Project Design Documentation

Project Design Documentation

Version 1

Table of Contents

[Purpose 3](#_Toc74047492)

[Class Diagram 3](#_Toc74047493)

[For Future Considerations (Task 2) 3](#_Toc74047494)

[Reusable Components (Task 3) 4](#_Toc74047495)

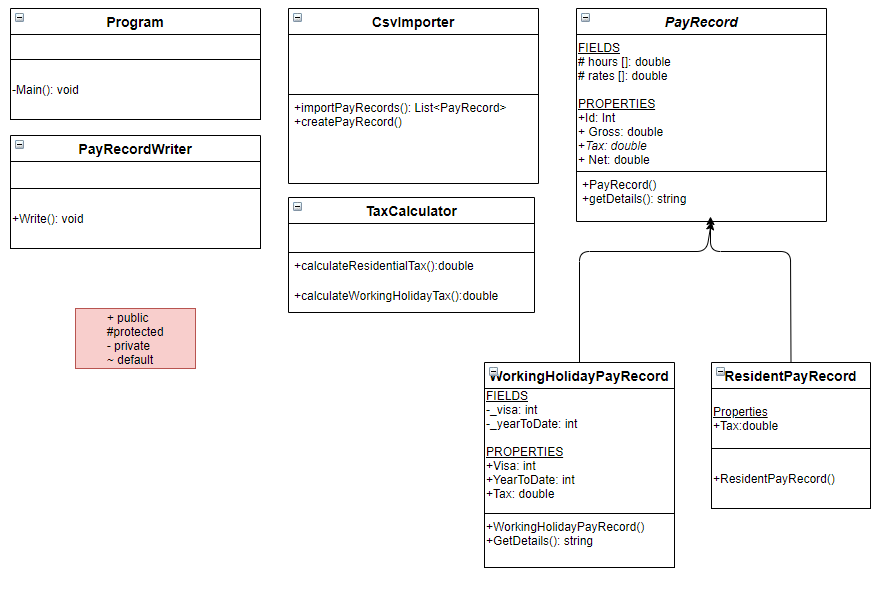
**[update Table of Contents when finished]**

# Purpose

The purpose of this document is to illustrate the design elements and initial testings that will be used to construct the main application.

# Class Diagram

Below is a UML class diagram that is based off the classes tables provided from the analyst. We will be using this as a reference guide for when start to build our application and it’s classes.



# For Future Considerations (Task 2)

As for the main purpose, this application will be run on desktop computers, however as instructed by the client we are to consider what the requirements be if the application was to be converted into an app.

Below are some systems, operating systems and devices that have potential to fit this requirement.

|  |  |
| --- | --- |
| Systems | Devices |
| xcode | MAC |
| Windows Operating System | PC/Laptops |
| Bizness Apps | Android, iPhone, Web-based |
| AppSheet | Windows, Mac, Linux |
| Appery.io | Windows, Mac, iPhone, Android & Web-based |

# Reusable Components (Task 3)

### Functionality Requirements and Reusable Components

As we have analysed the project, besides the main functionality requirements of the tax calculator, we have identified that the functionality for reading and writing to and from a CSV file can be implemented with ‘reusable components from either

The “System” namespace such as:

* System.IO
  + <https://docs.microsoft.com/en-us/dotnet/api/system.io?view=net-5.0>
  + System IO belongs to the system namespace which is a part of Microsoft .NET Framework therefore it is completely free to use.
  + <https://dotnet.microsoft.com/platform/free>

Or from an external library such as:

* CsvHelper
  + <https://joshclose.github.io/CsvHelper/>
  + <https://www.nuget.org/packages/CsvHelper/>
  + Package Download
    - <https://www.nuget.org/api/v2/package/CsvHelper/27.1.0>
  + Alternatively CsvHelper can be downloaded to your IDE (example VS Studio) through package manager.

Completely free for commercial use.

Dual License

* [Microsoft Public License (MS-PL)](https://opensource.org/licenses/MS-PL)
* [Apache License, Version 2.0](https://opensource.org/licenses/Apache-2.0)

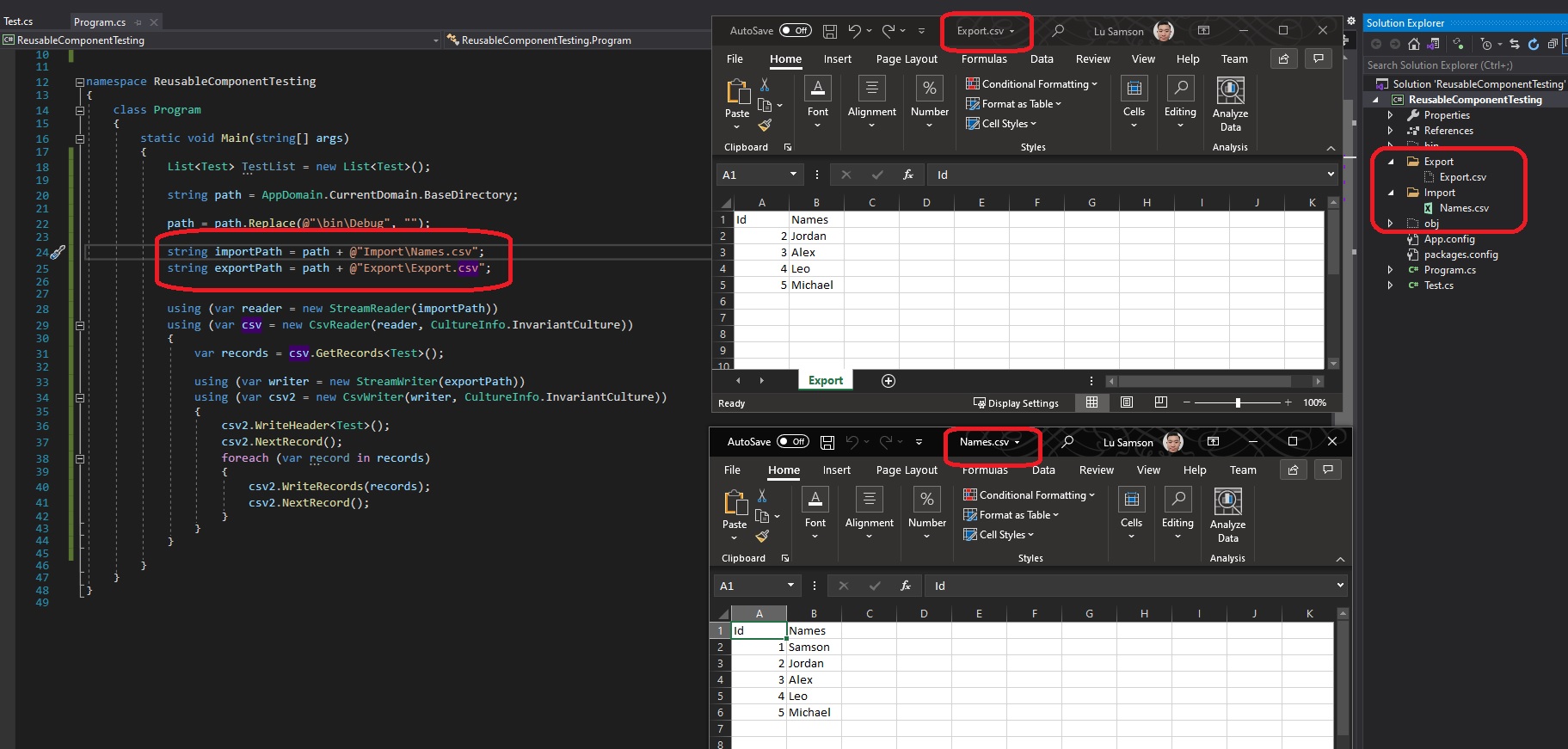
The reusable components are great way to quickly implement well known and supported code into your program.

Depending on which reusable component you are implementing, you should consider whether any future updates to these components would break your program. You should refer to their release notes for their changes/fixes or known bugs. Utilising external reusable components leaves your program with a possible flaw where a dependency relies on an external source. You are also required to learn how to utilise the components too.

However with utilising CsvHelper, it has a strong list of community contributors and provides well written documentation and support. It provides the right functionality for our requirements.

### Trialling the CsvHelper Reusable Component

The below screenshot illustrates the use of a reusable component (CsvHelper) to read and import data from a csv and export and write back the data into another csv.



# Code Metrics

Definition: Code metrics is a set of software measures that provide developers better insight into the code they are developing. By taking advantage of code metrics, developers can understand which types and/or methods should be reworked or more thoroughly tested.

Below indicators show the code is highly maintainable.

