

Verifikation und Validierung Genetic Algorithms

Contents

Contents	1
1 Testplanung	2
1.1 Testumfang	2
1.1.1 Zu testende Funktionen	2
1.1.2 Testziele	2
1.2 Ressourcenplanung	2
1.2.1 Planung und Zuweisung von Ressourcen	2
1.3 Teststrategie	2
1.3.1 Entwicklung der Strategie	2
2 Testarten während der Entwicklung	3
2.1 Testmatrix	3
2.2 Testfälle während der Entwicklung	3
3 Testspezifikationen	6
3.1 Testfallentwurf	6
3.1.1 Test Cases für vertikalen Prototypen	6
3.2 Testdaten und -bedingungen	6
3.2.1 Testdaten	6
3.3 Test Design Techniken	6
4 Testimplementierung / -durchführung	7
4.1 Testumgebung	7
4.1.1 Betriebssysteme & Browser	7
4.1.2 Netzwerk	7
4.1.3 Einschränkungen	7
5 Auswertung / Bericht	7
5.1 Testabdeckungsanalyse	7
5.2 Abschlussbewertung	7

1 Testplanung

1.1 Testumfang

1.1.1 Zu testende Funktionen

Im Rahmen dieses Testprozesses werden die folgenden Kernfunktionen der Anwendung getestet:

- Ladeverhalten der Website – Sicherstellen, dass die Anwendung vollständig und ohne Fehler geladen wird.
- Navigationselemente – Überprüfung von allen Buttons auf ihre korrekte Funktionsweise.
- Benutzer-Authentifizierung – Prüfung der Login-Funktionalität.
- Benutzereinstellungen - Überprüfung, ob Benutzereinstellungen korrekt gespeichert und global aktiv sind.
- Rollenfunktion - Überprüfung der korrekten Berechtigungen für die verschiedenen Benutzerrollen.
- Datenbank - Prüfung der korrekten Datenspeicherung und -verarbeitung.

1.1.2 Testziele

Das Ziel der Tests ist es, die grundlegenden Funktionen der Webanwendung unter verschiedenen Bedingungen zu verifizieren und sicherzustellen, dass:

- alle Kernfunktionen fehlerfrei ausgeführt werden,
- die Website in verschiedenen Browsern korrekt läuft,
- der Login-Prozess zuverlässig funktioniert,
- Rollenrechte richtig verteilt und angezeigt werden.

Diese Tests sollen sicherstellen, dass der Testprozess die Anforderungen abdeckt und die Software den gewünschten Qualitätsstandards genügt.

1.2 Ressourcenplanung

1.2.1 Planung und Zuweisung von Ressourcen

Für den Testprozess sind folgende Ressourcen eingeplant und zugewiesen:

- Testteam: Ein Team von 2 Testern wird für die Durchführung der Tests verantwortlich sein.
- Testsysteme: Die Tests werden auf Windows 11 und macOS 14 durchgeführt.
- Software: Aktuelle Versionen der Browser Google Chrome, Mozilla Firefox und Safari sind auf den jeweiligen Betriebssystemen installiert.

1.3 Teststrategie

1.3.1 Entwicklung der Strategie

- Testansätze:
 - Manuelle Tests: Die Benutzeroberfläche und die Navigation der Webanwendung werden manuell getestet, um sicherzustellen, dass alle Benutzerinteraktionen wie erwartet funktionieren.
 - Automatisierte Tests: Die Benutzeroberfläche und Navigation der Webanwendung werden automatisiert getestet, um größere Benutzerinteraktionsszenarien effizient zu überprüfen.
- Testtechniken
 - Regressions-Tests – Sicherstellen, dass bereits getestete Funktionen nach Änderungen weiterhin fehlerfrei arbeiten.
- Testmethoden:
 - Black-Box-Tests: Die Funktionalität der Anwendung wird getestet, ohne dass ein Blick auf den internen Code geworfen wird. Dies gewährleistet, dass die Anwendung aus Benutzersicht funktioniert.
 - White-Box-Tests: Falls erforderlich, werden auch Teile des Quellcodes betrachtet, um sicherzustellen, dass die Implementierung den festgelegten Spezifikationen entspricht.
- Teststufen:
 - Systemtests: Die gesamte Anwendung wird als System getestet, um zu verifizieren, dass sie wie gewünscht funktioniert.

2 Testarten während der Entwicklung

Während der Entwicklung werden unabhängig kontinuierlich Tests durchgeführt, um Fehler frühzeitig zu identifizieren. Dies umfasst sowohl Backend- als auch Frontend-Tests.

2.1 Testmatrix

Testart	Backend	Frontend
Unit-Tests	API-Endpunkte, Datenbanklogik	Einzelne UI-Komponenten
Integrationstests	API↔Datenbank, <i>API</i> ↔ <i>Frontend</i>	Datenfluss zwischen Komponenten
Manuelle Tests	API-Aufrufe mit Postman prüfen	UI-Elemente auf Funktionalität testen
End-to-End-Tests	-	Gesamtanwendung mit echten Nutzeraktionen testen

2.2 Testfälle während der Entwicklung

Table 1: Backend - Testfälle während der Entwicklung

Testfall-ID	Beschreibung	Erwartetes Ergebnis	Status	Gefundener Fehler	Datum	Bemerkung
TC-BE-001	Überprüfen ob hinzugefügte Testnutzerdaten korrekt gespeichert und verarbeitet werden.	Es ist durch verschiedene Funktionen möglich User hinzuzufügen, ihre Existenz zu überprüfen, Passwort und Role zu checken	✓	checkpasswort und getrole fehlerhaft	08.03.2025	adduser, userexist fine - Passwort und Role vielleicht vertausch.
TC-BE-002	checkpasswort und getrole durch hinzufügen eines bool'schen Rückgabewertes bei adduser fixen.	Durch Rückgabewerte sollte der Datenbank klar sein, ob der Nutzernamen schon existiert, oder erfolgreich hinzugefügt wurde	✓	-	08.03.2025	PW und Role tatsächlich vertauscht, weil die Datenbank nicht gelöscht und so Nutzer nicht korrekt hinzugefügt wurden. Deshalb gab adduser keinen Rückgabewert zurück.
TC-BE-003	hashingCheck	Hashing wird korrekt durchgeführt.	✓	-	-	-
TC-BE-004	Überprüfen, ob Verbindung korrekt getrennt wird	Verbindung wird korrekt getrennt.	✓	-	-	-

TC-BE-003	Durch hinzufügen von 5000 Testusern (mit Dopplungen) wird getestet, ob redundante Userdaten abgelehnt werden.	Die User-Funktionen sollten nach wie vor korrekt funktionieren.	✗	Ab Datensatz 900 ist ein Fehler aufgetreten. Checkpasswort gibt Fehlermeldung aus.	-	Nach Testlauf haben jedoch Daten korrekte Form.
-----------	---	---	---	--	---	---

Table 2: Frontend - Testfälle während der Entwicklung

Testfall-ID	Beschreibung	Erwartetes Ergebnis	Status	Gefundener Fehler	Datum	Bemerkung
TC-FE-001	Überprüfung, ob die Website durch Klick auf den Link geöffnet wird	Die Seite wird korrekt und vollständig geladen, ohne Fehler.	✓	-	04.03.2025	-
TC-FE-002	Login-Button klicken	Nach Klicken des Buttons sollte sich die Login Seite öffnen	✓	-	04.03.2025	
TC-FE-003	Data Visualization-Button klicken	Nach Klicken des Buttons sollte sich die Data Visualizations Seite öffnen	✓	-	04.03.2025	Border um Home Button.
TC-FE-004	Settings-Button klicken	Nach Klicken Buttons sollte sich die Settings Seite öffnen.	✓	-	04.03.2025	Home Button wird in Bar blau angezeigt.
TC-FE-005	Proceed to Simulation-Button klicken	Nach Klicken Buttons sollte sich die Data Visualization Seite öffnen.	✓	-	04.03.2025	
TC-FE-006	Home-Button klicken	Nach Klicken des Buttons sollte sich die Home Seite öffnen.	✓	-	04.03.2025	
TC-FE-007	Überprüfen, ob das "Passwort"-Feld im Login-Formular anklickbar ist und Text eingegeben werden kann.	Nach dem Klicken auf das "Passwort"-Feld sollte es möglich sein, ein Passwort einzugeben. Der eingegebene Text sollte maskiert sein.	✓	-	06.03.2025	-
TC-FE-008	Überprüfen, ob das "Username"-Feld im Login-Formular anklickbar ist und Text eingegeben werden kann.	Nach dem Klicken auf das "Username"-Feld sollte es möglich sein, Text einzugeben. Der eingegebene Text sollte im Feld sichtbar sein.	✓	-	06.03.2025	-

TC-FE-009	Account Login	Nach Eingaben der Daten und drücken des Login-Buttons sollte der User eingeloggt und auf die Datenvisualisierungsseite weitergeleitet werden.	✗	Es passiert nichts.	06.03.2025	Wahrscheinlich Fehler in der Datenbank bei der Datensicherung.
TC-FE-010	Account Login	Nach Eingaben der Daten und drücken des Login-Buttons sollte der User eingeloggt und auf die Datenvisualisierungsseite weitergeleitet werden.	✗	Login funktioniert, aber man wird nicht weitergeleitet und bleibt auf der Login Seite.	06.03.2025	Login erfolgreich, da in der Navigationsbar "Login" zu "Logout" wurde
TC-FE-011	Account Information-Button klicken	Nach Klicken des Buttons sollten bei eingeloggten Usern der Username sichtbar werden.	✓	-	04.03.2025	
TC-FE-012	FAQ buttons anklicken	Nach dem Klicken der einzelnen Fragen, sollten sich die jeweiligen Antworten unter den Fragen aufklappen	-	-	-	-
TC-FE-013	Auswahl Generations buttons anklicken	Nach dem Klicken des Buttons sollte sich ein Dropdown-Menü öffnen und alle 10 Generationen sichtbar/auswählbar werden.	✗	Menü öffnet sich, aber Generationen nicht anklickbar.	-	Angemeldete User haben noch alte Namenskonvention.
TC-FE-014	Logout Button anklicken	User wird ausgeloggt und verbleibt auf der aktuellen Seite.	✓	-	05.03.2025	-
TC-FE-015	Continue as Simulator Button in der Login Seite anklicken	User sollte als Gast auf die Data Visualizations Seite weitergeleitet werden.	✗	Weiterleitung funktioniert, jedoch wird der User als eingeloggt angezeigt.	05.03.2025	Login Button wird zu Logout Button.
TC-FE-016	Julius haut auf die Tasten, um zu überprüfen, ob der Server bei zu schneller Eingabe abstürzt.	Eingabe wird erkannt und übernommen.	✓	-	11.03.2025	-
TC-FE-017	Werte für die Daten Visualisierung eingeben und auf Start Simulation klicken.	Datenart und zugehöriger eingegebener Wert sollte erscheinen	✗	Anstatt Mutation-rate wird AEP ausgegeben	11.03.2025	-

TC-FE-018	Werte für die Daten Visualisierung eingeben und wieder rauslöschen.	Bei falscher Eingabe sollte es möglich sein Werte wieder komplett rauszulöschen.	✗	bei 4/7 kann man den letzten Stellenwert nach Eingabe nicht mehr rauslöschen.	11.03.2025	-
TC-FE-019	Werte für die Daten Visualisierung eingeben und wieder rauslöschen.	Bei falscher Eingabe sollte es möglich sein Werte wieder komplett rauszulöschen.	✓	-	11.03.2025	kurze Fehlerbehebung im Input.
TC-FE-020	Extremwerte für die Daten Visualisierung eingeben.	Sytem sollte zu große Werte nicht zulassen.	✗	Eingabe hat keinen cap, weshalb die Seite beim laden der Extremwerte sehr fehlerhaft ist - zerstörte Größenverhältnisse und Blackscreen.	11.03.2025	Eingabe einfach cappen
TC-FE-021	Extremwerte für die Logindaten eingeben.	Sytem sollte zu große Werte nicht zulassen.	✗	Eingabe hat keinen cap, weshalb die Seite beim laden der Extremwerte sehr fehlerhaft ist - zerstörte Größenverhältnisse und Blackscreen.	11.03.2025	Eingabe einfach cappen
TC-FE-022	Admins sollten über die Homepage über den Proceed to Simulation Button zur Datenvisualisierungsseite weitergeleitet werden.	Dies sollte passieren, ohne die Rolle zu ändern.	✗	Admins verlieren ihre Rolle und werden zu Simulatoren.	12.03.2025	Wurde schnell in 2 Minuten gefixt.

3 Testspezifikationen

3.1 Testfallentwurf

3.1.1 Test Cases für vertikalen Prototypen

Bei den Test Cases für das Endprodukt haben wir uns dazu entschieden, diese mit Hilfe von Selenium als automatisierte Tests zu schreiben. Davor wurde jeder Button und jedes Eingabefeld manuell getestet und auf die richtige Funktionalität geprüft. Genauer zu den genutzten Funktionen und erstellten Test Cases befindet sich auf GitHub.

3.2 Testdaten und -bedingungen

3.2.1 Testdaten

- Als Testdaten wurden ausschließlich von uns selbst erstellte Daten genutzt.
- Die Daten zum Testen waren Dateien.

3.3 Test Design Techniken

- Äquivalenzklassenpartitionierung: Eingabewerte werden in gültige und ungültige Äquivalenzklassen unterteilt, um sicherzustellen, dass alle wichtigen Eingabebereiche abgedeckt sind.
- Grenzwertanalyse: Tests werden auf Basis von Grenzwerten für Eingabefelder durchgeführt, um sicherzustellen, dass die Anwendung mit den Extremwerten korrekt umgeht.

4 Testimplementierung / -durchführung

4.1 Testumgebung

4.1.1 Betriebssysteme & Browser

Die Tests wurden unter folgenden Bedingungen durchgeführt:

- Windows 11
 - Browser: Google Chrome (Version 122.0), Mozilla Firefox (Version 124.0)
- macOS 14
 - Browser: Safari (Version 17.3), Google Chrome (Version 122.0)

4.1.2 Netzwerk

- Testumgebung: Staging-Server `staging.example.com`

4.1.3 Einschränkungen

- Keine Tests auf Internet Explorer durchgeführt.
- Tests nur mit simulierten Nutzerkonten, keine realen Kundendaten.

5 Auswertung / Bericht

5.1 Testabdeckungsanalyse

- Es wurden alle zuvor festgelegten funktionalen- und nicht-funktionalen Bedingungen getestet.
- Jeder Code-Abschnitt wurde direkt auf richtige Funktionalität manuell getestet.

5.2 Abschlussbewertung

- Alle Funktionen wurden getestet und sind voll funktionsfähig.
- Die frühzeitigen Tests haben uns dabei geholfen, Fehler frühzeitig zu erkennen und direkt in der Entwicklung auszubessern. Durch gefundene Fehler kamen dann während der Arbeit immer mal wieder Tests hinzu, die nötig waren, um die Verbesserung festzustellen.