GUI-Tests einer Webseite zur Registrierung von Durchflussmessgeräten mittels Protractor

Projektarbeit

an der

Hochschule für Technik und Wirtschaft
Fachbereich Wirtschaftswissenschaften II
Studiengang Angewandte Informatik

von

Monika Drößler s0547952 und Elena Gillich s0535893

28.03.2016

Inhaltsverzeichnis

Abbildungsverzeichnis

Worum geht es?

In einem Projektstudium wurde eine Webseite zur Registrierung von Durchflussmessgeräten auf Mitarbeiter verschiedener Unternehmen entwickelt. Ziel dieser Arbeit ist es, diese Webseite einigen GUI-Tests zu unterziehen. Dabei soll ein weitestgehend natürliches Nutzerverhalten mittels programmatisch implementierter Tests bei verschiedenen Interaktionen imitiert werden.

Diese Tests sollen sowohl positives als auch negatives Verhalten des Nutzers abdecken und die entsprechende Resonanz der Webseite auf ihre Korrektheit prüfen.

Warum Protractor?

Da es sich bei der zu prüfenden Anwendung um eine Webseite handelt, bietet sich als Testumgebung Selenium an. Dieses Tool wurde speziell für automatisiertes Testen von Webanwendungen entwickelt und basiert rein auf HTML und JavaScript.

Selenium lässt sich zum einen als Addon für Firefox verwenden, bei welchem ein manueller Test der Webseite aufgezeichnet und im Anschluss mehrfach automatisiert vom Addon wiederholt werden kann. Eine weitere Möglichkeit der Verwendung Seleniums ist die programmatische Implementierung und darauf folgende Ausführung von Tests.

Die in dieser Arbeit verwendete Anwendungsmöglichkeit ist die Nutzung Seleniums als Basis und WebDriver für ein spezielleres Testframework. Dieses speziell auf das, auf der Webseite genutzte, JavaScript Framework AngularJS abgestimmte Tool heißt Protractor und bietet vielerlei Vorteile. Die wohl ausschlaggebendsten Vorteile Protractors sind die unmittelbare Interaktion und Synchronisation mit der AngularJS Anwendung und die Nutzung weniger verschiedener Vorkenntnisse. Dies ermöglicht bereits den Frontend-Entwicklern die Implementierung einiger GUI-Tests.

Wie wird vorgegangen?

Zunächst muss ein Framework für die GUI-Tests ausgewählt werden. Zur Auswahl stehen dabei Selenium und Protractor. Da auf der Webseite AngularJS verwendet wird, wird zum Testen letztendlich Protractor verwendet.

Nach der Einigung auf eine Testumgebung werden die Testfälle entwickelt. Dabei wird die Seite zunächst manuell untersucht und mögliche Interaktionen notiert. Mithilfe der Notizen und einer genaueren Beobachtung während erneuter manueller Tests, werden die Diagramme zur detaillierteren Beschreibung der Testfälle entwickelt.

Da bereits im Projektstudium einige Testfälle überprüft wurden, sollen in dieser Arbeit hauptsächlich speziellere und negative Tests durchlaufen werden.

Im nächsten Schritt wird die Testumgebung eingerichtet. Dazu muss aufgrund der Nichterreichbarkeit der Webseite eine lokale Kopie der Webseite eingerichtet werden, indem der Quellcode aus dem Git Verzeichnis gezogen wird und NodeJS, NPM und Bower installiert werden. Um die Webseite lokal ausführbar zu machen, muss diese nun mittels des Konsolenbefehls *grunt serve* durch Grunt gebaut und gestartet werden.

Nach Einrichtung der Webseite muss Protractor installiert und der Selenium WebDriver aktualisiert werden. Da der Selenium WebDriver in der Protractor Installation bereits vorhanden ist, ist eine separate Installation Seleniums nicht notwendig.

Je nach Entwicklungsprogramm kann eine Unterstützung für die Protractor-Entwicklung, wie bei der IntelliJ IDEA, installiert werden, die die Entwicklung der GUI-Tests erleichtern soll.

Die Implementierung der Tests erfolgt auf Grundlage der zuvor erstellten Testfalldiagramme. Es werden explizit die definierten Tests umgesetzt und auf die entsprechend benötigten Objekte zugegriffen. Weitergehende Zugriffe auf die Websiete erfolgen in dieser Arbeit nicht.

Um die GUI-Tests nach der Implementierung zu starten, wird zunächst der Selenium WebDriver mithilfe des Konsolenbefehls *webdriver-manager start* gestartet. Danach werden mit dem Befehl *protractor conf.js* die Tests ausgeführt.

Bereits auf der Konsole gibt Protractor das Ergebnis der GUI-Tests zurück. Korrekt verlaufene Tests werden mit einem Punkt, inkorrekt verlaufene Tests mit einem F gekennzeichnet. Sollte ein Test fehlgeschlagen sein, so wird ein Hinweis auf die fehlerbehaftete Prüfung in der Konsole ausgegeben.

Welche Testfälle sollen abgedeckt werden?

In dieser Arbeit werden verschiedene Testfälle bearbeitet. Neben einem bereits im Projektstudium durchgeführten positiven Test des Logins, erfolgen hier Tests bezüglich der Meldungen bei fehlerhafter oder gar fehlender Eingabe der Login-Daten. Zudem werden Fehlermeldungen und Link-Verbindungen der Formulare bezüglich eines vergessenem Passworts und der Registrierung eines neuen Nutzers geprüft.

Weitere Testfälle behandeln den Logout und das Löschen von registrierten Geräten sowie des Nutzers.

Mittels Testfalldiagrammen werden die zu implementierenden GUI-Tests genauer definiert.

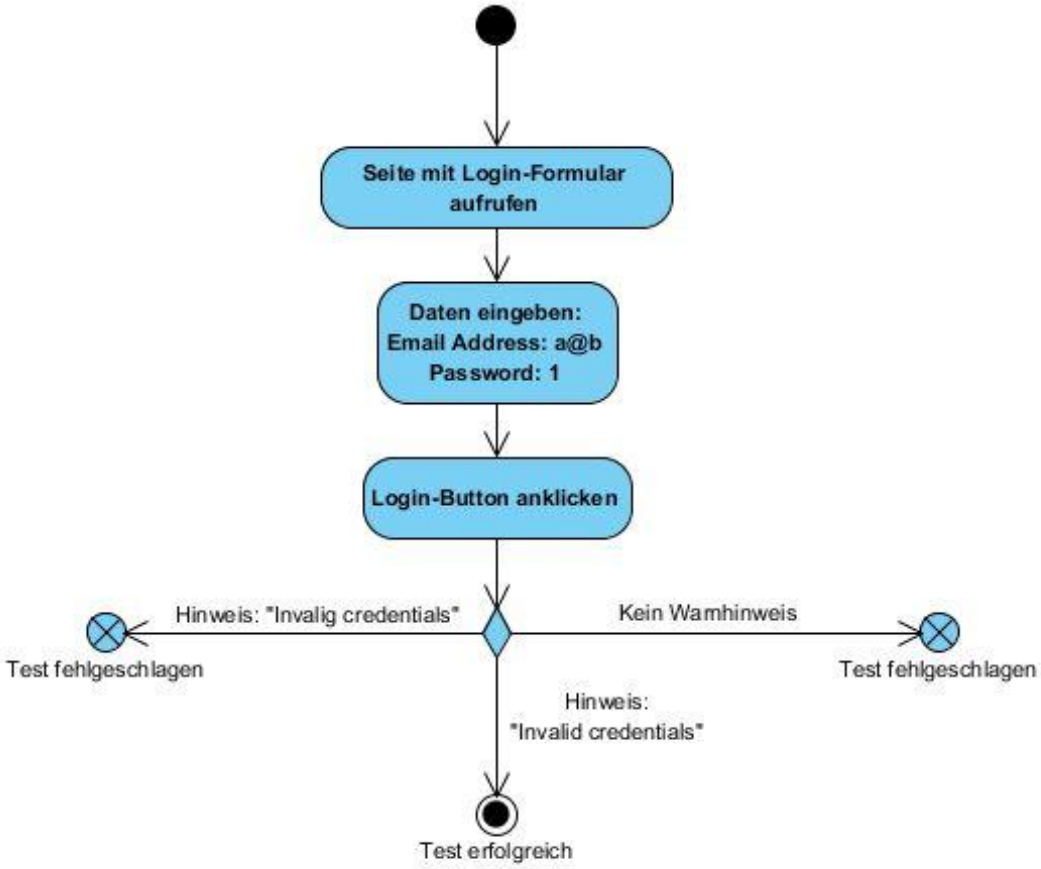

Abbildung 1: Testfall 1 - Login mit fehlerhaften Daten

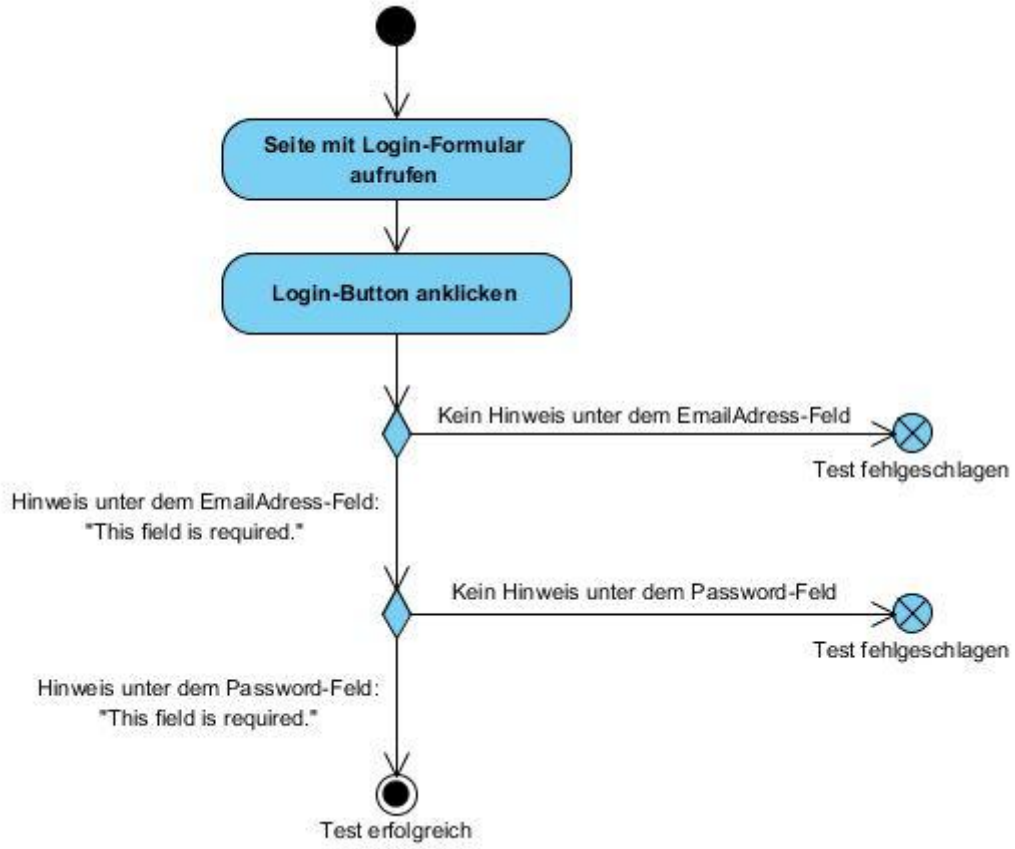

Abbildung 2: Testfälle 2, 4 - Login ohne Email bzw. Passwort

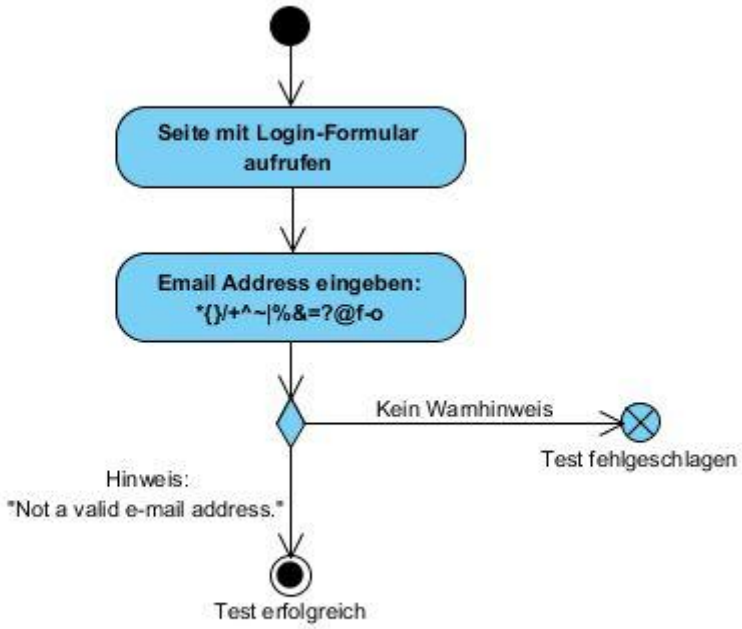

Abbildung 3: Testfall 3 - Login mit ungültiger Email

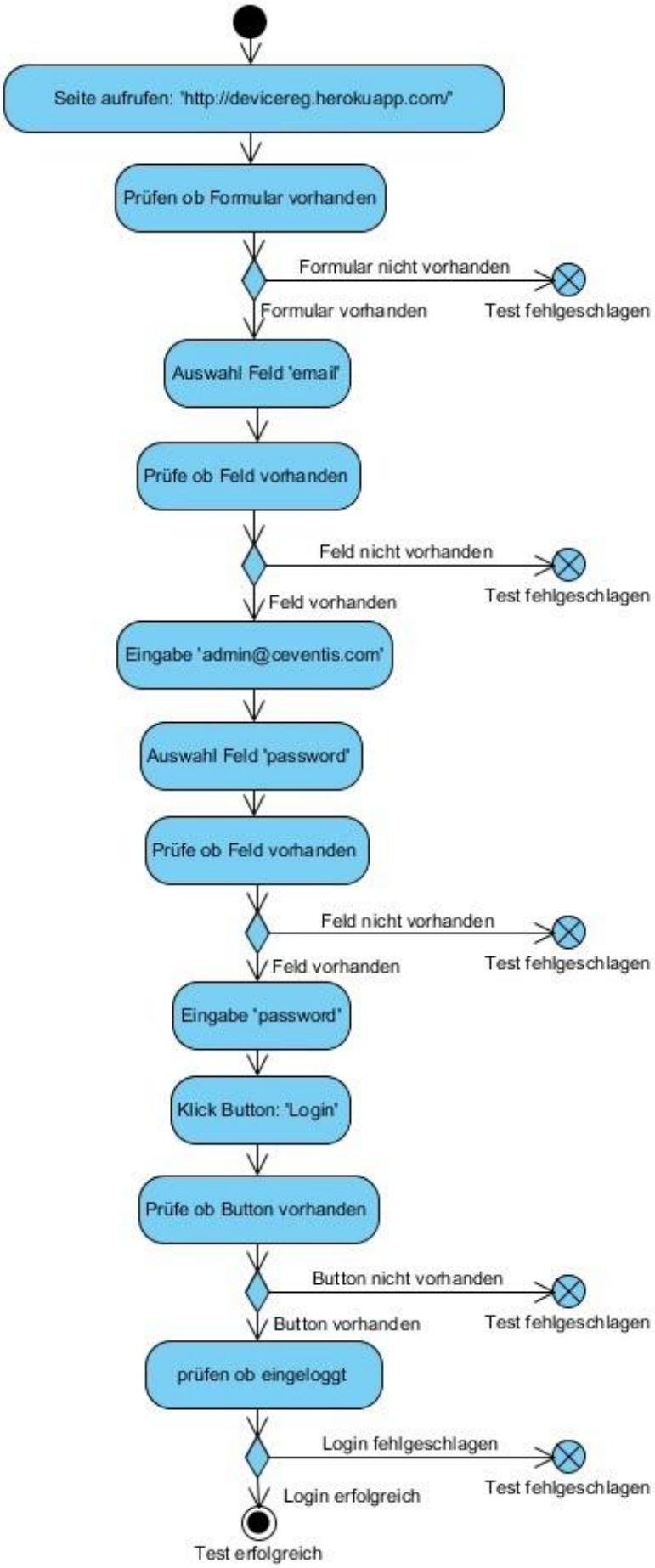

Abbildung 4: Testfall 5 - erfolgreicher Login

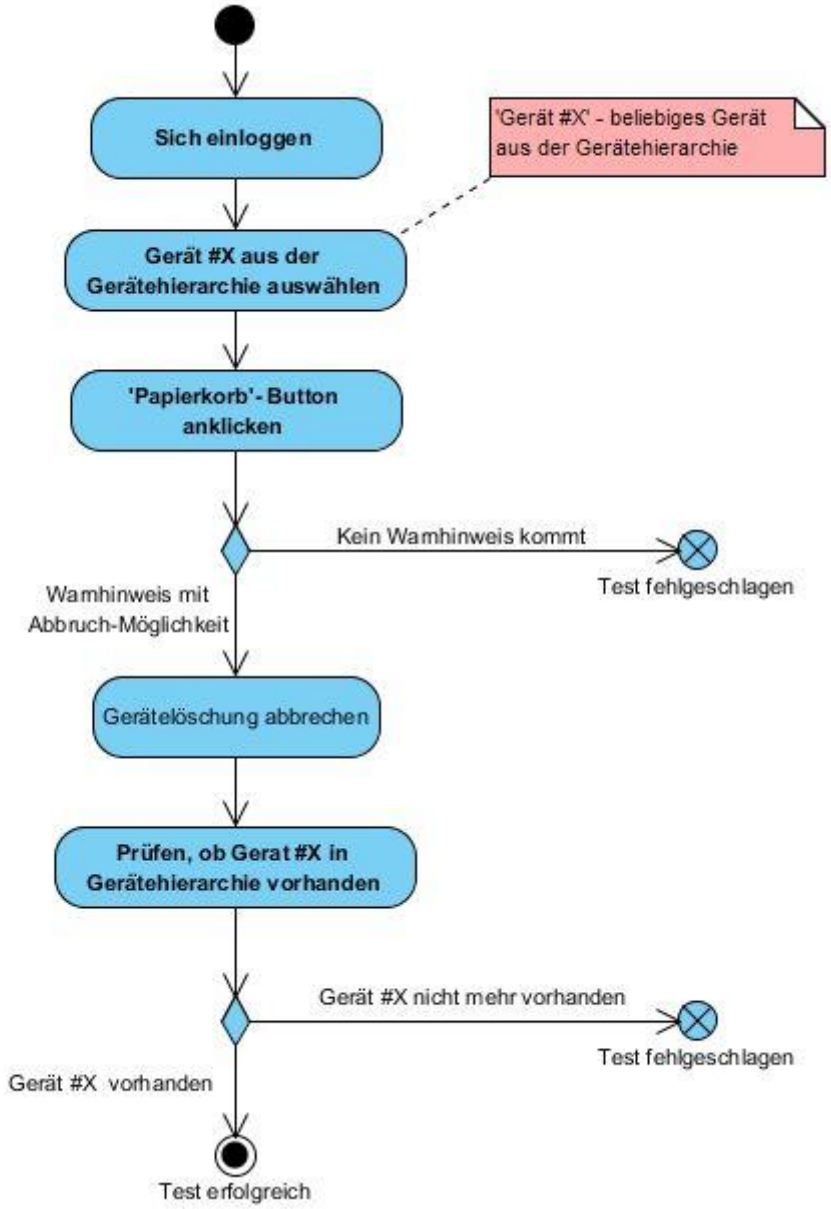

Abbildung 5: Testfall 6 - Löschen eines Gerätes

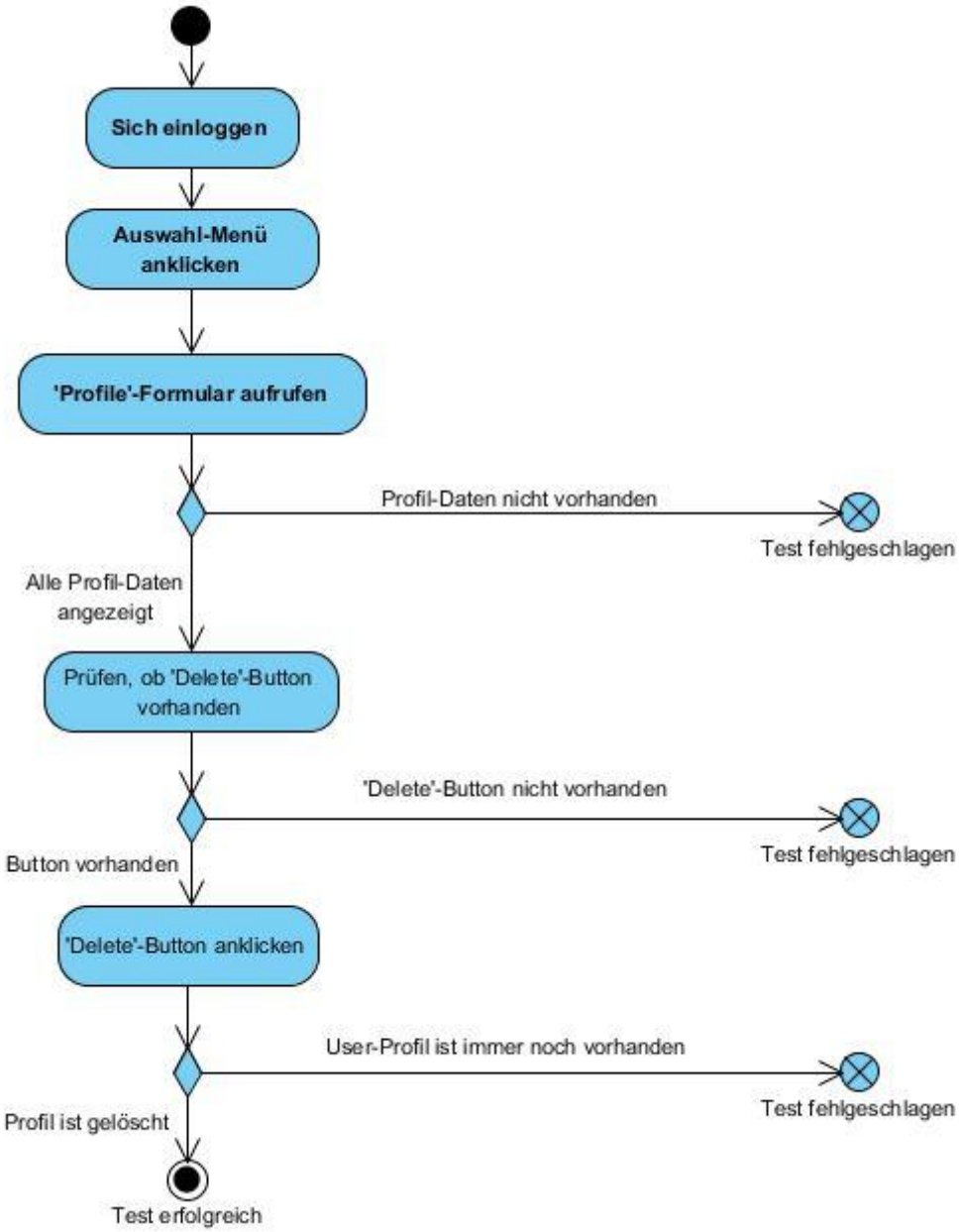

Abbildung 6: Testfall 7 - Löschen des Nutzers

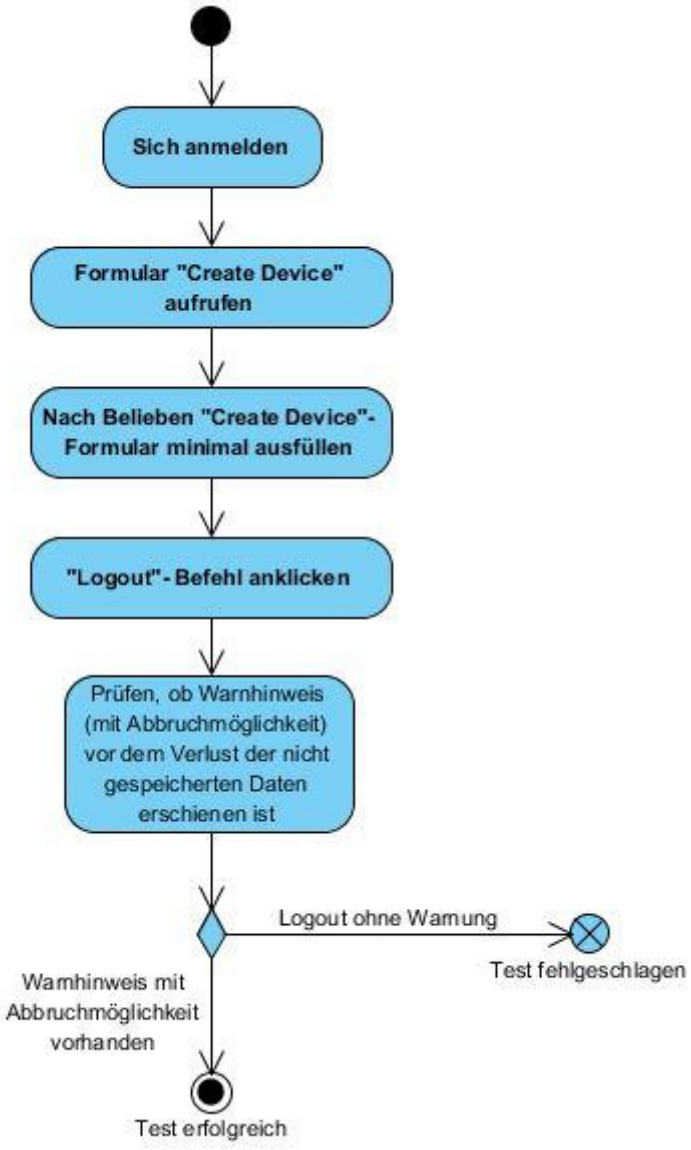

Abbildung 7: Testfall 8 - Logout während Geräteregistrierung

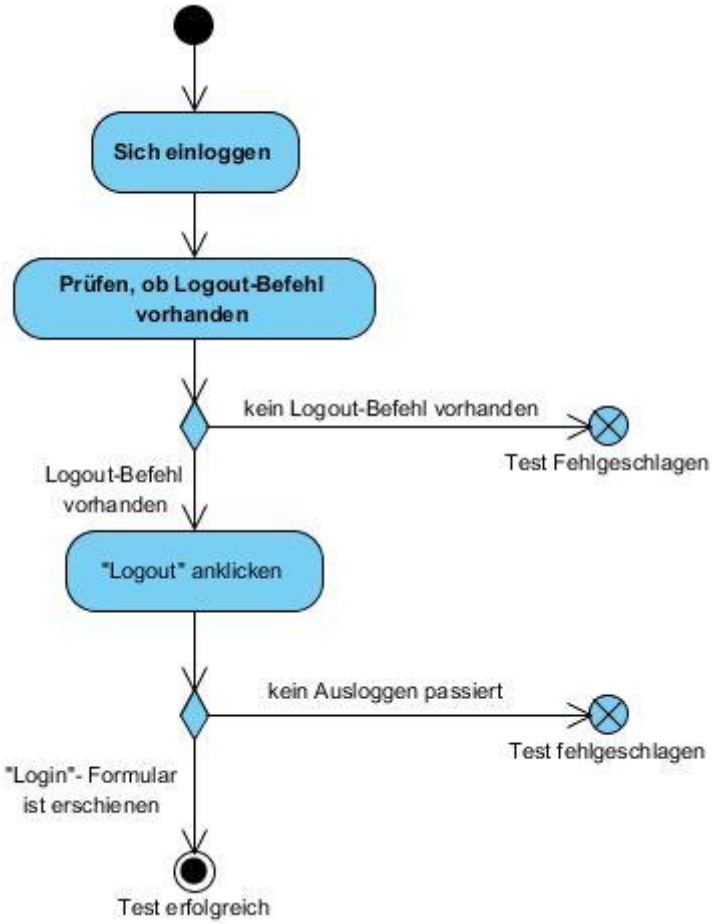

Abbildung 8: Testfall 9 - erfolgreicher Logout

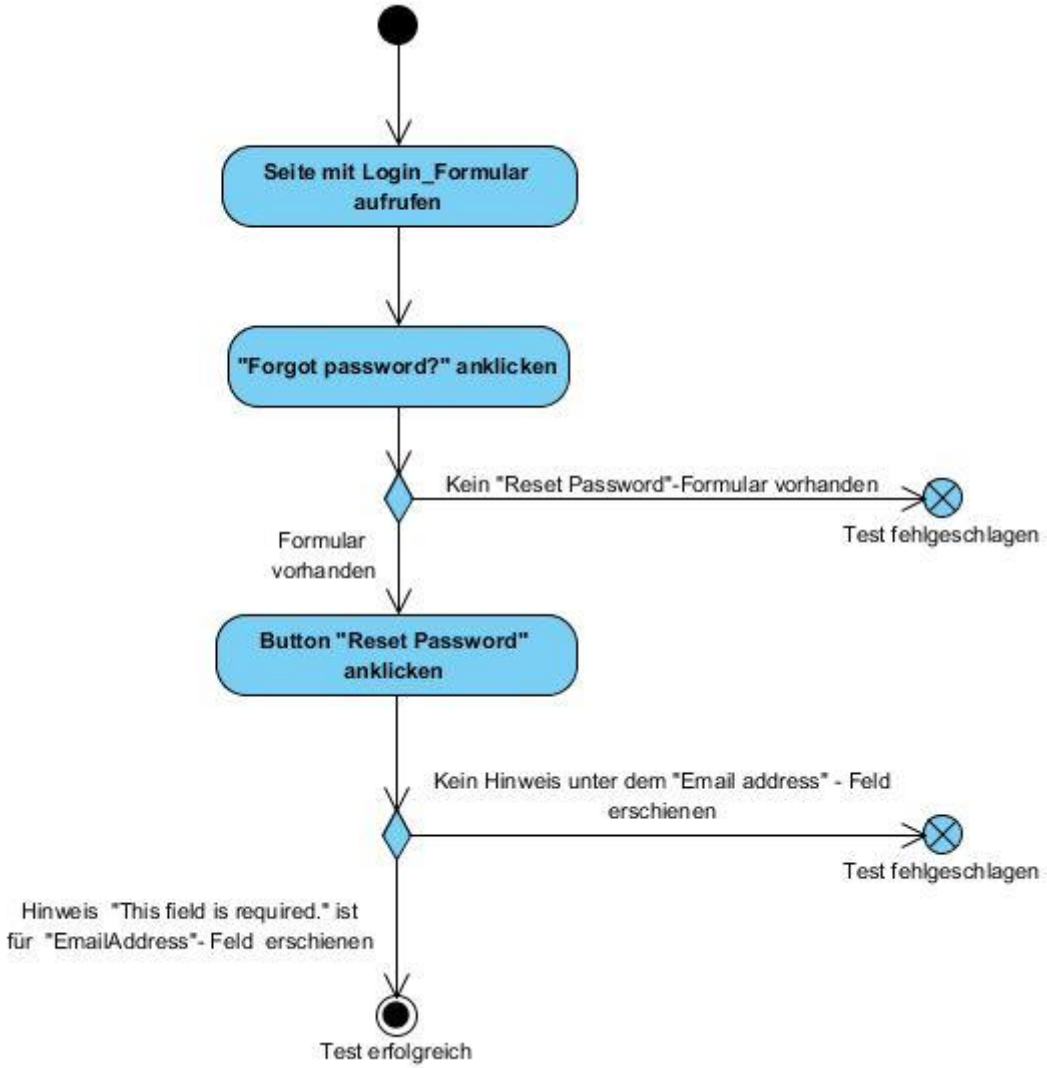

Abbildung 9: Testfall 10 - Passwort zurücksetzen ohne Email

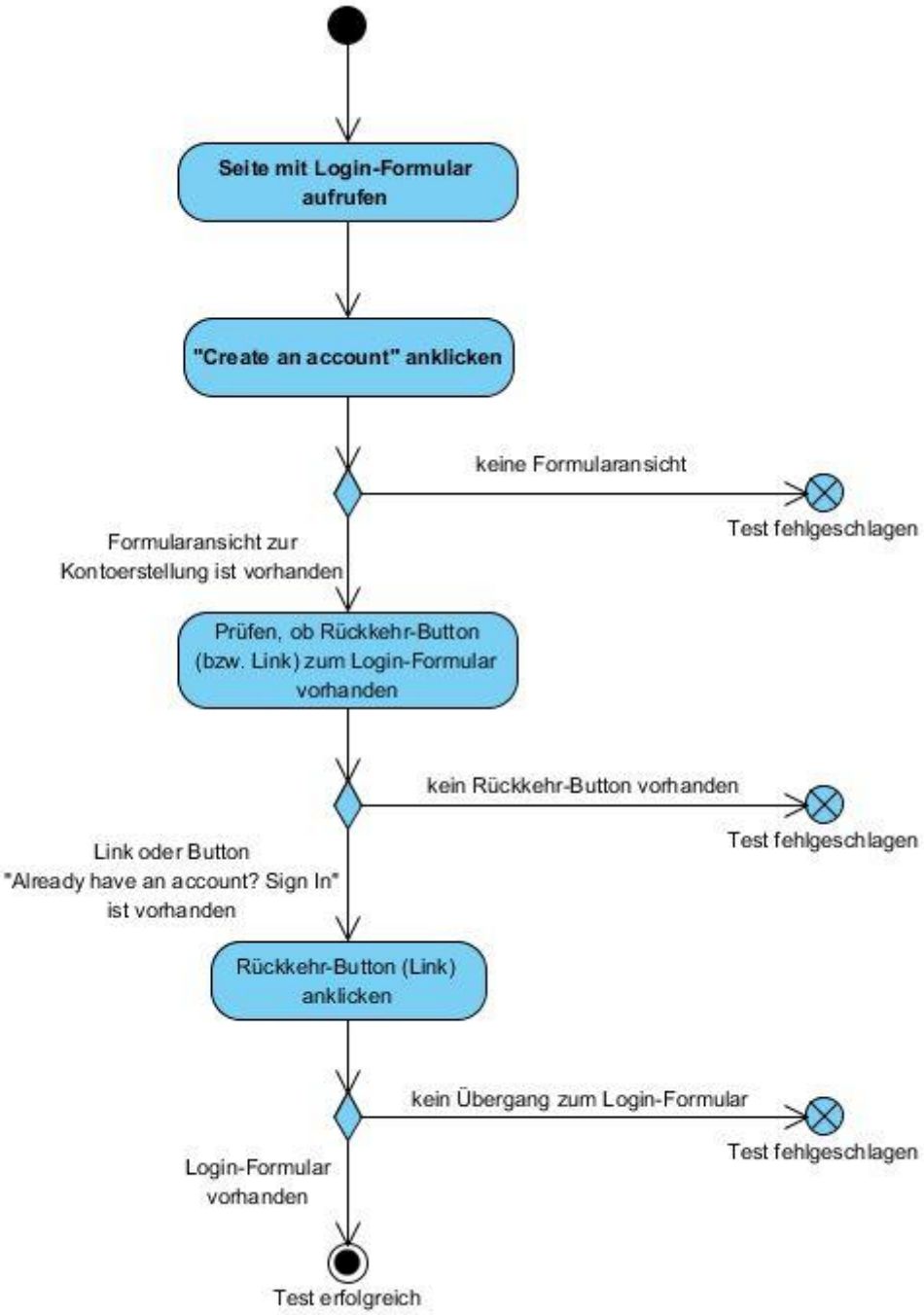

Abbildung 10: Testfall 11 - Rückker zum Login von Registrierungsformular

Wie wird implementiert?

Bei der Implementierung der Tests wird darauf geachtet, die Objekte der jeweiligen Unterseiten der Webseite mit entsprechenden Methoden von den Zusammenstellungen der Tests zu trennen. Dazu werden zunächst für jede Unterseite die, für die GUI-Tests benötigten, Objekte in jeweils einer Datei gesammelt. Zudem werden Methoden implementiert, um diese Objekte manipulieren zu können.

Mittels dieser Objekte und Methoden werden daraufhin die Tests zusammengestellt und an ihre Aufgabe angepasst. Zuletzt werden die Aufgaben der einzelnen Tests beschrieben und Abgleiche mit den erwarteten Ergebnissen durchgeführt. Ein Test kann dabei mehrere Abgleiche beinhalten, sofern dies von Nöten ist.

Um die Tests ausführbar zu machen, werden diese mittels einer Konfigurationsdatei zur Verfügung gestellt. In dieser Datei werden ebenso die Adresse für den Selenium WebDriver, die zu verwendenden Browser und eventuelle weitere Konfigurationsparameter definiert.

Sobald alle GUI-Tests implementiert und die Konfigurationen definiert sind, kann ein Testlauf über die Konsole gestartet werden.

Was waren die Problematiken?

Während der Umsetzung dieser Arbeit traten einige Problematiken auf, die zum Teil gelöst werden konnten.

Das zunächst grundlegendste Problem stellte die Webseite der Geräteregistrierung dar. Da die Live-Version im Internet nicht funktioniert, musste die Webseite lokal zur Verfügung gestellt werden. Durch Zugriff auf das Git-Verzeichnis bereits aus dem Projektstudium, war diese Lösung recht schnell umzusetzen.

Ein weiteres Problem stellte der Funktionsumfang der Webseite dar. Da die Webseite noch in der Entwicklung ist, ist der Umfang, der verfügbaren Funktionen, eingeschränkt. Somit konnten recht wenige Nutzerinteraktionen für die GUI-Tests berücksichtigt werden.

Da die Webseite mit dem Backend aus einem anderen Projektstudium gekoppelt ist, welches stetig weiter entwickelt wird und auf einem externen Server liegt, kam es während der Erstellung der Testfälle und den Ausführungen der GUI-Tests öfters zu Problemen und Stillständen. Der Server war während der gesamten Planungs- und Entwicklungszeit mehrfach und über längere Zeit nicht erreichbar und verhinderte somit einen reibungslosen Ablauf.

Die Änderungen im Backend verursachten zudem ein Problem mit dem Testfall bezüglich der Löschung eines registrierten Gerätes. Ein fehlerfreier Verlauf dieses Tests kann aufgrund fehlender Datenbankeingaben nicht garantiert werden.

Fazit

Während der Planung und Erstellung der Testfalldiagramme lag ein eingeschränkter Funktionsumfang der Webseite vor, weshalb das Hauptaugenmerk der Testfälle auf spezifischere Funktionen gelegt wurde. Zudem musste die Verwendung der Live-Version der Webseite aufgrund ihrer Nichtfunktion überdacht und eine Alternative gefunden werden, was einen größeren Arbeitsaufwand zur Ausführung der GUI-Tests zur Folge hat. Da am Backend der Webseite weiterhin gearbeitet wird, kam es zu Komplikationen und die Entwicklung der Tests geriet von Zeit zu Zeit ins Stocken. Dies ist für die Entwicklung korrekter GUI-Tests von Nachteil und sollte künftig unterbunden werden.
Trotz der Schwierigkeiten war es möglich GUI-Tests zu implementieren und diese Arbeit zu vollenden.