|  |  |
| --- | --- |
| **Test ID: 1** | |
| Name | Projekt laden mit Standardauswahl |
| Testvoraussetzungen | Mindestens ein Projekte vorhanden |
| Testablauf | 1. Projekt in Dropdown auswählen 2. Klick auf „Load“ 3. „Select existing Test-Project…“ auswählen |
| Erwartetes Resultat: | Die Auswahl der Dropdown sollte nicht geladen werden können. Der Load-Button sollte nicht aktiviert sein. |

|  |  |
| --- | --- |
| **Test ID: 2** | |
| Name | Unechtes Projekt |
| Testvoraussetzungen | - |
| Testablauf | 1. In „Projects“ ein Ordner mit beliebigem Namen anlegen. 2. GUI öffnen 3. Auswahl des vorher vergebenen Namens in der Drop-Down 4. Klick auf „Load“ |
| Erwartetes Resultat: | Die Auswahl darf nicht geladen werden können. Das GUI bleibt unverändert. Der „Load“-Button ist aktiviert und lässt, wenn er angeklickt wird, eine entsprechende Meldung erscheinen. |

|  |  |
| --- | --- |
| **Test ID: 3** | |
| Name | Control-Model verändern |
| Testvoraussetzungen | Ein Projekt mit Control-Model |
| Testablauf | 1. Projekt laden (Control-Model wird geladen) 2. Die Datei im Ordner „Control\_Model“ umbenennen oder löschen. 3. Klick auf „Modify Control-Model“ |
| Erwartetes Resultat: | Die Datei sollte nur bearbeitet werden können, wenn die Anzeige mit dem Befund im Verzeichnis übereinstimmt. Ansonsten sollte das Öffnen verweigert werden mit dem entsprechenden Hinweis.  Nach erneutem Laden des Projekts sollte die Datei, falls sie gelöscht wurde nicht mehr angezeigt werden und sonst mit dem neuen Namen. |

|  |  |
| --- | --- |
| **Test ID: 4** | |
| Name | Dateiauswahl umgehen mit „All Files“ |
| Testvoraussetzungen | Ein Projekt mit Control-Model |
| Testablauf | 1. Projekt laden 2. Klick auf „Select existing Control-Model“ 3. Dateityp auf „All files“ umstellen 4. Beliebige Nicht-Modell-Datei auswählen 5. Klick auf „Öffnen“ |
| Erwartetes Resultat: | Die Datei mit dem falschen Dateityp darf weder im GUI angezeigt, noch ins Projektverzeichnis kopiert werden.  Der Benutzer sollte über die Umstände informiert werden. |

|  |  |
| --- | --- |
| **Test ID: 5** | |
| Name | Korrupte MDL-Datei |
| Testvoraussetzungen | Ein Projekt |
| Testablauf | 1. Leere Datei mit der Endung „.mdl“ im Ordner „Control\_Model“ erstellen 2. Projekt und Model laden und dann auf „RUN“ klicken |
| Erwartetes Resultat: | Entsprechende Meldung im Status-Textfeld |

|  |  |
| --- | --- |
| **Test ID: 6** | |
| Name | Datei-Auswahl abbrechen |
| Testvoraussetzungen | Projekt mit Control-Model |
| Testablauf | 1. Projekt auswählen und laden 2. Klick auf „Select existing Control-Model“ und dann auf „abbruch“ |
| Erwartetes Resultat: | Bisherige Auswahl steht immer noch im Textfeld (keine Änderung im GUI) |

|  |  |
| --- | --- |
| **Test ID: 7** | |
| Name | Test 3,4,5 und 6 mit TsNet-File anstatt Control-Model |
| Testvoraussetzungen | Ein Projekt mit TsNet-File |
| Testablauf | Wie in den jeweiligen Tests. |
| Erwartetes Resultat: | Wie in den jeweiligen Tests. |

|  |  |
| --- | --- |
| **Test ID: 8** | |
| Name | Ändern des TsNet-File und der Drop-Down |
| Testvoraussetzungen | Ein Projekt mit mindestens zwei TsNet-Files |
| Testablauf | 1. TsNet-File laden 2. Drop-Down betrachten 3. Anderes TsNet-File laden 4. Drop-Down betrachten |
| Erwartetes Resultat: | Die Drop-Down passt den Inhalte den Dateien entsprechend an. |

|  |  |
| --- | --- |
| **Test ID: 9** | |
| Name | Datei entfernt vor Klick auf RUN |
| Testvoraussetzungen | Vollständiges Projekt |
| Testablauf | 1. Projekt laden 2. Control-Model, TsNet-File und Sheet auswählen 3. Entweder Control-Model oder TsNEt-Datei löschen 4. Klick auf „RUN“ |
| Erwartetes Resultat: | Wenn eine Datei fehlt, sollte die Simulation nicht durchgeführt werden können. Eine entsprechende Meldung weist darauf hin. |