

Individuell uppgift 3: Automatiserade UI-tester

Bakgrund

Bothniabladet, en framstående dagstidning med en expansiv bildbyrå, står inför växande utmaningar i hanteringen av sitt omfattande bildarkiv. Med en ökande volym av bildmaterial från interna och externa källor samt komplexa behov av rättighetshantering och arkivering, är effektiviteten och tillförlitligheten i deras bildhanteringssystem avgörande. I processen att utveckla ett nytt system som inte bara förbättrar dessa processer ingår också att erbjuda ett användarvänligt gränssnitt (GUI) anpassat för att möta både interna medarbetares och externa kunders behov.

I ljuset av dessa behov är utformningen av robusta testscenarier genom automatisering inte bara önskvärt utan nödvändigt. Automatiserade testflöden kan dramatiskt öka effektiviteten i testprocesserna genom att snabbt identifiera fel och brister i systemet innan de når slutanvändare.

För detta ändamål har man överväger man att nyttja Selenium IDE, ett verktyg som möjliggör snabb utveckling och exekvering av testfall direkt i webbläsaren. Verktöget tillåter snabb prototyping och iterering av testscenarier utan behov av djupgående programmeringskunskaper. Bothniabladet är dock öppna för att nyttja annat verktyg för att utföra enhetstesterna, exempelvis Playwright, Cypress eller liknande.

Uppgift

Syftet med denna uppgift är att designa och implementera en serie av automatiserade testflöden för att validera kritiska funktioner i Bothniabladets nya bildhanteringssystem. Genom att använda ett UI-testverktyg ska du skapa testfall som simulerar tester för nyckelfunktionerna inloggning och bildsökning. Testerna ska utföras på <https://ltu-i0015n-2024-web.azurewebsites.net/> och åtminstone följande funktioner ska testas:

Bildsökning

- **Mål:** Simulera tester där användaren söker efter bilder med specifika nyckelord och verifiera att sökresultaten matchar de angivna sökorden. Både positiva och negativa testfall ska inkluderas.
- En lista på alla bilder som finns i testdatabasen kan hittas genom att gå till adressen <https://ltu-i0015n-2024-web.azurewebsites.net/images>

Inloggning

- **Mål:** Simulera ett test där en användare försöker logga in och verifiera resultatet. Både positiva och negativa testfall ska inkluderas.
- Två testkonton har satts upp:
 - Username: "stina", password: "fåGelskådning"
 - Username: "johan", password: "FotoGrafeRing1!"

För varje test, dokumentera processen steg för steg och inkludera skärmbilder eller kodexempel från UI-testverktyget, som visar de viktiga stegen i testflödet. Beskriv förväntade resultat och faktiska resultat av testerna.

Tester ska inte bara säkerställa att systemet fungerar som förväntat utan också att det är robust nog att hantera både vanliga och ovanliga användarscenarier.

Glöm inte att testerna inte får vara beroende av att man utför dem i en viss ordning, utan man ska kunna utföra endast ett test, fristående från varandra. Så varje test behöver inkludera allt som behövs för att utföra testet från en helt ny webbläsare, men ska också klara av att man utför testerna ett i taget med samma webbläsare.

Du får gärna använda inspelningsfunktionen i det UI-testverktyg du nyttjar, för att skapa simulerade testfall genom att utföra åtgärder i webbläsaren. Om det behövs, uppdatera sedan testfallen för att göra testfallen mer robusta och minska risken att testfallen misslyckas på grund av exempelvis estetiska ändringar.

Installation och konfiguration av Selenium IDE: <https://www.selenium.dev/selenium-ide/docs/en/introduction/getting-started>

Installation och konfiguration av Playwright: <https://playwright.dev/docs/intro> (eller VS Code extension likt den som nyttjats under lektion: <https://playwright.dev/docs/getting-started-vscode>)

Inlämning

En rapport som innehåller:

- En kort beskrivning av Bothniabladets behov och syftet med uppgiften.
- Testplan:
 - En detaljerad beskrivning av varje testfall inklusive syfte, genomförandesteg, förväntade resultat och risker.
 - Om du använt ett verktyg som behöver ytterligare konfiguration för att kunna utföra dina tester, inkludera Instruktioner för hur man konfigurerar och använder verktyget för att utföra testerna.
- Detaljerad presentation av varje testfall med namn och vad testet syftar till att verifiera samt detaljerad steg-för-steg beskrivning av varje testfall som utfördes, inklusive skärmbilder.
- Reflektion över användningen av automatiserade UI-tester som inkluderar fördelar och utmaningar.
- Reflektion kring specifika önskemål om hur applikationen skulle kunna förändras för att göra testerna mer robusta eller på andra sätt göra systemet mer testbart.
- Eventuella defekter era tester funnit i systemet

Bedömning

Uppgiften kommer att bedömas baserat på följande kriterier:

- Relevansen och omfattningen av teststrategin och testplanen
- Visad förståelse för hur UI-testverktyg kan användas för automatiserade tester i förhållande till specificerad funktionalitet (inloggning och bildsökning).
- Testfallens utformning inklusive täckningsgrad.
- Reflektion över automatiseringens fördelar och utmaningar.
- Diskutera potentiella förbättringar för de genomförda testerna.
- Diskutera potentiella förändringar i själva webbapplikationen som kan förbättra för testerna.