CSC 450

Team 2

US-02 Documentation

Version 1.0

10/07/2021

Files Included: us-02.py, \data-sets\trip-test.csv, \data-sets\responses.csv

Inputs:

* ‘r’, ‘R’, ‘c’, ‘C’
* .csv file
* Typed response less than or equal to 500 characters.

Functions:

* Main()
* csvOption()
* csvRead()
* bomValidation()
* responseOption()

Overview:

This user story was to allow a user and the development team to upload .csv file of responses or a single, typed in response to our application. For the time being, and for the sake of modularity and separation from other user stories, the code implementing US-02 is a command line tool. A prompt is given to the user asking if they would like to enter a path to a .csv file or type in a single response. The user has the option to type ‘r’ or ‘R’ for a single response and ‘c’ or ‘C’ for a path to a .csv file. The single response and .csv file are then validated to make sure they are in the correct format for our application. If they are not, an error message will pop up in the command line and prompt the user to re-enter a valid path or a valid response.

US-02 is entirely housed in the us-02.py file. There is also code in there pertaining to US-03 but is separated out into separate functions. us-02.py was tested with trip-tests.csv, a dataset downloaded from Kaggle.com and edited to our accepted formatting in Microsoft Excel.

Validation:

There are multiple layers of validation in us-02.py. The first step of validation happens in main(). This is when the user is prompted on what option they would like to do. There is a list of valid responses that has a length of 4 elements; ‘r’, ‘R’, ‘c’, ‘C’. the program checks the input to see if it is in that list. If it is not, the user is asked again for input until correct input is entered.

The next layer of validation depends on the option that the user takes. In the .csv option (in csvOption() ) , the next layer of validation is checking that the path to the .csv file the user provided is valid and exists. Here, we use the os module to call path.exists(path) to validate the path. If it is not a valid path, a prompt will appear telling the user that the input was an invalid path and will prompt the user for a path until a valid path is provided. Next, in csvRead(), the .csv file is iterated over and is scanned to see if each row is a single value. If it is not, an error message will appear telling the user that the .csv file is in an invalid format. The program will then prompt the user for another .csv until a valid .csv is provided.

In the case that the user chooses the single response option, the next layer of validation is in responseOption(). Here, it prompts the user for input. If the input is greater than 500 characters, a message will appear letting the user know that the input was too long and will prompt for input again until a valid length of input is entered.

There also a separate function called bomValidation(). This function scans the first 3 characters of the first row in a .csv file to check what encoding the file has. It then gets the encoding and passes that value to the csvRead() function to read the file appropriately. If there is no encoding characters in the file, the file is read as a UTF-8 file as default.