SE 211: Specification and Design II

29 January 2020

Design Document for Assignment #1

Mohamed Mohamed

Table of Contents

Introduction

1.1 Purpose of Document:

The purpose of this documentation is specify the implementation of the CSV main application and the library. The main application serves as the user's method of interaction, with the CSV library's functionality.

1.2 Scope of Document:

This document discusses the design architecture and implementation details for the CSV application and library. As previously stated, the software consists of two articles of code: the main application and the CSV library. The CSV library is the source of functions that the main application utilizes. The main application serves as the main interface for the human user to utilize the library's functions.

1.3 Definitions and Abbreviations:

CSV library: Contains the functions that modify the CSV text file, and serves as the core functionality of the main application

Main CSV application: The main interface for the general user to modify, read, and write to CSV text files.

CSV text file: The entity that is being changed, serves as the input for main application and library.

Client: The human user.

System Overview:

2.1 Description of Software:

The main CSV application serves as a simple, lightweight method in manipulating CSV text files. Users will interact with a CLI, or command line interface, to access file contents and data. From the main menu, users will have the ability to write, replace, and even read values from the command line interface.

2.2 Technologies Used

This program (the CSV library and main application) has rather basic requirements to ensure functionality. The client must use a machine that can run Python 3.7 as an input device. The recommended operating system to run is Microsoft Windows 10, however the library and application should be compatible with any machine running Python 3.7. The development environment for both articles of code were created in Thonny, a Python integrated development environment.

System Architecture:

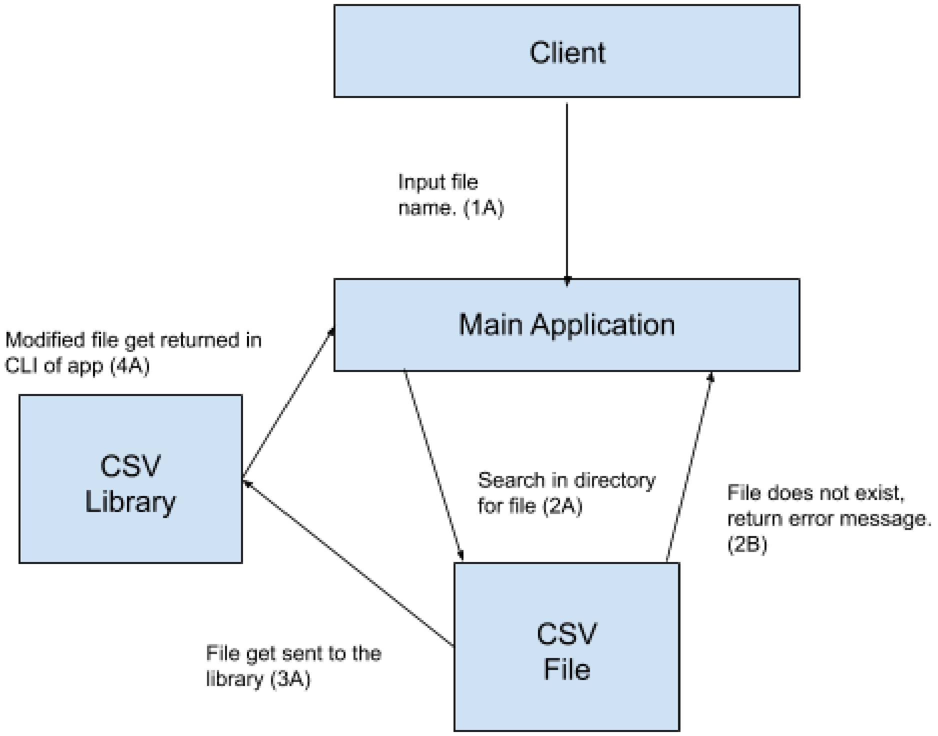
3.1 Architectural Design Components

Client - The client's role is the one that declares an input into the interface of the program, in this case, the main application.

Main Application - Transmits the Client's input into the machine and communicates with the CSV library under certain circumstances

CSV library - This part of the system works to perform designated by the client. This part of the system

The CSV Text File - The file that is being modified through all of the other design components.


Client
Input file name. (1A)
Main Application
Modified file get returned in CLI of app (4A)
CSV Library
Search in directory for file (2A)
File does not exist, return error message. (2B)
CSV File
File get sent to the library (3A)

(The architectural design of the entire CSV system, with an overview of the step that the system goes through after the client selects a command.)

3.2 Design Rationale

Why support text files?

Python possesses native support for text files, and as such makes implementation rather straight-forward. CSV files can also be supported through text files, specifically those with the “txt” file extension. Native support will lead to quickly execution time among all supported devices, and will reduce the amount of requirements to run the system of programs.

Component Design:

4.1 Overview

This section will go into further detail through the two articles of code introduced in the Scope of Document section of the Design Document.

4.2 CSV library

As previously stated, the CSV library is the source of functionality within the system. It contains the majority of the functions needed to modify the CSV files that the user requires. It also contains various methods that culminate the library’s various functionalities, like read() and replace ().

Name	Type	Description
concsv()	function	Called when user wants to change separators in file.
csvprint()	function	Called when user wants to read a file.
csvreplace()	function	Called when a user wants to replace an existing value in a file.
csvappend()	function	Called when a user wants to add more data into a CSV file.

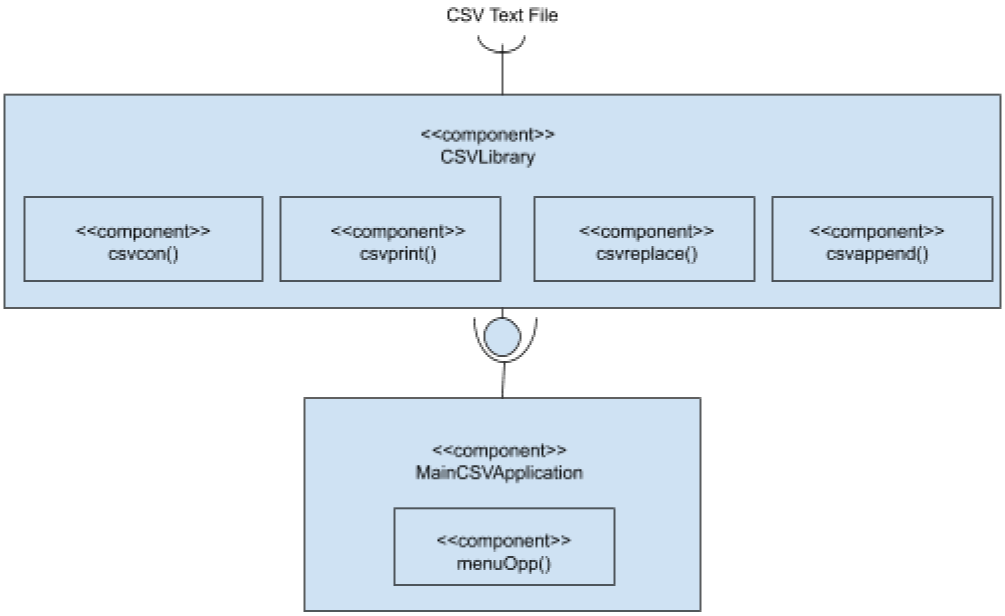
Name	Type	Description
.read()	method	Reads the inputted file.
.write()	method	Writes the content within the parameter into the file.
.replace()	method	Takes in two parameters, one value that is replaced, the other replacing the original

		value.
.close()	method	Closes the file.

4.3 Main CSV Application

The main CSV application utilizes an abstract architecture structure. The CSV application consists of a single function that is recursively called, with various if-elif statements that determine what set of commands the program will execute. The main application is responsible for being the interface the user uses to access the functionalities of the library. As such it refers to the library's function.

Name	Type	Description
concsv()	function	Called when user wants to change separators in file.
csvprint()	function	Called when user wants to read a file.
csvreplace()	function	Called when a user wants to replace an existing value in a file.
csvappend()	function	Called when a user wants to add more data into a CSV file.
menuOpp()	function	The internal structure that holds all of the contents of the main application.



(The component diagram that details that relationship between the component of the CSV library and main application.)