

Ilari Kemppainen

**VIRRAMITTAUSTULOSTEN TRENDIKUVAN LISÄÄMINEN
TESTITULOSPORTAALIIN**

**VIRRAMITTAUSTULOSTEN TRENDIKUVAN LISÄÄMINEN
TESTITULOSPORTAALIIN**

Ilari Kemppainen
Opinnäytetyö
Syksy 2021
Tietotekniikan tutkinto-ohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu

Tietotekniikan tutkinto-ohjelma, ohjelmistokehityksen suuntautumisvaihtoehto

Tekijä: Ilari Kemppainen

Opinnäytetyön nimi: Virranmittaustulosten trendikuvan lisääminen testitulosportaaliin

Työn ohjaajat: Jukka Jauhiainen, Juha Savolainen

Työn valmistumislukukausi ja -vuosi: Syksy 2021

Sivumäärä: 17

Opinnäytetyön tavoitteena oli lisätä valmiiseen testitulosportaaliin näkymä, jossa näkyy Polarin urheilukellojen virranmittaustulosten trendikuva halutuilla parametreilla. Uuden näkymän oli tarkoitus olla monikäyttöinen ja siitä haluttiin alusta asti valmis pohja, joka voidaan myöhemmin laajentaa myös muiden testitulosten visualisointiin.

Toteutus tehtiin lisäämällä uusi näkymä valmiiseen testitulosten seuranta varten kehitettyyn Portaaliin. Valmiiseen testiautomaatioympäristöön tutustuminen ja valmiiden komponenttien hyödyntäminen oli keskeisessä roolissa koko projektin ajan. Ensimmäisenä uuteen näkymään lisättiin valintanäkymä, jolla voidaan hakea ja rajata halutut testitulokset testattavan laitteen ja testin mukaan. Seuraava vaihe oli hakea tietokannasta testiajot halutuilla parametreilla ja niiden data. Viimeisenä vaiheena oli muodostaa testiajojen datasta graafi, josta voidaan helposti havaita virrankulutuksen muutokset. Näitä vaiheita toistettiin useita kertoja, kun huomattiin, että jonkin vaiheen toteutusta voisi parantaa tai tehdä paremmalla tavalla.

Lopputuloksena saatiin helposti muihinkin testeihin laajennettava sovellus, josta voidaan valita halutut testiajot ja niiden virrankulutus visualisoidaan selkeästi. Graafista näkyy selkeästi X-akselilla aika ja Y-akselilla virrankulutus. Lisäksi testattavan laitteen ohjelmistoversio on luettavissa osoittamalla kursorilla kuvaajan datapistettä.

Trendinäkymää on tarkoitus jatkokehittää tulevaisuudessa visualisoimaan myös muita mittaustuloksia. Lisäksi muun muassa ulkonäköä ja käytettävyyttä on tarkoitus parantaa tulevaisuudessa.

Asiasanat: JavaScript, ReactJS, käyttöliittymät, ApexChart, testiautomaatio, ohjelmistokehitys, graafit

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Information Technology, Option of Software Development

Author: Ilari Kemppainen
Title of thesis: Trend Graph Extension to Test Results Portal
Supervisors: Jukka Jauhiainen, Juha Savolainen
Term and year when the thesis was submitted: Autumn 2021
Number of pages: 17

The goal of the thesis was to create a new page of current consumption trend charts to existing test automation results portal. Purpose of the feature is to help tracking changes in power consumption of Polar sport watches between different firmware versions.

Feature was implemented into existing result portal web page as a new page. The result is working multifunctional custom trend view of test results that can be used to track changes in current consumption of selected sport watch. In the future trend graph will be further developed to visualize several measurement test results in addition to current consumption.

Keywords: JavaScript, React, Software testing, Test automation

SISÄLLYS

TIIVISTELMÄ.....	3
ABSTRACT.....	4
SISÄLLYS.....	5
SANASTO.....	6
1 JOHDANTO.....	7
2 TESTIAUTOMAATIO.....	8
2.1 Jenkins.....	8
2.2 Result Portal.....	9
3 TRENDIGRAAFIN KEHITTÄMINEN.....	10
3.1 Tavoitteet.....	10
3.2 Toteutus.....	10
3.2.1 Testiajojen valitseminen.....	11
3.2.2 Trendigraafien lisääminen.....	12
3.3 Lopputulos.....	12
4 JATKOKEHITTÄMINEN.....	15
5 YHTEENVETO.....	16
LÄHTEET.....	17

SANASTO

ApexCharts.js	JavaScript-kirjasto erilaisten visualisointien toteuttamiseen verkkosivuilla
API	Application Programming Interface. Ohjelmointirajapinta, joka mahdollistaa sovellusten välisen kommunikoinnin
Back-end	Palvelimella ajettava koodi, joka ei suoraan näy käyttäjälle, vastaa web-sovelluksen taustalla vaadittavasta tietoliikenteestä ja datan käsittelystä
Front-end	Verkkoselaimella ajettava koodi, joka näkyy käyttäjälle, esimerkiksi verkkosivun ulkoasu ja toiminnallisuudet verkkosivulla
JavaScript	Web-kehityksessä yleinen ohjelmointikieli, joka käytetään yleensä Front-end -kehityksessä
Käyttöliittymä	User Interface (UI). Ohjelmiston osa, jonka kautta käyttäjä käyttää tuotetta
Query	Tietokantakysely, joka palauttaa halutun datan tietokannasta
ReactJS	JavaScript-kirjasto käyttöliittymien rakentamiseen
Testiautomaatio	Sovellus, joka testaa toisen sovelluksen tai ohjelmiston toimivuutta
Tietokanta	Tietokoneen tallentama järjestelty kooste dataa

1 JOHDANTO

Opinnäytetyön tilaajana toimii Polar Electro Oy. Polar on perustettu Kempeleessä vuonna 1977 ja se on ollut alusta asti edelläkävijä langattomien sykemittarien kehittämisessä. Vuonna 1979 Polar sai ensimmäisen patentin langattomalle sykkeenmittaukselle ja lanseerasi maailman ensimmäisen langattoman sykemittarin vuonna 1982. (1.) Nykyään Polarin palveluksessa on 1200 ihmistä ja Polar kehittää urheilukellojen sekä sykemittareiden lisäksi erilaisia sovelluksia ja palveluita tukemaan urheilua (2).

Koska Polar kehittää jatkuvasti uusia laitteita, on laadunvarmistus olennaista. Kaikki uudet ohjelmistoversiot testataan laadun varmistamiseksi. Testauksen toistuvuuden vuoksi automaatiotestaus on todella tärkeässä roolissa. Muun muassa laitteiden eri ohjelmistoversioiden virrankulutus mitataan säännöllisesti.

Tämän opinnäytetyön tarkoituksena oli laajentaa nykyiseen testiautomaatioympäristöön uusi graafinäkymä helpottamaan virranmittaustulosten seurantaa. Tavoitteena oli, että virrankulutustestien tulos eri laitteilla tuodaan omalle helposti seurattavalle graafille, josta näkyy virranmittaustestin keskimääräinen virrankulutus, testin päivämäärä ja testattava ohjelmisto urheilukellossa. Tämä helpottaa huomattavasti virrankulutuksen tarkastelua eri ohjelmistoversioiden välillä.

2 TESTIAUTOMAATIO

Kaupallisen tuotteen menestymisen kannalta laadun varmistaminen on erityisen tärkeää. Huono käyttökokemus antaa ohjelmistosta huonon ensivaikutelman ja vaikuttaa suoraan ostopäätökseen. Ohjelmistotestaus on laaja kokonaisuus ja se sisältää laajasti eri osa-alueita, joista tulee huolehtia. Yleistettynä voidaan kuitenkin sanoa, että laadunvarmistuksessa ja testauksessa on tarkoitus varmistaa, että ohjelmisto toimii, kuten sen halutaan toimivan. Testaus ei varsinaisesti suoraan tuota mitään ja se saattaa jopa hidastaa kehitystä. Laadunvarmistus näkyy asiakkaalle vain, kun sitä ei ole ja tuotteen laatu on huono. Testauksen hyödyt näkyvätkin tuotekehityksessä vasta sen jälkeen, kun tuote on julkaistu. Mitä aiemmin mahdollisen virheet löydetään ja korjataan, sitä paremman tuotteen voi viedä tuotantoon. (3.)

Laadun varmistamisessa automaatiotestaus on erityisen tärkeässä asemassa helpon skaalautuvuuden vuoksi. Uusia testinodeja (-solmuja) on helppo lisätä ja automaatiotestejä voidaan tehdä juuri tietylle tuotteelle. Lisäksi jatkuvassa testauksessa voidaan varmistaa se, etteivät uudet ominaisuudet riko mitään toiminnallisuuksia ohjelmistoissa sekä virheet ja puutteet ohjelmistossa havaitaan mahdollisimman aikaisin, kun automaatiotestit suoritetaan laitteiden ohjelmistoille jo kehitysvaiheessa.

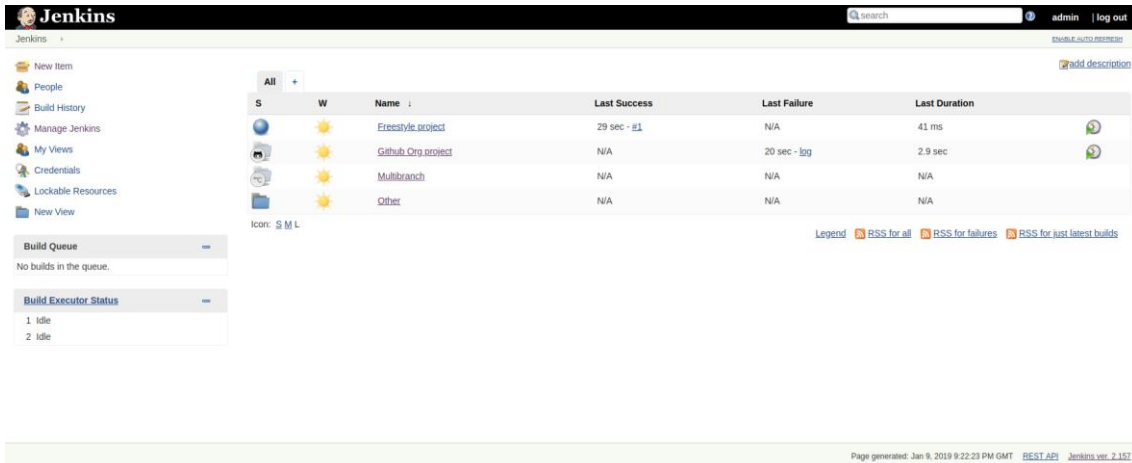
Testiautomaatio tarkoittaa sitä, että ohjelmistoa ja tuotetta ei testaa ihminen vaan sovellus. Testiautomaation olennaisimpia etuja manuaalitestaukseen ovat se, että kone on aina saatavilla ja sillä voidaan suorittaa testejä jokaisena päivänä ympäri vuorokauden. Tästä johtuen se on lähes aina halvempaa, kun jotain asiaa pitää testata useasti. Testausautomaation hyötyjä ovat testauksen nopeus, kattavuus, toistettavuus ja se, kuinka nopeasti uusi ohjelmistoversio saadaan testaukseen. Testiautomaatio voi aloittaa ohjelmiston testaamisen käytännössä heti, kun uusi versio on valmis. Tämän lisäksi kustannusten ja työtuntien säästäminen parantaa työn tehokkuutta. (4.)

2.1 Jenkins

Jenkins on avoimeen lähdekoodiin perustuva käännösautomaation palvelinohjelmisto, mikä mahdollistaa automaatiotestien automatisointia halutuilla tietokoneilla ja laitteilla. Testejä voidaan

asettaa ajettavaksi esimerkiksi haluttuun aikaan tai aina kun uusi ohjelmistoversio julkaistaan, mikä nopeuttaa testaamista ja virheiden löytymistä. Jenkins on yksi yleisimmistä työkaluista testiautomaation parissa sen laajennettavuuden ja erilaisten työkalujen laajan valikoiman vuoksi. (5.)

Jenkins mahdollistaa sen, että testejä voidaan ajaa yhtä aikaa todella monella ja erilaisilla testikoneilla, joissa on erilaisia urheilukelloja. Lisäksi nodelle voidaan lisätä älypuhelin ja suorittaa testejä kellon ja puhelimen välisistä toiminnoista. Nämä mahdollistavat sen, että testejä voidaan ajaa jatkuvasti ja todella laajalla variaatiolla erilaisten tietokoneiden ja puhelimien yhteensopivuuden testaamiseksi. Jenkins-käyttöliittymä (kuva 1) on helposti hallittava ja sen avulla voidaan tarkastella muun muassa luotuja testejä, testiajojen tuloksia, nodeja sekä niiden statusa.



The screenshot shows the Jenkins web interface. At the top, there is a search bar and user information (admin | log out). Below the navigation menu, there is a table of build jobs. The table has columns for Status (S), Web icon (W), Name, Last Success, Last Failure, and Last Duration. The jobs listed are 'Ereestiyte.rojject', 'GitHub_Corp.rojject', 'Multibranch', and 'Other'. The 'Ereestiyte.rojject' job shows a last success of '29 sec - #1' and a last duration of '41 ms'. The 'GitHub_Corp.rojject' job shows a last failure of '20 sec - log' and a last duration of '2.9 sec'. The 'Multibranch' and 'Other' jobs show 'N/A' for both last success and last failure, and 'N/A' for last duration. Below the table, there is a legend for RSS feeds and a footer indicating the page was generated on Jan 9, 2019 9:22:23 PM GMT.

S	W	Name	Last Success	Last Failure	Last Duration
		Ereestiyte.rojject	29 sec - #1	N/A	41 ms
		GitHub_Corp.rojject	N/A	20 sec - log	2.9 sec
		Multibranch	N/A	N/A	N/A
		Other	N/A	N/A	N/A

KUVA 1. Jenkins-käyttöliittymä (6)

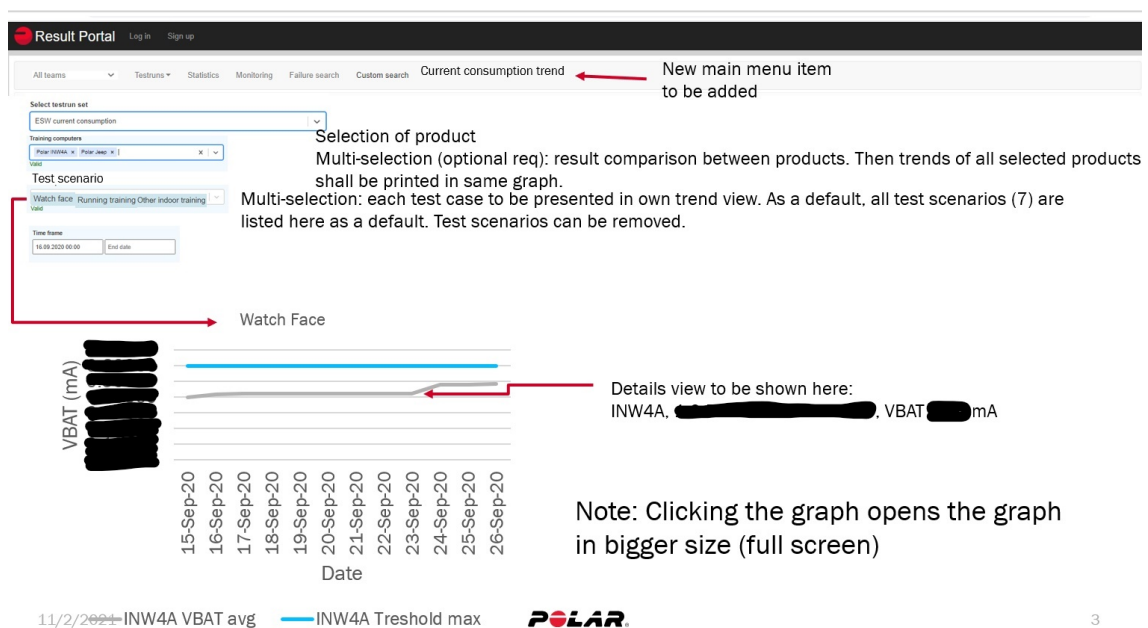
2.2 Result Portal

Jenkinsin lisäksi Polarilla on tehty sisäiseen käyttöön erillinen testiautomaatioympäristö Result Portal, joka helpottaa testitulosten seuranta ja analysointia. Jenkins-testien data lähetetään Portalin päähän, jossa dataa käsitellään, tallennetaan tietokantaan ja sitä voi tarkastella sekä muokata. Portalista on Jenkinsiä helpompi seurata tietyn testin historiaa ja tulosten läpikäyntiä. Lisäksi Portaliin voi helpommin kirjata, miksi testi ei ole onnistunut sekä linkata Jira linkin virheen aiheuttajaksi.

3 TRENDIGRAAFIN KEHITTÄMINEN

3.1 Tavoitteet

Polarilla on tehty testejä mittaamaan eri laitteiden virrankulutusta. Testissä virrankulutukselle on määritetty tietty raja-arvo, jonka ylittäessä testi ei mene läpi liian suuren virrankulutuksen vuoksi. Ongelmana on kuitenkin, se että virrankulutuksen muutoksia on todella hidasta ja työlästä seurata. Tähän ongelmaan haluttiin ratkaisuksi selkeä näkymä, jossa virrankulutus näkyy trendigraafina aikajaksolla. Näkymään haluttiin helppokäyttöinen ja selkeä valinta halutuista testeistä, jotta haluttu näkymä on helppo ja nopea valita (kuva 2). Näin tätä näkymää voi käyttää mahdollisimman pienellä kynnyksellä ja tuloksia tulee seurattua enemmän.



KUVA 2. Hahmotelma halutusta näkymästä

3.2 Toteutus

Toteutus tehtiin valmiin Result Portalin front-end -puolelle, johon back-endiltä saatiin haluttua dataa rajapintaa hyödyntämällä. Portaliin lisättiin uusi sivu trendinäköille. Valmiit rajapinnat ja kyselyt mahdollistivat hyvin pitkälti kaiken halutun datan saaminen. Jossain pienissä tapauksissa back-endiin piti tehdä pieniä muutoksia.

Koska Result Portal on hyvin pitkälti tehty Reactia käyttämällä, myös trendikuvien lisääminen tehtiin ReactJS:llä (6). Itse graafien tuottamiseen valittiin ApexCharts (7), koska sitä on jo Portal ympäristössä käytetty jonkin verran. Näin valmiiksi käytössä olevia kirjastoja ja eri komponentteja saadaan luotua yhdenmukainen kokonaisuus koko testiautomaatioympäristölle eikä uusien toimintojen käyttäminen poikkea merkittäväksi käyttäjälle tutuista vanhemmista toiminnoista.

3.2.1 Testiajojen valitseminen

Testiajojen valintaan haluttiin mahdollisimman yksinkertainen näkymä, josta saadaan graafeja näkyviin mahdollisimman vähällä määrällä klikkauksia. Tässä käytettiin hyödyksi Result Portalissa jo aiemmin käytettyjä komponentteja testiajojen valintaan, joka nopeutti toteutuksen tekemistä ja samalla visuaalinen ulkoasu on yhdenmukainen muihin Portalin sivuihin (kuva 3).

The screenshot displays a user interface for selecting test components. It features three main sections:

- Test Component:** A dropdown menu currently showing "Current Consumption".
- Training Computer:** A dropdown menu currently showing "Polar INW4A".
- Test Scenarios:** A list of three selected scenarios, each with a close button (x):
 - test_001_trainingComputer_currentConsumption_inWatchFace
 - test_003_trainingComputer_currentConsumption_inOtherIndoorT...
 - test_004_trainingComputer_currentConsumption_inRunningTraini...

Below the scenarios is a **Time frame** section with two input fields: "27.10.2021 00:00" and "End date". At the bottom left is a "Search" button.

KUVA 3. Testiajojen valintakomponentti

Virrankulutuksen muutokset haluttiin näkyviin niin, että mahdollisimman vähillä klikkauksilla saa näkyviin yleisimmät ja laitteen akun keston kannalta tärkeimmät testiskenaariot. Valintakomponenttia muutettiin siten, että siihen on valmiiksi asetettu seitsemän yleisintä testiskenaariota. Testitulosten analysoija voi halutessaan valita lisää testiskenaarioita näkymään, joiden virrankulutuksen trendi on kiinnostava.

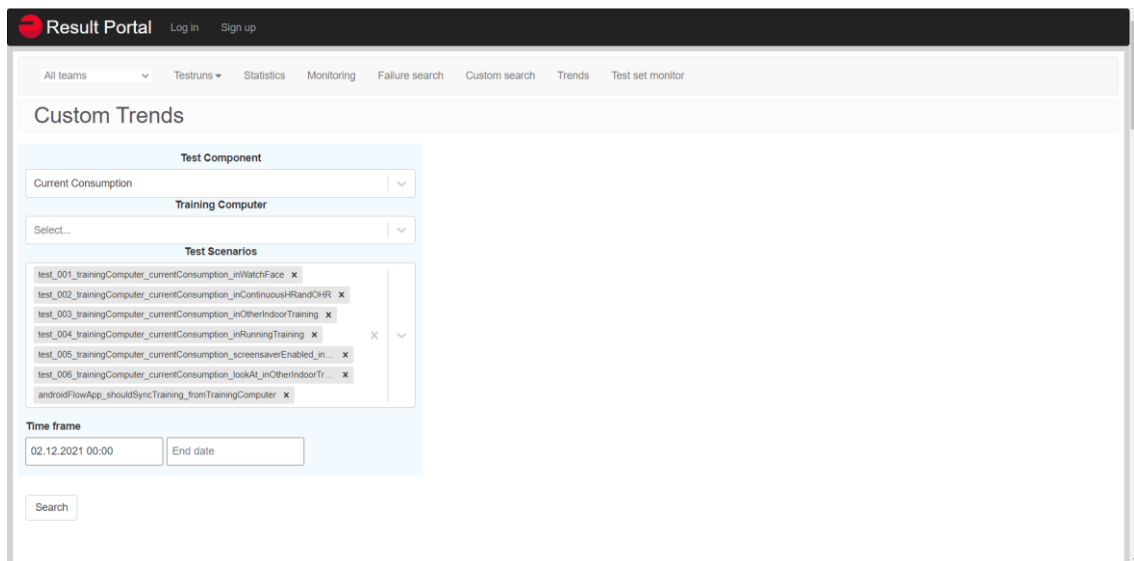
3.2.2 Trendigraafien lisääminen

Kun halutut parametrit on valittu valintanäkymästä, "Search" -painiketta painamalla sivulle tulostetaan oma graafi jokaiselle valitulle testiskenaariolle. X-akselilla näkyy aikajana ja Y-akselilla virrankulutuksen keskiarvo eri mittauskerroilla. Lisäksi graafin Z-akselille on lisätty testattu laiteohjelmistoversio. Projektin kuluessa tässä havaittiin ongelmia liittyen siihen, että päivämäärän formaatti ei ollut yhteensopiva ApexChartsin kanssa. Tästä johtuen päivämäärä näkyi jokaisen testin alla X-akselilla, minkä vuoksi se ei ollut luettavissa, kun testituloksia tarkasteltiin pitkältä aikaväliltä. Tämä korjattiin väliaikaisesti piilottamalla X-akselin päivämäärä kuvaajasta, koska päivämäärä on kuitenkin luettavissa osoittamalla kursorilla kuvaajaa. Myöhemmin tähän tehtiin parannus siten, että päivämäärän formaatti muokataan yhteensopivaksi ja päivämäärä on helposti luettavissa kuvan alapuolelta. Projektin edetessä myös virrankulutuksen trendi muutettiin viivagraafista tolppagraafiksi, jolloin ennen kaikkea pitkät aikavälit testitulosten välillä on selkeämmin visualisoitu.

3.3 Lopputulos

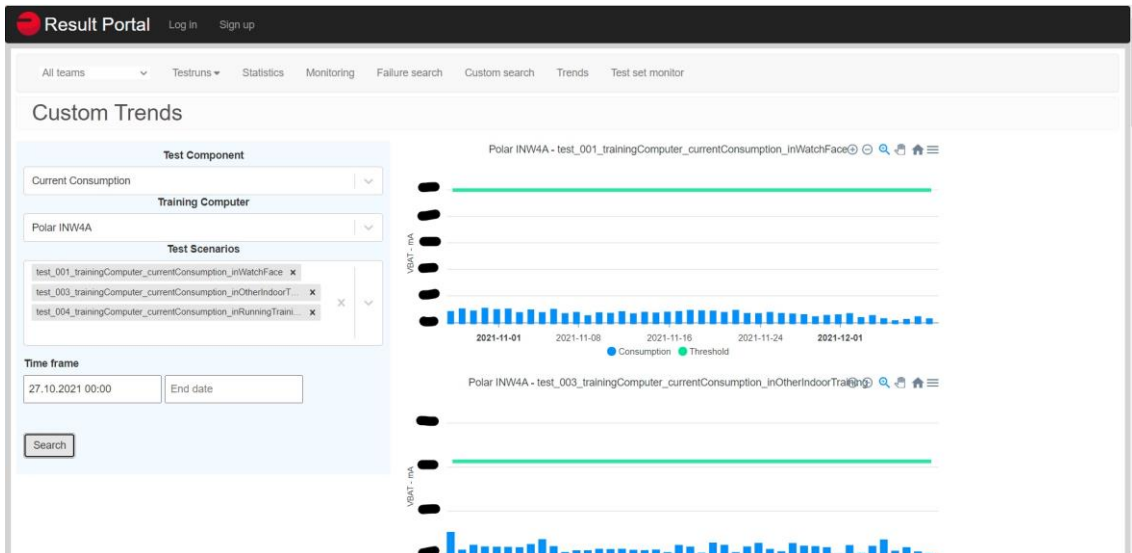
Lopputuloksena saatiin selkeä, helposti käytettävä ja laajennettava sovellus, jolla automaatio-testien data visualisoidaan selkeäksi graafiksi tulosten tarkastelua varten eri ohjelmistoversioiden välillä. Graafien valinta haluttiin mahdollisimman nopeaksi siten, että tarvittavien klikkausten määrä olisi mahdollisimman vähäinen trendikuvien luomiseen. Näkymästä haluttiin kuitenkin mahdollisimman yleiskäyttöinen, joka voidaan laajentaa muidenkin testitulosten tarkasteluun sekä joustava käyttäjän tarpeiden mukaan riippuen siitä, mitä testejä halutaan tarkastella. Nämä seikat otettiin huomioon käyttöliittymän suunnittelussa alusta asti ja lopputuloksena syntyikin suoraviivainen ja yksinkertainen käyttöliittymä, jonka näkymä on kuitenkin helposti käyttäjän muokattavissa ja joka on helposti laajennettavissa.

Aloituspäätöksessä on oletuksena valittu komponentiksi virrankulutus, joka on myös toistaiseksi ainoa komponentti, jonka tuloksia voi trendikuvaan valita. Tätä on tarkoitus kuitenkin laajentaa jatkossa myös muille testituloksille, joten sivu tehtiin mahdollisimman helposti laajennettavaksi ja eri testikomponentin valinta on tehty valmiiksi. Tämän lisäksi myös seitsemän yleisintä testiä on valittu valmiiksi ja aikaväliksi on asetettu yksi viikko. Tämä vähentää tarvittavien klikkausten määrää ja graafit saadaan näkyviin nopeasti valitsemalla haluttu rannelaite ja painamalla "Search" -painiketta. Kaikkia parametreja voidaan kuitenkin vaihtaa (kuva 4).



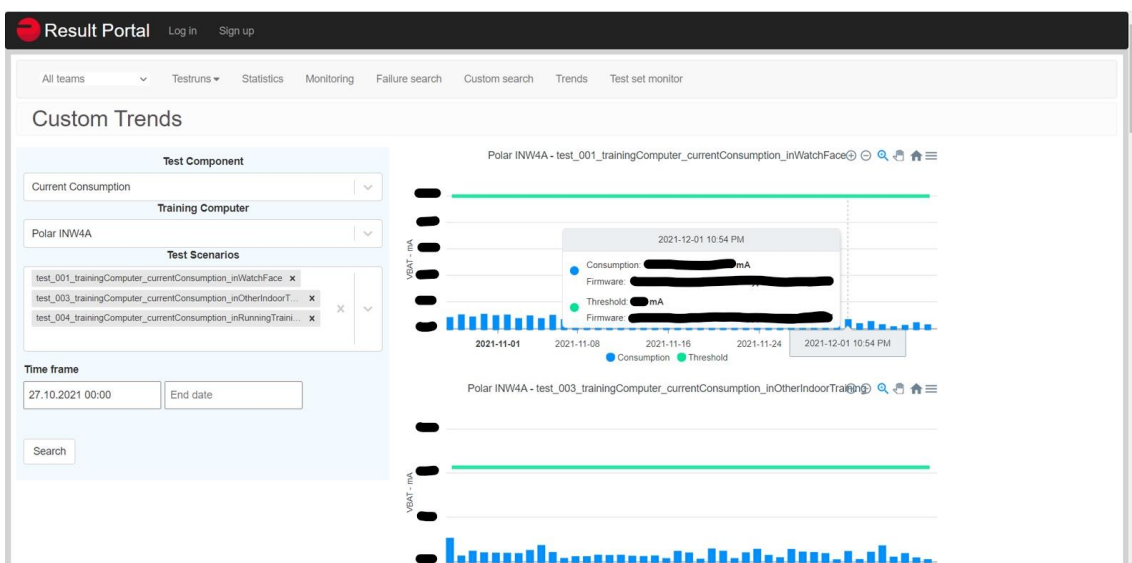
KUVA 4. Sivun aloitusnäky

Kun testit on valittu halutuilla parametreilla ja painetaan "Search" -painiketta, sivulle luodaan trendigraafi jokaiselle valitulle testiskenaariolle (kuva 5). Graafin sivuilla näkyy X-akselilla päivämäärät selkeästi näkyvässä muodossa. Kaikkia päivämääriä ei graafin alapuolelle mahdu, mikäli tuloksia tarkastellaan pidemmältä aikaväliltä, mutta viisi päivämäärää riittää hyvin havainnollistamaan miltä aikaväliltä tuloksia tarkastellaan. Y-akselilla näkyy virrankulutuksen määrä ja yksikkö. Testeille asetettu raja-arvo, jonka yli virrankulutus ei saa mennä, näkyy vihreänä viivana graafissa ja virrankulutus testin tulos näkyy sinisinä tolppina.



KUVA 5. Valmis näkymä

Kun kursorin vie graafin päälle, nähdään yksittäisen testin tulokset tarkemmin (kuva 6). Ikkunassa näkyy testin päivämäärä, tarkka virrankulutus, virrankulutuksen raja-arvo sekä testattu ohjelmistoversio. Laiteohjelmistoversio näkyy ApexChartsin toimintaan liittyvistä syistä kahteen kertaan. Virrankulutus ja virrankulutuksen raja-arvo ovat kummatkin trendikäyriä, joilla on data X-, Y- ja Z-akselille. Tästä johtuen Z-akselilla on data sekä virrankulutuksen, että virrankulutuksen raja-arvon data. Tämä voidaan myöhemmin muokata siten, että toisen Z-akselille lisätään jotain muuta dataa. Tämä ei kuitenkaan vaikuta sovelluksen toimintaan, vaikka laiteohjelmisto näkyy kahteen kertaan, eikä sen muokkaamista nähty kriittisenä osana sovelluksen toimintaa.



KUVA 6. Kursori trendikuvan päällä

4 JATKOKEHITTÄMINEN

Tulevaisuudessa trendikuvia on tarkoitus laajentaa myös muiden testausalueiden tulosten visualisointiin. Tämän lisäksi käyttöliittymää ja käytettävyyttä on tarkoitus parantaa saadun palautteen perusteella, jotta se palvelee mahdollisimman hyvin Result Portalin käyttäjiä.

Jatkokehittäminen ja työn priorisointi toteutetaan palautteen ja kokemusten perusteella. Usein kehittämisvaiheessa on vaikea ennakoida minkä loppukäyttäjä kokee tärkeimmäksi asiaksi sovelluksessa. Testien valintaan tullaan lisäämään toiminto, jolla voidaan filteröidä vain halutun ohjelmistohaaran versiot graafille.

5 YHTEENVETO

Opinnäytetyön tavoitteena oli kehittää uusi näkymä nykyiseen testiautomaatioympäristöön helpottamaan virranmittaustulosten seuranta. Lähtökohtana oli helpottaa testien tulosten seuraamista ja vähentää tulosten analysointiin tarvittavaa aikaa. Projektiin kuului suunnitella tarpeeseen sopivia ratkaisuja, jotta ne auttavat Result Portalin käyttäjiä. Olennaisena osana projektia oli tutustua valmiiseen testiautomaatioympäristöön ja opetella käyttämään sekä soveltamaan siinä valmiina olevia komponentteja.

Työn lopputulos vastaa sen tavoitteita ja vaatimuksia. Uusi näkymä luotiin oikeaan tarpeeseen ja se onkin ollut käytössä testitulosten seuraamisessa. Opinnäytetyö tarjosi sopivasti haasteita ja oli todella opettavainen kokemus. Projektin aikana opin paljon web-kehityksestä ReactJS:ää ja Apex-Chartsia hyödyntämällä sekä käsittelemään testiautomaation tuottamaa dataa. Samalla työn aikana opin paljon testiautomaation kokonaisuudesta, joka ei minulle ennen projektia ollut kovinkaan tuttu.

LÄHTEET

1. Polarin ensimmäinen sykemittari Heurekassa. 2016. Uusiteknologia.fi. Saatavissa: <https://www.uusiteknologia.fi/2016/02/09/polarin-ensimmainen-sykemittari-heurekassa/>. Hakupäivä 21.11.2021.
2. Keitä olemme. Polar Electro Oy. Saatavissa: https://www.polar.com/fi/tietoa_polarista/keita_olemme. Hakupäivä 7.11.2021.
3. Laadunvarmistus ja ohjelmistotestaus. Ite wiki. Saatavissa: <https://www.itewiki.fi/opas/laadunvarmistus-ja-ohjelmistotestaus/>. Hakupäivä 11.11.2021.
4. Mitä on testausautomaatio? 2021. Qentinel. Saatavissa: <https://value.qentinel.com/fi/mita-on-testausautomaatio>. Hakupäivä 7.11.2021.
5. How to Put Quality in the Build With Jenkins Test Automation. 2021. Testim. Saatavissa: <https://www.testim.io/blog/jenkins-test-automation/>. Hakupäivä 10.12.2021.
6. Yanniks 2019. Startseite von Jenkins (CyanogenMod Jenkins Instanz). Wikipedia. Saatavissa [https://fi.wikipedia.org/wiki/Jenkins_\(ohjelmisto\)#/media/Tiedosto:Jenkins_Home.png](https://fi.wikipedia.org/wiki/Jenkins_(ohjelmisto)#/media/Tiedosto:Jenkins_Home.png). Hakupäivä 11.11.2021.
7. React 2021. Saatavissa <https://reactjs.org/>. Hakupäivä 9.12.2021.
8. ApexCharts.js. 2021 Saatavissa <https://apexcharts.com/>. Hakupäivä 9.12.2021