Software Design Document

CSVapp

Cole Hoener

February 28, 2021

Contents

[1 Introduction 1](#_Toc66647577)

[1.1 Purpose 1](#_Toc66647578)

[1.2 Scope 1](#_Toc66647579)

[2 Design Overview 2](#_Toc66647580)

[2.1 Description of Problem 2](#_Toc66647581)

[2.2 Technologies Used 2](#_Toc66647582)

[2.3 System Architecture 2](#_Toc66647583)

[2.4 System Operation 3](#_Toc66647584)

[3 CSVapp Class Diagram 4](#_Toc66647585)

[3.1 Application 5](#_Toc66647586)

[3.1.1 Attributes 5](#_Toc66647587)

[3.1.2 Methods 5](#_Toc66647588)

[3.2 StartPage 5](#_Toc66647589)

[3.2.1 Attributes 5](#_Toc66647590)

[3.2.2 Methods 5](#_Toc66647591)

[3.3 MainPage 6](#_Toc66647592)

[3.3.1 Attributes 6](#_Toc66647593)

[3.3.2 Methods 6](#_Toc66647594)

# Introduction

## Purpose

The purpose of this document is to describe the implementation of the CSVapp described in the SRS documents from Assignment #1. The CSVapp is designed to interact, alter, and read CSV files through a user interface.

## Scope

This document describes the implementation details of the CSVapp. The software will consist of 2 major functions, to read CSV files and to alter CSV files.

# Design Overview

## Description of Problem

CSV files are not easy to manually alter, and view based on their text-based format. To view and alter CSV files efficiently and accurately requires software that is able to perform the repetitive tasks to do so.

## Technologies Used

The CSVapp will be a GUI based system with predefined actions that the user can perform. All the predefined actions will be outlined in the README file.

The intended platform is any machine capable of running python3.

## System Architecture

Figure 1 depicts the high-level system architecture. The system will be constructed from multiple distinct components:

* Command line Interface — Handles user directed interactions from the GUI to read and alter CSV files.
* CSVapp - The GUI the user interacts with.
* CSVFileReader – Can read CSV files
* CSVEditor – Can edit and interact with CSV files

Diagram

Description automatically generated

Figure 1

## System Operation

Figure 2 is the typical sequence of events that occur during a CSVapp session.

Diagram

Description automatically generatedFigure 2

# CSVapp Class Diagram

Figure 3 depicts the UML model for the CSVapp. The CSVapp deals with all user interactions and has multiple screens (/classes) with different methods and interactions. Users only interact with one class at a time.

Diagram

Description automatically generated

Figure 3: CSVapp Class Diagram

\*Note: There are many other attributes within each class, but they are not required for initalizion. They are only attributes in case of need of use throughout the class\*

## Application

### Attributes

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| master | Tk.Frame | The window for the application |
| frame | Tk.frame | The frame for the application |

### Methods

|  |  |
| --- | --- |
| Switch\_frame(frame\_class): void |  |
| Input: | A frame for the application to switch to |
| Output: |  |
| Description: | Changes the frame the application displays |

## StartPage

### Attributes

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| master | String | The name of the file |
| File | Tk.Frame | The window for the application |

### Methods

|  |  |
| --- | --- |
| createWidgets() |  |
| Input: | None |
| Output: | Void |
| Description: | Creates widgets and puts them in the frame |

|  |  |
| --- | --- |
| openFile() |  |
| Input: | None |
| Output: | Void |
| Description: | Opens the selected file |

## MainPage

### Attributes

|  |  |  |
| --- | --- | --- |
| Name | Type | Description |
| master | Tk.frame | The window for the application |
| newFrame | Tk.frame | The grid frame for the CSV view |

### Methods

|  |  |
| --- | --- |
| createWidgets() |  |
| Input: | None |
| Output: | Void |
| Description: | Creates widgets and puts them in the frame |

|  |  |
| --- | --- |
| updateTable() |  |
| Input: | Void |
| Output: | void |
| Description: | Updates the table widgets and displays them in the frame |

|  |  |
| --- | --- |
| changeSepecify() |  |
| Input: | None |
| Output: | Void |
| Description: | Sets the new separator to read from the file and updates the table |

|  |  |
| --- | --- |
| getCell() |  |
| Input: | None |
| Output: | Void |
| Description: | Gets the user entered cell and displays the data from that cell |

|  |  |
| --- | --- |
| changeCell() |  |
| Input: | None |
| Output: | Void |
| Description | Changes the data in the user specified cell |

|  |  |
| --- | --- |
| exit() |  |
| Input: | None |
| Output: | Void |
| Description | Changes the frame to StartPage |