Tampereen ammattikorkeakoulu

Verkkosivujen suorituskykytes-taus QAutomate-työkalulla

Aleksi Peltonen

OPINNÄYTETYÖ
Toukokuu 2020

Tieto- ja viestintätekniikka
Sulautetut järjestelmät ja elektroniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tieto- ja viestintätekniikka
Sulautetut järjestelmät ja elektroniikka

PELTONEN, ALEKSI:
Verkkosivujen suorituskykytestaus QAutomate-työkalulla

Opinnäytetyö 25 sivua
Toukokuu 2020

Opinnäytetyössä tutustuttiin verkkosivujen suorituskykytestauksen periaatteisiin ja suunniteltiin suorituskykytestaustoiminnallisuus Python-pohjaiseen QAutomate-työkaluun. Toiminnallisuuden ideana oli hyödyntää aiemmin työkalulla luotuja Robot Framework -käyttöliittymätestejä kuormitustestauksen käyttötapauksina ja näin ollen laajentaa tuotteen käyttömahdollisuuksia. Lopullisena tavoitteena pyrittiin löytämään mahdollisimman toimiva ja edullinen pohjaratkaisu suorituskykytestauksen toteuttamiseksi.

Toiminnallisuudelle määritettiin vähimmäisvaatimukset, ja sen kehittäminen pohjautui yhteistyökumppani Techila Technologiesin toimittamaan esimerkkitoteutukseen, jolla pystyttiin ajamaan yksittäinen käyttöliittymätesti pilviympäristössä. Esimerkkitoteutusta ja työkalun käyttöliittymää muokattiin vaatimusten mukaisesti niin, että toiminnallisuuden käyttäminen on mahdollisimman helppoa.

Toiminnallisuuden testaamiseksi luotiin suorituskykytesti QAutomaten ohjelmistorobottien testaukseen tarkoitetulle dashboard-sivustolle. Testi ajettiin kolme kertaa eri käyttäjämäärillä. Yksittäiset tulokset kerättiin pilvikoneilta, ja niistä koostettiin yhteenvetoraportti. Suorituskykytestin luomiseen käytettiin QAutoflow-työkalua.

Työstä syntynyt toiminnallisuus täyttää sille asetetut alustavat vaatimukset, ja se lisättiin mukaan pysyväksi osaksi QAutomate-työkalua. Toiminnallisuuden avulla työkalun käyttäjä pystyy ajamaan valmiita käyttöliittymätestejä pilvikoneilla haluamillaan käyttäjämäärillä. Lisäksi testeihin voi halutessaan lisätä mittauspisteitä, jotka keräävät ja tallentavat tietokantaan dataa resurssien hakemiseen kuluneesta ajasta. Tuloksia voidaan visualisoida suoraan tietokannasta Grafana-alustan avulla. Työn toteutusvaiheessa kirjattiin ylös jatkokehitysmahdollisuuksia, joita voidaan työstää tulevaisuudessa riittävän testaus- ja laadunvarmistusjakson jälkeen.

Toiminnallisuus toteutettiin QAutomate Oy:lle yhteistyössä Techila Technologies Oy:n kanssa. Toteutuksen lähdekoodi on luottamuksellista tietoa, joten sitä ei esitellä tässä opinnäytetyössä.

Asiasanat: suorituskykytestaus, robot framework, käyttöliittymätestit

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in ICT Engineering
Embedded Systems and Electronics

PELTONEN, ALEKSI:
Testing Website Performance with QAutomate Tool

Bachelor's thesis 25 pages
May 2020

In this thesis the principles of website performance testing were studied and a performance testing functionality for the Python-based QAutomate tool was designed. The idea of the functionality was to utilize already existing user interface tests created with Robot Framework and expand the tools affordance. The tests would be utilized as use cases for load testing. The study also determined the most functional and inexpensive base solution for performance testing implementation.

Minimum requirements were defined for the functionality and its development was based on the example implementation, provided by Techila Technologies. The example implementation could be used to run a single user interface test in the cloud environment. The example implementation and user interface of the tool were modified according to the requirements to make the functionality user-friendly.

For testing purposes, a load test was created for QAutomate's control panel site which is meant for software robot testing. The test was executed three times with different number of users for each run. Individual results were collected from the cloud computers and a summary report was compiled. The load test was created using QAutoflow tool.

The resulting functionality meets the requirements and it was permanently added to the QAutomate tool. Using the functionality the user can run existing user interface tests on cloud computers with desired number of users. In addition, measuring points can be added to the test. These can be used to measure and save timing data regarding the loading of an application's resources to database. The results can be visualized straight from the database with Grafana platform. During the implementation phase, several ideas for further development were written down. These ideas can be addressed in the future.

The functionality was implemented for QAutomate Ltd in cooperation with Techila Technologies Ltd. Source code for the implementation is confidential and therefore not displayed in this thesis.

Key words: performance testing, robot framework, user interface tests

SISÄLLYS

LYHENTEET JA TERMIT

Dashboard	ohjausnäkymä tai hallintapaneeli, visuaalisesti toteutettu näkymä
Framework	kehys, runko tai pohja kehitettävälle asialle, esimerkiksi ohjelmistolle tai hyväksymistestille
HTML	Hypertext Markup Language, hypertekstin kuvauskieli
Javascript	erityisesti web-ympäristöissä käytössä oleva dynaaminen ohjelmointikieli
Jenkins	avoimen lähdekoodin jatkuvan integraation palvelinohjelmisto
JSON	Javascript Object Notation, tiedonvälitykseen tarkoitettu tiedostomuoto
Käyttäjäpolku	vaiheittainen kuvaus reitistä, jota pitkin käyttäjä etenee verkkosivulla kohti päämääräänsä
Lokaattori	komento tai tunniste, jolla viitataan verkkosivuilla esiintyviin elementteihin
PoC	Proof of Concept, konseptitodistus, menetelmän tai idean osoittaminen toteuttamiskelpoiseksi
Pullonkaula	käsite rajoittavalle tekijälle, joka estää järjestelmän toimimisen täydellä kapasiteetilla
Python	korkean tason ohjelmointikieli
RPA	Robotic Process Automation, ohjelmistorobotiikka

1 JOHDANTO

Internetissä on paljon erilaisia verkkopalveluja, jotka helpottavat arkisten asioiden hoitamista. Esimerkiksi pankki- ja veroasiat voidaan hyvin suurelta osin hoitaa sähköisenä asiointina. Tulevaisuudessa verkkopalvelujen kattavuus tulee todennäköisesti vain kasvamaan. Palveluihin kohdistuu odotuksia, joista yksi on nopea vasteaika. Suorituskykytestaus on yksi keino, jolla pyritään etsimään ja korjaamaan verkkopalveluiden ongelmakohtia, jotka aiheuttavat negatiivisesti käyttäjäkokemukseen vaikuttavia viiveitä.

Opinnäytetyössä tutkitaan suorituskykytestauksen merkitystä käyttäjän näkökulmasta sekä kehitetään pohja suorituskykytestaustoiminnallisuudelle QAutomate-työkaluun ja esitetään sen toimintaa esimerkkitestillä. Pohjimmaisena tarkoituksena on parantaa olemassa olevaa tuotetta ja tehdä siitä monipuolisempi testauksen apuväline. Toiminnallisuus jää pysyväksi osaksi tuotetta, ja sitä tullaan jatkokehittämään tulevaisuudessa tarpeiden mukaisesti.

2 SUORITUSKYKYTESTAUS

Ihmisillä on käytössään suurempi valikoima verkkopalveluja kuin koskaan aikaisemmin. Siksi ei olekaan ihme, että niiden käyttöön liittyy tietynlaisia odotuksia ja vaatimuksia. Tavallisesti verkkopalveluissa ei ole aukioloaikoja, joten niiden tulisi olla saatavilla ja käyttövalmiita kellon ympäri. Tämän lisäksi palveluiden pitäisi vastata käyttäjilleen nopeasti, ideaalisesti ilman minkäänlaisia ylimääräisiä viiveitä. Käyttäjän näkökulmasta hitaasti latautuva tai pahimmillaan kaatuileva palvelu on toimimaton palvelu. (Sogeti Finland Oy n.d). Tämä voi yrityksille tarkoittaa käytännössä käyttäjien ja potentiaalisten asiakkuuksien menettämistä.

Suorituskykytestauksella tarkoitetaan menetelmiä ja keinoja, joilla selvitetään järjestelmän kykyä suoriutua tehtävistään kuormituksen alaisena. Suorituskykyvaatimuksiin lukeutuu mm. vasteajat sekä käyttäjä- ja datamäärät. Järjestelmän tulee siis nopean toiminnan lisäksi pystyä käsittelemään vaadittu määrä dataa ja käyttäjiä kaatumatta. Systemaattisella suorituskykytestauksella voidaan tunnistaa järjestelmän toimintaa hidastavia pullonkauloja sekä kartoittaa nykyisen laitteiston riittävyyttä tulevaisuudessa. (Vuori 2010, 6–8). Palveluiden käytön kasvaessa myös resurssitarpeet kasvavat, jolloin testaamaton järjestelmä saattaa aiheuttaa ikäviä seurauksia sanoessaan itsensä irti. Sama koskee myös uusia käyttöönotettavia palveluita. Suorituskykytestaus onkin harmillisen usein vähemmälle huomiolle jäänyt osa-alue järjestelmien kehitysvaiheessa ja laadunvarmistuksessa.

2.1 Suorituskykytestauksen alalajit

Suorituskykytesteihin liittyen puhutaan usein testaustyypeistä tai alalajeista. Suorituskykytestaus voi olla luonteeltaan kokonaisvaltaista tai sitten voidaan keskittyä tiettyjen alalajien testien toteuttamiseen. Kuvassa 1 on esitetty erilaisia suorituskykytestauksen alalajeja.

KUVA 1. Suorituskykytestauksen alalajeja (Vaidya 2019)

Kuormitustestauksella (load testing) tutkitaan järjestelmän kykyä toimia odotetun suuruisen kuorman alaisuudessa. Sitä käytetään enimmäkseen pullonkaulojen löytämiseen ennen järjestelmän julkaisua. Kestävyystestauksessa (endurance testing) järjestelmää kuormitetaan tasaisesti pitkällä aikavälillä, jolloin varmistetaan järjestelmän stabiilisuus. Kestävyystestin kesto vaihtelee muutamista kymmenistä minuuteista jopa kuukausiin. Volyymitestaus (volume testing) keskittyy suuriin datamääriin ja tietokantojen kuormittamiseen. Skaalautuvuustestauksessa (scalability testing) selvitetään, miten tehokkaasti järjestelmä skaalautuu eli reagoi käyttäjämäärien muutoksiin. Piikkitestauksessa (spike testing) kuormituksessa käytetään lyhytkestoisia mutta korkeita käyttöpiikkejä ja tutkitaan järjestelmän vastetta. Stressitestauksessa (stress testing) pyritään löytämään järjestelmän kestämä maksimikuorma ts. kohta, jossa järjestelmä kaatuu tai hajoaa. (Vaidya 2019).

3 TYÖKALUT JA LÄHTÖKOHDAT

3.1 QAutomate

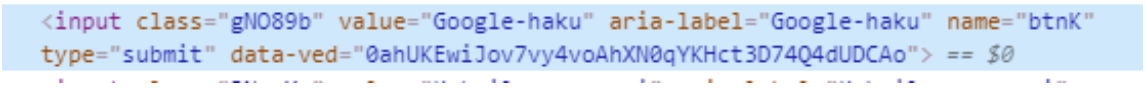
QAutomate on vuonna 2015 perustettu startup-vaiheessa oleva kansainvälisille markkinoille tähtäävä ohjelmistotuoteyhtiö, joka toimii vahvassa yhteistyössä Q-Factoryn kanssa. Q-Factory on vuonna 2010 perustettu testauskonsultointiin keskittyvä yritys, joka työllistää yli 100 henkeä mm. Helsingissä, Tampereella, Oulussa ja Ylivieskassa.

QAutomate tarjoaa yrityksille tuoteratkaisuna webbimaailmassa testauksen automatisointia, RPA-robotteja ja DevOps-konsultointia edullisesti. Asiakkaisiin lukeutuu niin isoja verkko-operaattoreita kuin pienempiä ja paikallisempia yrityksiä. QAutomaten tuoteperhe on keskittynyt kahteen päätuotteeseen: QAutomate ja QAutoRPA, jotka hyödyntävät suosittuja kehitystyökaluja, kuten Python, Robot Framework ja Jenkins.

3.2 QAutoflow

QAutomate-tuote jakaantuu kahteen päätyökaluun, joista ensimmäinen on QAutoflow. QAutoflow on ilmainen laajennus Google Chrome -selaimeen, joka on tarkoitettu automaatiotestien ja ohjelmistorobotiikan käyttäjäpolkujen suunnitteluun. Työkalun toimintaperiaatteena on nauhoittaa käyttäjän (esimerkiksi testin suunnittelijan) toimintaa verkkosivulla ja kerätä talteen olennaiset käyttäjän tekemät toiminnot. Lähtökohtaisesti työkalu seuraa liikkumista verkkosivulla eli klikkauksia ja käyttäjän syötteitä esimerkiksi tekstikenttiin. Haluttaessa voidaan myös verifioida sivulla esiintyviä elementtejä, kuten tekstiä tai kuvia, ottaa talteen arvoja tai suorittaa API-kutsuja ja valmiita Robot Framework -avainsanoja. Toisin sanoen QAutoflowlla luodaan käyttäjäpolku eli flow. Tämä flow on mahdollista muuntaa toimivaksi ja ajovalmiiksi Robot Framework -testiksi maksullista lisenssiä vastaan.

Käyttäjän toiminnan nauhoittaminen perustuu lokaattoreiden tallentamiseen. Lokaattorit ovat komentoja tai tunnisteita, joilla viitataan verkkosivuilla esiintyviin elementteihin. Esimerkiksi napin painamiseksi sille täytyy löytää yksilöllinen lokaattori, jotta painallus voidaan kohdistaa oikeaan kohteeseen. Lokaattori voidaan muodostaa usealla tavalla, tyypillisesti kuitenkin elementin tyypin ja attribuuttien avulla. Verkkosivujen elementtien attribuutteja pääsee vapaasti tutkimaan esimerkiksi Google Chrome -selaimen kehittäjätyökalujen avulla. Kuvassa 2 tarkastellaan Googlen etusivun hakupainikkeen attribuutteja.



KUVA 2. Googlen etusivun hakupainikkeen attribuutteja

Kuvasta huomataan, että kyseessä on input-tyyppinen elementti, jolla on class-, value-, aria-label-, name-, type- ja data-ved-nimiset attribuutit. Näitä kaikkia voidaan hyödyntää lokaattoreiden muodostamisessa. Yksittäiselle elementille voidaan siis muodostaa useita vaihtoehtoja lokaattoreiksi, mutta kaikki niistä eivät välttämättä ole hyviä. Sivuston elementit saattavat sisältää dynaamisia attribuutteja, joiden arvot vaihtelevat aina, kun sivustolla käydään. Tämä tarkoittaa käytännössä sitä, että nauhoitushetkellä tallennetut lokaattorit eivät enää toimi seu-

raavalla kerralla, kun testiä oikeasti ajetaan. Hyvä lokaattori muodostetaan tyypillisesti id-attribuutista, jonka tulisi olla uniikki jokaiselle elementille. QAutoflow tallentaa automaattisesti eri lokaattoreita nauhoitushetkellä. Työkalu pyrkii myöskin järjestämään vaihtoehdot paremmuusjärjestykseen ja tarjoamaan käyttäjälle testin käytettävyyden kannalta parhaan vaihtoehdon ensimmäisenä.

Nauhoituksen päätteeksi työkalu luo vielä yhteenvetoraportin, jossa näkyy tietoja jokaisesta vaiheesta, esimerkiksi mitä on klikattu tai mihin kenttään on syötetty tekstiä. Raportti sisältää lisäksi nauhoitusvaiheessa kaapattuja kuvia selaimesta, joihin on merkitty korostettuna ne elementit, joita on käsitelty. Koko nauhoitusprosessi on myös mahdollista tallentaa videomuodossa.

QAutoflowssa on valittavana kaksi nauhoitusmoodia: regressio (regression) ja tutkiva (exploratory). Regressiomoodissa nauhoitus pysähtyy jokaisen käyttäjän toiminnon jälkeen ja pyytää nimeämään mm. näkymän ja toiminnon kuvauksen. Tutkivassa moodissa näkymän nimet ja kuvaukset muodostuvat automaattisesti verkkosivun elementeistä eli käyttäjä vain liikkuu verkkosivulla niin kuin nauhoitusta ei olisi edes käynnissä. Tutkiva moodi on vaivattomampi ja nopeampi, mutta vaiheiden nimistä tulee mahdollisesti sekavia. Tämä moodi sopiikin paremmin prosessikuvauksen luomiseen. Regressiomoodissa nauhoitus on hieman hitaampaa, mutta vaiheiden nimistä tulee paljon kuvaavampia ja selkeitä muunnettavaa lopullista testiä ajatellen. Tästä syystä tämä moodi on sopivampi testausautomaatioon ja ohjelmistorobotiikan toteuttamiseen.

3.3 QAutorobot

Toinen päätyökalu QAutomate-tuotteessa on QAutorobot. Se on tarkoitettu pääasiassa luotujen testien hallintaan ja ylläpitoon. Työkalu tarjoaa käyttöliittymän, jolla voi mm. muuntaa QAutoflowlla luodun testin ajettavaan muotoon, luoda uusia vaiheita projektiin sekä ajaa Robot Framework -testejä. Myös virheiden etsiminen helpottuu, kun testin voi suorittaa vaikkapa tiedettyyn ongelmakohtaan asti tai yksittäisiä vaiheita voi kokeilla eristetysti tietyssä kohtaa ilman että testiä tarvitsee aina ajaa alusta. Tämä on erityisen hyödyllistä pitkien testien tapauksissa.

3.4 QAutolibrary

QAutolibrary on avoimen lähdekoodin Python-kirjasto, jota QAutoflow ja QAutorobot hyödyntävät. Se sisältää tiedostojen käsittelyyn liittyviä apufunktioita sekä toimii testauskirjastona Robot Frameworkille. QAutolibraryn verkkosivujen testaamiseen liittyvät komponentit pohjautuvat vahvasti Robot Frameworkin testauskirjastoon Selenium Library. Kirjasto pyrkii laajentamaan Seleniumin tarjoamia testauksen ja automaation työkaluja.

3.5 Robot Framework

Robot Framework on suomalaislähtöinen avoimen lähdekoodin hyväksymistestaukseen ja ohjelmistorobotiikkaan käytetty kehys (framework). Se hyödyntää avainsana-tyyppistä rakennetta mallintamaan testejä sekä ohjelmistorobotiikan käyttäjäpolkuja. Robot Framework on kehitetty Python-ohjelmointikielellä, ja siihen on tarjolla lukuisia ulkoisia kirjastoja laajentamaan sen testauskyvykkyyttä. Kirjastot sisältävät valmiita avainsanoja, jotka auttavat testien ja automaation kehittämisessä. Esimerkiksi AppiumLibrary mahdollistaa mobiilikäyttöjärjestelmien (Android ja iOS) testaamisen. (Robot Framework 2020).

3.6 Techila Distributed Computing Engine

Suorituskykytestauksen toteutuksessa hyödynnettiin Techila Technologies Oy:n hajautetun tietojenkäsittelyn alustaa. Alustan avulla otettiin käyttöön pilvikoneita testien ajamista varten. Testit ajettiin pilvikoneilla samanaikaisesti, jolloin niillä voitiin simuloida suurta käyttäjämäärää tai ruuhkautumista verkkosivuilla. Pilvikoneet käyttävät niin sanottuja pre-emptible-resursseja, jolloin niillä on normaalia rajoitetumpi saatavuus. Niiden hinta on kuitenkin kiinteä ja matalampi, eli niiden käyttäminen tulee edullisemmaksi.

4 SUORITUSKYKYTESTAUSTOIMINNALLISUUDEN TOTEUTUS

4.1 Suunnitteluvaihe ja vaatimukset

Suorituskykytestaustoiminnallisuutta alettiin suunnitella maltillisesti. Ensimmäiseksi haluttiin toteuttaa yksinkertainen konseptitodistustyyppinen (Proof of Concept) kokonaisuus, joka olisi helposti testattavissa ja jatkokehitettävissä. Vaatimuksena toiminnallisuuden pitäisi alussa ainakin pystyä

- määrittämään, monellako koneella testiä ajetaan (käyttäjämäärä)
- käynnistämään ja sammuttamaan testiä ajavat koneet pilvessä
- monitoroida testien etenemistä jollakin tavalla
- palauttamaan yksittäisten testien raportit ja yhdistää ne yhteenvedoksi
- tallentamaan tietokantaan mittauspisteiden tuloksia
- esittämään Grafana-alustan avulla visualisointia mittauspisteistä

Techila Technologies tarjosi esimerkkikoodin, jolla yksittäinen mallitesti voitiin lähettää Techilan alustalle ajettavaksi. Esimerkkiä tutkimalla selvitettiin minkälaisia asetuksia ja parametreja työkalulla tulee olla tiedossa ajojen mahdollistamiseksi. Esimerkin ajamiseksi asennettiin Techilan ohjelmistokehityspaketista Python-kirjasto, joka sisälsi myös projektin tilaikkunan, jolla voidaan seurata pilvikoneiden ja testien tilaa.

Pilvessä ajettavista testeistä haluttiin myös kerätä muutakin dataa kuin lopputulos, joten QAutolibrary-kirjastoon luotiin uusi funktio, joka mittaa verkkosivun resurssien lataamiseen kuluvaa aikaa käyttäen javascriptin Resource Timing -rajapintaa. Funktion palauttamat tulokset tallennettaisiin InfluxDB-tietokantaan ohjelmointirajapinnan kautta ja tuloksia visualisoitaisiin Grafana-alustalla.

4.2 Asetustiedosto

Tarvittavat asetukset koottiin JSON-muotoiseen asetustiedostoon (kuva 3), josta työkalu voisi lukea ne ajon käynnistämiseksi.

```
{
"techila_path": "C:\\techila",
"workers": "10",
"image_name": "techilaworker07032020-1583469005262",
"machinetype": "e2-small",
"influx_host": "localhost",
"influx_port": "8086",
"influx_http_mode": "http",
"influx_username": "",
"influx_password": "",
"grafana_host": "localhost",
"grafana_port": "8084",
"grafana_http_mode": "http",
"grafana_username": "admin",
"grafana_password": "admin"
}
```

KUVA 3. Asetustiedoston sisältö

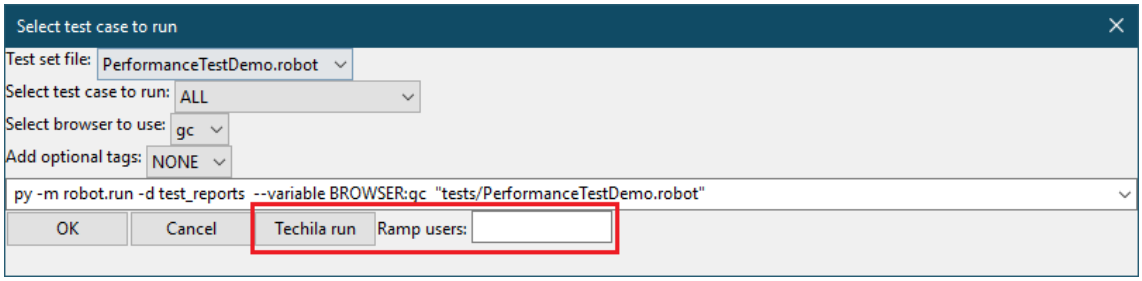
Asetustiedostossa määritettiin polku Techilan ohjelmistokehityspakettiin, käyttäjämäärä, näköistiedoston nimi, pilvikoneiden tyyppi sekä InfluxDB-tietokannan ja Grafanan yhteysparametrit. Näköistiedosto määrittää ympäristön, jossa testi ajetaan, joten sen tulee sisältää kaikki testin tarvitsemat riippuvuudet esimerkiksi ulkoiset kirjastot. Grafana ja InfluxDB asennettiin omalle koneelle, ja asetustiedostoon konfiguroitiin niiden käyttämät palvelinosoitteet ja portit.

4.3 Integrointi QAutorobottiin

Alkuperäistä esimerkkikoodia muokattiin siten, että pilvikoneet osaisivat pakata ja lähettää testien tulokset takaisin paikalliselle koneelle. Tämän lisäksi koodiin lisättiin pilvikoneiden automaattinen sammutus ja palautettujen testiraporttien yhdistäminen ajon päätteeksi. Tässä vaiheessa koodi oli toimiva itsenäinen kokonaisuus ja valmis siirrettäväksi työkalun koodiin omana funktionaan. Siirron yhteydessä tehtiin vielä pieniä parannuksia polkuasetuksiin ja virheenkäsittelyyn.

QAutorobotin testin ajamista edeltävään dialogiin lisättiin painike ja tekstikenttä ajon käynnistämistä varten (kuva 4). Tekstikenttä luotiin korvaamaan käyttäjämäärän määritystä asetustiedoston kautta. Samalla toteutettiin ominaisuus, jossa useita käyttäjämääriä voitaisiin syöttää tekstikenttään pilkulla erotettuna. Näin yhdellä käynnistyksellä voitaisiin ajaa testi useilla käyttäjämäärillä. Suorituskykytestaustoiminnallisuudesta haluttiin tehdä valinnainen, joten työkalun koodi sisältää

tarkistuksen Techilan asennukselle ja piilottaa napin sekä tekstikentän tarvittaessa pois käyttäjän näkyviltä.

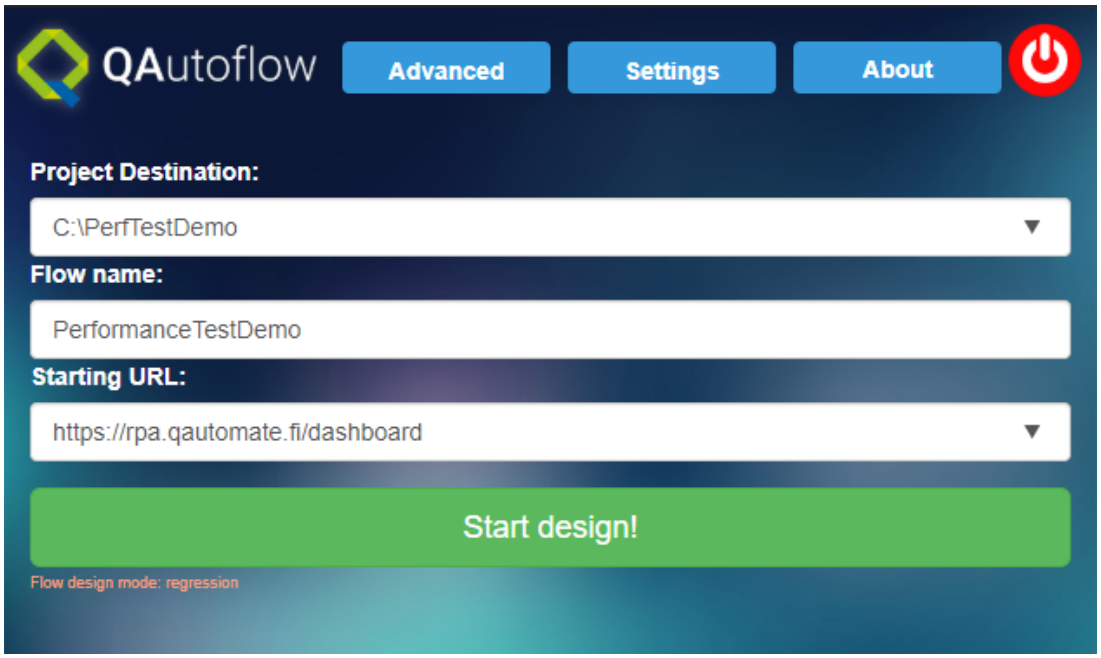


KUVA 4. Pilviajoa varten toteutetut käyttöliittymäelementit

5 KUORMITUSTESTIN LUOMINEN QAUTOMATELLA

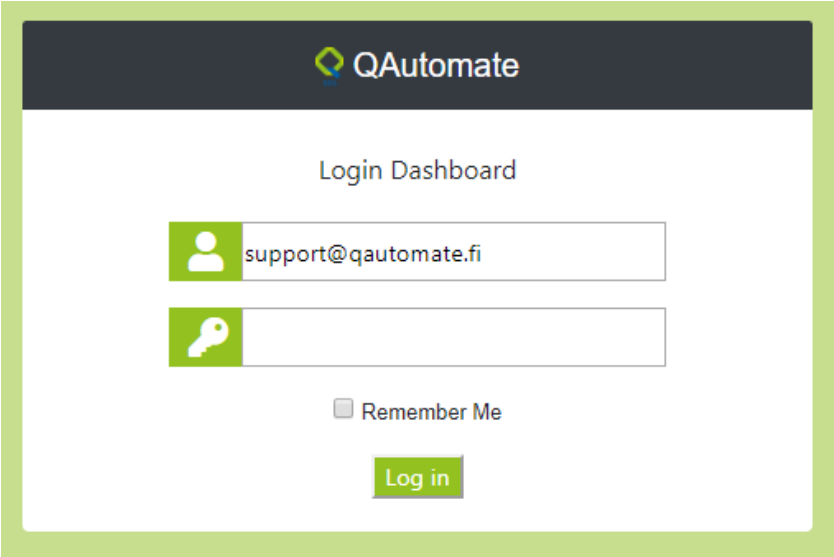
5.1 Käyttäjäpolun (Flown) nauhoittaminen

Suorituskykytestaustoiminnallisuuden testausta varten luotiin kuormitustesti QAutoflow-työkalulla. Testi luotiin QAutomaten RPA-robottien testaukseen tarkoitetulle dashboard-sivustolle. Kyseessä ei ole tuotantoympäristö, joten sivuston palvelin toimii rajoitetulla suorituskyvyllä ja ilman kuormantasaajaa. Työkalu käynnistettiin, jonka jälkeen määritettiin asetukset kohdilleen avaamalla työkalulaajennus selaimen oikeasta yläreunasta. Alussa valittiin projektikansion polku, käyttäjäpolun nimi sekä sivusto jolta nauhoitus aloitetaan (kuva 5). Työkalua käytettiin regressiomoodissa, jotta vaiheiden nimistä saatiin mahdollisimman kuvaavia.



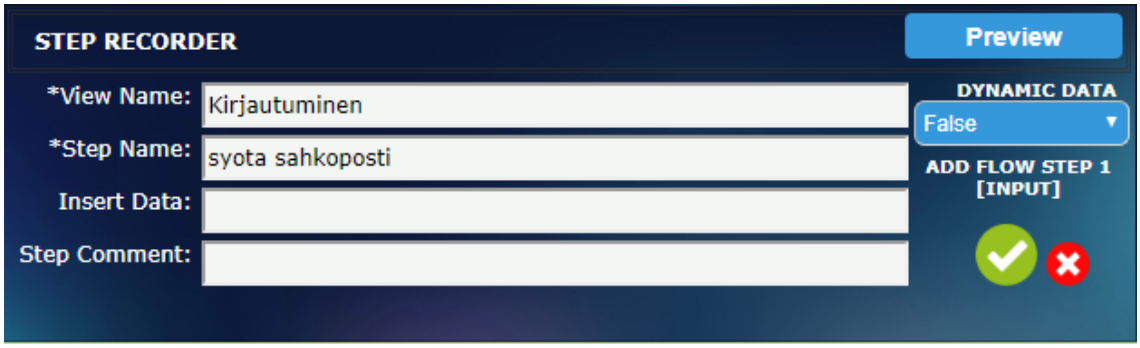
KUVA 5. Alkumääritykset käyttäjäpolun nauhoitusta varten

Nauhoittamisen aloituksen jälkeen dashboard-sivun kirjautumisnäkymä avautui, ja ensimmäinen vaihe tallennettiin. Kirjautumisvaiheessa klikattiin sähköpostikenttä aktiiviseksi ja syötettiin siihen testikäyttöön varattu sähköpostiosoite (kuva 6).



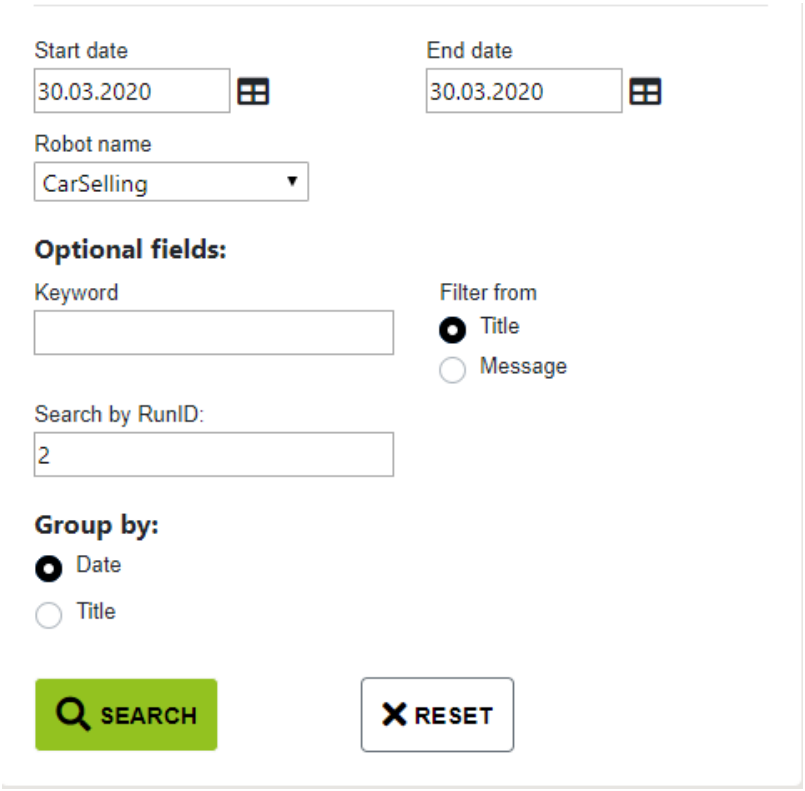
KUVA 6. Sähköpostin syöttäminen kirjautumisvaiheessa

Samaan aikaan työkalu avaa dialogin sivun oikeaan alareunaan. Tähän syötettiin näkymän ja vaiheen nimi (kuva 7).



KUVA 7. Dialogi vaiheen lisäämiseksi

Salasanan syöttövaihe ja sitä seuraavienkin vaiheiden lisääminen tapahtui vastaavalla tavalla. Nauhoituksessa siirryttiin robottien raporttinäkymään, tehtiin yksi haku ja kirjauduttiin ulos. Hakuvaiheeseen (kuva 8) lisättiin kommentti joka testiä muunnettaessa livemonitor-asetuksella muuttuisi mittauspisteeksi. Mittauspisteellä tarkoitetaan kohtaa, jossa kutsutaan aikaisemmin luotua mittausfunktiota.



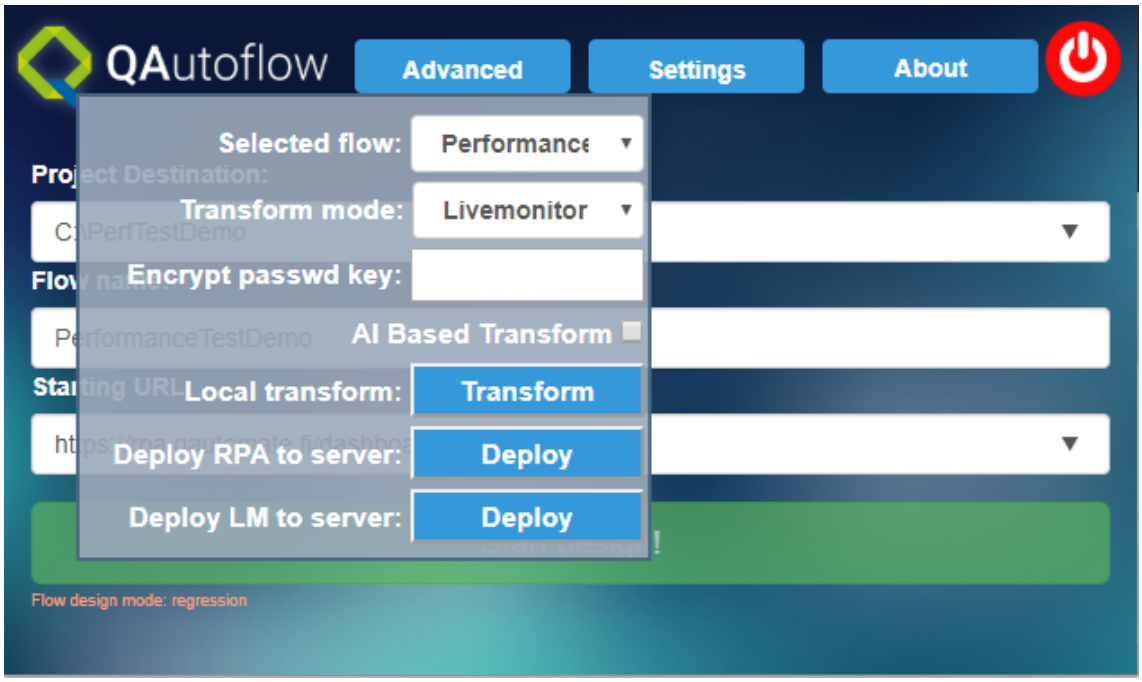
KUVA 8. Raporttinäkymässä toteutettiin yksi haku raporttiarkistosta

Viimeisen vaiheen jälkeen käyttäjäpolun nauhoitus lopetettiin. Käyttäjäpolusta avautui yhteenveto, joka sisälsi lisätietoa vaiheista, kuten toiminnon tyyppi ja vaiheelle annettu nimi (kuva 9).

KUVA 9. Nauhoitetun käyttäjäpolun vaiheet

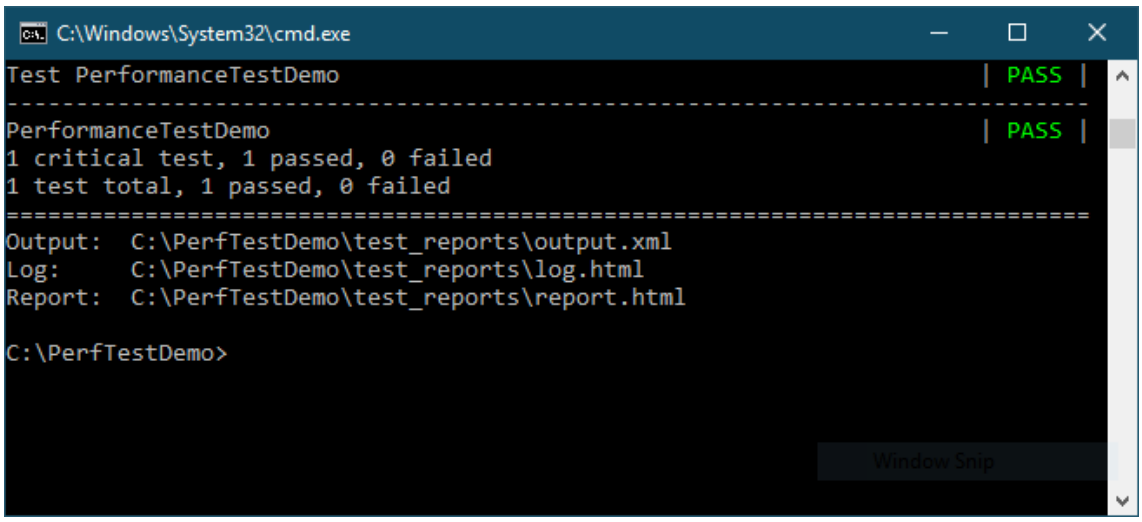
5.2 Testin toiminnan varmistaminen

Käyttäjäpolun nauhoittamisen jälkeen testi muunnettiin ajettavaan muotoon. Tämä tehtiin suoraan QAutoflown advanced-valikosta (kuva 10). Testi muunnettiin livemonitor-asetuksella, jolloin nauhoitusvaiheessa asetettuja kommentteja voidaan hyödyntää mittauspisteinä testin suorituksen aikana. Testi muunnettiin onnistuneesti.



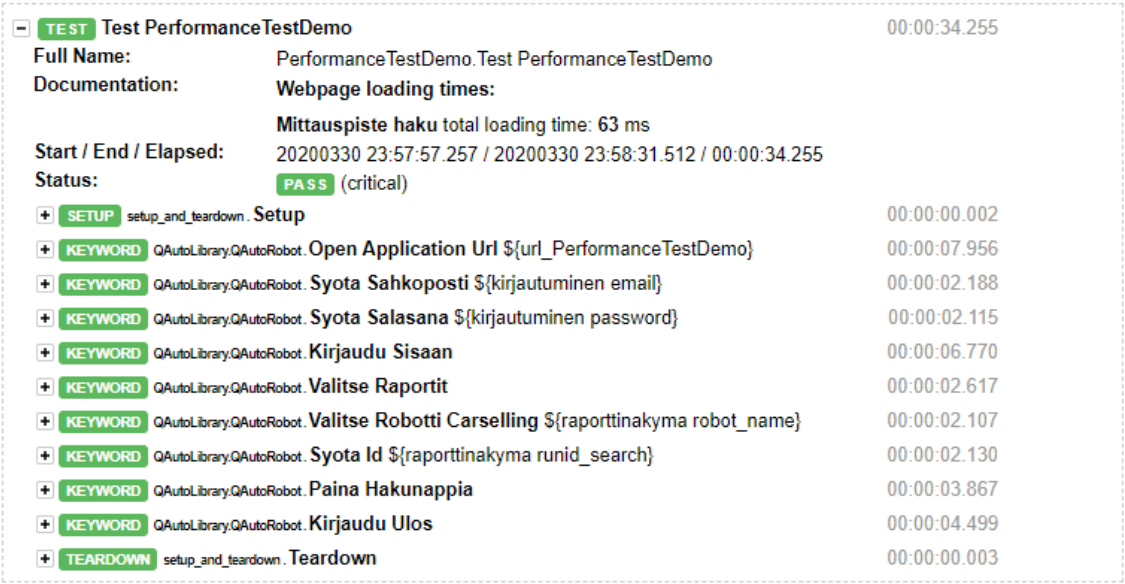
KUVA 10. QAutoflown advanced-valikko

Ennen pilvikoneille lähettämistä testin toiminta varmistettiin ajamalla se paikallisesti omalla tietokoneella. Testin voi ajaa esimerkiksi QAutorobotilla, projektikansioon generoituvalla skriptitiedostolla tai komentokehotteella. Ajo käynnistettiin komentokehotteella, jonka jälkeen tarkastettiin, että testi etenee oikein ja suunnitellusti. Ajon päätteeksi komentokehotteesta oli luettavissa testin lopputulos ja tiedostopolut raportteihin (kuva 11).



KUVA 11. Onnistuneen testin tulos

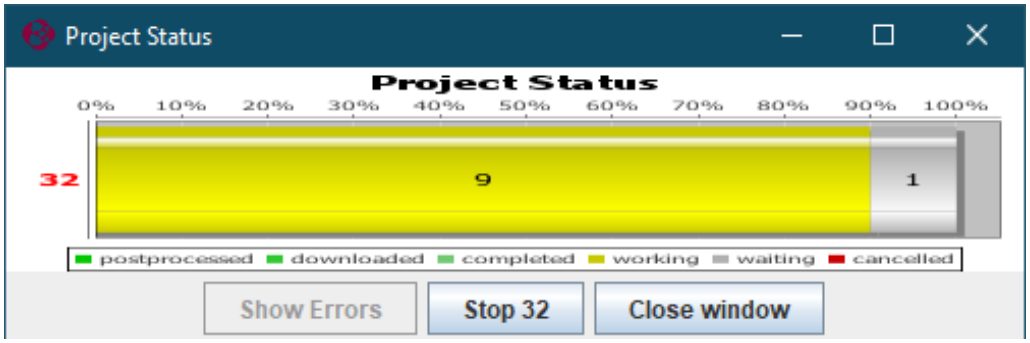
Testin kokonaiskestoa ja yksittäisiin vaiheisiin kulunutta aikaa voidaan tarkastella HTML-muotoisesta lokitiedostosta (kuva 12). Lokista on myös havaittavissa hakuvaiheen mittauspisteen palauttama resurssien lataamiseen kulunut aika. Pilvikoneilla ajettaessa tulokset tallennetaan myös tietokantaan, jolloin niitä voi tarkastella Grafanan avulla.



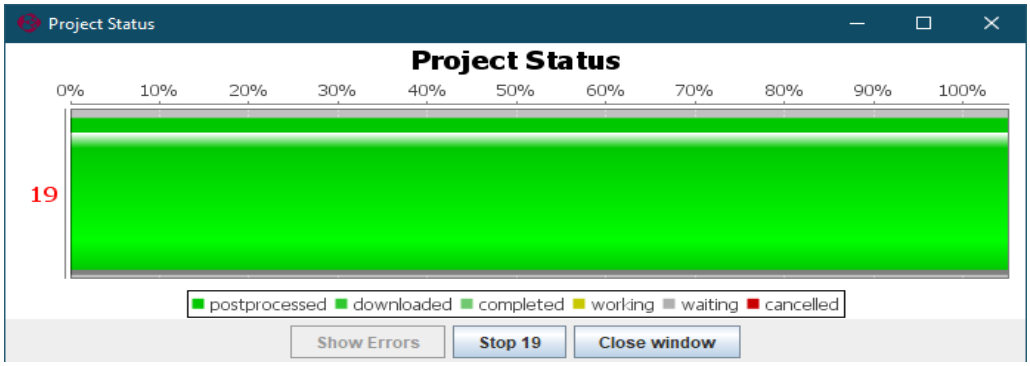
KUVA 12. Statistiikkaa ajetusta testistä

5.3 Testin ajaminen pilvikoneilla

Testin toiminnan varmistamisen jälkeen se lähetettiin ajettavaksi pilvikoneille. Testi päätettiin ajaa käyttäjämäärillä 1, 10 ja 20. Aluksi QAutorobotista valittiin käyttäjämäärät ja käynnistettiin ajot. Koneet nousivat ajovalmiiksi muutaman minuutin kuluessa, ja projektin etenemistä seurattiin tilaikkunasta (kuva 13). Koneet siirtyivät odottavasta tilasta (harmaa) ajovalmiiseen tilaan (keltainen) ja testien loppuessa valmiiseen tilaan (vihreä) (kuva 14). Projektin valmistuttua tilaikkuna sulkeutui automaattisesti. Projektin jokaisen testin yksittäinen tulosraportti ladattiin projektihakemistoon, ja ne yhdistettiin yhteenvedoksi.



KUVA 13. Pilvikoneiden tila on seurattavissa tilaikkunassa (10 käyttäjää)

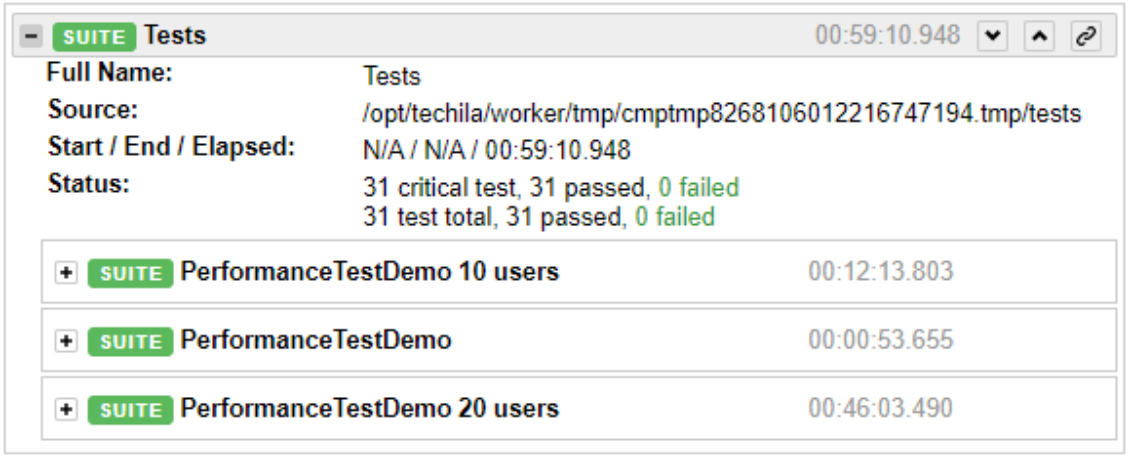


KUVA 14. Tilaikkuna projektin valmistuessa

Projektin valmistuessa testien tuloksista koostettu yhteenveto avautuu selaimessa automaattisesti. Testit on ryhmitelty käyttäjämäärien mukaan omiksi kokonaisuuksiksi.

6 KUORMITUSTESTIN TULOKSET

Ajon päätteeksi avautuneesta raportista (kuva 15) näkyy, että kaikki 31 testiä ajettiin onnistuneesti loppuun eli yhteyden aikakatkaisuja ei ainakaan tapahtunut.



KUVA 15. Testit on ryhmitelty käyttäjämäärien mukaan

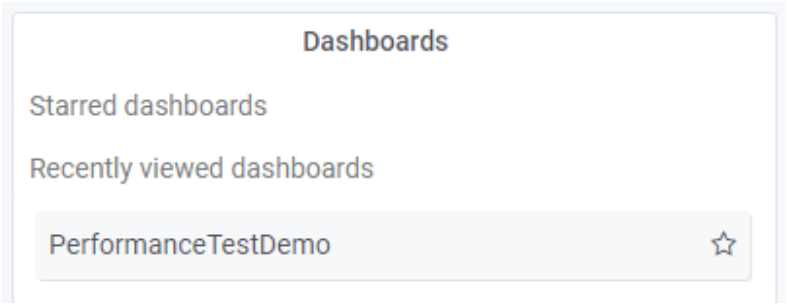
Käyttäjämäärän vaikutus sivuston suorituskykyyn näkyi jo testeihin kuluneesta ajasta. Taulukossa 1 on laskettu yksittäisen testin keskiarvollinen suoritusaika eri käyttäjämäärillä. Yleisimpänä ongelmakohtana havaittiin sisäänkirjautumisvaihe, jonka kesto kolminkertaistui useissa 20:n käyttäjän testeissä verrattuna 1:n käyttäjän testiin.

TAULUKKO 1. Testien suoritusaikoja

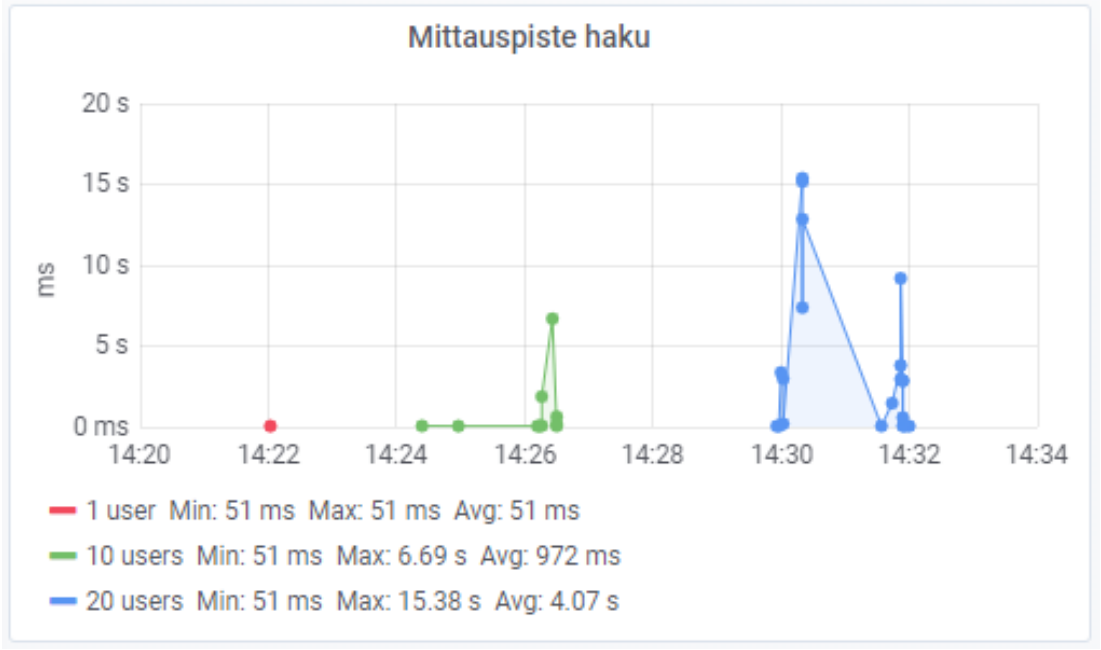
Käyttäjämäärä	Testien yhteenlaskettu aika	Yksittäisen testin suoritusaika (keskiarvo)	Suoritusajan kasvu
1	54 s	54 s	0 %
10	12 min 14 s	1 min 13 s	35 %
20	46 min 3 s	2 min 18 s	156 %

Grafanan avulla päästään tutkimaan tietokantaan kirjattuja mittauspisteiden tuloksia. Grafana-palvelin on käynnissä konfiguraation mukaisesti paikallisella koneella portissa 8084. Tässä tapauksessa palveluun kirjauduttiin kirjoittamalla selaimeen osoite http://localhost:8084/. Työkalu on luonut automaattisesti testien

ajon päätteeksi Grafanaan dashboardin (kuva 16), joka sisältää paneelin mittauspisteistä kerätyistä arvoista (kuva 17).



KUVA 16. Työkalu on luonut Grafanaan dashboardin



KUVA 17. Resurssien hakemiseen kulunut aika mittauspisteessä

Kuviosta on havaittavissa, miten käyttäjämäärän kasvaessa resurssien hakeminen kestää kauemmin. On huomioitavaa, että pilvikoneet ovat saavuttaneet mittauspisteen eri ajankohtina. Suurimmat viiveet resurssien hakemisessa esiintyvätkin silloin kun useamman käyttäjän kuormitus kohdistuu lyhyelle aikavälille, toisin sanoen silloin, kun pisteet ovat horisontaalisesti lähellä toisiaan. Jos sivustoa olisi kuormitettu vielä suuremmilla käyttäjämäärillä, havaittaisiin todennäköisesti yhteyden aikakatkaisuja sekä yksittäisiä epäonnistuneita testejä.

7 POHDINTA

Suorituskykytestaustoiminnallisuuden ensimmäisen versio täyttää suunnitteluvaiheessa annetut tavoitteet. Käyttäjäpolun nauhoitusvaiheessa lisättyjen mittauspisteiden hyödyntäminen tuntuu käyttäjän näkökulmasta vaivattomalta ja helppokäyttöiseltä. Lisäksi niiden tarjoamasta datasta tuotettiin helposti tulkittavia visualisointeja Grafanaan. Toiminnallisuuden lisääminen työkaluun laajentaa sen käyttömahdollisuuksia ja luo uusia tapoja toteuttaa testausautomaatiota valmiiden testien pohjalta.

Jatkokehitysideoita muodostui toiminnallisuuden kehitys- ja testausvaiheessa. QAutolibrary-kirjastoon lisättyä mittausfunktiota voitaisiin laajentaa keräämään monipuolisempaa dataa verkkosivulta. Grafanassa yksittäisessä näkymässä mahtuu esittämään paljon erilaisia kaavioita, joita voitaisiin hyödyntää verkkosivujen ongelmakohtien määrittämisessä. Esimerkiksi yksittäisten vaiheiden kestoa eri käyttäjämäärillä voisi olla mielenkiintoista analysoida raporttien lisäksi myös graafisesti. Toiminnallisuuden yleinen käyttäjäkokemus on myös yksi ylläpidettävistä kokonaisuuksista koko tuotteen tulevaisuutta ajatellen.

Yhteenvetona työkaluun onnistuttiin lisäämään ominaisuus, jonka hyödyt ovat selkeästi demonstroitavissa. Työkalua voi hyödyntää uudella tavalla verkkosivujen testaamisessa ja ongelmakohtien havaitsemisessa. Suorituskykytestaukselle löytyy varmasti tulevaisuudessa kasvavissa määrin kysyntää, kun pienemmätkin toimijat laajentavat palvelujaan enemmän internetin puolelle.

8 LÄHTEET

QAutomate Oy. Etusivu. Luettu 24.2.2020. https://qautomate.fi/

Q-Factory Oy. Etusivu. Luettu 24.2.2020. https://q-factory.fi/

Robot Framework. Etusivu. Luettu 25.2.2020. https://robotframework.org/

Sogeti Finland Oy. n.d. Suoritus- ja kuormitustestaus. Luettu 27.2.2020. https://www.sogeti.fi/palvelumme/testauspalvelumme/suorituskyky--ja-kuormitustestaus/

Techila Technologies Oy. Etusivu. Luettu 24.2.2020. http://www.techilatechnologies.com/

Vaidya, N. 2019. Performance testing tutorial – What is it & it's types? Luettu 28.2.2020. https://www.edureka.co/blog/performance-testing-tutorial/

Vuori, M. 2010. Suorituskykytestaus / kuormitustestaus. 6–8. Luettu 27.2.2020. https://www.mattivuori.net/julkaisuluettelo/liitteet/suorituskykytestaus.pdf