

name-sorter

Author: Evi Juwita
Date: 20 October 2018

This project is generated as an employee skill assesment for PT. Kreatifitas Sinergisme Teknoindo.

Table of Contents

1	Introduction	3
2	Installation Manual	4
3	Technical Design Manual	6

Table of Figures

Drawing 1	Command Prompt Window	4
Drawing 2	Navigate to [name-sorter] Installation Location	4
Drawing 3	Running [name-sorter]	5
Drawing 4	[name-sorter] Class Diagram (Overall)	6
Drawing 5	[name-sorter] Class Diagram (Validator & Logger	7
Drawing 6	[name-sorter] Sequence Diagram	7

1. Introduction

The program [name-sorter] is a console application that will sort a list of names. User can submit a file path in text form, and [name-sorter] will extract the names listed on the file. Next, [name-sorter] sorts names based on surname, then by given name respectively. User can then view the result on screen and on an output file, called [sorted-names-list.txt] on the same directory as the input file.

There are two main points discussed in this document; installation manual and technical design manual. Installation manual elaborates on how to install [name-sorter] on PC / laptop (running on Windows for now), technical design manual describes more on [name-sorter] design that acts as technical design guidance and reference for future development.

2. Installation Manual

The generated [.exe] file should be available before starting the [name-sorter] installation. Besides, there is a folder named [name-sorter.installer] along with this document for this [name-sorter] installation purposes.

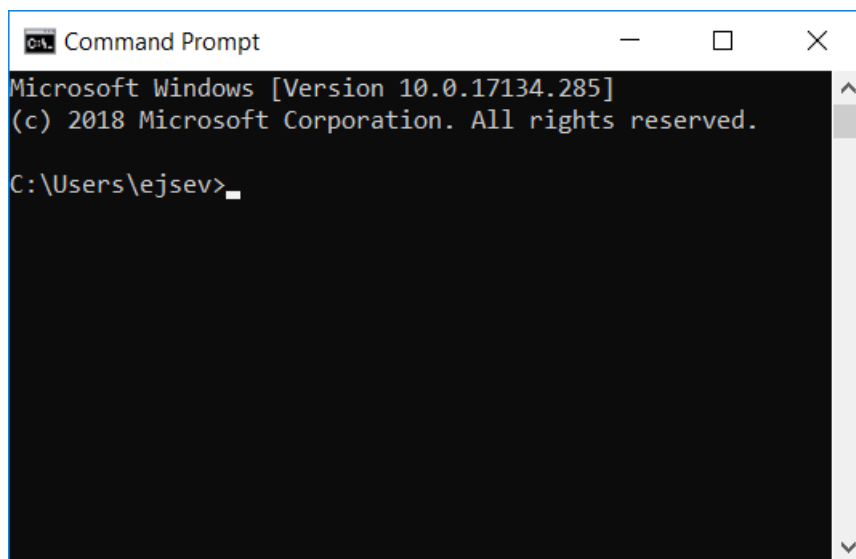
Installing [name-sorter] requires a user to simply copy and paste the entire files under [name-sorter.installer] directory. The same method also applies when installation files exist in a different folder. In the case when the installation files exist in a compressed form (or zipped form), then user may extract the installer to a desired location.

To check if the installation went properly, simply run [name-sorter] by following three steps below:

1. Open command prompt
[Windows key + r] ➤ [Type: cmd]

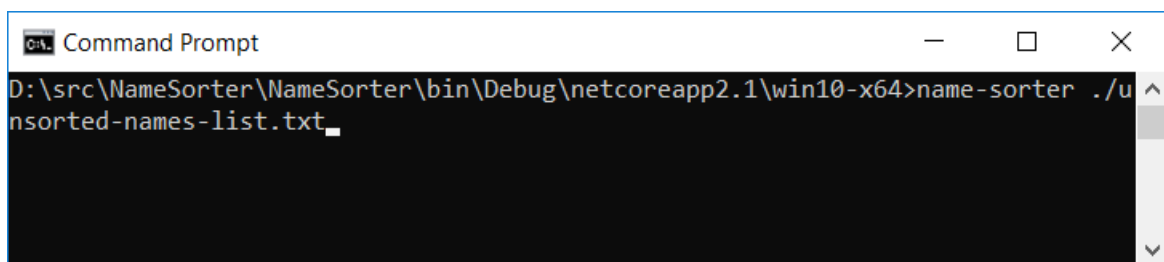
or

[Go to Start menu] ➤ [Windows System] ➤ [Command Prompt]



Drawing 1: Command Prompt Window

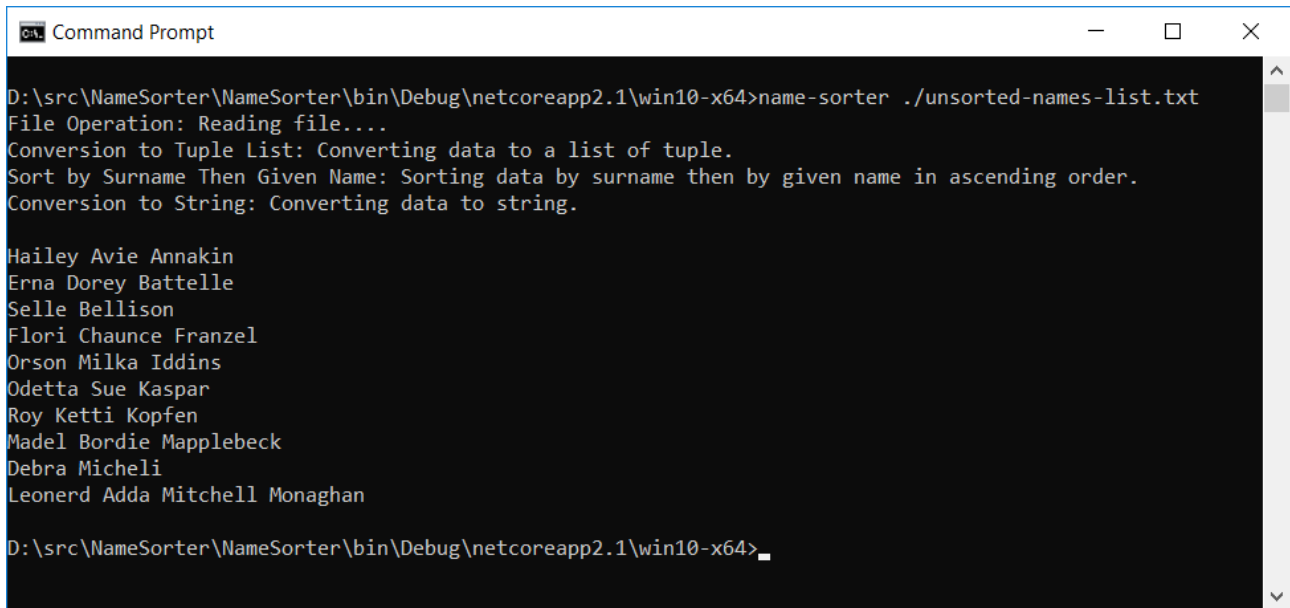
2. Run [name-sorter]
On command prompt, navigate to [name-sorter] folder ➤ [Type: name-sorter ./unsorted-names-list.txt]



Drawing 2: Navigate to [name-sorter] Installation Location

3. View result

There should be names listed in orderly manner printed on screen. For certainty, there should also be a file called [sorted-names-list.txt] under the same folder as [unsorted-names-list.txt] file.



```
Command Prompt
D:\src\NameSorter\NameSorter\bin\Debug\netcoreapp2.1\win10-x64>name-sorter ./unsorted-names-list.txt
File Operation: Reading file....
Conversion to Tuple List: Converting data to a list of tuple.
Sort by Surname Then Given Name: Sorting data by surname then by given name in ascending order.
Conversion to String: Converting data to string.

Hailey Avie Annakin
Erna Dorey Battelle
Selle Bellison
Flori Chaunce Franzel
Orson Milka Iddins
Odetta Sue Kaspar
Roy Ketti Kopfen
Madel Bordie Mapplebeck
Debra Micheli
Leonerd Adda Mitchell Monaghan

D:\src\NameSorter\NameSorter\bin\Debug\netcoreapp2.1\win10-x64>
```

Drawing 3: Running [name-sorter]

3. Technical Design Manual

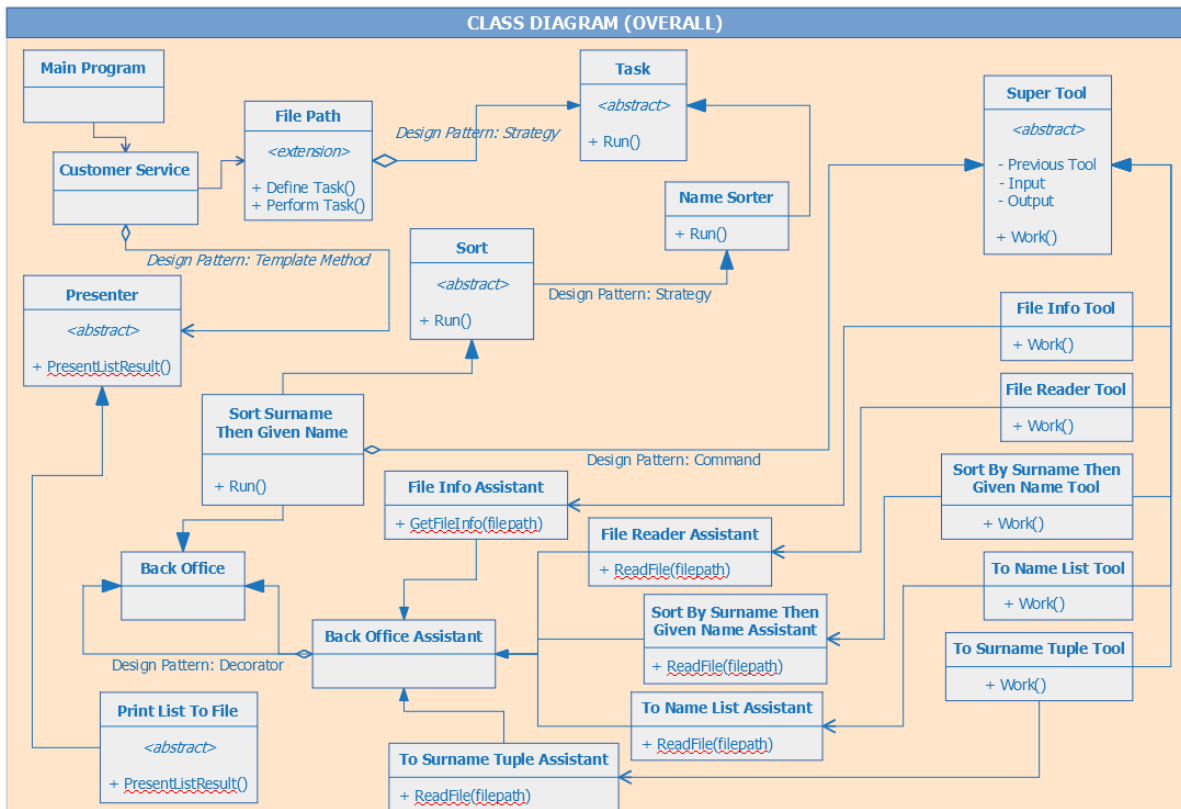
Program [name-sorter] follows design pattern concept. Following OOP concept, [name-sorter] follows an analogy where a regular customer goes to a bank to open a new account. The customer enters the bank, and meets the [*Customer Service*] officer that tackles all client requests. Knowing the type of request, [*Customer Service*] officer will give the appropriate form. The customer fills out the form and submits it to [*Customer Service*] officer for further processing.

There should be a smart person in the background that assists [*Customer Service*] officer that handles requests, including but not limited to approving or rejecting the process of generating an account for the customer. When a bank account number is created, bank should prepare a record book that customer can keep to track his banking transaction. Depending on whether or not the customer requires an ATM card, the [*Customer Service*] should ask the appropriate [*Back Office*] to generate an ATM card for the customer.

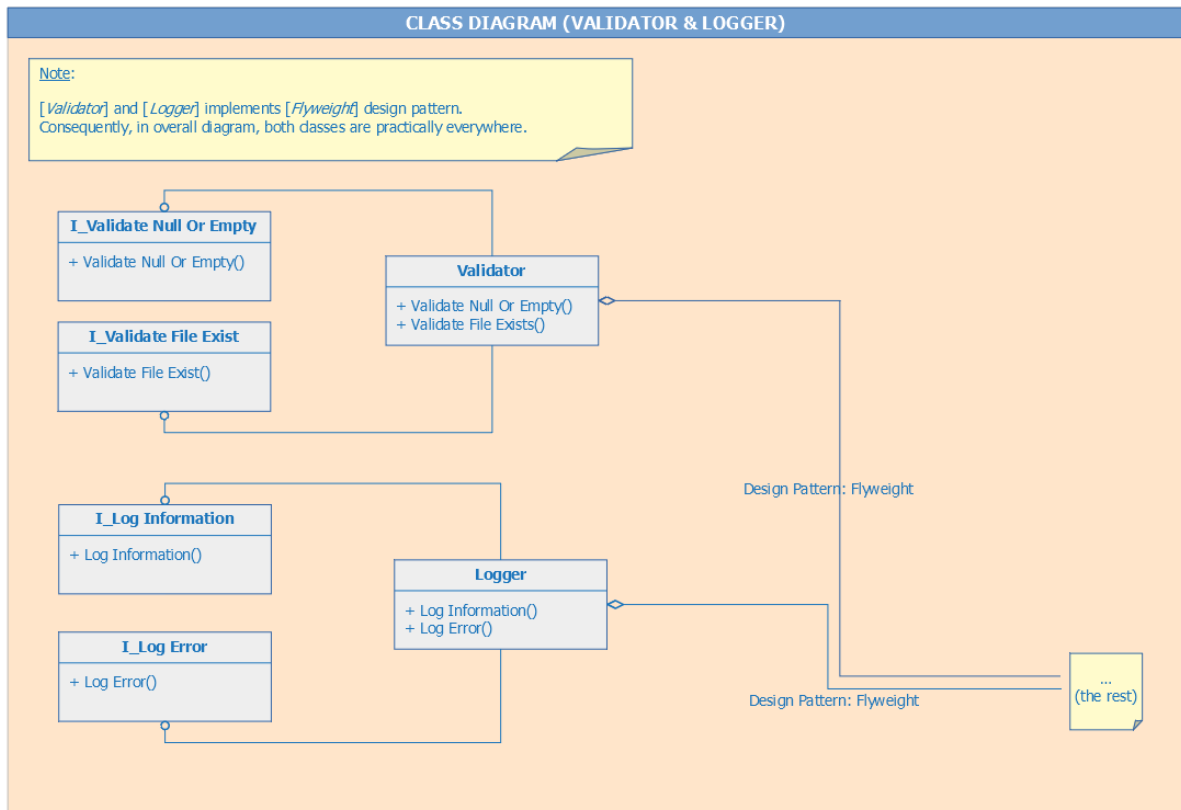
The same story applies for [name-sorter]. As soon as the [*Main Program*] receives a request to sort names on a certain file, it will call for the [*Customer Service*] officer to tackle the request. [*Customer Service*] officer shall choose among all available services, which one is the suitable service for this particular sort request.

Then, sort request goes to the [*Back Office*] that sorts data technically, and return the result back to the [*Customer Service*] officer. [*Customer Service*] will go to the *writer* staff, to prepare a notification letter for the customer. The letter contains the result of the sort process.

Roughly, the class diagram of [name-sorter] is as follows:

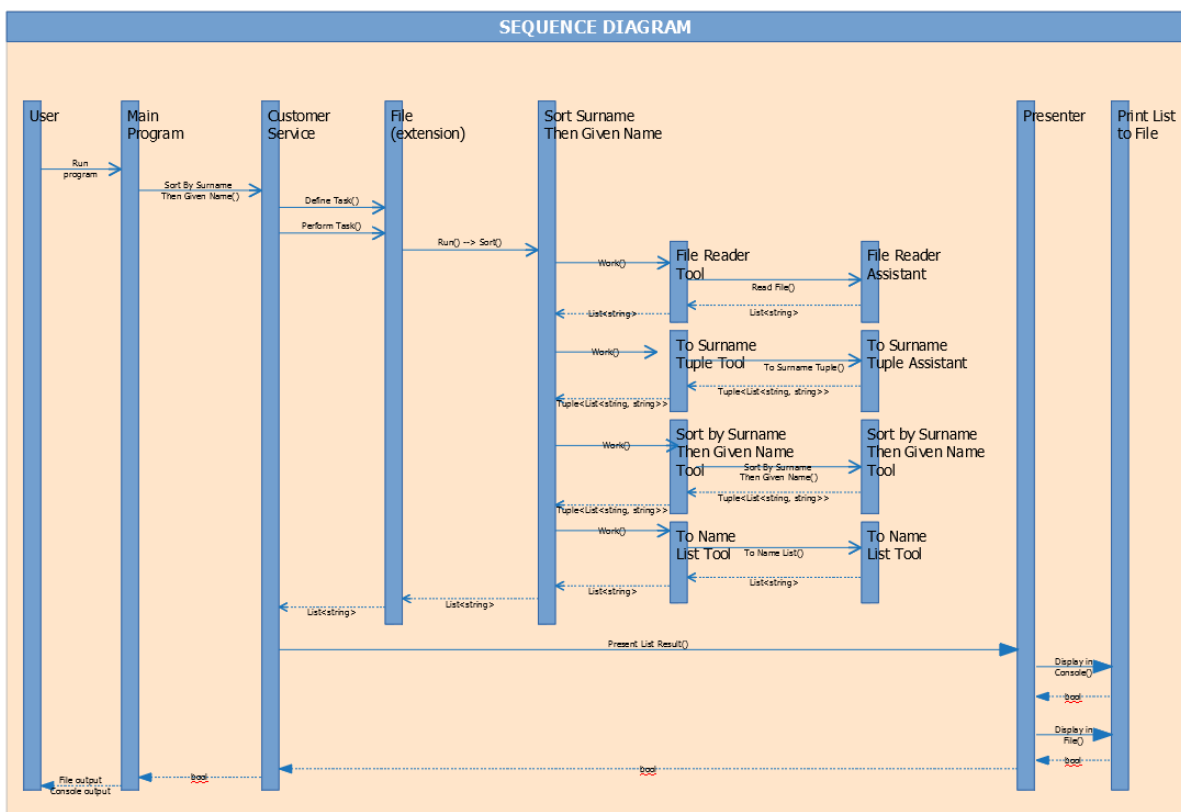


Drawing 4: [name-sorter] Class Diagram (Overall)



Drawing 5: [name-sorter] Class Diagram (Validator & Logger)

As proof of concept, the following is a Sequence Diagram that depicts the flow of the program:



Drawing 6: [name-sorter] Sequence Diagram