

| Test Nr. | Aktion | Testobjekt | Methode | Erwartetes Ergebnis | Resultat |
|----------|------------------------------|------------------------------|---|---|----------|
| 101.1 | CREATE | Frontend (JS) | Add Button in Navigation öffnet Modal zur Erstellung => showModal(0) | Modal wird geöffnet mit Titel "New" | OK |
| 101.2 | | Backend (PHP) | Nach dem Absenden gibt die API eine success Response zurück => AJAX | Toast "success" im frontend | OK |
| 101.3 | | Frontend (JS) | Discard Button schliesst das Modal mit dem Formular | Das Formular wird geschlossen | OK |
| 102.1 | READ | Frontend (JS) | Die Daten werden beim Seitenaufruf automatisch geladen => getData() | Liste mit den aktuellen Daten | OK |
| 102.2 | | Frontend (JS) | Nach erfassen neuer Einträge werden die Daten neu geladen => getData() | Liste mit den aktuellisierten Daten wird geladen | OK |
| 102.3 | | Frontend (JS) | Limit Buttons limitiert die Data Table; Bei absenden von Daten bleibt die Limitierung bestehend => getData() | Die Liste wird limitiert und mittels Pagination kann navigiert werden. Nach einem Neuen Datensatz / Updaten wird die limiterung beibehalten | OK |
| 102.4 | | Backend (PHP) | Endpoint "action.php?action=get" gibt alle Daten aus | Es werden alle Daten zurückgegeben | OK |
| 102.5 | | Backend (PHP) | Endpoint "action.php?action=get&id=1" gibt Daten aus | Es wird nur den Datensatz für die Row 1 ausgegeben | OK |
| 103.1 | UPDATE | Frontend (JS) | Edit Button beim Datensatz öffnet das Modal mit den entsprechenden Daten prefilled | Im Titel ist "Edit ..." erkennbar; Die Daten werden korrekt übernommen | OK |
| 103.2 | | | Speichern sendet Request an API; Discard schliesst das Modal | Beim Speichern werden die Daten an die API übermittelt => Response; Close werden keine übermittelt | OK |
| 104.1 | DELETE | Frontend (JS) | Delete Button öffnet das Modal zum Löschen => showDeleteModal(id: int) | Das Modal wird geöffnet mit rotem Akzent zur Warnung vom Benutzer; Korrekte id ist im Modal erkennbar | OK |
| 104.2 | | Frontend (JS) | Löschen Button löscht den Datensatz | Der Datensatz wird gelöscht und mittels Toast bestätigt; Die Daten werden anschliessens Aktualisiert | OK |
| 104.3 | | | Cancel verhindert das Löschen | Das Modal wird geschlossen ohne Aktion | OK |
| 105.1 | Validation - CREATE & UPDATE | Frontend (JS); Backend (PHP) | Required Field: Name | Es wird ein Toast ausgegeben wenn nichts eingegeben wurde | OK |
| 105.2 | | Frontend (JS); Backend (PHP) | Required Field: Kaufdatum | Es wird ein Toast ausgegeben wenn nichts eingegeben wurde | OK |
| 105.3 | | Frontend (JS); Backend (PHP) | Field: Kaufdatum | Es wird auf ein gültiges Datum geprüft (YYYY-MM-DD) | OK |
| 105.4 | | Frontend (JS); Backend (PHP) | Required Field: Kategorie | Es wird ein Toast ausgegeben wenn nichts eingegeben wurde | OK |
| 105.5 | | Frontend (JS); Backend (PHP) | Field: Kaufpreis | Überprüfen ob die Eingabe nicht > 999.99; Kein Fehler soll ausgegeben werden | OK |
| 105.6 | | Frontend (JS); Backend (PHP) | Field: Rating | Überprüfen ob die Eingabe keine Komma zahl ist; Falls ja, soll ein Fehler ausgegeben werden | OK |
| 106.1 | Initialisierung | Frontend (JS); Backend (PHP) | DB wird erstellt beim ersten Aufruf; Gibt allfällige Fehler aus per JS Toast => initializeDB() | DB wird mit Strukur und 3 Beispiel Datensätze erstellt; Keine Fehler | OK |
| 106.2 | | Backend (PHP) | DB Connection wird geprüft beim API Aufruf | Die Connection wird geprüft; Kein Error wird ausgegeben | OK |