# Test plan

## Date Unit Test

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Description | Expected Output | Passed |
| 1 | Check constructor correctly initializes the data, and all getters can retrieve the data | Default Constructor  Day: -1  Month: -1  Year: -1  Constructor  Day: 2  Month: 2  Year: 2000 | Pass |
| 2 | Check that setters are working correctly | Test Setters  Day: 24  Month: 10  Year: 1995 | Pass |

A screen shot of a computer

Description automatically generated

## Time Unit Test

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Description | Expected Output | Passed |
| 1 | Check constructor correctly initializes the data, and all getters can retrieve the data | Default Constructor  Hour: -1  Minute: -1  Constructor  Hour: 10  Minute: 30 | Pass |
| 2 | Check that setters are working correctly | Test Setters  Hour: 9  Minute: 56 | Pass |

A screen shot of a computer

Description automatically generated

## Weather Unit Test

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Description | Expected Output | Passed |
| 1 | Check that constructor correctly initialize the data and all getters can retrieve the data | Check constructor.  Date: -1/-1/-1  Time: 0-1:0-1  Wind Speed: -1  Solar Radiation-1  Air Temperature: -1 | Pass |
| 2 | Check that setters are working correctly | Test Setters  Date: 16/3/2016  Time: 09:15  Wind Speed: 11.1  Solar Radiation23.15  Air Temperature: 78.25 | Pass |

A screenshot of a computer

Description automatically generated

## Vector Unit Test

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Description | Expected Output | Passed |
| 1 | Check that Constructor can correctly initialize an empty Vector and Operator [] will throw out of range exception | Check Constructor  Try retrieve: Unable to retrieve any elements | Pass |
| 2 | Check that the copy constructor can correctly deep copy another Vector | Test Copy Constructor  Retrieving test Vector: test pointer: <pointer>  Retrieving test Vector: test2pointer: <pointer>  Retrieving test Vector: test3pointer: <pointer>  Retrieving copy Vector: test pointer: <pointer>  Retrieving copy Vector: test2pointer: <pointer>  Retrieving copy Vector: test3pointer: <pointer>  \*ensure pointer values are all different | Pass |
| 3 | Check that Size function can correctly return the size of the vector | Check Size of test vector: 3 | Pass |
| 4 | Check that Capacity function can correctly return the capacity of the vector | Check Capacity of test Vector: 4 | Pass |
| 5 | Check that the destructor or can successfully delete the Vector | Tests destructor or  Size of copy Vector before destructing: 3  Size of copy Vector after destructing: 0 | Pass |
| 6 | Check that Operator= can correctly deep copy another Vector | Test Operator=  Retrieve test Vector pointer: <pointer>  Retrieving test Vector [0]: test, pointer: <pointer>  Retrieving test Vector [1]: test2, pointer: <pointer>  Retrieving test Vector [2]: test3, pointer: <pointer>  Retrieve copy Vector pointer: <pointer>  Retrieving copy Vector [0]: test, pointer: <pointer>  Retrieving copy Vector [1]: test2, pointer: <pointer>  Retrieving copy Vector [2]: test3, pointer: <pointer>  \*Ensure all pointers are different | Pass |
| 7 | Check that Resizes function can correctly increase the capacity of the vector by multiple of 2 if it isn’t enough | Check Resize(int) function.  current size and capacity of vector, size (3) capacity (4)  size and capacity of vector after resizing to 5, size (5) capacity (6)  size and capacity of vector after resizing to 3, size (3) capacity (6) |  |
| 8 | Check that Erase function can correctly remove an element in a vector while shifting all elements after the index to the left | Check Erase(int)  Retrieving Vector index 0: test  Retrieving Vector index 1: test2  Retrieving Vector index 2: test3  After erasing element at index 1  Retrieving Vector index 0: test  Retrieving Vector index 1: test3 | Pass |
| 9 | Check that Inserts function can correctly insert an element in a vector while shifting all elements from the index to the right | Check Insert (int, <T>)  Retrieving copy Vector index 0: test  Retrieving copy Vector index 1: test3  Inserted new element at index 1  Retrieving Vector index 0: test  Retrieving Vector index 1: Test  Retrieving Vector index 2: test3 | Pass |
| 10 | Check that clear function can correctly remove all elements in a vector | Check clear ()  Size of Vector before clearing: 3  Size of Vector after clearing 0 | Pass |

A computer screen with white text

Description automatically generated

## Logic Unit Test

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Description | Expected Output | Passed |
| 1 | Check that RemoveLeadingZeros function can correctly remove the leading zeros when given a string | Test RemoveLeadingZeros (string)  Test RemoveLeadingZeros (08): 8  Test RemoveLeadingZeros (08.9): 8.9 | Pass |
| 2 | Check that DisplayMenu function can successfully display the menu | Test DisplayMenu ()  1. The average wind speed and average ambient air temperature for a specified month and year.  2. Average wind speed and average ambient air temperature for each month of a specified year.  3. Total solar radiation in kWh/m2 for each month of a specified year.  4. Write average wind speed (km/h), average ambient air temperature and total solar radiation in kWh/m2 for each month of a specified year to CSV.  5. Exit the program. | Pass |
| 3 | Check that IntMonthToString function can correctly return the month in string when provided with a month in integer | Test IntMonthToString (unsigned int)  Test IntMonthToString (8): August  Test IntMonthToString (13):  Error converting int month to string.  ERROR | Pass |
| 4 | Check that ConvertWindMStoKMH function can correctly convert windspeed provided in m/s to km/h | Test ConvertWindMStoKMH (float)  Test ConvertWindMStoKMH (10): 36  Test ConvertWindMStoKMH (10.5): 37.8 | Pass |
| 5 | Check that ConvertSolarRadiationWMtoKWH function can correctly convert solar radiation provided in W/m2 to kWh/m2 | Test ConvertSolarRadiationWMtoKWH (float)  Test ConvertSolarRadiationWMtoKWH (120): 0.02  Test ConvertSolarRadiationWMtoKWH (120.42): 0.02007 | Pass |
| 6 | Check that LoadDataFileCSV function can correctly load csv data file | Test LoadDataFileCSV function  31/3/2016,09:00,6,512,20.74  31/3/2016,09:10,5,565,20.97  31/3/2016,09:20,5,574,20.92  31/3/2016,09:30,5,623,21.63  31/3/2016,09:40,6,617,22.39 | Pass |
| 7 | Check that AverageWindSpeedMonth function can correctly find and calculate the average windspeed for a month | Test AverageWindSpeedMonth (vector, int, int)  Test AverageWindSpeedMonth (TestLog,3,2016): 5.4  Test AverageWindSpeedMonth (TestLog,1,2020): -1 | Pass |
| 8 | Check that AverageAirTemperatureMonth function can correctly find and calculate the average ambient air temperature for a month | Test AverageAirTemperatureMonth (vector, int, int)  Test AverageAirTemperatureMonth (TestLog,3,2016): 21.33  Test AverageAirTemperatureMonth (TestLog,1,2020): -1 | Pass |
| 9 | Check that SumSolarRadiationMonth function can correctly find and calculate the total solar radiation for a month | Test SumSolarRadiationMonth (vector, int, int)  Test SumSolarRadiationMonth (TestLog,3,2016): 2891  Test SumSolarRadiationMonth (TestLog,1,2020): -1 | Pass |
| 10 | Check that StandardDeviationWindSpeedMonth function can correctly find and calculate standard deviation of wind speed for a month | Test StandardDeviationWindSpeedMonth (vector, int, int)  Test StandardDeviationWindSpeedMonth (TestLog,3,2016): 0.55  Test StandardDeviationWindSpeedMonth (TestLog,1,2020): -1 | Pass |
| 11 | Check that StandardDeviationAirTemperatureMonth function can correctly find and calculate standard deviation of wind speed for a month | Test StandardDeviationAirTemperatureMonth (vector, int, int)  Test StandardDeviationAirTemperatureMonth (TestLog,3,2016): 0.68  Test StandardDeviationAirTemperatureMonth (TestLog,1,2020): -1 | Pass |
| 12 | Check that WriteAppendFile able to correctly write to "WindTempSolar.csv" | Test WriteAppendFile (string)  \*Look for the WindTempSolar.csv file  teststring  teststring2 | Pass |

**Test 1-5**

A screenshot of a computer program

Description automatically generated

**Test 6-12**

A screenshot of a computer program

Description automatically generated

**Test 10**

A screenshot of a computer

Description automatically generated

## Validator Unit Test

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Description | Expected Output | Passed |
| 1 | Check that IsValidDate function can correctly determine if a date is valid when given the day, month, and year | Checking IsValidDate (int day, int month, int year)  Positive case (29/2/2020): 1  Negative case, wrong day (31/2/2020): 0  Negative case, wrong month (29/13/2020): 0  Negative case, wrong year (29/2/99): 0  Negative case, not leap year (29/2/2022): 0 | Pass |
| 2 | Check that IsValidDate function can correctly determine if a date is valid when given the month and year only | Checking IsValidDate (int month, int year)  Positive case (2/2020): 1  Negative case, wrong month (13/2020): 0  Negative case, wrong year (2/99): 0 | Pass |
| 3 | Check that IsValidMonth function can correctly determine if a int is valid month | Checking IsValidMonth (int month)  Positive case (2): 1  Negative case, wrong month (13): 0 | Pass |
| 4 | Check that IsLeapYear function can correctly determine if a year is a leap year | Checking IsLeapYear (int year)  Positive case (2020): 1  Negative case (2022): 0 | Pass |
| 5 | Check that IsValid24HourTime function can correctly determine if a given hour and minute is valid 24-hour time | Checking IsValid24HourTime (int hour, int minute)  Positive case (8,50): 1  Negative case, wrong hour (25,50): 0  Negative case, wrong minute (24,60): 0 | Pass |
| 6 | Check that IsNumber function can correctly determine if a provided string is positive int/float | Checking IsNumber (string input)  Positive case (415): 1  Positive case (63.41): 1  Negative case, negative int (-1): 0  Negative case, negative float (-1.2): 0  Negative case, too many decimal points (123456.123456789012345): 1  Negative case, octal numbers (08): 0 | Pass |
| 7 | Check that IsInteger function can correctly determine if a provided string is a positive integer | Checking IsInteger (string input)  Positive case (415): 1  Negative case (45.21): 0  Negative case (-1): 0  Negative case (08): 0 | Pass |

A screenshot of a computer

Description automatically generated

## Main SIT (Using Test.csv)

|  |  |  |  |
| --- | --- | --- | --- |
| Test | Description | Expected Output | Passed |
| 1 | Check that data/data\_source.txt file is in the wrong syntax. | Failed to open data\_source.txt. | Pass |
| 2 | Check if file name in data/data\_source.txt is in the wrong syntax. | Failed to open data file: tet.csv | Pass |
| 3 | Check that invalid month or year input while using function 1, it will result in error | Error: invalid month. Expects integer. | Pass |
| 4 | Check that invalid month and year combination will result in error and stopping of program | Error: invalid date. | Pass |
| 5 | Check that function 1 can print correctly average wind speed of the specific month and year in km/h and standard deviation of wind speed when data is found. | 2016  March 2016:  Average speed: 19.44 km/h  stdev: 0.55 | Pass |
| 6 | Check that function 1 will show no data if no data is found for the specific month and year | January 2016: No Data | Pass |
| 7 | Check that if user entered invalid year while using function 2, it will result in error | Error: invalid year. Expects integer. | Pass |
| 8 | Check that function 2 can correctly print the average ambient air temperature and standard deviation of air temperature of all 12 months in the specific year in degrees C and no data if no data is found for the month | 2016  January 2016: No Data  February 2016: No Data  March 2016: average:21.33 degrees C, stdev: 0.68  April 2016: No Data  May 2016: No Data  June 2016: No Data  July 2016: No Data  August 2016: No Data  September 2016: No Data  October 2016: No Data  November 2016: No Data  December 2016: No Data | Pass |
| 9 | Check that if user entered invalid year while using function 3, it will result in error | Error: invalid year. Expects integer. | Pass |
| 10 | Check that function 3 can correctly print the total solar radiation for each month for a specific year in kWh/m2 and no data if no data is found for the month | 2016  January 2016: No Data  February 2016: No Data  March 2016: 0.481833 kWh/m²  April 2016: No Data  May 2016: No Data  June 2016: No Data  July 2016: No Data  August 2016: No Data  September 2016: No Data  October 2016: No Data  November 2016: No Data  December 2016: No Data | Pass |
| 11 | Check that if user entered invalid year while using function 4, it will result in error and stopping of program | Error: invalid year. Expects integer. | Pass |
| 12 | Check that function 4 can correctly output the average wind speed, standard deviation of wind speed, average air temperature, standard deviation of air temperature and total solar radiation for each month of a specific year in km/h, degrees C and kWh/m2 and no line for months with no data | \*Check WindTempSolar.csv  2016  March,19.440001(0.550000),  21.329998(0.680000),0.481833 | Pass |
| 13 | Check that function $ can correctly output the blank field if data is not available for a particular field. | 2016  March,19.200001(0.520000),,0.585667 | Pass |
| 14 | Check that function 5 can quit the program | \*Program exits | Pass |
| 15 | Check that inputting any function other than 1-5 will return in error and repeatedly display the menu | Error: Unknown command. Only numbers 1-5 accepted.  1. Average wind speed and standard deviation for a given specified month and year.  2. Average ambient air temperature and sample standard deviation for each month of a specified year.  3. Total solar radiation in kWh/m2 for each month of a specified year.  4. Average wind speed in km/h (standard deviation), average ambient air temperature (standard deviation) and total solar radiation in kWh/m2 for each month of a specified year.  5. Exit the program. | Pass |

Test 1

A screen shot of a computer

Description automatically generated

Test 2

A screen shot of a computer

Description automatically generated

Test 3, 4, 5, 6

A black screen with white text

Description automatically generated

Test 7, 8

A computer screen with a black background

Description automatically generated

Test 9, 10, 11

A black screen with white text

Description automatically generated

Test 12

A screenshot of a computer

Description automatically generated

Test 13

A screenshot of a computer

Description automatically generated

Test 14

A black and white text

Description automatically generated

Test 15

A computer screen shot of a black screen

Description automatically generated

# Evaluation

+ Able to find and print average wind speed and standard deviation of wind speed of user's choice of month and year based on data file using function 1.

+ Able to find and print average ambient air temperature and standard deviation of air temperature of user's choice year based on data file using function 2.

+ Able to find total solar radiation of each month of user's choice of year based on data file using function 3.

+ Able to outfile file with data of average wind speed, standard deviation of wind speed, average ambient air temperature, standard deviation of air temperature and total solar radiation of each month of user's choice of year based on data file using function 4.

+ Able to convert wind speed from m/s to km/h.

+ Able to convert W/10min/m^2 to kWh/m^2.

+ Able to validate all data file's expected input and store in memory.

+ Able to validate all user input to ensure input is expected.

**- There are no requirements that this program does not meet.**

**Exception**

-when using function 4, if the empty column is in last place, need to put another comma to load the file. If not then, the program works as intended.

e.g.

WAST,DP,Dta,Dts,EV,QFE,QFF,QNH,RF,RH,S,SR,ST1,ST2,ST3,ST4,Sx,T

31/03/2016 9:00,14.6,175,17,0,1013.4,1016.9,1017,0,68.2,6,512,22.7,24.1,25.5,26.1,8,,

31/03/2016 9:10,14.6,194,22,0.1,1013.4,1016.9,1017,0,67.2,5,565,22.7,24.1,25.5,26.1,8,,