|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case Id:** | **UC\_01** | **Use Case Name:** | Choose . . . | Test Date: |  |
| **Test Case Id:** | **TC\_01** | **Test Case Name:** | Store (navbar) | Tester: |  |
| **Steps** | **Inputs** | **Expected Output** | Actual Result | (Pass/Fail) | Comment |
| 1. | Open the Code Clone Detection Tool | Tool interface opens successfully | Correctly Functioned | Pass | Tested by - Mal |
| 2. | Select a code file for analysis for the Open Code File | Codebase loads into the tool | Correctly Functioned | Pass | Tested by - Mal |
| 3. | Adjust detection similarity scale and apply detection similarity | Settings update successfully | Correctly Functioned | Pass | Tested by - Mal |
| 4. | Initiate clone detection by clicking on Run Clone Detection and selecting same file as previously selected | Tool starts analyzing the codebase | Correctly Functioned | Pass | Tested by - Mal |
| 5. | Review the generated report | Report displays detected clones with similarity percentages. | Correctly Functioned | Pass | Tested by - Dem |
| 6. | Assess the types of clones detected | Correct clone types are identified | Correctly Functioned | Pass | Tested by - Dem |
| 7. | Save the report either PDF/CSV | Report is saved successfully | Correctly Functioned | Pass | Tested by - Dem |
|  | **UC\_02** |  |  |  |  |
| 1. | Open the previously saved clone report | Report opens successfully | Correctly Functioned | Pass | Tested by - Mal |
| 2. | Review Detected clones in the report | Clones are displayed with relevant details | Correctly Functioned | Pass | Tested by - Mal |
| 3. | Plan refactoring strategy | Identified redundant code blocks are selected for refactoring | Correctly Functioned | Pass | Tested by - Mal |
| 4. | Modify the code to remove identified clones | Code is refactored without affecting functionality | Correctly Functioned | Pass | Tested by - Mal |
| 5. | Test the refactored codebase | Code works with no new bugs | Correctly Functioned | Pass | Tested by - Mal |
| 6. | Rerun clone detection analysis | Updated report shows reduced redundant clones | Correctly Functioned | Pass | Tested by - Dem |
| 7. | Save the updated report | Report saves successfully with new data | Correctly Functioned | Pass | Tested by - Dem |
|  | **UC\_03** |  |  |  |  |
| 1. | Open the tool’s GUI | GUI opens and displays detected clones | Correctly Functioned | Pass | Tested by - Dem |
| 2. | View all detected clones | Clones are sorted by type and similarity percentage | Correctly Functioned | Pass | Tested by - Dem |
| 3. | Apply filter to show only Type 1 clones with 100 similarity | Filter works correctly | Correctly Functioned | Pass | Tested by – Mal |
| 4. | Click on a specific clone to navigate to its code location | The tool highlights the clone in the database | Correctly Functioned | Pass | Tested by – Mal |
| 5. | Mark clones for refactoring | Clones are marked for future refactoring | Correctly Functioned | Pass | Tested by – Mal |
| 6. | Save the session with current settings | Session is saved successfully | Correctly Functioned | Pass | Tested by – Mal |