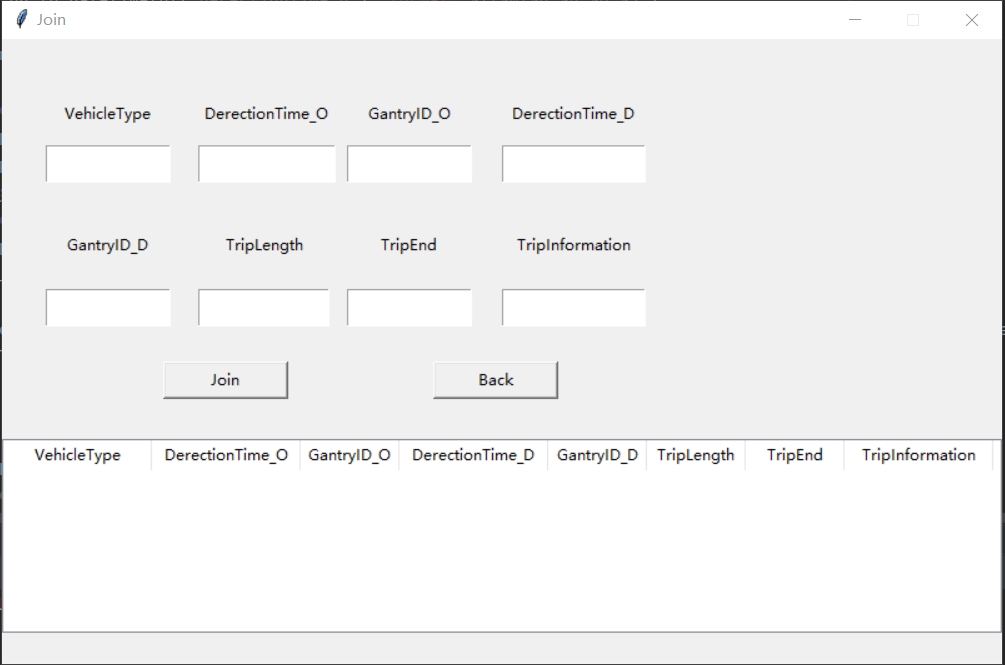
**Search window:**



**Sort window:**

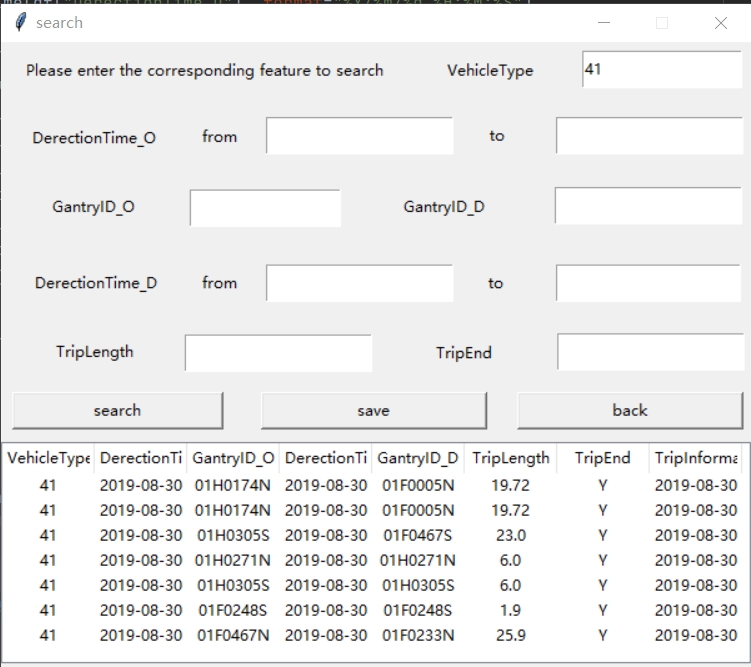


**Join window:**

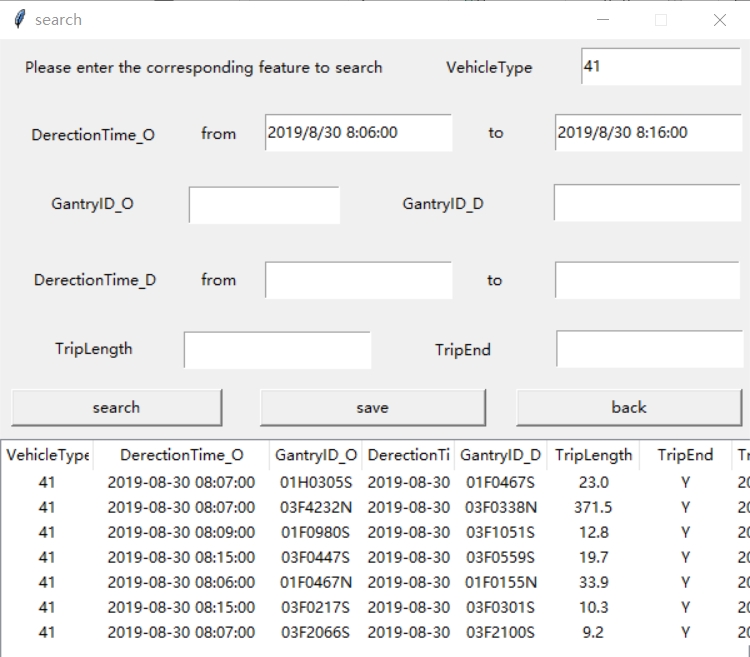


**Search function:**

Here we have selected the dataset with VehicleType=41 for the search operation.

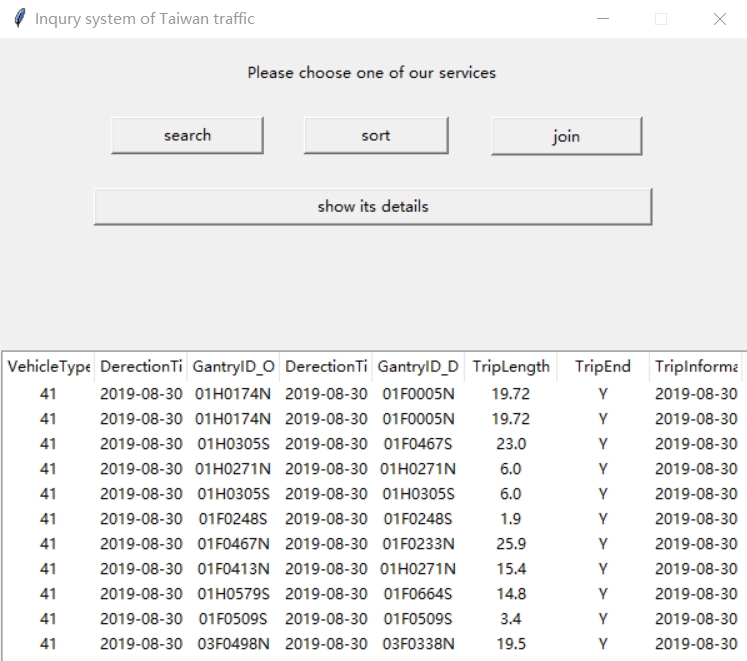


We have performed a SEARCH operation and selected data with ‘VehicleType’=41, ’DerectionTime\_O’ between 2019/8/30 8:06:00-2019/8/30 8:16:00



The Search window returns the search results to the main windows.

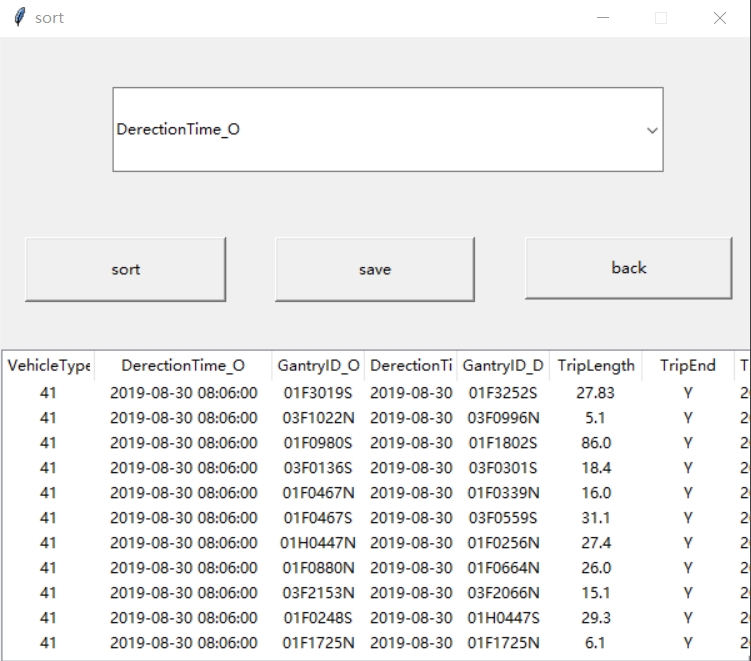
Then you can perform further operations on this processed data.



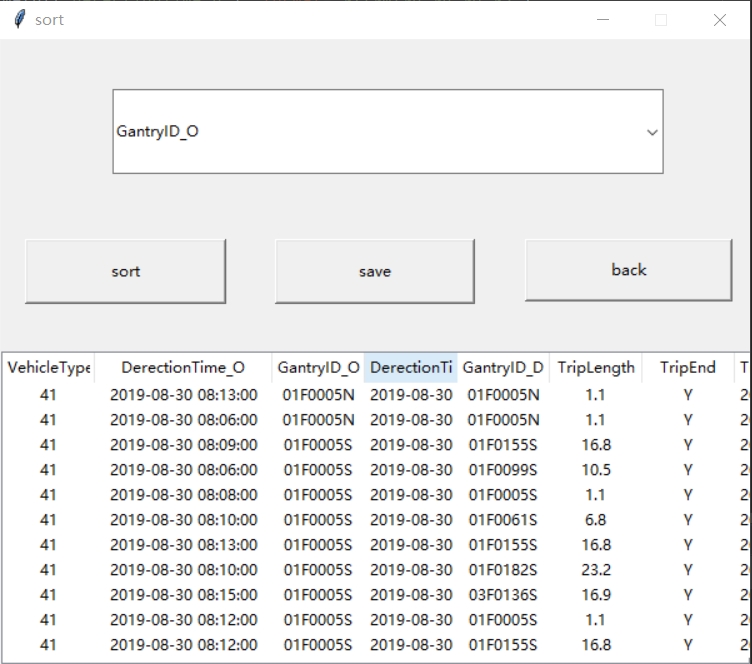
**Sort function:**

(In this section we have performed a SEARCH operation and selected data with ‘VehicleType’=41, ’DerectionTime\_O’ between 2019/8/30 8:06:00-2019/8/30 8:16:00)

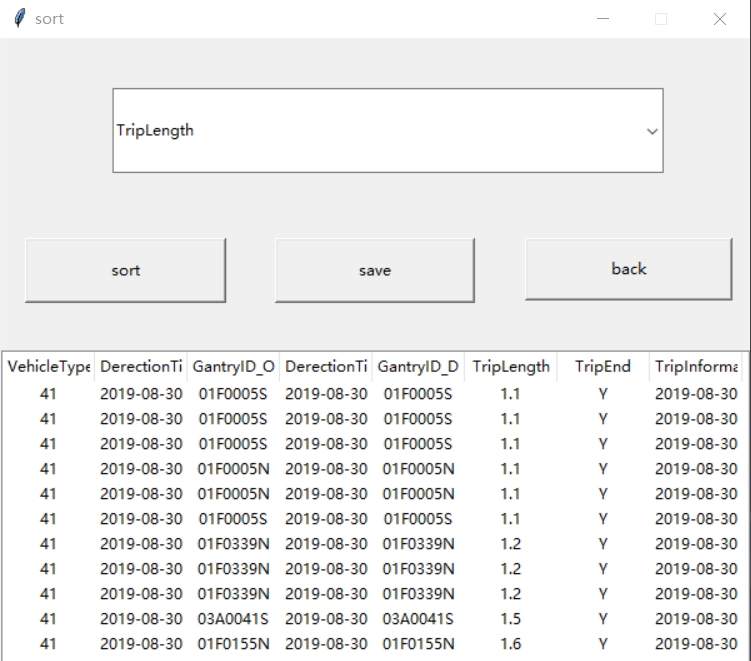
First we try to sort the DerectionTime\_O, because of the processed data, the least value is 2019-08-30 08:06:00.



Then we try the sort function on GantryID\_O.



At last, we try the sort function on the variable ‘TripLength’.



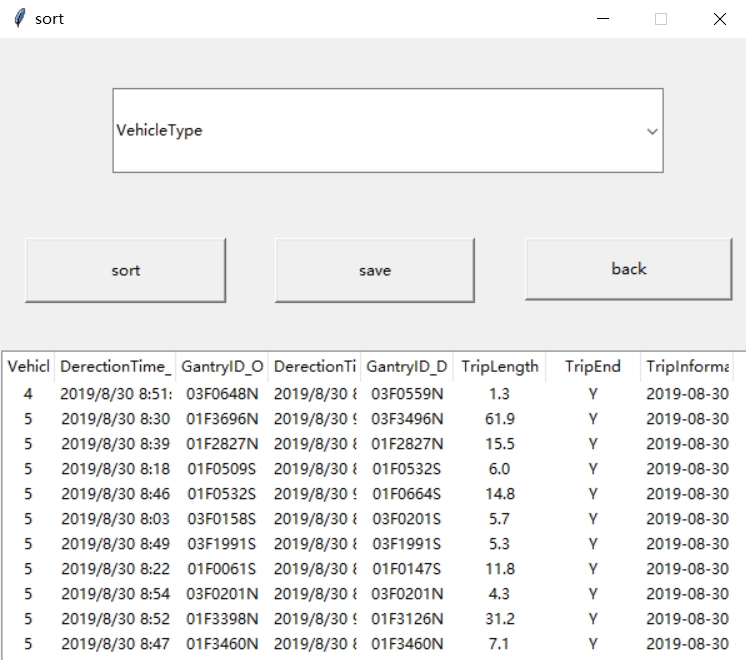
**Join function:**

Enter one piece of data to check the join function.

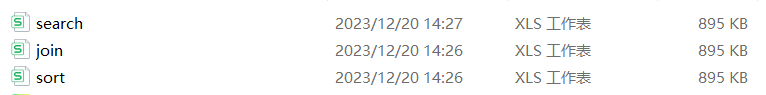


Then we use the sort function to check whether the join function can be run normally.

(From the picture we can check that there is a VehicleType=4 which is the observation we added just now.)



**Save function in three windows:**

****