

SE 211: Software Specification and Design II

Software Design Document

CSVapp

Assignment #3

Cassidy Ashe

3/14/2021

Revision 1

SDD Revisions

Date	Description	Revision
3/14/2021	Created the document	0

Contents

1 Introduction	4
1.1 Purpose	4
1.2 Scope	4
1.3 Definitions, Acronyms, Abbreviations	4
2 Design Overview	4
2.1 Description of Problem	4
3 CSVlibrary Interface	5
3.1 CSVlibrary	5
3.2 CSVFile	6
3.3 CSVLibraryInterpreter	7
3.4 CSVlibraryValidator	8
3.5 CSVGetter	9
3.6 CSVSetter	10
6 References	12

1 Introduction

1.1 Purpose

The purpose of this document is to describe the implementation of the CSVapp as described in Assignment 1, the SRS document for this project. The purpose of this application is to allow users to access and modify CSV files using a graphical user interface.

1.2 Scope

This document describes the implementation details of the CSVapp. CSVapp is an application that allows for the access of CSV files with the intention of being modified. CSVapp serves the purpose of being the user interface for the CSVlibrary application.

1.3 Definitions, Acronyms, Abbreviations

CSVlibrary The name of the product being developed to be used in conjunction with CSVapp

CSV Comma-separated Value

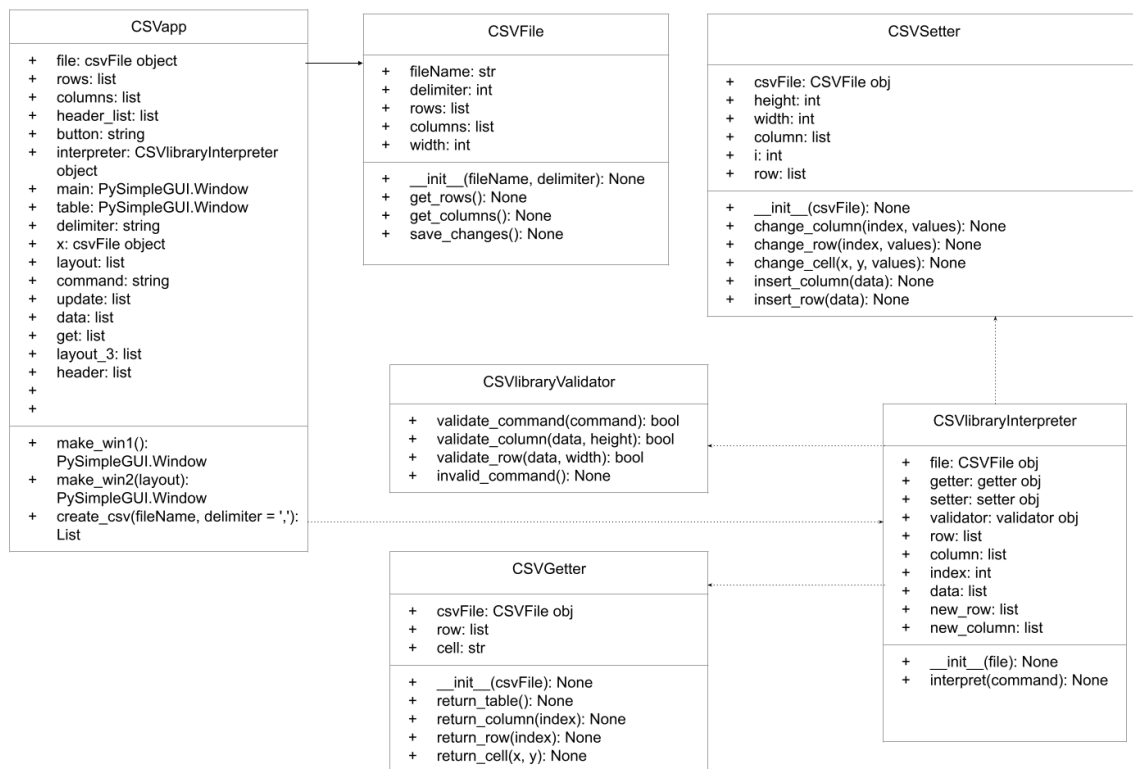
CSVapp The name of the product being developed to be used in conjunction with CSVlibrary

2 Design Overview

2.1 Description of Problem

The CSVFile class will serve the purpose of supporting the CSVapp, allowing users to have the ability to access their CSV files in a simple and efficient way. CSVapp contributes to both personal and professional uses, as it not only could be used by casual users, but has the potential to be used by businesses to effectively access and modify data.

3 CSVapp Interface



3.1 CSVapp

3.1.1 Attributes

Name	Type	Description
file	csvFile object	Used in the create_csv function to create the object
rows	list	A list of the rows created in create_csv
columns	list	A list of the columns created in create_csv
header_list	list	A list for the header row for table creation
button	String	The input from a yes/no popup
interpreter	CSVlibraryInterpreter object	Instance of interpreter object to perform functions on the CSV File
main	PySimpleGUI.Window	Main window for the CSVapp program

table	PySimpleGUI.Window	Table window and help section for the CSVapp program
delimiter	String	A delimiter used to separate the values of the CSV file
x	csvFile object	The object to be manipulated
layout	list	A list to define the layout of a window
command	string	The input from a user
update	list	Updates the CSV file to match the changes
data	list	The part of a command that contains the data for a modification
get	list	The data retrieved when the user requests a certain bit of data to be output
layout_3	list	A list to define the layout of a window
header	list	The header of a column

3.1.2 Methods

PySimpleGUI.Window make_win1()	
Input:	None
Output:	The main PySimpleGUI.Window that the user will interact with
Description:	Creates a window

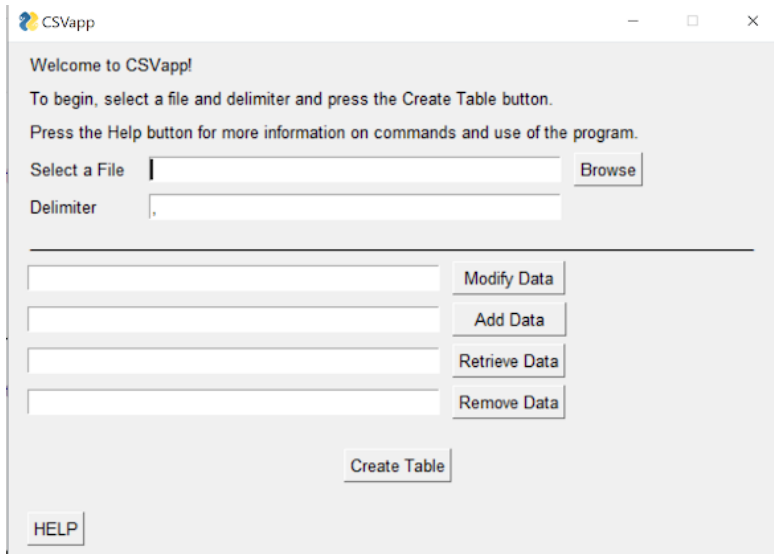
PySimpleGUI.Window make_win2(layout)	
Input:	Layout: a list of PySimpleGUI objects to be displayed in the second window
Output:	A PySimpleGUI.Window to display tables or the help section
Description:	Creates a window

List create_csv(fileName, delimiter = ',')	
Input:	fileName: the name of the file to make a CSVFile object from delimiter: the delimiter to use in creating the CSVFile object

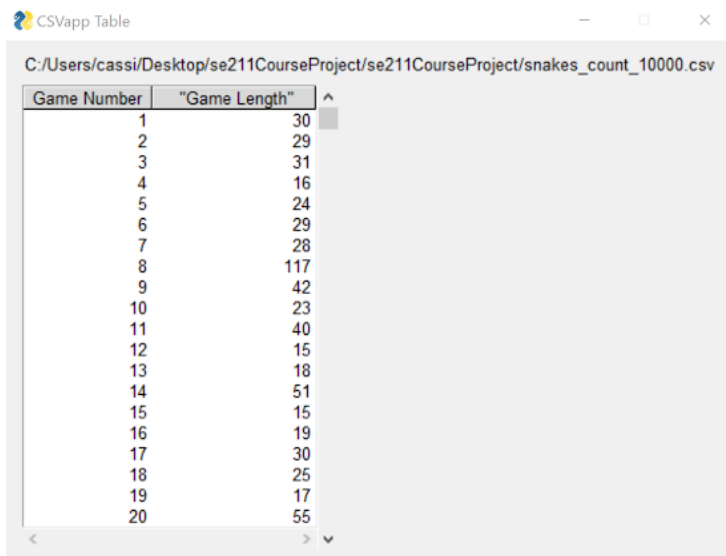
Output:	A list containing the rows list as well as the header_list
Description:	Creates a CSVFile object, and asks the user if the data contains a header row. If yes, the first row will be the header list and the rest will be in the rows list. Otherwise, the program creates a header_list for the user. Also creates an interpreter object to modify the CSV file.

3.1.3 Graphical User Interface

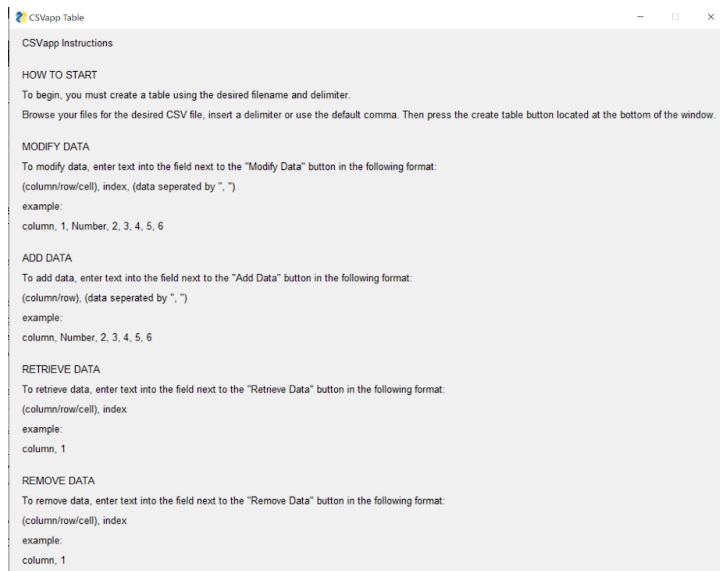
CSVApp main Window:



CSVApp Table Window with Example Data



CSVApp Help Window



3.2 CSVFile

3.2.1 Attributes

<i>Name</i>	<i>Type</i>	<i>Description</i>
fileName	str	User input that determines what file to open
rows	list	List of lists containing the data for each row in the file
columns	list	List of lists containing the data for each column in the file
width	int	Number of items in rows[0], describes how many columns there should be

3.2.2 Methods

None __init__(fileName:String, delimiter:String)	
Input:	The name of the file and delimiter to create a CSVFile object
Output:	A CSVFile object is created
Description:	The initializer for the CSVFile object.

None get_rows()

Input:	None
Output:	Adds the values of each row to the rows list.
Description:	Reads rows from CSV file and adds them to rows list.

None get_columns()	
Input:	None
Output:	Adds the values of each column to the columns list.
Description:	Takes column values from rows and adds them to list.

None save_changes	
Input:	None
Output:	None
Description:	Updates changes made to data by writing to the CSV file

3.3 CSVLibraryInterpreter

3.3.1 Attributes

<i>Name</i>	<i>Type</i>	<i>Description</i>
file	CSVFile object	The CSV file to be used
getter	csvGetter object	Instantiates getter object to return values
setter	csvSetter object	Instantiates setter object to set values
validator	validator object	Instantiates validator object to validate commands and inputs
index	int	Index value
data	list	List of values to change/add to a row or column
row	list	A row to modify

column	list	A column to modify
--------	------	--------------------

3.3.2 Methods

None __init__(file: CSVFile object)	
Input:	CSVFile object to be manipulated
Output:	None
Description:	Initializes an interpreter object.

None interpret(command: String)	
Input:	A command string from user input to be interpreted.
Output:	None
Description:	Interprets and executes commands given by the user.

3.4 CSVlibraryValidator

3.4.1 Methods

Bool validate_command(command: String)	
Input:	A command string from user input to be validated.
Output:	A boolean value indicating the validity of the command.
Description:	A validation method that determines if the user input a valid command.

Bool validate_column(data: list, height: int)	
Input:	A data list from user input and height of column to be validated.
Output:	A boolean value indicating the validity of the data and height.
Description:	A validation method that determines if the user input a valid data set.

Bool validate_row(data: list, width: int)	
Input:	A data list from user input and width of column to be validated.
Output:	A boolean value indicating the validity of the data and width.
Description:	A validation method that determines if the user input a valid data set.

None invalid_command()	
Input:	None
Output:	A print to the user indicating the input of an invalid command.
Description:	A method that serves the purpose of being an error message.

3.5 CSVGetter

3.5.1 Attributes

<i>Name</i>	<i>Type</i>	<i>Description</i>
csvFile	CSVFile object	The CSV file to be used
row	list	A row to return
cell	str	A column to return

3.5.2 Methods

None __init__(csvFile: CSVFile object)	
Input:	CSVFile object to retrieve data from.
Output:	A CSVFile object.
Description:	Initializes a CSVFile object.

None return_table()

Input:	None
Output:	Returns the CSV file data to the user.
Description:	A method to return the table created in the CSV file.

None return_column(index: int)	
Input:	Index of the column.
Output:	Returns the contents of the row to the user.
Description:	A method to return a specific column from the CSV file.

None return_row(index: int)	
Input:	Index of the row.
Output:	Returns the contents of the column to the user.
Description:	A method to return a specific row from the CSV file

None return_cell(x: int, y: int)	
Input:	Row (x) and column (y) of the cell that is meant to be modified.
Output:	Returns the contents of the cell to the user.
Description:	A method to return a specific cell from the CSV file

3.6 CSVSetter

3.6.1 Attributes

<i>Name</i>	<i>Type</i>	<i>Description</i>
csvFile	CSVFile object	The CSV file to be used
height	int	Length of first column, determines number of rows

width	int	Length of first row, determines number of columns
column	list	A list of data to be modified
i	int	Index value
row	list	A list of data to be modified

3.6.2 Methods

None __init__(csvFile: CSVFile object)	
Input:	A CSVFile object to be manipulated.
Output:	A setter object to make changes to the CSV file.
Description:	Initializes the setter object.

None change_column(index: int, values: list)	
Input:	The index of the column that the user would like to change as well as the values to change that column to.
Output:	None
Description:	Updates the CSV file with whatever valid modifications the user has made.

None change_row(index: int, values: list)	
Input:	The index of the row that the user would like to change as well as the values to change that row to.
Output:	None
Description:	Updates the CSV file with whatever valid modifications the user has made.

None change_cell(x: int, y: int, values: list)	
Input:	The cell (row, column) that the user would like to change as well as the value to change that cell to.
Output:	None

Description:	Updates the CSV file with whatever valid modifications the user has made.
--------------	---

None insert_column(data: list)	
Input:	A list of data in order to create a new column.
Output:	None
Description:	Updates the CSV file with whatever valid modifications the user has made.

None insert_row(data: list)	
Input:	A list of data in order to create a new row.
Output:	None
Description:	Updates the CSV file with whatever valid modifications the user has made.

6 References

[Sample Design Document, BBLearn](#)