

OptiLift

Tietotekniikan Sovellusprojekti

Lauri Laasala
Olli Lukkarinen
Ville Räisänen
Vesa Tanhua-Tyrkkö

Testaussuunnitelma

Versio 1.0

29.4.2004

Jyväskylän yliopisto
Tietotekniikan laitos

- Tekijät:** Laasala Lauri, lvlaasal@cc.jyu.fi, 044-3375831
Lukkarinen Olli, olliluk@cc.jyu.fi, 040-7668234
Räisänen Ville, vtraisan@cc.jyu.fi, 044-3060999
Tanhua-Tyrkkö Vesa, vttanhua@cc.jyu.fi, 050-3475670
- Projektin tiedot:** OptiLift-projekti
Projektitila: Agora C225.3, 014-260 4971
Kotisivu: <http://sovellusprojektit.it.jyu.fi/optilift/>
- Työn nimi:** OptiLift-projektin testaussuunnitelma.
- Työn kuvaus:** Testaussuunnitelma tietotekniikan Sovellusprojektiin.
- Tilaaaja:** Kilpa- ja huippu-urheilun tutkimuskeskus KIHU.
- Teettäjä:** Jyväskylän yliopisto, tietotekniikan laitos.
Vastaava ohjaaja: Markus Inkeroinen.
Tekninen ohjaaja: Ville Tirronen.
- Tiivistelmä:** Tämä dokumentti on Jyväskylän yliopiston keväällä 2004 toteutettavan OptiLift-Sovellusprojektin testaussuunnitelma. Dokumentissa kuvataan projektin toteuttaman sovelluksen testaamiseen käytettäviä menetelmiä ja periaatteita sekä testitapauksia ja ympäristöä, jossa testaus suoritetaan.
- Avainsanat:** alfa- ja beta-testaus, KIHU, kilpa- ja huippu-urheilun tutkimuskeskus, testausympäristö, testitapaus.

Versiohistoria

Versio	Päiväys	Tehnyt	Muutokset
0.1	7.4.2004	Ville Räisänen	Ensimmäinen versio, tehty runko dokumentille.
0.2	14.4.2004	Ville Räisänen	Lisätty testitapauksia, testauksen periaatteita, kuvauksia testilaitteistoista ja -ohjelmistoista.
0.3	22.4.2004	Ville Räisänen	Lisätty testitapauksia, muokattu testauksen periaatteita.
0.4	26.4.2004	Ville Räisänen	Lisätty testitapauksia, tarkennettu niiden kuvauksia, tarkennettu testauksen periaatteita.
0.5	27.4.2004	Ville Räisänen	Lisätty testitapausten ja virheiden raportointikäytäntö, korjattu testitapauksia ja dokumentin ulkoasu.
1.0	28.4.2004	Ville Räisänen	Korjattu testitapausten kuvauksia, lisätty pari uutta, korjattu kirjoitusvirheet.
1.0	29.4.2004	Ville Räisänen	Tehty viimeiset korjaukset hyväksytyyn versioon.

Sisältö

1	Johdanto	1
2	Termistöä	2
3	Testausympäristö	3
3.1	Laitteistot	3
3.2	Ohjelmistot ja käyttöjärjestelmät	4
4	Testauksen periaatteet	5
4.1	Testauksen tavoite	5
4.2	Ohjelmakoodin testaus	5
4.2.1	Moduulitestaus	5
4.2.2	Integrointitestaus	6
4.3	Testitapausten toteuttaminen	6
4.4	Sovelluksen testikäyttö	6
4.5	Virheiden raportointikäytäntö	7
5	Testitapaukset	9
5.1	Sovelluksen toiminnallisuus	9
5.1.1	Nostajien lisääminen ja poistaminen	9
5.1.2	Kalibrointi	9
5.1.3	Suorituksen läpivienti ja vertailu	10
5.1.4	Virhetilanteet	11
5.2	Sovelluksen ikkunoiden ja valikoiden käsittely	13
5.3	Sovelluksen analyysiosa	14
6	Lähteet	16

1 Johdanto

OptiLift-projekti suunnittelee ja toteuttaa Kilpa- ja huippu-urheilun tutkimuskeskukselle painonnoston levytankoharjoitteluun nostotekniikan automaattisen mittaus- ja analysointisovelluksen. Järjestelmän avulla analysoidaan urheilijan levytankoharjoittelua, joka on olennainen osa monien eri urheilulajien voimaharjoittelua. Oikean tekniikan hallitseminen on tärkeää niin vammattoman kuin mahdollisimman tehokkaankin harjoittelun kannalta. Monilla valmentajilla ei kuitenkaan ole riittävästi tietoa oikeista nostotekniikoista, jotta mahdollisimman hyödyllisen palautteen antaminen olisi mahdollista [3].

Projekti toteutetaan Jyväskylän yliopiston tietotekniikan sovellusprojektina kevään 2004 aikana. Projektiin liittyvistä käytännöistä ja toimintatavoista sekä projektiin kuuluvista henkilöistä on kerrottu tarkemmin projektisuunnitelmassa [4].

Tässä dokumentissa kuvataan sovelluksen testaamisessa käytettäviä menetelmiä ja käytäntöjä. Dokumentin avulla testaus pyritään suorittamaan järjestelmällisesti, luotettavasti ja toistettavasti. Sovellusta testataan vaatimusmäärittelyssä [5] mainittujen vaatimusten pohjalta. Luvussa 3 on kerrottu sovelluksen testausympäristöstä, eli testauksessa käytettävistä laitteistoista ja ohjelmistoista. Luvussa 4 kerrotaan testauksen periaatteista, ja luvussa 5 käydään läpi testaamisessa käytettävät testitapaukset.

2 Termistöä

Tässä luvussa selitetään yleisimmät sovelluksen testaamiseen liittyvät termit.

AVI	Microsoftin tiedostomuoto äänelle ja liikkuvalla kuvalla. Tiedostot ovat yhteensopivia sekä PC:lle että Applen Macintoshille. AVIin sisältyy CODEC-tiedosto, jota tarvitaan videotiedostoja käyttävissä ohjelmissa tiedon pakkaukseen ja purkuun.
Alfa-testaus	Alfa-testauksessa tutkitaan täyttääkö esimerkiksi tilaustyönä tehty ohjelmisto asiakkaan asettamat vaatimukset. Tätä vaihetta jatketaan, kunnes systeemin kehittäjä ja asiakas hyväksyvät tuotteen ominaisuudet [2].
Beta-testaus	Yleensä laaja testausjakso ohjelmistolle ennen sen varsinaista käyttöönottoa ja jakelua [1].
FireWire (IEEE 1394)	Apple Computerin rekisteröimä tuotemerkki lisälaitteiden liitäntäteknikaksi mikrotietokoneille. FireWire tunnetaan paremmin nimellä IEEE 1394. Se on nopea (100, 200, 400 ja 800 Mbps), digitaalista sarjamuotoista dataa välittävä väylä [1].
Koodekki	Ohjelma tai laite, jota käytetään informaation pakkaamiseen ja purkamiseen. (engl. <i>code & decode= codec</i>).

3 Testausympäristö

Tässä luvussa esitellään sovelluksen testaukseen käytettävät laitteistot ja ohjelmistot. Testilaitteistona käytetään pääasiassa KIHU:ta ja tietotekniikan laitoksen suunnittelija Jani Kurhiselta lainattavia kannettavia tietokoneita.

3.1 Laitteistot

Nostoanalysointijärjestelmään kuuluva digitaalinen videokamera liitetään tietokoneeseen yleensä FireWire-liitännän kautta. Jotkin kamerat pystytään liittämään myös USB-väylän uudemman 2.0 version kautta. Projektiryhmän käytössä olevista pöytätietokoneista ei löydy kumpaakaan edellä mainituista liitännöistä. Niitä ei ole myöskään sovellusprojektien käytössä olevassa kannettavassa tietokoneessa. Tämän takia sovelluksen testaamiseksi aidoissa käyttötapauksissa projektiryhmä lainaa KIHU:ta tai suunnittelija Jani Kurhiselta tietotekniikan laitokselta kannettavan tietokoneen. Sellaisten testitapausten osalta, joissa ei tarvita videokameraa, testataan sovellusta projektiryhmän koneilla valmiiksi nostosuorituksista nauhoitettujen videoiden avulla.

Todellisia testitapauksia varten tarvitaan tietokoneen ja siihen asennetun sovelluksen lisäksi digitaalinen videokamera, joka kytketään tietokoneeseen, sekä ledi, jota videokameran välittämästä kuvasta seurataan. Videokamera ja ledi saadaan lainaksi KIHU:ta. Testaamista varten KIHU on luvannut projektiryhmän käyttöön tilat, joissa sovellusta voidaan testata käyttäen oikeaa painonnostotankoa ja siihen kiinnitettyä lediä.

Testaamisessa käytettyjen tietokoneiden tekniset tiedot mainitaan sovelluksen testaamisen yhteydessä laadittavassa testausraportissa.

3.2 Ohjelmistot ja käyttöjärjestelmät

Sovellus on suunniteltu Windows-ympäristöön ja testauksessa käytettävissä tietokoneissa on joko Windows XP tai 2000 -käyttöjärjestelmä. Testaamista ei voida suorittaa Windowsin vanhemmilla versioilla. On oletettavaa, että kaikissa koneissa, joista löytyy kameran liittämiseen tarvittava liitäntä, on myös ajan tasalla oleva käyttöjärjestelmä. Käytännössä tämä tarkoittaa Windows XP:tä tai 2000:a, jotka ovat ominaisuuksiltaan hyvin pitkälti toisiaan vastaavia.

Sovelluksella nauhoitettuja videoita testataan KIHUn KIHUVIEWER-ohjelmalla.

4 Testauksen periaatteet

Tässä luvussa kerrotaan testauksen yleisistä periaatteista. Tärkein periaate on sovelluksen jatkuva testaaminen. Ohjelmakoodia, käyttöliittymää ja ohjelmanosien toimivuutta yhdessä testataan jatkuvasti. Testitapaukset toteutetaan sovelluksen ohjelmoinnin päätyttyä.

4.1 Testauksen tavoite

Sovelluksen testauksen tavoitteena on tehdä sovelluksesta mahdollisimman virheetön ja käyttötarkoitustaan vastaava. Täydelliseen virheettömyyteen pääseminen on mahdotonta, joten testaamisessa keskitytään olennaisimpiin asioihin eli testataan ensisijassa sovellukselle vaatimusmäärittelyssä [5] asetetut vaatimukset.

4.2 Ohjelmakoodin testaus

Ohjelmakoodia testataan jatkuvasti sitä kirjoitettaessa. Useimmat virheet löydetään näin mahdollisimman aikaisessa vaiheessa, jolloin niiden korjaaminen helpottuu. Ohjelmankehitysympäristön kääntäjän asetukset laitetaan mahdollisimman tiukalle, tällöin kääntäjä löytää ohjelmakoodista pahimpia virheitä. Ohjelmakoodin testaus suoritetaan moduuli- ja integrointitestauksena.

4.2.1 Moduulitestaus

Sovelluksen luokkia ja ikkunoita ohjelmoidaan ja testataan erillisinä yksiköinä eli moduuleina. Moduulitestauksen avulla on mahdollista löytää virheet mahdollisimman aikaisessa vaiheessa jo ennen kuin yksiköt integroidaan yhteen.

4.2.2 Integroititestausta

Sovelluksen osien integroimista yhteen testataan jatkuvasti sovelluksen kehittämisen aikana. Lopullinen integroititestausta suoritetaan ennen testitapausten suorittamista. Tällöin sovelluksen yksiköitä käsitellään jo kokonaisuutena. Integroititestausta jälkeen suoritetaan järjestelmätestausta, jossa testataan sovellusta osana kokonaisuutena. Nostoanalysointijärjestelmää testitapausten avulla.

4.3 Testitapausten toteuttaminen

Sovelluksen testaamisessa käytetyt testitapausta on esitelty luvussa 5. Ne toteutetaan järjestyksessä projektiryhmän jäsenten toimesta, kun sovelluksen katsotaan olevan vaatimusten osalta valmis. Testitapaustailla pyritään paitsi testaamaan sovelluksen eri osat, myös erityisesti niiden toimivuus yhdessä.

4.4 Sovelluksen testikäyttö

Sovellus luovutetaan tilaajalle testikäyttöä varten samaan aikaan, kun projektiryhmä alkaa toteuttamaan testitapausta. Testausta tilaajan kanssa suoritetaan ensin alfa-testausta, jossa joku projektiryhmän jäsen seuraa tilaajan edustajaa käyttämässä sovellusta antaen samalla käyttäjäkoulutusta. Alfa-testaustailla pyritään selvittämään vastaavatko sovelluksen ominaisuudet sille asetettuja vaatimuksia. Alfa-testausta jatketaan niin kauan kun on aikaa ja se nähdään tarpeelliseksi.

Kun projektiryhmä on saanut kaikki testitapausta suoritettua ja niiden mukaiset sekä alfa-testaustaessa esille tulleet korjaukset tehtyä sovellukseen, aloitetaan sovelluksen beta-testausta. Siinä loppukäyttäjä eli tilaaja käyttää sovellusta sen oikeassa käyttötarkoituksessa raportoiden itsenäisesti projektiryhmälle mahdollisista virheistä ja ongelmatilanteista.

Beta-testaustauksen päätyttyä korjataan siinä ilmenneet virheet ja ongelmat ja sovelluksesta tehdään virallinen julkaisu.

4.5 Virheiden raportointikäytäntö

Toteutetut testitapaukset kirjataan ylös testiraporttiin, josta esimerkki taulukossa 4.1. Testiraportit sisällytetään liitteenä projektin testausraporttiin.

Testiraportti		
Testaajan tiedot		
Nimi:	Ville Räisänen (VR)	
Pvm:	26.4.2004	
Laitteisto:	1	
Testitapaukset		
OK	Virhe	Kuvaus
1		Toimii oikein.
2		Toimii oikein.
	3	Syötettäessä tekstiä kenttään, johon pitäisi syöttää numeroita, kaatuu koko ohjelma antaen virheilmoituksen.

Taulukko 4.1: Testiraportti.

Kustakin testiraportista selviää testauksen suorittaja, päivämäärä ja laitteisto, jolla testaus on suoritettu. Testilaitteistojen kuvaukset mainitaan testausraportissa erillisessä taulukossa, josta esimerkkinä taulukko 4.2. Testitapaukset erotellaan virheittä suoritettuihin ja sellaisiin, joiden suorituksen yhteydessä havaitaan virheitä. Havaituista virheistä kirjataan kuvaus virheen aiheuttaneesta toiminnasta ja sen seurauksista. Virheet ja niiden korjaukset kirjataan ylös myös virheraporttiin, josta esimerkki taulukossa 4.3.

Kukin testaaja siis täyttää testitapauksia suorittaessaan testiraportin ja kirjaa virheet ylös virheraporttiin. Testitapauksia käydään läpi ja niissä havaitut virheet korjataan, kunnes kaikki tapaukset voidaan suorittaa oikein. Toteutetuista testitapauksista tehdään yhteenveto testausraporttiin.

Nro	Laitteisto:
1	kannettava Intel P4 3,2 GHz, 512Mb ram, näytönohjain GeForce3, firewire-liitäntä, kamera Sony Digital VideoCam DV400, käyttöjärjestelmä Windows XP
2	kannettava Intel Celeron 2 GHz, 256Mb ram, näytönohjain GeForce2, firewire-liitäntä, kamera Sony Digital VideoCam DV400, käyttöjärjestelmä Windows 2000
3	pöytäkone Intel Celeron 800 MHz, 256Mb ram, näytönohjain GeForce2 MX400, ei kameraa, käyttöjärjestelmä Windows XP

Taulukko 4.2: Testilaitteistot.

Virheraportti		
ID	Havaittu	Korjattu
3	26.4.2004, VR	27.4.2004, VR
9	26.4.2004, VR	
11	26.4.2004, VR	

Taulukko 4.3: Virheraportti.

5 Testitapaukset

Tässä luvussa kuvataan testaamisessa toteutettavat testitapaukset. Niissä pyritään mahdollisimman suureen kattavuuteen. Kustakin testitapauksesta on kerrottu edellytykset eli vaaditut esiehdot, toiminto ja lopputila sekä mahdollisia huomautuksia testitapauksen suorittamiseen liittyen.

5.1 Sovelluksen toiminnallisuus

5.1.1 Nostajien lisääminen ja poistaminen

ID	Testitapaus	
1	Edellytys:	Sovellus toimintavalmis, ei tarvita kameraa.
	Toiminto:	Uuden nostajan lisääminen suoritusryhmään / nostajan lisääminen rekisteritiedostosta.
	Lopputila:	Uusi nostaja lisätty suoritusryhmään / rekisteritiedostoon.
	Huomautuksia:	
2	Edellytys:	Sovellus toimintavalmis, ei tarvita kameraa.
	Toiminto:	Nostajan poistaminen suoritusryhmästä / rekisteritiedostosta.
	Lopputila:	Valittu nostaja poistettu.
	Huomautuksia	Jos on nauhoitettu suorituksia joita ei ole tallennettu, sovellus kysyy niiden tallennetaanko ne.
3	Edellytys:	Sovellus toimintavalmis, ei tarvita kameraa.
	Toiminto:	Toisen samannimisen nostajan lisääminen rekisteritiedostoon / suoritusryhmään.
	Lopputila:	Uusi nostaja lisätty suoritusryhmään / rekisteritiedostoon.
	Huomautuksia:	Käyttäjä voi halutessaan lisätä toisen samannimisen nostajan.

5.1.2 Kalibrointi

4	Edellytys:	Sovellus toimintavalmis, kamera kytkettynä ja sen kuva näkyy ruudulla.
	Toiminto:	Kalibrointi: valitaan kaapatusta kuvasta neljä pistettä ja syötetään pisteiden väliset etäisyydet.
	Lopputila:	Sovellus on laskenut annettujen pisteiden ja etäisyyksien perusteella kalibrointiparametrit ja on valmiina nauhoittamaan suorituksia.
	Huomautuksia:	

5	Edellytys:	Sovellus toimintavalmis, kamera kytkettynä ja sen kuva näkyy ruudulla.
	Toiminto:	Kalibrointi: valittujen mittapisteiden siirtäminen, ja parametrien laskeminen uudelleen.
	Lopputila:	Sovellus on laskenut uudet kalibrointiparametrit ja on valmiina nauhoittamaan suorituksia.
	Huomautuksia:	Kalibrointiparametrit voidaan laskea uudelleen mittapisteiden siirtämisen jälkeen.

5.1.3 Suorituksen läpivienti ja vertailu

6	Edellytys:	Sovellus toimintavalmiina, kamera kytkettynä, nauhoitetaan nostosuoritus. Tiedetään tarvittavat asiat.
	Toiminto:	Oikea testitilanne: ledi kiinnitettynä levytangon päähän, tiedossa jonkin esineen mitat kalibrointia varten, nostajan paino ja pituus sekä levytangon paino.
	Lopputila:	Sovellus seuraa lediä ja piirtää liikeratakäyrän, nauhoittaa suorituksesta videon sekä laskee biomekaanisten muuttujien arvot.
	Huomautuksia:	
7	Edellytys:	Sovellus toimintavalmis, kamera kytkettynä, kamerassa digitaalinen videokasetti paikoillaan.
	Toiminto:	Videonpätkien tallentaminen kameran kasetille ja siirtäminen tietokoneelle.
	Lopputila:	Kameran kasetille tallennetut videot on siirretty tietokoneen levyille haluttuun paikkaan.
	Huomautuksia:	
8	Edellytys:	Sovellus toimintavalmis, nauhoitettu nostosuorituksia, jotka ovat liftscontainerissa.
	Toiminto:	Harjoituksesta nauhoitettujen videoiden tallentaminen.
	Lopputila:	Tallennettaviksi valitut videot tallennettu ja loput poistettu sovelluksen sulkemisen yhteydessä.
	Huomautuksia:	Myös muut suorituksen tiedostot tallennetaan / poistetaan samalla.
9	Edellytys:	Sovellus toimintavalmis, kamera kytkettynä.
	Toiminto:	Nauhoittaminen eri koodekeilla.
	Lopputila:	Suoritus on nauhoitettu valittua koodekkia käyttäen.
	Huomautuksia:	Kaikkien koodekkien käyttö ei ole välttämättä järkevää, tulee ilmoittaa jotenkin, minkä koodekkien käyttöä suositellaan.

10	Edellytys:	Sovellus toimintavalmis, kamera kytkettynä, yksi harjoitus tehty aikaisemmin samana päivänä.
	Toiminto:	Uuden harjoituksen läpivienti samana päivänä. (tiedostonnimien käyttäytyminen)
	Lopputila:	Suoritukset on nauhoitettu ja analysoitu.
	Huomautuksia:	Tiedostonnimien perusteella pystytään yksikäsitteisesti tunnistamaan nostojen tiedostot.
11	Edellytys:	Sovellus toimintavalmis, ei tarvita kameraa, nauhoitettu nostosuorituksia, jotka liftscontainerissa / tiedostossa.
	Toiminto:	Kahden nostosuorituksen vertailu, molemmat liftscontainerissa / toinen / molemmat avataan tiedostosta.
	Lopputila:	Valitut nostosuoritukset haettu ja valmiina vertailua varten.
	Huomautuksia:	
12	Edellytys:	Sovellus toimintavalmis, ei tarvita kameraa, valitaan kaksi videota vertailtavaksi.
	Toiminto:	Vertailtavien videoiden pysyminen synkronissa, kun niitä kelataan mielivaltaisesti.
	Lopputila:	Videot pysyvät synkronissa.
	Huomautuksia:	
13	Edellytys:	Sovellus toimintavalmis, löytyy kaksi suoritusta joita halutaan vertailla.
	Toiminto:	Kahden nostosuorituksen vertailu, kun vertailuikkuna ei ole auki.
	Lopputila:	Sovellus on avannut vertailuikkunan automaattisesti.
	Huomautuksia:	
14	Edellytys:	Sovellus toimintavalmiina, nauhoitetaan nostosuoritus, josta analysoidaan muuttujat ja liikeratakäyrä.
	Toiminto:	Suorituksen tiedostojen tallentuminen haluttuun paikkaan.
	Lopputila:	Kaikki tiedostot tallennettu samaan, haluttuun paikkaan.
	Huomautuksia:	Tallennushakemisto voidaan valita optionsista.

5.1.4 Virhetilanteet

15	Edellytys:	Sovellus toimintavalmis, ei tarvita kameraa.
	Toiminto:	Väärin arvojen syöttäminen uutta nostajaa lisättäessä / nostajan tietoja muutettaessa.
	Lopputila:	Väärää syöttötietoja ei ole hyväksytty, vaaditaan oikeanlaisia tilalle.
	Huomautuksia:	

16	Edellytys:	Sovellus toimintavalmis, ei tarvita kameraa, valittu suoritus näytettäväksi, jonka muuttujien arvot näkyvät taulukossa.
	Toiminto:	Arvojen syöttäminen käsin muuttujataulukkoon.
	Lopputila:	Käsin syötettyjen arvojen päälle on kirjoitettu uudet arvot, tai arvojen syöttäminen ei ole onnistunut.
	Huomautuksia:	Ei ole mahdollista / ei sekoita sovelluksen toimintaa.
17	Edellytys:	Sovellus toimintavalmis, ei tarvita kameraa.
	Toiminto:	Nostajan lisääminen tiedostoon, kun tiedostoa ei ole avattuna.
	Lopputila:	Sovellus huomauttaa, ettei tiedostoa ole avattuna.
	Huomautuksia:	
18	Edellytys:	Sovellus toimintavalmis, kamera kytkettynä ja sen kuva näkyy ruudulla.
	Toiminto:	Nauhoituksen aloittaminen ennen kuin kalibrointi suoritettu.
	Lopputila:	Nauhoitusta ei ole aloitettu, sovellus pyytää suorittamaan ensin kalibroinnin.
	Huomautuksia:	Nauhoittamista ei voi aloittaa ennen kalibrointia.
19	Edellytys:	Sovellus toimintavalmis, kamera kytkettynä.
	Toiminto:	Nauhoituksen aloittaminen, kun nostajaa / painoja / nostotekniikkaa ei ole valittuna.
	Lopputila:	Nauhoitusta ei ole aloitettu. Sovellus pyytää antamaan puuttuvan tiedon.
	Huomautuksia:	
20	Edellytys:	Sovellus toimintavalmis, kamera kytkettynä, nauhoitetaan suoritus ja tallennustila loppuu kesken nauhoituksen.
	Toiminto:	Tallennustilan loppuminen kesken.
	Lopputila:	Nostosuorituksen tallentaminen on katkaistu, sovellus ilmoittaa asiasta.
	Huomautuksia:	Sovelluksen tulee säilyä toimintavalmiina.
21	Edellytys:	Sovellus toimintavalmis, kamera kytkettynä, nauhoitetaan nostosuoritus.
	Toiminto:	Kameran välittämän kuvan katoaminen kesken nauhoituksen. (esim. johto irtoaa)
	Lopputila:	Nostosuorituksen tallentaminen on katkaistu, sovellus palautunut tilanteesta.
	Huomautuksia:	Sovelluksen tulee säilyä toimintavalmiina.

22	Edellytys:	Sovellus toimintavalmis, nauhoitettu suoritus, josta tarvittavat tiedostot olemassa, tiedostoissa väärää arvoja esim. käsin syötettynä.
	Toiminto:	Korruptoituneen tiedoston lukeminen.
	Lopputila:	Sovellus palautunut tiedoston lukemisesta, väärät arvot sivuutettu.
	Huomautuksia:	rekisteritiedosto, liikeratakäyrätiedosto
23	Edellytys:	Sovellus toimintavalmis, ei tarvita kameraa, nauhoitettu kaksi nostosuoritusta joita halutaan vertailla.
	Toiminto:	Kahden nostosuorituksen vertailu, toisesta ei löydy liikeratatiedostoa tai analysointitulostiedostoa.
	Lopputila:	Vertailu käynnistetty, vaikka toisesta suorituksesta puuttuikin tiedostoja.
	Huomautuksia:	

5.2 Sovelluksen ikkunoiden ja valikoiden käsittely

24	Edellytys:	Sovellus toimintavalmis, ei tarvita kameraa.
	Toiminto:	Sovelluksen ikkunoiden sulkeminen mielivaltaisissa järjestyksissä ja luominen uudelleen.
	Lopputila:	Sovellus säilynyt toimintavalmiina, kaikki ikkunat saatu palautettua / luotua uudelleen.
	Huomautuksia:	Suoritusryhmäikkunaa ei voi sulkea, jos ryhmässä on nostajia.
25	Edellytys:	Sovellus toimintavalmis, ei tarvita kameraa, details-ikkuna auki.
	Toiminto:	Details-ikkunan käsittely samanaikaisesti suoritusryhmä- ja nostajarekisteri-ikkunoista.
	Lopputila:	Details-ikkunan tiedot päivitetty oikein.
	Huomautuksia:	Details-ikkuna pitää avata nostajarekisteri-ikkunasta, muuten voidaan avata kaksi details-ikkunaa.
26	Edellytys:	Sovellus toimintavalmis, ei tarvita kameraa.
	Toiminto:	Menujen testaus, kaikkien valintojen kokeileminen.
	Lopputila:	Menuvalinnat ovat toteuttaneet halutun toiminnon.
	Huomautuksia:	Kaikki painikkeiden toiminnot saatavilla myös menusta, toiminta täsmälleen sama. Menuissa ei tyhjiä valintoja.

27	Edellytys:	Sovellus toimintavalmis, kamera kytkettynä / ei kameraa.
	Toiminto:	Options -valintojen testaaminen.
	Lopputila:	Options -valinnat muuttaneet asetuksia halutuiksi.
	Huomautuksia:	Videonkaappausasetuksia voi muuttaa vain, jos kamera kytkettynä, muita asetuksia aina.
28	Edellytys:	Sovellus toimintavalmis, kamera kytkettynä / ei kameraa, voidaan muuttaa näytön resoluutiota.
	Toiminto:	Käytettävyys erilaisilla resoluutioilla. Muutetaan näytön resoluutiota ja testataan sovelluksen käyttämistä.
	Lopputila:	Käytettävyys säilynyt kohtuullisena resoluutiosta riippumatta.
	Huomautuksia:	

5.3 Sovelluksen analyysiosa

29	Edellytys:	Sovellus toimintavalmiina, kamera kytkettynä, nauhoitetaan nostosuoritus.
	Toiminto:	Analyysitulostiedoston kirjoittaminen ja lukeminen.
	Lopputila:	Analyysitulostiedoston käsittely onnistunut oikein.
	Huomautuksia:	
30	Edellytys:	Sovellus toimintavalmiina, nauhoitettu nostosuoritus, jonka analysointi halutaan tehdä uudestaan.
	Toiminto:	Suorituksen uudelleenanalysointi. Avataan vanha suoritus, joka analysoidaan uudelleen uusilla painoilla. Myös liikeratakäyrä voidaan laskea uudelleen.
	Lopputila:	Suoritus on analysoitu uudelleen, uudet analyysitulostiedostot ja / tai liikeratakäyrätiedostot kirjoitettu.
	Huomautuksia:	Levytangon painoa voidaan muuttaa uudelleenanalysointia varten. Liikeratakäyrän uudelleenlaskemiseksi sovellus täytyy kalibroida.
31	Edellytys:	Sovellus toimintavalmiina, kamera kytkettynä, nauhoitetaan nostosuoritus, muuttujien viitteelliset arvot tiedossa.
	Toiminto:	Mitattujen biomekaanisten muuttujien arvojen tarkastaminen.
	Lopputila:	Muuttujien arvot eroavat oikeista vain ilmoitettujen virhemarginaalien verran.
	Huomautuksia:	

32	Edellytys:	Sovellus toimintavalmiina, kamera kytkettynä, nauhoitetaan nostosuoritus. Mahdollisuus käyttää erivärisiä ja -kokoisia ledejä.
	Toiminto:	Virhemarginaalin käyttäytyminen erilaisilla ledeillä. Nauhoitetaan ja analysoidaan suorituksia käyttäen levytangossa erilaisia ledejä.
	Lopputila:	Nostosuuritukset nauhoitettu ja analysoitu, virhemarginaalit laskettu.
	Huomautuksia:	Virhemarginaalin pitäisi pysyä tietyissä rajoissa erilaisia ledejä käytettäessä.
33	Edellytys:	Sovellus toimintavalmiina, nauhoitettu nostosuoritus, jonka analyysitulostiedosto luetaan.
	Toiminto:	Korruptoituneen analyysitulostiedoston lukeminen. (väärää arvoja)
	Lopputila:	Väärät arvot analyysitulostiedostossa sivuutettu.
	Huomautuksia:	
34	Edellytys:	Sovellus toimintavalmiina, nauhoitettu nostosuoritus, josta olemassa analyysitulokset.
	Toiminto:	Eri muuttujien arvojen näyttäminen graafisella kuvaajalla.
	Lopputila:	Valitut muuttujat näytetään kuvaajana ja numeroina taulukossa.
	Huomautuksia:	

6 Lähteet

- [1] Jaakohuhta, Hannu, IT Ensyklopedia, Sanasto, Edita Oyj, Helsinki, 2001.
- [2] Kautto Tuomas, Ohjelmistotestaus ja siinä käytettävät työkalut, Ohjelmistotekniikan seminaariesitelmä, Jyväskylän yliopisto, saatavilla WWW-muodossa <URL: <http://www.mit.jyu.fi/opiskelu/seminaarit/ohjelmistotekniikka/testaus/>>, 21.11.1996.
- [3] Keränen Tapani, Viitasalo Jukka ym., Nostotekniikan automaattinen mittaus- ja analysointijärjestelmä-esittelymoniste, KIHU, Jyväskylä, 2003.
- [4] Laasala Lauri, Lukkarinen Olli, Räisänen Ville ja Tanhua-Tyrkkö Vesa, OptiLift-Sovellusprojektin projektisuunnitelma, Jyväskylän yliopisto, Tietotekniikan laitos, 2004.
- [5] Laasala Lauri, Lukkarinen Olli, Räisänen Ville ja Tanhua-Tyrkkö Vesa, OptiLift-Sovellusprojektin vaatimusmäärittely, Jyväskylän yliopisto, Tietotekniikan laitos, 2004.