Software Testing Report

<Project Name>

Student Names: **Zhaocheng Dong, Zihao Cheng, Yidan Zhang**

Table of Contents

[1.0 Unit Tests 3](#_Toc49779837)

[2.0 Coverage Report 4](#_Toc49779838)

[3.0 Requirements Acceptance Testing 5](#_Toc49779839)

# Unit Tests

Delete the RED text and replace with your own

(In this table you fill out details about what unit tests you have done using the unittest module)

| **No** | **Test Case** | **Expected Results** | **Actual Results** |
| --- | --- | --- | --- |
| **1.0** | **CSV file reading and drawing** |  |  |
| 1.1 | First attempt at limiting output content | Depending on the constraints, the output should be what we need. | Problems with the code |
| 1.2 | Second re-code to limit output | Required content | Required content |
| 2.0 | Histogram Functions |  |  |
| 2.1 | Empty input dictionary | Display error message and exit | Display error message and exit |

# Coverage Report

A description of the coverage of your unit tests, including how you evaluated coverage (function, statement, branch, condition)

The main difficulty with the encoding process for the third and fourth problems is to read the contents of the CSV file and obtain the required content under conditions similar to keyword restrictions. To a certain extent, the initial refinement of the reading of the CSV file and the mapping of the data after acquisition led to difficulties in the subsequent coding process, where the initial experience of using the same coding method was taken for granted, and the keyword restriction was not implemented in a way that accurately identified or read the content sought. During the re-coding process, I reversed my original assumptions and started over using ‘mask’ to help me better distinguish keywords to achieve the desired effect.

# Requirements Acceptance Testing

(You will need to fill out the column on the left with the requirements listed in software design documents and the columns on the right with the results of your own testing)

| **Software  Requirement No** | **Test** | **Implemented (Full /Partial/ None)** | **Test Results (Pass/ Fail)** | **Comments (for partial implementation or failed test results)** |
| --- | --- | --- | --- | --- |
| 1 | Try to enter a CSV file and read the contents to draw. | Full | Pass |  |
| 2 | Distinguish keywords and output | None | Fail | Because adding something like a keyword restriction on top of the original drawing alone failed, other ways are being sought to achieve the purpose. |
| 3 | Re-coding(Distinguish keywords and output) | Full | Pass |  |
| 4 | Display a message if an argument is a directory instead of a file |  |  |  |
| 5 | File name can be a simple file name or include the full path of the file with one or more levels |  |  |  |
| 6 | file names must start with an alphabetical character |  |  |  |
| 7 | Valid file name extensions must be 3 or 4 alphabetical characters preceded by a dot) |  |  |  |
| 8 | Directory/level names must start with an alphabetical character to be considered valid |  |  |  |
| 9 | The program should be able to accept as many levels for each file name as the user wants to input. This is limited only by the number of levels allowed in Windows (approximately 120) |  |  |  |