CSV Library Software Requirements

## Introduction

The CSV Library is an easy to use Java library used for parsing and editing CSV files. The CSV Library provides standard CSV formats for reading and for the ability of the user to define their own.

## Overall Description

This document outlines the functional and non-functional requirements of the CSV Library. In addition, this document will outline the format required of the CSV files, the interface provided by the library, and error handling.

| **Actor** | **Description** |
| --- | --- |
| Developer | Will use the library to read and write files that use the CSV file format |

## Design and Implementation

The CSV Library is implemented in Java and is compatible with Java versions 11+. The CSV Library can be distributed as a .jar or as a .zip file with the separate .java files of the library.

## CSV File Format

The CSV Library allows the developer a certain degree of control over how the CSV files are parsed. The CSVFormat class will allow the developer to choose multiple options such as choosing to use the first line as a record (i.e. no header in the CSV file), choosing to trim whitespace, choosing the delimiter, etc. The full list of available options will be outlined in the library interface section. Outside of the provided options the library will read from input streams until EOF and separate the stream by new line delimiters. Each separate line will be processed into a CSVRecord based off of the CSVFormat options.

## Library Interface

### CSVFormat

The CSVFormat class will have two constructors. The first constructor will take no arguments and initializes the instance data of the class to their defaults. The other constructor will take four arguments for each field and will assign the provided value to the corresponding field. There will be a getter and setter for each of the four fields. There will be a copy method that returns a new CSVFormat instance with the same field values as the object that invoked the method.

### CSVReader

The CSVReader class will have five constructors. All five constructors will have two arguments: the first is an object used to read/open the csv file and the second is a CSVFormat object for formatting input. The developer will have the choice to either provide a file/path to a file to be opened or to provide a direct InputStream object to the csv file. The CSVReader will have getters for the List objects containing the headers and CSVRecord objects. The CSVReader will have a close method that closes the InputStream used for reading the csv file.

### CSVPrinter

The CSVPrinter class will have five constructors. All five constructors will have three arguments: an object to open/write the csv file, a CSVFormat object for dictating printing, and an arbitrary amount of String objects representing the header of the csv file to be printed. The developer will have the choice to provide a file/path to the file or to provide an OutputStream to the file. The CSVPrinter class will have a method for printing an entire record at once and printing individual entries of a record. There will be a method for ending the current record and continuing to the next. The CSVPrinter will have a close method that closes the OutputStream used for printing the csv file.

### CSVRecord

The CSVRecord class will have no public constructor. The developer can obtain instances of CSVRecord objects from the CSVPrinter class. The CSVRecord will have two methods to get values within a record: one that takes an index to the value and one that takes the header name of the value’s column. Similarly, the class will also have two methods to check if a value is set within a record. These methods will also use either an index or header name to determine if a value was set. The CSVRecord will have a method for determining the size of the record. The class will have a method to determine if the record size is consistent with the header size. The class will have a method to get a List object with the values of the CSVRecord. The CSVRecord will override the toString method to allow for easily displaying the CSVRecord.

## Error Recovery

It is expected of the developer to be prepared for IOExceptions. The constructors of the CSVReader and CSVPrinter classes throw IOExceptions in the event there’s an error creating the streams for reading and writing the csv file. The CSVRecord class' get methods will return null to signal to the developer that an error occurred in reading the value from the record.

## Functional Requirements

### CSVFormat

| **No.** | **Requirement** |
| --- | --- |
| FMT-01 | The library shall allow the developer to instantiate a CSVFormat object with default values |
| FMT-02 | The library shall allow the developer to instantiate a CSVFormat object of a custom format defined by the developer |
| FMT-03 | The library shall allow the developer to mutate a CSVFormat object’s fields |
| FMT-04 | The library shall allow the developer to copy a CSVFormat object to make a new instance with identical values |

### CSVReader

| **No.** | **Requirement** |
| --- | --- |
| READ-01 | The library shall allow the developer to instantiate a CSVReader object from a InputStream object and a CSVFormat object |
| READ-02 | The library shall allow the developer to instantiate a CSVReader object with a File object instead of a InputStream object |
| READ-03 | The library shall allow the developer to instantiate a CSVReader object with a Path object instead of a InputStream object |
| READ-04 | The library shall allow the developer to instantiate a CSVReader object with a String object representing a path instead of a InputStream object |
| READ-05 | The library shall allow the developer to obtain an unmodifiable List object containing the CSVRecord objects read from the InputStream |
| READ-06 | The library shall allow the developer to obtain a List object containing the header values store as String objects |
| READ-07 | The library shall throw an IOException if an error occurs creating or reading the InputStream |
| READ-08 | The CSVReader class shall implement the AutoCloseable interface so that it can be used within a try-with-resources statement |
| READ-09 | The library shall allow the developer to obtain an unmodifiable List object containing the headers |

### CSVPrinter

| **No.** | **Requirement** |
| --- | --- |
| PRNT-01 | The library shall allow the developer to instantiate a CSVPrinter object from a OutputStream object, a CSVFormat object, and an arbitrary amount of String objects to represent the header |
| PRNT-02 | The library shall allow the developer to instantiate a CSVPrinter object with a File object instead of a OutputStream object |
| PRNT-03 | The library shall allow the developer to instantiate a CSVPrinter object with a Path object instead of a OutputStream object |
| PRNT-04 | The library shall allow the developer to instantiate a CSVPrinter object with a String object representing a path instead of a OutputStream object |
| PRNT-05 | The library shall print the headers, if any, to the OutputStream within the CSVPrinter constructor |
| PRNT-06 | The library shall allow the developer to print an arbitrary amount of String objects to the OutputStream as a record |
| PRNT-07 | The library shall allow the developer to print a singular value into current record in the OutputStream |
| PRNT-08 | The library shall allow the developer to print the end of line String to the OutputStream |
| PRNT-09 | The library shall throw an IOException if an error occurs printing to the OutputStream |
| PRNT-10 | The CSVPrinter class shall implement the AutoCloseable interface so that it can be used within a try-with-resources statement |

### CSVRecord

| **No.** | **Requirement** |
| --- | --- |
| REC-01 | The library shall allow the developer to read all values within the CSVRecord object |
| REC-02 | The library shall return null for any value in the record that is unset |
| REC-03 | The library shall allow the developer to read a value by its index within the CSVRecord object |
| REC-04 | The library shall allow the developer to read a value by its header name within the CSVRecord object |
| REC-05 | The library shall allow the developer to check if a value at a given index is has a set value |
| REC-06 | The library shall allow the developer to check if a value with a given header name has a set value |
| REC-07 | The library shall allow the developer to obtain the number of values within the CSVRecord object |
| REC-08 | The library shall allow the developer to check if the size of the CSVRecord is the same as the size of the header |
| REC-09 | The library shall allow the developer to obtain an unmodifiable List object containing the CSVRecord values |
| REC-10 | The CSVRecord class shall override the toString method |

## Non-Functional Requirements

### Performance

| **No.** | **Requirement** |
| --- | --- |
| PR-01 | The CSVReader class shall use a BufferedInputStream object to improve library performance |
| PR-02 | The CSVPrinter class shall use a BufferedOutputStream object to improve library performance |

### Security

| **No.** | **Requirement** |
| --- | --- |
| SR-01 | The CSVReader class shall make an internal copy of the provided CSVFormat object to prevent modification of format while reading the CSV file |
| SR-02 | The CSVPrinter class shall make an internal copy of the provided CSVFormat object to prevent modification of format while writing the CSV file |