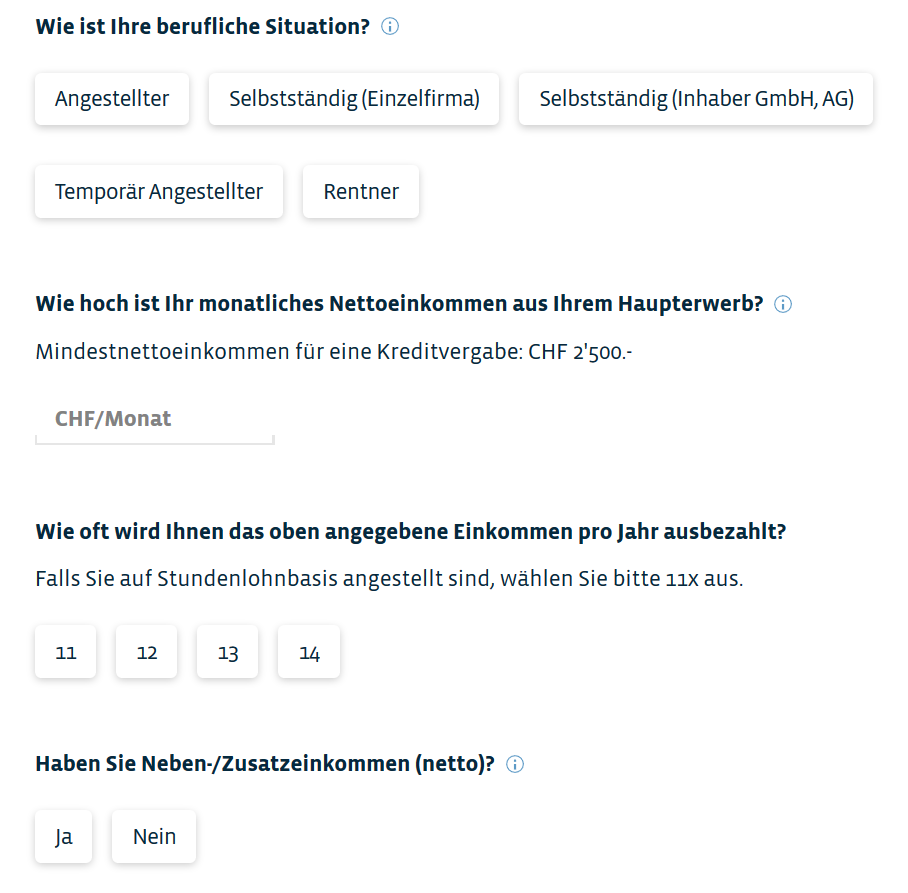
# Realisierungskonzept

Das Programm wird aus der Hauptklasse «App.java» und der Klasse «SeleniumHandler.java» bestehen. Das Formular welches es testen soll ist auf einer URL im Cembra Intranet ersichtlich. Das Formular wurde mit «React» erstellt und ist Dynamisch aber hat über HTML ersichtliche Tags für die Eingaben und Buttons. Durch die Verwendung von Selenium können auf diese Tags zugegriffen werden um Interaktionen zu automatisieren. Zuerst werden die Tags auf den ersten fünf Formular Seiten ermittelt und entsprechend der Struktur des Formulars sollen die automatischen Handlungen im Programm nachgebildet werden.

Vor und nach der Betätigung jedes Buttons im Formular werden Screenshots erstellt um die Eingaben welche gemacht worden sind zu überprüfend und mögliche Fehlermeldung einzusehen. Diese werden an einem spezifizierten Ort gespeichert und nach Zeitpunkt und Reihenfolge in welcher die Screenshots gemacht wurden sortiert.

In der «SeleniumHandler.java» Klasse werden Methoden erstellt um sich repetierenden Code möglichst gering zu halten. Spezifisch angepasste Methoden werden erstellt um die Buttons oder Eingabe Felder anzusprechen welche keine ID haben. Bei solchen Fällen muss das ganze über einen Array geregelt werden weshalb eine Methode dafür gelegen kommt.

Um überhaupt auf die Seite zuzugreifen oder die Kontrollen über einen Browser zu haben wird ein WebDriver benötigt. Ich werde das Ganze mit Google Chrome realisieren weshalb ich den Chrome Driver Verwenden werde.

Selenium stellt eine Schnittstelle zwischen dem Browser und meinem Java Code dar. Durch Selenium kann ich direkt den WebDriver von Chrome steuern und die Inhalte auf einer Webseite einlesen und ansprechen. In diesem Fall werden diese Option hier welche Textfelder darstellen angesprochen und ihnen wird ein gewisses Attribut gegeben wodurch

# Testkonzept

|  |  |
| --- | --- |
| Testfall-Nr. 1 | |
| Anforderung: | Selenium ist korrekt installiert |
| Beschreibung: | Selenium muss korrekt installiert sein damit die entsprechenden «Import» Anweisungen funktionieren. |
| Testvorgang: | Im Source Code zeigen die «Import» Anweisung keine Fehler an. Und die «JAR» Dateien von Selenium sind unter der Library ersichtlich |
| Erwartetes Resultat: | Selenium ist korrekt installiert und die «Import» Anweisung sind entsprechend fehlerfrei. |

|  |  |
| --- | --- |
| Testfall-Nr. 2 | |
| Anforderung: | Chrome Driver ist korrekt installiert |
| Beschreibung: | Der Chrome Driver wird benötigt um den Browser zu steuern weshalb die Installation dessen und seine Verwendung im Code korrekt sein müssen |
| Testvorgang: | Der Driver wird installiert und an einem Ort abgelegt. Der Driver wird im Code definiert und die URL zum Formular geöffnet |
| Erwartetes Resultat: | Der Browser öffnet sich mit der erwünschten URL |

|  |  |
| --- | --- |
| Testfall-Nr. 3 | |
| Anforderung: | Das Programm greift auf «uat3.cembra.ch» zu und setzt einen Kreditbetrag und betätigt den Button «Next» |
| Beschreibung: | Das Programm öffnet den Chrome Browser und greift auf die URL «uat3.cembra.ch» zu. Anschliessend wird der Kreditbetrag auf 2000 gesetzt und der Button «Next» wird betätigt. |
| Testvorgang: | Das Programm wird ausgeführt mit dem entsprechenden Source Code für einen Kreditbetrag von 2000 CHF. |
| Erwartetes Resultat: | Das Programm greift erfolgreich auf «uat3.cembra.ch» zu, setzt eine Kreditbetrag von 2000 CHF und betätigt anschliessend den Button «Next». |

|  |  |
| --- | --- |
| Testfall-Nr. 4 merge 5 | |
| Anforderung: | Screenshots werden Gemacht vor und nach dem betätigen des «Next» Buttons |
| Beschreibung: | Zur Überprüfung der Error welche von der Webseite erzeugt werden und zum Bestätigen das gewisse Eingaben fehlerfrei funktionieren müssen Screenshots vor und nach der Betätigung eines Buttons erstellt werden. |
| Testvorgang: | Das Programm muss mit dem entsprechenden Source Code ausgeführt werden und anschliessend der «Selenium\_Image» Ordner überprüft. |
| Erwartetes Resultat: | Das Programm macht die entsprechenden Screenshots und diese sollten im Ordner «Selenium\_Image» ersichtlich sein. |

|  |  |
| --- | --- |
| Testfall-Nr. 5 | |
| Anforderung: | Fehlermeldungen wird korrekt eingelesen |
| Beschreibung: | Bei einer Eingabe zum Kreditbetrag wird absichtlich ein ungültiger Wert eingetragen um zu überprüfen ob die Fehlermeldung korrekt einlesen werden kann. |
| Testvorgang: | Das Programm wird ausgeführt mit einem Kreditbetrag von 10 CHF und es sollte eine entsprechende Fehlermeldung wiedergegeben werden. Das Programm sollte nicht weiterlaufen und |
| Erwartetes Resultat: | Der Vergleichstest ist erfolgreich |