

VERKKOKAUPAN REGRESSIOTESTAUKSEN AUTOMATISOINTI

Miela Designroom

Perkkiö Antti

Opinnäytetyö

Tietojenkäsittelyn koulutus
Tradenomi (AMK)

2024

Tietojenkäsittelyn koulutus
Tradenomi (AMK)

| | | | |
|-----------------------|--|--------------|------|
| Tekijä | Antti Perkkiö | Vuosi | 2024 |
| Ohjaaja(t) | Yrjö Koskenniemi | | |
| Toimeksiantaja | Miela Designroom | | |
| Työn nimi | Verkkokaupan regressiotestauksen automatisointi: Miela Designroom | | |
| Sivumäärä | 36 | | |

Tämä opinnäytetyön aiheena oli kehittää automatisoitu regressiotestaus dynaamiseen verkkokauppaan. Tavoitteena oli suunnitella sekä toteuttaa testitapaukset ja automatisoida ne. Kehitystyössä painotettiin kokonaisuuden ylläpidettävyyttä.

Tutkimustyö tehtiin käytännönläheisyytensä vuoksi tutkimuksellisen kehityksen keinoin toimintatutkimuksen tutkimusotteella. Työhön sisältyi käyttötapausten, skenaarioiden sekä testitapausten suunnittelun lisäksi testitapausten automatisointi. Automatisointiin käytettiin ennalta määriteltyä Robot Frameworkia sekä Browser-kirjastoa.

Opinnäytetyön tuloksena syntyi kaksi käyttötapausta automatisoitune testitapauksineen. Testitapaukset kehitettiin niin, että ne tukevat automatisointia ja automaatio kehitettiin puolestaan modulaariseksi ylläpidettävyyden takia. Opinnäytetyön tulokset olivat positiivisia, joten ne ovat hyödynnettävissä vastaavissa tutkimuksissa tai muissa tarpeissa.

Avainsanat
Muita tietoja

testaus, automaatio, ohjelmistokehitys, modulaarisuus
Työhön liittyy toimeksiantajalle kehitetyt käyttötapaukset, skenaariot sekä automatisoidut testitapaukset.

Business Information Technology
Bachelor of Business Administration

| | | | |
|------------------------|---|-------------|------|
| Author | Antti Perkkiö | Year | 2024 |
| Supervisor(s) | Yrjö Koskenniemi | | |
| Commissioned by | Miela Designroom | | |
| Title | Automating regression testing for e-commerce: Miela Designroom | | |
| Number of pages | 36 | | |

The topic of this thesis is to develop automated regression testing for a dynamic ecommerce store. The goal was to design and implement test cases and automate them. The emphasis in the development work was on maintainability of the whole.

The research was conducted through practical development methods with an action research approach, due to its practical nature. In addition to the design of use cases, scenarios, and test cases, the automation of test cases was included. Pre-defined Robot Framework and Browser library were used for automation.

As a result of the thesis, two use cases with automated test cases were developed. The test cases were designed to support automation, and the automation was developed to be modular for maintainability. The results of the thesis were positive and therefore, can be utilized in similar studies or other needs.

Keywords testing, automation, software development, modularity
Special remarks The thesis includes developed use cases, scenarios, and automated test cases for the commissioner

SISÄLLYS

| | | |
|------|--|----|
| 1 | JOHDANTO | 5 |
| 2 | OPINNÄYTETYÖN TAVOITTEET JA MENETELMÄLLINEN TOTEUTUS | 7 |
| 2.1 | Tutkimusmenetelmät..... | 7 |
| 2.2 | Eettiset lähtökohdat ja luotettavuus | 8 |
| 3 | TUTKIMUSAINEISTON ESITTELY | 9 |
| 3.1 | Regressiotestaus ja testiautomaatio | 9 |
| 3.2 | Ylläpidettävyyden määritelmä | 10 |
| 3.3 | Aiheeseen liittyvät opinnäytetyöt..... | 10 |
| 4 | TESTAUKSEN SUUNNITTELU | 12 |
| 4.1 | Verkkokauppa-alusta | 12 |
| 4.2 | Ohjelmisto- ja WEB-testaus | 12 |
| 4.3 | Ohjelmistotestauksen menetelmät..... | 13 |
| 4.4 | Ohjelmistotestauksen tasot..... | 15 |
| 4.5 | WEB-testauksen tyypit..... | 16 |
| 4.6 | Regressiotestaus | 17 |
| 4.7 | Automatisoitu testaus..... | 18 |
| 4.8 | Tietojärjestelmien ylläpidettävyys | 19 |
| 4.9 | Page Object Model -suunnittelumalli | 20 |
| 4.10 | Keyword Driven Framework -viitekehys | 21 |
| 4.11 | Automatisoinnin työkalut..... | 21 |
| 4.12 | Testiympäristön suunnittelu | 22 |
| 4.13 | Käyttötapausten suunnittelu | 23 |
| 4.14 | Testiskenaarioiden suunnittelu | 25 |
| 4.15 | Testitapausten suunnittelu..... | 25 |
| 5 | TESTAUKSEN TOTEUTUS..... | 27 |
| 5.1 | Käyttötapausten ja testiskenaarioiden toteutus..... | 27 |
| 5.2 | Testitapausten toteutus..... | 29 |
| 5.3 | Testien automatisointi..... | 30 |
| 6 | POHDINTA | 32 |
| | LÄHTEET..... | 34 |

1 JOHDANTO

Verkko-ostamisesta on tullut pysyvä osa suomalaisten elämää. Vuonna 2021 57 % 16–89-vuotiaista suomalaisista oli ostanut jotain verkosta edellisen kolmen kuukauden aikana. (Suomen virallinen tilasto (SVT) 2021.) Vuonna 2022 yleinen kustannusten ja hintojen nousu sekä pandemian tuoman ylimääräisen kasvusysäyksen poistuminen kuitenkin pysäytti yli 10 vuotta kestäneen kuluttajien tekemien tavaroiden digiostosten arvon nousun myös Suomessa. Samaan aikaan kotimaasta tehtyjen verkko-ostosten osuus pieneni ja ulkomailta ostaminen kasvoi. (Kaupan liitto 2023.) Näiden tietojen valossa käy ilmi, että verkko-ostamisen suosioista huolimatta verkkokauppojen keskinäinen kilpailu asiakkaista on viime aikoina koventunut ja kotimaiset verkkokaupat ovat menettämässä osuutensa kaupasta ulkomaisille toimijoille.

Opinnäytetyön tavoitteena on suunnitella sekä automatisoida dynaamisella verkkokauppa-alustalla toimivan verkkokaupan regressiotestaus. Regressiotestauksen tarkoituksena on määrittää, onko ohjelmaan tehty toiminnallinen parannus tai korjaus vaikuttanut ohjelman muihin osa-alueisiin (Myers, Sandler & Badgett 2012, 134). Testitapausten automatisointi mahdollistaa testien uudelleenkäytön useilla alustoilla, ympäristöissä, käyttöjärjestelmissä, selaimissa ja mobiililaitteilla, mikä vähentää kustannuksia ja tehostaa testausta (Jose 2021, 8).

Opinnäytetyön toimeksiantajana toimii Miela Designroom. Mielan liiketoimintaan kuuluu myyntipaikkojen vuokraaminen kotimaisille muoti- ja designbrändeille, joiden puolesta Miela hoitaa tuotteiden myynnin ja markkinoinnin. Tuotteiden myynti tapahtuu Tampereella sijaitsevan kivijalkamyymälän lisäksi verkkokaupassa, joka toimii Shopify-alustalla. Verkkokauppaa voidaan käyttää useilla eri laitteilla ja selaimilla, kuten tietokoneella, tai mobiililaitteella Chrome- tai Opera-selainta hyödyntäen. Alustan yleisten päivitysten lisäksi Miela kehittää ja päivittää verkkokaupan toimintoja jatkuvasti. Verkkokaupan toimintoja testataan nykyään manuaalisesti ja vain kehitystyön yhteydessä resurssien takia. (Minkkinen, A. & Keski-Heikkilä, I. 2024.)

Opinnäytetyön päätutkimuskysymys on, miten saadaan suunniteltua ja toteutettua ylläpidettävä automatisoitu regressiotestaus dynaamiseen verkkokauppaan.

Ylläpidettävyys määrittelee, kuinka tehokkaasti ja tuloksellisesti tuotetta tai järjestelmää voidaan muuttaa sen parantamiseksi, korjaamiseksi tai mukauttamiseksi ympäristön ja vaatimusten muutoksiin (SFS-ISO/IEC 25010:2019). Työtä rajataan laajuuden vuoksi keskittymällä liiketoiminnan kannalta kriittisten toimintojen toiminnan varmistamiseen. Samalla opinnäytetyön puitteissa regressiotestit automatisoidaan yhteen Mielan kanssa erikseen määriteltyyn liiketoiminnan kannalta keskeiseen alustaan.

Opinnäytetyön aihe valittiin tekijän oman mielenkiinnon sekä toimeksiantajan tarpeen mukaan. Tekijä on saanut opinnoissaan laajan ymmärryksen muun muassa digitaalisten palvelujen suunnittelusta ja hyödyntämisestä. Tämän lisäksi tekijä työskentelee Test Engineerinä ohjelmistotuotannon parissa. Nämä seikat antavat hyvän pohjan opinnäytetyön tekemiseen.

2 OPINNÄYTETYÖN TAVOITTEET JA MENETELMÄLLINEN TOTEUTUS

Opinnäytetyön tarkoituksena on suunnitella ja toteuttaa automatisoitu regressiotestaus verkkokauppaan. Tavoitteena on saavuttaa standardin SFS-ISO/IEC 25010:2019 mukaisesti mahdollisimman ylläpidettävä kokonaisuus toimeksiantajan resursseista johtuen. Regressiotestaamisen tarpeeseen ollaan vasta havahduttu ja tämän työ vastaa siihen. Työstä tulee olemaan hyötyä opinnäytetyön tekijän ammattitaidon kehittymisen lisäksi toimenantajalle sekä todennäköisesti myös muille aiheetta tutkiville.

Opinnäytetyön päätutkimuskysymys on, miten saadaan suunniteltua ja toteutettua ylläpidettävät automatisoidut testitapaukset dynaamiseen verkkokauppaan. Tarpeita ei ole määritelty etukäteen, joten määrittelyä tehdään työn edetessä kirjallisuuteen perustuen sekä toimeksiantajan kanssa neuvotellen. Apututkimuskysymys on, miten suunnitella regressiotestaus niin, että se on kohtuullisella työllä laajennettavissa sekä niin, että se tukee automatisointia.

2.1 Tutkimusmenetelmät

Opinnäytetyö toteutetaan laadullisena tutkimuksena. Laadullisen tutkimuksen tarkoituksena on kuvailla jotain ilmiötä seikkaperäisesti, saada jokin asia ymmärretyksi tai kehittää uutta teoriaa todellisuutta vastaavasta aineistosta (Anttila 2014). Laadullisessa tutkimuksessa pyritään tarkastelemaan teorian, empirian ja käytännön keskinäistä yhteyttä (Juuti & Puusa 2020, 56–57). Tässä opinnäytetyössä käytännön kautta saavutettu kokemuksellinen tieto on keskeisessä asemassa, sillä vaikka aiheesta on runsaasti tietoa saatavilla, keskeinen kysymys liittyy siihen, miten tätä tietoa kannattaa käytännössä soveltaa.

Opinnäytetyö tehdään tutkimuksellisen kehityksen keinoin, sillä aihe on hyvin käytännönläheinen. Tutkimuksellinen kehittäminen erottuu perinteisestä tieteellisestä tutkimuksesta käytännönläheisyytensä vuoksi. Sen tarkoituksena on saada teorian tuotannon lisäksi aikaan myös käytännön parannuksia tai uusia ratkaisuja. (Ojasalo, Moilanen & Ritalahti 2015, 17–19.)

Tutkimukselliseen kehitystyöhön sovelletaan opinnäytetyössä tapaustutkimuksen tutkimusotetta. Tapaustutkimus soveltuu hyvin kehittämistyön lähestymistavaksi, kun tavoitteena on tuottaa kehittämissuhteita tai -ideoita. Tutkimuksen kohde, eli tapaus voi olla esimerkiksi tuote, palvelu tai prosessi. Tapaustutkimuksen tavoitteena on tuottaa syvällistä ja yksityiskohtaista tietoa tutkittavasta tapauksesta, jonka avulla on mahdollista ymmärtää kehittämisen kohdetta kokonaisvaltaisesti realistisessa ympäristössä. (Ojasalo ym. 2015, 52.)

2.2 Eettiset lähtökohdat ja luotettavuus

Hyvän tieteellisen käytännön kannalta on myös opinnäytetyön arviointi tehtävä rehellisesti ja tarkasti. Arvioinnissa ei riitä se, että itse toteaa noudattaneensa eettisiä periaatteita, vaan myös lukijalle on kerrottava, miten se tehtiin. (Vilkkä 2021, 142.) Tässä opinnäytetyössä käsitellään vain työn tekemisen kannalta välttämättömiä tietoja. Tutkimusaineisto pyritään keräämään niin, ettei se sisällä henkilötietoja tai liiketoiminnalle kriittistä tietoa.

Luotettavuuden arvioinnissa järjestelmällinen ja johdonmukainen työskentely korostuu läpi prosessin. Valintojen johdonmukaisuutta ja tarkoituksellisuutta tulee arvioida kaiken aikaa suhteessa kohteeseen ja tavoitteisiin. (Vilkkä 2021, 138.) Tutkimuksen johdonmukaisuus ja tarkoituksellisuus pyritään säilyttämään läpi prosessin ja niitä reflektoidaan määriteltyihin tavoitteisiin sekä tutkimuksen kohteeseen.

Tutkimuksessa lähteen ja aineiston sisäinen laatu vaikuttaa suoraan opinnäytetyön luotettavuuteen ja laatuun, jonka takia lähdekritiikki on olennaisessa osassa tutkimuksen tekemistä (Vilkkä 2021, 84). Tämän tutkimuksen aiheesta on saatavilla laajasti kirjallista aineistoa. Työn tekijän omakohtainen kokemuksellinen tieto aiheesta auttaa osaltaan lähdekritiikkiin, jonka lisäksi huomiota kiinnitetään niin kirjoittajaan kuin julkaisijaankin. Tutkimuksessa pyritään käyttämään mahdollisimman ajantasaista tietoa niiltä osin, kuin sen katsotaan olevan tarpeellista laadun ja luotettavuuden kannalta.

3 TUTKIMUSAINIESTON ESITTELY

3.1 Regressiotestaus ja testiautomaatio

Ohjelmistokehitystä ja sen myötä ohjelmistotestausta sekä testauksen automatisointia on tehty jo pitkään, joten aiheista on kirjoitettu lukuisissa eri lähteissä. Käsitteet liittyvät läheisesti toisiinsa, joten niitä usein myös käsitellään samoissa teoksissa.

Jussi Pekka Kasurisen (2013) kirjoittama ”Ohjelmistotestauksen käsikirja” esittelee ohjelmistojen testauksen peruseriaatteen, tärkeimmät testauksen työkalut sekä testauksen työvaiheet. Kirjassa käsitellään myös testauksen hallintoa, esitellään ammattitestaajille tärkeimmät alan sertifikaatit ja standardit sekä käydään läpi ohjelmistotuotannon perusteet siinä laajuudessa, kuin on tarpeen kirjan sisällön ymmärtämiseksi.

Myersin ym. (2012) ”The Art of Software Testing” -kirjan kolmas painos tarjoaa tiiviin ja vaikuttavan katsauksen vakiintuneisiin lähestymistapoihin ohjelmistojen testauksessa. Kirja toimii lyhyenä ja kattavana oppaana testauksen peruseriaatteisiin ja parhaisiin käytäntöihin.

Boby Josen (2021) kirjoittama ”Test Automation: A manager’s guide” on laaja-alainen opas, joka on suunniteltu erityisesti johtajille ja kattaa perusteellisesti testiautomaation eri näkökulmat alkaen keskisuurista aina suuriin toteutuksiin. Oppaassa tarkastellaan perusteellisesti testiautomaation etuja ja keskitytään asianmukaisten testiautomaatiomenetelmien määrittelyyn, niiden rakentamiseen ja toteuttamiseen. Lisäksi opas käsittelee testien toteuttamistyökalujen ja -puitteiden hankkimista sekä niiden käyttöönottoa.

Regressiotestaus on yleisnimitys kaikelle uudelleentestaamiselle ja sen keskeinen tehtävä on todentaa, että uusi versio toimii halutulla tavalla (Kasurinen 2013, 68–69). Myersin ym. (2012, 134) mukaan regressiotestaus suoritetaan sen jälkeen, kun ohjelmaan on tehty toiminnallinen parannus tai korjaus. Sen tarkoituksena on määrittää, onko muutos vaikuttanut ohjelman muihin osa-alueisiin.

Automaattinen testaus on luotettavampaa kuin manuaalinen testaus, sillä se eliminoi inhimillisen virheen mahdollisuuden. Manuaalinen testaus on aikaa vievää ja altis virheille. (Myers ym. 2012, 178–179.) Testauksen automatisointi on prosessi, jossa käytetään muita ohjelmistoja manuaalisen testauksen tai sovelluksessa suoritettavien manuaalisten käyttäjän toimien automatisointiin. Automatisointi mahdollistaa samojen testitapausten uudelleenikäytön suorittamalla samat testit useita kertoja pienin kustannuksin. (Jose 2021, 3.)

3.2 Ylläpidettävyyden määritelmä

SFS-ISO/IEC 25010:2019 on kansainvälinen standardi, joka määrittelee laatuomallin ohjelmistojen ja järjestelmien laadulle. Se tarjoaa yleisen viitekehyksen ohjelmistojen ja järjestelmien laadun määrittelemiseen ja arviointiin. Laatuomalli antaa organisaatioille mahdollisuuden asettaa selkeitä tavoitteita ja arvioida niiden saavuttamista näiden laatuominaisuusalueiden avulla. Standardin esittelemä ylläpidettävyysominaisuus määrittelee, kuinka tehokkaasti ja tuloksellisesti tuotetta tai järjestelmää voidaan muuttaa sen parantamiseksi, korjaamiseksi tai muuttamiseksi ympäristön ja vaatimusten muutoksiin. (SFS-ISO/IEC 25010:2019.)

3.3 Aiheeseen liittyvät opinnäytetyöt

Aiheeseen soveltuvia opinnäytetöitä on saatavilla useita. Anniina Aaltonen (2019) lähestyy opinnäytetyössään ”Verkkokaupan regressiotestauksen automatisointi” aihetta ketterän ohjelmistokehityksen näkökannasta. Aaltonen työssä käydään testiautomaation lisäksi läpi ohjelmistokehitystä yleisesti sekä testauksen eri tasoja ja vaiheita. Työn tuotoksena oli yhdeksän automatisoitua testitapausta verkkomaksamisen toiminnan varmistamiseksi.

Tupaq Castro (2016) käsittelee aihetta lopputyössään ”Verkkokaupan automaattiotestaus”. Työssä Castro tähtää manuaalisten testitapausten automatisointiin ja tuotoksena syntyy kattava määrä automatisoituja testitapauksia, joista suurin osa ei toimi opinnäytetyön valmistumisvaiheessa verkkokaupan muuttuneen ulkoasun takia.

Joona Leppälahden (2018) opinnäytetyössä ”Regressiotestauksen automatisointi Robot Frameworkilla” käsitellään ohjelmistotestausta ja sen eri tasoja, testausautomaatiota, erilaisia testauskehyksiä sekä Robot Frameworkia. Työn toiminnallisessa osiossa tarkastellaan regressiotestauksen käytännön toteutusta.

4 TESTAUKSEN SUUNNITTELU

4.1 Verkkokauppa-alusta

Shopify on pilvipohjainen verkkokauppa-alusta, joka tarjoaa mahdollisuuden myydä tuotteita verkkokaupan lisäksi monissa muissa kanavissa, kuten sosiaalisessa mediassa ja kivijalkamyymälässä. Koska Shopify on isännöity palvelu, verkkokaupan perustaminen ja sen toimintojen kehittäminen ovat vaivatonta ilman tarvetta huolehtia ohjelmistojen tai verkkopalvelimien ylläpidosta. (Shopify 2024a.)

SaaS (Software as a Service) on ohjelmistojen jakelumalli, jossa pilvipalveluntarjoaja isännöi sovelluksia ja mahdollistaa niiden käytön loppukäyttäjille internetin välityksellä. SaaS vähentää yritysten kustannuksia ja helpottaa kulujen ennakoimista, kun ne voivat luopua paikallisista asennuksista ja ylläpidosta. (Chai & Casey 2022.) SaaS-palveluiden riskeistä suurin osa liittyy siihen, että kolmannen osapuolen palveluntarjoaja vastaa palvelujensa turvallisuudesta ja käytettävyydestä (Carey 2021).

Mielan verkkokauppa toimii Shopify-alustalla, joka on isännöity SaaS-palvelu. Palvelun tuottaja vastaa ohjelmiston ja palvelimien ylläpidosta, kun taas Mielan vastuulle jää verkkokaupan kehittäminen valmiiksi kehitettyjen toiminnallisuuksien avulla. Miela käyttää verkkokaupassaan alustan toiminnallisuuksien lisäksi siihen integroituja kolmannen osapuolen SaaS-palveluita.

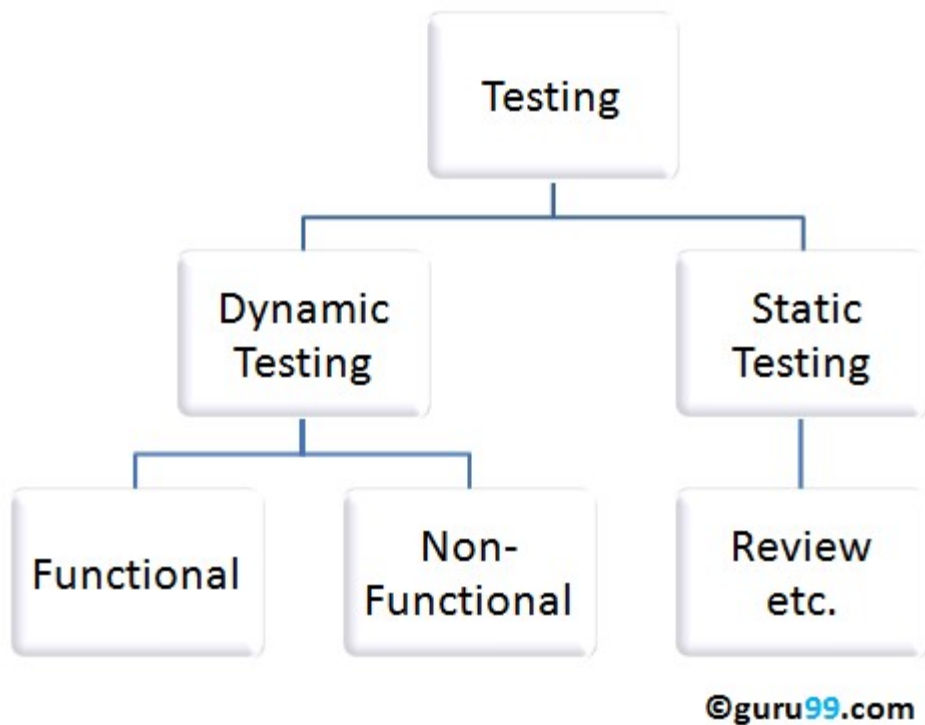
4.2 Ohjelmisto- ja WEB-testaus

Ohjelmistotestaus voidaan kuvata prosessina tai joukkona prosesseja, joiden tarkoituksena on varmistaa, että ohjelmistokoodi toimii suunnitellulla tavalla. Ohjelmiston toiminnan tulisi olla mahdollisimman johdonmukaista ja helposti ennustettavaa, eikä sen tulisi aiheuttaa käyttäjille suuria yllätyksiä. (Myers ym. 2012, 2.) WEB-testaamisella tarkoitetaan verkkosovelluksen tai -sivujen tarkistamista mahdollisten virheiden varalta ennen kuin se otetaan käyttöön yleisön saataville (Hamilton 2023a).

Verkkokaupassa asiakkaan kynnys siirtyä toiseen kauppaan on erittäin pieni ja kilpailijan sivu saattaakin olla valmiiksi auki toisessa ikkunassa (Lahtinen 2013, 113). Opinnäytetyön tavoitteena automatisoida regressiotestaus Mielan verkkokaupan toimivuuden varmistamiseksi päivitysten jälkeen.

4.3 Ohjelmistotestauksen menetelmät

Testausta voidaan suorittaa monin eri tavoin, jotka vaihtelevat ohjelmistotuotteen kehitysvaiheen sekä testauksen näkökulman mukaan. Testausmenetelmät voidaan jakaa karkeasti staattiseen ja dynaamiseen testaamiseen (kuvio 1).



Kuvio 1. Testausmenetelmät (Hamilton 2023b)

Staattisella testaamisella tarkoitetaan järjestelmän testaamista niin, että järjestelmää tutkitaan esimerkiksi koodianalysaattorien, koodiarviointien tai arkkitehtuurisuunnittelun näkökulmasta. Staattisten menetelmien tarkoituksena on poistaa ilmiselvät ongelmat järjestelmästä ennen kuin tarkempi testaus aloitetaan. Tämä testaaminen voidaan aloittaa jo varhaisessa vaiheessa, jopa jo pelkkien arkkitehtuurisuunnitelmien pohjalta. (Kasurinen 2013, 65.) Hamiltonin (2023b) mukaan

staattinen testaaminen sisältää kooditarkastuksien lisäksi dokumenttisuunnitelmien arvioinnin, jonka tarkoituksena on löytää niistä mahdollisia virheitä sekä saada niistä palautetta.

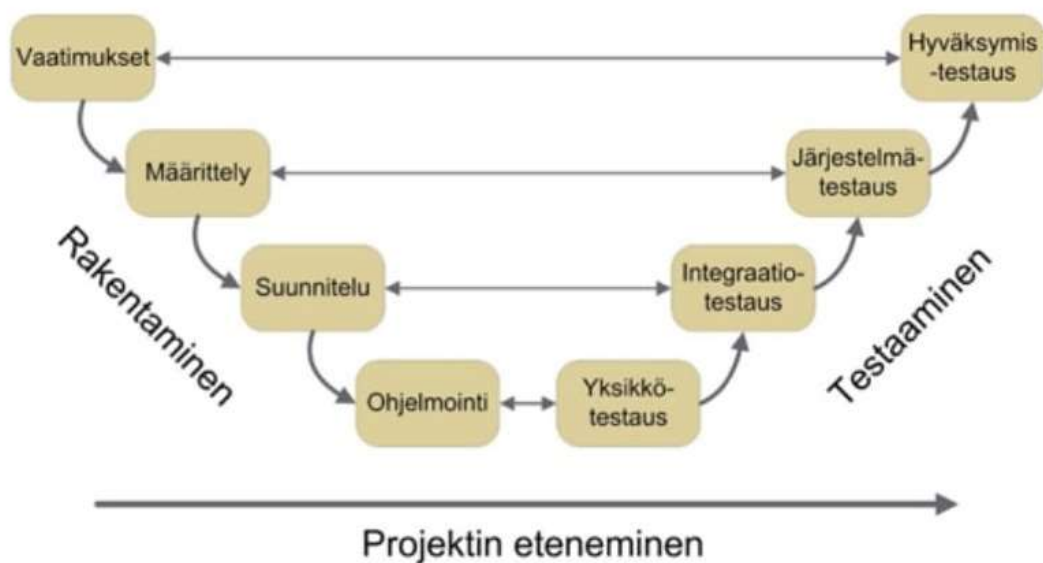
Dynaaminen testaus on staattisen testauksen vastakohta, jossa testattavaa järjestelmää käytetään ja sen reaktiota annettuihin syötteisiin seurataan. Tämä testaaminen voidaan aloittaa siinä vaiheessa, kun järjestelmää voidaan koekäyttää prototyyppeinä tai osittain toteutettuna järjestelmänä. (Kasurinen 2013, 65.) Dynaamisen testauksen päätavoitteena on varmistaa, että järjestelmä toimii liiketoimintavaatimusten mukaisesti, eli validoida se. Testaaminen jakautuu toiminnalliseen ja ei-toiminnalliseen testaamiseen. Toiminnallisen testauksen tavoitteena on varmistaa jokainen järjestelmän toiminto, kun taas ei-toiminnallinen testaaminen keskittyy esimerkiksi järjestelmän suorituskykyyn, käytettävyyteen tai luotettavuuteen. (Hamilton 2023c.)

Dynaamisen ja staattisen testausmenetelmien lisäksi ohjelmistotestaus voidaan jakaa myös testaustyylin mukaan Black Box ja White Box -testaukseen. Black Box -testaamisessa järjestelmää testataan antamalla sille syötteitä ja seuraamalla sen reaktiota ilman, että järjestelmän sisäistä toimintaa tarkastellaan. Testauksella voidaan varmentaa erilaisia käyttötapauksia, kuten esimerkiksi tiedostojen tallentamista ja avaamista, lomakkeiden syöttämistä tai painikkeiden toimintoja. White Box -testaamisella puolestaan tarkoitetaan testausmenetelmää, jossa järjestelmän sisäistä toimintaa tarkastellaan testauksen aikana. White Box -testaus on näin ollen syvempää ja tarkempaa kuin Black Box testaus. (Kasurinen 2013, 65–67.)

Opinnäytetyön tavoitteena on varmistaa, että Mielan kanssa erikseen sovitut asiakkaiden käyttämät toiminnot toimivat verkkokaupassa. Toimintojen varmistaminen jo valmiissa järjestelmässä on luonteeltaan dynaamista ja funktionaalista testaamista. Automatisoitu regressiotestaus on tarkoituksenaan toteuttaa isännöityyn SaaS-palveluun, jossa palvelun käyttäjien näkyvyys ohjelmistokoodiin on rajallista, joten se tullaan toteuttamaan Black Box testauksena.

4.4 Ohjelmistotestauksen tasot

Testauksen V-malli määrittelee ohjelmiston kehitysvaiheen testaamiselle kolme päävaihetta (kuvio 2). Vaiheet ovat yksikkötestaus, integrointitestaus ja järjestelmätestaus. Nämä vaiheet suoritetaan testausympäristössä. Kehitysvaiheen jälkeen seuraa hyväksymistestaus, joka suoritetaan kohdeympäristössä tai sen tarkassa simulaatiossa. (Kasurinen 2013, 51.)



Kuvio 2. Testauksen V-malli (Kasurinen 2013, 51)

Yksikkötestauksella tarkoitetaan ohjelmiston verifiointi- ja validointimenetelmää, jossa ohjelmoija testaa yksittäisten lähdekoodin moduulien käyttökelpoisuuden. Yksikkötestauksen suorittaa yleensä ohjelmoija. (Hamilton 2024a.) Yksikkötestauksen tavoitteena on varmistaa, että toteutettu muutos tai toiminto olemassa olevaan järjestelmään toimii ainakin periaatteessa (Kasurinen 2013, 51).

Integrointitestaus on ohjelmistokehityksen vaihe, jossa yksittäiset moduulit testataan yhdessä. Tämä vaihe suoritetaan yleensä testaajien toimesta. (Hamilton 2024a.) Integrointitestauksen tavoitteena on saada koko järjestelmä toimimaan kokonaisuutena (Kasurinen 2013, 54).

Järjestelmätestauksella varmistetaan integroidun järjestelmän vaatimustenmukaisuus. Yleensä testaajat suorittavat tämän vaiheen. (Hamilton 2024a.) Järjestelmätestaukseen siirrytään komponenttien yksikkötestauksen ja integrointitestauksen jälkeen ja sillä tarkoitetaan sitä testaustyötä, joka tehdään koko järjestelmälle (Kasurinen 2013, 56).

Hyväksymistestauksella arvioidaan täyttääkö järjestelmä hyväksymiskriteerit. Asiakas suorittaa tämän vaiheen ja päättää sen avulla, hyväksyykö järjestelmän vai ei. (Hamilton 2024a.) Hyväksymistestausta suoritetaan yleensä järjestelmän kohdeympäristössä, toisin kuin aiemman vaiheen testejä, jotka suoritetaan yleensä erikseen rakennetussa testiympäristössä (Kasurinen 2013, 57).

Shopify kehittää sekä ylläpitää verkkokauppa-alustaa ja tarjoaa sitä SaaS-palveluna Mielalle. Miela maksaa palvelusta ja kehittää omaa verkkokauppaansa valmiiden, Shopify:n toimesta testattujen toimintojen sekä kolmannen osapuolen integraatioiden avulla. Näin ollen voidaan katsoa, että Mielan verkkokaupan regressiotestaus on luonteeltaan pääosin hyväksymistestausta. Toisaalta SaaS-pohjaisen verkkokaupan regressiotestaamisen toimintojen päivittämisen jälkeen voidaan katsoa olevan osin myös järjestelmätestausta.

4.5 WEB-testauksen tyypit

WEB testauksella tarkoitetaan Hamiltonin (2023a) mukaan verkkosivuston tai -sovelluksen tarkastamista mahdollisten virheiden varalta ennen kuin se otetaan käyttöön yleisön saataville. Testauksen vaatimuksista riippuen WEB-testausta voidaan suorittaa joko toiminnallisuustestauksena, käytettävyydestestauksena, käyttöliittymättestauksena, tietokantatestauksena, yhteensopivuustestauksena, suorituskykytestauksena tai turvallisuustestauksena.

Verkkosivun toiminnallisuuden testaus on prosessi, joka sisältää useita erilaisia testausparametreja, kuten esimerkiksi käyttöliittymän, sovellusrajapintojen, tietokantojen ja verkkosivuston, perustoiminnallisuuksien testaamisen. Käytettävyydestestauksen tarkoituksena puolestaan on testata, että erilaiset verkkosivujen elementit, kuten esimerkiksi valikot, painikkeet ja linkit, ovat helposti löydettävissä ja johdonmukaisia jokaisella sivulla. Tämän lisäksi sen tarkoituksena on varmistaa

esimerkiksi verkkosivuilla olevan tekstin luettavuus ja oikeinkirjoitus. (Hamilton 2023a.)

Käyttöliittymätestauksen tarkoituksena on varmistaa, että verkkosovellus, WEB-palvelin ja tietokanta toimivat oikein ja luotettavasti. Testeillä voidaan tarkistaa esimerkiksi se, että tietokantaan lähetetyt kyselyt palauttavat käyttöliittymään oikean datan tai että WEB-palvelin käsittelee kaikki pyynnöt ilman palvelukatkoksia. Tietokannan testauksella puolestaan pyritään esimerkiksi varmistamaan, että kyselyiden käsittelyssä ei tapahdu virheitä ja datan eheys säilyy dataa lisätessä, päivittäessä tai poistaessa. Näiden lisäksi voidaan testata kyselyiden vasteaikaa sekä datan oikeellisuutta verkkosivuilla. (Hamilton 2023a.)

Yhteensopivuuden testauksella pyritään varmistamaan, että verkkosivut toimivat oikein kaikilla alustoilla ja selaimilla. Testien avulla voidaan varmistaa esimerkiksi painikkeiden toimivuus eri selaimilla ja laitteilla, kuten tietokoneella ja mobiililaitteella. Suorituskykytestauksella puolestaan voidaan varmistaa, että verkkosivusto toimii luotettavasti kaikilla kuormilla, kuten normaali- ja huippukuormilla. Tämän lisäksi voidaan testata verkkosivujen toimivuutta hitailla ja nopeilla yhteyksillä. Turvallisuustestauksen avulla pyritään varmistamaan verkkosivujen tietoturvasuus. Testeillä voidaan esimerkiksi varmistaa, että suojatuille sivuille ei pääse tai tiedostoja ei saa ladattua ilman asianmukaisia käyttöoikeuksia. (Hamilton 2023a.)

Opinnäytetyön tavoitteena on varmistaa Mielan verkkokaupan toimintojen toiminta päivitysten jälkeen. Testit pyritään automatisoimaan niin, että ne ovat laajennettavissa usealle eri alustalle ja selaimelle. Näin ollen WEB-testauksen tyypeiksi voidaan katsoa opinnäytetyön kontekstissa toiminnallisuus- ja yhteensopivuustestaus.

4.6 Regressiotestaus

Regressiotestaus on yleisnimitys kaikelle uudelleentestaamisella ja sen keskeinen tehtävä on todentaa, että uusi versio toimii halutulla tavalla (Kasurinen 2013, 68–69). Regressiotestaaminen suoritetaan järjestelmälle tehdyn toiminnallisen muutoksen tai parannuksen jälkeen. Sen tarkoituksena on varmistaa, että muu-

tos ei ole hajottanut järjestelmän muita osa-alueita. Regressiotestaaminen suoritetaan yleensä ajamalla jonkin osa-alueen testitapaukset uudestaan. Regressiotestaus on tärkeää, koska muutokset ja virheiden korjaukset ovat usein paljon virhealttiimpia kuin alkuperäinen ohjelmakoodi. (Myers ym. 2012, 98.)

Opinnäytetyön tavoitteena on varmentaa Mielan verkkokaupan toiminta verkkokauppa-alustan yleisten päivitysten sekä verkkokaupan kehitystyön jälkeen. Verkkokaupan toiminallisuuksia testataan nykyään manuaalisesti kehitystyön yhteydessä ennen julkaisua, mutta laajempaa testaamista ei suoriteta. Verkkokaupan kehitystyön lisäksi Shopify julkaisee verkkokauppa-alustalle yleisiä päivityksiä, jotka otetaan käyttöön jokaiselle asiakkaalle. Regressiotestaus pyritään tämän työn puitteissa toteuttamaan Mielan kanssa yhdessä sovituille liiketoiminnan kannalta tärkeimmille toiminnoille. Verkkokaupan toimintojen kehittämisen jälkeen suoritettua regressiotestaamista voidaan katsoa olevan järjestelmätestausta, joka varmistaa verkkokaupan toiminnan kehitystyön jälkeen.

4.7 Automatisoitu testaus

Automatisoidulla testauksella tarkoitetaan ohjelmistotestausta, joka suoritetaan erityisen automatisoidun testausohjelmiston avulla (Hamilton 2024b). Regressiotestaus on yksi yleisimmistä automatisoitavista testausprosesseista. Automatisoidulla testauksella on useita hyötyjä, kuten nopeus, luotettavuus ja toistettavuus. Automatisoitu testaus on luotettavampaa kuin manuaalinen testaus, koska se vähentää inhimillisiä virheitä testien suorittamisessa. Toistettavuuden ja nopeuden ansiosta esimerkiksi regressiotestausta voidaan suorittaa nopeasti, useammin ja eri ympäristöissä, mikä vähentää kustannuksia ja testauksen vaivaa. (Jose 2021, 8, 11.)

Opinnäytetyön tavoitteena on automatisoida Mielan verkkokaupan regressiotestaus. Asiakkaat käyttävät verkkokauppaa useassa eri ympäristössä (Miela 2024). Automatisoinnin avulla testausta on mahdollista laajentaa helposti kaikille liiketoiminnan kannalta tärkeille alustoille. Shopify (2024b) voi julkaista useita yleisiä päivityksiä jopa yhden vuorokauden aikana. Automatisoidun testauksen toistet-

tavuuden ja nopeuden avulla pyritään helpottamaan regressiotestauksen suorittamista, kun taas luotettavuuden avulla voidaan vähentää inhimillisten virheiden mahdollisuutta.

4.8 Tietojärjestelmien ylläpidettävyys

Tietojärjestelmien laatua käsittelevän SFS-ISO/IEC 25010:2019 standardin ylläpidettävyys-ominaisuus kuvaa, kuinka tehokkaasti ja tuloksellisesti tuotetta tai järjestelmää voidaan muokata, korjata tai sovittaa ympäristön ja vaatimusten muutoksiin. Ylläpidettävyys sisältää seuraavat ominaisuudet: modulaarisuus, uudelleenkäytettävyys, analysoitavuus, muunneltavuus ja testattavuus. (SFS-ISO/IEC 25010:2019.)

Ominaisuuksista modulaarisuus kuvaa sitä, missä määrin järjestelmä tai tietokoneohjelma koostuu erillisistä komponenteista siten, että yhden komponentin muutoksella on minimaalinen vaikutus muihin komponentteihin. Uudelleenkäytettävyydellä puolestaan tarkoitetaan sitä astetta, jolla hyödykettä voidaan käyttää useammassa kuin yhdessä järjestelmässä tai muiden hyödykkeiden rakentamisessa. Analysoitavuus kuvaa sitä vaikutus- ja tehokkuusastetta, jolla on mahdollista arvioida muutoksen vaikutusta yhteen tai useampaan tuotteen tai järjestelmän osaan, tuotteen puutteiden tai vikojen syiden diagnosoimiseksi, tai muutettavien osien tunnistamiseksi. (SFS-ISO/IEC 25010:2019.)

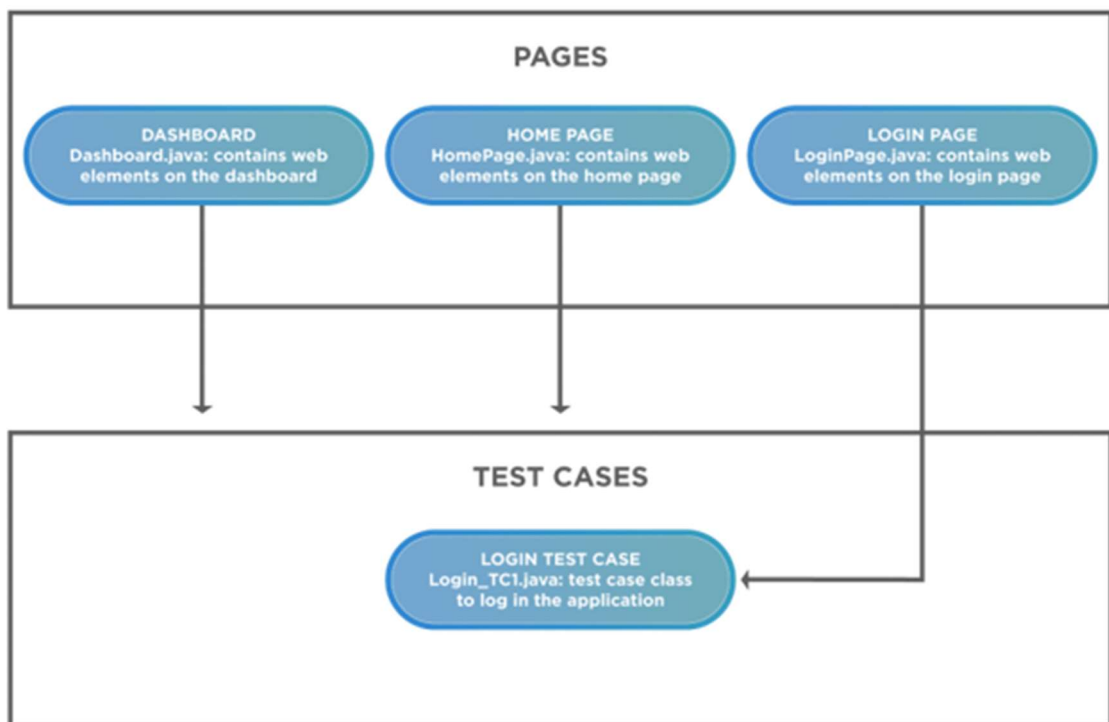
Ylläpidettävyys-ominaisuuden mukaan muunneltavuudella tarkoitetaan sitä, missä määrin tuotetta tai järjestelmää voidaan muuttaa tehokkaasti ja tuloksellisesti ilman, että se aiheuttaa vikoja tai heikentää olemassa olevan tuotteen laatua. Testattavuudella puolestaan tarkoitetaan sitä vaikutus- ja tehokkuusastetta, jolla järjestelmälle, tuotteelle tai komponentille voidaan laatia testauskriteerit ja suorittaa testit sen määrittämiseksi, täytyvätkö nämä kriteerit. (SFS-ISO/IEC 25010:2019.)

Verkkokaupan automatisoitu testaus pyritään suunnittelemaan modulaarisesti niin, että verkkosivuilla tapahtuvat muutokset voidaan korjata helposti muokkaamalla yhtä moduulia testaustyökalussa. Samalla pyritään uudelleenkäytettävyyteen niin, että automatisoitu testaus voidaan halutessa laajentaa useammalle

alustalle tai selaimelle. Muunneltavuuden avulla pyritään tekemään automatisoidun testauksen kattavuuden laajentaminen mahdollisimman helpoksi.

4.9 Page Object Model -suunnittelumalli

Page Object Model on suosittu automatisoidussa testaamisessa käytetty modulaarinen testien suunnittelumalli. Sen tarkoituksena on vähentää koodin päällekkäisyyksiä ja tehostaa testien ylläpidettävyyttä. Sivuoobjetti (Page Object) edustaa verkkosivun elementtejä sekä metodeja sisältävää luokkaa, joka tarjoaa rajapinnan testattavan sovelluksen verkkosivuun (kuvio 3). Se voi myös edustaa jotain sivustolla usein esiintyvää toimintoa, kuten navigointipalkkia. Testit käyttävät sivuoobjektin metodeja aina, kun niiden on oltava vuorovaikutuksessa kyseisen verkkosivun käyttöliittymän kanssa. Mallin eduiksi voidaan laskea se, että testikoodi ja sivukohtainen koodi on erotettu selkeästi toistaan, sekä se, että sivun tarjoamille elementeille ja metodeille on yksi ainoa säilytyspaikka. Tämä mahdollistaa sen, että käyttöliittymän muutosten edellyttämä korjaukset voidaan tehdä yhdessä paikassa. (Selenium 2023.)



Kuvio 3. Modulaarinen suunnittelumalli (Kaushik 2021)

Page Object Model -suunnittelumallin avulla Mielalle kehitettävä automatisoitu regressiotestaus saadaan modularisoitua. Modularisoinnin avulla saavutetaan parempi ylläpidettävyys. Mallin soveltaminen suunnitellaan testitapauksiin sopivaksi niin, että automaatio säilyy mahdollisimman helposti laajennettavana.

4.10 Keyword Driven Framework -viitekehys

Keyword Driven Framework on toiminnallisen automaatiotestauksen viitekehys, jossa testitapaukset jaetaan neljään eri osaan, jotta koodaus voidaan erottaa testitapauksista ja testausvaiheista paremman automaation varmistamiseksi (Hamilton 2024c). Keyword Driven -testauksen tarkoituksena on erottaa avainsanat, toiminnot ja käytetty data automatisoiduista testitapauksista. Avainsanat tallennetaan erilliseen tiedostoon, joka mahdollistaa tehokkaamman uudelleenkäytävyyden ja helpottaa ylläpitoa. Avainsanapohjaisen suunnittelun ajatuksena on jakaa testit pienempiin komponentteihin, kuten testivaiheisiin, objekteihin ja toimintoihin. (Jose 2021, 94.) Avainsanalähtöisessä testauksessa tunnistetaan ensin joukko avainsanoja, jonka jälkeen avainsanoihin liitetään niihin liittyvä tapahtuma tai toiminto. Tällöin jokainen testauksessa tapahtuva toiminto, kuten selaimen avaaminen tai sulkeminen, hiiren napsautus tai näppäimen painallus, kuvataan avainsanalla, kuten esimerkiksi "openbrowser" tai "click". (Hamilton 2024c.)

Mielan verkkokaupan regressiotestauksen automatisointi pyritään toteuttamaan Page Object Modelia soveltaen. Jokaiseen sivuobjektiin tallennetaan elementtien lisäksi metodit, joille annetaan kullekin oma avainsana. Näitä metodeja kutsutaan testitapauksissa aina tarpeen vaatiessa omalla avainsanallaan. Avainsanalla kutsuttava metodi suorittaa sille asetetun toiminnon, kuten esimerkiksi "open chrome", "adjust screen size" tai "log user in". Tämän lisäksi automatisointiin tul- laan käyttämään avainsanapohjaista työkalua, jonka avulla testien kukin yksittäinen toiminto, kuten hiiren klikkaus, kutsutaan avainsanalla erillisestä kirjastosta.

4.11 Automatisoinnin työkalut

Robot Framework on yleinen avoimeen lähdekoodiin perustuva ilmainen automaatiotyökalu, jota voidaan käyttää esimerkiksi testien automatisointiin. Robot

Framework on avoin, laajennettavissa ja integroitavissa käytännössä mihin tahansa muuhun työkaluun tehokkaiden sekä joustavien automaattioratkaisujen luomiseksi. Robot Frameworkissa käytetään ihmisen luettavissa olevia avainsanoja, joiden avulla kutsutaan metodeja erillisistä kirjastoista. Kirjastojen avulla Robot Frameworkin ominaisuudet ovat laajennettavissa toimimaan erilaisissa ympäristöissä. (Robot Framework 2024a.)

Browser Library on Robot Frameworkin kirjasto, joka on kehitetty nykyisille verkkoselaimille. Kirjasto käyttää JavaScript-pohjaista tekniikkaa nimeltä Playwright. Playwright on suoraan suoraan yhteydessä selaimen ohjelmointirajapintaan (API), ja sillä on täysi kontrolli selaimen ja sen sisältöön. Käyttäjillä on siis täysi pääsy kaikkiin sivun JavaScript-objekteihin. Browser Library tukee myös testausta erilaisilla laitteilla, kuten mobiililaitteilla. (Robot Framework 2024b.)

Mielan verkkokaupan regressiotestauksen automatisoinnin työkaluksi päätettiin Robot Framework Browser Libraryllä opinnäytetyön tekijän omakohtaisen kokemuksen ja työkalun maksuttomuuden perusteella. Browser Library avaa pääsyn suoraan selaimen APIin ja tukee myös testausta erilaisilla laitteilla ja selaimilla, jonka myötä automaation käyttö on myös laajennettavissa.

4.12 Testiympäristön suunnittelu

Testiympäristöllä tarkoitetaan ohjelmistojen ja laitteistojen muodostamaa kokonaisuutta, joka tukee testien suorittamista (Hamilton 2024d). Sovelluksen asianmukaista testaamista varten tulee testiympäristöksi kopioida sovelluksen kehitysympäristö. Tämä tarkoittaa, että testiympäristön tulisi käyttää web-palvelimia, sovelluspalvelimia ja tietokantapalvelimia, jotka ovat identtisiä tuotantolaitteiston kanssa. (Myers ym. 2012, 145.)

Mielan verkkokaupan regressiotestauksessa yhdistyy järjestelmätestauksen ja hyväksymistestauksen piirteitä. Testaus on tarkoituksenaan suorittaa verkkokaupan tuotantoympäristössä, jotta se vastaa mahdollisimman hyvin hyväksymistestauksen tarpeita. Tämä lähestymistapa mahdollistaa myös testauksen suorittamisen ilman erillisen ympäristön ylläpitotarvetta, eikä näin ollen vaadi sen osalta resursointia.

4.13 Käyttötapausten suunnittelu

Käyttötapaustapaus on skenaario, joka kuvaa järjestelmän käyttöä toimijan toimesta tietyn tavoitteen saavuttamiseksi. Toimija on käyttäjä, joka ottaa roolin suhteessa järjestelmään ja pyrkii käyttämään järjestelmää saavuttaakseen jotain merkityksellistä tietyssä kontekstissa. Käyttötapausten avulla voidaan tallentaa järjestelmän toiminnalliset vaatimukset käyttäjän näkökulmasta ja niitä voidaan hyödyntää esimerkiksi vaatimustenmäärittelyprosessissa sekä testitapausten kehittämisessä. (Copeland 2003, 132–133.) Kate Brushin (2022) mukaan käyttötapausten menetelmässä kuvataan kaikki käyttäjän suorittamat vaiheet jonkin tietyn aktiviteetin suorittamiseksi. Jokaisen käyttötapausten tulisi hänen mukaansa sisältää ainakin kolme olennaista elementtiä:

- Toimija – Toimijalla tarkoitetaan järjestelmän käyttäjiä, jotka voivat olla yksittäisen henkilön sijaan myös ryhmä ihmisiä, jotka ovat vuorovaikutuksessa prosessin kanssa.
- Tavoite – Tavoitteella tarkoitetaan onnistunutta lopputulosta, joka päättää prosessin.
- Järjestelmä – Järjestelmällä tarkoitetaan prosessia ja vaiheita, joiden avulla tavoite saavutetaan.

Näiden lisäksi käyttötapausta kirjoittaessa seuraavien elementtien käyttö on harjoittelun arvoista (Brush 2022):

- Sidosryhmät – Sidosryhmillä tarkoitetaan järjestelmän toiminnasta kiinnostuneita tai siihen sijoittaneita tahoja.
- Edellytykset – Elementit, joiden on toteuduttava ennen kuin käyttötapausta voi tapahtua.
- Liipaisimet – Liipaisimilla tarkoitetaan tapahtumia, jotka aloittavat käyttötapausten.
- Lopputulema – Lopputulemalla tarkoitetaan tapahtumia, mitä järjestelmän tulisi olla suorittanut käyttötapausten loppuessa.

Brushin (2022) mukaan käyttötapausten kirjoitusprosessi (kuvio 4) sisältää kuusi askelta. Ensimmäisessä askeleessa jokainen järjestelmän käyttäjä tunnustetaan ja profiloidaan. Tämä sisältää jokaisen järjestelmän kanssa vuorovaikutuksessa olevan käyttäjän roolin. Toisessa askeleessa määritellään yksi näistä käyttäjistä ja hänen tavoitteensa suhteessa järjestelmään. Jokaisesta tavoitteesta tulee käyttötapaus. Kolmannessa askeleessa kuvataan kunkin käyttötapausten tapahtumakulku tavoitteen saavuttamiseen asti. Neljäntenä vuorostaan tarkastellaan mahdollisia vaihtoehtoisia tapahtumakulkuja sekä laajennetaan niiden pohjalta käyttötapausta. Tämän jälkeen tunnustetaan käyttötapausten yhteisiä piirteitä ja luodaan mahdollisia yhteisiä käyttötapausta. Tässä askeleessa myös kirjoitetaan kullekin käyttötapauselle kuvaus. Viimeisessä askeleessa tehtävänä on toistaa jokaisen tunnustetun käyttäjän osalta askeleet 2–5.



Kuvio 4. Käyttötapausten kirjoitusprosessi (Brush 2022)

Mielan verkkokaupan käyttötapausten suunnitteluun pyritään soveltamaan käyttötapausmetodologiaa sekä Mielan ennakkoon kartoittamia verkkokaupan asiakaspolkuja. Käyttötapausta tullaan käyttämään testattavien toiminnallisuuksien tunnistamiseen ja ne laaditaan rajattuina, jotta niitä voidaan ylläpitää olemassa olevilla resursseilla. Käyttötapausta pidetään opinnäytetyön ja Mielan verkkokaupan kontekstissa myös riittävänä toimimaan testisuunnitelmana.

4.14 Testiskenaarioiden suunnittelu

Olha Holotan (2023) mukaan testiskenaariot ovat osa käyttötapausten dokumentaatiota. Testiskenaarioissa keskitytään toimiin, joita käyttäjä voi tehdä järjestelmän avulla. Niiden tavoitteena on kuvata, miten loppukäyttäjä on vuorovaikutuksessa järjestelmän kanssa. Testiskenaarioiden avulla varmistetaan, että kaikki sovelluksen tarjoamat toiminnallisuudet toimivat oikein. Testiskenaariot luodaan käyttötapausten pohjalta hahmottelemalla positiiviset ja negatiiviset skenaariot kaikkien käyttötapauksista ilmenevien järjestelmän toiminnallisuuksien osalta. Testiskenaariot ovat yksirivisiä, kuten esimerkiksi ”tarkista kirjautumistoiminnallisuus”.

Verkkokaupan testiskenaariot on tavoitteena tunnistaa Mielan kanssa yhdessä määriteltyjen käyttötapausten avulla. Testiskenaarioiden avulla saadaan luotua kattava perusta testitapausten suunnitteluun.

4.15 Testitapausten suunnittelu

Kate Brushin (2020) mukaan testitapauksella tarkoitetaan joukkoa järjestelmälle suoritettavia toimenpiteitä, joiden avulla voidaan määrittää täyttyvätkö vaatimukset ja toimiiko järjestelmä odotetulla tavalla. Testitapauksia voidaan käyttää vaiheittaisina ohjeina kutakin järjestelmätestiä varten. Niissä määritellään, mitä järjestelmän testaamiseksi on tehtävä, mukaan lukien siinä suoritettavat vaiheet, siihen syötetyt tiedot sekä testitapauksen suorituksen aikana odotetut tulokset. Testitapauksiin sisältyvät komponentit ovat nimi, ID, tavoite, viitteet, edellytykset, testausjärjestelyt, testiaskeleet ja odotetut tulokset.

Testin nimellä tarkoitetaan otsikkoa, joka kuvaa testattavan toiminnallisuuden tai ominaisuuden. Testin ID on puolestaan tyypillisesti kirjaimia ja/tai numeroita sisältävä identifikaattori, jonka avulla testitapauksia voidaan ryhmitellä joukkoihin. Tavoitteeseen kuvataan yhdellä tai kahdella lauseella se, mitä testin on tarkoitus todentaa. Viitteet ovat taas linkkejä käyttäjätapauksiin, määrittelyihin tai vaatimukseen, jotka testin odotetaan todentavan. Edellytyksillä puolestaan tarkoitetaan niitä välttämättömiä asioita, joiden pitää toteutua, jotta testi voidaan suorittaa. Testausjärjestelyissä määritellään ne seikat, joita vaaditaan testitapauksen

oikeanlaiseen suoritukseen, kuten esimerkiksi sovelluksen versio, käyttöjärjestelmä sekä tietoturvamääritykset. Testiaskeleet ovat kuvauksia yksittäisistä toimista, jotka on suoritettava testin loppuun saattamiseksi. Odotetut tulokset ovat puolestaan hahmotelma siitä, miten järjestelmän pitäisi reagoida kuhunkin testi-askeleeseen.

Opinnäytetyössä testitapaukset pyritään suunnittelemaan testiskenaarioiden pohjalta niin, että jokainen skenaario tulee testatuksi kokonaisvaltaisesti. Testitapausten suunnittelussa otetaan huomioon se, että ne tukisivat automatisointia mahdollisimman hyvin.

5 TESTAUKSEN TOTEUTUS

5.1 Käyttötapausten ja testiskenaarioiden toteutus

Työn toteuttaminen aloitettiin suunnittelemalla liiketoiminnan kannalta kriittisimmille toiminnoille käyttötapaukset. Tätä varten suunniteltiin tutkimustapaukseen sopiva pohja (taulukko 1). Valmis pohja auttaa yhteneväisten ja selkeiden käyttötapausten suunnittelussa. Suunnittelussa pyrittiin myös huomioimaan helppo ylläpidettävyys pitämällä käyttötapaukset mahdollisimman yksinkertaisena ilman, että olennaisesta tiedosta luovutaan.

Taulukko 1. Pohja käyttötapausten suunnitteluun

| USE CASE | |
|---------------|---|
| ID | [Unique identifier for the use case] |
| DESCRIPTION | [A brief description of the purpose of the use case] |
| ACTOR | [The actor initiating the use case] |
| PRECONDITIONS | [Conditions that must be true before the use case begins] |
| GOAL | [The objective or desired outcome of the use case] |
| SYSTEM | <p>STEPS AND SCENARIOS:</p> <p>[The actor's first step towards achieving the goal]</p> <p>[A scenario that must occur within the system to enable the actor to take the step]</p> <p>[A second scenario that must occur within the system to enable the actor to take the step]</p> <p>[An alternative scenario that must occur within the system to enable the actor to take the step]</p> |

| | |
|--|--|
| | <p>[A second alternative scenario that must occur within the system to enable the actor to take the step]</p> <p>[The actor's second step towards achieving the goal]</p> <p>[A scenario that must occur within the system to enable the actor to take the step]</p> <p>[A second scenario that must occur within the system to enable the actor to take the step]</p> <p>[An alternative scenario that must occur within the system to enable the actor to take the step]</p> <p>[A second alternative scenario that must occur within the system to enable the actor to take the step]</p> <p>[Fill in all needed steps and scenarios]</p> |
|--|--|

Kukin käyttötapaus yksilöitiin uniikin ID:n avulla, jolloin niiden seurattavuus ja hallittavuus saatiin paranemaan. Tämän jälkeen kukin käyttötapaus kuvattiin ytimekkäästi niin, että kuvauksesta selviää käyttötapausten tarkoitus ilman ylimääräistä informaatiota. Kuvauksen jälkeen käyttötapausten nimettiin toimija, joka opinäytetyön kontekstissa on verkkokaupan asiakas. Ennakkoehdoin kirjattiin asiat, joiden pitää olla toteutunut ennen kuin käyttötapaus voi alkaa. Tässä työssä käyttötapaukset pilkottiin osiin, jotta ne pysyisivät selkeämpinä ja helpommin hallittavina. Pilkkomisen takia yhden käyttötapausten suorittaminen saattoi taten olla ennakkoehtona toiselle. Tavoitteeseen kuvattiin tavoiteltu lopputulos, eli se mitä verkkokaupan asiakkaan tulisi saavuttaa käyttötapausten lopuksi.

Asiakkaan ottamat askeleet verkkokaupassa kohti tavoitetta kuvattiin askelten avulla. Jokainen askel kuvaa asiakkaan tekemää toimintaa, kuten esimerkiksi "asiakas voi hakea tuotteita hakukentän avulla". Askelia luotiin niin paljon, kuin käyttötapausten onnistunut suorittaminen vaati huolehtimalla kuitenkin siitä, että yhdestä käyttötapauksesta askeleineen ei tule liian pitkä. Tämän jälkeen kukin askel jaettiin skenaarioihin, jotka kuvaavat askeleen ottamiseen tarvittavia toimia

ohjelmiston näkökulmasta. Skenaariot sisällytettiin käyttötapaukseen, jotta niiden asianyhteys pysyy mahdollisimman selkeänä. Yksi skenaario aikaisemmalle esimerkille askeleesta voisi olla esimerkiksi ”hakukenttä on saavutettavissa”. Toisinaan asiakas voi suorittaa yhden askeleen kahta tai useampaa eri reittiä, joten jokaiselle askeleelle mietittiin myös vaihtoehtoisia skenaarioita.

5.2 Testitapausten toteutus

Testitapausten suunnitteluun laadittiin käyttötapauksien tapaan pohja (taulukko 2). Jokainen käyttötapauksissa kuvattu skenaario purettiin testitapauksiksi, joita muodostui skenaariosta riippuen yksi tai useampi. Testitapaukset pyrittiin pitämään sopivan pituisina, jotta niiden ylläpidettävyys ei kärsisi. Samalla testitapaukset suunniteltiin toisistaan riippumattomiksi niin, että yhden testitapauksen suorittamisen epäonnistuessa se ei vaikuta muihin testitapauksiin. Näiden keinojen avulla automaatiosta saatiin vakaampi, selkeämpi sekä helpommin ylläpidettävä.

Taulukko 2. Pohja testitapausten suunnitteluun

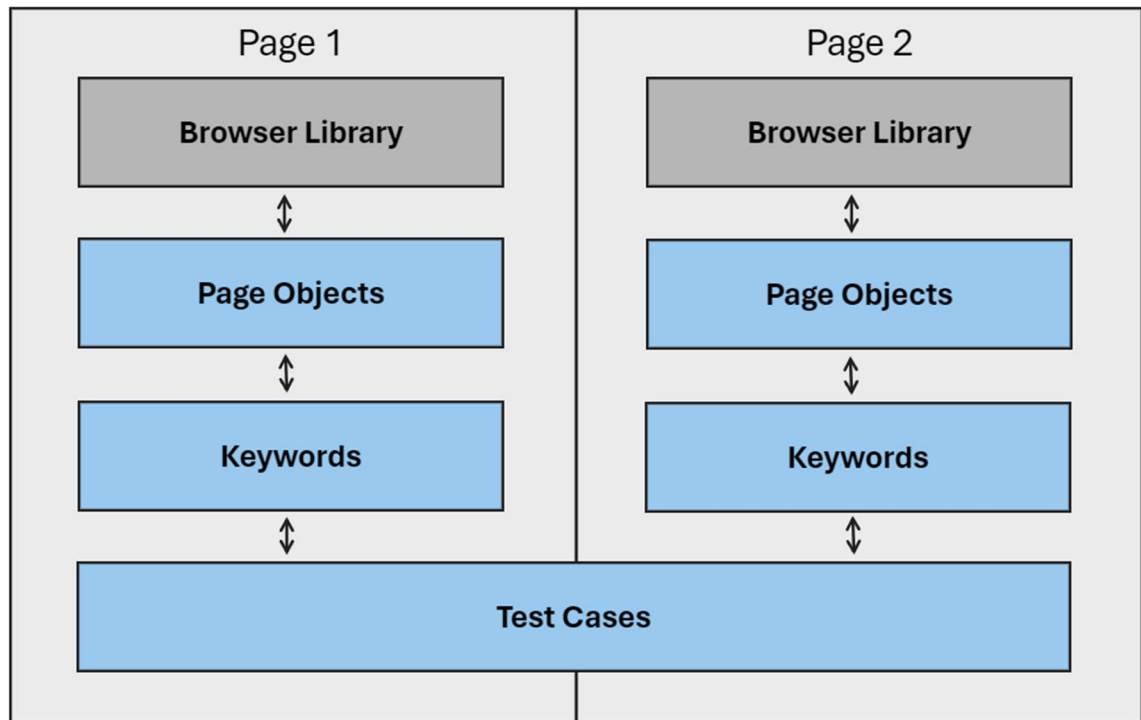
| TEST CASE | |
|---------------------|---|
| ID | [Unique identifier for the test case] |
| RELATED USE CASE(S) | [Unique identifier for the use case(s) associated with this test case] |
| TITLE | [Unique title for the test case] |
| DESCRIPTION | [A brief description of the purpose of the test case] |
| STEPS | [The first step of the test automation towards completing the test case] [The second step of the test automation towards completing the test case] [Fill in all needed steps] |

Testitapausten suunnittelu aloitettiin antamalla testeille uniikit ID:t, jotka toimivat samalla tunnisteena automaatiossa. Tämän lisäksi testitapauksessa viitattiin siihen liittyvään käyttötapaukseen, jotta asiayhteys pysyy selkeänä. Otsikko suunniteltiin testiautomaatioon sopivaksi, jotta sitä voidaan käyttää sellaisenaan testien automatisoinnissa. Tavoitteena oli, että automaatiosta jokaisen suorituksen jälkeen saatava raportti on mahdollisimman helppo ymmärtää. Kuvaukseen avattiin testitapauksen tarkoitusta, eli mitä testin on tarkoituksena verifioida. Askeleet suunniteltiin otsikon tapaan automaatioon soveltuvaksi niin, että yksi askel on mahdollista suorittaa yhden avainsanan puitteissa. Askelia tehtiin niin monta, kuin kunkin testitapauksen suorittaminen vaati, pyrkimällä kuitenkin pitämään niiden määrä alle 10 askeleessa.

5.3 Testien automatisointi

Testien automatisointiin käytettiin Robot Frameworkia sekä siihen suunniteltua Browser-kirjastoa. Automatisointi saatiin opinnäytetyön puitteissa laajennettua koskemaan useita erilaisia laitteita ja selaimia, kuten Android puhelinta ja tietokonetta Chrome-selaimella, sekä iPhonea Opera-selaimella. Tämä onnistui hyödyntämällä Browser-kirjastoon sisältyvää valmista luetteloja erilaisista laitteista, jonka avulla automaatio asettaa käytettävä näyttö oikean kokoiseksi ja valitsee selaimelle laitekohtaisen käyttäjäagentin.

Testien automatisointi suunniteltiin modulaariseksi Page Object Modelin avulla niin, että siinä on kolme tasoa (kuvio 5). Ylimmässä tasossa sivuobjektit hyödyntävät Browser-kirjaston valmiita avainsanoja, kuten "click" tai "wait until element is visible", ja ovat niiden avulla vuorovaikutuksessa suoraan verkkosivun kanssa. Jokaiselle testien kattamalle verkkosivulle tehtiin oma Page Object -tiedosto.



Kuvio 5. Testien automatisoinnin modulaarisuus

Seuraavassa tasossa sivuobjektit koottiin Keyword-tiedostoon niin, että yksi avainsana suorittaa yhden testitapauksissa määritellyn askeleen. Kullekin sivulle tehtiin oma Keyword-tiedosto, joka hyödyntää pääsääntöisesti vain kyseisen sivun Page Object -tiedostoa. Tämä paransi automaation selkeyttä ja vaikuttaa näin ollen myös ylläpidettävyyteen.

Alimmassa tasossa eri Keyword-tiedostoihin tallennetut avainsanat koottiin testitapausten alle askeliksi Test Cases -tiedostoon. Yksi testitapaus voi siis hyödyntää eri tiedostoihin tallennettuja avainsanoja testiaskeleina.

Automatisoidut testitapaukset suunniteltiin toisistaan riippumattomiksi, jotta yhden testin suorituksen epäonnistuminen ei vaikuttaisi muiden testien suoritukseen. Tämä toteutettiin tekemällä kaikille testeille yhteisiä avainsanoja, joita käytetään jokaisen testin alussa ja lopussa. Näiden avainsanojen avulla voidaan esimerkiksi avata selaimeen uusi sivu, valita käytettävä laite tai sulkea selain. Tämän lisäksi usealla sivulla toistuville elementeille tehtiin omat Page Object- ja Keyword-tiedostot, joiden avulla vältettiin sivuobjektien ja avainsanojen toistaminen useassa eri tiedostossa.

6 POHDINTA

Opinnäytetyön toteutus vastasi mielestäni hyvin niin päätutkimuskysymykseen kuin apututkimuskysymykseenkin. Dynaamiseen verkkokauppaan saatiin suunniteltua ja toteutettua ylläpidettävät automatisoidut testitapaukset. Myös regresiotestaus saatiin suunniteltua kohtuullisella työllä laajennettavaksi ja automaatiota tukevaksi.

Käyttötapauksien askeleet saatiin purettua skenaarioiksi, joiden ansiosta kunkin toiminnallisuuden testaaminen saatiin katettua hyvin. Myös testitapaukset saatiin toteutettua automaatiota tukeviksi ja toisistaan riippumattomiksi. Käyttö- ja testitapausten suunnitteluun luodut pohjat saatiin kompakteiksi niin, että ne kuitenkin kattavat kaiken tarpeellisen informaation. Lisäksi testiskenaariot saatiin sisällytettyä käyttötapauksiin, jolloin niiden asianyhteys säilyy selvänä ja informaatio pysyy paremmin hallittavissa.

Automatisoitu testaus saatiin toteutettua modulaariseksi sitä tukevaksi suunniteltujen testitapausten avulla. Automaatio myös säilyi selkeän rakenteen sekä valmiiden avainsanojen ansiosta helposti ylläpidettävänä ja laajennettavana. Alkuperäisistä tavoitteista poiketen automaatio saatiin myös tukemaan erilaisia laitteita ja selaimia, jonka ansiosta kattavuutta saatiin parannettua jo opinnäytetyön yhteydessä huomattavasti.

Tapaustutkimuksen tutkimusote tuki tämän opinnäytetyön tavoitteita ja toteuttamista hyvin. Työn tekeminen vaati laaja-alaista aiheeseen perehtymistä, sillä testauksen automatisointia käsittelevien opinnäytetöiden yleisyydestä huolimatta vastaavia ylläpidettävyyteen keskittyviä opinnäytteitä en löytänyt. Tutkimuksellisen kehittämisen keinoista oli suurta apua työn tekemisessä ja niiden avulla saatiin toteutettua mahdollisimman vähän toimeksiantajan resursseja vaativa lopputulos.

Erillistä tuotantoympäristöä vastaavaa testiympäristöä ei saatu opinnäytetyön yhteydessä toteutettua, joten testaaminen jouduttiin toteuttamaan verkkokaupan tuotantoympäristöön. Jatkotutkimuskysymyksenä toimisi näin ollen se, miten verkkokauppa-alustaan saa toteutettua tuotantoympäristöä vastaavan testausympäristön niin, että se on mahdollisimman helppo ylläpitää.

Automatisoitu testaus voi viedä huonosti suunniteltuna ja toteutettuna pohjan siitä saaduilta hyödyiltä. Tästä johtuen opinnäytetyöstä voivat hyötyä kaikki automatisoitua testausta suunnittelevat, sen parissa työskentelevät tai sen ylläpidettävyydestä kiinnostuneet tahot.

LÄHTEET

Aaltonen, A. 2019. Verkkokaupan regressiotestauksen automatisointi. Opinnäytetyö, Karelia ammattikorkeakoulu. Viitattu 17.01.2024 <https://urn.fi/URN:NBN:fi:amk-2019090918319>.

Anttila, P. 2014. Tutkimisen taito ja tiedon hankinta. Metodix – metoditietämystä kaikille. Viitattu 20.01.2024 <https://metodix.fi/2014/05/17/anttila-pirkko-tutkimisen-taito-ja-tiedon-hankinta/>.

Brush, K. 2020. Definition: Test Case. TechTarget. Viitattu 28.02.2024 <https://www.techtarget.com/searchsoftwarequality/definition/test-case>.

Brush, K. 2022. Definition: Use Case. TechTarget. Viitattu 28.02.2024 <https://www.techtarget.com/searchsoftwarequality/definition/use-case>.

Carey, S. 2021. What is SaaS? Software as a service defined. InfoWorld 22.06.2021. Viitattu 08.02.2024 <https://www.infoworld.com/article/3226386/what-is-saas-software-as-a-service-defined.html>.

Castro, T. 2016. Verkkokaupan automaatiotestaus. Opinnäytetyö, Jyväskylän ammattikorkeakoulu. Viitattu 17.01.2024 <https://urn.fi/URN:NBN:fi:amk-2016052910611>.

Chai, W. & Casey, K. 2022. Definition: Software as a Service (SaaS). TechTarget. Viitattu 08.02.2024 <https://www.techtarget.com/searchcloudcomputing/definition/Software-as-a-Service>.

Copeland, L. 2003. Practitioner's guide to software testing. Norwood: Artech House.

Holota, O. 2023. Test Scenario. Medium. Viitattu 26.02.2024 https://medium.com/@case_lab/test-scenario-92f39e31c9b1.

Hamilton, T. 2023a. Web Application Testing: How to Test a Website? Guru99. Viitattu 18.02.2024 <https://www.guru99.com/web-application-testing.html>.

Hamilton, T. 2023b. Static Vs Dynamic Testing: Difference Between Them. Guru99. Viitattu 14.02.2024 <https://www.guru99.com/static-dynamic-testing.html>.

Hamilton, T. 2023c. Functional Vs Non-Functional Testing – Difference Between Them. Guru99. Viitattu 18.02.2024 <https://www.guru99.com/functional-testing-vs-non-functional-testing.html>.

Hamilton, T. 2024a. Types of Software Testing (100 Examples). Guru99. Viitattu 20.02.2024 <https://www.guru99.com/types-of-software-testing.html>.

- Hamilton, T. 2024b. Automation Testing. Guru99. Viitattu 20.02.2024
<https://www.guru99.com/automation-testing.html>.
- Hamilton, T. 2024c. Keyword Driven Testing Framework with Example. Guru99. Viitattu 01.03.2024 <https://www.guru99.com/keyword-driven-testing.html>.
- Hamilton, T. 2024d. Test Environment in Software Testing. Guru99. Viitattu 25.02.2024 <https://www.guru99.com/test-environment-software-testing.html>.
- Jose, B. 2021. Test automation: a manager's guide. Swindon, England: BCS Learning and Development Ltd.
- Juuti, P. & Puusa, A. 2020. Laadullisen tutkimuksen näkökulmat ja menetelmät. Helsinki: Gaudeamus.
- Kasurinen, J. 2013. Ohjelmistotestauksen käsikirja. Jyväskylä: Docento.
- Kauppa liitto 2023. Kotimaisen verkko-ostamisen kasvu katkesi – digiostaminen on muutoksessa. 23.3.2023. Viitattu 15.01.2024
<https://kauppa.fi/uutishuone/2023/03/22/kotimaisen-verkko-ostamisen-kasvu-katkesi-digiostaminen-on-muutoksessa/>.
- Kaushik, G. 2021. Page Object Model. ToolsQA 11.11.2021. Viitattu 26.02.2024
<https://toolsqa.com/selenium-webdriver/page-object-model/>.
- Lahtinen, T. 2013. Verkkokaupan käsikirja. Helsinki: Yrityskirjat.
- Leppälahti, J. 2018. Regressiotestauksen automatisointi Robot Frameworkilla. Opinnäytetyö, Haaga-Helia ammattikorkeakoulu. Viitattu 17.02.2024
<https://urn.fi/URN:NBN:fi:amk-201902041892>.
- Myers, G., Sandler, C. & Badgett, T. 2012. The Art of Software Testing. 3., uudistettu painos. New Jersey: John Wiley & Sons, Inc.
- Minkkinen, A. & Keski-Heikkilä, I. 2024. Miela Designroom. Yrittäjien haastattelu 17.1.2024.
- Ojasalo, K., Moilanen, T. & Ritalahti, J. 2015. Kehittämistyön menetelmät: uudenlaista osaamista liiketoimintaan. Helsinki: Sanoma Pro.
- Owen, G. 2022. The crucial role of test scenarios, especially in automation. TechTarget. Viitattu 28.02.2024
<https://www.techtarget.com/searchsoftwarequality/tip/The-crucial-role-of-test-scenarios-especially-in-automation>.
- Robot Framework 2024a. Introduction. Viitattu 20.02.2024
<https://robotframework.org/>.
- Robot Framework 2024b. Browser Library. Viitattu 20.02.2024
<https://robotframework-browser.org/>.

Rungta, K. 2023. Page Object Model (POM) & Page Factory in Selenium. Guru99. Viitattu 26.02.2024 <https://www.guru99.com/page-object-model-pom-page-factory-in-selenium-ultimate-guide.html>.

Selenium 2023. Page object models. Viitattu 26.02.2024 https://www.selenium.dev/documentation/test_practices/encouraged/page_object_models/.

SFS-ISO/IEC 25010:2019. Systems and software engineering. Systems and software Quality Requirements and Evaluation (SQuaRE). System and software quality models. 1. painos. Suomen Standardoimisliitto SFS.

Shopify 2024a. FAQ. Viitattu 25.01.2024 <https://www.shopify.com/free-trial>.

Shopify 2024b. Changelog: What's New at Shopify?. Viitattu 25.01.2024 <https://changelog.shopify.com/>.

Suomen virallinen tilasto (SVT): Väestön tieto- ja viestintätekniikan käyttö 2021. ISSN=2341-8699. 2021, 1. Verkkokauppa murroksessa. Helsinki: Tilastokeskus. Viitattu 15.01.2024 https://www.stat.fi/til/sutivi/2021/sutivi_2021_2021-11-30_kat_001_fi.html.

Vilka, H. 2021. Näin onnistut opinnäytetyössä. Ratkaisut tutkimuksen umpikujiiin. Jyväskylä: PS-kustannus.