



Antti Vänskä

SCALABLE INVENTORY SYSTEM

-JÄRJESTELMÄN TESTAUS, DOKUMENTOINTI

JA KÄYTTÖÖNOTTO

SCALABLE INVENTORY SYSTEM

-JÄRJESTELMÄN TESTAUS, DOKUMENTOINTI

JA KÄYTTÖÖNOTTO

Antti Vänskä
Opinnäytetyö
Syksy 2013
Hyvinvointiteknologian koulutusohjelma
Oulun seudun ammattikorkeakoulu

TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu
Hyvinvointiteknologian koulutusohjelma

Tekijä: Antti Vänskä

Opinnäytetyön nimi: Scalable Inventory System -järjestelmän testaus, dokumentointi ja käyttöönotto

Työn ohjaaja: Juha Alakärppä

Työn valmistumislukukausi ja -vuosi: Syksy 2013

Sivumäärä: 50 + 6 liitettä

Tässä opinnäytetyössä järjestettiin jo aiemmin toteutetulle Scalable Inventory System -nimiselle inventaariojärjestelmälle käyttöönottestaus, laadittiin koko järjestelmän kattava dokumentaatio ja otettiin järjestelmä käyttöön Oamkin tekniikan yksikössä. Viivakoodipohjainen inventaariojärjestelmä koostuu kahdesta C#-kielellä kirjoitetusta yksiköstä, joista SIS Clientilla toteutetaan tuotteiden lainaus- ja palautustoiminnot ja SIS Managementilla hallitaan tuotteita, tuotekategorioita ja tuotteita varastoivia tiloja MySQL-tietokannassa. Tavoitteena oli lisäksi tunnistaa järjestelmän jatkokehitystarpeet.

Ennen käyttöönottestausta järjestelmä sisälsi sen toiminnan kannalta kriittisiä ohjelmavirheitä, jotka paikannettiin ja korjattiin. Käyttöönottestaus suunniteltiin kattamaan kaikki SIS-yksiköiden perustoiminnot ja lukuisia erityistapauksia. Kolme testihenkilöä suoritti itsenäisesti kaikki testitapaukset ja täytti testauspöytäkirjat, joiden pohjalta laadittiin testausraportti. Testauksen tulosten varmistuttua tehtiin tarvittavat korjaukset ja järjestettiin järjestelmän käyttöönotto asentamalla tarvittavat laitteistot ja ohjelmat erääseen hanketilaan ja inventoimalla sen tuotteet järjestelmään. Lisäksi järjestelmästä laadittiin englanninkielinen tekninen dokumentaatio ja sitä tukeva Doxygen-dokumentointityökalulla generoitu tiivistelmä lähdekoodin rakenteesta.

Opinnäytetyön tuloksena Scalable Inventory System saatettiin toimintakuntoon ja varsinaiseen käyttöön. Työn ohessa syntyi lukuisia kehitysideoita, joiden toteuttamista voidaan harkita tulevilla projekteilla. Avuksi uusille sovelluskehittäjille laadittiin kattava dokumentaatio, jonka avulla perehtyminen järjestelmän rakenteeseen on jatkossa helpompaa. Oheistuotteena syntyi myös ohje C#-ohjelmointikielen dokumentoinnista, jota hyödynnetään tulevilla projekteilla yhdenmukaistamaan lähdekoodin kommentointi- ja dokumentointikäytäntöjä.

Asiasanat:

Inventaariojärjestelmä, C#, MySQL, Doxygen, testaus, dokumentointi, käyttöönotto

ABSTRACT

Oulu University of Applied Sciences
Medical Engineering

Author: Antti Vänskä

Title of thesis: Deployment Testing, Documentation and Deployment of Scalable Inventory System

Supervisor: Juha Alakärppä

Term and year when the thesis was submitted: Autumn 2013

Pages: 50 + 6 appendices

In this bachelor's thesis formerly completed Scalable Inventory System was performed a deployment testing, documented thoroughly and finally deployed. The system consists of two C# written units: SIS Client offers borrow and return functions and SIS Management offers control over items, categories and locations in MySQL database. One goal was also to recognize needs for further development.

Before deployment testing the system had some major functional bugs that were located and fixed. Deployment testing was designed to cover all the basic functions and many special cases of SIS units. Three testers run all the test cases independently and filled the testing records that were later combined to testing report. After the conclusion of testing was clear requisite fixes were made and the system was deployed by installing the proper hardware and software to one project lab and adding its items to the system. Additionally, a technical documentation for Scalable Inventory System was written and completed with a software reference documentation generated by Doxygen.

As a result Scalable Inventory System was made operational and put into use. Numerous ideas were born to improve the system and carrying out those ideas can be considered in the future. An inclusive documentation was prepared to help new developers to get to know the structure of the system a lot easier than before. An instruction manual for C# source code documenting was also made as a by-product. It can be utilized in the forthcoming projects to standardize the use of comments and document comments in source code.

Keywords:

Inventory system, C#, MySQL, Doxygen, testing, documenting, deployment

ALKULAUSE

Opinnäytetyö käynnistyi helmikuun 2013 loppupuolella ja sen oli määrä valmistua kesään mennessä. Aikataulu ei kuitenkaan pitänyt ja kesän aikana edistyminen oli huomattavasti hitaampaa työkiireiden vuoksi. Scalable Inventory System saatiin heinä-elokuun vaihteessa vihdoinkin asennettua hanketilaan, jossa opinnäytetyötäni pääasiassa tein. Loppuaika onkin kulunut dokumentoinnin parissa ja valmistuminen vuoden loppuun mennessä näyttää väistämättömältä.

Haluan kiittää työni tilaajaa Oulun seudun ammattikorkeakoulun tekniikan yksiköä ja erityisesti sen yhdyshenkilönä toiminutta Tommi Sallista mielenkiintoisesta ja opettavaisesta opinnäytetyön aiheesta sekä kaikesta saamastani avusta ja ohjauksesta. Haluan myös kiittää työni ohjaajaa Juha Alakärppää, kielenohjauksesta vastannutta Eija Mustosta sekä testihenkilöinä toimineita tekniikan yksikön opiskelijoita Juha Nordia, Pirkko Kukkoilyä ja Marko Vesalaa.

Kiitokset kuuluvat myös ystäväilleni Mikalle ja Samulille, joiden kanssa työn dokumentointiosuus eteni tehokkaasti muodostamassamme opintopiirissä Oulun yliopistolla. Erityisen lämpimät kiitokset vaimolleni Suville, jonka kärsivällisyydellä ja luottamuksella oli vahva rooli tämän työn valmistumisen kannalta.

Oulussa 11.11.2013

Antti Vänskä

SISÄLLYS

TIIVISTELMÄ	3
ABSTRACT	4
ALKULAUSE	5
SISÄLLYS	6
SANASTO	8
1 JOHDANTO	10
2 SCALABLE INVENTORY SYSTEM	11
2.1 SIS Client	12
2.2 SIS Management	16
2.3 Tietokanta	19
3 KÄYTETYT MENETELMÄT	22
3.1 Ohjelmisto	22
3.2 Laitteisto	23
4 TOIMENPITEET	24
4.1 Ohjelmavirheiden korjaus	24
4.2 Käyttöönottestaus	27
4.3 Käyttöönotto	28
4.4 Dokumentointi	30
5 TULOKSET	33
6 JATKOKEHITYSTARPEET	34
6.1 SIS Client – Palautetta lainauksesta ja palautuksesta	34
6.2 SIS Client – Tietojen lajittelu	36
6.3 SIS Client – Kosketusnäyttöominaisuuksien hyödyntäminen	36
6.4 SIS Client – Lainaajan tietojen hyödyntäminen	37
6.5 SIS Client – Väärän viivakoodin lukeminen	38
6.6 SIS Management – Säilytyspaikat tilan sisällä	38
6.7 SIS Management – Olemattoman kategorian tai tilan poistaminen	40
6.8 SIS Management – Lainaajan tietojen tallentaminen	41
6.9 SIS Management – Suodatinten ja haun laajentaminen	41
6.10 SIS Management – Usean tuotteen muokkaaminen	42
6.11 SIS Management – Viivakoodin lukemisen hallinta	43

6.12 SIS Management – Väärän viivakoodin lukeminen	44
6.13 SIS Management – Tilan ja tuotekategorian uudelleennimeäminen	44
6.14 SIS Management – Tulostustapahtuman nimen muuttaminen	45
6.15 Tietoturvakorjaus	45
7 YHTEENVETO	47
LÄHTEET	49
LIITTEET	51

SANASTO

C# (C Sharp)	Microsoftin .NET-konseptin pääohjelmointikieli, joka kehitettiin yhdistämään C++:n tehokkuus ja Java-kielen helppokäyttöisyys.
COM	Component Object Model: Microsoftin kehittämä ohjelmistoarkkitehtuuri, joka helpottaa komponenttien yhteiskäyttöä. COM:in avulla eri valmistajien tekemät komponentit voidaan yhdistää toimivaksi kokonaisuudeksi kieliriippumattomasti.
DLL	Dynamic Link Library: Windows-ympäristössä käytetty ajonaikainen kirjastotiedosto, jolla voidaan jakaa ohjelmakoodia ja dataa useiden ohjelmien kesken.
Doxygen	Useita eri ohjelmointikieliä tukeva dokumentointityökalu, joka osaa tehdä dokumentoiduista lähdekooditiedostoista rakenteisen dokumentin HTML:nä, Latexina, RTF:nä, PostScriptina, PDF:nä, pakattuna HTML:nä ja Unixin man-sivuina.
HTML	HyperText Markup Language: kuvauskieli, jolla kuvataan hyperlinkkejä sisältävää tekstiä eli hypertekstiä. HTML on yleisin nettisivujen koodauksessa käytetty kieli.
MySQL	My Structured Query Language: SQL standardia noudattava relaatiotietokantaohjelmisto, joka on erittäin suosittu web-palveluiden käytössä.
.NET Framework	Microsoftin kehittämä ohjelmistokomponenttikirjasto, jota Microsoftin VisualStudio.NET-ympäristössä kehitetyt ohjelmistot käyttävät.
PHP	Hypertext Preprocessor: palvelinympäristöissä dynaa-

misten web-palveluiden tuottamiseen tarkoitettu ohjelmointikieli.

SIS Scalable Inventory System: keväällä 2012 opiskelijaprojekteina suunniteltu ja toteutettu joustava viivakoodipohjainen inventaariojärjestelmä, joka koostuu kahdesta yksiköstä: SIS Clientista ja SIS Managementista.

Windows Forms Microsoftin .NET-kehiksen graafinen ohjelmointirajapinta, joka tarjoaa perinteisen tavan luoda käyttöliittymäelementtejä Windows-ympäristössä.

WPF Windows Presentation Foundation: DirectX-rajapinnan päälle rakennettu kirjasto, joka tarjoaa modernin graafisen rajapinnan uudemille Windows-käyttöjärjestelmille. Windows Formsista poiketen WPF eriyttää käyttöliittymän ohjelmakoodista kuvaamalla sen XML-pohjaisella XAML-kielillä (eXtensible Application Markup Language).

XML eXtensible Markup Language: järjestelmäriippumaton kuvauskieli, joka on tarkoitettu rakenteisen tiedon esittämiseen ja laajojen tietomassojen jäsentämiseen. XML-kieltä käytetään sekä formaattina tiedonvälitykseen järjestelmien välillä että formaattina dokumenttien tallentamiseen.

1 JOHDANTO

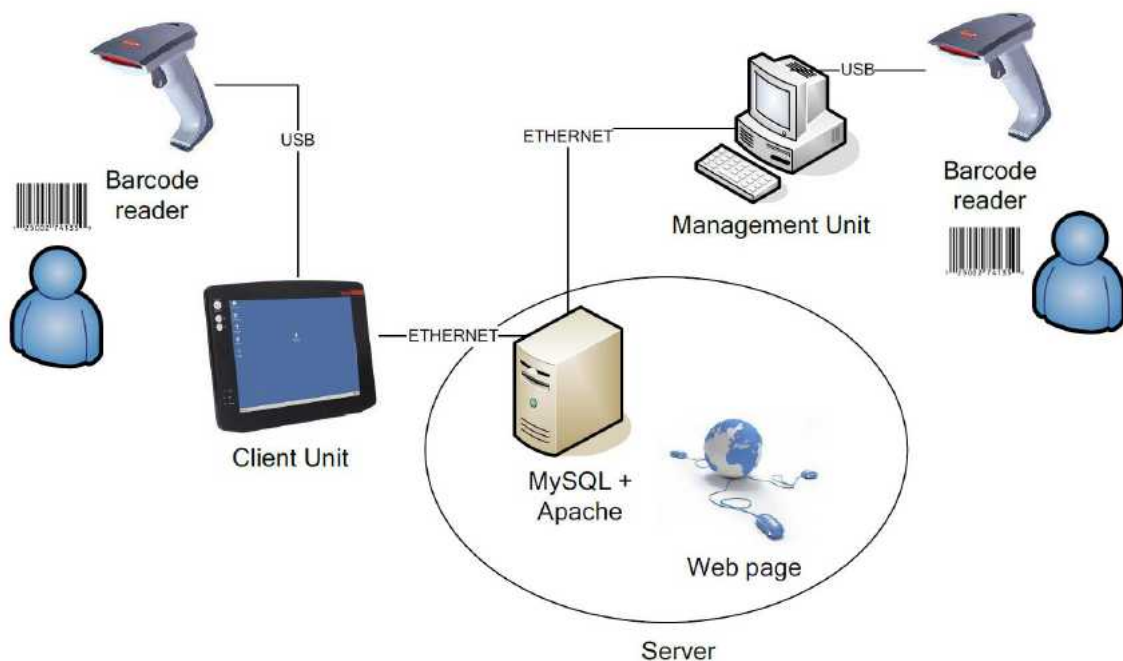
Oulun seudun ammattikorkeakoulun (Oamk) tekniikan yksikköön on vuosien saatossa kerääntynyt lukematon määrä erilaisia teknisiä tuotteita: laitteita, moduuleita, muuntajia, antureita, komponentteja, kehitystyökaluja ja muuta tavaraa. Tuotteita säilytetään useissa eri yksikön tiloissa ja niitä lainataan tarpeen mukaan toisiin tiloihin. Tuotteiden lisääntyessä kasvavaksi ongelmaksi on tunnistettu inventoinnin ja ajankohtaisen sijaintitiedon puuttuminen. Kun on tarvittu jotain tiettyä tuotetta, sitä ei ole välttämättä löydetty, koska se on ollut lainassa jossain muussa yksikön tilassa. Toisekseen tuote on voitu hankkia turhaan, koska ei ole ollut tietoa, että sellainen löytyy jo ennestään jostain toisesta yksikön tilasta. Ongelma on siis syönyt resursseja niin työajan kuin taloudellisten investointienkin muodossa.

Tätä ongelmaa ryhdyttiin ratkaisemaan keväällä 2012 kahden projektiryhmän voimin. Tuloksena syntyi tuotteiden hallintajärjestelmä nimeltä Scalable Inventory System (SIS). Se sisälsi kaksi itsenäistä yksikköä: tuotteiden lainaukseen ja palautukseen tarkoitetun SIS Clientin sekä tuotteiden hallinnointiin tarkoitetun SIS Managementin. Näiden lisäksi toteutettiin myös PHP-pohjainen verkkosivu, jolla tuotteita pystyi etsimään ja selaamaan nettiselaimen avulla. Sen osalta lopputulos ei kuitenkaan vastannut kaikkia vaatimuksia, joten sitä ei päätetty ottaa käyttöön, eikä sitä käsitellä tässä opinnäytetyössä. Client- ja Management-yksiköt valmistuivat ominaisuuksiensa osalta aikataulussa, mutta järjestelmää ei saatu otettua käyttöön muutamien kriittisten ohjelmavirheiden takia.

Opinnäytetyöni tilaaja, Oamkin yhdyshenkilö Tommi Sallinen, halusi järjestää Scalable Inventory System -järjestelmälle käyttöönottestauksen ja varsinaisen käyttöönoton. Tämä edellytti lähdekoodiin perehtymistä, ohjelmavirheiden paikantamista ja korjaamista sekä järjestelmän toimintakunnon varmistamista. Opinnäytetyön tavoitteisiin määriteltiin laatia koko järjestelmästä myös englanninkielinen dokumentaatio, joka sisälsi lisäksi Doxygen-ohjelmalla generoidun luokkadokumentaation tuottamisen. Lisäksi tarkoitus oli tunnistaa ja dokumentoida järjestelmään liittyviä jatkokehitystarpeita tulevien projektien hyödynnettäväksi. Tavoitteet löytyvät eriteltynä myös lähtötietomuistiosta (liite 1).

2 SCALABLE INVENTORY SYSTEM

Opinnäytetyön tekeminen käynnistyi järjestelmään perehtymisellä. SIS Clientin ja SIS Managementin lähdekoodit sekä asennustiedostot olivat saatavilla, mutta niitä ei ollut otettu varsinaiseen käyttöön. Ohjelmien asentaminen ja testaaminen PC:llä osoitti tilaajan mainitsemien ongelmien olemassaolon – järjestelmä ei ollut vielä toimintakunnossa. Järjestelmän kehitysvaiheessa laaditut raportit sekä lähdekoodien tutkiminen Microsoftin kehitysympäristö Visual Studiossa auttoivat avartamaan järjestelmän arkkitehtuuria (kuva 1).



KUVA 1. Scalable Inventory System -järjestelmän arkkitehtuuri (1, s. 1).

Järjestelmä koostuu kahdesta yksiköstä: SIS Clientistä ja SIS Managementista. Yksiköt ovat toiminnaltaan toisistaan riippumattomia, mutta niitä yhdistää yhteinen rajapinta: MySQL-tietokanta. Tietokantaa käytetään tuotetietojen säilyttämiseen ja yksiköillä on erilaiset mahdollisuudet esittää ja muokata sen sisältämää tietoa. Itse yksiköt koostuvat Windows-käyttöjärjestelmällä varustetusta PC-tietokoneesta, varsinaisesta ohjelmasta sekä USB-porttiin liitettävästä viivakoodinlukijasta (kuva 2). SIS Management sisältää tämän lisäksi USB-porttiin liitet-

tävän tarratulostimen, jonka avulla tuotteiden yksilöinti on mahdollista niiden tietokantaan luonnin jälkeen.



KUVA 2. SIS Client (vas.) ja SIS Management oheislaitteineen.

2.1 SIS Client

SIS Client on kevyt, Microsoftin .NET-kehiksen (framework) päällä toimiva Windows-ohjelma. Se suunniteltiin alun perin toimimaan Windows-pohjaisella tablet-laitteella, jonka ensisijainen osoitinlaite on kosketusnäyttö. Sen vuoksi päädyttiin käyttämään perinteisemmän Windows Forms -ohjelmointirajapinnan sijasta Windows Presentation Foundationia (WPF), jonka kosketusohjausta tukevat toiminnot ovat kehittyneemmät. Periaatteessa olisi hyvin voitu käyttää myös Windows Formsia, sillä kosketustoimintoja ei toteutuksessa lopulta kuitenkaan hyödynnetty. Tulevissa projekteissa WPF tarjoaa kuitenkin laajemmat mahdollisuudet käyttöliittymän kehittämiseen. Ohjelmointikielenä käytettiin .NETiä varten kehitettyä C#-kieltä. SIS Client toimii Vistassa ja sitä uudemmissa Windows-käyttöjärjestelmissä, mutta myös Windows XP:ssä, johon on asennettu WPF:n vaatimat kirjastot.

SIS Clientin käyttötarkoitus on mahdollistaa tuotteiden lainaaminen ja palauttaminen. Se on tarkoitettu asentaa jokaiseen tilaan, jossa säilytetään ja/tai johon

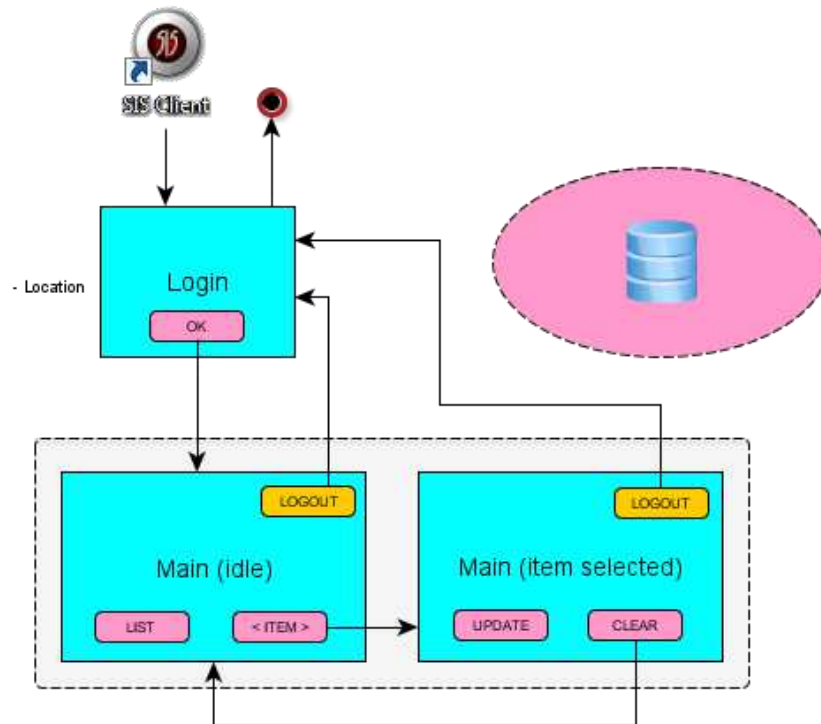
voidaan lainata tuotteita. Kun käyttäjä haluaa lainata tuotteen tilasta A tilaan B, hän käy läpi seuraavan tapahtumaketjun:

1. Käyttäjä noutaa tuotteen tilasta A.
2. Käyttäjä menee tilaan B ja käynnistää SIS Clientin.
3. Käyttäjä valitsee avautuvasta liukuvalikosta sijainnikseen tilan B.
4. Käyttäjä lukee viivakoodinlukijalla tuotteen laatikon kyljestä löytyvän viivakoodin.
5. Käyttäjä painaa Update-painiketta, jolloin tuote merkitään lainatuksi tilaan B.

Vaihtoehtoisesti käyttäjä voi suorittaa lainauksen jo tilassa A valitsemalla liukuvalikosta sijainnikseen tilan B. Tilan valintaa ei siis voi lukita tiettyyn tilaan, vaan ohjelma kysyy sitä joka käynnistyksen yhteydessä. Tuotteen palauttaminen tilasta B tilaan A toimii seuraavasti:

1. Käyttäjä palaa tuotteen kanssa tilaan A ja käynnistää SIS Clientin.
2. Käyttäjä valitsee avautuvasta liukuvalikosta sijainnikseen tilan A.
3. Käyttäjä lukee viivakoodinlukijalla tuotteen laatikon kyljestä löytyvän viivakoodin.
4. Käyttäjä painaa Update-painiketta, jolloin tuote merkitään palautetuksi tilaan A.

Myös palauttaminen voidaan suorittaa missä tilassa tahansa, mutta loogisinta se on tietysti tehdä siinä tilassa, johon tuote fyysisestikin palautetaan. SIS Clientillä voidaan ainoastaan lainata ja palauttaa tuotteita. Mikäli tilasta A tilaan B lainattua tuotetta yritetään palauttaa tilaan C, merkitään tuote vain lainatuksi tilasta A tilaan C. Tuotteen voi siis palauttaa vain sen alkuperäiseen säilytyspaikkaan, jota ei SIS Clientin avulla voida muuttaa. SIS Clientin käyttöliittymä koostuu kahdesta ikkunasta (kuva 3).



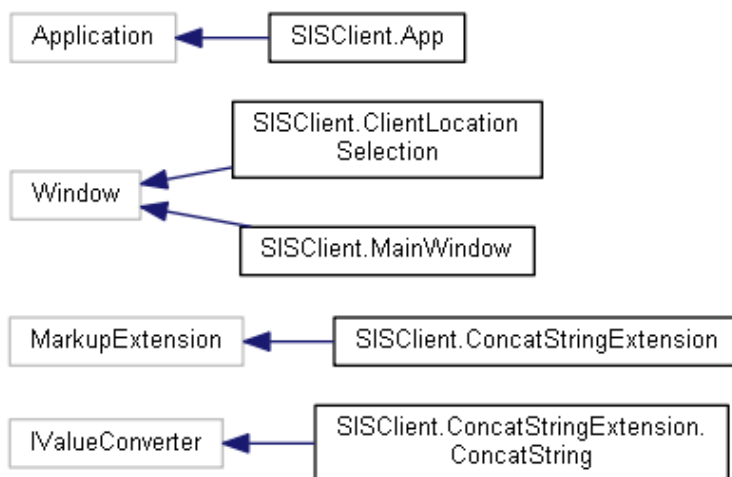
KUVA 3. SIS Clientin käyttöliittymän logiikka.

Kun SIS Client käynnistetään, avautuu ensimmäisenä kirjautumisikkuna (Login). Ohjelma pyytää valitsemaan liukuvalikosta tilan, jossa käyttäjä ohjelmaa käyttää. Valittuaan oikean tilan avautuu ohjelman pääikkuna (Main), jonka otsikkokentässä näkyy juuri valitun tilan nimi. Painamalla List-painiketta ohjelma tulostaa näytölle kaikki kyseiseen tilaan liittyvät tuotteet. Sekä kyseisessä tilassa säilytettävät tuotteet että sinne lainatut tuotteet tulostetaan näytölle. Tuotteita ei voida valita tai muokata tästä näkymästä, vaan ne ovat ainoastaan selattavissa. Logout-painikkeesta päästään takaisin kirjautumisikkunaan.

Varsinainen lainaaminen ja palauttaminen tapahtuu lukemalla käsiteltävän tuotteen viivakoodi, jolloin ohjelma tulostaa ainoastaan kyseisen tuotteen tiedot näytölle. Tämän jälkeen List-painike katoaa ja tilalle ilmestyy Update- ja Clear-painikkeet. Clear tyhjentää näytön ja palaa edelliseen tilaan. Update puolestaan joko lainaa tai palauttaa tuotteen riippuen siitä, missä tilassa ollaan. Ohjelma ei varsinaisesti anna mitään ilmoitusta lainauksen/palautuksen suorituksesta, vaan tuotteen tilan voi lukea Location- ja Borrowed Location -sarakkeista. Location

ilmaisee, mikä on tuotteen alkuperäinen sijainti. Jos Borrowed Location on tyhjä, tuote sijaitsee sarakkeen Location ilmaisemassa tilassa. Jos taas Borrowed Location sisältää jonkin tilan nimen, tuote on siinä tilassa lainassa.

SIS Clientin luokkahierarkia koostuu viidestä luokasta (kuva 4), joista Application-luokasta periytyvä App-luokka on WPF:n generoima. Sen tarkoitus on toimia ohjelman lähtöpisteenä eli se ajetaan ennen yhdenkään ikkunan avaamista. Siihen voidaan määritellä esimerkiksi tapahtumia ja poikkeustilakäsittelyitä, mutta yleensä sitä käytetään globaaleiden resurssien määrittämiseen (2, Working with App.xaml). SIS Clientin tapauksessa siihen ei ole tehty muutoksia. Window-luokasta periytyvät MainWindow ja ClientLocationSelection sisältävät kaiken SIS Clientia varten kirjoitetun lähdekoodin. ConcatStringExtension on valmis luokka, jota hyödynnetään valitun tilan nimen liittämiseksi tietokannasta ohjelman pääikkunan otsikkokenttään. Lähdekoodissa se sijaitsee MainWindow-luokan määrittävän tiedoston perässä. ConcatString on puolestaan ConcatStringExtension-luokan sisällä määritetty luokka, joka huolehtii varsinaisesta merkkijonon käsittelystä.

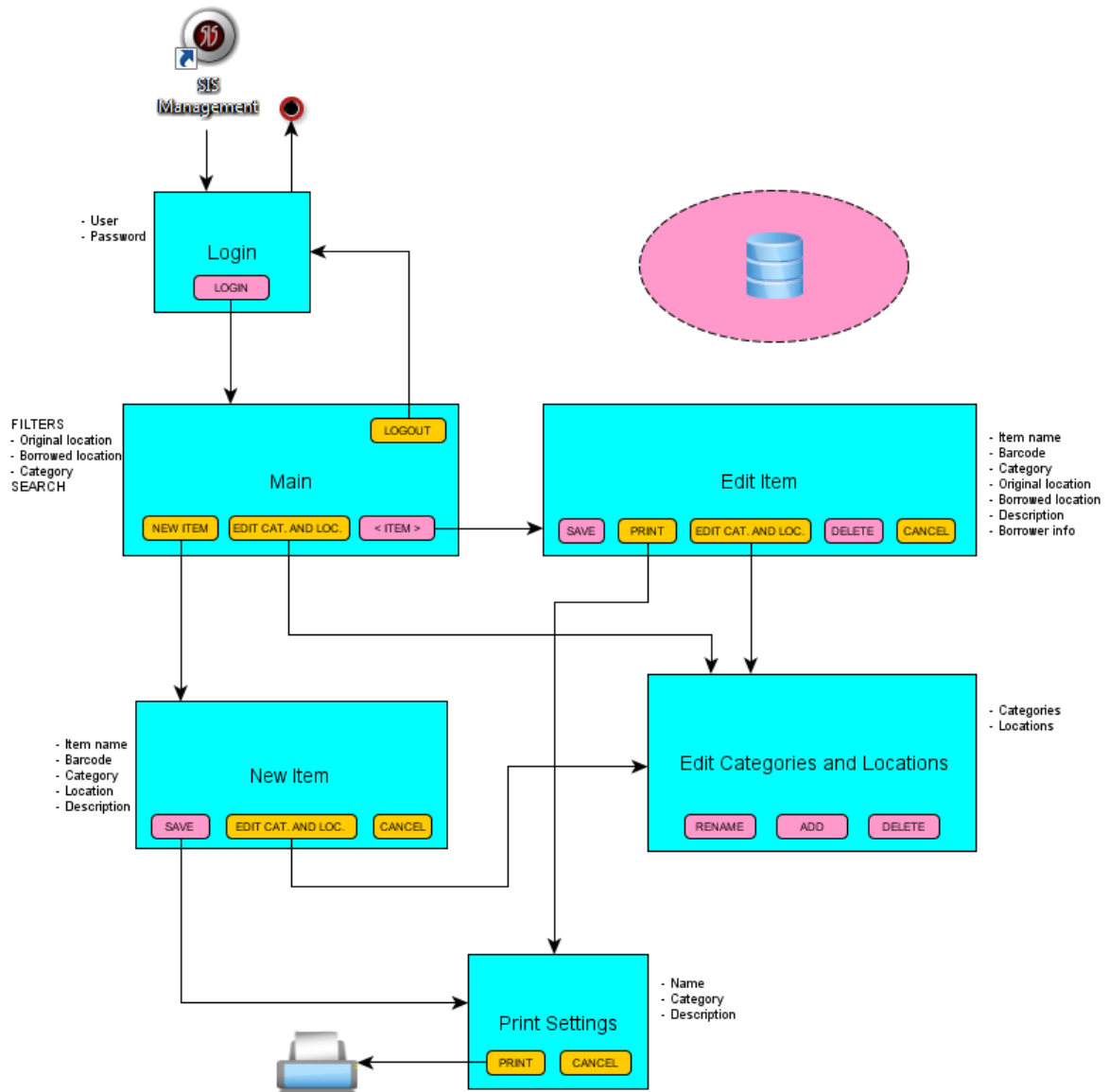


KUVA 4. SIS Clientin luokkahierarkia. Kuva on generoitu Doxygenilla.

2.2 SIS Management

SIS Management on SIS Clientiä laajempi ja monikäyttöisempi Microsoftin .NET-kehiksen (framework) päällä toimiva Windows-ohjelma. Kaikki Clientillä tehtävät toimenpiteet (lainaaminen, palauttaminen ja selaaminen) voidaan suorittaa myös Managementilla, mutta sen varsinainen käyttötarkoitus on tarjota tuotteiden hallintaan liittyvät työkalut. Clientistä poiketen Management käyttää perinteisempää Windows Forms -ohjelmointirajapintaa. Ohjelmointikielenä on käytössä C# ja ohjelmaa voidaan ajaa XP:ssä ja sitä uudemmissa Windows-käyttöjärjestelmissä. Periaatteessa myös vanhempiin käyttöjärjestelmiin ohjelma voidaan asentaa, mutta lisälaitteiden (viivakoodinlukija ja tarratulostin) rajoitusten kanssa saattaa ilmetä yhteensopivuusongelmia.

SIS Managementin avulla voidaan joustavasti hallita kaikkia tuotteisiin, tuotekategorioihin ja tiloihin liittyviä ominaisuuksia. Sen käyttöliittymä koostuu yhteensä kuudesta ikkunasta (kuva 5).



KUVA 5. SIS Managementin käyttöliittymän logiikka.

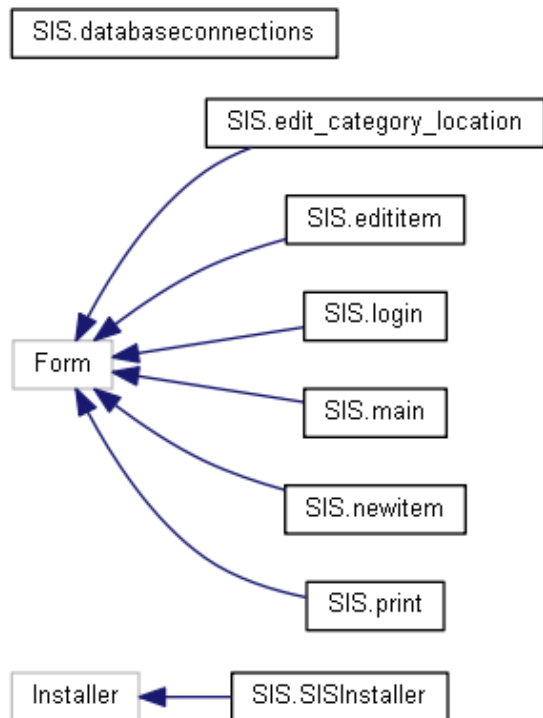
Kun SIS Management käynnistetään, avautuu ensimmäisenä kirjautumisikkuna (Login). Ohjelma pyytää syöttämään käyttäjätunnuksen ja salasanan, joita ilman ohjelman käyttö ei ole mahdollista. Managementin käytön rajoittaminen on sikäli perusteltua, että Clientin rajoitetuista toiminnoista poiketen Managementilla voidaan vaikka tyhjentää koko tietokannan sisältö. Kun käyttäjä on tunnistettu, avautuu ohjelman pääikkuna (Main), joka automaattisesti tulostaa näytölle kaikki järjestelmästä löytyvät tuotteet ja niiden tiedot. Tuotteita voidaan lajitella eri ominaisuuksien mukaisiin järjestyksiin ja/tai suodattaa sijaintitietojen, kategorian

ja vapaasanahaun mukaan näytölle. Logout-painikkeesta päästään takaisin kirjautumisikkunaan.

Järjestelmästä löytyvän tuotteen voi avata muokattavaksi joko kaksoisnapsauttamalla sitä pääikkunassa tai lukemalla sen viivakoodin, jolloin tuotteen tiedot avautuvat Edit Item -ikkunaan. Tuotteen kaikki tiedot ovat muokattavissa lukuun ottamatta tuotteelle luonnin yhteydessä generoitua viivakoodin lukuarvoa. Tuotteen tiedot voidaan tallentaa, tulostaa tai poistaa kokonaan. Jos tiedot halutaan tulostaa, avautuu erillinen Print Settings -ikkuna, josta voidaan erikseen määrittää, mitkä tiedot tarralle tulostetaan. Kun halutaan lisätä järjestelmään uusi tuote, painetaan pääikkunasta löytyvää New Item -painiketta, joka aukaisee samannimisen ikkunan. Kun kaikki tarvittavat tiedot on täytetty, voidaan tuote tallentaa ja ohjelma ehdottaa samalla myös tarran tulostamista.

Tuotekategorioiden ja tilojen muokkaaminen onnistuu omasta Edit Categories and Locations -ikkunasta, joka avautuu samannimisestä painikkeesta, joka puolestaan löytyy lähes jokaisesta edellä kuvatussa ikkunasta. Kategorioita ja tiloja voidaan luoda, nimetä uudelleen tai poistaa.

SIS Managementin luokkahierarkia koostuu pääasiassa Windows Formsin Form-luokasta periytyvistä kuudesta ikkunasta, jotka määrittävät oman graafisen ulkoasunsa ja toimintansa (kuva 6). Poikkeuksena on luokka nimeltä databaseconnections, jonka kautta kaikki tiedonvälitys ohjelman ja tietokannan välillä tapahtuu. Lisäksi ohjelmasta löytyy geneerinen asennusohjelman luomisesta huolehtiva luokka nimeltä SISInstaller.

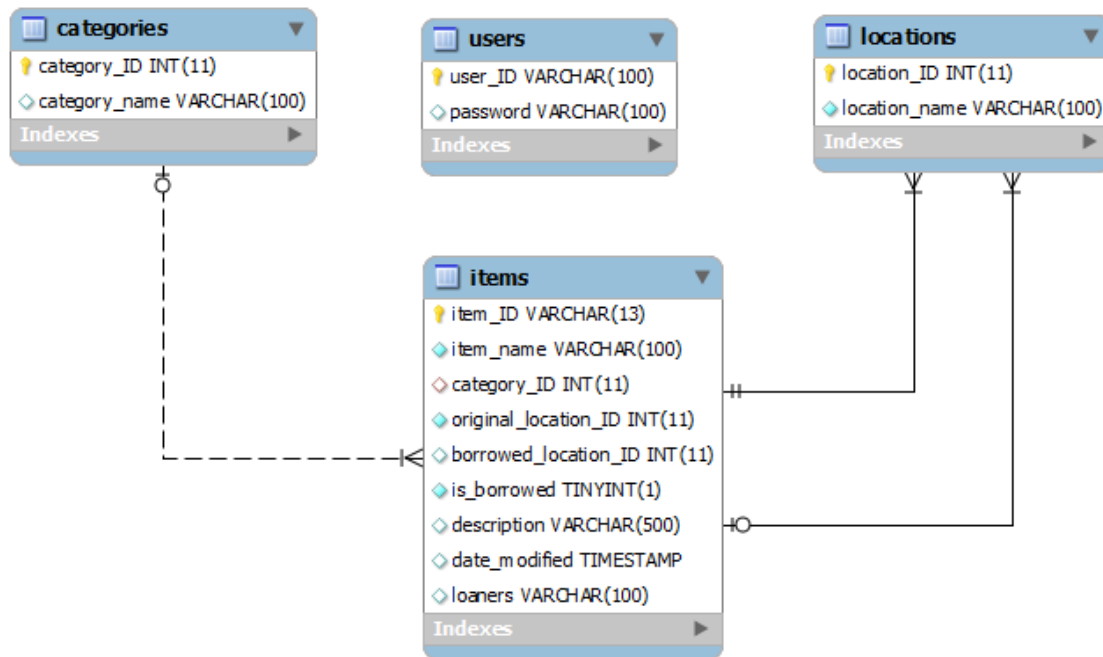


KUVA 6. SIS Managementin luokkahierarkia. Kuva on generoitu Doxygenilla.

2.3 Tietokanta

Scalable Inventory Systemin tietokanta on MySQL-relaatiotietokanta. Koska järjestelmän molemmat yksiköt pohjautuvat .NET-arkkitehtuuriin, on tietokannan ajuriksi valittu MySQL Connector/Net (3, linkit [ADO.NET Driver for MySQL \(Connector/NET\) -> Download](#)). Käytännössä se on .NET-kehiksen ohjelmistoprojektin viittaukseksi liitetty DLL-tiedosto, jonka avulla C#-lähdekoodiin voidaan sisällyttää MySQL-tietokannalle annettavia SQL-käskyjä.

Tietokanta suunniteltiin ja toteutettiin Scalable Inventory Systemin kehityksen yhteydessä, eikä siihen tullut muutoksia tämän opinnäytetyön aikana. Tietokanta koostuu neljästä taulusta (kuva 7).



KUVA 7. Scalable Inventory Systemin tietokannan taulut (4, s. 7).

Users-taulu sisältää hyväksytyt kirjautumistunnukset, joiden avulla SIS Managementin käyttäjä autentikoidaan. SIS Management vertaa käyttäjän syöttämää tunnus-salasana-yhdistelmää tämän taulun user_ID- ja password-kenttiin ja niiden ollessa identtiset avaa ohjelman pääikkunan.

Items-taulu on tietokannan päätaulu, johon tallennetaan kaikki järjestelmään luodut tuotteet ja niihin liittyvät tiedot. Näitä tietoja ovat tuotteen viivakoodi (item_ID), tuotteen nimi (item_name), tuotekategorian tunniste (category_ID), alkuperäinen sijainti (original_location_ID), lainatun tuotteen sijainti (borrowed_location_ID), lainatilanteen totuusarvo/lippu (is_borrowed), tuotteen kuvaus (description), muokauspäivämäärä (date_modified) ja lainaajan lisätiedot (loaners). Näistä viimeksi mainittua ei hyödynnetä järjestelmässä mitenkään, vaan se on lisätty tietokantaan tulevia projekteja varten. Siihen voi kyllä tallentaa merkkijonon muotoista tietoa SIS Managementin avulla, mutta se ei vaikuta järjestelmän toimintaan eikä näy SIS Clientin puolella.

Categories-taulu sisältää tuotekategorioiden nimet (category_name) ja niitä vastaavat tunnistenumerot (category_ID). Tämä taulu mahdollistaa kategorioiden

helpon lisäämisen, poistamisen ja uudelleennimeämisen. Jokaisella järjestelmään lisätyllä tuotteella on pakollisena tietona jonkin tuotekategorian tunnistenumero eli tuotteen täytyy kuulua aina johonkin kategoriaan. Items- ja categories-taulujen välillä on yhteys category_ID-kentän kautta, mikä mahdollistaa sen, että kategoriaa muokattaessa muutokset näkyvät myös kaikissa siihen kategoriaan kuuluvissa tuotteissa. Toisaalta yhteys myös estää poistamasta sellaista kategoriaa, johon yksi tai useampi tuote on luokiteltu.

Locations-taulu sisältää categories-taulun tapaan kaikkien tilojen nimet (location_name) ja niitä vastaavat tunnistenumerot (location_ID). Toiminta ja erillisen taulun suomat edut ovat samat kuin categories-taululla, mutta items- ja locations-taulujen välillä onkin kaksi yhteyttä: original_location_ID ja borrowed_location_ID. Jokaisella tuotteella täytyy olla alkuperäiseksi sijainniksi merkittynä jokin tila (original_location_ID), mutta lainatun tuotteen sijainti (borrowed_location_ID) sisältää jonkin tilan ainoastaan, jos tuote on todella lainassa. SIS pitää ohjelmallisesti huolta, etteivät nämä kentät sisällä samaa tilaa.

3 KÄYTETYT MENETELMÄT

Opinnäytetyön tekeminen vaati useiden ohjelmien ja laitteiden käyttöä. Osa liittyi kiinteästi itse järjestelmään ja osa toimi työkaluina projektin eri vaiheissa.

3.1 Ohjelmisto

Opinnäytetyössä keskeisimmät työmenetelmät liittyivät lähdekoodin tulkintaan, muokkaukseen ja dokumentointiin. Näiden toteuttaminen onnistui luontevasti Microsoftin kehitysympäristö Visual Studiolla, jolla myös järjestelmän asennusohjelmat luotiin. Varsinaisen lähdekoodin lisäksi asennusohjelmiin pakattiin päivitettyt ajurit tarratuloestimelle ja tietokannalle. Näiden ajureiden asentaminen järjestelmän asennuksen yhteydessä loppukäyttäjän tietokoneelle toteutetaan esilatausohjelmien (bootstrap) avulla. Esilatausohjelmaan voidaan määrittää asennuksessa käytettävät asetukset. Asennuksen yhteydessä voidaan esimerkiksi toteuttaa rekisteritarkistus jo mahdollisesti olemassa olevien ajureiden versiotiedon selvittämiseksi ja päättää sen perusteella, onko ajurin asentaminen tarpeellista. Esilatausohjelmien luomiseen käytettiin ohjelmaa nimeltä Bootstrapper Manifest Generator.

Lähdekoodin dokumentoinnin jälkeen lähdekoodista generoitiin HTML-pohjainen dokumentaatio, jonka tärkeintä antia on sen kattava kuvaus kaikista järjestelmässä käytetyistä luokista. Tähän käytettiin dokumentointityökalua nimeltä Doxygen. Projektin edetessä laadittiin lukuisia muitakin dokumentteja, kuten järjestelmän englanninkielinen dokumentaatio (liite 3), testaussuunnitelma (liite 5), -pöytäkirjat ja -raportti (liite 6), muutama käyttöohje ja useita projektin hallintaan liittyviä dokumentteja. Näissä käytettiin pääasiassa Microsoftin Office 2007 -ohjelmiston Word- ja Excel-sovelluksia. Dokumentointiin sisältyi myös kevyttä, GIMP-ohjelmalla toteutettua kuvankäsittelyä sekä kaavioiden piirtämistä yEd Graph Editor -ohjelmalla. Lisäksi projektitiedostojen varmuuskopiointiin käytettiin sekä varmuuskopiointiohjelmaa Cobian Backup että versionhallintaoh-

jelmaa Git. Kaikkien työssä käytettyjen ohjelmien versiotiedot löytyvät taulukosta 1.

TAULUKKO 1. Opinnäytetyössä käytetyt ohjelmat.

<i>Ohjelma</i>	<i>Versio</i>
Microsoft Visual Studio	2010 Professional (10.0.40219.1) SP1
Bootstrapper Manifest Generator	2.00
Doxygen GUI	1.8.3.1
Microsoft Office Word & Excel	2007 (12.0.6683.5001) SP3 MSO
GIMP	2.6.11
yEd Graph Editor	3.10.2
Cobian Backup	11 Gravity (11.2.0.582)
Git Gui	0.17.GITGUI (Git v1.8.3.msysgit.0)

3.2 Laitteisto

Järjestelmää kehitettiin Windows 7 -pohjaisella PC-kokoonpanolla. Testauksessa käytettiin useita vastaavia työasemia sekä Acer ICONIA -tablettia. Käyttöön oton yhteydessä järjestelmän toimivuus todettiin vielä kosketusnäytöllä varustetulla työasemalla. Itse järjestelmään kuuluvia lisälaitteita olivat Brother P-touch -tarratulostin sekä Marson-merkkinen viivakoodinlukija. Tarkemmat mallitiedot käytetyistä laitteista löytyvät taulukosta 2.

TAULUKKO 2. Opinnäytetyössä käytetyt laitteet.

<i>Laite</i>	<i>Malli</i>
PC-työasemat	Useita erilaisia (Windows 7, 32/64bit)
Acer ICONIA Tab	W500 (Windows 7)
Elo Touch Monitor	n/a (19")
Brother P-touch	QL-500
Marson Barcode Scanner	MT7945

4 TOIMENPITEET

Työ käynnistyi alkuperäisen Scalable Inventory System -projektin tuottamaan sisältöön perehtymisellä. Se sisälsi dokumentaatiota järjestelmän kehittämisen vaiheista sekä yksiköiden lähdekoodit ja asennusohjelmat. Asennusohjelmat toimivat moitteetta ja molemmat yksiköt saatiin asennettua työasemalle. Vaikka itse yksiköt eivät kaikkien ominaisuuksiensa osalta toimineetkaan oikein, sai ohjelmien toiminnasta ja käyttötarkoituksesta paremman kuvan kuin pelkästään lähdekoodia lukemalla. Kun yleiskuva oli hahmottunut, siirryttiin ohjelmointiprojektien käynnistämiseen ja itse lähdekoodin tutkimiseen. Tässä vaiheessa oli selvää, että ennen käyttöönottestauksen järjestämistä oli lähdekoodiin tehtävä korjauksia, mutta ohjelmavirheiden määrä tai laatu ei ollut vielä tiedossa. Opinäytetyön vaiheet olivat kuitenkin selvät ja niitä muodostui yhteensä neljä: ohjelmavirheiden korjaus, käyttöönottestaus, käyttöönotto ja dokumentointi.

4.1 Ohjelmavirheiden korjaus

Vakavin ohjelmavirhe liittyi SIS Managementin tulostusrajapintaan, sillä sen käyttäminen aiheutti ohjelman kaatumisen eli esti käytännössä tulostamisen kokonaan. Käyttöön valittu Brotherin tarratulostin vaatii toimiakseen loppukäyttäjän koneelle asennettavan ajurin nimeltä b-PAC Client Component, joka on ladattavissa Brotherin verkkosivuilta. Se on ns. COM-komponentti (Component Object Model) eli valmiiksi käännetty itsenäinen sovellus, jota käytetään ainoastaan sen tarjoamien rajapintojen kautta. COM-komponentti voi olla alun perin kirjoitettu eri kielellä kuin sen palveluja hyödyntävä ohjelma. Brotherin tarjoaman COM-komponentin avulla voidaan suoraan lähdekoodista antaa yksinkertaisia tulostuskäskyjä, joilla merkkijonumuotoista tietoa siirretään sopivaan mallipohjatiedostoon liitettynä tulostimelle. COM-komponentti voidaan sisällyttää ohjelmaan joko omaksi DLL-tiedostokseen tai upottaa itse ajettavaan ohjelmaan.

Syy SIS Managementin kaatumiselle tätä ajuria käytettäessä tiivistyy seuraavaan koodiriviin, jossa luodaan tulostustoiminnon mahdollistava ilmentymä/olio:

```
bpac.DocumentClass doc = new DocumentClass();
```

Koodirivi on alun perin kopioitu Brotherin sovelluskehityspaketin (b-PAC SDK) mukana tulevasta lähdekoodiesimerkistä, jossa tulostustoimintoa on esitelty (5, linkit Full b-PAC SDK for developers -> 3.0.017 -> bsdkw30017eu.exe). Koodissa ei sinänsä ole mitään vikaa, mutta jos kehitysympäristönä on Visual Studio 2010 ja Brotherin b-PAC-ajurin määrittelemät tietotyypit upotetaan SIS Managementin ajettavaan ohjelmaan, ei rajapinta toimi enää oikein. Tämä johtuu siitä, että upotus tapahtuu VS 2010 tarjoaman uuden ominaisuuden Embed Interop Types avulla, eikä se tietoturvasyistä hyväksy luokkien upottamista ajettavaan ohjelmaan. Luokka voi metatiedon lisäksi sisältää ajettavaa ohjelmakoodia, kun taas rajapinta sisältää ainoastaan metatietoa. Ratkaisu on yksinkertaisesti poistaa luokkaan viittaava pääte *Class* ja kutsua suoraan rajapintaa *Document*, jossa luokka *DocumentClass* on muutenkin määritelty:

```
bpac.Document doc = new Document();
```

Ilmentymän toiminnallisuus ei muutu, vaan kaikki sen tarjoamat metodit ovat käytettävissä. Tämän yksinkertaisen korjauksen pitäisi toimia kaikissa tilanteissa, joissa rajapinta vain on määritelty oikein. (6, VS 2010 compiler error: Interop type XXX cannot be embedded. Use the applicable interface instead.)

SIS Managementin kohdalla tulostamista häiritsi myös muutama tulostusalueen määrittäviin mallipohjatiedostoihin jäänyt ohjelmavirhe. Ne olivat kuitenkin vain virheellisiä viittauksia muuttujiin ja tulostusalueen rajauksen ongelmia, jotka oli helppo paikantaa ja korjata.

SIS Client vaikutti aluksi toimivan sille määriteltyjen vaatimusten mukaisesti, eikä sitä saanut kaadettua millään syötteillä. Lainaus- ja palautustoiminnot toimivat päällisin puolin oikein. Käyttöliittymän ja lähdekoodin lähempi tarkastelu paljasti kuitenkin yhden kriittisen virheen. Jokaisella tuotteella on tietokannassa kaksi kenttää, jotka sisältävät tuotteen alkuperäisen tilan sekä tilan, jossa kyseinen tuote on kulloinkin lainassa. Jos tuote sijaitsee alkuperäisessä tilassaan eli

sitä ei ole lainattu mihinkään, on näistä kentistä jälkimmäinen tyhjä. Kun SIS Clientilla lainattiin tuote tilasta A tilaan B, tuote merkittiin normaalisti lainatuksi eli tilan B tunniste kopioitiin tuotteen tietoihin. Jos samaa lainausta yritettiin heti uudelleen, siirtyikin tilan B tunniste tuotteen alkuperäisen tilan kenttään. Käytännössä tämä siis tarkoitti, että käyttäjä vaihtoi tuotteen alkuperäistä tilaa, mikä ei pitäisi SIS Clientilla olla mahdollista. Mikäli ohjelma olisi toiminut oikein, se ei olisi tehnyt mitään, koska tuote oli jo valmiiksi lainassa.

SIS Clientin dokumentaatiota tutkimalla selvisi, että lainaamisen ja palauttamisen toteuttava Update-painike luotiin aivan projektin loppuvaiheessa (7, s. 1). Sitä ennen toiminnot oli toteutettu erillisillä Borrow- ja Return-painikkeilla. Todennäköisesti ohjelmavirhe johtui juuri tästä viime hetken muutoksesta, jota ei ehditty testata riittävän kattavasti. Virheen korjaaminen vaati muutaman lisärivin lähdekoodiin.

Toisen kriittisen virheen aiheutti SIS Clientin pääikkunan otsikkokenttä, jonka klikkaaminen sai aikaan ohjelman kaatumisen. Sen korjaaminen onnistui yksinkertaisesti estämällä otsikkokentän tekstin muokkaus käyttäjältä. SIS Client on suunniteltu alun perin käytettäväksi kosketusnäytöllä, jolloin hiiren kursoria ei yleensä näytöllä tarvita. Tästä syystä se oli myös oletuksena piilotettu ohjelmallisesti. Ohjelman toistuva testaaminen kosketusnäytöttömällä työasemalla aiheutti kuitenkin siinä määrin ongelmia, että lisäsin pikanäppäinyhdistelmän (CTRL+M), jolla kursorin saa tarvittaessa näkyviin heti ohjelman käynnistymisen jälkeen. Toinen näkyvämpi muutos on ohjelmaan lisätty Logout-painike, jolla oletuksena koko näytön tilaan avautuva pääohjelma sulkeutuu. Aiemmin ohjelmaa ei saanut suljettua kuin Windowsin pikanäppäinten avulla.

Muut muutokset liittyivät lähinnä yksiköiden yhdenmukaistamiseen. Clientin ja Managementin kehittivät eri kehittäjäryhmät, joten niiden välille jäi lukuisia visuaalisia ja osittain myös toiminnallisia eroja, joita tässä opinnäytetyössä yhdenmukaistettiin. SIS Clientin pääikkunan sulkeminen ei sulje enää koko ohjelmaa, vaan SIS Managementin tavoin aukaisee taustalle jääneen kirjautumisikkunan. SIS Clientin kirjautumisikkunasta ei enää pysty valitsemaan niin sanottua tyhjää valintaa. Molempien yksiköiden asennusohjelmien toiminta, visuaalinen ilme ja

asetukset vastaavat nyt toisiaan. Samoin tekevät myös monet yksiköiden käyttöliittymissä käytetyt termit.

4.2 Käyttöönottestaus

Scalable Inventory System oli korjausten jälkeen valmis käyttöönottestausta varten. Tiedossa oli, että järjestelmää oli testattu kehitysvaiheessa useaan otteeseen. Varsinaista testausdokumentaatiota ei ollut saatavilla, mutta pienimuotoisten omien testien jälkeen kävi selväksi, että yksiköt toimivat hyvin niin erillään kuin samaan aikaan käytettynäkin. Käyttöönottestauksella oli tarkoitus kaivaa esiin viimeisetkin normaaliin käyttöön liittyvät virheet ja luokitella ne vakavuusasteen mukaisesti vakaviin, merkittäviin, vähäisiin ja kosmeettisiin virheisiin. Mikäli merkittäviä tai vakavia virheitä löytyisi, ne korjattaisiin ennen käyttöönottoa.

Ennen varsinaista käyttöönottestausta suunniteltiin sen toteutus. Tuloksena syntyi kattava testaus suunnitelma (liite 5), jossa kuvattiin 14 kappaletta SIS Clientiä ja 40 kappaletta SIS Managementia käsittelevää testitapausta. Testitapauksilla testattiin yksiköiden kaikki normaalit toiminnot sekä lukuisia poikkeustilanteita. Testaus suunnitelma sisälsi myös kaiken tarvittavan tiedon testausjärjestelyistä. Itse testauksen tueksi luotiin myös testauspöytäkirja, jossa samat testitapaukset ohjeineen löytyivät järjestyksessä, mutta sisälsivät lisäksi tyhjät kentät toistojen, kommenttien ja kehitysehdotusten varalle.

Käyttöönottestaus suoritettiin kahden päivän aikana, jolloin kolme testaajaa kävi läpi kaikki SIS Clientin ja SIS Managementin testitapaukset. Testausympäristönä toimi kolme pöytätietokonetta sekä yksi Acer ICONIA Tab W500 -tabletti. Kaikissa laitteissa käyttöjärjestelmänä oli Windows 7. Tuloksena syntyi kolme täytettyä testauspöytäkirjaa, joista koottiin testausraportti (liite 6). Merkittäviä tai vakavia virheitä ei löytynyt, mutta yhteensä 54 testitapauksesta kahdeksan ei täyttänyt läpipääsykriteereitä (taulukko 3).

TAULUKKO 3. Käyttöönottestauksen tulosten yhteenveto.

		SIS Client	SIS Management
Testitapaukset yhteensä		14	40
Hylätyt testitapaukset		1	7
Virheen luokittelu	Vakava	-	-
	Merkittävä	-	-
	Vähäinen	1	6
	Kosmeettinen	-	1

Havaitut virheet eivät olleet erityisen vakavia, eivätkä ne vaikuta ohjelman normaaliin käyttöön. Lähinnä ne ilmenevät erikoistilanteissa, joissa käyttäjä yrittää tehdä jotain odottamatonta.

Yksi hylätyistä testitapauksista liittyi tuotteen tietojen tulostamiseen. Viivakoodi tulostui huomattavan kapealle alueelle, kun viivakoodin lisäksi tulostettavaksi valittiin tuotteen nimi ja kuvaus. Tämä saattaa vaikeuttaa viivakoodin lukemista. Virhe luokiteltiin kuitenkin vähäiseksi, koska se esiintyy todellisessa käytössä erittäin harvoin. Oletuksena tulostukseen valitaan kaikki tiedot eli viivakoodin ja tuotteen nimen ja kuvauksen lisäksi myös tuotekategoria. Vain erittäin harvinaisissa tapauksissa tuotekategoria jätettäisiin valitsematta. Vika oli kuitenkin helposti paikannettavissa kyseisiä tulostusasetuksia vastaavaan mallipohjatiedostoon, joten se korjattiin SIS Managementin versioon 1.3.2. Muiden virheiden korjaaminen jätettiin mahdollisten tulevien projektien harkittavaksi. Ne on sisällytetty kappaleesta 6 löytyviin jatkokehitysehdotuksiin. Järjestelmän todettiin olevan toimiva kokonaisuus ja valmiina käyttöönottoa varten.

4.3 Käyttöönotto

Scalable Inventory System oli määrä ottaa käyttöön aluksi muutamassa tekniikan yksikön tilassa, joissa säilytetään laajaa valikoimaa erilaisia tuotteita ja tuotekokonaisuuksia. Tämän opinnäytetyön osalta sovittiin, että käyttöönotto rajoittuu erään hanketilan tuotteiden inventointiin, tuotteiden lisäämiseen järjestelmään ja järjestelmän toimivuuden varmistamiseen. Kyseiseen hanketilaan

asennettiin 31.7.2013 kosketusnäyttöinen pääte, oheislaitteet sekä molemmat SIS-yksiköt. Tämä työasema on tavallaan pilottikäytössä eli siinä molempien yksiköiden toimintaa voidaan havainnollistaa. Seuraavana päivänä valtaosa tilan tuotteista inventoitiin, lisättiin SIS Managementin avulla tietokantaan ja tuostettiin kullekin tuotteelle omat viivakooditarrat. Tarrat liimattiin tuotteen/tuotekokonaisuuden sisältävien muovilaatikoiden kylkeen (kuva 8) ja muovilaatikot järjesteltiin niille varatuille hyllyille säilytykseen.



KUVA 8. SIS-järjestelmään lisätty tuote viivakoodilla varustetussa säilytyslaatikossa.

Tulevaisuudessa on tarkoitus laajentaa järjestelmän käyttöä mahdollisimman moneen yksikön tilaan, jolloin myös SIS Clientin merkitys korostuu. Ideaali tilanne olisi, että jokaisessa tilassa, jossa säilytetään lainattavaksi tarkoitettuja tuotteita, olisi SIS Client käytettävissä joko kannettavalla laitteella tai kiinteällä työasemalla. Se vaatii kuitenkin tilakohtaista suunnittelua ja todennäköisesti myös laiteinvestointeja. Järjestelmää markkinoidaan myös mahdollisesti jatkossa muiden yksiköiden käyttöön.

4.4 Dokumentointi

Dokumentoinnilla tarkoitetaan tässä Scalable Inventory Systemin rakenteeseen ja toimintaan liittyvää dokumentointia. Se voidaan luokitella kahteen osaan: tekniseen dokumentaatioon ja Doxygen-dokumentaatioon. Tekninen dokumentaatio sisältää mahdollisimman kattavasti muun muassa järjestelmän kuvauksen, käyttötarkoituksen, arkkitehtuurin, tekniset tiedot, käyttöliittymän rakenteen ja komponenttien kuvaukset. Se on periaatteessa järjestelmän käyttöopas, josta on apua uusien projektityöntekijöiden perehdyttämisessä järjestelmän kehittäjiksi ja ongelmatilanteiden ratkaisemisessa. Tekninen dokumentaatio on kirjoitettu englanniksi (liite 3).

Doxygen-dokumentaatio on Doxygen-dokumentointityökalun generoima tiivistelmä lähdekoodin rakenteesta. Lähdekoodit syötetään Doxygenille ja se muodostaa niiden pohjalta rakenteisen dokumentaation, jossa kaikki lähdekoodin tyypit ja muut elementit on luetteloitu, lajiteltu ja linkitetty toisiinsa ohjelmakoodin hierarkian mukaisesti. Jotta Doxygen-dokumentaatiosta saataisiin kaikki hyöty irti, sen kaikki tyypit ja elementit tulee dokumentoida perusteellisesti suoraan lähdekoodiin, jolloin niiden kuvaukset siirtyvät suoraan dokumentaatiossa oikeille paikoilleen. Doxygen osaa myös luoda lähdekoodin pohjalta erilaisia kaavioita ja kuvaajia, kuten luokkahierarkia- ja yhteistyökaavioita. Dokumentaation voi tallentaa joko HTML:nä, Latexina, RTF:nä, PostScriptina, PDF:nä, pakattuna HTML:nä tai Unixin man-sivuina. Doxygenin tukemia ohjelmointikieliä ovat tällä hetkellä: C, C++, C#, Java, Python, VHDL, PHP IDL, Fortran, TCL, Objective-C 2.0 ja osittain myös D. (8, Doxygen manual)

Dokumentointi kannattaa tehdä tarpeeksi kattavaksi, jotta siitä olisi käytännössä apua Doxygen-dokumentaation lukijalle. Lähdekoodin dokumentointi eroaa perinteisestä kommentoinnista lähinnä syntaksin ja käyttötarkoituksen osalta. Seuraavassa on esimerkki SIS Managementin versiosta 1.2, jossa on kommentoitu GetLocation-metodia C-sukuisista ohjelmointikielistä tutun //-merkinnän jälkeen:

```
public DataSet GetLocation(string location_name) //get location by location name
{
...
}
```

Seuraavassa on sama metodi tällä kertaa dokumentoituna SIS Managementin versiossa 1.3.2:

```
/// <summary>
/// Gets a location matching the location name string.</summary>
/// <remarks>
/// This method is used for checking if a location name already exists in database.
/// Notice that it just compares the names of locations, not IDs of any kind.</remarks>
/// <param name="location_name">Location name string value added by user.</param>
/// <returns>Dataset containing the possible location name found from database.</returns>
public DataSet GetLocation(string location_name)
{
    ...
}
```

Dokumentointiin liittyvät rivit merkitään joko ///-merkinnällä, jolloin koko kyseinen rivi merkinnän jälkeen tulkitaan dokumentoinniksi. Tarvittaessa voidaan myös käyttää monirivistä merkintää /** ... */, jolloin kaikki aloitusmerkin /** ja lopetusmerkin */ väliin jäävä tieto tulkitaan dokumentoinniksi riippumatta rivinvaihdosta. Dokumentointi sijoitetaan aina dokumentoitavan tyyppin edelle eli käytännössä sen yläpuolelle. Pelkkä dokumentoinnin merkintä ei vielä aiheuta mitään toimenpiteitä dokumentointityökalulle, mutta se kertoo sijaintinsa puolesta, mitä ollaan dokumentoimassa, ja ilmaisee samalla kääntäjälle, ettei kyseisiä rivejä tarvitse käsitellä. Itse dokumentointi voidaan toteuttaa ohjelmointikielestä riippuen monella tavalla. Koska molemmat SIS-yksiköt on ohjelmoitu C#-kielellä, oli loogista käyttää C#:lle kehitettyä dokumentointimenetelmää, joka hyödyntää XML-syntaksia. Yllämainitussa esimerkissä on käytetty neljää eri XML-tagia: summary on lyhyt kuvaus, remarks on tarkempaa kuvausta tai lisätietoja varten, param kuvaa metodin tietyn parametrin tarkoitusta ja returns kuvaa, minkälaisen arvon metodi palauttaa. C# sallii myös monien muiden ja jopa itse tehtyjen tagien käytön, mutta mikäli Doxygen ei niitä tunnista, se jättää ne huomioimatta. (9, s. 455–457)

Dokumentoinnilla on tarkoitus selittää ohjelmoijan määrittämiä tyyppejä, kuten luokkia, delegaatteja ja rajapintoja, sekä niiden toimintaa tarkentavia osia kuten muuttujia, tapahtumia, metodeita/funktioita ja olio-ohjelmoinnissa tyyppillisiä ominaisuuksia (property). Myös mitä tahansa muita elementtejä voidaan dokumentoida, mutta tällöin Doxygen ei kohdista kuvauksia mihinkään elementtiin vaan esittää ne omina riveinään esimerkiksi elementin sisältävän luokan toimintoina.

Liitteessä 2 on esimerkki metodin login_check kommentoidusta lähdekoodista ja sen pohjalta generoidusta Doxygen-dokumentaatiosta.

5 TULOKSET

Opinnäytetyön näkyvin tulos oli Scalable Inventory System -järjestelmän ottaminen viimein käyttöön. Järjestelmän vakavimmat viat ja puutteet korjattiin ja käyttöönottestaus suoritettiin hyväksytysti. Sekä SIS Client että SIS Management toimivat hyvin ja ovat asennettavissa mille tahansa PC-tietokoneelle, jonka käyttöjärjestelmänä on vähintään Windows XP ja jolta löytyy toimiva internet-yhteys. Tilaajalle toimitettiin uusimmat asennuspaketit SIS Clientista (v1.2.0) ja SIS Managementista (v1.3.2) ja ne sisältävät aiemmista versioista poiketen myös pikaohjeet asennukselle ja yleisimpien ongelmien ratkaisuun (readme.txt). Lisäksi lähdekoodien mukaan liitettiin versionhallinnan seuranta tulevaisuudessa helpottavat tekstitiedostot, joihin on merkitty tämän projektin aikana syntyneet muutokset.

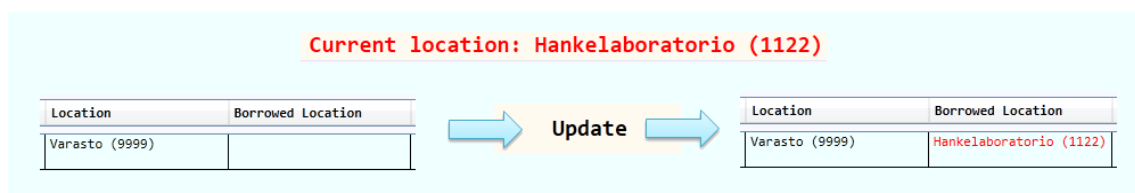
Toimivan järjestelmän lisäksi syntyi paljon sen käyttöä ja kehitystä tukevaa dokumentaatiota. Englanniksi kirjoitettu tekninen dokumentaatio (liite 3) auttaa tulevissa projekteissa järjestelmään perehtymisessä. Samoin lähdekoodi dokumentoitiin aiempaa perusteellisemmin myös englanniksi ja sen pohjalta laadittu Doxygen-dokumentaatio antaa kattavan kuvauksen lähdekoodin rakenteesta ja erityisesti sen luokkahierarkiasta. Doxygenin käyttöön ja C#-kielen dokumentointiin tutustuminen synnytti tilaajan toiveesta myös sivutuotteen: ohjedokumentin nimeltä C#-lähdekoodin kommentointi ja dokumentointi. Tämä ohje löytyy sekä suomeksi että englanniksi (liite 4) ja sitä tullaan hyödyntämään tulevaisissa projekteissa, joissa kirjoitetaan C#-koodia. Myös projektin hallintaan liittyvistä dokumenteista voi hyvin olla hyötyä järjestelmää jatkokehitettäessä – erityisesti testaussuunnitelmasta (liite 5) ja -raportista (liite 6).

6 JATKOKEHITYSTARPEET

Vaikka Scalable Inventory System on täysin toimintakuntoinen järjestelmä, voidaan sitä monin tavoin kehittää edelleen entistä tehokkaammaksi, virheettömämmäksi ja mukavammaksi käyttää. Käyttöönottotestaus paljasti muutamia vakavuudeltaan vähäiseksi luokiteltua ohjelmavirheitä, joita ei opinnäytetyön aikana ehditty korjata. Käyttöönoton yhteydessä huomattiin myös, ettei järjestelmä tarjoa aivan niin laajoja mahdollisuuksia tuotteiden käsittelyyn kuin kaivattaisiin. Lisäksi järjestelmän ja sen lähdekoodin kanssa työskentely synnytti muutamia ideoita, joiden avulla järjestelmän käyttö olisi entistä monipuolisempaa ja mielekkäämpää. Tässä kappaleessa esitellään kaikki nämä jatkokehitysideat, joita voidaan tulevissa kehitysprojeekteissa harkita ja tarpeen mukaan toteuttaa.

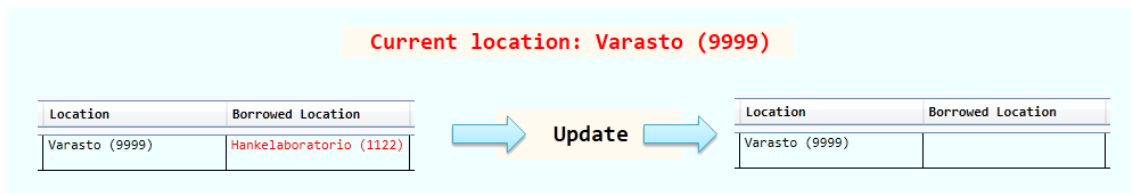
6.1 SIS Client – Palautetta lainauksesta ja palautuksesta

Nykyisessä versiossa lainaaminen ja palauttaminen tapahtuvat seuraavalla kaavalla: luetaan tuotteen viivakoodi ja painetaan Update-painiketta. Mikäli tilaksi on valittu jokin muu kuin tuotteen alkuperäinen tila, kyseessä on lainaus. Tällöin valittu tila merkitään tuotteen Borrowed Location -kenttään (kuva 9) ja tuote merkitään tietokannassa lainatuksi.



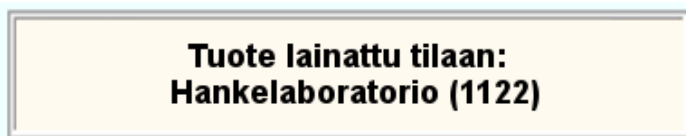
KUVA 9. Tuotteen lainaaminen SIS Clientilla.

Mikäli tilaksi on valittu tuotteen alkuperäinen tila, kyseessä on palautus. Borrowed Location -kenttä tyhjenee (kuva 10) ja tuote merkitään tietokannassa palautetuksi eli ei-lainatuksi.



KUVA 10. Tuotteen palauttaminen SIS Clientilla.

Ongelma liittyy käytettävyyteen ja tarkemmin Nielsenin heuristiikoista tuttuun riittävän palautteen antamiseen. ”Käyttäjän tulee saada palautetta tekemistään toiminnoista. Käyttäjän tulee saada tietää onnistuiko vai epäonnistuiko toiminta.” (10, Nielsenin säännöt – Riittävä palaute). SIS Clientiä aiemmin käyttänyt henkilö näkee varmasti heti, onko tuote lainassa vai ei, mutta uudet käyttäjät saattavat hyvin Update-painikkeen painamisen jälkeen ihmetellä, ettei mitään tapahtukaan. Yksinkertainen ratkaisu on lisätä ohjelmaan esimerkiksi oikeasta alakulmasta liukuen esiin nouseva ponnahdusikkuna, joka katoaa itsestään noin kahden sekunnin kuluttua (kuva 11).



KUVA 11. Esimerkki lainaustapahtuman aktivoimasta ponnahdusikkunasta.

Ponnahdusikkuna voisi luontevasti kertoa lainauksen ja palautuksen onnistumisen lisäksi myös epäonnistuneista tapahtumista. Esimerkiksi tilanteessa, jossa yritetään lainata tuotetta tilaan, jossa se on jo lainassa. Vaihtoehtoisesti ponnahdusikkunan voisi korvata vaikka pelkällä punaisella, suurehkolla fontilla kirjoitetulla tekstillä, joka käy hetken näkyvissä keskellä ikkunaa. Lainaus- ja palautustilanteissa suurin osa ikkunan pinta-alasta kun on muutenkin hyödyntämättä, kun näkyvissä on vain kyseisen tuotteen tiedot.

6.2 SIS Client – Tietojen lajittelu

Kun halutaan nähdä kaikki tilan sisältämät tuotteet, ne saadaan näkyviin List-painikkeella. Tuotteet lajitellaan automaattisesti viivakoodin numeroarvon perusteella. Kyseessä on vain staattinen lista, jonka järjestystä ei voi muuttaa. Tietokantaan lisättyjen tuotteiden määrän kasvaessa lajittelun mahdollistaminen on kuitenkin perusteltua. Yksinkertainen ratkaisu olisi mahdollistaa tuotteiden lajittelu SIS Managementin tyyliin sarakkeiden otsikkokenttiä (kuva 12) painamalla.

Current location: Hankelab(1122)						
Item ID	Item Name	Category	Location	Borrowed Location	Date Modified	Description
100000000061	Vuzix Wrap Video Eyewear (HMD1)	3D Framework	Hankelab(1122)		01.08.2013 11:18	1x Vuzix Wrap Video Ey 1x Vuzix Wrap 920 1x Vuzix Wrap Tracker 1x Vuzix Wrap VGA Adap 1x Vuzix AA Controller 1x Vuzix iWear Composi 1x Vuzix RCA Cable
100000000085	Vuzix Wrap Video Eyewear(HMD8)	3D Framework	Hankelab(1122)		01.08.2013 11:12	1x Vuzix Wrap 920 1x Vuzix AA Controller 1x Vuzix VGA Controlle 1x Vuzix Wrap Video Ey 1x Vuzix iWear Composi 1x Vuzix RCA Cable

KUVA 12. SIS Clientin sarakkeiden otsikkokentät.

Tällöin esimerkiksi Item name -kenttää painamalla ohjelma lajittelee tuotteet aakkosjärjestykseen tuotteen nimen perusteella. Toinen painallus lajittelee tuotteet käänteiseen aakkosjärjestykseen. Valitun sarakkeen otsikon viereen kannattaa myös laittaa pieni nuolen kuva kuvaamaan lajitteluperustetta ja aakkosjärjestyksen suuntaa.

6.3 SIS Client – Kosketusnäyttöominaisuuksien hyödyntäminen

Ensisijaisesti kosketusnäytölliselle laitteelle suunniteltu SIS Client toimii selkeästi parhaiten, kun sitä ohjataan hiiren sijasta kosketusnäytöllä. Itse kosketusnäyttöominaisuuksia ei kuitenkaan ole ohjelmassa hyödynnetty. Ohjelman yksinkertaisuuden vuoksi kaikkien kosketusnäyttöeleiden hyödyntäminen on liioiteltua, mutta esimerkiksi listan sulava vierittäminen sormen pyyhkäisyllä toisi

sen käyttöön heti aivan erilaisen tuntuman. Sulavalla vierityksellä tarkoitetaan kosketusnäyttökäyttöliittymissä tyypillistä liikettä, jossa vieritys jatkuu vielä hetken kosketuksen päättymisen jälkeen hitaasti hidastuen. Tämän toiminnallisuuden toteuttava esimerkkikoodi itse asiassa löytyy SIS Clientin lähdekoodista, mutta sitä ei ole liitetty mihinkään käyttöliittymän elementtiin, eikä se siis toimi.

6.4 SIS Client – Lainaajan tietojen hyödyntäminen

Scalable Inventory System toimii sillä periaatteella, että jokaiselle tuotteelle on määritetty tila, jossa sitä säilytetään, ja tuotetta voidaan lainata mihin tahansa muuhun tilaan. Järjestelmän suunnitteluvaiheessa syntyi visio, että tuotteen lainaajat tunnistettaisiin esimerkiksi henkilökortin avulla ennen lainaustapahtumaa, mutta se jäädytettiin kustannussyistä. SIS Management sisältää kuitenkin mahdollisuuden lisätä vapaamuotoista tietoa lainaajasta, mikäli kyseinen tuote on merkitty lainatuksi. Tätä tietoa ei hyödynnetä SIS Clientissä lainkaan.

Lainaajatiedon noutaminen tietokannasta ja esittäminen omalla sarakkeellaan on helppo toteuttaa. Se yksistään ei kuitenkaan tuo mitään lisäarvoa, mikäli lainaus- ja palautustapahtumiin käytetään pääasiassa SIS Clientia, jolla lainaajatietoja ei voi lisätä. Yksi tapa hyödyntää niitä on lisätä kirjautumisikkunaan kenttä, johon vaaditaan käyttäjältä vähintään nimen kirjoittamista. Joissain tapauksissa myös yhteystietojen lisääminen on järkevää. Kun käyttäjä omien tietojen ja tilan valinnan jälkeen pääsee lainaamaan tuotetta SIS Clientin pääikkunaan, näkyy lainaajan nimi siinä esimerkiksi tilan nimen yhteydessä. Kun käyttäjä lainaa tuotteen, siirtyy tämän nimi myös tuotteen tietoihin ja on nähtävillä omassa sarakkeessaan.

Tämä ratkaisu ei varmasti ole kaikissa tilanteissa ihanteellinen eikä varmista oikeiden tietojen antamista, mutta toimisi hyvin välivaiheen ratkaisuna, mikäli käyttäjän tunnistamista jossain vaiheessa kehitettäisiinkin pidemmälle. Käyttönoton jälkeen tuli myös ilmi, että joissain tilanteissa voi olla tarpeellista antaa tuote lainaan suoraan henkilölle ilman, että sitä merkitään lainatuksi mihinkään tilaan. Nykyisessä versiossa tällainen ongelma on kierretty lisäämällä lainaajan

nimi SIS Managementilla yhdeksi tiloista, mikä ei ole tarkoituksenmukaista. Mikäli lainaajatietojen lisääminen tehdään SIS Clientissa mahdolliseksi, täytyy tällaisten tilanteiden varalta varmistaa, että vähintään joko tila tai lainaaja merkitään tuotteen tietoihin aina lainauksen yhteydessä.

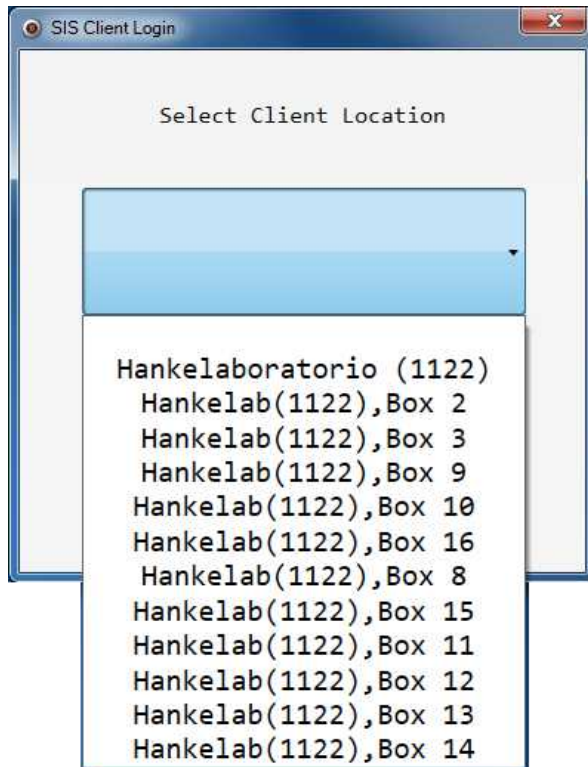
6.5 SIS Client – Väärän viivakoodin lukeminen

Scalable Inventory System käyttää tuotteiden yksilöintiin Suomessakin yleisintä viivakoodistandardia EAN-13, joka koostuu 13 numerosta. SIS Client lukee oikeassa formaatissa olevat viivakoodit oikein, mutta ei osaa käsitellä muissa formaateissa olevia viivakoodeja. Formaateista riippuen ohjelma saattaa tunnistaa, ettei kyseessä ole käyttökelpoinen viivakoodi, mutta sen lukeminen voi sekoittaa seuraavien viivakoodien lukemista. Jossain tilanteissa väärästä viivakoodista jää jokin osa ohjelman muistiin, mikä aiheuttaa virheen myös luettaessa seuraavia oikeassa formaatissa olevia viivakoodeja. Ohjelman uudelleenkäynnistys korjaa tämän ongelman, mutta virheenkäsittelyä voisi silti parantaa tältä osin.

6.6 SIS Management – Säilytyspaikat tilan sisällä

Käyttöönoton yhteydessä selvisi, ettei tuotteen sijainnin tietäminen tilan tarkkuudella välttämättä riitä. Tuote voi periaatteessa olla varastoituna vaikka suureen teollisuushalliin, mutta sen tarkempaa tietoa järjestelmä ei kykene tällä hetkellä antamaan. Tuotteen etsiminen ilman tarkkoja tilakohtaisia järjestelyitä olisi turhauttavaa. Ongelma on tällä hetkellä kierretty lisäämällä säilytyspaikka tilan nimen yhteyteen, jolloin ne yhdessä tulkitaan järjestelmän kannalta omiksi tiloikseen. Tämä aiheuttaa kuitenkin uuden ongelman varsinkin SIS Clientin kohdalla, koska kirjautumisvaiheessa täytyy myös valita tilan lisäksi, mihin säilytyspaikkaan haluaa ”kirjautua” (kuva 13). Tämä tekee mahdolliseksi myös lainaukset tilan sisällä esimerkiksi hyllystä 2 hyllyyn 5, mikä ei tietenkään ole toivottavaa. Toisekseen tilojen ja säilytyspaikkojen lisääntyessä tietokannan tilat

sisältävä locations-taulu paisuisi niin suureksi, että se haittaisi jo SIS-yksiköiden käytettävyyttä.



KUVA 13. Säilytyspaikkojen lisääminen tilan nimen perään aiheuttaa monia ongelmia erityisesti SIS Clientissa.

Kaikkein perusteellisin ratkaisu olisi luoda tietokantaan oma taulu, johon voisi lisätä kunkin tilan numeroidut säilytyspaikat, kuten kaapit, lokerot, hyllyt tai laatikot. Kun tuotetta lisättäisiin järjestelmään, sille merkittäisiin alkuperäisen tilan lisäksi tarvittaessa myös säilytyspaikka. Säilytyspaikat yhdistettäisiin tiloihin niin, että tilan valinnan jälkeen vain siihen liittyvät säilytyspaikat olisivat käytettävissä. Tuotteen tarraan säilytyspaikkaa ei tarvitsisi lisätä, koska säilytyspaikka voi tuotteen elinkaaren aikana helposti muuttua inventaarioiden ja tilan järjestelyiden seurauksena. SIS Clientin toimintaan tällä ei olisi muuta vaikutusta kuin yhden lisäsarakeen ilmestyminen taulukkoon. Lainaus- ja palautustapahtumiin tällä ei olisi vaikutusta, koska säilytyspaikka on vain tuotteen alkuperäisen tilan ominaisuus eikä siis liity lainauksen kohteena olevaan tilaan mitenkään. Tuotet-

ta palautettaessa näkisi suoraan SIS Clientista, mihin säilytyspaikkaan tuote on tarkoitettu jättää. Ratkaisu vaatii monia muutoksia tietokannan lisäksi myös molempiin SIS-yksiköihin. Kevyempi ratkaisu olisi sisällyttää SIS Managementiin mahdollisuus lisätä tarvittaessa vapaamuotoinen kuvaus säilytyspaikasta tuotteen luonnin yhteydessä. Tällöin säilytyspaikka olisi ainoastaan tuotteen ominaisuus, joka olisi toteutettavissa ilman uuden taulun luomista. Se ei kuitenkaan mahdollista säilytyspaikkojen keskitettyä hallintaa, vaan säilytyspaikaksi voisi valita mitä tahansa.

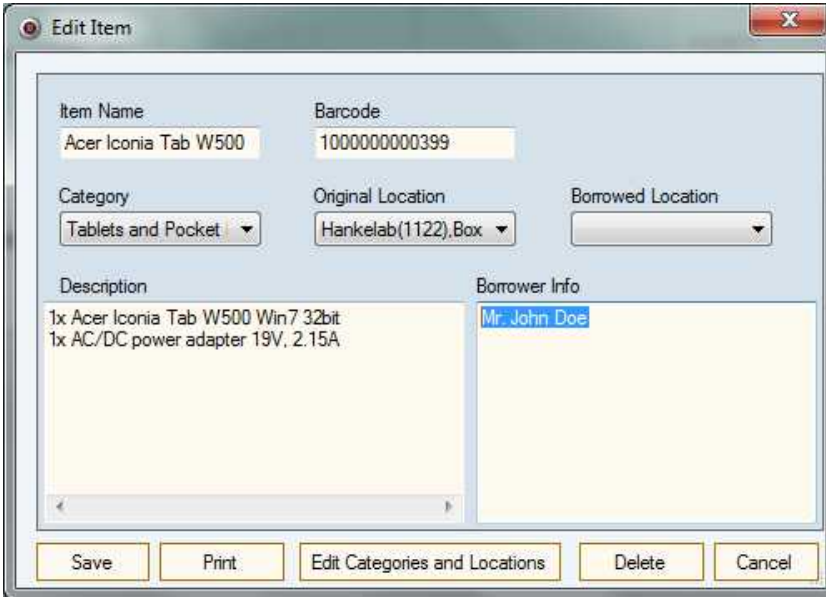
6.7 SIS Management – Olemattoman kategorian tai tilan poistaminen

SIS Managementilla voidaan muokata tuotekategorioita ja tiloja. Jos ne ovat käytössä yhdenkin tuotteen kohdalla, ohjelma estää niiden poistamisen. Ohjelma sisältää kuitenkin pienen ohjelmavirheen, joka liittyy sekä tuotekategorioiden että tilojen oletusvalintaan. Oletusvalintana on niin sanottu tyhjä muuttuja, joka on käytännössä yksi välilyönti " ". Kun SIS Management aukeaa pääikkunaan, tuotekategorioita ja tiloja seuloivat suodattimet viittaavat oletuksena tähän tyhjään muuttujaan, jolloin ohjelma näyttää kaikki tietokannan tuotteet. Kun tuotekategoriaksi valitaan suodattimesta jokin oikea tuotekategoria, ohjelma näyttää ainoastaan ne tuotteet, jotka kuuluvat tähän tuotekategoriaan.

Jos tyhjää muuttujaa " " yritetään poistaa, ohjelma antaa virheilmoituksen "Cannot delete this category!". Ohjelmavirhe ilmenee tilanteessa, jossa tyhjän muuttujan sisältämä yksi välilyönti pyyhitään pois ja painetaan sitten Delete-painiketta. Sama virhe ilmenee myös tilanteessa, jossa on ensin valittu jokin tietty tuotekategoria/tila, pyyhitty se pois ja painettu Delete-painiketta. Molemissa tapauksissa tyhjä muuttuja poistetaan tietokannasta, jolloin aakkosjärjestyksessä ensimmäinen tuotekategoria/tila siirtyy oletusvalinnaksi. Tämä aiheuttaa puolestaan sen, että kaikkia tuotteita ei voi mitenkään enää saada näkyviin yhtä aikaa, mikä luonnollisesti hankaloittaa ohjelman käyttöä. Virheen korjaaminen vaatii tarkistuksien lisäämistä lähdekoodiin.

6.8 SIS Management – Lainaajan tietojen tallentaminen

SIS Managementilla voidaan lisätä tietoa lainaajasta joko lainauksen yhteydessä tai tuotteen jo ollessa lainassa. Tiedot syötetään Borrower Info -kenttään ja ne tallentuvat tietokantaan Save-painikkeella. Jos tuote ei ole lainassa eli Borrowed Location -kenttä on tyhjä, lainaajatiedot eivät tallennu (kuva 14).



Item Name	Barcode	
Acer Iconia Tab W500	1000000000399	
Category	Original Location	Borrowed Location
Tablets and Pocket	Hankelab(1122),Box	
Description	Borrower Info	
1x Acer Iconia Tab W500 Win7 32bit 1x AC/DC power adapter 19V, 2.15A	Mr. John Doe	

Save Print Edit Categories and Locations Delete Cancel

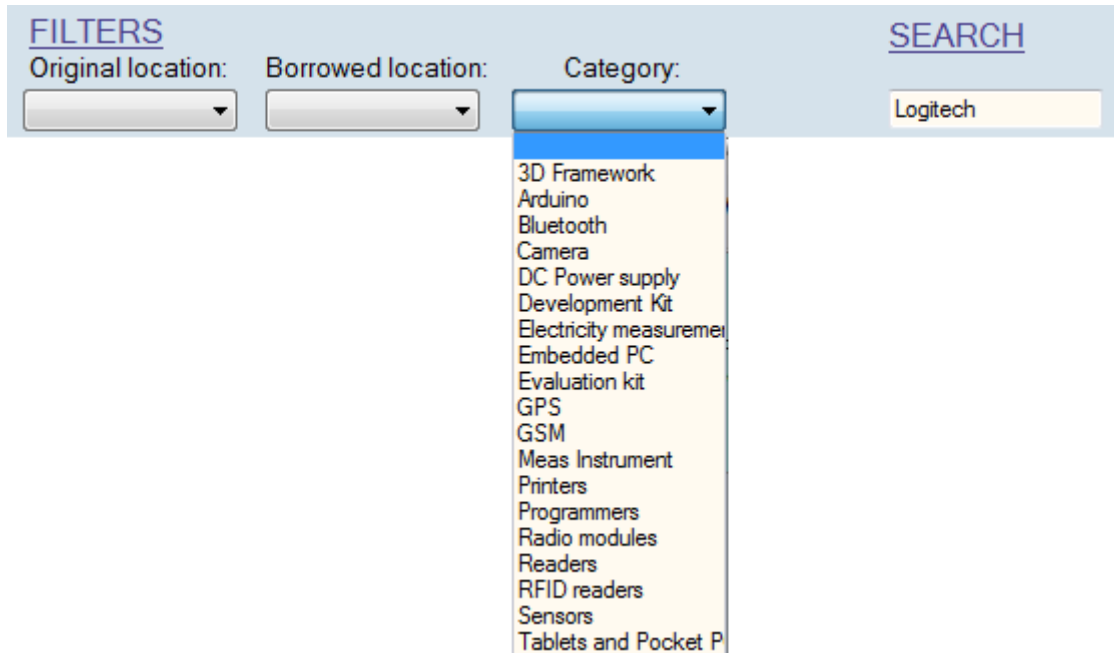
KUVA 14. Lainaajatiedot (Borrower Info) eivät tallennu Save-painikkeella, koska tuotetta ei ole merkitty lainatuksi (Borrowed Location).

Ohjelma toimii loogisesti täysin oikein, mutta jättää antamatta palautetta käyttäjälle poistaessaan käyttäjän syöttämää tietoa. Toimiva ratkaisu olisi avata uusi dialogi-ikkuna, jossa ohjelma kertoo aikovansa poistaa lainaajatiedot ja antaa käyttäjälle mahdollisuuden vielä peruuttaa tallennuksen.

6.9 SIS Management – Suodatinten ja haun laajentaminen

Suodattimilla voidaan suodattaa SIS Managementin näyttämiä tuotteita tilojen ja tuotekategorioiden mukaan. Haku-ominaisuudella voidaan suodattaa tuotteita käyttäjän syöttämän hakusanan mukaan. Suodattimet ja haku toimivat myös

yhtä aikaa, mutta ainoastaan, jos suodattimet valitaan ensin ja hakusana sen jälkeen. Toisin päin valittaessa suodatinten valinta ei vaikuta näytettäviin tuotteisiin mitenkään (kuva 15).



The image shows a search interface with a light blue header. On the left, under the heading 'FILTERS', there are three dropdown menus: 'Original location:', 'Borrowed location:', and 'Category:'. The 'Category:' dropdown is open, displaying a list of categories including '3D Framework', 'Arduino', 'Bluetooth', 'Camera', 'DC Power supply', 'Development Kit', 'Electricity measurement', 'Embedded PC', 'Evaluation kit', 'GPS', 'GSM', 'Meas Instrument', 'Printers', 'Programmings', 'Radio modules', 'Readers', 'RFID readers', 'Sensors', and 'Tablets and Pocket P'. On the right, under the heading 'SEARCH', there is a search input field containing the text 'Logitech'.

KUVA 15. Suodattimet (filters) eivät toimi, jos hakukenttään (search) on ensin syötetty hakusana.

Tuotteiden määrän pysyessä maltillisena suodattimia ja hakua käytetään todennäköisesti melko harvoin samaan aikaan. Jos tuotteiden määrä lisääntyy jossain vaiheessa tuhansiin, kannattaa ohjelmavirhe ehdottomasti jo korjata. Lisäksi haku kohdistuu tällä hetkellä ainoastaan tuotteen nimeen ja kuvaukseen. Sitä kannattaisi laajentaa vähintään lainaajatietoihin ja harkinnan mukaan myös muihin kenttiin.

6.10 SIS Management – Usean tuotteen muokkaaminen

SIS Managementin tuotteiden muokkausominaisuudet toimivat tällä hetkellä vain yhden tuotteen kohdalla kerrallaan. Tuote valitaan muokkaustilaan joko kaksoisnapsauttamalla sitä ohjelmassa tai lukemalla sen viivakoodi. Mahdollis-

suus usean tuotteen muokkaamiseen ja poistamiseen kerralla parantaisi ohjelman tehokkuutta. Muokattavat tuotteet voisi valita joko hiirellä aktivoimalla tuotetietojen vasemmalta puolelta löytyvät valintaruudut tai painamalla pikanäppäinyhdistelmää CTRL+A, joka valitsee kaikki näkyvillä olevat tuotteet. Suodatinten ja haun vaikutus huomioitaisiin myös eli niiden ollessa käytössä kaikkien tuotteiden valitseminen vaikuttaisi ainoastaan suodatuksen ja haun kautta valikoituihin tuotteisiin.

Muokkaustilassa kaikki kentät, joiden sisältö on valituilla tuotteilla identtinen, esitettäisiin normaalisti mustalla fontilla. Samoin kaikki tyhjät kentät esitettäisiin tyhjinä. Sellaiset kentät, jotka sisältävät eri informaatiota eri tuotteiden kohdalla, esitettäisiin harmaalla fontilla ja esimerkiksi tekstillä "misc." (miscellaneous, kuvaa monenlaista tietoa). Kun tällaista kenttää muokattaisiin, muuttuisi sen fontti normaalisti mustaksi ja tallennuksen jälkeen kaikki valitut tuotteet sisältäisivät saman tiedon. Tällaisen toiminnallisuuden toteuttaminen vaatii kuitenkin raskasta uudelleensuunnittelua ja paljon uutta koodia.

6.11 SIS Management – Viivakoodin lukemisen hallinta

Viivakoodin lukeminen SIS Managementin pääikkunassa tallentaa viivakoodin muuttujaan, hakee tietokannasta sitä vastaavan tuotteen ja avaa sen tiedot muokkausikkunaan. Käytännössä ennen tietokantahakua viivakoodi tallennetaan käyttäjältä piilossa olevaan tekstikenttään ja hyväksytään syöttämällä sen perään Enter-käsky. Ongelman aiheuttaa tilanne, jossa viivakoodi luetaan heti perään uudestaan tai viivakoodi luetaan jonkin muun ikkunan kuin pääikkunan ollessa aktiivisena. Tällöin viivakoodi käsitellään täysin samalla tavalla, mutta Enter-käsky ei avaakaan uutta muokkausikkunaa, vaan painaa aktiivisen muokkausikkunan oletuspainiketta eli Savea. Tämän jälkeen ohjelma kysyy, haluaako käyttäjä tallentaa kyseisen eli ensiksi luetun tuotteen tiedot.

Viivakoodin lukemisen hallintaa voisi kehittää niin, että muokkaustilassa uuden viivakoodin luku avaisi uuden dialogi-ikkunan, jossa käyttäjältä kysyttäisiin, ha-

luaako tämä vaihtaa valitun tuotteen. Muissa ikkunoissa pääikkunaa lukuun ottamatta viivakoodin lukemisen voisi estää kokonaan.

6.12 SIS Management – Väärän viivakoodin lukeminen

SIS Management generoi uusille tuotteille EAN-13-standardin mukaiset viivakoodit automaattisesti. Mikäli luettu viivakoodi on jotain muuta formaattia, ei ohjelma reagoi siihen mitenkään. Jos viivakoodi sisältää oikean määrän merkkejä – olivatpa ne numeroita tai kirjaimia – ja kyseistä viivakoodia ei löydy tietokannasta, ohjelma ehdottaa sellaisen lisäämistä avaamalla New Item -ikkunan. Väärässä formaatissa olevasta viivakoodista kannattaisi antaa virheilmoitus, jotta käyttäjä saa tapahtumasta palautetta. Samalla kannattaa myös pohtia, ovatko kirjaimia sisältävät, mutta oikean mittaiset viivakoodit hyväksyttäviä, kun ohjelma ei niitä itse kykene generoimaan.

6.13 SIS Management – Tilan ja tuotekategorian uudelleennimeäminen

Uuden tilan tai tuotekategorian lisääminen ei SIS Managementilla onnistu, jos samanniminen tila/tuotekategoria löytyy jo ennestään tietokannasta. Yritys kaa tuu virheilmoitukseen. Ongelman aiheuttaa kuitenkin tilan/tuotekategorian uudelleennimeäminen, sillä esimerkiksi tilan "Auditorio" voi muuttaa muotoon "Varasto", vaikka toinen "Varasto" löytyisikin jo tietokannasta. Tuloksena tietokannasta löytyy kaksi "Varasto"-nimistä tilaa. Ikävän ongelmasta tekee erityisesti se, että toiseen näistä tiloista kohdistuvat muutokset vaikuttavat samalla molempiin, koska ohjelmassa tiloja/tuotekategorioita ei jostain syystä käsitellä niiden ID-numeroiden vaan nimien perusteella. Vaikka virheen toteutumistodennäköisyys on erittäin pieni, sen korjaaminen on suositeltavaa, koska toteutusessaan virheen seurausten paikkaaminen voi olla työlästä.

6.14 SIS Management – Tulostustapahtuman nimen muuttaminen

Tuotteen tiedot sisältävän tarran tulostaminen SIS Managementilla tapahtuu erillisten .LBX-mallipohjatiedostojen avulla. Ne sisältävät kaiken tarvittavan tiedon tarran elementtien asettelusta ja tarvittavat rajapinnat muuttujien vastaanottamiseen itse ohjelmalta. Tiedostoja löytyy kahdeksan erilaista viivakoodin, tuotteen nimen, tuotekategorian, tuotteen kuvauksen sekä kaikkien kombinaatioiden varalle. Halutut tiedot käyttäjä valitsee juuri ennen tulostamista ja sen mukaan oikea mallipohjatiedosto tietoineen välitetään tulostusjonoon. Kuvassa 16 on nähtävillä tulostusjonoon edennyt mallipohjatiedosto, joka tulostaisi tarralle viivakoodin lisäksi tuotteen nimen ja kuvauksen.



KUVA 16. Mallipohjatiedosto tulostusjonossa.

Tämä ei varsinaisesti ole mikään ohjelmavirhe, mutta tulostusjonossa voisi tiedostonimen sijasta käyttää vaikka tulostuksen kohteena olevan tuotteen nimeä. Tämä olisi lähinnä kosmeettinen parannus, sillä tarratulostimen tulostusjonossa tuskin koskaan on todellista ruuhkaa.

6.15 Tietoturvakorjaus

Tietoturvan kannalta katsottuna molempiin SIS-yksiköihin kannattaisi tehdä pieniä korjauksia. Alkuperäisen Scalable Inventory System -projektin vaatimusmäärittelyn mukaan SIS Managementilla tulisi olla luku- ja kirjoitusoikeudet ja SIS Clientilla lukuoikeuden lisäksi rajoitettu kirjoitusoikeus tietokantaan (1, s. 3).

Lähdekoodin kannalta tilanne on hallinnassa, sillä SIS Clientilla ei voi kirjoittaa tietokantaan kuin lainaus- ja palautustapahtumiin liittyvää tilatietoa. Silti molemmat yksiköt käyttävät samoja tunnuksia ottaessaan yhteyttä tietokantaan eli SIS Clientilla on periaatteessa täysin perusteettomat kirjoitusoikeudet. Toiseen käyttäjän kirjautumistunnukset SIS Managementille kannattaisi vaihtaa paremmin oikeaa käyttöä vastaavaksi ja luovuttaa vain rajoitetun käyttäjäryhmän tietoon.

7 YHTEENVETO

Opinnäytetyön tavoitteena oli saattaa aiemmissa opiskelijaprojekteissa suunniteltu ja toteutettu Scalable Inventory System -järjestelmä toimintakuntoon ja ottaa se käyttöön Oamkin tekniikan yksikössä. Järjestelmä oli sikäli jo valmis kokonaisuus, että kaikki sen toiminnot oli toteutettu. Järjestelmän asentamisessa oli kuitenkin havaittu ongelmia, jotka estivät sen käyttöönoton. Ongelmat johtuivat puhtaasti ohjelmavirheistä, jotka tämän opinnäytetyön aikana paikannettiin ja korjattiin. Järjestelmälle tehtiin sen jälkeen kattava kolmen testihenkilön suorittama käyttöönototestaus, jonka tulokset osoittivat järjestelmän olevan toimintakunnossa ja valmiina käyttöön. Käyttöönotto järjestettiin asentamalla tarvittavat laitteistot ja ohjelmat tekniikan yksikön erääseen hanketilaan ja inventoimalla sen tuotteet järjestelmään. Järjestelmä on toiminnassa ja sen käyttöä laajennetaan tämän opinnäytetyön ulkopuolella. Lähdekoodit ja asennuspaketit luovutettiin tilaajalle.

Lähdekoodiin tehtyjen korjausten lisäksi se myös dokumentoitiin C#-kieltä varten suunnitellulla XML-syntaksilla. Koodin alkuperäistä kattavampi dokumentointi oli välttämätöntä, jotta sen pohjalta saatiin generoitua Doxygen-dokumentointityökalulla HTML-pohjainen lähdekoodin rakennetta kuvaava dokumentaatio. Tämän lisäksi koko SIS-järjestelmän käyttötarkoituksesta, arkkitehtuurista, käyttöliittymästä, komponenteista ja rajapinnoista laadittiin englanninkielinen tekninen dokumentaatio (liite 3). Molempien dokumentaatioiden tarkoituksena on helpottaa uusia ohjelmistokehittäjiä SIS-järjestelmää kehittämissä tulevilla projekteilla. Lähdekoodin dokumentoinnin yhteydessä syntyi lisäksi oheistuotteena ohjedokumentti C#-lähdekoodin kommentoinnista ja dokumentoinnista. Se laadittiin suomen lisäksi myös englanniksi (liite 4) ja luovutettiin tilaajan määrittelemään käyttöön.

Opinnäytetyön viimeisenä tavoitteena oli jatkokehitystarpeiden tunnistaminen. Lähdekoodia tutkiessa ja testatessa niitä syntyi helposti ja osa jopa toteutettiin ennen käyttöönototestausta. Suurin osa jäi kuitenkin toteuttamatta ajan puutteen vuoksi, ja niiden toteuttaminen jääkin tulevien projektien harkinnan varaan. Scalable Inventory System ei ole inventaariojärjestelmänä täydellinen, mutta se

on kokonaisuutena toimiva ja nykyisessä käytössään erittäin hyödyllinen verrattuna tilanteeseen, jossa tuotteita ei seurattaisi mitenkään tai seurattaisiin vaihtelevilla menetelmillä. Vaikka järjestelmästä löytyykin paljon kehityksen kohteita, kannattaa jatkossa ensisijaisesti keskittyä järjestelmän käytön laajentamiseen. Järjestelmän todellinen potentiaali saadaan käyttöön vasta sitten, kun se on asennettu valtaosaan tekniikan yksikön tuotteita säilyttäviä tiloja, jolloin lainaus- ja palautustoiminnot ovat aidosti hyödynnettävissä. Samalla myös tuotekannan seuraaminen ja tuotteiden etsiminen helpottuu huomattavasti. Mahdolliset parannukset ja uudet toiminnot voidaan myöhemmin asentaa päivityksinä SIS-yksiköiden vanhojen versioiden päälle. Tärkeää on myös varmistaa, että kaikki uudet hankinnat muistetaan lisätä järjestelmään, jotta siitä saatava tieto olisi ajantasaista ja luotettavaa. Vaikka Scalable Inventory Systemin tietokannan selaamiseen suunniteltua verkkosivua ei koskaan otettu käyttöön, kannattaisi sen toteuttamista harkita vielä uudelleen, sillä sen avulla erityisesti tuotteiden etsiminen olisi käytännöllisempää kuin salasanasuojatun SIS Managementin käyttäminen samaan tarkoitukseen.

Opinnäytetyön valmistuminen viivästyí huomattavasti alkuperäisestä tavoitteesta, joka oli kieltämättä epäsäännölliset osa-aikatyöt huomioon ottaen hieman epärealistinen. Edistystä tapahtui kuitenkin tasaisella tahdilla, eikä pitkiä taukoja päässyt syntymään. Suurin huolenaihe starttipalaverin jälkeen oli SIS Managementin tulostusrajapintaan liittyvä ongelma, jonka ratkaiseminen ei silloisilla tiedoilla ollut lainkaan itsestään selvää. Lähdekoodia tutkimalla ongelma kuitenkin ratkesi kohtuullisen nopeasti, eivätkä myöhemmin esiin nousseet ongelmat enää uhanneet opinnäytetyön tavoitteiden toteutumista. Ennakoitua suuremman osan aikaa vei lähdekoodin dokumentointi, mutta myös kaiken projektin aikana syntyneen dokumentaation määrä ja sen vaatima työmäärä ylittivät reippaasti ennako-odotukset. Kaikki lähtötietomuistiossa määritetyt tavoitteet saavutettiin, eikä kokonaistyöaikakaan lopulta ylittynyt kuin marginaalisesti.

LÄHTEET

1. Sallinen, Tommi 2012. Scalable Inventory System (SIS) - project description and system requirements v1.2. Oulu: Oamkin tekniikan yksikkö.
2. Working with App.xaml. 2013. The complete WPF tutorial. Saatavissa: <http://www.wpf-tutorial.com/wpf-application/working-with-app-xaml/>. Hakupäivä 31.10.2013.
3. Oracle Corporation. 2013. MySQL Connectors. Saatavissa: <http://www.mysql.com/products/connector/>. Hakupäivä 10.9.2013.
4. Ridell, Taneli – Parkkila, Jyri – Koski, Jani 2012. Scalable Inventory System, Management-yksikkö. Loppuraportti. Oulu: Oamkin tekniikan yksikkö.
5. Brother Industries, Ltd. 2013. Full b-PAC SDK for developers v3.0.017 (SDK, User Guide & sample programs). Saatavissa: <http://www.brother.com/product/dev/label/bpac/download/index.htm>. Hakupäivä 1.10.2013.
6. Shneerson, Misha 2009. VS 2010 compiler error: Interop type XXX cannot be embedded. Use the applicable interface instead. MSDN Blogs. Saatavissa: <http://blogs.msdn.com/b/mshneer/archive/2009/12/07/interop-type-xxx-cannot-be-embedded-use-the-applicable-interface-instead.aspx>. Hakupäivä 1.10.2013.
7. Matero, Henri – Parkkila, Jyri 2012. Scalable Inventory System, Client-yksikkö. Viikkoraportti (vko 19). Oulu: Oamkin tekniikan yksikkö.
8. Doxygen manual. 2013. Features. Saatavissa: <http://www.stack.nl/~dimitri/doxygen/manual/features.html>. Hakupäivä 15.10.2013.
9. Microsoft. 2013. C# Language Specification 5.0. Saatavissa: <http://www.microsoft.com/en-us/download/confirmation.aspx?id=7029>. Hakupäivä 31.10.2013.
10. Auer, Liisa 2006. Nielsenin säännöt. Opintojakson Johdatus käytettävyyteen materiaali. VirtuaaliAMK-portaali. Saatavissa:

<http://www2.amk.fi/digma.fi/www.amk.fi/opintojaksot/030308/1111676348138/1111677021119/1161290796532/1161290917294.html>. Hakupäivä 18.10.2013.

LIITTEET

- Liite 1 Lähtötietomuistio
- Liite 2 Esimerkki dokumentoidusta luokasta
- Liite 3 Tekninen dokumentaatio
- Liite 4 C#-lähdekoodin kommentointi ja dokumentointi
- Liite 5 Testaussuunnitelma
- Liite 6 Testausraportti



LÄHTÖTIETOMUISTIO

Työn tiedot	Tekijä ¹ Antti Vänskä	Tilaaaja ² Oulun seudun ammattikorkeakoulu / Tekniikan yksikkö
	Tilaaajan yhdyshenkilö ja yhteystiedot ³ Tommi Sallinen, _____, puh. _____	
	Työn nimi ⁴ Scalable Inventory System -järjestelmän testaus, dokumentointi ja käyttöönotto	
	Työn kuvaus ⁵ Olemassa olevan järjestelmän testaus eri laitteissa sekä pysyvä asennus eri koneille. Käyttöönotto ja sen todentaminen. Järjestelmän dokumentointi, Doxygen-luokka dokumentoinnin sekä jatkokehitystarpeiden tunnistaminen käyttöönottestauksen kautta (testihenkilöt). Testaus vaatii C#-tietämystä sekä lähdekoodiin perehtymistä.	
	Työn tavoitteet ⁶ - Käyttöönottestaus onnistuu ilman ongelmia - Järjestelmä on toimiva kokonaisuus - Erillinen dokumentointi mikä kattaa koko järjestelmän käytön (client ja management -yksiköt) - Järjestelmädokumentointi sekä luokkadokumentointi - Jatkokehitystarpeiden tunnistaminen	
	Tavoiteaikataulu ⁷ Opinnäytetyön arvioitu valmistumisaika on 31.5.2013.	
	Päiväys ja allekirjoitukset ⁸ _____/_____/_____ Tekijän allekirjoitus	
<ol style="list-style-type: none"> 1. Tekijän nimi, puhelinnumero ja sähköpostiosoite. 2. Työn teettävän yrityksen virallinen nimi. 3. Sen henkilön nimi ja yhteystiedot, joka yrityksessä valvoo työn suoritusta. 4. Työn nimi voi olla tässä vaiheessa työnimi, jota myöhemmin tarkennetaan. 5. Työ kuvataan lyhyesti. Siinä esitetään muun muassa työn tausta, lähtötilanne ja työssä ratkaistavat ongelmat. 6. Esitetään lyhyesti ja selvästi työn tavoitteet. 7. Esitetään projektin tavoiteaikataulu. Silloin, kun työllä on väitavoitteita, myös ne merkitään aikatauluun. Tavoiteaikataulun ja oppilaitoksen yleisaikataulun perusteella tekijä laatii oman aikataulunsa. 8. Lähtötietomuistio päivätään ja sen allekirjoittavat tekijä ja tilaaajan yhdyshenkilö 		

Lähdekoodissa:

```
/// <summary>
/// Login authentication method.</summary>
/// <remarks>
/// Checks that username and password are valid.</remarks>
/// <param name="user">Username entered by user.</param>
/// <param name="password">Password entered by user.</param>
/// <returns>A boolean value that defines if the user has privileges to access database.</returns>
public bool login_check(string user, string password)
{
    bool access = false;

    OpenConnection();

    /// <summary>
    /// Forms the SQL command and parameters.</summary>
    MySqlCommand mysqlcmd = new MySqlCommand();
    MySqlDataAdapter mysqladapt = new MySqlDataAdapter();
    mysqladapt.SelectCommand = new MySqlCommand("select * from users where user_ID=@user and
password=@password;", mysqlCon);
    mysqladapt.SelectCommand.Parameters.Add(new SqlParameter("@user", user));
    mysqladapt.SelectCommand.Parameters.Add(new SqlParameter("@password", password));
    dataset = new DataSet();

    try
    {
        /// <summary>Runs the SQL command.</summary>
        MySqlDataReader mysqlreader = mysqladapt.SelectCommand.ExecuteReader();

        if (mysqlreader.Read() != false)
        {
            if (mysqlreader.IsDBNull(0) == true)
            {
                /// <summary>If no results are found, access is denied.</summary>
                access = false;
            }
            else
            {
                /// <summary>If username and password match, access is granted.</summary>
                access = true;
            }
        }
        else
        {
            access = false;
        }

        mysqlreader.Close();
    }
    catch (Exception Er)
    {
        Console.WriteLine(Er);
    }

    CloseConnection();
    return access;
} // End of login_check()
```

Doxygen-dokumentaatioissa (HTML):

```
bool SIS.databaseconnections.login_check( string user,
                                         string password
                                         )
```

Login authentication method.

Checks that username and password are valid.

Parameters

user Username entered by user.
password Password entered by user.

Returns

A boolean value that defines if the user has privileges to access database.

Forms the SQL command and parameters.

Runs the SQL command.

If no results are found, access is denied.

If username and password match, access is granted.

Definition at line 46 of file databaseconnections.cs.

```
47     {
48
49         bool access = false;
50
51         OpenConnection();
52
53         MySqlCommand mysqlcmd = new MySqlCommand();
54         MySqlDataAdapter mysqladapt = new MySqlDataAdapter();
55         mysqladapt.SelectCommand = new MySqlCommand("select * from users where user_ID=@user and password=@password;", mysqlCon);
56         mysqladapt.SelectCommand.Parameters.Add(new MySqlParameter("@user", user));
57         mysqladapt.SelectCommand.Parameters.Add(new MySqlParameter("@password", password));
58         dataset = new DataSet();
59
60         try
61         {
62             MySqlDataReader mysqlreader = mysqladapt.SelectCommand.ExecuteReader();
63
64             if (mysqlreader.Read() != false)
65             {
66                 if (mysqlreader.IsDBNull(0) == true)
67                 {
68                     access = false;
69                 }
70                 else
71                 {
72                     access = true;
73                 }
74             }
75             else
76             {
77                 access = false;
78             }
79             mysqlreader.Close();
80         }
81         catch (Exception Ex)
82         {
83             Console.WriteLine(Ex);
84         }
85         CloseConnection();
86         return access;
87     } // End of login_check()
```

Scalable Inventory System (SIS)

Technical Document

Version 1.1

TABLE OF CONTENTS

1	PREFACE.....	2
1.1	Purpose of this document	2
1.2	Use of this document	2
1.3	Overview	2
2	INTRODUCTION.....	3
2.1	Purpose.....	3
2.2	Scope.....	3
2.3	References	4
3	SYSTEM OVERVIEW	5
3.1	System characteristics	6
3.2	System architecture	7
4	USER INTERFACE DESIGN	10
4.1	SIS Client UI.....	12
4.2	SIS Management UI.....	15
5	SYSTEM DESIGN.....	21
5.1	Class hierarchy	21
5.2	Database	22
5.3	Documentation standards	25
5.4	Software development tools	26
6	COMPONENT DESCRIPTION	27
6.1	MySQL Connector/Net.....	27
6.2	b-PAC Client Component	28
7	INSTALLATION INSTRUCTIONS	29
7.1	SIS Client Installation.....	29
7.2	SIS Management Installation.....	29
	DOCUMENT CONTROL.....	31
	DOCUMENT SIGNOFF	32
	DOCUMENT CHANGE RECORD.....	33

1 PREFACE

1.1 PURPOSE OF THIS DOCUMENT

This document is a complete technical overview of Scalable Inventory System. It describes handling, functionality and architecture of a system that was designed and developed by a group of students at Oulu University of Applied Sciences in spring 2012. The document provides guidance and reference material which is intended to assist the relevant management or designers in upcoming projects concerning Scalable Inventory System or other products as well.

1.2 USE OF THIS DOCUMENT

This document can be used to get a better understanding on what the Scalable Inventory System is, for what purpose it was developed, what does it do and how does it do it. This might help the end-users to solve some problems, but it's primarily intended to help developers to get wind of the system faster than just staring at the source code.

1.3 OVERVIEW

This document consists of the following sections:

Section 1 is the preface for this document.

Section 2 is the introduction and includes a description of the project and a list of reference documents.

Section 3 provides a system overview.

Section 4 demonstrates the user interfaces of the system.

Section 5 presents the system design, database structure and documentation standards.

Section 6 contains the component descriptions.

Section 7 provides the instructions for system installation.

2 INTRODUCTION

Scalable Inventory System was designed and developed at Oulu University of Applied Sciences (OUAS) in spring 2012. It consists of three units: Management, Client and Web page. The units were made by two groups of students. The first group (Taneli Ridell, Jani Koski and Juha-Pekka Pikki) made the Management unit and the other group (Henri Matero and Jyri Parkkila) made the Client and the Web page. However, the outcome of the Web page project didn't meet the original requirements and it will not be examined in this document.

2.1 PURPOSE

School of Engineering (OUAS) has a large amount of devices, modules, components and other equipment in numerous spaces around the campus. There isn't any formal protocol on how to keep track of different items. The problem is that no one might know where a certain item is located or to whom it's borrowed. Scalable Inventory System was made to meet this arisen need.

The idea is that every piece of equipment (or group of them) is stored in plastic containers that have unique labels with a barcode and additional information. Items have their original locations and they can be borrowed to other locations. With Scalable Inventory System they are always traceable, because every loan is stored in a database.

2.2 SCOPE

Both units, Management and Client, are used to handle the process of equipment loans. Client is meant to be installed in every location that has some SIS labeled equipment stored. Original idea is that Client is used on Windows tablet, but it can also be installed to any PC that runs at least Windows XP. However, the user interface is designed to work best with a touch screen.

When a person needs to borrow some item, he searches it and views its current location with SIS Client and then goes and gets it. After getting back to location in which he needs to use the item, he just reads the barcode on its container, updates the information and the item is then stored as borrowed to the current location. Returning the item works similarly: he takes the item back to its original location, reads the barcode, updates the information and it's done.

Management unit is used for managing all the items, item categories and locations. It can be installed to any PC that runs at least Windows XP. It can do the same tasks than Client, but the main reason for using SIS Management is adding new items and printing labels for them. Every basic admin functions (add, delete, rename, search, etc.) are included in SIS Management, so the problems that users confront with the Client can usually be solved by contacting an admin who has SIS Management privileges. Any relevant piece of information in database tables can be modified with SIS Management.

2.3 REFERENCES

The documents presented below provide miscellaneous information about the different phases of Scalable Inventory System development. All documents can be found from the project folders of SIS development: documents 7-8 from the SIS design and development folder (2012) and documents 1-6 from the testing, documentation and deployment folder (2013).

#	Title	Author(s)	Date	Language
1	Bachelor's thesis: Scalable Inventory System -järjestelmän testaus, dokumentointi ja käyttöönotto	Antti Vänskä	1.11.2013	Finnish
2	Using comments and document comments in C#	Antti Vänskä	20.9.2013	English / Finnish
3	SIS Management Documentation	Antti Vänskä	23.8.2013	English
4	SIS Client Documentation	Antti Vänskä	21.8.2013	English
5	Testausraportti (SIS)	Antti Vänskä	18.6.2013	Finnish
6	Testaussuunnitelma (SIS)	Antti Vänskä	29.5.2013	Finnish
7	Loppuraportti (SIS Client)	Henri Matero	29.5.2012	Finnish
8	Loppuraportti (SIS Management)	Taneli Ridell, Juha-Pekka Pikki, Jani Koski	24.5.2012	Finnish

3 SYSTEM OVERVIEW

SIS Client is based on Microsoft's .NET Framework technology called Windows Presentation Foundation (WPF). It's a modern alternative for older GDI based rendering systems (such as Windows Forms) since it utilizes DirectX. A significant difference between WPF and Windows Forms is that WPF provides a separation between the user interface and the business logic. This means that all the UI elements are defined and possibly linked with Extensible Application Markup Language (XAML). WPF works with all versions of Microsoft Windows since Windows Vista, but Windows XP (SP2/SP3) users can also install the necessary runtime libraries.

The main reason for using WPF in SIS Client was that it was supposed to work with touch screen and the developer came to conclusion that Windows Touch controls don't work with Windows Forms. That isn't actually true since there has been full multitouch functionality available for Windows Forms too since 2010. Still, it might have been more convenient to use WPF. SIS Management is instead based on Windows Forms. Both units are programmed in C#.

	SIS Client	SIS Management
Version	1.2.0	1.3.2
Localization	English	English
Platform	x86	x86
Framework	.NET Framework 4 Client	.NET Framework 4 Client
API	Windows Presentation Foundation (WPF)	Windows Forms (WinForms)
Languages	C#, XAML, SQL	C#, SQL
Prerequisites	.NET Framework 4 Client Profile Windows Installer 4.5	b-PAC Client Component 3.0.016 .NET Framework 4 Client Profile MySQL Connector/Net 6.6.5 Windows Installer 4.5

Prerequisites were added in the first versions of SIS Client and Management. They might not be required if a target PC is up to date. The only exceptions are b-PAC Client Component, which is essential for printing labels with printers manufactured by Brother, and MySQL Connector/NET that offers the interface for easy communication with MySQL database. More information about these components can be found from section 6.

3.1 SYSTEM CHARACTERISTICS

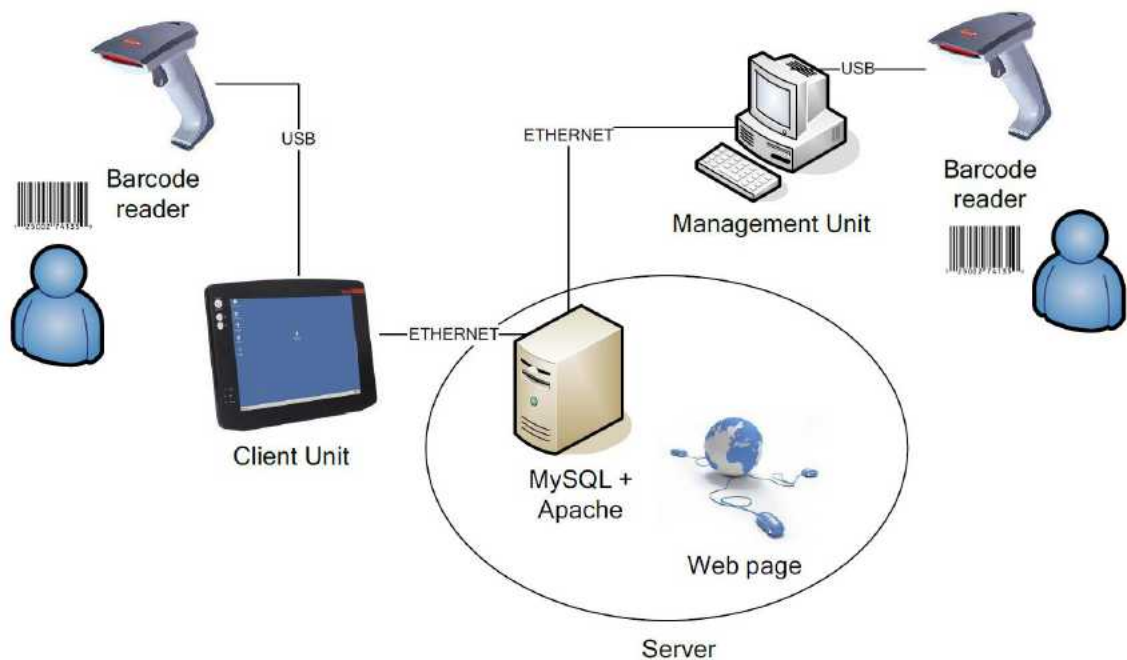
Scalable Inventory System is designed to be used in irregular intervals. This basically means the short events of borrowing or returning items with Client and adding new items or changing some database data with Management. Connections to database are opened only when needed so transactions (queries and updates) are done in bursts.

There might be multiple users at the same time, but Client users don't interfere with each other, because they are dealing with different items (and they have different barcodes). Collisions between Client and Management users are possible, but unlikely. The possibility that both users are making changes to the same item might end up one or the other losing his new data. Changes concerning locations or categories aren't shown in real-time in Client UI, but they won't cause any real trouble. The limit of concurrent users is a matter of MySQL server, not the Client/Management units (they are not linked to each other in any other ways than a shared database).

The system has some minor bugs, but they won't disturb the normal use. Known issues are described extensively in testing report (document 5) and SIS-related Bachelor's thesis (document 1) presented in section 2.3. System isn't very fault tolerant when it comes to network problems. Both units have issues with recovering from connections drops, because they don't have a proper exception handling for broken connection. It might show up as multiple unhandled error messages and might need application reboot after the connection comes back. It's not a major problem, because the system in offline mode is useless anyway. When online the system is quite resilient with incorrect inputs and actions.

3.2 SYSTEM ARCHITECTURE

Scalable Inventory System is based upon client-server architecture. The server is a MySQL server and both units (Client and Management) work as clients over internet (picture 1). Both units have their own USB barcode readers and SIS Management also has a USB label printer for adding new items to the system.



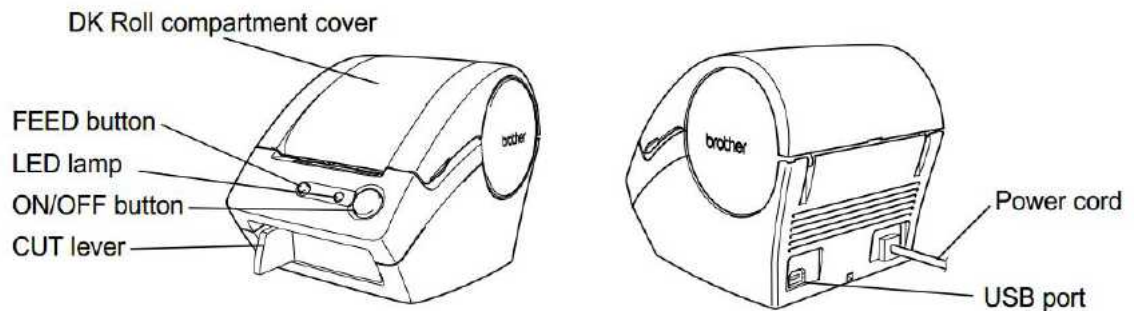
PICTURE 1. Architecture design for Scalable Inventory System.

The barcode reader used with SIS Client can basically be whatever reader that supports EAN-13 barcode standard. The model that is currently used with SIS is Marson MT7945 laser barcode reader (picture 2). It's a basic easy-to-use USB accessory that Windows automatically installs when plugged in.



PICTURE 2. Marson™ MT7945 Barcode Reader.

The label printer used with SIS Management is Brother P-Touch QL-500 (picture 3).



PICTURE 3. Brother™ P-Touch QL-500 Label Printer. (© Brother QL500/QL-560 Label Printer User's Guide)

There is a wide range of DK Rolls (Brother's label rolls) in different sizes and materials available for QL-500. SIS Management uses size 29mm x 90mm (1-1/7" x 3-1/2") die-cut (pre-sized) labels. There's an example of one test label printed with SIS Management in picture 4. All information was selected for printing: name, category and the barcode with its numeric value are on the left and the description is on the right. The sizes and locations of these elements may vary depending on what pieces of information is selected to be printed.



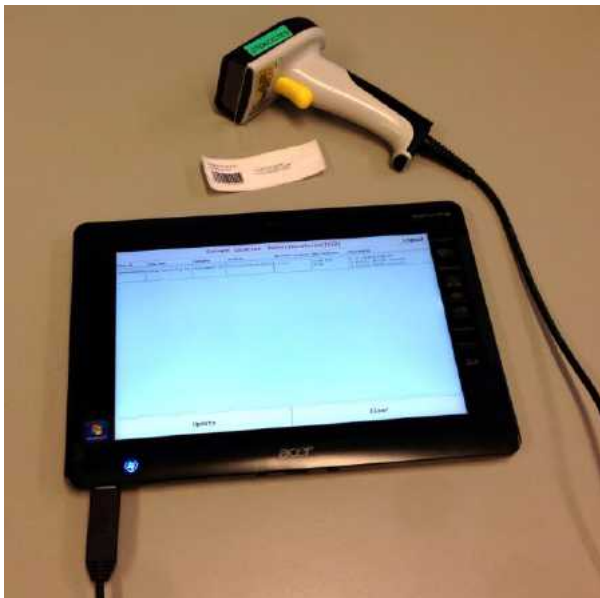
PICTURE 4. Label printed with SIS Management.

Before use QL-500 needs drivers installed (see section 7.2). Printing with user's program like SIS Management requires also the b-PAC Client Component installed (see section 6.2). Modifying label template files (*.lbr, *.lbl) can be done with P-touch

Editor, but it isn't required when using SIS Management. Picture 5 shows what SIS Management looks like when it's installed to a PC with label printer and barcode reader. Picture 6 shows the same with SIS Client on a tablet with barcode reader.



PICTURE 5. SIS Management with label printer and barcode reader installed.

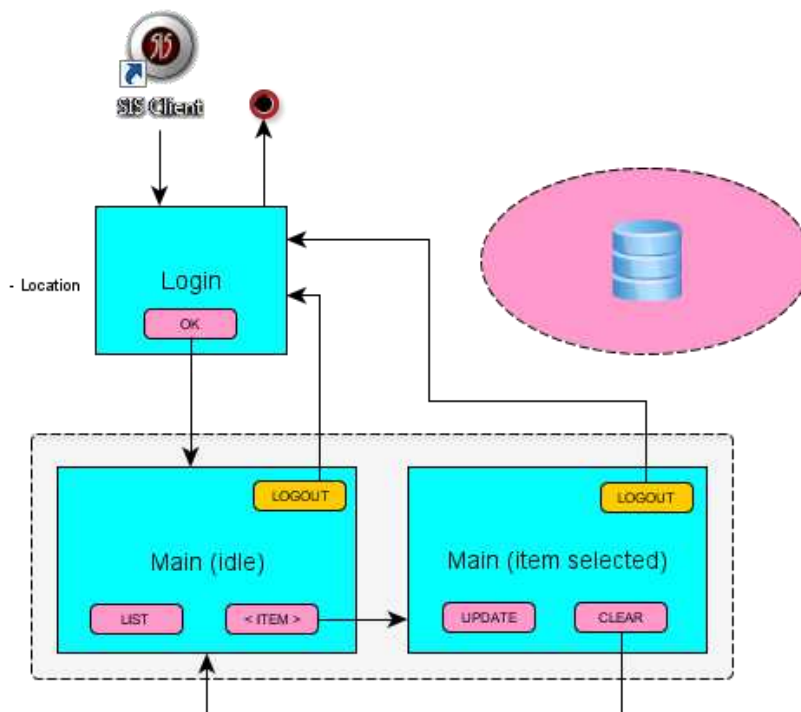


PICTURE 6. SIS Client with barcode reader installed.

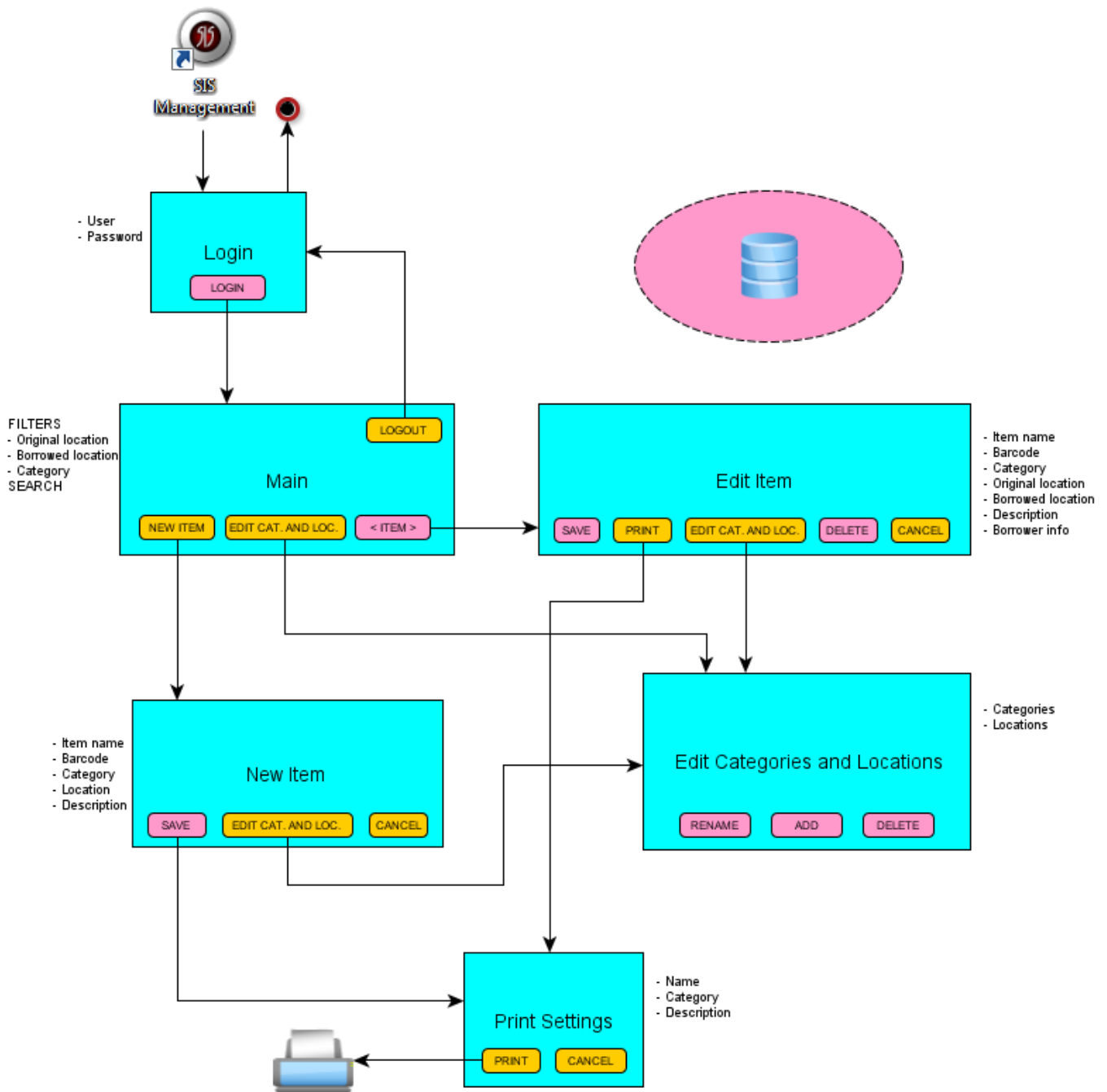
4 USER INTERFACE DESIGN

The user interfaces of SIS Client and SIS Management vary to some extent, mainly because of the different target platforms they were designed for and partly because of the different graphical application programming interfaces (APIs) and different designer teams.

The form logic of SIS Client (picture 7) and SIS Management (picture 8) is shown below. Blue rectangles present forms, yellow rounded rectangles present buttons and pink rounded rectangles present buttons that have any interaction with the database. The pink rectangle that's labelled < ITEM > isn't actually a button, but an element that can be selected with a barcode reader (works in both units) or double clicking the corresponding row in the list (only in Management).



PICTURE 7. Form logic of SIS Client.

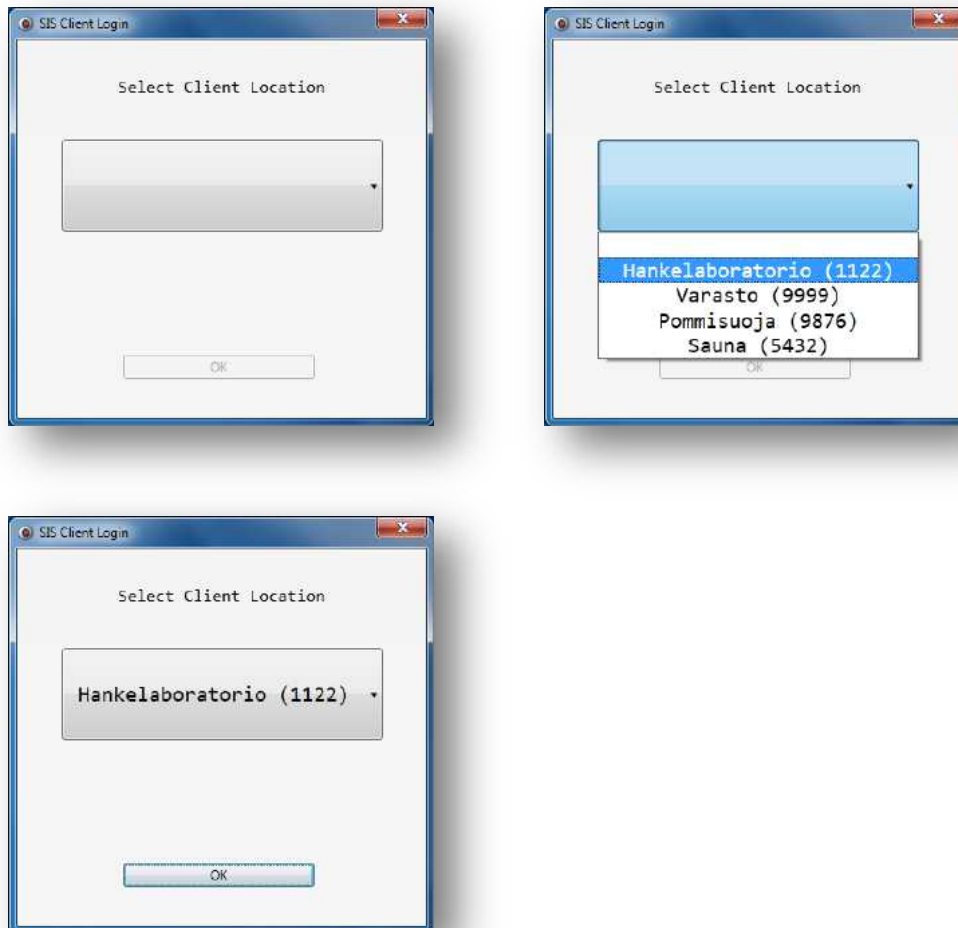


PICTURE 8. Form logic of SIS Management.

The next subsections demonstrate what the user interfaces of SIS Client and SIS Management actually look like.

4.1 SIS CLIENT UI

Client unit consists of only two forms: login and the actual program (main form). When SIS Client is run, the login form pops up and asks for user to select in which location he is currently using the client (picture 9). After selecting the location, OK button enables.



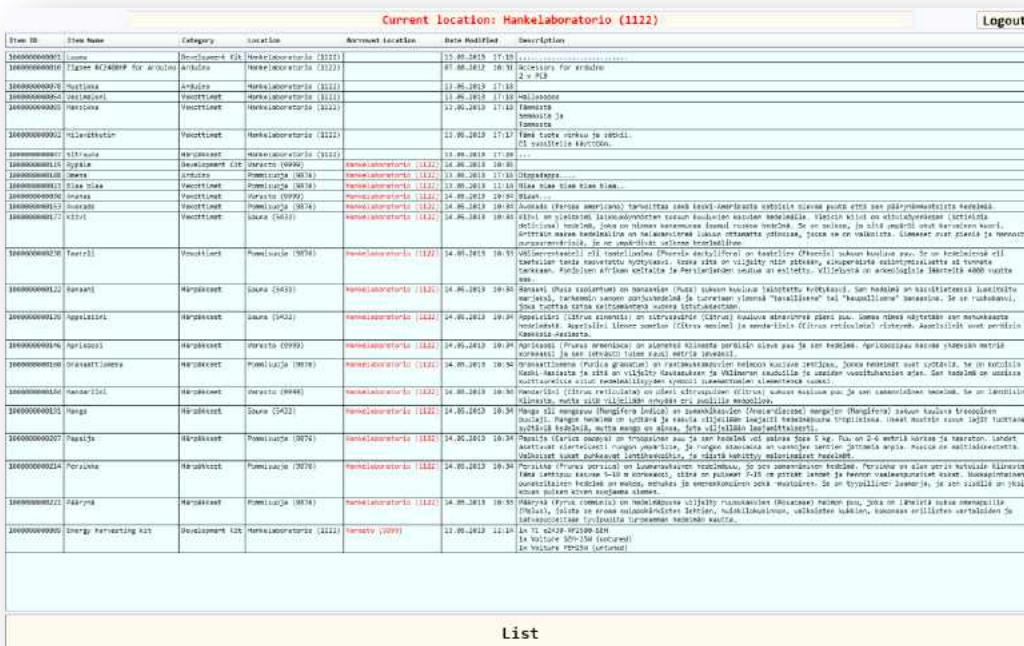
PICTURE 9. Login form of SIS Client.

When the user has confirmed the location, login form hides to the background and the main form fills the screen (picture 10).



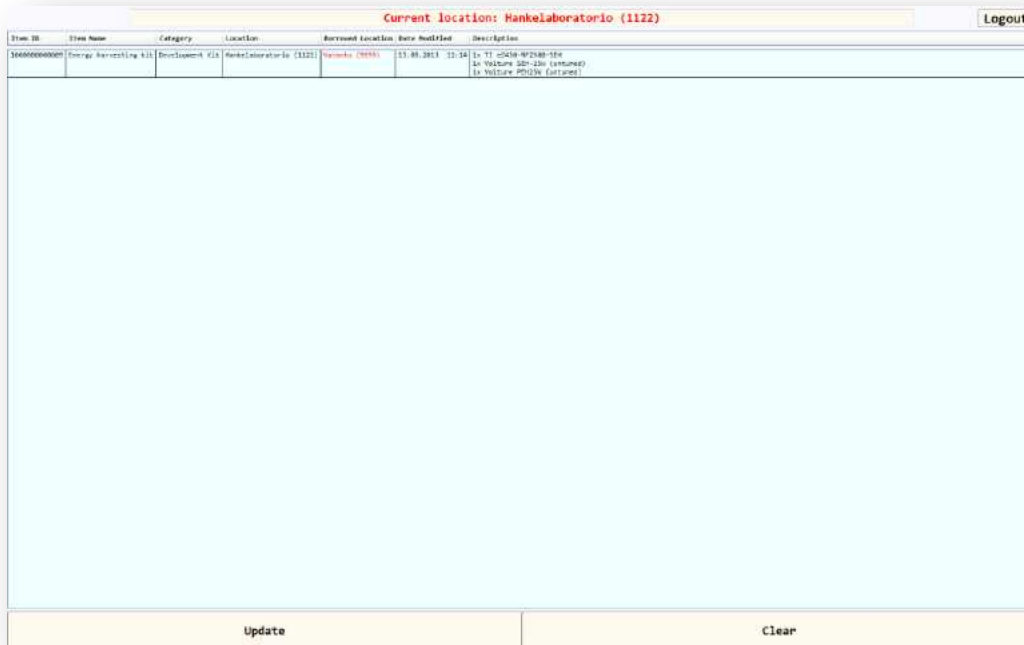
PICTURE 10. Main form of SIS Client.

The form shows the newly selected location on top of the screen. There's a grid in the centre, but it's currently empty. The user has now three choices: to hit the 'List' or 'Logout' buttons or use the barcode reader to read some existing barcode input. Hitting the 'List' button makes a query to the database and opens up all the items that either originate from the current location or are borrowed there (picture 11).



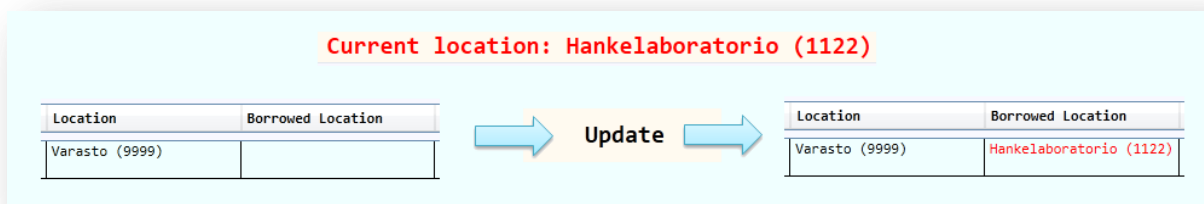
PICTURE 11. Main form of SIS Client after hitting the 'List' button.

The 'List' function is made only for review purposes for now. The list of items can only be viewed, not sorted in any way or even select single items. The only way to select an item is to read its barcode. After reading a barcode of some particular item, main form shows only that item (picture 12).



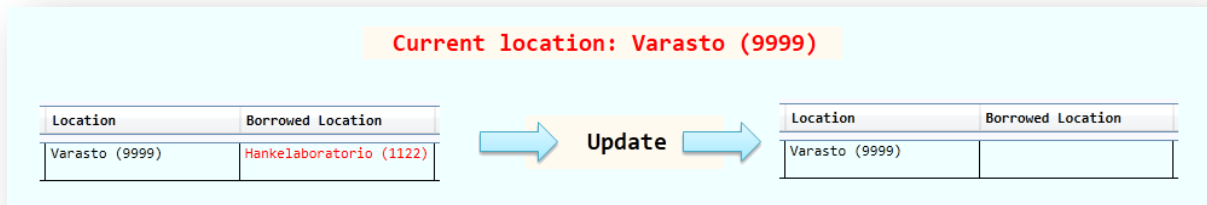
PICTURE 12. Main form of SIS Client after reading a barcode.

Notice that the 'List' button is now replaced with 'Update' and 'Clear' buttons. 'Clear' deselects the item, clears the grid and goes back to the normal state with only 'List' button visible. 'Update' instead works in two ways. If the current location is something else than the location the item is originally stored to ('Location' column), then 'Update' marks the item as borrowed to current location (picture 13).



PICTURE 13. Borrowing an item with SIS Client.

On the contrary, if the current location is the same location that the item is borrowed from ('Location' column) to some other location, 'Update' marks the item as returned (not borrowed anymore) and clears the 'Borrowed Location' field (picture 14).



PICTURE 14. Returning an item with SIS Client.

An item can't be borrowed to its original location. Already borrowed item can be borrowed to other locations without returning it first. Multiple clicks on 'Update' are ignored.

The 'Logout' button closes the main form and returns to the login form. So changing the current location is possible without restarting the application. After closing the login form, the whole application shuts down.

4.2 SIS MANAGEMENT UI

Management unit consists of six forms: login, main, new item, edit item, edit categories and locations and print form. When SIS Management is run, the login form pops up and asks for user to enter the correct login credentials (picture 15).



PICTURE 15. Login form of SIS Management.

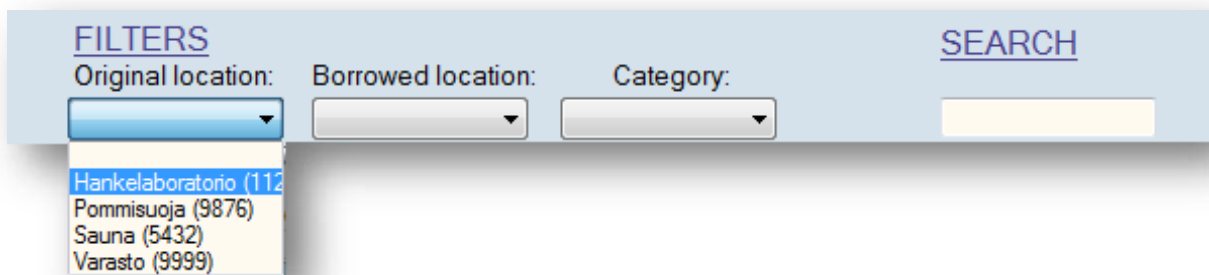
If the credentials match with the ones defined in the database, the main form opens and lists all the items found in database automatically (picture 16).

barcode	name	category	original location	borrowed location	borrower info.	edit date	description
100000000082	Hävikukuin	Veikotimet	Vaasto (0999)			14.5.2013 14:21	Tämä tuote viivakoodilla. Ei suositella jatkuvaa käyttöä.
100000000112	Banani	Häpäkkäet	Sauna (0430)	Hankalatoimisto (112)		14.5.2013 14:31	Banani (Musa sapientum) on tropiikissa (Musa) sukuisen kasvien ja kasvien ryhmä. Sen hedelmä on karvattomissa kuituissa nautittava, lakonemisen on pölyallergeniä sisältävä ja yleensä "mangola" tai "kaapillinen" kasveksi. Se on murtokivi, joka tuottaa säilyneenä kuumaa sokeria sisältävää.
100000000113	Apelsini	Häpäkkäet	Sauna (0430)			14.5.2013 14:30	Apelsini (Citrus sinensis) on sitruunin (Citrus) sukuisen kasvien ryhmä. Sen hedelmä sisältää runsaasti sokeria ja on murtokivi, joka tuottaa säilyneenä kuumaa sokeria sisältävää.
100000000146	Aprikoosi	Häpäkkäet	Vaasto (0999)			14.5.2013 14:30	Aprikoosi (Prunus avium) on pensaskasvi, josta peräisin oleva puu ja sen hedelmä. Aprikoosi on kirkkaan värinen, kirkkaan ja sen sisältöä tulee kuumaa sokeria sisältävää.
100000000115	Rypäle	Development Ki	Vaasto (0999)	Hankalatoimisto (112)		14.5.2013 10:35	
100000000030	Ananas	Veikotimet	Vaasto (0999)	Hankalatoimisto (112)		14.5.2013 10:34	Blaah...
100000000153	Avokado	Veikotimet	Pormuuse (0676)	Hankalatoimisto (112)		14.5.2013 10:34	Avokado (Persea americana) on kasvi, joka kuuluu kasvien ryhmään. Sen hedelmä on kirkkaan värinen ja sisältää runsaasti sokeria.
100000000160	Granatomena	Häpäkkäet	Pormuuse (0676)	Hankalatoimisto (112)		14.5.2013 10:34	Granatomena (Punica granatum) on kasvi, joka kuuluu kasvien ryhmään. Sen hedelmä on kirkkaan värinen ja sisältää runsaasti sokeria.

PICTURE 16. Main form of SIS Management.

'Logout' button on the top right side of the form closes the main form and brings back the login form. Same way that in SIS Client, closing that form will cause the application finally shut down.

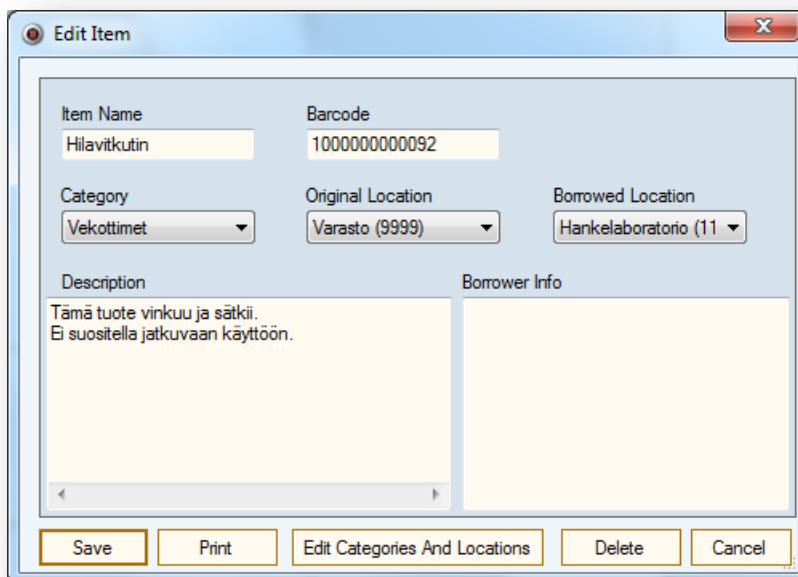
Unlike in SIS Client, the user can now sort the list by clicking any of the column headers (ascending/descending order), select any item from the list and start editing item properties by double clicking the item. The list is also modifiable with filters and a search field (picture 17). They are all combined together with AND operators so they work as a sum of terms defining the contents of the list. Search is set to work only with 'Name' and 'Description' columns and is not case-sensitive.



The image shows a user interface for filtering and searching items. It includes three dropdown menus labeled 'Original location:', 'Borrowed location:', and 'Category:'. The 'Original location:' dropdown is currently open, displaying a list of location options: 'Hankelaboratorio (112)', 'Pommisuoja (9876)', 'Sauna (5432)', and 'Varasto (9999)'. To the right of these filters is a search input field with a 'SEARCH' button.

PICTURE 17. Filters and Search on Main form of SIS Management.

Double clicking the item pops up the 'Edit Item' form (picture 18). All properties of the item are modifiable except the barcode value, which is generated automatically when the item is created. If the barcode label of the item is at hand, it's usually much faster to just read it to enter the 'Edit Item' form.



The 'Edit Item' form displays the following information:

- Item Name: Hilavitkutin
- Barcode: 1000000000092
- Category: Vekottimet
- Original Location: Varasto (9999)
- Borrowed Location: Hankelaboratorio (11)
- Description: Tämä tuote vinkuu ja sätkii. Ei suositella jatkuvaan käyttöön.
- Borrower Info: (Empty field)

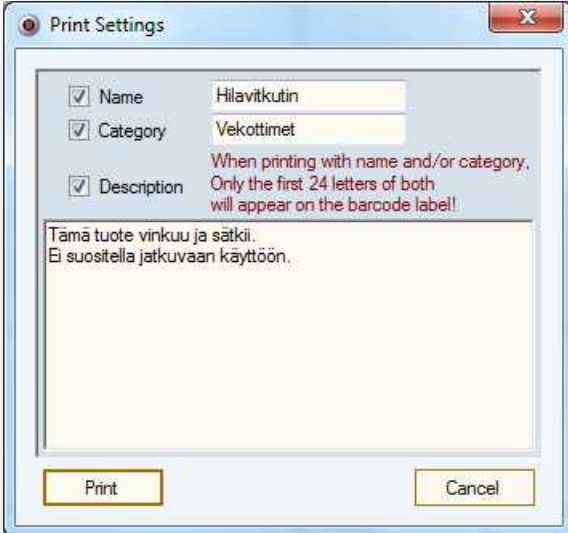
Buttons at the bottom: Save, Print, Edit Categories And Locations, Delete, Cancel.

PICTURE 18. Edit Item form of SIS Management.

The field 'Borrower Info' is not currently utilized, but it could contain information about the person who has borrowed the item. If the 'Borrowed Location' is empty, the data written to 'Borrower Info' won't be saved. 'Borrower Info' is not implemented in SIS Client, so it can't be used in normal borrowing procedures. It may be used in the future, when every SIS Client user is perhaps authenticated with some ID card or number.

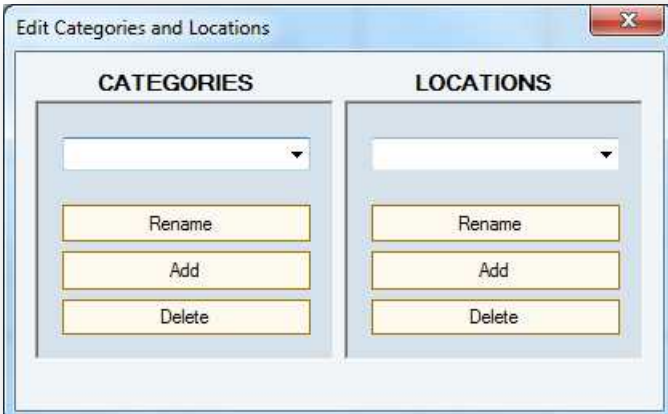
Deleting an item can be done by 'Delete' button on this form, but it can be done by selecting the item from main form (picture 16) and pressing Delete button on keyboard as well. 'Save' button saves all the changes to database as long as every required field is filled.

A label for the item can be printed by clicking the 'Print' button, that opens the 'Print Settings' form (picture 19). Usually it's printed when the new item is created, but it can also be printed here, if the old one is damaged or lost or the printing was skipped earlier on purpose. Print settings allow user to define which fields of information he likes to print. The barcode is always printed, but 'Name', 'Category' and 'Description' are voluntary. The default setting prints them all. After clicking the 'Print' button on this form, SIS Management sends the data for label printer to print and closes the 'Print Settings' form.



PICTURE 19. Print Settings form of SIS Management.

There's also a button labelled 'Edit Categories and Locations' on the 'Edit Item' form. It opens the corresponding form (picture 20).



PICTURE 20. Edit Categories and Locations form of SIS Management.

Rename works so that first the right category or location is selected from the dropdown list and then renamed in the same element. After clicking the 'Rename' button the new name is saved to database (picture 21).



PICTURE 21. *Renaming a category with SIS Management.*

Adding new categories or locations works quite similar. The name of a new category/location is put to dropdown list's empty field and it's added by clicking 'Add' button. Already existing names can't be added, but there's currently (v1.3.2) a bug that allows renaming an existing category/location over another so that there remain two similar categories/locations on the list, but they can't be handled individually anymore.

Deleting categories/locations is easy. Just select the unnecessary category/location from the dropdown list and click 'Delete'. If some item still belongs to that category or location, deleting won't work.

On the main form there's a button for adding new items to the database. Clicking that button opens a 'New Item' form (picture 22). Again, all properties of the item are modifiable except the barcode value, which is generated automatically. 'Item Name' and 'Location' are the only fields that are required for a new item. There's also a button for editing categories and locations on this form as well as there is another one on the main form. Both buttons open the 'Edit Categories and Locations' form (picture 20).

The image shows a software window titled "New Item" with a close button (X) in the top right corner. The window contains the following fields and controls:

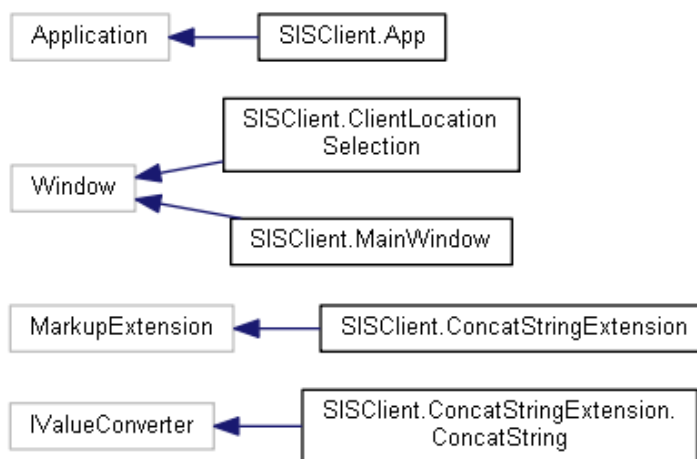
- Item Name:** A text input field that is currently empty.
- Barcode:** A text input field containing the value "1000000000245".
- Category:** A dropdown menu that is currently empty.
- Location:** A dropdown menu that is currently empty.
- Description:** A large text area for entering a description, which is currently empty.
- Buttons:** Three buttons are located at the bottom of the window: "Save", "Edit Categories And Locations", and "Cancel".

PICTURE 22. New Item form of SIS Management.

5 SYSTEM DESIGN

5.1 CLASS HIERARCHY

The class hierarchy of SIS Client consists of five classes (picture 23). **App** is generated by WPF and it's the declarative starting point of the application, so it's run before any form. It's the place for important application events, like application start and unhandled exceptions, but also the place for global resources. No changes have been made to the default App class of SIS Client.

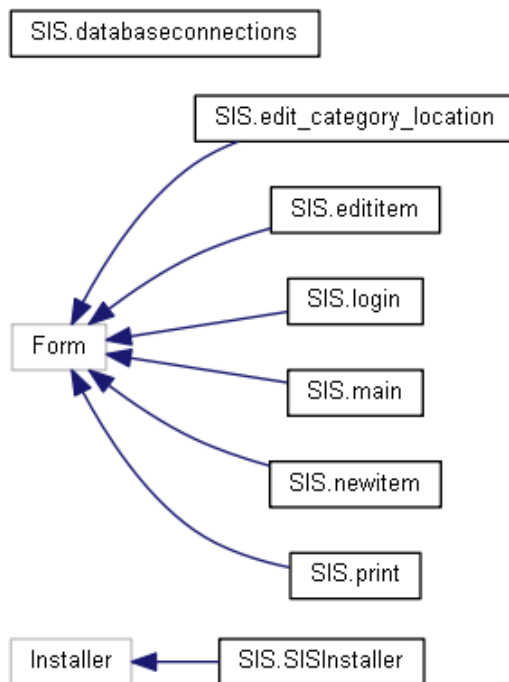


PICTURE 23. Class hierarchy of SIS Client.

The only visible forms, inherited from the Window class, are **ClientLocationSelection** and **MainWindow**. Those classes include all the source code of SIS Client.

ConcatStringExtension is a pre-coded class that is used to bind the chosen location from database to title field of **MainWindow**. It's located in the end of **MainWindow.xaml.cs**. **ConcatString** is a class defined inside **ConcatStringExtension** and it handles the actual string manipulation.

The class hierarchy of SIS Management looks a bit different because of the Windows Forms. It doesn't have the base classes **Application** and **Window** as all visible forms/windows derive from base class **Form**. SIS Management has six forms (picture 24).



PICTURE 24. Class hierarchy of SIS Management.

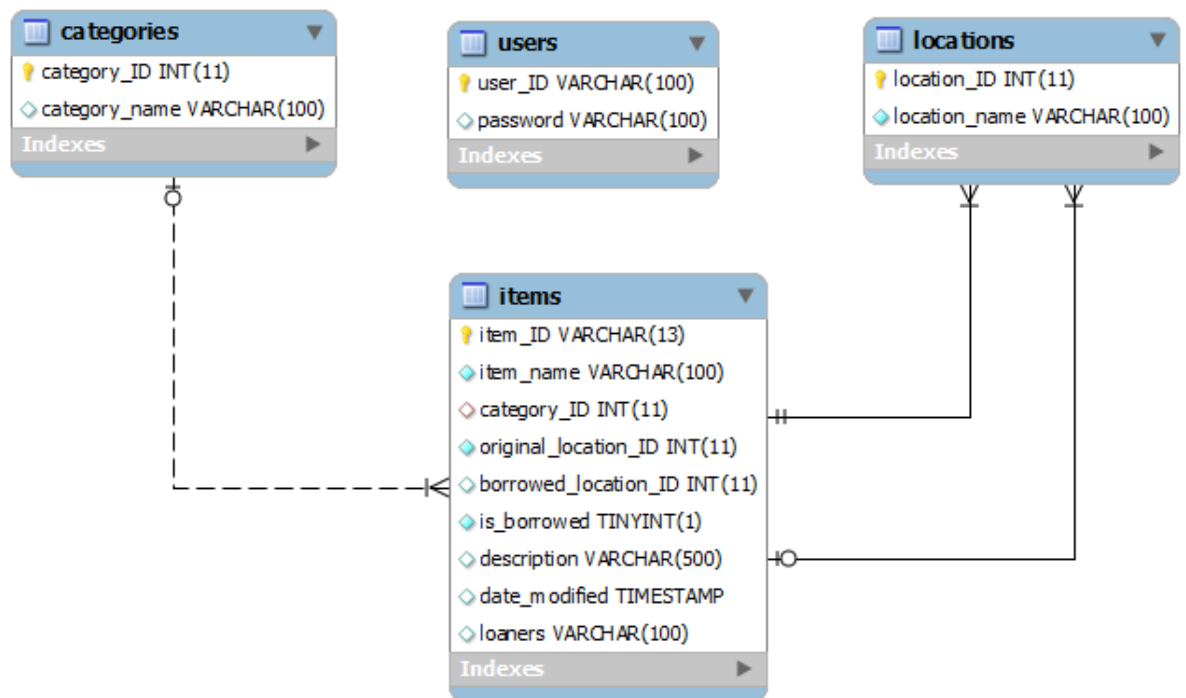
The class called **databaseconnections** defines all communication between the database and SIS Management. Every form uses its public methods when they need to get/set some values from/to database tables. The last class, **SISInstaller**, contains the data required to create the installer for SIS Management.

More information about the structure of SIS source code can be found from the HTML-formatted Doxygen documentation (references 3 & 4 in section 2.3).

5.2 DATABASE

The MySQL database was designed alongside the development of Scalable Inventory System itself and there were no changes made in testing and deployment stages. Because both SIS units use the .NET framework, the obvious choice for database driver was MySQL Connector/Net. It can be downloaded from <http://www.mysql.com/products/connector/>.

The database consists of four tables: users, items, categories and locations (picture 25).



PICTURE 25. The tables of SIS database.

Users table contains the valid credentials for SIS Management user authentication. SIS Management compares user’s input and the user_ID and password fields of this table. If they match, user is permitted access to open the main window. The table currently contains only one user whose credentials are shown below (censored in this public document):

Parameter	Specification
user_ID	*****
password	*****

Items table is the main table of the database. Every item added to the system is stored here. The data contains item’s barcode (item_ID), name (item_name), ID of category (category_ID), ID of original location (original_location_ID), ID of borrowed location (borrowed_location_ID), Boolean value of loan (is_borrowed), description (description), date of last modification (date_modified) and further details about the loaner (loaners). The last-mentioned isn’t utilized in any way in the system, but it’s added there for the upcoming development projects. With SIS Management it’s still possible to store string data to it, but it doesn’t affect the functions of the system nor won’t it be shown on SIS Client.

Categories table contains the names of the categories (category_name) and the corresponding IDs (category_ID). Swift adding, deleting and renaming of categories is

possible due to this table. Every item must belong to some category so that category's ID is stored to the item's data. The field category_ID is a link between items and categories tables. Therefore, if a category is modified, the changes affect all the items that belong to that category. The link also rules out the possibility to delete a category that one or several items belong to.

Locations table contains the names of the locations (location_name) and the corresponding IDs (location_ID). The behaviour and the benefits of a separate table are the same as with categories table. The difference is that there are two connections between items and locations: original_location_ID and borrowed_location_ID. Every item must have some original location, but borrowed_location_ID contains some location only if the item is borrowed at the same time. Scalable Inventory System makes sure that these fields don't ever contain the same location.

The SQL query to retrieve all the data of particular item is presented below:

```
SELECT
item_ID as 'barcode',
item_name as 'name',
c1.category_name as 'category',
l1.location_name as 'original location',
l2.location_name as 'borrowed location',
is_borrowed as borrowed,
date_modified as 'edit date',
description FROM items i1
JOIN categories c1 ON c1.category_ID = i1.category_ID
JOIN locations as l1 ON l1.location_ID = i1.original_location_ID
JOIN locations as l2 ON l2.location_ID = i1.borrowed_location_ID
```

The login information for MySQL server that holds the SIS database at OUAS is as follows (censored in this public document):

Parameter	Specification
Address	*****
User id	*****
Password	*****
Database	*****
Permissions	Read / Write

At the moment both units use the same login credentials even if it's not necessary for SIS Client to have unlimited write permissions.

5.3 DOCUMENTATION STANDARDS

The source code of Scalable Inventory System is documented using a special comment syntax that contains XML tags. It's a standard mechanism for documenting C# code. In source code files, comments having a certain form can be used to direct a tool to produce XML from those comments and the source code elements, which they precede. They can also be analysed with a documentation generator that generates software reference documentation from the structure of source code elements and their descriptions. Comments using such syntax are called documentation comments. They must immediately precede a user-defined type (such as a class, delegate, or interface) or a member (such as a field, event, property, or method).

The next example is a method that has a normal C-style comment:

```
// Saves all the item data to database
public DataSet update(string item_ID, string item_name, string category_ID, string
original_location_ID, string borrowed_location_ID, string loaners, string is_borrowed, string
description)
{
...
}
```

The example below is the same method that's been documented with XML tags:

```
/// <summary>
/// Saves all the item data to database.</summary>
/// <remarks>
/// Used when user presses Save button in New Item or Edit Item forms.</remarks>
/// <param name="item_ID">Barcode value of an item.</param>
/// <param name="item_name">Title of an item.</param>
/// <param name="category_ID">ID of the category to which current item belongs to.</param>
/// <param name="original_location_ID">ID of the location to which current item originally
belongs to.</param>
/// <param name="borrowed_location_ID">ID of the location to which current item is lent
to.</param>
/// <param name="loaners">Undefined information about the person who is about to borrow
current item.</param>
/// <param name="is_borrowed">Boolean value that defines if current item is borrowed or
not.</param>
/// <param name="description">Further information about the contents of current item.</param>
/// <returns>Dataset containing all the item data that matches with the item ID.</returns>
public DataSet update(string item_ID, string item_name, string category_ID, string
original_location_ID, string borrowed_location_ID, string loaners, string is_borrowed, string
description)
{
...
}
```

This method is much more structured and specific. An instruction manual for C# source code documenting was made during the documentation of SIS source code

and it's available in both English and Finnish. It contains more information about using document comments and documentation in general. Another great resource is C# Language Specification and its appendix A, Documentation comments: <http://www.microsoft.com/en-us/download/confirmation.aspx?id=7029>

5.4 SOFTWARE DEVELOPMENT TOOLS

The environment used to develop Scalable Inventory System was Microsoft Visual Studio. Installers for both SIS units were also made with Visual Studio which requires the Professional edition. Adding drivers of MySQL Connector and Brother Printer to the installers required that they were first wrapped to bootstrappers. The solution used to accomplish this was Bootstrapper Manifest Generator. Doxygen GUI was used to generate the Doxygen documentation. The version information of these programs shown below refers to the versions used in testing and deployment stages:

Program	Version
<i>Microsoft Visual Studio</i>	<i>2010 Professional (10.0.40219.1) SP1</i>
<i>Bootstrapper Manifest Generator</i>	<i>2.00</i>
<i>Doxygen GUI</i>	<i>1.8.3.1</i>

6 COMPONENT DESCRIPTION

The system has two components: **MySQL Connector/Net** and **b-PAC Client Component**. Both SIS units use the MySQL Connector/Net to communicate with MySQL database, but b-PAC Client Component is used only in SIS Management to provide the interface for printing labels.

6.1 MYSQL CONNECTOR/NET

MySQL Connector/NET is a fully managed ADO.NET driver written in C# that enables developers to easily create .NET applications with MySQL. It implements the required ADO.NET interfaces and integrates into ADO.NET aware tools.

MySQL Connector/NET uses the next four components to interact with the database:

MySqlConnection	Establishes a connection to a specific MySQL database
MySqlCommand	Executes a command against a MySQL database
MySqlDataReader	Reads a stream of data from a MySQL database
MySqlDataAdapter	Populates a data set and resolves updates with the MySQL database

SIS Client source code files that use MySQL Connector/NET:

- ClientLocationSelection.xaml.cs
- MainWindow.xaml.cs

SIS Management source code files that use MySQL Connector/NET:

- databaseconnections.cs

Connector/Net Download

<http://dev.mysql.com/downloads/connector/net/>

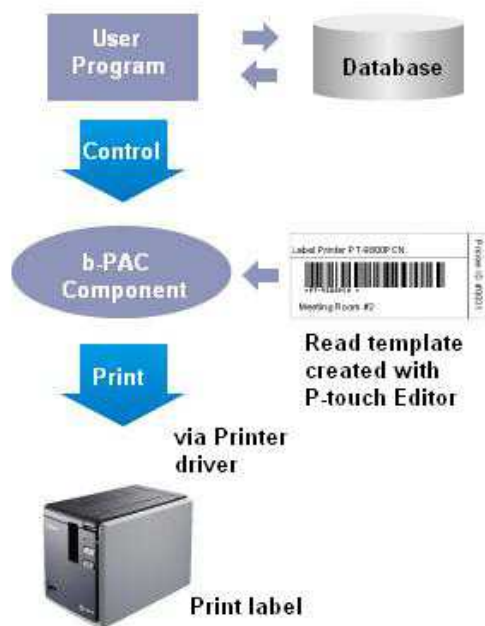
Connector/Net Tutorials

<http://dev.mysql.com/doc/refman/5.7/en/connector-net-tutorials.html>

6.2 B-PAC CLIENT COMPONENT

b-PAC is a component for outputting to Brother's printers (P-touch/Mobile Printer/Stampcreator series). It is used to replace text, barcodes and images within template files created with P-touch Editor. b-PAC is packaged as a COM automation server. If an application that can handle COM automation servers, such as Visual Basic, Visual C++, or Visual C# or a development environment is used, outputting is possible without using P-touch Editor.

The b-PAC provides an automation function for using P-touch Editor document files (.lbx, *.lbl) as templates and for outputting them from the user's program to a P-touch series printer without using the Editor (picture 26).*



PICTURE 26. The printing process with b-PAC Component. (© b-PAC 3.0 SDK help file)

SIS Management source code files that use b-PAC Component:

- `print.cs`

b-PAC Client Component (v3.0.017) Download

<http://www.brother.com/product/dev/label/bpac/download/bcciw30015/index.htm>

b-PAC 3.0 SDK Help

http://www.brother.com/pub/bpac/download/bPAC30_eng.zip

7 INSTALLATION INSTRUCTIONS

7.1 SIS CLIENT INSTALLATION

1. *Run setup.exe and follow the installation instructions.*
2. *Connect barcode reader to USB port. Windows should take care of the proper driver installation.*
3. *Run SIS Client.*

Default location:

Start > All Programs > OAMK > SIS Client > SIS Client.exe

C:\Program Files\OAMK\SIS Client\SIS Client.exe

Problem Solving

- P:** *SIS Client fails to connect to database and freezes/crashes.*
- S:** *Internet connection is probably cut off. Close the program, fix the connection and try again.*
-
- P:** *Barcode reader states an error or can't find my item, when reading a valid barcode.*
- S:** *Check that the keyboard language is set to FI or EN (can be changed from system tray).*

7.2 SIS MANAGEMENT INSTALLATION

1. *Download and install drivers for Brother QL-500 label printer:*
http://welcome.solutions.brother.com/bsc/public/eu/fi/fi/dlf/download_top.html?req=eu&c=fi&lang=fi&prod=lpq1550euk
2. *Run setup.exe and follow the installation instructions.*
3. *Connect barcode reader and label printer to USB ports, if they're not already connected and turn label printer on.*
4. *Run SIS Management*

Default location:

Start > All Programs > OAMK > SIS Management > SIS Management.exe

C:\Program Files\OAMK\SIS Management\SIS Management.exe

Problem Solving

P: *SIS Management crashes when trying to print.*

S: *1. Check that label printer is connected and turned on.*

2. Check that Brother QL-500 is recognised as a printer under Devices and Printers (Control Panel > Hardware and Sound > Devices and Printers).

Try to select it as default printer. If it can't be found, reinstall Brother QL-500 drivers and reboot.

P: *SIS Management fails to connect to database and freezes/crashes.*

S: *Internet connection is probably cut off. Close the program, fix the connection and try again.*

P: *Barcode reader states an error or can't find my item, when reading a valid barcode.*

S: *Check that the keyboard language is set to FI or EN (can be changed from system tray).*

DOCUMENT CONTROL

Title: Technical Document
Version: 1.1
Date Modified: 14.11.2013
Author: Antti Vänskä
Distribution: Oulu University of Applied Sciences
Filename: sis.technical.documentation.v1.1.docx

DOCUMENT SIGNOFF

Nature of Signoff	Person	Signature	Date	Role
Authors	Antti Vänskä			Project Member
Reviewers	Tommi Sallinen			Client

DOCUMENT CHANGE RECORD

Date	Version	Author	Change Details
02 April 2013	0.1	Antti Vänskä	Sections 1, 2 and 3
15 May 2013	0.2	Antti Vänskä	User Interface Design
10 September 2013	0.3	Antti Vänskä	Form logic diagrams for UI Design
5 November 2013	0.4	Antti Vänskä	System Design, Installation Instructions
6 November 2013	0.5	Antti Vänskä	Component Description
7 November 2013	1.0	Antti Vänskä	Minor fixes
11 November 2013	1.1	Antti Vänskä	SIS Management login credentials

Using comments and document comments in C#

Document created by:
Antti Vänskä
HYV8SN
Medical Engineering
School of Engineering
Oulu University of Applied Sciences

Introduction

Using **comments** in C# source code is done the same way as in any C based language:

single-line comment	<code>// this is a single-line comment</code>
multiple lines comment	<code>/* this is a comment that spans multiple lines */</code>

There's a much more versatile alternative for comments that can achieve the same and more: **document comments**. The same way as comments, document comments are also wiped out when compiling the code. Using document comments is more organized and accurate, which makes it possible to run the code through [a documentation generator](#). It generates a software documentation which basically is an easy read summary of the structure of source code with plots and tables. This kind of documentation can be very useful for developers that are just studying the code made by someone else. The most popular documentation generators are e.g. Doxygen (C++, C, C#, Objective-C, Java, Python, Fortran, PHP, etc.) and Javadoc (Java).

C# provides a method for using document comments in XML syntax. Before adding any XML tags, we have to tell the compiler that it should handle our tags as document comments:

single-line document comment	<code>/// document comment</code>
multiple lines document comment	<code>/** document comments on multiple lines */</code>

The style for multiple lines above should be avoided, because integrated development environments (IDE, such as Microsoft Visual Studio) usually can generate the necessary rows ahead of the element (e.g. class) automatically so that every line starts with `///`.

The actual documenting is done inside XML tags. With different tags we can define which element of the code we are documenting. The most common tags are listed below, but creating custom tags is also possible. The most used tags are marked in bold type.

Tag	Purpose
<c>	Set text in a code-like font
<code>	Set one or more lines of source code or program output
<example>	Indicate an example
<exception>	Identifies the exceptions a method can throw
<include>	Includes XML from an external file
<list>	Create a list or table
<para>	Permit structure to be added to text
<param>	Describe a parameter for a method or constructor
<paramref>	Identify that a word is a parameter name
<permission>	Document the security accessibility of a member
<remark>	Describe additional information about a type
<returns>	Describe the return value of a method
<see>	Specify a link
<seealso>	Generate a <i>See Also</i> entry
<summary>	Describe a type or a member of a type
<value>	Describe a property
<typeparam>	Describe a generic type parameter
<typeparamref>	Identify that a word is a type parameter name

Notice that all documentation generators don't necessarily recognize all the tags, and if so, those tags will simply be skipped over.

More specific information about C# documenting and XML tags can be found from C# Language Specification:

C# Language Specification (VS 2012)

Appendix A: Documentation comments

<http://msdn.microsoft.com/en-us/library/vstudio/ms228593.aspx>

What should be documented?

The purpose of documenting code is to explain how user-defined types such as classes, delegates and interfaces, and the members such as variables, events, methods and properties actually work and/or what they do. A document comment always precedes the element being documented. For example, documenting a variable:

```
/// <summary>Row height of the main grid.</summary>
int rowHeight = 110;
```

Number of rows doesn't matter as long as XML syntax remains intact. More important is to keep the notation consistent so that the code is easier to read. The document comment below works as well:

```
/// <summary>
/// Row height of the main grid.</summary>
int rowHeight = 110;
```

It may be smart to keep document comment of a single variable in one line and span larger entities (such as method documentation) to multiple lines. For example, a method that receives multiple variables:

```
/// <summary>
/// Saves all the item data to database.</summary>
/// <remarks>
/// Used when user presses Save button in New Item or Edit Item forms.</remarks>
/// <param name="item_ID">Barcode value of an item.</param>
/// <param name="item_name">Title of an item.</param>
/// <param name="category_ID">ID of the category to which current item belongs to.</param>
/// <param name="original_location_ID">ID of the location to which current item originally belongs to.</param>
/// <param name="borrowed_location_ID">ID of the location to which current item is lent to.</param>
/// <param name="loaners">Undefined information about the person who is about to borrow current item.</param>
/// <param name="is_borrowed">Boolean value that defines if current item is borrowed or not.</param>
/// <param name="description">Further information about the contents of current item.</param>
/// <returns>Dataset containing all the item data that matches with the item ID.</returns>
public DataSet update(string item_ID, string item_name, string category_ID, string original_location_ID, string
borrowed_location_ID, string loaners, string is_borrowed, string description)
{
    ...
}
```

There's no point to write all those rows manually, because IDE usually recognizes when user wants to document something. When the method is finished, just write three slashes (///) above the definition of your type or member, and the IDE adds all the empty fields in place automatically:

```

/// <summary>
///
/// </summary>
/// <param name="item_ID"></param>
/// <param name="item_name"></param>
/// <param name="category_ID"></param>
/// <param name="original_location_ID"></param>
/// <param name="borrowed_location_ID"></param>
/// <param name="loaners"></param>
/// <param name="is_borrowed"></param>
/// <param name="description"></param>
/// <returns></returns>
public DataSet update(string item_ID, string item_name, string category_ID, string original_location_ID, string
borrowed_location_ID, string loaners, string is_borrowed, string description)
{
...

```

This automation doesn't necessarily work in all IDEs nor in the situations when document comments are placed above something else than definition of a class/variable/method/etc. For example, if it's meaningful to clarify the behavior of if sentence, the tags must be written manually:

```

/// <summary>
/// If SQL query didn't find any results, opens a New Item form.</summary>
if (!mysqlreader.HasRows)
{
    newitem newitem = new newitem();
    newitem.barcodetxt.Text = item_ID;
    newitem.ShowDialog();
}
/// <summary>
/// If SQL query found the matching item,
/// opens an Edit Item form and copies all the data to it.</summary>
else
{
    edititem edit = new edititem();
    edit.barcodetxt.Text = item_ID;
    edit.itemnametxt.Text = dataset.Tables[0].Rows[0][1].ToString();
    edit.edittxt.Text = dataset.Tables[0].Rows[0][2].ToString();
    edit.edittxt2.Text = dataset.Tables[0].Rows[0][3].ToString();
    edit.edittxt3.Text = dataset.Tables[0].Rows[0][4].ToString();
    edit.descriptiontxt.Text = dataset.Tables[0].Rows[0][8].ToString();
    edit.borrowinfotxt.Text = dataset.Tables[0].Rows[0][6].ToString();
    edit.ShowDialog();
}

```

In this way the IDE doesn't bind document comments to any element, but they still help user to interpret code and are also included to generated documentation as separated lines inside the current type. There are also other benefits than generated documentation when using document comments. The method above named 'update' has a tag <summary>, which contains a short description of that method. When we examine the part of code where the 'update' is called, we can see its description by hovering mouse cursor on it (picture below from Visual Studio):

```
/// <summary>
/// Asks for confirmation for updating the item's data to database.</summary>
DialogResult dlgresult = MessageBox.Show("Are you sure you want to update " + item_name + "?", "Confirm up

if (dlgresult == DialogResult.Yes)
{
    dbcons.update(item_ID, item_name, category_ID, original_location_ID, borrowed_location_ID, loaners, is
    this.Di
    dbcons. DataSet databaseconnections.update(string item_ID, string item_name, string category_ID, string original_location_ID, string
    dbcons. Saves all the item data to database.
}
```

We won't have to jump to the definition of the method (which can be located in other source file) to see what it does, when we can see that from the tooltip.

Best practices

When documenting (or adding comments to) source code, it's good to keep in mind why we're doing it. The purpose is to help others understand the code, so that a developer who's new to the project can get wind of the code without wasting precious time. Furthermore, it's easier to examine even own code that was written long time ago, when it's documented properly.

It's recommended to keep descriptions (summary) short and unambiguous, because experienced developers usually read only the description from the documentation of a class/method. However it's smart to describe the element behavior more specifically as additional information (remarks), because the first impression gained from the short description may vary due to e.g. different interpretations.

There's no need to document every line of code, because it just complicates reading it. For example, if a short method doesn't contain any complex algorithms, its behavior can be explained entirely in description (summary). Details shouldn't be documented line by line if the same information can be included explicitly in to names (titles) of variables/methods/etc. The next if sentence is so clear that the document comment above doesn't give any added value, but is just useless repetition:

```
/// <summary>If request doesn't exist...</summary>
if ( !$request ) {
    ...
}
```

Similarly the next document comment above the definition of an integer variable is pointless:

```
/// <summary>Limit for the grid height.</summary>
int grid_height_limit = 100;
```

Instead, if the variable isn't named explicitly, the document comment can show essential information about its purpose:

```
/// <summary>Limit for the grid height.</summary>
int height = 100;
```

But still the best way is to name the variable explicitly and leave it undocumented:

```
int grid_height_limit = 100;
```

Consideration is advised though; at least if there are lots of similar variables that have differences in behavior/purpose. When writing the code documentation it could be useful to imagine yourself as a rookie developer and think about can you understand the code without additional details. If not, you should add some document comments.

Remember to use enough line breaks and intend the code, so that it's even nicer to read!

Other comments

Even if the behavior of source code is documented properly (with document comments), there's still many ways to make use of normal comments. You can add notes, which may not be important by content, but are useful because of their location. For example, the curly bracket that ends a long method can be marked with a comment, so that it's easily recognizable:

```
    ...

    CloseConnection();
    return dataset;

} // End of EditLocation()

} // End of class databaseconnections

} // End of namespace SIS
```

Another useful tip is to sort similar methods to own sections with comments:

```
//-----KEYBOARD-----//

private void barcodetext_KeyPress(object sender, KeyPressEventArgs e)
{ ... }

private void barcodetext_KeyDown(object sender, KeyEventArgs e)
{ ... }

private void main_KeyPress(object sender, KeyPressEventArgs e)
{ ... }

//-----MOUSE -----//

private void main_MouseEnter(object sender, EventArgs e)
{ ... }

private void main_MouseWheel(object sender, MouseEventArgs e)
{ ... }

private void mainlistview_CellMouseDoubleClick(object sender, DataGridViewCellMouseEventArgs e)
{ ... }

private void mainlistview_MouseClick(object sender, MouseEventArgs e)
{ ... }

//-----BUTTONS-----//

private void logoutbtn_Click(object sender, EventArgs e)
{ ... }

private void newbtn_Click(object sender, EventArgs e)
{ ... }

private void editcategoriesbtn_Click(object sender, EventArgs e)
{ ... }
```



```
//-----TIMERS-----//  
  
private void focus_timer_Tick(object sender, EventArgs e)  
{ ... }  
  
private void search_timer_Tick(object sender, EventArgs e)  
{ ... }  
  
private void barcode_timer_Tick(object sender, EventArgs e)  
{ ... }
```

Other notes outside the documentation can be added too. It's a common habit to e.g. add the name of the source file to the first line of the file with comment to inform which file we're modifying:

```
// edititem.cs  
  
using System;  
using System.Collections.Generic;  
...
```

Example of documented class

In source code:

```
/// <summary>
/// Login authentication method.</summary>
/// <remarks>
/// Checks that username and password are valid.</remarks>
/// <param name="user">Username entered by user.</param>
/// <param name="password">Password entered by user.</param>
/// <returns>A boolean value that defines if the user has privileges to access database.</returns>
public bool login_check(string user, string password)
{
    bool access = false;

    OpenConnection();

    /// <summary>
    /// Forms the SQL command and parameters.</summary>
    MySqlCommand mysqlcmd = new MySqlCommand();
    MySqlDataAdapter mysqladapt = new MySqlDataAdapter();
    mysqladapt.SelectCommand = new MySqlCommand("select * from users where user_ID=@user and
password=@password;", mysqlCon);
    mysqladapt.SelectCommand.Parameters.Add(new MySqlParameter("@user", user));
    mysqladapt.SelectCommand.Parameters.Add(new MySqlParameter("@password", password));
    dataset = new DataSet();

    try
    {
        /// <summary>Runs the SQL command.</summary>
        MySqlDataReader mysqlreader = mysqladapt.SelectCommand.ExecuteReader();

        if (mysqlreader.Read() != false)
        {
            if (mysqlreader.IsDBNull(0) == true)
            {
                /// <summary>If no results are found, access is denied.</summary>
                access = false;
            }
            else
            {
                /// <summary>If username and password match, access is granted.</summary>
                access = true;
            }
        }
        else
        {
            access = false;
        }

        mysqlreader.Close();
    }
    catch (Exception Er)
    {
        Console.WriteLine(Er);
    }
}
```

```
        CloseConnection();
        return access;

    } // End of login_check()
```

Doxygen generated documentation (html):

```
bool SIS.databaseconnections.login_check ( string user,
                                           string password
                                           )
```

Login authentication method.

Checks that username and password are valid.

Parameters

user Username entered by user.
password Password entered by user.

Returns

A boolean value that defines if the user has privileges to access database.

Forms the SQL command and parameters.

Runs the SQL command.

If no results are found, access is denied.

If username and password match, access is granted.

Definition at line 46 of file databaseconnections.cs.

```
47     {
48
49         bool access = false;
50
51         OpenConnection();
52
53         MySqlCommand mysqlcmd = new MySqlCommand();
54         MySqlDataAdapter mysqladapt = new MySqlDataAdapter();
55         MySqlCommand selectCommand = new MySqlCommand("select * from users where user_ID=@user and password=@password;", mysqlCon);
56         mysqladapt.SelectCommand.Parameters.Add(new MySqlParameter("@user", user));
57         mysqladapt.SelectCommand.Parameters.Add(new MySqlParameter("@password", password));
58         DataSet dataset = new DataSet();
59
60         try
61         {
62             MySqlDataReader mysqlreader = mysqladapt.SelectCommand.ExecuteReader();
63
64             if (mysqlreader.Read() != false)
65             {
66                 if (mysqlreader.IsDBNull(0) == true)
67                 {
68                     access = false;
69                 }
70                 else
71                 {
72                     access = true;
73                 }
74             }
75             else
76             {
77                 access = false;
78             }
79
80             mysqlreader.Close();
81
82             catch (Exception Ex)
83             {
84                 Console.WriteLine(Ex);
85             }
86
87             CloseConnection();
88             return access;
89
90         } // End of login_check()
```

Testaussuunnitelma

Scalable Inventory System (SIS)

Versio 1.0

Muutoshistoria

Versionumero	Pvm	Selitys	Tekijä(t)
1.0	18.06.2013	Päivityksiä vastuisiin ja liitteisiin	Antti
0.4	23.05.2013	SIS Clientin testitapausten kuvaukset	Antti
0.3	13.05.2013	Testitapausten luettelo	Antti
0.2	10.05.2013	Kappaleet 1 ja 2	Antti
0.1	07.05.2013	Testaussuunnitelmapohja	Antti

OAMK HYVINVOINTITEKNOLOGIA		Testaussuunnitelma		2 (34)
Prepared ANTTI VÄNSKÄ	File.doc liite5.testaussuunnitelma.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

Sisällysluettelo

1	JOHDANTO	4
2	TESTAUKSEN SISÄLTÖ JA RAJAUS	5
2.1	LÄHESTYMISTAPA	5
2.2	LÄPIPÄÄSYKRITEERIT	5
2.3	KESKEYTTÄMIS- JA JATKAMISVAATIMUKSET	6
2.4	TESTAUKSEN RAJAUS	6
2.5	TESTAUSTEHTÄVÄT	6
2.6	TESTAUSYMPÄRISTÖ	6
2.7	TESTAUSDOKUMENTAATIO	7
2.8	VASTUUT	7
2.9	AIKATAULU	7
3	TESTITAPAUKSET: SIS CLIENT	8
3.1	ASENNUS	8
3.2	TILAN VALITSEMINEN	8
3.3	TILAAAN LIITTYVIEN TUOTTEIDEN LISTAAMINEN	9
3.4	VIIVAKOODIN LUKEMINEN	9
3.4.1	<i>Lukeminen tyhjällä listalla</i>	9
3.4.2	<i>Lukeminen listauksen jälkeen</i>	9
3.4.3	<i>Lukeminen toisen tuotteen ollessa valittuna</i>	10
3.4.4	<i>Lukeminen useaan kertaan</i>	10
3.4.5	<i>Puuttuvan tuotteen viivakoodin lukeminen</i>	10
3.4.6	<i>Väärän viivakoodin lukeminen</i>	11
3.5	LAINAAMINEN	12
3.5.1	<i>Lainaaminen toiseen tilaan</i>	12
3.5.2	<i>Lainatun lainaaminen kolmanteen tilaan</i>	12
3.5.3	<i>Lainaaminen useaan kertaan</i>	12
3.6	PALAUTTAMINEN	13
3.6.1	<i>Lainatun palauttaminen alkuperäiseen tilaan</i>	13
3.6.2	<i>Palautetun palauttaminen uudestaan</i>	13
4	TESTITAPAUKSET: SIS MANAGEMENT	14
4.1	ASENNUS	14
4.2	KIRJAUTUMINEN	15
4.2.1	<i>Kirjautuminen väärillä tunnuksilla</i>	15
4.2.2	<i>Kirjautuminen oikeilla tunnuksilla</i>	15
4.3	VIIVAKOODIN LUKEMINEN	16
4.3.1	<i>Normaali lukeminen</i>	16
4.3.2	<i>Lukeminen toisen tuotteen ollessa valittuna</i>	16
4.3.3	<i>Lukeminen useaan kertaan</i>	16
4.3.4	<i>Puuttuvan tuotteen viivakoodin lukeminen</i>	17
4.3.5	<i>Väärän viivakoodin lukeminen</i>	17
4.4	UUDEN TUOTTEEN LISÄÄMINEN	18
4.4.1	<i>Puutteellisilla tiedoilla</i>	18
4.4.2	<i>Pitkillä ja erikoismerkkejä sisältävillä kentillä</i>	18

Prepared ANTTI VÄNSKÄ	File.doc liite5.testaussuunnitelma.v1.0.docx	Version 1.0
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11
	Printed 2013-11-11	

4.5	TUOTTEEN POISTAMINEN	19
4.5.1	<i>Tuotteen muokkaustilassa</i>	19
4.5.2	<i>Tuotteen ollessa valittuna</i>	19
4.6	KATEGORIAN MUOKKAAMINEN	20
4.6.1	<i>Uuden kategorian luominen</i>	20
4.6.2	<i>Uuden kategorian luominen olemassa olevan kategorian nimellä</i>	20
4.6.3	<i>Uudelleennimeäminen.....</i>	20
4.6.4	<i>Uudelleennimeäminen toisen kategorian päälle</i>	21
4.6.5	<i>Käytössä olevan kategorian poistaminen</i>	21
4.6.6	<i>Vapaan kategorian poistaminen</i>	22
4.7	TILAN MUOKKAAMINEN.....	23
4.7.1	<i>Uuden tilan luominen</i>	23
4.7.2	<i>Uuden tilan luominen olemassa olevan tilan nimellä.....</i>	23
4.7.3	<i>Uudelleennimeäminen.....</i>	23
4.7.4	<i>Uudelleennimeäminen toisen tilan päälle</i>	24
4.7.5	<i>Käytössä olevan tilan (original location) poistaminen.....</i>	24
4.7.6	<i>Käytössä olevan tilan (borrowed location) poistaminen</i>	24
4.7.7	<i>Vapaan tilan poistaminen.....</i>	25
4.8	TUOTTEEN MUOKKAAMINEN	26
4.8.1	<i>Puutteellisilla tiedoilla.....</i>	26
4.8.2	<i>Pitkillä ja erikoismerkkejä sisältävillä kentillä.....</i>	26
4.8.3	<i>Lainaaminen alkuperäiseen tilaan</i>	26
4.9	TUOTTEEN TULOSTAMINEN	28
4.9.1	<i>Pelkällä viivakoodilla</i>	28
4.9.2	<i>Name</i>	28
4.9.3	<i>Name + Category</i>	29
4.9.4	<i>Name + Description</i>	29
4.9.5	<i>Name + Category + Description</i>	29
4.9.6	<i>Category</i>	30
4.9.7	<i>Category + Description.....</i>	30
4.9.8	<i>Description.....</i>	31
4.10	FILTTEREIDEN KÄYTTÄMINEN	32
4.10.1	<i>Original location</i>	32
4.10.2	<i>Borrowed location</i>	32
4.10.3	<i>Category</i>	32
4.11	HAUN KÄYTTÄMINEN	33
LIITTEET.....		34

OAMK HYVINVOINTITEKNOLOGIA		Testaussuunnitelma		4 (34)
Prepared ANTTI VÄNSKÄ	File.doc liite5.testaussuunnitelma.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

1 JOHDANTO

Tämä testaussuunnitelma toimii pohjana Scalable Inventory System -järjestelmän käyttöönottotestaukselle. Järjestelmää on testattu kattavasti koko sen kehityksen ajan, mutta tässä dokumentissa kuvattujen toimenpiteiden jälkeen järjestelmä on määrä ottaa varsinaiseen käyttöön. Käyttöönottotestaus järjestetään tämän testaussuunnitelman mukaisesti ja tuloksista laaditaan erikseen testausraportti.

Scalable Inventory System on kahdesta yksiköstä koostuva inventaariojärjestelmä, jonka avulla on tarkoitus ylläpitää viivakoodein yksilöityjen tuotteiden varastointia ja lainaamista tilasta toiseen. Management-yksiköllä järjestetään tuotteiden, tilojen ja kategorioiden luominen, muokkaaminen, selaaminen ja poistaminen. Lisäksi sillä voidaan tulostaa tuotteille viivakoodit laitteistoon liitetyn tarratulostimen avulla. Client-yksiköllä puolestaan järjestetään tuotteiden normaalit lainaus- ja palautusrutiinit ja voidaan myös selata valitun tilan sisältämiä ja sinne lainattuja tuotteita. Käyttöönottotestaus käsittää molemmat yksiköt.

Testaussuunnitelma koostuu tämän johdannon lisäksi kahdesta osasta: kappaleessa 2 on määritelty kaikki testauksen järjestämiseen ja siinä käytettäviin menetelmiin liittyvät tekijät ja kappaleesta 3 löytyvät varsinaiset testitapaukset, jotka testauksessa suoritetaan.

OAMK HYVINVOINTITEKNOLOGIA		Testaussuunnitelma		5 (34)
Prepared ANTTI VÄNSKÄ	File.doc liite5.testaussuunnitelma.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

2 TESTAUKSEN SISÄLTÖ JA RAJAUS

2.1 Lähestymistapa

Kyseessä on käyttöönottotestaus, jossa tietojärjestelmän toimintojen testaamisella pyritään varmistamaan, että järjestelmä toimii sopimuksen mukaisesti eli vastaa sille asetettuja vaatimuksia. Testauksella pyritään myös löytämään ja korjaamaan mahdolliset virheet ennen käyttöönottoa. Toimintoja testataan suoraan järjestelmän käyttöliittymien kautta ns. musta laatikko -testauksena.

2.2 Läpikäyskriteerit

Testattava kohde on läpäissyt testin, kun se on suoritettu onnistuneesti viisi (5) kertaa ilman virhetilanteita. Onnistunut suoritus tarkoittaa, että ohjelman vaste vastaa suunniteltua toimintaa. Jokaisen testitapauksen viimeinen suoritus tehdään ilman internet-yhteyttä (kaapeli irti) ja varmistetaan, että ohjelma antaa asianmukaisen virheilmoituksen, eikä kaadu omia aikojaan. Ohjelmien asennusta ei tarvitse testata moneen kertaan, kuten ei myöskään kappaleessa 4.9. käsitellyjä tulostustoimintoja (tarrojen säästämiseksi).

Mikäli joidenkin testien kohdalla käytettävissä oleva testausmateriaali tai jokin muu tekijä rajoittaa suoritettavien toistojen määrää, kyseiset testit suoritetaan kattavimmalla mahdollisella tavalla. Kaikista poikkeamista toistojen määrässä ilmoitetaan testausraportissa sekä arvioidaan niiden vaikutusta testauksen luotettavuuteen.

OAMK HYVINVOINTITEKNOLOGIA		Testaussuunnitelma		6 (34)
Prepared ANTTI VÄNSKÄ	File.doc liite5.testaussuunnitelma.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

2.3 Keskeyttämis- ja jatkamisvaatimukset

Mikäli jossain testissä läpipääsykriteerit eivät täyty, testaus sen osalta päättyy. Testausraporttiin kirjataan virhetilanteiden aiheuttajat ja seuraukset sekä arvioidaan lyhyesti virheen laatua. Ohjelman kaatuminen on aina syy keskeyttää testi. Projektin tilaaja päättää jatkotoimenpiteistä ja mahdollisista lisätestauksista.

2.4 Testauksen rajaus

Testaus kattaa ohjelmien asennuksen ja normaalin käytön. Laitteistokokoonpanojen ja käyttöjärjestelmien vaikutusta ohjelmien käyttäytymiseen ei varsinaisesti testata, vaan pääpaino on itse ohjelmien toiminnoilla. Mikäli yhteensopivuusongelmia havaitaan, ne kuvataan mahdollisimman kattavasti testausraporttiin ja otetaan jatkoselvitykseen. Myöskään kuormittavuus- tai tietoturvanäkökulmia ei erikseen testata, mikäli niitä ei testausilanteessa syystä tai toisesta koeta välttämättömiksi.

2.5 Testaustehtävät

1. Testaussuunnitelman laatiminen
2. Käyttöönottotestaus
3. Raportointi

2.6 Testausympäristö

Testaus suoritetaan OAMK:n tekniikan yksikössä testaukseen soveltuvassa tilassa. Se suoritetaan yhdellä tai mielellään kahdella laitteistokokoonpanolla. Tämä tarkoittaa sitä, että ajan ja muiden resurssien salliessa sekä Clientiä että Managementia testattaisiin yhteensä neljässä eri koneessa. Käyttöjärjestelmänä toimii mikä tahansa

OAMK HYVINVOINTITEKNOLOGIA		Testaussuunnitelma		7 (34)
Prepared ANTTI VÄNSKÄ	File.doc liite5.testaussuunnitelma.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

työasemakäyttöön tarkoitettu Windows-käyttöjärjestelmä (vähintään XP).
 SIS Clientiä voidaan testata työasemalla, mutta kuitenkin vähintään yhdellä tabletilla, jolle se on ensisijaisesti suunniteltu.

2.7 Testausdokumentaatio

Testausdokumentaatio sisältää tämän *testaussuunnitelman* lisäksi testauksessa täytettävän *testauspöytäkirjan* sekä sen pohjalta erikseen laadittavan *testausraportin*.

2.8 Vastuut

Testaajina toimivat OAMK:n tekniikan yksikön opiskelijat Juha Nord, Pirkko Kukkoyle ja Marko Vesala. Jokainen testaa molemmat yksiköt, joten kumpaakin yksikköä testataan kolmeen eri otteeseen.

2.9 Aikataulu

Testaus suoritetaan yhdellä istumalla viikon 22 alussa. Raportointi tapahtuu saman viikon loppuun mennessä.

OAMK HYVINVOINTITEKNOLOGIA		Testaussuunnitelma		8 (34)
Prepared ANTTI VÄNSKÄ	File.doc liite5.testaussuunnitelma.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

3 TESTITAPAUKSET: SIS Client

3.1 Asennus

Testattavat ominaisuudet	SIS Clientin asentaminen Windows-työasemalle/-tabletille.
Tarkempi kuvaus	Käynnistetään SIS Client v1.1.2 -paketin sisältämä 'setup.exe' ja noudatetaan asennusohjeita. Asennuksen jälkeen käynnistetään asennuskansiosta löytyvä 'SIS Client.exe'.
Mitä testataan	Asentuuko SIS Client ilman ongelmia? Käynnistyykö asennettu ohjelma normaalisti?
Läpikäytkriteerit	SIS Clientin asennus onnistuu ilman virheilmoituksia ja ohjelman käynnistys onnistuu normaalisti.

3.2 Tilan valitseminen

Testattavat ominaisuudet	SIS Clientin tilan valinta.
Tarkempi kuvaus	Valitaan 'SIS Client Login' -ikkunan alavetovalikosta satunnainen tila ja painetaan OK-painiketta.
Mitä testataan	Onnistuuko tilan valitseminen normaalisti ja näkyykö valittu tila pääohjelman 'Current location'-kentässä?
Läpikäytkriteerit	Tila valitaan onnistuneesti ilman virheilmoituksia.

OAMK HYVINVOINTITEKNOLOGIA		Testaussuunnitelma		9 (34)
Prepared ANTTI VÄNSKÄ	File.doc liite5.testaussuunnitelma.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

3.3 Tilaan liittyvien tuotteiden listaaminen

Testattavat ominaisuudet	Tilassa sijaitsevien (location) ja sinne lainattujen (borrowed location) tuotteiden listaaminen.
Tarkempi kuvaus	Valitaan tilaksi joko 1) Ruokala tai 2) Hissi. Painetaan 'List'-painiketta. Varmistetaan, että ohjelma näyttää kaikki kyseiseen tilaan liittyvät tuotteet.
Mitä testataan	Löytääkö ohjelma oikean määrän kyseiseen tilaan liittyviä tuotteita? 1) Ruokala: 3 tuotetta (Tammi, Tuomi ja Saarni) 2) Hissi: 4 tuotetta (Tervaleppä, Hieskoivu, Pihlaja ja Mänty)
Läpikäytkriteerit	Kaikki valittuun tilaan liittyvät tuotteet saadaan listattua ilman virhetilanteita ja ylimääräisiä tuotteita.

3.4 Viivakoodin lukeminen

3.4.1 Lukeminen tyhjällä listalla

Testattavat ominaisuudet	Järjestelmässä olevan tuotteen viivakoodin lukeminen ikkunan ollessa tyhjänä.
Tarkempi kuvaus	Tilan valinnan jälkeen ei paineta mitään, vaan luetaan jonkin tuotteen viivakoodi.
Mitä testataan	Avautuvatko kyseisen tuotteen tiedot ohjelmaan?
Läpikäytkriteerit	Vain kyseisen tuotteen tiedot avautuvat ongelmitta näytölle eli näkyviin ilmestyy vain yksi rivi.

3.4.2 Lukeminen listauksen jälkeen

Testattavat ominaisuudet	Järjestelmässä olevan tuotteen viivakoodin lukeminen 'List'-painikkeen painamisen jälkeen.
Tarkempi kuvaus	Tilan valinnan jälkeen painetaan 'List'-painiketta, jolloin näytölle ilmestyy kyseiseen tilaan liittyvät tuotteet. Tämän jälkeen luetaan jonkin tuotteen viivakoodi.
Mitä testataan	Avautuvatko kyseisen tuotteen tiedot ohjelmaan?

OAMK HYVINVOINTITEKNOLOGIA		Testaussuunnitelma		10 (34)
Prepared ANTTI VÄNSKÄ	File.doc liite5.testaussuunnitelma.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

Läpikäsitteerit	Vain kyseisen tuotteen tiedot avautuvat ongelmitta näytölle eli näkyviin ilmestyy vain yksi rivi.
-----------------	---

3.4.3 Lukeminen toisen tuotteen ollessa valittuna

Testattavat ominaisuudet	Kahden tuotteen lukeminen peräkkäin.
Tarkempi kuvaus	Luetaan tuotteen 'Tammi' viivakoodi, jolloin sen tiedot ilmestyvät näytölle. Heti perään luetaan tuotteen 'Tervaleppä' viivakoodi.
Mitä testataan	Ilmestyvätkö tuotteiden tiedot normaalisti näytölle yksi kerrallaan?
Läpikäsitteerit	Vain ja ainoastaan juuri luetun tuotteen tiedot näkyvät näytöllä.

3.4.4 Lukeminen useaan kertaan

Testattavat ominaisuudet	Järjestelmässä olevan tuotteen viivakoodin lukeminen useaan kertaan peräkkäin.
Tarkempi kuvaus	Luetaan satunnaisen tuotteen viivakoodi kerran, jolloin sen tiedot ilmestyvät näytölle. Tämän jälkeen toistetaan lukeminen vielä 2-4 kertaa.
Mitä testataan	Eihän viivakoodin lukeminen useaan kertaan aiheuta mitään poikkeustilanteita?
Läpikäsitteerit	<ul style="list-style-type: none"> - Vain ja ainoastaan kyseisen tuotteen tiedot näkyvät näytöllä. - Ei virheilmoituksia tai muuta poikkeavaa toiminnallisuutta.

3.4.5 Puuttuvan tuotteen viivakoodin lukeminen

Testattavat ominaisuudet	Järjestelmästä puuttuvan tuotteen viivakoodin lukeminen.
Tarkempi kuvaus	Luetaan järjestelmästä poistetun tuotteen 'Bonsai' viivakoodi.
Mitä testataan	Ilmoittaako ohjelma normaalisti, ettei kyseistä tuotetta löydy järjestelmästä?
Läpikäsitteerit	Ohjelma antaa asianmukaisen virheilmoituksen.

OAMK HYVINVOINTITEKNOLOGIA		Testaussuunnitelma		11 (34)
Prepared ANTTI VÄNSKÄ	File.doc liite5.testaussuunnitelma.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

3.4.6 Väärän viivakoodin lukeminen

Testattavat ominaisuudet	Väärässä formaatissa olevan viivakoodin lukeminen.
Tarkempi kuvaus	Luetaan jokin väärässä formaatissa oleva (ei EAN-13) viivakoodi.
Mitä testataan	Antaako ohjelma virheilmoituksen väärästä viivakoodista?
Läpikäyäkriteerit	Ohjelma antaa asianmukaisen virheilmoituksen.

OAMK HYVINVOINTITEKNOLOGIA		Testaussuunnitelma		12 (34)
Prepared ANTTI VÄNSKÄ	File.doc liite5.testaussuunnitelma.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

3.5 Lainaaminen

3.5.1 Lainaaminen toiseen tilaan

Testattavat ominaisuudet	Tuotteen normaali lainaaminen paikasta toiseen.
Tarkempi kuvaus	Valitaan jokin tuote, joka ei ole lainassa. Valitaan tila, joka on eri kuin tuotteen nykyinen tila. Luetaan tuotteen viivakoodi ja painetaan 'Update'-painiketta.
Mitä testataan	Onnistuuko lainaaminen ilman ongelmia?
Läpikäsitteerit	Valitun tuotteen 'Borrowed location'-kenttään ilmestyy tilan nimi, joka näkyy ohjelman 'Current location'-kentässä.

3.5.2 Lainatun lainaaminen kolmanteen tilaan

Testattavat ominaisuudet	Lainassa olevan tuotteen lainaaminen toiseen tilaan.
Tarkempi kuvaus	Valitaan jokin tuote, joka on lainattu tilasta A tilaan B. Siirrytään ohjelmalla tilaan C, luetaan viivakoodi ja painetaan 'Update'-painiketta.
Mitä testataan	Onnistuuko lainaaminen ilman ongelmia?
Läpikäsitteerit	Valitun tuotteen 'Borrowed location'-kenttään vaihtuu tila, joka näkyy ohjelman 'Current location'-kentässä.

3.5.3 Lainaaminen useaan kertaan

Testattavat ominaisuudet	Yritetään lainata tuotetta useita kertoja samaan paikkaan.
Tarkempi kuvaus	Valitaan jokin tuote, joka ei ole lainassa. Valitaan tila, joka on eri kuin tuotteen nykyinen tila. Luetaan tuotteen viivakoodi ja painetaan 'Update'-painiketta. Painetaan 'Update'-painiketta uudestaan useita kertoja välillä myös lukien viivakoodi uudestaan.
Mitä testataan	Eihän uudelleen lainaaminen aiheuta mitään ongelmia?
Läpikäsitteerit	<ul style="list-style-type: none"> - 'Borrowed location'-kenttä pysyy muuttumattomana. - Ei virheilmoituksia tai poikkeavaa toiminnallisuutta.

OAMK HYVINVOINTITEKNOLOGIA		Testaussuunnitelma		13 (34)
Prepared ANTTI VÄNSKÄ	File.doc liite5.testaussuunnitelma.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

3.6 Palauttaminen

3.6.1 Lainatun palauttaminen alkuperäiseen tilaan

Testattavat ominaisuudet	Lainatun tuotteen normaali palauttaminen sen alkuperäiseen tilaan.
Tarkempi kuvaus	Valitaan jokin tuote, joka on lainassa paikasta A paikkaan B. Siirrytään tilaan A, luetaan tuotteen viivakoodi ja painetaan 'Update'-painiketta. Tuotteen pitäisi palautua paikkaan A eli kentän 'Borrowed location' tulisi tyhjentyä.
Mitä testataan	Onnistuuko palauttaminen ilman ongelmia?
Läpikäyäkriteerit	<ul style="list-style-type: none"> - 'Borrowed location' kenttä tyhjenee. - Ei virheilmoituksia tai poikkeavaa toiminnallisuutta.

3.6.2 Palautetun palauttaminen uudestaan

Testattavat ominaisuudet	Lainatun tuotteen palauttaminen useammin kuin kerran.
Tarkempi kuvaus	Valitaan jokin tuote, joka on lainassa paikasta A paikkaan B. Siirrytään tilaan A, luetaan tuotteen viivakoodi ja painetaan 'Update'-painiketta. Painetaan 'Update'-painiketta uudestaan useita kertoja välillä myös lukien viivakoodi uudestaan.
Mitä testataan	Eihän uudelleen palauttaminen aiheuta mitään ongelmia?
Läpikäyäkriteerit	<ul style="list-style-type: none"> - 'Borrowed location'-kenttä tyhjenee normaalisti. - 'Location'-kenttä pysyy muuttumattomana. - Ei virheilmoituksia tai poikkeavaa toiminnallisuutta.

OAMK HYVINVOINTITEKNOLOGIA		Testaussuunnitelma		14 (34)
Prepared ANTTI VÄNSKÄ	File.doc liite5.testaussuunnitelma.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

4 TESTITAPAUKSET: SIS Management

4.1 Asennus

Testattavat ominaisuudet	SIS Managementin asentaminen Windows-työasemalle.
Tarkempi kuvaus	<p>Avataan linkki http://welcome.solutions.brother.com/bsc/public/eu/fi/fi/dlf/download_top.html?reg=eu&c=fi&lang=fi&prod=lpql550euk (asennuspaketissa linkki: Brother QL-500 drivers) ja ladataan oikealle käyttöjärjestelmälle sopivat tulostimen ajurit ja asennetaan ne. Asennuksen jälkeen voi olla tarpeellista varmistaa, että Windowsin 'Devices and Printers'-valikosta löytyy 'Brother QL-500'-tulostin ja se viittaa nimenomaan fyysiseen tulostimeen.</p> <p>Käynnistetään sen jälkeen SIS Management v1.3.1 -paketin sisältämä 'setup.exe' ja noudatetaan asennusohjeita. Asennuksen jälkeen käynnistetään asennuskansiosta löytyvä 'SIS Management.exe'.</p>
Mitä testataan	Asentuuko SIS Management ilman ongelmia? Käynnistyykö asennettu ohjelma normaalisti?
Läpikäytkriteerit	<ul style="list-style-type: none"> - SIS Managementin asennus onnistuu ilman virheilmoituksia. - Ohjelman käynnistys onnistuu normaalisti.

OAMK HYVINVOINTITEKNOLOGIA		Testaussuunnitelma		15 (34)
Prepared ANTTI VÄNSKÄ	File.doc liite5.testaussuunnitelma.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

4.2 Kirjautuminen

4.2.1 Kirjautuminen väärillä tunnuksilla

Testattavat ominaisuudet	SIS Managementin vaste väärillä tunnuksilla kirjautumiseen.
Tarkempi kuvaus	Käynnistetään ohjelma. Yritetään kirjautua järjestelmään sattumanvaraisilla tunnuksilla/salasanoina.
Mitä testataan	Antaako ohjelma oikean virheilmoituksen ja estää etenemisen?
Läpikäytkriteerit	<ul style="list-style-type: none"> - Virheilmoitus väärillä tunnuksilla kirjautumisesta. - Ohjelma jatkaa toimintaansa samassa tilassa.

4.2.2 Kirjautuminen oikeilla tunnuksilla

Testattavat ominaisuudet	Kirjautuminen järjestelmään.
Tarkempi kuvaus	Käynnistetään ohjelma. Kirjaututaan järjestelmään oikeilla tunnuksilla.
Mitä testataan	Päästääkö SIS Management etenemään ohjelman pääikkunaan ilman virheilmoituksia?
Läpikäytkriteerit	Kirjautuminen onnistuu ilman virheilmoituksia.

OAMK HYVINVOINTITEKNOLOGIA		Testaussuunnitelma		16 (34)
Prepared ANTTI VÄNSKÄ	File.doc liite5.testaussuunnitelma.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

4.3 Viivakoodin lukeminen

4.3.1 Normaali lukeminen

Testattavat ominaisuudet	Viivakoodin lukeminen ohjelman perustilassa.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Luetaan jonkin järjestelmästä löytyvän tuotteen viivakoodi.
Mitä testataan	Aukeaako kyseinen tuote 'Edit Item'-tilaan?
Läpikäytkriteerit	<ul style="list-style-type: none"> - 'Edit Item'-ikkuna aukeaa. - Kyseisen tuotteen tiedot ovat esillä. - Ei virheilmoituksia.

4.3.2 Lukeminen toisen tuotteen ollessa valittuna

Testattavat ominaisuudet	Viivakoodin lukeminen toisen tuotteen ollessa valittuna.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Aukaistaan tuote A 'Edit Item'-tilaan joko tuplaklikkaamalla sitä listalta tai lukemalla sen viivakoodi. Sen jälkeen luetaan tuotteen B viivakoodi.
Mitä testataan	Antaako ohjelma virheilmoituksen tai avaaako se tuotteen B 'Edit Item'-tilaan?
Läpikäytkriteerit	<ul style="list-style-type: none"> - Ohjelma antaa virheilmoituksen. TAI - Tuote B avautuu 'Edit Item'-tilaan. TAI - Mitään ei tapahdu.

4.3.3 Lukeminen useaan kertaan

Testattavat ominaisuudet	Viivakoodin lukeminen useaan kertaan peräkkäin.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Luetaan jonkin tuotteen viivakoodi, jolloin 'Edit Item'-ikkuna avautuu. Luetaan heti perään sama viivakoodi uudestaan.
Mitä testataan	Antaako ohjelma virheilmoituksen vai tekeekö se mitään?
Läpikäytkriteerit	<ul style="list-style-type: none"> - Ohjelma antaa virheilmoituksen. TAI - Mitään ei tapahdu.

OAMK HYVINVOINTITEKNOLOGIA		Testaussuunnitelma		17 (34)
Prepared ANTTI VÄNSKÄ	File.doc liite5.testaussuunnitelma.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

4.3.4 Puuttuvan tuotteen viivakoodin lukeminen

Testattavat ominaisuudet	Järjestelmästä puuttuvan viivakoodin lukeminen.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Luetaan sellainen viivakoodi, jota vastaavaa tuotetta ei ole lisätty järjestelmään. Esimerkiksi järjestelmästä poistettu tuote 'Bonsai'.
Mitä testataan	Yrittääkö ohjelma lisätä kyseistä tuotetta järjestelmään vai antaa se virheilmoituksen?
Läpipääsykriteerit	<ul style="list-style-type: none"> - 'New Item'-ikkuna avautuu sisältäen pelkästään kyseisen viivakoodin numeerisessa muodossa. TAI - Ohjelma antaa virheilmoituksen.

4.3.5 Väärän viivakoodin lukeminen

Testattavat ominaisuudet	Väärässä formaatissa olevan viivakoodin lukeminen.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Luetaan sellainen viivakoodi, joka ei ole EAN-13-standardin mukainen.
Mitä testataan	Antaako ohjelma virheilmoituksen väärästä viivakoodista?
Läpipääsykriteerit	Ohjelma antaa asianmukaisen virheilmoituksen.

OAMK HYVINVOINTITEKNOLOGIA		Testaussuunnitelma		18 (34)
Prepared ANTTI VÄNSKÄ	File.doc liite5.testaussuunnitelma.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

4.4 Uuden tuotteen lisääminen

4.4.1 Puutteellisilla tiedoilla

Testattavat ominaisuudet	Uuden tuotteen lisääminen puutteellisilla tiedoilla.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Painetaan 'New Item'-painiketta. Yritetään lisätä uusi tuote, jonka tiedot ovat puutteelliset. Ainoastaan 'Item Name' ja 'Location' -kenttien kuuluisi olla pakolliset.
Mitä testataan	Estääkö ohjelma puutteellisilla tiedoilla varustetun tuotteen lisäämisen?
Läpipääsykriteerit	Ohjelma antaa virheilmoituksen, jossa se kertoo, mikä tieto ei ole kunnossa.

4.4.2 Pitkillä ja erikoismerkkejä sisältävillä kentillä

Testattavat ominaisuudet	Uuden tuotteen lisääminen pitkiä merkkijonoja ja erikoismerkkejä sisältävillä kentillä.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Painetaan 'New Item'-painiketta. Yritetään lisätä tuote, jonka tietoihin on lisätty huomattavan pitkiä ja erikoismerkkejä sisältäviä merkkijonoja.
Mitä testataan	Selviääkö ohjelma mahdollisesti haastavista merkkijonoista?
Läpipääsykriteerit	<ul style="list-style-type: none"> - Ohjelma ei kaadu. - Ohjelma joko hyväksyy kaikki syötteet, antaa niistä virheilmoituksen tai leikkaa liian pitkän merkkijonon sopivan mittaiseksi.

OAMK HYVINVOINTITEKNOLOGIA		Testaussuunnitelma		19 (34)
Prepared ANTTI VÄNSKÄ	File.doc liite5.testaussuunnitelma.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

4.5 Tuotteen poistaminen

4.5.1 Tuotteen muokkaustilassa

Testattavat ominaisuudet	Tuotteen poistaminen tuotteen muokkaustilassa.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Painetaan 'New Item'-painiketta, luodaan uusi tuote ja tulostetaan siitä tarra (1 kpl riittää, tarran tiedoilla ei väliä). Tuplaklikataan kyseistä tuotetta listalta, jolloin 'Edit Item'-ikkuna avautuu. Painetaan 'Delete'-painiketta ja hyväksytään tuotteen poistaminen. Yritetään myös lukea poistetun tuotteen viivakoodi.
Mitä testataan	Onnistuuko tuotteen poistaminen ilman ongelmia ja katoaako se myös listalta? Tulkitseeko ohjelma poistetun tuotteen viivakoodin uudeksi tuotteeksi (eli avaa 'New Item'-ikkunan)?
Läpipääsykriteerit	<ul style="list-style-type: none"> - Poistaminen onnistuu ilman ongelmia. - Tuote katoaa listalta. - Poistetun tuotteen viivakoodin lukeminen avaa 'New Item'-ikkunan.

4.5.2 Tuotteen ollessa valittuna

Testattavat ominaisuudet	Tuotteen poistaminen suoraan listalta.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Painetaan 'New Item'-painiketta ja luodaan uusi tuote (ei tarvitse tulostaa tarraa). Sitten aktivoidaan kyseinen tuote listalta painamalla hiiren 1-näppäintä kerran sen kohdalla. Painetaan 'Delete'-näppäintä näppäimistöltä ja hyväksytään poistaminen.
Mitä testataan	Onnistuuko tuotteen poistaminen ilman ongelmia ja katoaako se myös listalta?
Läpipääsykriteerit	<ul style="list-style-type: none"> - Poistaminen onnistuu ilman ongelmia. - Tuote katoaa listalta.

OAMK HYVINVOINTITEKNOLOGIA		Testaussuunnitelma		20 (34)
Prepared ANTTI VÄNSKÄ	File.doc liite5.testaussuunnitelma.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

4.6 Kategorian muokkaaminen

4.6.1 Uuden kategorian luominen

Testattavat ominaisuudet	Uuden kategorian luominen.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Painetaan 'Edit Categories and Locations'-painiketta. Kirjoitetaan 'Categories'-sarakkeen alta löytyvään kenttään uuden kategorian nimi 'Kaapelit'. Painetaan sen jälkeen 'Add'-painiketta.
Mitä testataan	Onnistuuko kategorian lisääminen ongelmitta?
Läpipääsykriteerit	<ul style="list-style-type: none"> - Ei virheilmoituksia. - Uusi kategoria 'Kaapelit' löytyy filttereiden 'Category'-kentästä.

4.6.2 Uuden kategorian luominen olemassa olevan kategorian nimellä

Testattavat ominaisuudet	Uuden kategorian luominen samalla nimellä kuin jo olemassa oleva kategoria.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Painetaan 'Edit Categories and Locations'-painiketta. Edellisessä testitapauksessa lisättiin uusi kategoria 'Kaapelit'. Varmistetaan, että se löytyy edelleen listalta ja yritetään lisätä toinen samanniminen kategoria.
Mitä testataan	Estääkö ohjelma samannimisen kategorian luomisen?
Läpipääsykriteerit	Ohjelma estää kategorian luomisen ja antaa virheilmoituksen.

4.6.3 Uudelleennimeäminen

Testattavat ominaisuudet	Kategorian uudelleennimeäminen.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Painetaan 'Edit Categories and Locations'-painiketta. Valitaan aiemmin lisätty 'Kaapelit'-kategoria liukuvalikosta ja kirjoitetaan sen päälle uusi nimi 'Kytkimet'. Painetaan 'Rename'-painiketta.
Mitä testataan	Onnistuuko kategorian uudelleennimeäminen ongelmitta?
Läpipääsykriteerit	<ul style="list-style-type: none"> - Ei virheilmoituksia.

OAMK HYVINVOINTITEKNOLOGIA		Testaussuunnitelma		21 (34)
Prepared ANTTI VÄNSKÄ	File.doc liite5.testaussuunnitelma.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

	- Katgoria 'Kaapelit' on kadonnut listalta ja sen tilalle on tullut 'Kytkimet'.
--	---

4.6.4 Uudelleennimeäminen toisen kategorian päälle

Testattavat ominaisuudet	Kategorian nimeäminen uudelleen samalla nimellä kuin jo olemassa oleva kategoria.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Painetaan 'Edit Categories and Locations'-painiketta. Luodaan uusi kategoria nimeltä 'Muuntajat'. Valitaan sen jälkeen liukuvalikosta tämä 'Muuntajat' ja kirjoitetaan sen päälle nimi 'Kytkimet', jonka niminen kategoria löytyy jo listalta. Painetaan 'Rename'-painiketta.
Mitä testataan	Estääkö ohjelma uudelleennimeämisen toisen kategorian päälle?
Läpikäsykriteerit	<ul style="list-style-type: none"> - Ohjelma estää kategorian uudelleennimeämisen ja antaa virheilmoituksen. - Listalta löytyy edelleen yksi 'Muuntajat'- ja yksi 'Kytkimet'-kategoria.

4.6.5 Käytössä olevan kategorian poistaminen

Testattavat ominaisuudet	Käytössä olevan kategorian poistaminen, kun jokin tuote on luokiteltu kyseiseen kategoriaan.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Luodaan uusi kategoria nimeltä 'Komponentit'. Luodaan uusi tuote nimeltä 'Transistori', joka kuuluu kategoriaan 'Komponentit' (ei tulosteta tarraa). Sitten painetaan jälleen 'Edit Categories and Locations'-painiketta, valitaan liukuvalikosta kategoria 'Komponentit' ja painetaan 'Delete'-painiketta.
Mitä testataan	Estääkö ohjelma käytössä olevan kategorian poistamisen?
Läpikäsykriteerit	<ul style="list-style-type: none"> - Ohjelma estää kategorian poistamisen ja antaa virheilmoituksen. - 'Komponentit'-kategoria löytyy edelleen listalta.

OAMK HYVINVOINTITEKNOLOGIA		Testaussuunnitelma		22 (34)
Prepared ANTTI VÄNSKÄ	File.doc liite5.testaussuunnitelma.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

4.6.6 Vapaan kategorian poistaminen

Testattavat ominaisuudet	Kategorian poistaminen, kun mikään tuote ei ole luokiteltu kyseiseen kategoriaan.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Vaihdetaan tuotteen 'Transistori' kategoriaksi jokin muu kuin 'Komponentit'. Sen jälkeen painetaan 'Edit Categories and Locations'-painiketta, valitaan liukuvalikosta kategoria 'Komponentit' ja painetaan 'Delete'-painiketta.
Mitä testataan	Onnistuuko vapaan kategorian poistaminen ongelmitta?
Läpikäyäkriteerit	<ul style="list-style-type: none"> - Ei virheilmoituksia. - Kategoria 'Komponentit' katoaa listalta.

OAMK HYVINVOINTITEKNOLOGIA		Testaussuunnitelma		23 (34)
Prepared ANTTI VÄNSKÄ	File.doc liite5.testaussuunnitelma.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

4.7 Tilan muokkaaminen

4.7.1 Uuden tilan luominen

Testattavat ominaisuudet	Uuden tilan luominen.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Painetaan 'Edit Categories and Locations'-painiketta. Kirjoitetaan 'Locations'-sarakkeen alta löytyvään kenttään uuden tilan nimi 'Ullakko'. Painetaan sen jälkeen 'Add'-painiketta.
Mitä testataan	Onnistuuko tilan lisääminen ongelmitta?
Läpipääsykriteerit	<ul style="list-style-type: none"> - Ei virheilmoituksia. - Uusi tila 'Ullakko' löytyy filttereiden 'Original location'- ja 'Borrowed location'-kentistä.

4.7.2 Uuden tilan luominen olemassa olevan tilan nimellä

Testattavat ominaisuudet	Uuden tilan luominen samalla nimellä kuin jo olemassa oleva tila.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Painetaan 'Edit Categories and Locations'-painiketta. Edellisessä testitapauksessa lisättiin uusi tila 'Ullakko'. Varmistetaan, että se löytyy edelleen listalta ja yritetään lisätä toinen samanniminen tila.
Mitä testataan	Estääkö ohjelma samannimisen tilan luomisen?
Läpipääsykriteerit	Ohjelma estää tilan luomisen ja antaa virheilmoituksen.

4.7.3 Uudelleennimeäminen

Testattavat ominaisuudet	Tilan uudelleennimeäminen.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Painetaan 'Edit Categories and Locations'-painiketta. Valitaan aiemmin lisätty 'Ullakko'-tila liukuvalikosta ja kirjoitetaan sen päälle uusi nimi 'Kahvio'. Painetaan 'Rename'-painiketta.
Mitä testataan	Onnistuuko tilan uudelleennimeäminen ongelmitta?
Läpipääsykriteerit	<ul style="list-style-type: none"> - Ei virheilmoituksia. - Tila 'Ullakko' on kadonnut listalta ja sen tilalle on tullut 'Kahvio'.

OAMK HYVINVOINTITEKNOLOGIA		Testaussuunnitelma		24 (34)
Prepared ANTTI VÄNSKÄ	File.doc liite5.testaussuunnitelma.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

4.7.4 Uudelleennimeäminen toisen tilan päälle

Testattavat ominaisuudet	Tilan nimeäminen uudelleen samalla nimellä kuin jo olemassa oleva tila.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Painetaan 'Edit Categories and Locations'-painiketta. Luodaan uusi tila nimeltä 'Navetta'. Valitaan sen jälkeen liukuvalikosta tämä 'Navetta' ja kirjoitetaan sen päälle nimi 'Kahvio', jonka niminen tila löytyy jo listalta. Painetaan 'Rename'-painiketta.
Mitä testataan	Estääkö ohjelma uudelleennimeämisen toisen tilan päälle?
Läpikäytkriteerit	<ul style="list-style-type: none"> - Ohjelma estää tilan uudelleennimeämisen ja antaa virheilmoituksen. - Listalta löytyy edelleen yksi 'Navetta'- ja yksi 'Kahvio'-tila.

4.7.5 Käytössä olevan tilan (original location) poistaminen

Testattavat ominaisuudet	Käytössä olevan tilan poistaminen, kun jokin tuote sijaitsee siellä.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Luodaan uusi tila nimeltä 'Siivouskomero'. Luodaan uusi tuote nimeltä 'Pölynimuri', jota säilytetään ('Location'-kenttä) tilassa 'Siivouskomero' (ei tulosteta tarraa). Sitten painetaan jälleen 'Edit Categories and Locations'-painiketta, valitaan liukuvalikosta tila 'Siivouskomero' ja painetaan 'Delete'-painiketta.
Mitä testataan	Estääkö ohjelma käytössä olevan tilan poistamisen?
Läpikäytkriteerit	<ul style="list-style-type: none"> - Ohjelma estää tilan poistamisen ja antaa virheilmoituksen. - 'Siivouskomero'-tila löytyy edelleen listalta.

4.7.6 Käytössä olevan tilan (borrowed location) poistaminen

Testattavat ominaisuudet	Käytössä olevan tilan poistaminen, kun jokin tuote on siellä lainassa.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Mikäli tila 'Siivouskomero' on poistettu, luodaan se uudestaan. Muokataan tuotetta 'Pölynimuri' niin, että sitä säilytetään ('Location'-kenttä)

OAMK HYVINVOINTITEKNOLOGIA		Testaussuunnitelma		25 (34)
Prepared ANTTI VÄNSKÄ	File.doc liite5.testaussuunnitelma.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

	esimerkiksi tilassa 'Kahvio' ja se on lainassa ('Borrowed location'-kenttä) tilassa 'Siivouskomero'. Sitten painetaan 'Edit Categories and Locations'-painiketta, valitaan liukuvalikosta tila 'Siivouskomero' ja painetaan 'Delete'-painiketta.
Mitä testataan	Estääkö ohjelma käytössä olevan tilan poistamisen?
Läpikäyskriteerit	<ul style="list-style-type: none"> - Ohjelma estää tilan poistamisen ja antaa virheilmoituksen. - 'Siivouskomero'-tila löytyy edelleen listalta.

4.7.7 Vapaan tilan poistaminen

Testattavat ominaisuudet	Vapaan eli tuotteista tyhjennetyt tilan poistaminen.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Poistetaan tuote 'Pölynimuri'. Sitten painetaan 'Edit Categories and Locations'-painiketta, valitaan liukuvalikosta tila 'Siivouskomero' ja painetaan 'Delete'-painiketta.
Mitä testataan	Onnistuuko vapaan tilan poistaminen ongelmitta?
Läpikäyskriteerit	<ul style="list-style-type: none"> - Ei virheilmoituksia. - Tila 'Siivouskomero' katoaa listalta.

OAMK HYVINVOINTITEKNOLOGIA		Testaussuunnitelma		26 (34)
Prepared ANTTI VÄNSKÄ	File.doc liite5.testaussuunnitelma.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

4.8 Tuotteen muokkaaminen

4.8.1 Puutteellisilla tiedoilla

Testattavat ominaisuudet	Tuotteen muokkaaminen puutteellisilla tiedoilla.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Luodaan uusi tuote 'Helikopteri'. Täytetään kaikki kentät sattumanvaraisella datalla (ei tarvitse tulostaa tarraa). Kun tuote on luotu, avataan se muokkaustilaan ('Edit Item'-ikkuna) tuplaklikkaamalla sitä listalta. Yritetään tyhjentää eri kenttiä ja tallentaa tiedot. 'Item Name' ja 'Original Location' -kenttien tallentamisen tyhjänä ei pitäisi onnistua.
Mitä testataan	Estääkö ohjelma aina tuotteen tallentamisen, jos vaaditut kentät eivät sisällä dataa?
Läpikäyäkriteerit	<ul style="list-style-type: none"> - Ohjelma antaa virheilmoituksen, jossa se kertoo, mikä tieto ei ole kunnossa. - Tietojen tallentaminen ei onnistu, jos joko 'Item Name' tai 'Original Location' on tyhjä.

4.8.2 Pitkillä ja erikoismerkkejä sisältävillä kentillä

Testattavat ominaisuudet	Tuotteen muokkaaminen pitkiä merkkijonoja ja erikoismerkkejä sisältävillä kentillä.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Tuplaklikataan tuotetta 'Helikopteri'. Yritetään lisätä tuotteen tietoihin huomattavan pitkiä ja erikoismerkkejä sisältäviä merkkijonoja.
Mitä testataan	Selviääkö ohjelma mahdollisesti haastavista merkkijonoista?
Läpikäyäkriteerit	<ul style="list-style-type: none"> - Ohjelma ei kaadu. - Ohjelma joko hyväksyy kaikki syötteen, antaa niistä virheilmoituksen tai leikkaa liian pitkän merkkijonon sopivan mittaiseksi.

4.8.3 Lainaaminen alkuperäiseen tilaan

Testattavat ominaisuudet	Tuotteen lainaaminen sen alkuperäiseen tilaan.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Tuplaklikataan tuotetta 'Helikopteri'. Valitaan kentän 'Borrowed Location' arvoksi

OAMK HYVINVOINTITEKNOLOGIA		Testaussuunnitelma		27 (34)
Prepared ANTTI VÄNSKÄ	File.doc liite5.testaussuunnitelma.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

	sama kuin 'Original Location'. Tuote olisi siis merkitty lainatuksi samaan tilaan, josta se on peräisin, minkä ei pitäisi olla mahdollista.
Mitä testataan	Estääkö ohjelma tuotteen lainaamisen alkuperäiseen tilaansa?
Läpikäsykriteerit	<ul style="list-style-type: none"> - Ohjelma ei tallenna valittua tilaa 'borrowed location'-kenttään. TAI <ul style="list-style-type: none"> - Ohjelma antaa virheilmoituksen.

OAMK HYVINVOINTITEKNOLOGIA		Testaussuunnitelma		28 (34)
Prepared ANTTI VÄNSKÄ	File.doc liite5.testaussuunnitelma.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

4.9 Tuotteen tulostaminen

Tuotteen viivakoodin ja lisätietojen tulostamista ei tarvitse toistaa viittä (5) kertaa, kuten muissa testitapauksissa.

4.9.1 Pelkällä viivakoodilla

Testattavat ominaisuudet	Tuotteen pelkän viivakoodin tulostaminen tarratulostimella.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Tuplaklikataan tuotetta 'Pihlaja'. Painetaan 'Print'-painiketta. Tyhjennetään kaikki valintaruudut (checkbox) ja painetaan 'Print'-painiketta. Yritetään myös lukea tulostettu viivakoodi.
Mitä testataan	Onnistuuko tuotteen tulostaminen pelkällä viivakoodilla? Entä onnistuuko tulostetun viivakoodin lukeminen?
Läpipääsykriteerit	<ul style="list-style-type: none"> - Ei virheilmoituksia. - Tulostus onnistuu ja tuloste on käyttökelpoinen. - Viivakoodin lukeminen avaa tuotteen 'Edit Item'-ikkunaan.

4.9.2 Name

Testattavat ominaisuudet	Tuotteen viivakoodin ja nimen tulostaminen tarratulostimella.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Tuplaklikataan tuotetta 'Pihlaja'. Painetaan 'Print'-painiketta. Valitaan ainoastaan valintaruutu 'Name' ja tyhjennetään muut. Painetaan 'Print'-painiketta. Yritetään myös lukea tulostettu viivakoodi.
Mitä testataan	Onnistuuko tuotteen tulostaminen viivakoodilla ja nimellä varustettuna? Entä onnistuuko tulostetun viivakoodin lukeminen?
Läpipääsykriteerit	<ul style="list-style-type: none"> - Ei virheilmoituksia. - Tulostus onnistuu ja tuloste on käyttökelpoinen. - Viivakoodin lukeminen avaa tuotteen 'Edit Item'-ikkunaan.

OAMK HYVINVOINTITEKNOLOGIA		Testaussuunnitelma		29 (34)
Prepared ANTTI VÄNSKÄ	File.doc liite5.testaussuunnitelma.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

4.9.3 Name + Category

Testattavat ominaisuudet	Tuotteen viivakoodin, nimen ja kategorian tulostaminen tarratulostimella.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Tuplaklikataan tuotetta 'Pihlaja'. Painetaan 'Print'-painiketta. Valitaan valintaruudut 'Name' ja 'Category', tyhjennetään 'Description'. Painetaan 'Print'-painiketta. Yritetään myös lukea tulostettu viivakoodi.
Mitä testataan	Onnistuuko tuotteen tulostaminen viivakoodilla, nimellä ja kategorialla varustettuna? Entä onnistuuko tulostetun viivakoodin lukeminen?
Läpipääsykriteerit	<ul style="list-style-type: none"> - Ei virheilmoituksia. - Tulostus onnistuu ja tuloste on käyttökelpoinen. - Viivakoodin lukeminen avaa tuotteen 'Edit Item'-ikkunaan.

4.9.4 Name + Description

Testattavat ominaisuudet	Tuotteen viivakoodin, nimen ja kuvauksen tulostaminen tarratulostimella.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Tuplaklikataan tuotetta 'Pihlaja'. Painetaan 'Print'-painiketta. Valitaan valintaruudut 'Name' ja 'Description', tyhjennetään 'Category'. Painetaan 'Print'-painiketta. Yritetään myös lukea tulostettu viivakoodi.
Mitä testataan	Onnistuuko tuotteen tulostaminen viivakoodilla, nimellä ja kuvauksella varustettuna? Entä onnistuuko tulostetun viivakoodin lukeminen?
Läpipääsykriteerit	<ul style="list-style-type: none"> - Ei virheilmoituksia. - Tulostus onnistuu ja tuloste on käyttökelpoinen. - Viivakoodin lukeminen avaa tuotteen 'Edit Item'-ikkunaan.

4.9.5 Name + Category + Description

Testattavat ominaisuudet	Tuotteen viivakoodin ja kaikkien lisätietojen tulostaminen tarratulostimella.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Tuplaklikataan tuotetta 'Pihlaja'.

OAMK HYVINVOINTITEKNOLOGIA		Testaussuunnitelma		30 (34)
Prepared ANTTI VÄNSKÄ	File.doc liite5.testaussuunnitelma.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

	Painetaan 'Print'-painiketta. Valitaan kaikki valintaruudut. Painetaan 'Print'-painiketta. Yritetään myös lukea tulostettu viivakoodi.
Mitä testataan	Onnistuuko tuotteen tulostaminen viivakoodilla ja kaikilla lisätiedoilla varustettuna? Entä onnistuuko tulostetun viivakoodin lukeminen?
Läpikäytkriteerit	<ul style="list-style-type: none"> - Ei virheilmoituksia. - Tulostus onnistuu ja tuloste on käyttökelpoinen. - Viivakoodin lukeminen avaa tuotteen 'Edit Item'-ikkunaan.

4.9.6 Category

Testattavat ominaisuudet	Tuotteen viivakoodin ja kategorian tulostaminen tarratulostimella.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Tuplaklikataan tuotetta 'Pihlaja'. Painetaan 'Print'-painiketta. Valitaan valintaruutu 'Category' ja tyhjennetään muut. Painetaan 'Print'-painiketta. Yritetään myös lukea tulostettu viivakoodi.
Mitä testataan	Onnistuuko tuotteen tulostaminen viivakoodilla ja kategorialla varustettuna? Entä onnistuuko tulostetun viivakoodin lukeminen?
Läpikäytkriteerit	<ul style="list-style-type: none"> - Ei virheilmoituksia. - Tulostus onnistuu ja tuloste on käyttökelpoinen. - Viivakoodin lukeminen avaa tuotteen 'Edit Item'-ikkunaan.

4.9.7 Category + Description

Testattavat ominaisuudet	Tuotteen viivakoodin, kategorian ja kuvauksen tulostaminen tarratulostimella.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Tuplaklikataan tuotetta 'Pihlaja'. Painetaan 'Print'-painiketta. Valitaan valintaruudut 'Category' ja 'Description', tyhjennetään 'Name'. Painetaan 'Print'-painiketta. Yritetään myös lukea tulostettu viivakoodi.
Mitä testataan	Onnistuuko tuotteen tulostaminen viivakoodilla, kategorialla

OAMK HYVINVOINTITEKNOLOGIA		Testaussuunnitelma		31 (34)
Prepared ANTTI VÄNSKÄ	File.doc liite5.testaussuunnitelma.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

	ja kuvauksella varustettuna? Entä onnistuuko tulostetun viivakoodin lukeminen?
Läpikäytkriteerit	<ul style="list-style-type: none"> - Ei virheilmoituksia. - Tulostus onnistuu ja tuloste on käyttökelpoinen. - Viivakoodin lukeminen avaa tuotteen 'Edit Item'-ikkunaan.

4.9.8 Description

Testattavat ominaisuudet	Tuotteen viivakoodin ja kuvauksen tulostaminen tarratulostimella.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Tuplaklikataan tuotetta 'Pihlaja'. Painetaan 'Print'-painiketta. Valitaan valintaruutu 'Description' ja tyhjennetään muut. Painetaan 'Print'-painiketta. Yritetään myös lukea tulostettu viivakoodi.
Mitä testataan	Onnistuuko tuotteen tulostaminen viivakoodilla ja kuvauksella varustettuna? Entä onnistuuko tulostetun viivakoodin lukeminen?
Läpikäytkriteerit	<ul style="list-style-type: none"> - Ei virheilmoituksia. - Tulostus onnistuu ja tuloste on käyttökelpoinen. - Viivakoodin lukeminen avaa tuotteen 'Edit Item'-ikkunaan.

OAMK HYVINVOINTITEKNOLOGIA		Testaussuunnitelma		32 (34)
Prepared ANTTI VÄNSKÄ	File.doc liite5.testaussuunnitelma.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

4.10 Filttereiden käyttäminen

4.10.1 Original location

Testattavat ominaisuudet	'Original location'-filtterin toiminta.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Testataan 'Original location'-filtterin toimintaa valitsemalla sen liukuvalikosta eri tiloja. Listalla tulisi näkyä ainoastaan tuotteet, joiden 'original location'-sarakkeen arvo on sama kuin valittu tila. Testataan myös valitsemalla samasta liukuvalikosta tila 'Varasto (9999)'. Listalla tulisi näkyä tasan kuusi (6) tuotetta. Huomioi, että filtrit nollautuvat, kun focus siirtyy pois pääikkunasta.
Mitä testataan	Toimiiko 'Original location'-filtteri oikein?
Läpikäsitteerit	<ul style="list-style-type: none"> - Ei virheilmoituksia. - Filtteri suodattaa listan sisältöä oikein.

4.10.2 Borrowed location

Testattavat ominaisuudet	'Borrowed location'-filtterin toiminta.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Testataan 'Borrowed location'-filtterin toimintaa valitsemalla sen liukuvalikosta eri tiloja. Listalla tulisi näkyä ainoastaan tuotteet, joiden 'borrowed location'-sarakkeen arvo on sama kuin valittu tila. Testataan myös valitsemalla samasta liukuvalikosta tila 'Varasto (9999)'. Listalla tulisi näkyä tasan kaksi (2) tuotetta. Huomioi, että filtrit nollautuvat, kun focus siirtyy pois pääikkunasta.
Mitä testataan	Toimiiko 'Borrowed location'-filtteri oikein?
Läpikäsitteerit	<ul style="list-style-type: none"> - Ei virheilmoituksia. - Filtteri suodattaa listan sisältöä oikein.

4.10.3 Category

Testattavat ominaisuudet	'Category'-filtterin toiminta.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Testataan 'Category'-filtterin toimintaa valitsemalla sen liukuvalikosta eri kategorioita.

OAMK HYVINVOINTITEKNOLOGIA		Testaussuunnitelma		33 (34)
Prepared ANTTI VÄNSKÄ	File.doc liite5.testaussuunnitelma.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

	Listalla tulisi näkyä ainoastaan tuotteet, joiden 'category'-sarakkeen arvo on sama kuin valittu kategoria. Testataan myös valitsemalla samasta liukuvalikosta kategoria 'Development Kit'. Listalla tulisi näkyä tasan kolme (3) tuotetta. Huomioi, että filtrit nollautuvat, kun focus siirtyy pois pääikkunasta.
Mitä testataan	Toimiiko 'Category'-filteri oikein?
Läpikäytkriteerit	<ul style="list-style-type: none"> - Ei virheilmoituksia. - Filteri suodattaa listan sisältöä oikein.

4.11 Haun käyttäminen

Testattavat ominaisuudet	'Search'-toiminnon käyttäminen.
Tarkempi kuvaus	Kirjaututaan järjestelmään. Testataan hakuä syöttämällä 'Search'-kenttään erilaisia merkkijonoja. Hakuominaisuus kohdistuu tuotteiden 'Name'- ja 'Description'-kenttiin. Listalla tulisi näkyä ainoastaan tuotteet, joiden joko 'Name'- tai 'Description'-kenttä sisältää kyseisen merkkijonon. Merkkikoolla ei ole merkitystä. Testataan myös kirjoittamalla kenttään merkkijono 'puun'. Listalla tulisi näkyä tasan kolme (3) tuotetta. Voit kokeilla myös filtereiden ja haun yhteistoimintaa.
Mitä testataan	Toimiiko 'Search'-toiminto oikein?
Läpikäytkriteerit	<ul style="list-style-type: none"> - Ei virheilmoituksia. - 'Search'-kenttään lisätty merkkijono suodattaa listan sisältöä oikein.

OAMK HYVINVOINTITEKNOLOGIA		Testaussuunnitelma		34 (34)
Prepared ANTTI VÄNSKÄ	File.doc liite5.testaussuunnitelma.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

LIITTEET

Testauspöytäkirja

SIS_testauspoytakirja_locked.xlsx

Testausraportti

Scalable Inventory System (SIS)

Versio 1.0

Muutoshistoria

Versionumero	Pvm	Selitys	Tekijä(t)
1.0	23.09.2013	Ei muutoksia, hyväksyntä lisätty	Antti
0.4	18.06.2013	Korjauksia, liitteet merkitty	Antti
0.3	12.06.2013	Loput tulokset ja arviointi	Antti
0.2	06.06.2013	Lisätty tuloksia	Antti
0.1	04.06.2013	Testausraportin pohja	Antti

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		2 (38)
Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

Sisällysluettelo

1	JOHDANTO	5
2	POIKKEAMAT	6
3	SIS CLIENT - TULOKSET	8
3.1	ASENNUS	8
3.2	TILAN VALITSEMINEN	8
3.3	TILAAAN LIITTYVIEN TUOTTEIDEN LISTAAMINEN	9
3.4	VIIVAKOODIN LUKEMINEN	9
3.4.1	<i>Lukeminen tyhjällä listalla</i>	<i>9</i>
3.4.2	<i>Lukeminen listauksen jälkeen</i>	<i>10</i>
3.4.3	<i>Lukeminen toisen tuotteen ollessa valittuna</i>	<i>10</i>
3.4.4	<i>Lukeminen useaan kertaan</i>	<i>11</i>
3.4.5	<i>Puuttuvan tuotteen viivakoodin lukeminen</i>	<i>11</i>
3.4.6	<i>Väärän viivakoodin lukeminen</i>	<i>11</i>
3.5	LAINAAMINEN	12
3.5.1	<i>Lainaaminen toiseen tilaan</i>	<i>12</i>
3.5.2	<i>Lainatun lainaaminen kolmanteen tilaan</i>	<i>14</i>
3.5.3	<i>Lainaaminen useaan kertaan</i>	<i>14</i>
3.6	PALAUTTAMINEN	15
3.6.1	<i>Lainatun palauttaminen alkuperäiseen tilaan</i>	<i>15</i>
3.6.2	<i>Palautetun palauttaminen uudestaan</i>	<i>15</i>
4	SIS MANAGEMENT - TULOKSET	16
4.1	ASENNUS	16
4.2	KIRJAUTUMINEN	16
4.2.1	<i>Kirjautuminen väärillä tunnuksilla</i>	<i>16</i>
4.2.2	<i>Kirjautuminen oikeilla tunnuksilla</i>	<i>17</i>
4.3	VIIVAKOODIN LUKEMINEN	17
4.3.1	<i>Normaali lukeminen</i>	<i>17</i>
4.3.2	<i>Lukeminen toisen tuotteen ollessa valittuna</i>	<i>18</i>
4.3.3	<i>Lukeminen useaan kertaan</i>	<i>19</i>

Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx	Version 1.0	
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11

4.3.4	Puuttuvan tuotteen viivakoodin lukeminen.....	20
4.3.5	Väärän viivakoodin lukeminen.....	20
4.4	UUDEN TUOTTEEN LISÄÄMINEN	21
4.4.1	Puutteellisilla tiedoilla.....	21
4.4.2	Pitkillä ja erikoismerkkejä sisältävillä kentillä.....	21
4.5	TUOTTEEN POISTAMINEN	22
4.5.1	Tuotteen muokkaustilassa	22
4.5.2	Tuotteen ollessa valittuna	22
4.6	KATEGORIAN MUOKKAAMINEN	23
4.6.1	Uuden kategorian luominen	23
4.6.2	Uuden kategorian luominen olemassa olevan kategorian nimellä	24
4.6.3	Uudelleennimeäminen.....	24
4.6.4	Uudelleennimeäminen toisen kategorian päälle	24
4.6.5	Käytössä olevan kategorian poistaminen	26
4.6.6	Vapaan kategorian poistaminen	26
4.7	KATEGORIAN MUOKKAAMINEN	26
4.7.1	Uuden tilan luominen	26
4.7.2	Uuden tilan luominen olemassa olevan tilan nimellä	27
4.7.3	Uudelleennimeäminen.....	27
4.7.4	Uudelleennimeäminen toisen tilan päälle	28
4.7.5	Käytössä olevan tilan (original location) poistaminen.....	28
4.7.6	Käytössä olevan tilan (borrowed location) poistaminen	29
4.7.7	Vapaan tilan poistaminen.....	29
4.8	TUOTTEEN MUOKKAAMINEN	30
4.8.1	Puutteellisilla tiedoilla.....	30
4.8.2	Pitkillä ja erikoismerkkejä sisältävillä kentillä.....	30
4.8.3	Lainaaminen alkuperäiseen tilaan.....	30
4.9	TUOTTEEN TULOSTAMINEN	31
4.9.1	Pelkällä viivakoodilla	31
4.9.2	Name	32
4.9.3	Name + Category	32
4.9.4	Name + Description	32
4.9.5	Name + Category + Description	33

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		4 (38)
Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

4.9.6	Category	33
4.9.7	Category + Description.....	34
4.9.8	Description.....	34
4.10	FILTTEREIDEN KÄYTTÄMINEN	35
4.10.1	Original location	35
4.10.2	Borrowed location	35
4.10.3	Category	35
4.11	HAUN KÄYTTÄMINEN	36
5	ARVIOINTI.....	37
	LIITTEET.....	38

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		5 (38)
Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx		Version 1.0	
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

1 JOHDANTO

Tämä testausraportti sisältää Scalable Inventory System -järjestelmän käyttöönottotestauksen tulokset. Testaus kohdistui järjestelmän kahteen erilliseen yksikköön, SIS Clientiin ja SIS Managementiin. Lisätietoa testauksen sisällöstä löytyy testaussuunnitelmasta. Testitapaukset kommentteineen löytyvät testauspöytäkirjoista.

Testaus järjestettiin 29.-31.5.2013. Testaajina toimivat OAMK:n tekniikan yksikön opiskelijat Juha Nord, Pirkko Kukkoyle ja Marko Vesala. Kaikki ajoivat molempien yksiköiden kaikki testitapaukset toistoinen kerran läpi.

Testausympäristö		
Testaaja	SIS Client v1.1.2	SIS Management v1.3.1
Juha	Pöytätietokone Windows 7 Enterprise SP1 (64 bit) Intel Celeron E3300 2.50GHz / 4GB	Pöytätietokone Windows 7 Enterprise SP1 (64 bit) Intel Celeron E3300 2.50GHz / 4GB
Pirkko	Pöytätietokone Windows 7 Enterprise SP1 (64 bit) Intel Celeron E3300 2.50GHz / 4GB	Pöytätietokone Windows 7 Enterprise SP1 (64 bit) Intel Celeron E3300 2.50GHz / 4GB
Marko	Acer ICONIA Tab W500 Windows 7 Home Premium SP1 (32 bit) AMD C-60 1.00GHz / 2GB	Pöytätietokone Windows 7 Enterprise SP1 (32 bit) Intel Core 2 Duo E8400 3.00GHz / 4GB

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		6 (38)
Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx		Version 1.0	
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

2 POIKKEAMAT

Testitapaukset käytiin läpi testauspöytäkirjassa määritetyssä järjestyksessä lukuun ottamatta Pirkon suorittamaa testausta. Hänen työasemassaan havaittiin odottamaton ongelma tulostimen ajureiden kanssa, sillä tulostaminen aiheutti aina SIS Managementin kaatumisen virheilmoitukseen. Tästä syystä Pirkon testauksesta puuttuvat tulostamiseen liittyvät testitapaukset. Kyseisellä työasemalla oli ilmeisesti ollut muitakin ongelmia johtuen jonkun edellisen opiskelijan omista virityksistä.

Ongelmaa kuitenkin selvitettiin jälkeenpäin ja ajurit asennettiin toistamiseen, jolloin tulostaminen alkoikin yhtäkkiä toimia normaalisti. Ongelmaa ei siis saatu enää toistettua, joten oletettavasti asennusvaiheessa Pirkko on tehnyt jotain asennusohjeista poikkeavaa, kuten asentanut epähuomiossa väärälle tulostimelle tai käyttöjärjestelmäversiolle sopivat ajurit. Tulostaminen on toiminut moitteettomasti kaikissa muissa testiympäristöissä, joten on epätodennäköistä, että lähdekoodissa olisi mitään kyseiseen poikkeustilanteeseen johtavaa virhettä.

Lisäksi Pirkon suorittamassa testauksessa (SIS Management) hän ymmärsi yhteydettömän tilan (viimeinen toisto) merkityksen väärin. Tarkoitus oli, että yhteys internetiin katkaistaan juuri ennen testitapauksen toiminnallisen vaiheen suorittamista ja yhteys palautetaan vasta, kun ohjelma on huomannut yhteyden puuttumisen tai testitapaus on muuten suoritettu. Pirkko oli kuitenkin vain irrottanut verkkokaapelin hetkeksi ja laittanut sen sitten takaisin ennen varsinaisen testitapauksen suorittamista. Tästä syystä yhteydettömän tilan toistoja ei Pirkon testauksessa huomioida tässä testausraportissa.

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		7 (38)
Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx		Version 1.0	
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

Testauksessa ei löydetty yhtään testitapausta, jonka testaaminen olisi aiheuttanut satunnaisia virheitä. Tästä syystä testitapauksen yksittäisiä toistoja ei ole syytä käsitellä erikseen tässä raportissa.

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		8 (38)	
Prepared ANTTI VÄNSKÄ		File.doc liite6.testausraportti.v1.0.docx		Version 1.0	
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ		Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

3 SIS CLIENT - TULOKSET

3.1 Asennus

Juha: ok

Pirkko: ok

Marko: ok

Yhteenveto: ok

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	-

3.2 Tilan valitseminen

Juha: " Ilman verkkoyhteyttä ilmoituksena ettei saa yhteyttä. Ohjelma palautuu hyvin kun verkkoyhteys kytketään päälle. "

Pirkko: " Ilman verkkoyhteyttä 'No connection could be made, please contact administration' ja OK-painikkeesta tulee päävalikko. "

Marko: ok

Yhteenveto: Ohjelma toimii kuten pitääkin ja antaa oikean virheilmoituksen yhteydettömässä tilassa.

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	-

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		9 (38)
Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

3.3 Tilaan liittyvien tuotteiden listaaminen

- Juha: " Yhteydetön tila => kohdan 3.2 virheilmoitus. "
- Pirkko: " Ruokala tilassa 2 tuotetta ja hissi tilassa 3 tuotetta. Sama ilmoitus viidennessä kohdassa kuin 3.2. "
- Marko: " Ruokala ei näyttänyt Tammaa. "
- Yhteenveto: Ohjelma toimii oikein. Testitapauksessa määritetyt tuotteiden lukumäärät eivät Pirkolla ja Markolla täsmää, mutta se johtuu vain huonosta testitapauksen suunnittelusta. Puuttuvia tuotteita oli testauksen aikaan jo ehditty muuttaa Managementilla toisessa testauksessa.

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	-

3.4 Viivakoodin lukeminen

3.4.1 Lukeminen tyhjällä listalla

- Juha: " Yhteydetön tila => kohdan 3.2 virheilmoitus. "
- Pirkko: " Sama kuin 3.2. "
- Marko: ok
- Yhteenveto: ok

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	-

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		10 (38)
Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

3.4.2 Lukeminen listauksen jälkeen

Juha: " Yhteydetön tila => kohdan 3.2 virheilmoitus. "

Pirkko: " Sama kuin 3.2 ssa. "

Marko: ok

Yhteenveto: ok

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	-

3.4.3 Lukeminen toisen tuotteen ollessa valittuna

Juha: " Yhteydetön tila => kohdan 3.2 virheilmoitus. "

Pirkko: " Sama kuin 3.2 ssa. "

Marko: ok

Yhteenveto: ok

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	-

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		11 (38)
Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

3.4.4 Lukeminen useaan kertaan

Juha: " Yhteydetön tila => kohdan 3.2 virheilmoitus. "

Pirkko: " Sama kuin 3.2 ssa. "

Marko: ok

Yhteenveto: ok

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	-

3.4.5 Puuttuvan tuotteen viivakoodin lukeminen

Juha: " Yhteydetön tila => kohdan 3.2 virheilmoitus. "

Pirkko: " Sama kuin 3.2 ssa. "

Marko: ok

Yhteenveto: ok

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	-

3.4.6 Väärän viivakoodin lukeminen

Juha: " Yhteydetön tila => kohdan 3.2 virheilmoitus. "

Pirkko: " Sama kuin 3.2 ssa. "

Marko: " kun väärän viivakoodin lukee ilman internet yhteyttä, ei tule virheilmoitus: no internet connection, vaan barcode is incomplete. Mutta jos scannaa sen jälkeen

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		12 (38)
Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

esim.tammen viivakoodin niin ohjelma ilmoittaa barcode is incomplete ”

Yhteenveto: Markon huomio oikean viivakoodin lukemisesta väärän viivakoodin lukemisen jälkeen ei koske ainoastaan yhteydetöntä tilaa. Ohjelma ei osaa käsitellä väärässä formaatissa olevia viivakoodeja oikein, vaan formaatista riippuen se ilmoittaa joko ”No internet connection...” (mitä ei luonnollisesti pitäisi tulla) tai ”Barcode is incomplete...”. Jostain syystä väärän viivakoodin lukemisen jälkeen johonkin muistiin jää jokin merkkijonon pätkä, joka sotkee seuraavia viivakoodin lukuja. Ongelma poistuu uudelleenkäynnistämällä pääikkunan.

Tila	HYLÄTTY
Jatkotoimenpiteet	Ongelma on vähäinen. Jätetään toistaiseksi korjaamatta.

3.5 Lainaaminen

3.5.1 Lainaaminen toiseen tilaan

Juha: ” Yhteydetön tila => kohdan 3.2 virheilmoitus. ”

Pirkko: ” Sama kuin 3.2 ssa. ”

Marko: ” yhden ja saman tuotteen lainaaminen onnistuu, mutta kun lainaa ilman internet-yhteyttä ohjelma ilmoittaa että ei ole internet-yhteyttä. Mutta tämän jälkeen se kumminkin näyttää sen tuotteen mitä olet lainannut ”

Yhteenveto: Markon huomio tuotteen näyttämisestä yhteydettömässä tilassa on erikoista, muttei

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		13 (38)
Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

varsinaisesti ole virhe. Jos yhteys katkeaa juuri ennen Update-painikkeen painamista, ohjelma antaa virheilmoituksen (toisinaan useaan keraan) ja jättää sitten valitun tuotteen näytölle. Jos näytön tyhjentää tämän jälkeen Clear-painikkeella ja lukee viivakoodin uudestaan, antaa ohjelma edelleen virheilmoituksen ja näyttää sen jälkeen tuotteen tiedot jostain välimuistista. Jos yhteydettömässä tilassa lukee jonkin toisen tuotteen viivakoodin, se näyttää edelleen sen yhteydellisessä tilassa aiemmin luetun tuotteen viivakoodin. Ongelma poistuu, kun yhteys palautetaan.

Ongelma ei liity suoraan tähän testitapaukseen, eikä se vaikuta lainaamiseen, koska lainaaminen on mahdotonta ilman verkkoyhteyttä. Toisaalta ei ohjelma yritäkään kertoa, että tuote olisi lainattu, jos se ei todella tietokannan mukaan näin ole. Kyseessä on lähinnä kosmeettinen virhe, joka korjaantuisi parantamalla ohjelman poikkeustilakäsittelyä.

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	Jätetään poikkeustilakäsittelyn kehittäminen tuleviin projekteihin.

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		14 (38)
Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

3.5.2 Lainatun lainaaminen kolmanteen tilaan

Juha: " Yhteydetön tila => kohdan 3.2 virheilmoitus. "

Pirkko: " Sama kuin 3.2 ssa. "

Marko: ok

Yhteenveto: ok

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	-

3.5.3 Lainaus useaan kertaan

Juha: " Yhteydetön tila => kohdan 3.2 virheilmoitus. "

Pirkko: " Sama kuin 3.2 ssa. "

Marko: ok

Yhteenveto: ok

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	-

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		15 (38)
Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

3.6 Palauttaminen

3.6.1 Lainatun palauttaminen alkuperäiseen tilaan

Juha: " Yhteydetön tila => kohdan 3.2 virheilmoitus. "

Pirkko: " Sama kuin 3.2 ssa. "

Marko: ok

Yhteenvedo: ok

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	-

3.6.2 Palautetun palauttaminen uudestaan

Juha: " Yhteydetön tila => kohdan 3.2 virheilmoitus. "

Pirkko: " Sama kuin 3.2 ssa. "

Marko: ok

Yhteenvedo: ok

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	-

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		16 (38)
Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

4 SIS MANAGEMENT - TULOKSET

4.1 Asennus

Juha: ok

Pirkko: ok

Marko: ok

Yhteenveto: ok

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	-

4.2 Kirjautuminen

4.2.1 Kirjautuminen väärillä tunnuksilla

Juha: ok

Pirkko: ok

Marko: " Kun ohjelman käynnistää ilman internet-yhteyttä, ohjelma kaatuu kokonaan siihen. Mutta jos käynnistää ohjelman internetin kanssa ja vasta käynnistymisen jälkeen laittaa internet-yhteyden kiinni, niin tulee ilmoitus, että internet-yhteyttä ei ole. "

Yhteenveto: Ohjelma toimii kuten pitääkin eli ei päästä kirjautumaan väärillä tunnuksilla. Markon huomio ohjelman kaatumisesta yhteydettömässä tilassa liittyy huonoon poikkeustilakäsittelyyn.

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		17 (38)
Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	Jätetään poikkeustilakäsittelyn kehittäminen tuleviin projekteihin.

4.2.2 Kirjautuminen oikeilla tunnuksilla

- Juha: " ILMAN YHTEYTTÄ OHJELMAA EI SAA LOPETTETUA ILMAN OHJAUSPANEELIA. "
- Pirkko: ok
- Marko: " sama kuin 4.2.1 "
- Yhteenveto: Ohjelma toimii oikein eli hyväksyy oikeat käyttäjätunnukset. Toiminta yhteydettömässä tilassa vaikuttaa aiheuttavan ongelmia eli se joko jumittaa ohjelman tai kaataa sen kokonaan.

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	Jätetään poikkeustilakäsittelyn kehittäminen tuleviin projekteihin.

4.3 Viivakoodin lukeminen

4.3.1 Normaali lukeminen

- Juha: " Ilman nettiyhteyttä sovellus ei toivu lukemisen jälkeen tapahtuneesta uudelleenkytkennästä. Tehtävien hallinnalla lopetettava sovellus ja käynnistettävä uudestaan. "
- Pirkko: " Edit Item - ikkunassa voisi olla close nappi. "
- Marko: " Kun ei ole internet-yhteyttä ja yrittää lukea tuotteen viivakoodia tulee ilmoitus ei ole internet-yhteyttä, mutta

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		18 (38)
Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

ok napin jälkeen se antaa uudelleen saman ilmoituksen.
Ja tämä loppuu vasta kun ohjelma kaatuu ”

Yhteenveto: Normaalitilassa lukeminen toimii kuten pitääkin.
Yhteydettömässä tilassa ei. Pirkon mainitsema Close-painike ei toisi ikkunalle mitään lisäarvoa, sillä ikkunan saa ilman tallentamista kiinni jo neljällä eri tavalla: Cancel-painikkeella, ikkunan Close-painikkeella, ikkunan alavetovalikon Close-valinnalla ja Alt+F4-pikavalinnalla.

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	Jätetään poikkeustilakäsittelyn kehittäminen tuleviin projekteihin.

4.3.2 Lukeminen toisen tuotteen ollessa valittuna

Juha: ok

Pirkko: ” Ilmoitus 'Are you sure you want to update xxxx?' ”

Marko: ” ohjelma haluaa päivittää ensimmäisen tuotteen. Sama virhe kuin 4.3.1 ilman internettiä ”

Yhteenveto: Ohjelma ei toimi odotetulla tavalla. Se ei osaa käsitellä viivakoodin lukemista, mikäli jokin tuote on jo entuudestaan Edit Item -tilassa. Viivakoodin lukeminen on käytännössä merkkijonon lukemista ja sen hyväksymistä (Enter). Koska Save-painike on oletusvalintana Edit Item -ikkunassa, ohjelma lukee viivakoodin merkkijonon johonkin muuttujaan ja painaa sen jälkeen Enteriä, jolloin ohjelma yrittää siis tallentaa auki olevaa tuotetta, eikä aukaista juuri luettua tuotetta.

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		19 (38)
Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

Ohjelman tulisi joko antaa virheilmoitus tai kysyä käyttäjältä, haluaako hän vaihtaa muokattavaa tuotetta. Itse asiassa sama ongelma on havaittavissa myös minkä tahansa muun ikkunan ollessa auki lukuhetkellä. Ongelma ei kuitenkaan ole vakava kuin tilanteissa, joissa käyttäjä olisi lisännyt virheellistä tietoa ensimmäisen tuotteen kenttiin ja sen jälkeen epähuomiossa lukenut toisen viivakoodin ja erehtynyt vielä tämän jälkeen painamaan Enteriä tai hyväksymään tallennuksen hiirellä. Tämäkään ei tosin vaikuttaisi kuin yhteen tuotteeseen.

Tila	HYLÄTTY
Jatkotoimenpiteet	Ongelma on vähäinen. Jätetään toistaiseksi korjaamatta.

4.3.3 Lukeminen useaan kertaan

- Juha: ok
- Pirkko: " Sama ilmoitus kuin kohdassa 4.3.2. "
- Marko: " yrittää päivittää kyseisen tuotteen. Sama virhe kuin 4.3.1 ilman internettiä "
- Yhteenveto: Ohjelma ei toimi odotetulla tavalla. Samalla tavalla kuin **testitapauksessa 4.3.2**, tässäkin tapauksessa viivakoodin lukeminen Edit Item -ikkunan ollessa auki aiheuttaa Save-painikkeen painamisen. Ohjelman tulisi oikeasti joko antaa jokin virheilmoitus tai olla tekemättä mitään.

Tila	HYLÄTTY
Jatkotoimenpiteet	Ongelma on vähäinen. Jätetään toistaiseksi korjaamatta.

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		20 (38)
Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

4.3.4 Puuttuvan tuotteen viivakoodin lukeminen

- Juha: ok
- Pirkko: ok
- Marko: " new item ikkuna, jossa on vain viivakoodin numeerinen muoto. Sama virhe kuin 4.3.1 ilman internettiä "
- Yhteenveto: Ohjelma tarjoaa mahdollisuutta lisätä kyseisellä viivakoodilla olevaa tuotetta tietokantaan eli se toimii täysin oikein.

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	Jätetään poikkeustilakäsittelyn kehittäminen tuleviin projekteihin.

4.3.5 Väärän viivakoodin lukeminen

- Juha: ok
- Pirkko: " Ei anna virheilmoitusta. Ilmoittaa uudesta tuotteesta. "
- Marko: " ei anna mitään ilmoitusta väärästä koodista. Sama virhe kuin 4.3.1 ilman internettiä "
- Yhteenveto: Ohjelma toimii periaatteessa oikein, mutta läpipääsykriteeri ei täyty, sillä ohjelma ei anna virheilmoitusta väärässä formaatissa (ei EAN-13) olevan viivakoodin lukemisesta. Jos viivakoodi on jotain muuta formaattia, ohjelma ei yksinkertaisesti tee mitään. Jos taas viivakoodi sisältää oikean määrän merkkejä (13 kappaletta), ohjelma hyväksyy sen ja tarjoaa mahdollisuutta lisätä se tietokantaan. Tämä toimii myös

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		21 (38)
Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

viivakoodeilla, joiden 13 merkkiä sisältävät myös kirjaimia (jollaisia ohjelma ei itse generoi). Ne kuitenkin tallentuvat tietokantaan täysin oikein, eivätkä aiheuta ongelmia. On toisaalta hieman tulkinnanvaraista, että onko tämä menettelytapa hyväksyttävä.

Tila	HYLÄTTY
Jatkotoimenpiteet	Ongelma on kosmeettinen. Jätetään toistaiseksi korjaamatta.

4.4 Uuden tuotteen lisääminen

4.4.1 Puutteellisilla tiedoilla

- Juha: " Ilman yhteyttä New Item -painike ei toimi. "
- Pirkko: ok
- Marko: " Sama virhe kuin 4.3.1 ilman internettiä "
- Yhteenveto: Ohjelma antaa virheilmoituksen, mikäli pakollisia kenttiä ei ole täytetty. Eli se toimii kuten pitääkin. Yhteydetön tila aiheuttaa tässäkin tapauksessa ongelmia.

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	Jätetään poikkeustilakäsittelyn kehittäminen tuleviin projekteihin.

4.4.2 Pitkillä ja erikoismerkkejä sisältävillä kentillä

- Juha: ok
- Pirkko: ok

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		22 (38)
Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

Marko: " ohjelma hyväksyy kaikki syötteen. Sama virhe kuin 4.3.1 ilman internettiä "

Yhteenveto: ok

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	Jätetään poikkeustilakäsittelyn kehittäminen tuleviin projekteihin.

4.5 Tuotteen poistaminen

4.5.1 Tuotteen muokkaustilassa

Juha: ok

Pirkko: " Tällä koneella ei tulostus onnistunut. "

Marko: " Sama virhe kuin 4.3.1 ilman internettiä "

Yhteenveto: Ohjelma toimii normaalisti yhteydetöntä tilaa lukuun ottamatta. Pirkon ongelmat tulostamisen kanssa liittyvät tulostimen ajureihin, eivät testattavaan ohjelmaan.

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	Jätetään poikkeustilakäsittelyn kehittäminen tuleviin projekteihin.

4.5.2 Tuotteen ollessa valittuna

Juha: ok

Pirkko: " Kaipasin delete nappia kyseisellä sivulla. "

Marko: " Sama virhe kuin 4.3.1 ilman internettiä "

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		23 (38)
Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

Yhteenveto: Tuotteen poistaminen toimii normaalisti. Pirkon mainitseman Delete-painikkeen voisi helposti lisätä pääikkunaan, mutta toisaalta tuotteen voi jo nykyisin poistaa joko suoraan näppäimistön Delete-näppäimellä tai avaamalla tuotteen ensin Edit Item -ikkunaan. Käytettävyyden kannalta erilliselle Delete-painikkeelle pitäisi myös lisätä oma työkalupalkkinsa, sillä se olisi muiden toimintojen vierellä ainoa tuotteen muokkaamiseen liittyvä toimenpide. Mikäli käyttäjälle halutaan lisätä enemmän oikopolkuja, olisi ehkä parasta mahdollistaa hiiren oikeanpuoleisella näppäimellä aukeava pikavalikko (context menu), jolla pystyisi suoraan poistamaan ja muokkaamaan valitun tuotteen tietoja.

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	Jätetään poikkeustilakäsittelyn kehittäminen tuleviin projekteihin.

4.6 Kategorian muokkaaminen

4.6.1 Uuden kategorian luominen

Juha: ok

Pirkko: ok

Marko: " Sama virhe kuin 4.3.1 ilman internettiä "

Yhteenveto: ok

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	Jätetään poikkeustilakäsittelyn kehittäminen tuleviin projekteihin.

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		24 (38)
Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

4.6.2 Uuden kategorian luominen olemassa olevan kategorian nimellä

Juha: ok

Pirkko: ok

Marko: ” ohjelma kertoo että löytyy jo saman niminen kategoria. Sama virhe kuin 4.3.1 ilman internettiä ”

Yhteenveto: ok

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	Jätetään poikkeustilakäsittelyn kehittäminen tuleviin projekteihin.

4.6.3 Uudelleennimeäminen

Juha: ok

Pirkko: ok

Marko: ” Sama virhe kuin 4.3.1 ilman internettiä ”

Yhteenveto: ok

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	Jätetään poikkeustilakäsittelyn kehittäminen tuleviin projekteihin.

4.6.4 Uudelleennimeäminen toisen kategorian päälle

Juha: ” Jos uusi nimi on jo järjestelmässä, on uudelleen nimeämisen jälkeen samoja nimiä useampi. ”

Pirkko: ” Ohjelma ei estä nimen muuttamista, vaan listalta löytyi kaksi kytkimet kategoriaa. ”

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		25 (38)
Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

Marko: ” ohjelma ei estä mitenkään kategorian uudelleennimeämisen toisen kategorian nimiseksi. Sama virhe kuin 4.3.1 ilman internettiä ”

Yhteenveto: Ohjelma ei toimi odotetulla tavalla, vaan siinä on selkeä bugi. Vaikka uuden kategorian lisääminen olemassa olevan kategorian nimellä ei onnistu, onnistuu kategorian nimen uudelleennimeäminen toisen kategorian nimellä, jolloin tuloksena on kaksi samannimistä kategoriaa. Tästä seuraa myös se ongelma, että ohjelma ei enää kykene yksilöimään näitä kategorioita, vaan kaikki jompaankumpaan kohdistuvat muutokset vaikuttavat samalla molempiin.

Ongelmaa ei olisi, jos ohjelmassa käytettäisiin kategorioiden omia ID-tunnisteita, mutta tällä hetkellä ohjelma vertailee vain niiden nimiä. Toisaalta vaikka kategorioilla olisikin eri ID:t, ei kahta samannimistä kategoriaa saisi silti tietokannasta löytyä.

Ongelma ei silti ole merkittävä, koska todennäköisyys sen ilmenemiselle on hyvin pieni. Kategorian nimen muuttaminen on harvinainen toimenpide ja nimen muuttaminen jo olemassa olevalla nimellä äärimmäisen harvinainen vahinko.

Tila	HYLÄTTY
Jatkotoimenpiteet	Ongelma on vähäinen. Jätetään toistaiseksi korjaamatta.

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		26 (38)
Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx		Version 1.0	
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

4.6.5 Käytössä olevan kategorian poistaminen

Juha: ok

Pirkko: ok

Marko: ” Sama virhe kuin 4.3.1 ilman internettiä ”

Yhteenvedo: ok

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	Jätetään poikkeustilakäsittelyn kehittäminen tuleviin projekteihin.

4.6.6 Vapaan kategorian poistaminen

Juha: ok

Pirkko: ok

Marko: ” Sama virhe kuin 4.3.1 ilman internettiä ”

Yhteenvedo: ok

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	Jätetään poikkeustilakäsittelyn kehittäminen tuleviin projekteihin.

4.7 Kategorian muokkaaminen

4.7.1 Uuden tilan luominen

Juha: ok

Pirkko: ok

Marko: ” Sama virhe kuin 4.3.1 ilman internettiä ”

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		27 (38)
Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx		Version 1.0	
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

Yhteenveto: ok

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	Jätetään poikkeustilakäsittelyn kehittäminen tuleviin projekteihin.

4.7.2 Uuden tilan luominen olemassa olevan tilan nimellä

Juha: ok

Pirkko: ok

Marko: " Sama virhe kuin 4.3.1 ilman internettiä "

Yhteenveto: ok

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	Jätetään poikkeustilakäsittelyn kehittäminen tuleviin projekteihin.

4.7.3 Uudelleennimeäminen

Juha: ok

Pirkko: ok

Marko: " Sama virhe kuin 4.3.1 ilman internettiä "

Yhteenveto: ok

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	Jätetään poikkeustilakäsittelyn kehittäminen tuleviin projekteihin.

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		28 (38)	
Prepared ANTTI VÄNSKÄ		File.doc liite6.testausraportti.v1.0.docx		Version 1.0	
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ		Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

4.7.4 Uudelleennimeäminen toisen tilan päälle

- Juha: " KAHVIO == kahvio eli valikossa voi olla usealla eri tavalla uudelleennimetty rivi ja kaikki eri tavalla siot/pienet variaatioina => filter aina sama. Ohjelma EI ESTÄ uudelleennimeämisellä duplikaatteja. "
- Pirkko: " Ei estä uudelleennimeämistä, listalla kaksi kahvio tilaa. "
- Marko: " ei estä uudelleennimeämistä. Tekee samanlailla kuin 4.6.4 kohdassa. Sama virhe kuin 4.3.1 ilman internettiä "
- Yhteenveto: Ohjelma ei toimi odotetulla tavalla, vaan siinä on selkeä bugi. Kyseessä on täsmälleen sama ohjelmavirhe, joka ilmeni testitapauksessa **4.6.4 Uudelleennimeäminen toisen kategorian päälle**. Kategorioiden ja tilojen muokkaukseen liittyvät lähdekoodin osat ovat suurimmaksi osaksi copypatea toisistaan.

Tila	HYLÄTTY
Jatkotoimenpiteet	Ongelma on vähäinen. Jätetään toistaiseksi korjaamatta.

4.7.5 Käytössä olevan tilan (original location) poistaminen

- Juha: ok
- Pirkko: ok
- Marko: " Sama virhe kuin 4.3.1 ilman internettiä "
- Yhteenveto: ok

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		29 (38)
Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	Jätetään poikkeustilakäsittelyn kehittäminen tuleviin projekteihin.

4.7.6 Käytössä olevan tilan (borrowed location) poistaminen

Juha: ok

Pirkko: ok

Marko: " Sama virhe kuin 4.3.1 ilman internettiä "

Yhteenveto: ok

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	Jätetään poikkeustilakäsittelyn kehittäminen tuleviin projekteihin.

4.7.7 Vapaan tilan poistaminen

Juha: ok

Pirkko: ok

Marko: " Sama virhe kuin 4.3.1 ilman internettiä "

Yhteenveto: ok

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	Jätetään poikkeustilakäsittelyn kehittäminen tuleviin projekteihin.

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		30 (38)
Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx		Version 1.0	
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

4.8 Tuotteen muokkaaminen

4.8.1 Puutteellisilla tiedoilla

Juha: ok

Pirkko: ok

Marko: ” Sama virhe kuin 4.3.1 ilman internettiä ”

Yhteenveto: ok

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	Jätetään poikkeustilakäsittelyn kehittäminen tuleviin projekteihin.

4.8.2 Pitkillä ja erikoismerkkejä sisältävillä kentillä

Juha: ok

Pirkko: ok

Marko: ” ohjelma hyväksyy kaikki syötteet. Sama virhe kuin 4.3.1 ilman internettiä ”

Yhteenveto: ok

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	Jätetään poikkeustilakäsittelyn kehittäminen tuleviin projekteihin.

4.8.3 Lainaaminen alkuperäiseen tilaan

Juha: ” Lainaaminen originaaliin tilaan => laina poistuu lainatusta tilasta == palautettu. Meh?!?!?!? ”

Pirkko: ok

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		31 (38)
Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx		Version 1.0	
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

Marko: " ohjelma ei anna mitään virheilmoitusta, mutta se ei laita borrowed location sarakkeeseen mitään tietoa. Sama virhe kuin 4.3.1 ilman internettiä "

Yhteenveto: Ohjelma toimii oikein. Jos tuote on lähtöisin paikasta A ja se lainataan paikkaan A, se on silloin palautettu. Ohjelma ei ehkä ilmaise käyttäjälle erityisen selvästi, mitä tapahtuu, mutta se kuitenkin tekee sen mitä käyttäjä yrittää eli siirtää tuotteen sen alkuperäiseen tilaan.

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	Jätetään poikkeustilakäsittelyn kehittäminen tuleviin projekteihin.

4.9 Tuotteen tulostaminen

4.9.1 Pelkällä viivakoodilla

Juha: ok

Pirkko: " Tulostaminen ei onnistu tällä koneella. "

Marko: ok

Yhteenveto: Tulostaminen toimii oikein. Pirkon koneella oli ongelmia tulostimen ajureiden kanssa.

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	-

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		32 (38)
Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

4.9.2 Name

Juha: ok

Pirkko: " Tulostaminen ei onnistu. "

Marko: ok

Yhteenveto: Tulostaminen toimii oikein. Pirkon koneella oli ongelmia tulostimen ajureiden kanssa.

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	-

4.9.3 Name + Category

Juha: ok

Pirkko: " Tulostaminen ei onnistu. "

Marko: ok

Yhteenveto: Tulostaminen toimii oikein. Pirkon koneella oli ongelmia tulostimen ajureiden kanssa.

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	-

4.9.4 Name + Description

Juha: " Viivakoodi on aika tiheä, joten se on herkkä häiriöille (sotkuinen). "

Pirkko: " Tulostaminen ei onnistu. "

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		33 (38)
Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

Marko: " viivakoodi tulee printattua läjään eikä viivakoodin lukija välttämättä saa luettua "

Yhteenveto: Nimen ja kuvauksen tulostaminen aiheuttaa viivakoodin tulostumisen hyvin kapealle alueelle. Vika on todennäköisesti kyseisen tulostusvalinnan templatessa. Viivakoodi on vielä juuri ja juuri luettavissa, mutta se on parempi silti korjata. Pirkon koneella oli ongelmia tulostimen ajureiden kanssa.

Tila	HYLÄTTY
Jatkotoimenpiteet	Ongelma on vähäinen, mutta helposti korjattavissa. Korjataan versioon 1.3.2.

4.9.5 Name + Category + Description

Juha: ok

Pirkko: " Tulostaminen ei onnistu. "

Marko: ok

Yhteenveto: Tulostaminen toimii oikein. Pirkon koneella oli ongelmia tulostimen ajureiden kanssa.

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	-

4.9.6 Category

Juha: ok

Pirkko: " Tulostaminen ei onnistu. "

Marko: ok

OAMK HYVINVOINTITEKNOLOGIA Testausraportti			34 (38)
Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx	Version 1.0	
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11

Yhteenveto: Tulostaminen toimii oikein. Pirkon koneella oli ongelmia tulostimen ajureiden kanssa.

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	-

4.9.7 Category + Description

Juha: ok

Pirkko: " Tulostaminen ei onnistu. "

Marko: ok

Yhteenveto: Tulostaminen toimii oikein. Pirkon koneella oli ongelmia tulostimen ajureiden kanssa.

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	-

4.9.8 Description

Juha: ok

Pirkko: " Tulostaminen ei onnistu. "

Marko: ok

Yhteenveto: Tulostaminen toimii oikein. Pirkon koneella oli ongelmia tulostimen ajureiden kanssa.

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	-

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		35 (38)
Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

4.10 Filttareiden käyttäminen

4.10.1 Original location

Juha: ok

Pirkko: ok

Marko: " Sama virhe kuin 4.3.1 ilman internettiä "

Yhteenvedo: ok

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	-

4.10.2 Borrowed location

Juha: ok

Pirkko: ok

Marko: " Sama virhe kuin 4.3.1 ilman internettiä "

Yhteenvedo: ok

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	-

4.10.3 Category

Juha: ok

Pirkko: ok

Marko: " Sama virhe kuin 4.3.1 ilman internettiä "

Yhteenvedo: ok

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		36 (38)
Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx		Version 1.0	
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

Tila	HYVÄKSYTTY
Jatkotoimenpiteet	-

4.11 Haun käyttäminen

- Juha: ok
- Pirkko: ok
- Marko: ” jotta haku ja filterit toimisi yhdessä, täytyy ekana filletterit valita ennen hakusanaa. Sama virhe kuin 4.3.1 ilman internettiä ”
- Yhteenveto: Haku toimii muuten oikein, mutta kun sitä käytetään yhdessä filtereiden kanssa, täytyy filletterit valita ennen haku-kentän merkkijonoa, jotta ne molemmat vaikuttaisivat suodatukseen. Päinvastaisessa järjestyksessä valittuna ainoastaan haku-kenttä vaikuttaa suodatukseen. Ongelma ei kuitenkaan ole kovin merkittävä, sillä filtereitä ja hakua tuskin käytetään kovin usein yhtä aikaa, ja vaikka käytettäisiin, niin todennäköisesti niitä käytettäisiin juuri tuossa järjestyksessä, jossa ne toimivat tällä hetkellä oikein.

Tila	HYLÄTTY
Jatkotoimenpiteet	Ongelma on vähäinen. Jätetään toistaiseksi korjaamatta.

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		37 (38)
Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx	Version 1.0		
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

5 Arviointi

Käyttöönottestauksessa havaittiin kaikki bugit, jotka olivat tiedossa jo testaussuunnitelmaa tehdessä. Tämän lisäksi ilmeni muutama uusi bugi sekä erityisesti SIS Managementin poikkeustiläkäsittelyn heikohko toiminta. Havaitut ohjelmavirheet eivät olleet erityisen vakavia, eivätkä ne vaikuta ohjelman normaaliin käyttöön. Lähinnä ne ilmenevät erikoistilanteissa, joissa käyttäjä yrittää tehdä jotain odottamatonta. Alla olevassa taulukossa on nähtävillä tilastoa testauksessa käytetyistä testitapauksista.

	SIS Client	SIS Management
Testitapaukset yhteensä	14	40
Hylätyt testitapaukset	1	7
Virheen luokittelu	Vakava	-
	Merkittävä	-
	Vähäinen	1
	Kosmeettinen	-

Testitapauksen 4.9.4 bugi korjataan SIS Managementin seuraavaan versioon (1.3.2). Muut virheet jätetään tällä erää korjaamatta, mutta ne lisätään muiden kehitysehdotusten listaan tulevien projektien työstettäväksi.

Käyttöönottestauksen tulos on hyväksytty. Sekä SIS Client että SIS Management ovat valmiita käyttöönottoa varten.

OAMK HYVINVOINTITEKNOLOGIA		Testausraportti		38 (38)
Prepared ANTTI VÄNSKÄ	File.doc liite6.testausraportti.v1.0.docx		Version 1.0	
Approved TOMMI SALLINEN JUHA ALAKÄRPPÄ	Created 2011-10-19	Last saved 2013-11-11	Printed 2013-11-11	

LIITTEET

Täytetyt testauspöytäkirjat

- SIS_testauspoytakirja_Juha.xlsx
- SIS_testauspoytakirja_Pirkko.xlsx
- SIS_testauspoytakirja_Marko.xlsx

Testaussuunnitelma

- SIS_testaussuunnitelma_v1.0.pdf