|  |
| --- |
| Four Walls Software |
| Test Summary & Defect Report |
| Landfill Labs Worker Service Prototype |

|  |
| --- |
| Husnain Ahmed (21308666)  1-8-2025 |

A logo for a software company

Description automatically generated

Contents

[Test Summary 3](#_Toc183187117)

[System Overview 4](#_Toc183187118)

[Testing Scope 4](#_Toc183187119)

[Test Environment & Tools 9](#_Toc183187120)

[Statistics 9](#_Toc183187121)

[Recommendations & Conclusion 13](#_Toc183187122)

[Defect Reports 16](#_Toc183187123)

Appendices …………………………………………………………………………………… ….……… 27

# Test Summary

The purpose of this report is to evaluate the testing process of the Landfill Labs Waste Management System. The system is designed to assist the council in managing waste from landfill sites by estimating waste distribution, identifying optimal recycling centres, and calculating transportation and processing times. The objective is to ensure that the system performs its functions as it should. It should handle various waste scenarios and have a user-friendly interface.

The testing process has revealed several defects, including, but not limited to, crashes caused by invalid inputs, incorrect travel time calculations, and minor spelling errors in the interface. To address these, it is recommended to fix these defects, standardise the output format, and implement better error-handling to improve system reliability and usability. These improvements will help the system meet its intended purpose.

## System Overview

## 

The Landfill Labs Waste Management System, developed by Four Walls Software for the council, is a user-friendly program that will allow the council to be more efficient when managing and disposing of any recyclable waste from historic landfill sites.

**Core System Logic:**

* **RunModel:**
* This is the core method for modelling waste scenarios.
  + It returns key results: travel duration, process duration, and total duration. These are needed to know how long it will take to transport and process your waste.
* **‘EstimateWasteSplit’:**
* This estimates the split of waste based on total volume.
* If total waste is less than 1250 cubic meters:
  + The ratio of waste is 50% plastic/glass, 50% paper.
* If total waste is greater than 1250 cubic meters:
  + The ratio of waste is 30% plastic/glass, 50% paper, and 20% metallic.
* **‘FindViableCentres’:**
* This identifies viable recycling centres based on waste type and distance.
* For metallic waste it includes centres within 3 hours travel time.
* For no metallic waste it only Alpha and Beta centres are viable.
* **‘FindOptimalCentre’:**
* This orders viable centres by proximity and priority.
* **Priority order:**
  1. Nearest centres.
  2. If multiple centres are equally near, the one with the highest generation (Gamma > Beta > Alpha).
  3. If generation is equal, the centre with fewer years of activity is prioritised.
  4. In case of ties, the optimal centre is chosen randomly.
* **‘CalculateTravelDuration’:**
* This calculates travel time based on the number of round trips needed for waste transport.
* **‘CalculateProcessDuration’:**
* This calculates processing time based on centre type and waste type.
* **Type Alpha:** 1 cubic meter per hour.
* **Type Beta:** 1.5 cubic meters per hour.
* **Type Gamma:**
  + 1.5 cubic meters per hour for plastic.
  + 2.0 cubic meters per hour for metallic.
  + 3.0 cubic meters per hour for paper.

Calculates travel and processing durations using specific formulas.

Orders recycling centres by priority, considering proximity, type, and operational age.

**Key features:**

* **The ability to model scenarios:**
  + The user can input data to create Historic Sites (Landfills) and Recycling centres.
  + The system uses this data to calculate the amount of time it will take to fill the recycling centre with waste and to process the waste.
  + It then outputs these times to the user as well as the total time.
* **Command Line Interface (CLI):**
  + The user can navigate through the CLI to interact with the system.
  + The CLI has a specific page (About page) to give the user information on how the system works and how to use it.
* **System constraints:**
  + Only 1 user at a time can currently use this system.
  + Only 1 transport truck is available at any time.
  + Each transport truck can only carry a maximum of 20 cubic metres of waste at a time.
* **Security:**
  + Only 1 user at a time can currently use this system as it is in a initial stage, this will be changed in the future and we will need to take steps to ensure that each users data remains secure and only accessible to them and them alone.
  + Data is not stored locally or on the cloud at this moment in time so it cannot be accessed my malicious third parties, in future iterations of this system we will need to ensure data privacy and protection by decentralising data and/or encrypting it.
  + The system follows the Single Responsibility Principle (SRP) and each function is responsible for a single task each.
  + The system will have to follow well established best security practices.
* **Error handling:**
  + The system is designed to fail gracefully in most cases and should prompt users to retry, proper messages should be displayed for invalid or missing inputs.
  + Numeric values aren't always formatted to 2 decimal places as they should be.
  + Some invalid scenarios result in exceptions or crashes.

## Testing Scope

The scope of testing for this project is designed to ensure that the system requirements are met and the developed features function as expected. The testing is focused on the key system logic outlined previously in this document and will be executed to verify the proper operation of the critical features.

**In-scope**

The following features and aspects of the system are considered in-scope for testing:

* **Worker Service**:
  + The system must be able to model waste scenarios.
  + Estimate waste splits and identify viable recycling centres.
  + Order recycling centres based on proximity or the highest waste generation.
  + Calculate travel duration and processing time for each selected recycling centre.
  + Checking if there are any spelling errors.
  + Checking if any numbers are outputted to 2 decimal points.
  + Testing that the program fails gracefully.
* **Security Criteria**:
  + Ensure that the system adheres to established best practices for security.
  + Protect against privacy breaches and ensure secure handling of user data.

The files that I will code tests for are as follows: Alpha.java, Beta.java, Gamma.java, Historic.java, Location.Java, Recycling.Java, Site.java, Transport.java, ScenarioConfiguration.java and Utils.java.

**Out-of-scope**

The following aspects are considered out-of-scope for testing as they are not critical to the current implementation of the system:

* **System Performance**:
  + Scalability and responsiveness under stress.
  + Load testing and determining response times under heavy usage are not required at this stage.

I will not code any tests for the Main.java file as all its functions are private, this will instead be tested using manual tests.

**Types of testing used:**

* **Manual Testing:** This was used to manually test the CLI. The data provided in the ‘TechMemo.md’ provided a source of information to check against the system. I used this data to check for any spelling errors in the text that the system outputs to the user. I also tested if the system handled errors gracefully without throwing exceptions and whether the outputted numbers were concatenated to 2 decimal places.
* **Unit tests:** These were used to test individual methods like ‘findViableCentres’ to ensure that they work as expected.
* **Integration tests:** These were used to Test functions that call other functions and classes like ‘findOptimalCentres’ which calls 3 functions, or other functions that use instances of the historic and recycling classes.
* **Parameterised tests with CSV data:** These were used to save time and improve code readability, I could pass in extra data for more code coverage and to test different scenarios for functions such as null values, empty values, different waste splits and distributions of waste types.

## Test Environment & Tools

I’ve thoroughly tested the CLI, checking all commands, handling edge cases, and ensuring clear feedback and smooth performance. I focused on usability, making sure it’s intuitive, responsive, and works well with the backend.

Below are the tools I have used for my automated tests:

|  |  |
| --- | --- |
| **Tool** | **Explanation** |
| Laptop | My personal laptop (Razer Blade 14, 2014 model) that I have used to run the SUT on and write tests on. This uses windows 10 as an operating system. |
| IntelliJ | The IDE of choice for testing the SUT. |
| JUnit5 (Java) | The testing framework I have used to write tests. |
| CSV data | I have used this to parameterise some of my tests and reduce code redundancy / repetition. |
| TestRecycling | A subclass of Recycling I have used for better ease of testingas the original class is abstract. |
| Test data | The following mock data was used for testing:**centre1**: Location.B, 5 years active, generation "Alpha", rates [1.0, 1.0, 1.0]**centre2**: Location.B, 10 years active, generation "Beta", rates [1.5, 1.5, 1.5]**centre3**: Location.C, 15 years active, generation "Gamma", rates [1.5, 2.0, 3.0] This data was set up in a void setup() with the ‘@BeforeEach’ notation, this allowed me to save space and stop repetitive code by not having to recreate this code for every test that required it. |
| GitHub | Used for version control as seen in figure 1. |
| Microsoft Excel | Spreadsheet used to document test cases. |
| Microsoft Word | Notepad application used to write this defect report. |

Table 1 — Tools used for testing

I used the information in the following documents to write my tests:

* **Spec\_v1.docx:** This is the design spec for the system.
* **TechMemo.md:** This is an explanation of the SUT and provided some test data in then form of examples of spelling error free text that I could use for my manual tests.
* **MemorandumOfUnderstanding.md:** An explanatory email that briefed me on the system and its requirements, this included some examples of how the system should work.

A screenshot of a computer

Description automatically generated

Figure 1 — GitHub for version control

## Statistics

I have created 16 Manual tests and 126 automated tests.

Of the manual tests 5 passed, 1 was obsolete and 10 failed.

Of the 126 automated tests 109 passed and 16 failed.

A screenshot of a computer

Description automatically generated

Figure 2 — Tests run

Of the 10 failed manual tests, 5 were of a minor severity as they are due to spelling errors and do not impact the core functionality of the system.

The other 5 are of critical severity.

4 of these are caused by invalid or unexpected inputs causing exceptions to be thrown.

1 of these is caused by an inability to manually add a ‘Gamma’ generation recycling centre to the scenario.

These 5 defects directly impact the core functionality of the system and ruin the user experience so they must be rectified before the final handover of the system to the client.

Of the 16 failed automated tests, 3 are of minor severity as they are due to the user being able to enter negative values for initial waste or other parameters in the creation of Historic or Recycling classes.

5 are of moderate severity as they are due to incorrect calculations in the ‘getTravelTime’ functions, this impacts the final calculation of the total time it takes to dispose of the waste but can be easily rectified as will be explained later.

2 are of critical severity as they pertain to runtime exceptions being thrown when functions are invoked with empty parameter values.

The remaining 6 are of moderate severity as they are due to invalidly inputted generation name (not alpha, beta or gamma) causing unexpected results, in practice the system will not allow the user to enter an invalid input and will ask them to re-enter their data.

100% test coverage has been achieved for all files besides the main file which is covered by manual testing.

A screenshot of a computer program

Description automatically generated

Figure 3 — Test coverage report

The Main.java file consists solely of private functions and so I have not written unit tests for it.

## Recommendations & Conclusion

I recommend being careful and double checking any AI generated code / unit tests as it is very rarely 100% correct, for example the SUT only has Locations A, B and C. The AI generated Tests were including Location D among others.

I also recommend changing all the incorrect text on the CLI to be the same as the proofread versions that are on the TechMemo.md sheet. And fix any other spelling errors that have been found such as the misspelling of ‘Please’ as ‘Peeas’.

Another Recommendation that I have is to concatenate all outputs to 2 decimal points as specified in TechMemo.md.

We could also try to fix the code that throws exceptions and change it so that when invalid data is entered the system asks the user to re-enter the data in a correct format instead of crashing. This could be extended to preventing the user from entering a negative integer for waste values.

I also recommend altering the code for ‘Location.TravelTime’. As shown in Appendices A (Line 36) the travel time between the same locations (A to A, B to B, C to C) is ‘0.0’. This is incorrect, according to the Memorandum of understanding this should instead be ‘1.0’.

I also recommend altering the code for ‘Transport.getTravelTime’. As shown in Appendices B (Line 47) the travel time between the locations B to B is set to 3. This is incorrect, according to the Memorandum of understanding this should instead be 1.

To fix this we must change the second clause of the following if statement.

‘((start == Location.*B* && end == Location.*C*) || (start == Location.*B* && end == Location.*B*) )’

Should be changed to

‘((start == Location.*B* && end == Location.*C*) || (start == Location.*C* && end == Location.*B*) )’

I also recommend altering the code in the Main.java file in the ‘collectGeneration’ function. As Shown in Appendices C (Line 231) The ‘Gamma’ generation centre is misspelt as ‘Gama’, this causes the user to be unable to set any recycling centres to that generation as seen in CreateRecycling\_AllGeneration\_TC001 (Appendices D).

I also recommend altering the code of the 'Utils.findNearestCentres' and the 'Utils. 'findOptimalCentre' to have some exception handling and to return an empty array list or an empty value when no candidates are provided. It currently throws an exception in that case as seen in ‘Utils\_FindNearestCentres\_NoCandidatesProvided\_DoesNotThrowException’ and ‘Utils\_FindOptimalCentre\_NoCandidatesProvided\_DoesNotThrowException’. See the errors in Appendices E.

And finally, I recommend altering the code of ‘Utils.CompareGenerations’ to not allow any values not in the generations list (see Appendices F) to be passed in. This will ensure that even if the users manage to get past the error handling and enter invalid data the system will work as expected.

Currently this causes defects such as ‘Utils\_CompareGenerations\_ DF\_002. I believe it is for the best if we remedy this by only allowing data existing in the generations list to be passed in and if that is not the case the system should return 0 as the generation is invalid.

In conclusion, these recommendations can serve as guidelines for the development of the next iteration of the system, perfecting it even further and remedying any issues with spelling mistakes, error / exception handling and system functionality. By following these recommendations, the system will become more user friendly and will not have any errors with its calculations.

# Defect Report - Manual Tests

Table 2: CLI Defects

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Defect ID** | **Description** | **Steps to Reproduce** | **Expected Behaviour** | **Actual Behaviour** | **Priority** | **Severity** | **Status** | **Source** |
| MenuSpellingError\_DF\_001 | The text on the ‘About’ page is not the same as the text shown in Appendices B (TechMemo.md), This makes the user experience slightly worse. | 1) Start the program. 2) Enter '2' to navigate to the 'About' screen. 3) View the 'About' Text and manually compare them to Appendices B (TechMemo.md),. | The 'About' Text is displayed. | The word 'application' is misspelt as 'aplicato'. This does not meet the expected requirements outlined in appendix B (TechMemo.md),. | Low | Minor | In progress | MenuSpellingError\_TC\_002 |
| MenuSpellingError\_DF\_002 | The text on the 'Historic Site Created’ page is not the same as the text shown in Appendices D (TechMemo.md), This makes the user experience slightly worse. | 1) Start the program. 2) Enter '1' to navigate to the 'Scenario Options' screen. 3) Enter '1' to navigate to the 'Historic Site' screen. 4) Enter Test Data based on Appendices D (TechMemo.md),. 5) View the 'Historic Site Created' Text and manually compare them to Appendices D (TechMemo.md),. | The 'Historic Site Created' Text is displayed. | The text is correct with no spelling errors but this does not meet the expected requirements outlined in appendix D (TechMemo.md), as the outputted number is not displayed to 2 decimal points as it should instead it is shown as '5000.000000' with 6 dp. | Low | Minor | In progress | MenuSpellingError\_TC\_004 |
| MenuSpellingError\_ DF\_003 | The text on the 'Recycling Centres Created' page is not the same as the text shown in Appendices E (TechMemo.md), This makes the user experience slightly worse. | 1) Start the program. 2) Enter '1' to navigate to the 'Scenario Options' screen. 3) Enter '2' to navigate to the 'Recycling Centres' screen. 4) Enter Test Data based on Appendices E (TechMemo.md),. 5) View the 'Recycling Centres Created' Text and manually compare them to Appendices E. | The 'Recycling Centres Created' Text is displayed. | The actual response is only '2 recycling centres created', this does not meet the expected requirements outlined in appendix E (TechMemo.md),. | Low | Minor | In progress | MenuSpellingError\_TC\_005 |
| MenuSpellingError\_ DF\_004 | The text on the ‘Scenario Completion Created' page is not the same as the text shown in Appendices F (TechMemo.md), This makes the user experience slightly worse. | 1) Start the program. 2) Enter '1' to navigate to the 'Scenario Options' screen. 3) Enter '1' to navigate to the 'Historic Site' screen. 4) Enter Test Data based on Appendices D. 5) Enter '2' to navigate to the 'Recycling Centres' screen. 6) Enter Test Data based on Appendices E. 7) View the 'Scenario Completion' Text and manually compare them to Appendices F. | The 'Scenario Completion Created' Text is displayed. | The text is correct with no spelling errors but this does not meet the expected requirements outlined in appendix F (TechMemo.md), as the outputted numbers is are displayed to 2 decimal points as it should instead it is shown as '250.000000' with 6 dp. | Low | Minor | In progress | MenuSpellingError\_TC\_006 |
| MenuSpellingError\_ DF\_005 | The invalid input text has spelling errors. This makes the user experience slightly worse. | 1) Start the program. 2) Enter '1' to navigate to the 'Scenario Options' screen. 3) Enter '1' to navigate to the 'Historic Site' screen. 4) Enter Incorrect Test Data. 5) Check the text for any spelling errors. | The system will run without crashing and display the invalid input text. | The actual response is 'Invalid choice. Peeas try again.' this is spelt incorrectly. | Low | Minor | In progress | MenuSpellingError\_TC\_008 |
| ExceptionHandling\_ DF\_001 | When no data is inputted an exception is thrown, this does not meet the expectation in the tech memo that the CLI fails gracefully, which is a critical issue for user experience and functionality. | 1) Start the program. 2) Enter '1' to navigate to the 'Scenario Options' screen. 3) Enter '3' to run the scenario. 4) Check if an error has been thrown or if the system asks the user to try again and re-enter a valid input. | The system will run without crashing. | Throws exception and crashes. | High | Critical | In progress | ExceptionHandling\_TC\_001 |
| ExceptionHandling\_ DF\_002 | When insufficient data is inputted. (requires both 'Historic Site' and 'Recycling Centre' to be created but only 1 is) an exception is thrown, this does not meet the expectation in the tech memo that the CLI fails gracefully, which is a critical issue for user experience and functionality. | 1) Start the program. 2) Enter '1' to navigate to the 'Scenario Options' screen. 3) Enter '1' to navigate to the 'Historic Site' screen. 4) Enter Test Data. 5) Check if an error has been thrown or if the system asks the user to try again and re-enter a valid input. | Check if an error has been thrown or if the system asks the user to try again and re-enter a valid input. | Throws exception and crashes. | High | Critical | In progress | ExceptionHandling\_TC\_002 |
| ExceptionHandling\_ DF\_003 | When insufficient data is inputted. (requires both 'Historic Site' and 'Recycling Centre' to be created but only 1 is) an exception is thrown, this does not meet the expectation in the tech memo that the CLI fails gracefully, which is a critical issue for user experience and functionality. | 1) Start the program. 2) Enter '1' to navigate to the 'Scenario Options' screen. 3) Enter '2' to navigate to the 'Recycling Centre' screen. 4) Enter Test Data. 5) Check if an error has been thrown or if the system asks the user to try again and re-enter a valid input. | Check if an error has been thrown or if the system asks the user to try again and re-enter a valid input. | Throws exception and crashes. | High | Critical | In progress | ExceptionHandling\_TC\_003 |
| ExceptionHandling\_ DF\_004 | When expecting an integer and inputting a unexpected character such as a symbol or string an exception is thrown, this does not meet the expectation in the tech memo that the CLI fails gracefully, which is a critical issue for user experience and functionality. | 1) Start the program. 2) Enter incorrect test data. 3) Check if an error has been thrown or if the system asks the user to try again and re-enter a valid input. | Check if an error has been thrown or if the system asks the user to try again and re-enter a valid input. | Throws exception and crashes. | High | Critical | In progress | ExceptionHandling\_TC\_004 |
| CreateRecycling\_AllGeneration\_ DF\_001 | When trying to initialise Recycling centres with all possible generations Gamma generation centres cannot be created due to a spelling error in main.java misspelling ‘Gamma’ as ‘gama’. This causes the core functionality of the runmodel to fail and is a critical issue for user experience and functionality. | 1) Start the program. 2) Enter '1' to navigate to the 'Scenario Options' screen. 3) Enter '2' to navigate to the 'Recycling Centres' screen. 4) Enter Incorrect Test Data. 5) Check the text for any spelling errors. | Recycling centres are all created. | 'Alpha' and 'Beta' generation centres have been created successfully whilst 'Gamma' centres have not and the system responds with "Invalid choice. Please try again." . The system does not allow 'Gamma' Generation centres to be created. | High | Critical | In progress | CreateRecycling\_AllGeneration\_TC\_001 |

# Defect Report - Model Tests

Table 3: Models Defects

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Defect ID** | **Description** | **Steps to Reproduce** | **Expected Behaviour** | **Actual Behaviour** | **Priority** | **Severity** | **Status** | **Source** |
| Location\_TravelTime\_ DF\_001 | When testing 'Location.travelTime' between the same locations, it should return 1.0 but instead returns 0, this is not in accordance with the memorandum of understanding. | 1) Invoke 'the travelTime' method with the test data. 2) Assert that the actual result is the same as the expected result. | '1.0' | The travel time is '0.0', this is not in accordance with the memorandum of understanding. | Mid | Moderate | In progress | Location\_TravelTime\_SameLocation\_Returns1 |
| Transport\_getTravelTime\_ DF\_001 | When testing 'Transport.getTravelTime' from location C to B, it should return 3.0 but instead returns 1.0, this is not in accordance with the memorandum of understanding. | 1) Initialise an instance of the transport with the parameterised test data. 2) Invoke the 'getTravelTime' method with the test data. 3) Assert that the actual result is the same as the expected result. | '3.0' | The travel time between locations "B,C" = 3.0 as expected but the travel time between locations "C,B" = 1.0 which is incorrect. | Mid | Moderate | In progress | Transport\_getTravelTime\_BC\_Returns3 |
| Transport\_getTravelTime\_ DF\_002 | When testing 'Transport.getTravelTime' from location B to B, it should return 1.0 but instead returns 3.0, this is not in accordance with the memorandum of understanding. | 1) Initialise an instance of the transport with the parameterised test data. 2) Invoke the 'getTravelTime' method with the test data. 3) Assert that the actual result is the same as the expected result. | '1.0' | The travel time between locations "A, A" and "C, C" = '1.0' as expected but the travel time between locations "B,B" = 3.0 which is incorrect. | Mid | Moderate | In progress | Transport\_getTravelTime\_SameLocation\_Returns1 |
| Transport\_SetWasteTypes\_ DF\_001 | Waste types can be set as negative values, this does not make sense as you cannot have negative cubic metres of waste, the code should reflect this. | 1) Initialise an instance of the transport with the test data. 2) Invoke the 'setPaperWaste' method with the test data. 3) Invoke the 'setPlasticGlassWaste' method with the test data. 4) Invoke the 'setMetallicWaste' method with the test data. 5) Invoke the 'getTotalWaste' method. 4) Assert that the actual result from the 'getTotalWaste' method is the same as the expected result. | If the result is equal to -1500 then the test will fail | The value of the total waste is '-1500'. Which should not be the case. | Low | Minor | In progress | Transport\_SetWasteTypesAsNegative\_DoesNotInitialise |
| Recycling\_setYearsActiveAsNegative\_ DF\_001 | YearsActive can be set as a negative value, this does not make sense as you cannot have something that has been active for - 10 years and is set in the future. The code should reflect this. | 1) Initialise an instance of the recycling class with the test data. 2) Invoke the 'getYearsActive' method. 3) Assert that the actual result from the 'getYearsActive' method is the same as the expected result. | If the result is equal to -10 then the test will fail | The value of the total waste is '-10'. Which should not be the case. | Low | Minor | In progress | Recycling\_setYearsActiveAsNegative\_DoesNotInitialise |
| Historic\_SettingValuesToNegative\_ DF\_001 | Waste can be set as negative values, this does not make sense as you cannot have negative cubic metres of waste, the code should reflect this. | 1) Initialise an instance of the historic class at 'location.A' with -500 inital waste. 2) Invoke the 'setPlasticGlass', 'setMetallic', 'setPaper', and 'setRemainingWaste' methods with the test data. 3) Invoke the get methods and store the values that were set as variables . 4) Assert that the actual rvalue of those variables is the same as the expected result. | If the variables are equal to -500 then the test will fail | The values of all the waste types is '-500'. Which should not be the case. | Low | Minor | In progress | Historic\_SettingValuesToNegative\_DoesNotInitialise |

# Defect Report - Utils Tests

Table 4: Utils Defects

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Defect ID** | **Description** | **Steps to Reproduce** | **Expected Behaviour** | **Actual Behaviour** | **Priority** | **Severity** | **Status** | **Source** |
| Utils\_FindNearestCentres\_ DF\_001 | The 'findNearestCentres' function throws an exception when no centres are passed in as parameters, which is a critical issue for user experience and functionality. | 1) Initialise an instance of the historic class at 'location.A' with 10 inital waste. 2) Initialise an empty arraylist of recycling centres. 3) Assert that 'findNearestCentres' function does not throw an exception when called with the historic class and empty arraylist as its parameters. | An exception is not thrown. | A runtime exception is thrown. (java.util.NoSuchElementException). | High | Critical | In Progress | Utils\_FindNearestCentres\_NoCandidatesProvided\_DoesNotThrowException |
| Utils\_FindOptimalCentre\_ DF\_001 | The 'findOptimalCentre' function throws an exception when no candidate centres are provided, which is a critical issue for user experience and functionality. | An instance of the historic class is initalised. Initialise an empty array list of recycling centres. | An exception is not thrown. | A runtime exception is thrown. (java.util.NoSuchElementException). | High | Critical | In Progress | Utils\_FindOptimalCentre\_NoCandidatesProvided\_DoesNotThrowException |
| Utils\_CompareGenerations\_ DF\_001 | The 'compareGenerations' function does not always return 0 when invalid (non existing) generations are passed in. | 1) Invoke the 'compareGeneration' method with the passed in parameters as the value of the 2 generations that will be compared. 2) Assert that the returned integer value is equal to what is expected. | '0' | Not all returned values were equal to '0'. "'Alpha', 'xxx'" returns '1' and "'xxx', 'Alpha'" returns '-1'. But only "'xxx', 'xxx'" returns '0' as expected. | Mid | Moderate | In Progress | Utils\_CompareGenerations\_InvalidGenerations\_ReturnsZero |
| Utils\_CompareGenerations\_ DF\_002 | The 'compareGenerations' function does not always return 0 when empty generations are passed in. | 1) Invoke the 'compareGeneration' method with the passed in parameters as the value of the 2 generations that will be compared. 2) Assert that the returned integer value is equal to what is expected. | '0' | Not all returned values were equal to '0'. "'', 'Alpha'" returns '-1' and "'Alpha', ''" returns '1'. But only "'', ''" returns '0' as expected. | Mid | Moderate | In Progress | Utils\_CompareGenerations\_EmptyInputGeneration\_ReturnsZero |
| Utils\_CompareGenerations\_ DF\_003 | The 'compareGenerations' function does not always return 0 when null generations are passed in. | 1) Invoke the 'compareGeneration' method with the passed in parameters as the value of the 2 generations that will be compared. 2) Assert that the returned integer value is equal to what is expected. | '0' | Not all returned values were equal to '0'. null, 'Alpha'" returns '-1' and "'Alpha', null returns '1'. But only null, null returns '0' as expected. | Mid | Moderate | In Progress | Utils\_CompareGenerations\_NullInputGeneration\_ReturnsZero |

**Appendices A:**  
  
  
  
A screen shot of a computer

Description automatically generated

**Appendices B:**

A screen shot of a computer program

Description automatically generated

**Appendices C:**

A screenshot of a computer program

Description automatically generated

**Appendices D:**

A screen shot of a computer

Description automatically generated

**Appendices E:**

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

**Appendices F:**

A screen shot of a computer

Description automatically generated