Table 1, A summary of test case designed

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case No | Test case description | Test data | Expected results | Actual results | Status (Pass/Fail) |
| 01 | Test for menu display when running the code | N/A | The menu displays in the console to allow user input |  | Pass |
| 02 | Test for adding a pet | Menu option ‘1’  ‘dog’ for pet type  ‘Monty’ for name  ‘2’ for age  ‘Brown’ for colour  ’10.2’ for weight  ‘Retriever’ for breed  ‘4’ to view the pet added | The test data will be saved and then displayed correctly |  | Pass |
| 03 | Test for adding a pet | Menu option ‘1’  ‘cat for pet type  ‘Milly’ for name  ‘5’ for age  ‘Brown’ for colour  ‘6’ for weight  ‘British shorthair’ for breed  ‘4’ to view the pet added | The test data will be saved and then displayed correctly |  | Pass |
| 04 | Error handling test for adding a pet | Menu option ‘1’  ‘parrot’ to test error handling | An appropriate error message will be displayed, then allowing the user to continue |  | Pass |
| 05 | Test for modifying pet details | Menu option ‘2’    ‘Monty’ to select the pet  menu option ‘4’  ‘11’ for new weight  ‘4’ to list and view the new weight | The new weight will be saved and a appropriate message will be displayed to the user to let them know |  | Pass |
| 06 | Test for viewing the current pets added | Menu option ‘4’ | It will display the added pet ‘Monty’ and the appropriate details |  | Pass |
| 07 | Test for searching pets by name | Menu option ‘5’  ‘Monty’ to search for the name | It will display the details of the pet, and the ‘speak’ method will be called, making the animal noise |  | Pass |
| 08 | Error testing for searching pets by name | Menu option ‘5’  ‘Max’ to search for the name. which doesn’t exist | It will display an appropriate error message telling the user the pet was not found |  | Pass |
| 09 | Test for clinic reporting | Menu option ‘6’ | It will display the clinic report, with 1 dog, 1 cat and the dominant colour being ‘Brown’ |  | Pass |
| 10 | Test for removing a pet | Menu option ‘3’  ‘Milly’ as the pet to remove  ‘4’ to view the current pets | After Milly has been removed, it will show an appropriate confirmation message, and only Monty will be left |  | Pass |
| 11 | Error testing for removing a pet | Menu option ‘3’  ‘Max’ as the pet to remove  ‘4’ to view the current pets | After Max is inputted, it will show an appropriate error message saying the pet is not found |  | Pass |
| 12 | Testing of exit and save data to disk | Menu option ‘7’ | After the program exits, the data will be saved to ‘PetDetails.txt’ and the clinic report will be saved to ‘ClinicDetails.txt’ |  | Pass |
| 13 | Test of repopulating the program on start | Run the program  Menu option ‘4’ | After ‘4’ is selected, it displays ‘Monty’ and the appropriate details |  | Pass |

Table 2, Unit tests

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case No | Unit test description | Unit test Code | Output | Status (Pass/Fail) |
| 1 | Test for pet speak method |  |  | Pass |
| 2 | Test for adding a pet |  |  | Pass |
| 3 | Test for removing a pet |  |  | Pass |
| 4 | Test for searching of pets |  |  | Pass |
| 5 | Test for clinic report |  |  | Pass |
| 6 | Test for saving and loading data to/from disk |  |  | Pass |

Table 3, A summary of the implementation of each requirement

|  |  |  |
| --- | --- | --- |
| Requirement | Implemented (Yes/No) | Test case ID designed |
| Ability to pre-populate the application  with any previously store data | Yes | Table 1, ID13  Table 2, ID6 |
| Ability to add a pet | Yes | Table 1, ID3,4  Table 2, ID2 |
| Ability to delete a pet | Yes | Table 1, ID10,11  Table 2, ID3 |
| Ability to modify a pet | Yes | Table 1, ID5 |
| Ability for a pet to ‘speak’ | Yes | Table 1, ID7  Table 2, ID1 |
| Ability to search for a pet | Yes | Table 1, ID7,8  Table 2, ID4 |
| Ability to view all pets | Yes | Table 1, ID6 |
| Report of the clinic | Yes | Table 1, ID9  Table 2, ID5 |
| Save all data to disk | Yes | Table 1, ID12  Table 2, ID6 |