

Ryhmä: TeamH

Tekijä: Henri Haverinen, henri.s.haverinen@student.jyu.fi

Julkisuustaso: Julkinen

Päivämäärä: 28.9.2017

Versio: 1.1.0

Opintojakso: TIES546 Ohjelmistotestaus, Jyväskylän yliopisto, informaatioteknologian tiedekunta

Versiohallinta: <https://github.com/hhaverinen/TIES546>

DatabaseClient

Testausprojektin suunnitelma

Järjestelmätestaus

Toiminnallisuustestaus

Sisällysluettelo

1 Ohjelmiston kuvaus ja kohderyhmät.....	2
2 Testattavat ominaisuudet.....	2
3 Testausprojektin tavoitteet.....	3
4 Suoritettavat testaustasot ja -tyypit.....	3
5 Testausprojektin jäsenen tietotaidot.....	4
6 Testausprojektin aikataulu.....	4

1 Ohjelmiston kuvaus ja kohderyhmät

Testattava ohjelmisto on Henri Haverisen kehittämä prototyyppi nimeltään “DatabaseClient”. Ohjelmisto on kehitetty harjoitustyönä Jyväskylän yliopiston informaatioteknologian tiedekunnan tarjoamalle kurssille *TIEA306 Ohjelmointityö*. Ohjelmisto on työasemasovellus.

Ohjelmisto on kehitetty tarjoamaan käyttäjälle graafisen käyttöliittymän tietokantayhteyksien muodostamiseen ja SQL-kyselyjen tekemiseen. Ohjelmisto on pääasiassa tarkoitettu henkilöille, jotka työskentelevät tietokantojen kanssa päivittäin, kuten ohjelmistokehittäjät ja IT-tuen henkilöt. Ohjelmisto ei kuitenkaan tuo suurta lisäarvoa esimerkiksi tietokantojen ylläpitäjille, sillä ohjelmisto ei tarjoa käyttöliittymää tietokantojen ylläpidollisille työkaluille. Lisäksi ohjelmisto on toistaiseksi suunnattu vain MySQL-tietokantojen käyttäjille, sillä se on tällä hetkellä ainut tuettu tietokanta.

Ohjelmiston olennaisimpia ominaisuuksia ovat tietokantakyselyjen tekeminen, tietokantayhteyksien tietojen tallentaminen ja lukeminen tiedostosta. SQL-lauseiden tallentaminen ja lukeminen tiedostosta sekä SQL-kyselyn tulosten vienti tiedostoon. Lisäksi käyttöliittymä pitää kirjaa käyttäjän syöttämistä SQL-kyselyistä session aikana. Käyttäjä voi valita ja suorittaa aiemmin tekemänsä kyselyn uudestaan alasvetovalikosta. SQL-kyselyiden syötekenttä suorittaa oletuksena SQL-kyselyn siltä riviltä, jolla kursori on. On myös mahdollista maalata suoritettava SQL-kysely.

Tietokantayhteyden muodustuttua käyttöliittymä näyttää myös listauksen tietokannassa olevista tauluista.

2 Testattavat ominaisuudet

Testattavat ominaisuudet ja perustelut testauksen tarpeelle ovat seuraavat:

- Tietokantayhteyksien muodostaminen on edellytys ohjelmiston käyttämiselle, ja siksi sen toiminnallisuutta on testattava.
- SQL-kyselyjen tekeminen tietokantaan on ohjelmiston pääasiallinen käyttötarkoitus ja siten sen toiminnallisuus tulee testata.

- Tietokantayhteyksien tietojen tallentaminen ja lukeminen tiedostosta on olennainen ominaisuus, jonka vuoksi sen toiminnallisuuden testaaminen on tärkeää.
- SQL-kyselyn tulosten vienti tiedostoon on usein haluttu ja tarpeellinen ominaisuus, jonka vuoksi ominaisuutta tulee testata.
- Käytöliittymä tuottaa myös käyttäjälle näkyvää lokia, jossa kerrotaan tarkemmin onnistumis- ja virheviesteistä. Lokin toimintaa ja viestien selkeyttä on tärkeä testata, koska se kertoo käyttäjälle ohjelmiston tilasta sekä auttaa selvittämään mahdollisia ongelmatilanteita.

Testauksesta pois rajatut ominaisuudet ja perustelut rajauksille ovat seuraavat:

- SQL-kyselyiden lukeminen ja tallentaminen tiedostoon ei ole kriittinen ja usein tarpeellinenkaan ominaisuus, joten se on rajattu pois testattavista ominaisuuksista.
- Käyttäjän syöttämien SQL-kyselyiden ylläpitäminen sekä niiden valitseminen alusvetolaatikosta ei ole myöskään käyttöä estävä ominaisuus, joten sen toiminnallisuutta ei tässä testausprojektissa testata.
- Tietokantataulujen näyttäminen visuaalisesti tietokantayhteyden muodostumisen jälkeen ei ole tärkeä ominaisuus. Lisäksi kyseinen tieto on käyttäjän mahdollista hakea myös SQL-kyselyllä. Näistä syistä kyseistä ominaisuutta ei testata.
- Syötekentän toiminnallisuuksia (maalatun SQL-kyselyn suorittaminen, kursorin rivillä olevan SQL-kyselyn suorittaminen) ei testata, koska niiden uupuminen ei estä ohjelmiston käyttöä.

Edellä olevien listauksen valinnoissa on vaikuttanut myös testausprojektin työmäärän vähentäminen.

3 Testausprojektin tavoitteet

Vaikka ohjelmisto on vain prototyyppi, on sitä tärkeää testata keskeisten ja käyttöä olennaisesti haittaavien puutteiden ja virheiden löytämiseksi. Varhaisessa vaiheessa löydettyjen ongelmien korjaaminen on edullisempaa, ja mikäli testauksessa havaittaisiin jokin ominaisuus tai toteutus käyttökelvottomasti, ei sitä turhaan alettaisi implementoimaan oikeaan tuotteeseen. Lisäksi jos prototyyppiä demottaisiin asiakkaalle, on ohjelmiston perustoiminnallisuuksien oltava kunnossa. Tällä testausprojektilla halutaan nimenomaan kiinnittää huomiota kriittisten ominaisuuksien varmaan toiminnallisuuteen ja mahdollisesti löytää parannus/korjaus ideoita.

4 Suoritettavat testaustasot ja -tyypit

Testausprojektin tavoitteita tukien ohjelmistolle suoritetaan järjestelmätason toiminnallisuustestaus. Testattavat ja pois rajatut ominaisuudet on esitelty luvussa 2.

5 Testausprojektin jäsenen tietotaidot

Allekirjoittanut on itse kehittänyt testattavan ohjelmiston, joten sen toiminnallisuudet ja toteutus ovat tuttuja. Lisäksi olen työskennellyt noin 2.5 vuotta ohjelmistotalossa, jossa olen saanut hyvän yleiskäsityksen ohjelmistojen kehittämisestä sekä vakaan ohjelmointiosaamisen. Testauksesta minulla ei ole aiempaa kokemusta, muuta kuin manuaalisesti ominaisuuksien testaaminen, eli “kyllähän se näyttäisi toimivan” -tyylinen testaus.

6 Testausprojektin aikataulu

Testaussuunitelmien suunniteltu toimituspäivä on viimeistään 20.10., testaus- ja analyysiraporttien suunniteltu toimituspäivä on viimeistään 3.11. ja viimeisteltyjen tulosten toimituspäivä viimeistään 10.11.