**Requirements:**

* Read representation of case diagram in xml format
* Store STRIDE matrix in code
* Generate candidate misuse cases using STRIDE matrix
* List the candidate misuse cases
* Prune candidate misuse cases to remove invalid cases
* List misuse case and threatened node in console

**Non-functional:**

* The program will be written in C++

**Test Cases**

**Application System:** Automatic Misuse Case Generation Tool

**Date:** 19/05/2016

**Tests Conducted By:** Alex Till

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test Description** | **Expected Behaviours/ Evaluation Criteria** | **Outcome** |
| Test-01 | Check that application correctly reads an xml representation of a use case diagram. | The application will be able to correctly read a representation of case diagram in xml format. | The xml file was read correctly and properly inserted into memory. |

**Test Passed?:** YES.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test Description** | **Expected Behaviours/ Evaluation Criteria** | **Outcome** |
| Test-02 | Check that the STRIDE matrix is stored in the program code. | The application will have a STRIDE matrix stored in the program code. | A STRIDE matrix is stored in the program code. |

**Test Passed?:** YES.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test Description** | **Expected Behaviours/ Evaluation Criteria** | **Outcome** |
| Test-03 | Check that the program generates candidate misuse cases for each use case. | Each use case will have 9 candidate misuse cases generated based on the STRIDE matrix. | 9 misuse cases were generated for each use case in the diagram. |

**Test Passed?:** YES.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test Description** | **Expected Behaviours/ Evaluation Criteria** | **Outcome** |
| Test-04 | Check that the program lists each misuse case and threatened node in the console. | The application will list all the candidate misuse cases. | The misuse cases were listed correctly. |

**Test Passed?:** YES.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test Description** | **Expected Behaviours/ Evaluation Criteria** | **Outcome** |
| Test-05 | Check that the pruning algorithm receives the candidate misuse cases. | After the candidate misuse cases are generated they will be sent to the pruning algorithm. | The candidate misuse cases were sent to the pruning algorithm correctly. |

**Test Passed?:** YES.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test Description** | **Expected Behaviours/ Evaluation Criteria** | **Outcome** |
| Test-06 | Check that the program prunes the candidate misuse cases correctly. | The program will prune the candidate misuse cases down so that all invalid ones are removed. | No pruning has occurred at all. |

**Test Passed?:** NO.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test ID** | **Test Description** | **Expected Behaviours/ Evaluation Criteria** | **Outcome** |
| Test-07 | Check that the program lists each misuse case and threatened node in the console. | The application will list all the misuse cases returned from the pruning algorithm. | The misuse cases were listed correctly. |

**Test Passed?:** YES.