

# **Gesture-tuki kliniseen tietojärjestelmään**

**Tero Voutilainen**

Opinnäytetyö

---



Koulutusala Tekniikan ja liikenteen ala	
Koulutusohjelma Tietotekniikan koulutusohjelma	
Työn tekijä(t) Tero Voutilainen	
Työn nimi Gesture-tuki kliiniseen tietojärjestelmään	
Päiväys 24.5.2013	Sivumäärä/Liitteet 31 / 1
Ohjaaja(t) tutkimuspäällikkö Matti Sipilä, ohjelmistosuunnittelija Mikko Pääkkönen, lehtori Veijo Pitkänen, käytettävyyssinööri Juha Kaasinen	
Toimeksiantaja/Yhteistyökumppani(t) GE Healthcare Finland Oy	
Tiivistelmä <p>Opinnäytetyön tavoitteena oli tutkia mahdolliset tavat toteuttaa Gesture-tuki eli ele-tuki olemassa olevaan kliiniseen tietojärjestelmään sekä suunnitella ja toteuttaa se. Minimivaatimuksena oli tuki yleisimmille yhden osoittimen eleille. Monikosketustuki toteutettaisiin, mikäli myöhemmin todettaisiin sen olevan mahdollista. Opinnäytetyöhön kuului myös tutkimus kosketusnäyttötekniikoista, erilaisista eleistä ja eleisiin liittyvistä patenteista.</p> <p>Kosketusnäyttötekniikoiden tutkimuksessa tavoitteena oli valita tekniikka, jota hyödyntävää kosketusnäyttöä voitaisiin käyttää työn toteutuksen yhteydessä. Valitun näytön täytyi myös tukea monikosketusta.</p> <p>Tuki toteutettiin Java-ohjelmointikielellä Eclipse-ohjelmointiympäristössä. Työn tukena käytettiin CVS-versionhallintajärjestelmää sekä ketterän ohjelmistokehityksen prosessia.</p> <p>Työn tuloksena toteutettiin tuki yleisimmille yhden osoittimen eleille. Lisäksi työssä toteutettiin myös yksinkertainen tukea esittelevä ja suorituskykyä vertaava sovellus sekä testitapaukset tuen käytettävyydelle.</p>	
Avainsanat gesture, kliininen, tietojärjestelmä	

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Information Technology			
Author(s) Tero Voutilainen			
Title of Thesis Gesture Support for Clinical Information System			
Date	24 May 2013	Pages/Appendices	31 / 1
Supervisor(s) Mr. Matti Sipilä, Research Manager, Mr. Mikko Pääkkönen, Software Designer, Mr. Veijo Pitkänen, Lecturer, Mr. Juha Kaasinen, Usability Engineer			
Client Organisation/Partners GE Healthcare Finland Oy			
<p>Abstract</p> <p>The aim of this thesis was to research possible ways to implement an gesture support for the existing clinical information system and also to design and develop the support. As a minimum requirement the gesture support should support common gestures with one pointer. Multi-touch support would be implemented if further investigations show it to be possible. In addition, a research on different touch screen technologies, gestures and patents concerning gestures was done.</p> <p>The aim of the research on touch screen technologies was to choose a technology which is used in an existing touch screen and also acquire the device to be used in the gesture support development. The requirement for the touch screen was that it should support Multi-touch.</p> <p>The gesture support was developed with the Java programming language using Eclipse as the software development environment. A version management system (CVS) was used as a supporting tool and the work was executed by applying agile methodologies.</p> <p>As a result of this thesis, the support was implemented meeting the minimum requirements. In addition, a simple application showing how the support should work and perform was developed. Test cases concerning the usability of the gesture support were also made.</p>			
Keywords gesture, clinical, information system			

## ALKUSANAT

Tämä opinnäytetyö tehtiin GE Healthcare Finland Oy:lle. Aloitin työn vuoden 2012 alussa ja sain sen valmiiksi keväällä 2013. Kiitän työn ohjaajia, käytettävyyssinsinööri Juha Kaasista GE Healthcare Oy:stä sekä tutkuspäällikkö Matti Sipilää, ohjelmistosuunnittelija Mikko Pääkköstä ja lehtori Veijo Pitkäästä Savonia-ammattikorkeakoulusta. Lisäksi iso kiitos kuuluu GE Healthcare Finland Oy:n henkilökunnalle työn aikana saamastani ohjauksesta ja tuesta.

Kuopiossa 24.5.2013

Tero Voutilainen

## SISÄLTÖ

1	JOHDANTO .....	8
2	GE HEALTHCARE FINLAND OY .....	9
3	KÄYTETYT TEKNIIKAT .....	10
3.1	Eclipse IDE .....	10
3.2	Java-ohjelmointikieli .....	10
3.3	CVS .....	11
4	GESTURE.....	12
5	KOSKETUSNÄYTTÖTEKNIIKAT .....	16
5.1	Resistiivinen .....	16
5.2	Kapasitiivinen .....	17
5.3	Kosketusnäytön valinta.....	18
6	TUTKIMUS.....	19
6.1	Yleiset älylaitteiden tukemat eleet ja eleisiin liittyvät patentit .....	19
6.2	Tutkitut vaihtoehdot tuen toteuttamiseksi .....	19
7	TOTEUTUS .....	23
7.1	Toteutuksen rakenne.....	23
7.2	Ongelmat ja ratkaisut.....	24
8	TESTAUS .....	26
8.1	Testitapaukset ja epävirallinen testaus .....	26
8.2	Toimistotesti.....	26
8.3	Agile-testaus.....	27
9	YHTEENVETO.....	28
	LÄHTEET .....	29

## LIITTEET

Liite 1. Toimistotestin tulokset

## TERMIT JA LYHENTEET

API	Application Programming Interface
CVS	Concurrent Versions System
Eclipse IDE	Eclipse Integrated Development Environment
Gesture	Ele. Sormella tehtävä ele kosketusnäytön pinnassa
iGesture	Gesture Recognition Framework
JAVA	Oliopohjainen ohjelmointikieli
MT4j	Multi-touch for Java. Kehitysympäristö, joka tukee usealla sormella tehtäviä eleitä
SDK	Software Development Kit
Sparsh UI	Monikosketusta tukeva sovellusrajapinta
SWING	Kevyt graafisen käyttöliittymän luontiin tarkoitettu kirjasto Javalle
VRAC	Virtual Reality Applications Center

## 1 JOHDANTO

Tämän opinnäytetyön tarkoituksena on kartoittaa mahdolliset työkalut, joilla voidaan toteuttaa ele-tuki olemassa olevaan kliiniseen tietojärjestelmään sekä suunnitella ja toteuttaa tuki. Minimivaatimuksena on, että yhden osoittimen yleisimmät toiminnot voidaan toteuttaa. Lisäksi tutustutaan kosketusnäyttöjen toimintaan liittyvään teknologiaan, erilaisiin eleisiin sekä, siihen miten eleitä käytetään älylaitteissa.

Oppimistavoitteena on syventää ohjelmistokehityksessä tarvittavia taitoja. Näihin luokituvat Java-ohjelmointikielen sujuva käyttö, projektinhallinta sekä testaus. Tavoitteena on myös oppia, miten olemassa olevaan kliiniseen tietojärjestelmään voidaan lisätä uusia ominaisuuksia.

Työn taustalla on GE Healthcare Finland Oy:n halu laajentaa olemassa oleva kliininen tietojärjestelmä tukemaan nykyaikaisista älypuhelimista ja taulutietokoneista tuttuja eleitä. Tavoitteena on tehdä järjestelmästä helppokäyttöisempi.

Raportissa käydään läpi käytetyt tekniikat ja erilaiset eleet sekä tutkimukseen, toteutukseen ja testaukseen liittyneet asiat. Vaiheet esitellään siinä järjestyksessä, kuin ne toteutuivat, sekä pohditaan lopussa työn onnistumista ja tehdään työstä yhteenveto.



## 2 GE HEALTHCARE FINLAND OY

GE Healthcare Finland Oy:n pääkonttori sijaitsee Helsingissä, Kuopion toimisto sijaitsee Technopolis Oy:n toimitiloissa. Yrityksen päätoimiala on säteilylaitteiden sekä elektronisten lääkintä- ja terapialaitteiden valmistus. (Suomen Asiakastieto.)

GE Healthcare on perustettu vuonna 2004, ja se on General Electricin ensimmäinen liiketoimintasegmentti Yhdysvaltojen ulkopuolella. GE Healthcare on yhdysvaltalaisen GE Technology Infrastructuren tytäryhtiö. GE Technology Infrastructureen kuuluvat myös GE Aviation ja GE Transportation. GE Technology Infrastructure on yhdysvaltalaisen pääyhtiön GE:in tytäryhtiö. (GE Healthcare; GE Healthcare UK.)

GE Healthcare työllisti yli 46 000 henkilöä maailmanlaajuisesti vuonna 2010, ja sen pääkonttori sijaitsee Yhdistyneen Kuningaskunnan Buckinghamshiren Little Chalfontissa. GE Healthcaren toimintaan kuuluu tuotteita ja palveluita mm. seuraavilta osialueilta: lääketieteellinen kuvantaminen ja diagnostiikka, informaatioteknologia, potilaan seurantajärjestelmät, lääketutkimus sekä biofarmaseuttiset tuotantotekniikat. (GE Healthcare; GE Healthcare UK.)

### 3 KÄYTETYT TEKNIIKAT

Tässä luvussa esitellään opinnäytetyössä käytetyt työkalut ja tekniikat. Ohjelmistokehityksessä käytettiin Eclipse-ohjelmointiympäristöä ja ohjelmointikielenä Javaa. Versionhallintaan käytettiin CVS-versionhallintajärjestelmää, joka toimii yhdessä Eclipse-ohjelmointiympäristön kanssa.

#### 3.1 Eclipse IDE

Eclipse IDE on ohjelmointiympäristö, jolla voidaan tehdä sovelluksia mm. Java-ohjelmointikielillä. Ympäristöä voidaan laajentaa myös monilla eri komponenteilla, kuten liitännäisillä. Monet avoimen lähdekoodin projektit ja yritykset ovat laajentaneet tätä ohjelmointiympäristöä. Ohjelmointiympäristöä kehitetään avoimen lähdekoodin lisenssillä, ja sen kehityksestä vastaa Eclipse Foundation säätiö. (Lars Vogel.)

Eclipseä käytettiin työn aikana pääasiallisena ohjelmointiympäristönä. Versionhallinta toteutettiin Eclipsestä käsin, koska CVS-versionhallintajärjestelmä tarjoaa Eclipseen soveltuvan liitännäisen.

#### 3.2 Java-ohjelmointikieli

Java-ohjelmointikieli on alustariippumaton, yleiskäyttöinen, rinnakkainen, vahvasti tyypitetty luokka- ja oliopohjainen kieli (Oracle.) Kielen kehittivät Bill Joy ja James Gosling työskennellessään Sun Microsystemsillä 1990-luvun alussa. Java-ohjelmointikielen käyttö alkoi vakiintua muiden kielten ohella vuodesta 1995 alkaen. Vakiintumisen taustalla ovat laitteistoriippumattomuus, kielioppi, oliopohjaisuus sekä kattava standardikirjasto. (Feizabadi.)

Java-ohjelmointikieli eroaa muista ohjelmointikielistä siten, että Javan lähdekoodia ei käännetä suoraan konekielille, vaan se käännetään tavukoodiksi ja suoritetaan virtuaalikoneessa. Tämä on Javan yksi hyvistä puolista, koska näin suoritettava Java-sovellus ei voi vaikuttaa muihin prosesseihin suoraan. Toisekseen Java-sovellukset eivät myöskään pääse virtuaalikoneen ns. hiekkalaatikon ulkopuolelle varmistaen tältä osin sovelluksen turvallisuuden.

### 3.3 CVS

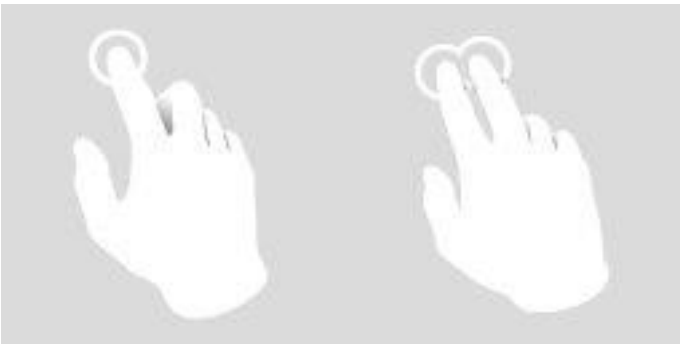
CVS on versionhallintajärjestelmä. Sillä voidaan tallentaa tiedostojen ja dokumenttien historia. Sitä käytetään myös tuotannon laatujärjestelmänä. (Derek Robert Price & Ximbiot.)

CVS seuraa ohjelmistokehityksessä lähdekoodin muutoksia sekä mahdollistaa usean ohjelmistokehittäjän yhteistyön pitäen kunkin kehittäjän tuotokset erillään versioimalla tiedostot (Cederqvist). Ohjelmistokehittäjä voi tehdä tiedostoon muutoksia, minkä jälkeen tiedosto päivitetään CVS:iin. Muutoksesta jää jälki järjestelmään, jolloin toinen kehittäjä näkee tehdyt muutokset. Tiedostojen versiointi voi estää mm. sen tilanteen, että sama kehitystyö tehdään useaan kertaan tai, että toinen vahingossa tuhoaa toisen kehittäjän tekemää ohjelmakoodia. CVS on myös siksi hyvä, että usean kehittäjän luomat lähdekooditiedostot voidaan automaattisesti yhdistää niin, ettei mitään tuhoudu tai jää uupumaan. Käytännössä CVS on myös varmuuskopiointimenetelmä kehittäjän omien varmuuskopioiden lisäksi.

## 4 GESTURE

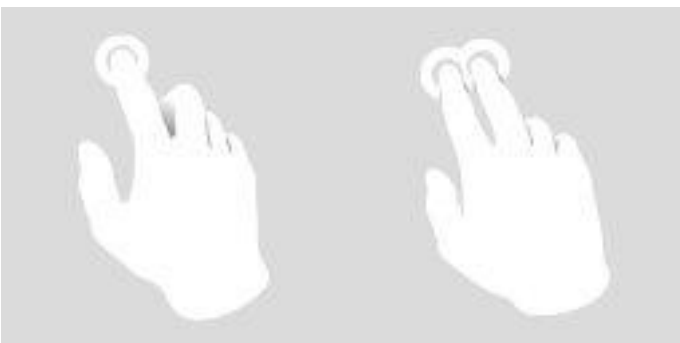
Gesturet eli eleet ovat osoittimella, joko käyttäjän sormella tai samaan tarkoitukseen tarkoitettulla esineellä esim. Stylus-kynällä, tehtäviä kosketuksia kosketusnäytöllä. Tässä luvussa esitellään eleitä, jotka tehdään kädellä eli yhdellä tai useammalla sormella. Esiteltävät eleet ovat näpäytys, kosketus ja pito, raahaus, pyyhkäisy, kaksoisnäpäytys, kierto, nipistys ja levitys. Jotkin kehitysympäristöt mahdollistavat myös omien eleiden käytön. Esimerkiksi sormella tehtävällä kuviolla voitaisiin vaikkapa laajentaa puurakenne ja samoin voitaisiin supistaa se takaisin alkuperäiseen kokoon.

Näpäytystä (kuva 1) käytetään valitsemaan jokin kohta tai painamaan nappia näytöllä. Tämä toiminto vastaa yhtä klikkausta hiiren vasemmanpuoleisella napilla suoritettuna. Näpäytys suoritetaan tekemällä sormella nopea kosketus ja irrotus kosketusnäytön pinnassa. (LukeW.)



KUVA 1. Näpäytys (Mobile Tuxedo.)

Kosketusta ja pitoa (kuva 2) käytetään pysäyttämään pyyhkäisyn laukaisema automaattinen listan vieritys pyyhkäisyn jälkeen tai aktivoimaan erityistoiminto. Tämä toiminto vastaa yleensä yhtä klikkausta hiiren oikeanpuoleisella napilla suoritettuna. Kosketus ja pito suoritetaan laskemalla sormi kosketusnäytölle ja pitämällä sormea paikallaan, kunnes haluttu toiminto on suoritettu. Käyttäjä voi jatkaa tämän jälkeen raahauksella tai pyyhkäisyllä niin halutessaan. (LukeW.)



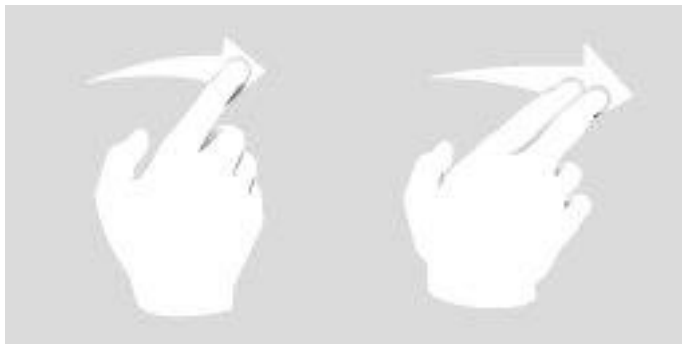
KUVA 2. Kosketus ja pito (Mobile Tuxedo.)

Raahausta (kuva 3) käytetään vierittämään listaa. Raahaus suoritetaan laskemalla sormi kosketusnäytölle ja liikuttamalla sormea heti kosketuksen yhteydessä haluttuun suuntaan nostamatta sormea kosketusnäytön pinnasta. (LukeW.)



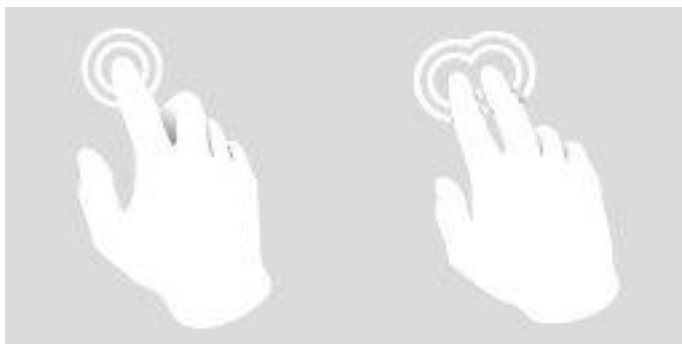
KUVA 3. Raahaus (Mobile Tuxedo.)

Pyyhkäisyä (kuva 4) käytetään vierittämään listaa tai muuta piilossa olevaa sisältöä näkyviin nopeasti. Pyyhkäisy suoritetaan laskemalla sormi kosketusnäytölle ja heti kosketuksen yhteydessä liikuttamalla sormea nopeasti haluttuun suuntaan ja irrottamalla lopuksi sormi kosketusnäytön pinnasta. (LukeW.)



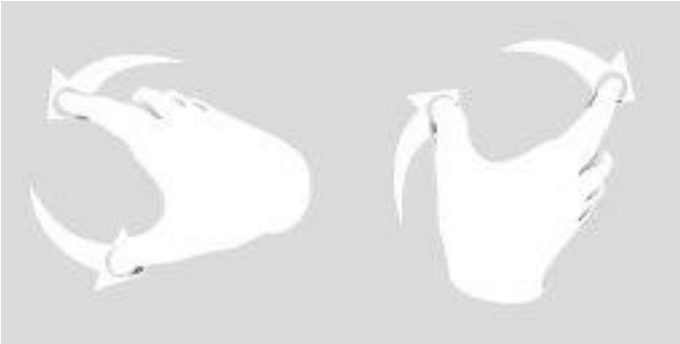
KUVA 4. Pyyhkäisy (Mobile Tuxedo.)

Kaksoisnäpätystä (kuva 5) käytetään suorittamaan jokin tietty toiminto. Tämä vastaa hiirellä tehtyä kaksoisnäpätystä. Kaksoisnäpätys suoritetaan tekemällä kaksi näpätystä peräkkäin. (LukeW.)



KUVA 5. Kaksoisnäpätys (Mobile Tuxedo.)

Kiertoa (kuva 6) käytetään kääntämään kuvaa tai muuta käännettävissä olevaa sisältöä. Kierto suoritetaan laskemalla kaksi sormea kosketusnäytön pintaan ja raahamalla sormia ympyrän muodossa myötä- tai vastapäivään. (LukeW.)



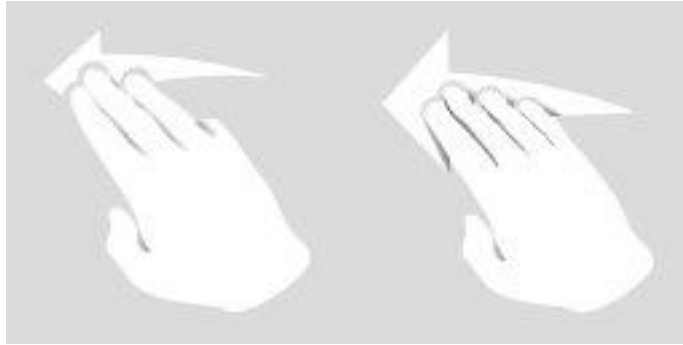
KUVA 6. Kierto (Mobile Tuxedo.)

Nipistystä ja levitystä (kuva 7) käytetään pienentämään ja suurentamaan kuvaa tai muuta pienennettävissä tai suurennettavissa olevaa sisältöä. Nipistys suoritetaan laskemalla kaksi sormea kosketusnäytön pintaan ja tuomalla sormia lähemmäksi toisiaan. Levitys suoritetaan samoin, mutta viemällä sormia kauemmaksi toisistaan. (LukeW.)

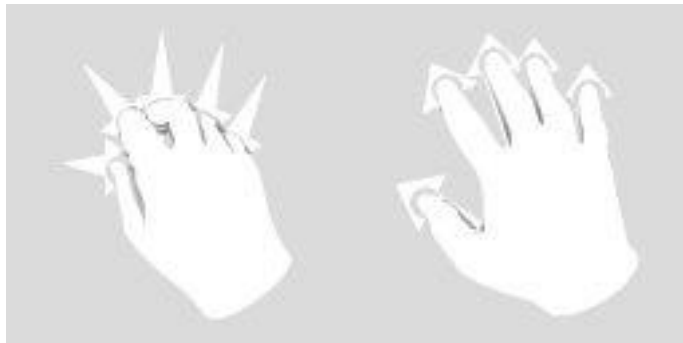


KUVA 7. Nipistys ja levitys (Mobile Tuxedo.)

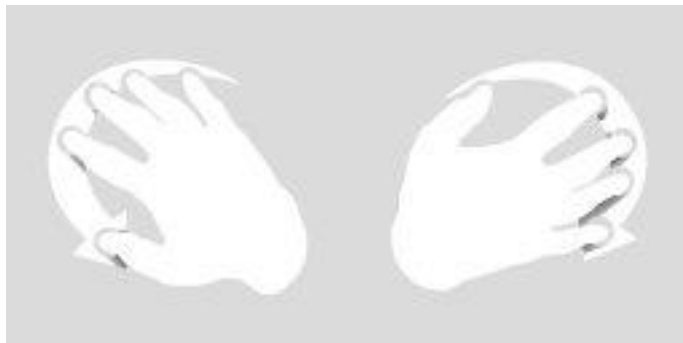
Lisäksi on mahdollista käyttää jopa 3 - 5 sormen eleitä, mutta nämä ovat huomattavasti harvinaisempia vielä toistaiseksi. Näin usean sormen käyttöä tukee ainakin pyyhkäisy OS X:ssä, jossa kolmen sormen pyyhkäisyllä (kuva 8) voidaan selata kuvakirjastoja ja neljän sormen pyyhkäisyllä (kuva 8) tuoda työpöytä tai kaikki avonaiset ikkunat näkyviin sekä myös siirtyä sovellusten välillä. Viidellä sormella Microsoft Surface:ssa voidaan käyttää nipistystä ja levitystä (kuva 9) sekä kiertoa (kuva 10). (Lu-keW.)



KUVA 8. Kolmen ja neljän sormen pyyhkäisy (Mobile Tuxedo.)



KUVA 9. Viiden sormen nipistys ja levitys (Mobile Tuxedo.)



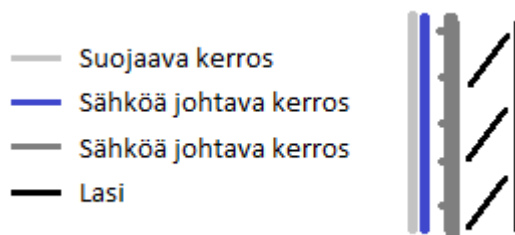
KUVA 10. Viiden sormen kierto (Mobile Tuxedo.)

## 5 KOSKETUSNÄYTTÖTEKNIIKAT

Etsiessä sopivaa kosketusnäyttöä opinnäytetyötä varten tutustuttiin muutamiin kosketusnäyttötekniikoihin, joista tässä luvussa hieman tarkempaa tietoa. Alaluku 5.3 käsittelee kosketusnäytön valintaa.

### 5.1 Resistiivinen

Resistiivinen kosketusnäyttö koostuu lasipaneelista, suojakerroksesta ja kahdesta metallikalvosta (kuva 11). Lasipaneelin uloin kerros suojaa näyttöä naarmuilta ja uloimman kerroksen alla ovat sähköä johtavat metallikerrokset. Metallisten kerrosten välissä on muovisia kohoumia, jotka pitävät kerrokset erillään. Kun kosketusnäyttöä kosketetaan, koskettavat metalliset kerrokset toisiaan ja kosketuskohdasta rekisteröidään jännitteen muutos. Tämän jännitemuutoksen perusteella kosketusnäytöltä saadaan koordinaatit, jossa kosketus tapahtui. Kosketuksen on oltava riittävän painava, jotta metalliset kerrokset koskettavat toisiaan. Tämän takia resistiivinen kosketusnäyttö ei ole niin herkkä, kuin kapasitiivinen kosketusnäyttö. (Jämsä 2010; PhoneArena.)



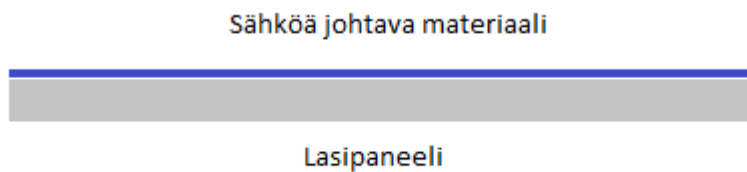
KUVA 11. Resistiivisen kosketusnäytön pelkistetty rakenne

Resistiiviset kosketusnäytöt ovat yleisesti ottaen edullisia valmistaa sekä hyvin kestäviä, jonka vuoksi niitä käytetäänkin ravintoloissa, tuotantolaitoksissa ja sairaaloissa. Resistiiviset kosketusnäytöt ovat käytännöllisiä, koska ne toimivat lähes esineestä riippumatta. Resistiivisiä kosketusnäyttöjä on myös käytetty älypuhelimissa ja käytetään edelleen, mutta enimmäkseen vain halvemmissä älypuhelimissa (Ion). Resistiiviset kosketusnäytöt eivät aiemmin normaalisti tukeneet monikosketusta, mutta sellaisia on kuitenkin nykyään olemassa. (PhoneArena, Johnson.)



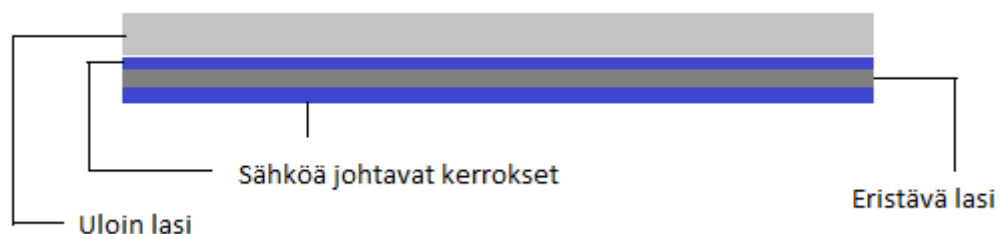
## 5.2 Kapasitiivinen

Kapasitiivinen kosketusnäyttö koostuu lasipaneelista, joka on päällystetty sähköä johtavalla materiaalilla (kuva 12). Koska ihminen johtaa sähköä, näytön koskettaminen aiheuttaa näytön sähkökenttään muutoksen, mikä voidaan mitata kapasitanssin muutoksena kosketetussa kohdassa. Kapasitiivisten kosketusnäyttöjen etuna on nopeus ja herkkyys, mutta niitä ei voida käyttää esimerkiksi tavallisella Stylus-kynällä tai käsineillä. (PhoneArena; Lion Precision 2012.)



KUVA 12. Kapasitiivisen kosketusnäytön pelkistetty rakenne

Projisoitu kapasitiivinen kosketusnäyttö koostuu lasipaneelista ja kahdesta sähköä johtavasta kerroksesta, joiden välissä on sähköä eristävä lasikerros (kuva 13). Sähköä johtavien kerrosten päissä on sähköä johtavat mittaliuskat, jotka muodostavat matriisin eli rakenteen, joka rakentuu riveistä ja kolumneista. Matriisi heijastaa sähkökentän lasipaneelin läpi. Kentän heijastumisen ja ihmiskehon sähkövarauksen ansiosta lasia kosketettaessa sähkökentän kapasitanssi muuttuu. Tämä mahdollistaa sen, että näyttö toimii myös ohuilla käsineillä. Suurin etu projisoiduissa kapasitiivisissä näytöissä on se, että ne pystyvät mittaamaan kapasitanssin useasta yhtäaikaista kosketuskohdasta ja siten näyttö tunnistaa samanaikaisia kosketuksia. Tällainen näyttö on myös hyvin tarkka ja sen käyttämiseen riittää hyvin kevyt kosketus, joten näyttö on optimaalinen eleiden käytölle. (Pitkäkangas 2012.)



KUVA 13. Projisoidun kapasitiivisen kosketusnäytön pelkistetty rakenne

### 5.3 Kosketusnäytön valinta

Opinnäytetyössä käytettiin aluksi resistiivistä kosketusnäyttöä, joka tukee myös monikosketusta. Näytön ollessa opinnäytetyön teon aikana myös muussa käytössä, päädyttiin ratkaisuun, että hankitaan erillinen näyttö ele-tuen kehitystä varten. Kosketusnäytön valintaa varten tehtiin tutkimus nykyaikaisista kosketusnäyttötekniikoista, jonka pohjalta päätettiin näytölle asetettavat vaatimukset.

Työssä käytettävän kosketusnäytön vaatimuksiksi määriteltiin, että sen on tuettava monikosketusta sekä sen olisi hyvä toimia myös ohuiden hanskojen kanssa. Näiden vaatimusten perusteella päädyttiin siihen lopputulokseen, että paras kosketusnäyttötekniikka opinnäytetyötä varten on projisoitu kapasitiivinen tekniikka.

Näytöksi valittiin Planar Systemsin PT1785P, joka on valmistettu valitulla tekniikalla ja sitä voi käyttää ohuilla käsineillä. Näyttö hankittiin opinnäytetyöprosessin alkupuolella ja käytettiin koko työn ajan kehityksen tukena. (Planar Systems.)

## 6 TUTKIMUS

Tutkimus tuen toteuttamiseksi aloitettiin tutustumalla olemassa olevien älypuhelinien ja taulutietokoneiden tukemiin eleisiin sekä niiden toimintaan yleisellä tasolla ja mahdollisiin eleisiin liittyviin patenteihin. Tämän jälkeen tutkittiin, millä työkaluilla olisi mahdollista toteuttaa ele-tuki kohteena olevaan kliiniseen tietojärjestelmään. Kliinisestä tietojärjestelmästä puhutaan tässä luvussa kohdejärjestelmänä.

### 6.1 Yleiset älylaitteiden tukemat eleet ja eleisiin liittyvät patentit

Yleisesti ottaen jokainen älypuhelin ja taulutietokone tukee kaikkia yhdellä osoittimella tehtäviä eleitä sekä kahdella sormella tehtävää suurennos- ja pienennyselettä (kuva 7). Esimerkiksi Android-älypuhelimessa raahausta (kuva 3) ja pyyhkäisyä (kuva 4) käytetään listojen vieritykseen sekä navigointiin käyttöjärjestelmässä, selaimessa sekä muissa sovelluksissa. Nipistystä ja levitystä (kuva 7) käytetään kuvien sekä selaimen sivujen suurentamiseen ja pienentämiseen. Muita kahden tai useamman sormen eleitä käytetään enemmän PC-tietokoneiden käyttöjärjestelmissä tai muissa PC-tietokoneella suoritettavissa ohjelmissa. (LukeW.)

Monissa kolmannen osapuolen ohjelmissa ja kehitystyökaluissa, esim. Android SDK:ssä, on mahdollisuus luoda omia eleitä. Omien eleiden lisääminen omaan sovellukseen on yksinkertaista Androidin Gesture API:n avulla. Käytännössä Androidin Gesture API mahdollistaa sen, että käyttäjä voi tallentaa kaikenlaiset osoittimella piirretyt kuviot käytettäväksi sovelluksessa. (Grec.)

Tutkimuksessa keskityttiin EU:n alueella käytössä oleviin patenteihin. EU:n alueelta ei tutkimuksessa löytynyt eleisiin liittyviä patenteja. Pääsyy tähän on oletettavasti se, että eleet ovat yleistä tietoa ja kaikkien vapaassa käytössä. Yhdysvalloissa sen sijaan Apple on saanut patentoitua toiminnon, jossa käytetään eleitä avaamaan lukitusnäyttö sekä monet monikosketusta vaativat toiminnot, kuten nipistys, kierto ja usean sormen pyyhkäisy. (US 7479949 B2; US 8046721 B2.)

### 6.2 Tutkitut vaihtoehdot tuen toteuttamiseksi

Työkalujen tutkinta aloitettiin kartoittamalla ensin kohdejärjestelmän tarpeet. Selvisi, että ainakin kaikkia yhden osoittimen eleitä ja yhtä kahden osoittimen elettä (kuva 7)

voitaisiin hyödyntää kohdejärjestelmässä. Täten työkaluja tutkittiin prioriteettina se, että tuki voitaisiin toteuttaa monikosketustuen kanssa.

Ensimmäisenä vaihtoehtona tutustuttiin NUI Groupin kehittämään MT4j-ohjelmointiympäristöön, joka käyttää Java-ohjelmointikieltä sekä omia luokkakirjastoja. Ohjelmointiympäristöllä voidaan luoda visuaalisesti näyttäviä sovelluksia ja se tukee monikosketusnäytön käyttöä ja siten myös usean osoittimen eleitä. (MT4j.)

Ensimmäiseksi alettiin selvittää voidaanko MT4j:n luokkakirjastoja periä tai muuten käyttää Java Swingillä toteutetussa sovelluksessa. Ongelmaksi kuitenkin osoittautui kehitettävien toiminnallisuuksien integrointi kohdejärjestelmään. MT4j:lla olisi mahdollisesti voitu toteuttaa eleet Appletina kohdejärjestelmään. Tässä tapauksessa MT4j:n luokkaa MTRectangle olisi käytetty peittämään kohdejärjestelmän komponentit (MT4j Core). MTRectangle-luokalla voidaan luoda läpinäkyvä suorakulmion muotoinen alue sovelluksen käyttöliittymän päälle. Tämä alue olisi rekisteröinyt kosketusnäytön kuunteluun tarkoitettujen metodien avulla käyttäjän suorittamia eleitä, jotka sitten olisi välitetty eteenpäin kohdejärjestelmän komponenteille. Ratkaisusta olisi tullut aivan liian raskas ja työläs käytettäväksi kohdejärjestelmän kaltaisessa ympäristössä. Näistä syistä tällaisen ratkaisun toteutus hylättiin jo tässä vaiheessa. Tukea ei myöskään voitu toteuttaa tällä työkalulla suoraan kohdejärjestelmään. Tämä johtui siitä, että työkalun käyttämät luokkakirjastot eivät ole yhteensopivia kohdejärjestelmän kirjastojen kanssa.

Seuraavana tutustuttiin Iowa Statein yliopiston VRAC:n kehittämään Sparsh UI -ohjelmointirajapintaan. Tällä ohjelmointirajapinnalla kehitettyjen sovellusten pääosana toimii Sparsh-UI Gesture Server, joka suorittaa ja välittää tiedon kosketuksista ja eleistä sovellukselle. Sparsh UI tukee sekä yleisimpiä että käyttäjän itse luomia kosketustoimintoja. Yhteys kohdejärjestelmän toiminnallisuuteen luodaan VRACin kehitettävien adaptereiden avulla. Kohdejärjestelmälle ei löytynyt sopivaa Java Swing -adapteria, joten tämäkään työkalu ei soveltunut tuen toteuttamiseen (Sparsh UI.)

Kolmantena vaihtoehtona tutustuttiin Smardecin kehittämään avoimen lähdekoodin lisensillä toteutettuun Mouse Gestures -luokkakirjastoon Javalle. Kirjastoa käyttämällä voidaan tunnistaa ja suorittaa hiirellä suoritettavia eleitä. Kirjaston tarjoamien, vierityksen hallintaan tarkoitettujen metodien käyttämien numeraalisten vakioiden arvoja ei päässyt muokkaamaan. Myöskään suurinta osaa nykyään olemassa olevista eleistä ei kirjastossa tuettu. Kirjasto osoittautui siten aivan liian suppeaksi ja joustamattomaksi, jotta sillä olisi voitu toteuttaa kohdejärjestelmän vaatimia toimintoja.

Kirjaston päätarkoitus onkin vain mahdollistaa hiirellä suoritettavien eleiden käyttö. (Smardec.)

Lopuksi tutustuttiin vielä yhteen mahdolliseen työkaluun nimeltä iGesture lähinnä vain siinä toivossa, että saataisiin jotain ratkaisevia ideoita toteuttamaan tarvittavat toiminnot. Työkalu kuitenkin osoittautui lähes saman tien soveltumattomaksi puuttuvan kosketusnäyttötuen takia. (iGesture.)

Tutkimuksen tässä vaiheessa tilanne näytti siltä, ettei tukea voitaisi toteuttaa ja työn suoritus opinnäytetyönä täytyisi arvioida uudelleen. Tutkimusta päätettiin kuitenkin vielä jatkaa ja seuraavana tutkittiin Javan luokkakirjastoja koskien normaaleja vieritystoimintoja. Lisäksi alettiin tutkia, minkälaisilla rajapinnoilla työssä käytetyn monikosketusnäytön tietovirtaa voitaisiin kuunnella ja käyttää normaalien listakomponenttien kanssa kehitettäessä sovellusta Javalla Eclipse-ohjelmointiympäristössä (Java SE API). Selvisi, että käytössä oleva kosketusnäyttö, joka toimii ilman omia laiteajureita, käyttää Windowsin laiteajureita. Nämä laiteajurit muuntavat kosketusnäytöltä tulevan tietovirran normaaleiksi hiiren toiminnoiksi, jotka ilmentyvät Java Swing -sovelluksessa hiiren liikkeitä kuuntelevien metodien laukeamisena (kuvio 1).

```
public class MouseMethods extends MouseAdapter {

    @Override
    public void mouseClicked(MouseEvent e) {
    }
    @Override
    public void mousePressed(MouseEvent e) {
    }
    @Override
    public void mouseReleased(MouseEvent e) {
    }
    @Override
    public void mouseDragged(MouseEvent e) {
    }
    @Override
    public void mouseMoved(MouseEvent e) {
    }
    @Override
    public void mouseEntered(MouseEvent e) {
    }
    @Override
    public void mouseExited(MouseEvent e) {
    }
}
```

KUVIO 1. Hiiren kuuntelun metodit

Kun osoitin koskettaa kosketusnäyttöä, sovelluksessa laukeaa mousePressed-metodi. Vastaavasti, kun osoitin irtoaa näytöstä, laukeaa mouseReleased-metodi. Näissä metodeissa suoritetaan esimerkiksi napin painallus tai jonkin kohdan valinta listakomponentissa (kuvio 2). Hiiren kuuntelun lisäys tapahtuu luomalla hiiren liikkeitä kuuntelevasta luokasta olio, joka asetetaan listakomponenttiin kuvion 3 mukaisesti.

```
@Override
public void mousePressed(MouseEvent e) {
    Point point = e.getLocationOnScreen();
    list.setSelectedIndex(list.locationToIndex(point));
}
```

KUVIO 2. Yksi tapa tehdä valinta listassa

```
JList list = new JList();
MouseMethods mm = new MouseMethods();
list.addMouseListener(mm);
list.addMouseMotionListener(mm);
```

KUVIO 3. Hiiren kuuntelijan lisäys

Kosketusnäytön toiminnan selvittyä alettiin tutkia, miten voitaisiin näiden metodien avulla toteuttaa tarvittavat eleet ja niiden tunnistusalgoritmi sekä miten näitä voidaan käyttää. Tutkimuksen aikana suunniteltu eleiden tunnistuksen logiikka ja toiminta ei ole julkista tietoa.

Tutkimuksen tuloksena ei siis löytynyt valmista työkalua, jolla ele-tuki voitaisiin toteuttaa. Tutkimus kuitenkin osoitti, että Javan omilla luokkakirjastoilla saadaan toteutettua ainakin yksinkertainen ele-tuki yhden osoittimen toiminnoille. Siispä päätettiin, että eleiden hallintatyökalu luodaan itse. Työkalun luonti ja ele-tuen lisäys käsitellään luvussa 7.

## 7 TOTEUTUS

Ele-tuen kehitys suoritettiin ketterän ohjelmistokehityksen hengessä eli lisättyjä ominaisuuksia ja toimintoja testattiin jatkuvasti kehityksen ohessa. Toteutuksen aikana kehitysprosessia seurattiin säännöllisin väliajoin pidettävien tarkistuspistein. Ohjelmakoodin tarkistuksia ja kohdejärjestelmän toteutukseen perehdyttäviä tilaisuuksia järjestettiin tarpeen vaatiessa. Työn tuotoksien versionhallinta toteutettiin käyttämällä CVS-versionhallintajärjestelmää. Tässä luvussa käsitellään ensin toteutuksen rakennetta ja sen jälkeen pohditaan toteutuksen ongelmia ja todetaan niiden ratkaisut. Eleiden tarkempi ohjelmallinen toteutus salataan.

### 7.1 Toteutuksen rakenne

Ensimmäiseksi toteutettiin kohdejärjestelmään erillinen ominaisuus, jolla voidaan hallita sitä, että onko tuki käytössä vai ei. Näin vältetään riski, että jokin alkuperäisen järjestelmän toiminnallisuuksista menisi rikki. Järjestelmän toimii normaalisti, jos tuki ei ole käytössä.

Seuraavaksi suunniteltiin ja toteutettiin työkalu, jolla eleet tunnistetaan ja niitä vastaavat toiminnot suoritetaan. Käytännössä työkalu on usean luokan paketti. Yksi näistä luokista määrittelee eleisiin liittyvät toiminnot. Luokassa on metodit eleiden tunnistamiseen ja suorittamiseen kosketusnäytöltä. Näiden lisäksi luokkaan suunniteltiin ja toteutettiin algoritmi, joka suorittaa pyyhkäisystä aktivoituvan automaattisen vieritystapahtuman säääten vierityksen nopeuden sekä kiihtyvyyden ja hidastuvuuden. Luokkaan toteutettiin myös yksi ainutlaatuinen ele, jonka käyttötarkoitus on laajentaa haluttu alue niin suureksi, kuin se mahtuu laajenemaan rajojen puitteissa sekä yhtälailla supistaa alue takaisin alkuperäiseen kokoonsa.

Paketissa on myös luokka, jonka tarkoitus on välittää komponentin vieritystoiminto ylemmälle tasolle. Tämä tapahtuu esimerkiksi silloin, kun vieritettävää listaa yritetään vierittää, mutta listassa ei ole enempää sisältöä näytettäväksi. Tällöin luokassa tutkitaan onko komponenttien hierarkiassa ylempanä jokin komponentti, jota voi vierittää. Silloin vieritystoiminto välitetään ylemmän tason komponentille ja, jos vieritettävää komponenttia ei löydy ylemmältä tasolta, vieritystoimintoa ei edes yritetä suorittaa.

Paketissa on myös luokka, jonka tarkoitus on muuttaa tekstilaatikoiden toimintaa niin, että niitä pystytään käyttämään eleiden kanssa paremmin. Luokka asettaa tekstilaati-

kon oletuksena tilaan, jossa tekstiä ei voi maalata suoraan vaan luokkaan toteutettiin ominaisuus, jolla voidaan aktivoida tekstin maalaus Kosketus ja pito -eleellä (kuva 2).

Muut pakettiin kuuluvat luokat parantavat kohdejärjestelmän alkuperäistä toimintaa eleiden toiminnan suhteen suotuisammaksi. Nämä luokat laajentavat komponenttien käyttöliittymäluokkien toimintaa. Toimintoihin lukeutuvat mm. nappien aktivointi, listoissa tehtävien valintojen suoritus sekä valintalaatikoiden ponnahtusikkunoiden avaaminen ja sulkeminen. Yksi näistä luokista laajentaa komponenttien ulkoasun määrittävän luokan. Laajennuksen tärkein ominaisuus on muuttaa vierityspalkkien ulkoasu pelkästään informatiiviseksi eli vierityspalkeista tehdään ohuita ja niitä on vaikea käyttää sormella. Tämä kannustaa käyttäjää käyttämään eleitä komponenttien vieritykseen. Muutoksen taustalla on älylaitteiden esimerkki. Vierityspalkkien vieritysominaisuus kuitenkin säilytettiin ennallaan.

Itse ele-tuen lisäys kohdejärjestelmään toimii niin, että eleiden suorituslogiikan sisältävästä luokasta luodaan olio, joka voidaan implementoida mihin tahansa Java-komponenttiin. Sitten kunkin komponentin toimintaa muutetaan laajentamalla komponentin toimintoja ohjaavaa luokkaa. Lopuksi komponenttiin vielä sisällytetään kuuluteliija, joka ohjaa tarvittaessa vieritystoiminnan ylemmälle tasolle.

Ele-tuen toiminnan kehitystä ja kehityksen aikaista testausta varten luotiin erillinen sovellus, johon rakennettiin kustakin kohdejärjestelmän komponentista omat pelkistetyt versiot. Tässä käyttöliittymässä pystyttiin testaamaan kunkin komponentin vierityksen toimintaa tai muuta haluttua toiminnallisuutta ennen varsinaista lisäystä kohdejärjestelmään. Sovelluksella voidaan myös esitellä ele-tuen toimintaa teknisessä mielessä paremmin, kun taas järjestelmän realistisen toiminnan esittelyssä on varsinainen toimiva järjestelmä parempi.

## 7.2 Ongelmat ja ratkaisut

Eleiden toiminnan sisältävää luokkaa kehittäessä mietittiin, kuinka saataisiin toiminoista mahdollisimman helppokäyttöisiä ja sulavia. Siksi oli hyvä tutustua älypuhelimien ja taulutietokoneiden tapaan suoriutua mm. vieritystapahtumista sulavasti. Tutustumisen pohjalta kehitettiin omat kiihtyvyyss- ja hidastuvuuslaskelmat. Näiden laskelmien saaminen kohdilleen kesti koko toteutuksen ajan. Tämä johtui siitä, että kun automaattisen vierityksen sai toimimaan yhden komponentin kanssa hyvin, niin seuraavassa komponentissa vieritys ei välttämättä toiminut aivan yhtä hyvin. Pitkällisen



hienosäädön jälkeen ja ottamalla huomioon pyyhkäisyn nopeus ja pituus sekä kunkin komponentin vieritysalueen pituuden vaihtelu saatiin automaattinen vieritys toimimaan hyvin. Tärkeää oli myös asettaa tietyt rajat, että missä kulmassa raahaus tai pyyhkäisy täytyy suorittaa kuhunkin suuntaan vieritettäessä, jotta haluttu vaste saadaan aikaan virheettää.

Toinen paljon pohdintaa aiheuttanut asia oli vieritystoiminnan siirto ylemmän tason käsittelyyn. Esimerkiksi eleiden käyttö nappien päällä suoritti oletuksena napin painalluksen alas, vaikka tarkoitus olisikin ollut vierittää napin taustalla olevaa aluetta. Toinen missä tällaista toiminnallisuutta tarvittiin, oli vieritettävät komponentit, joissa ei ole vieritettävää sisältöä joko ollenkaan tai vieritystapahtuman hetkellä. Nämä ongelmat selvitettiin luomalla luokka, joka kuuntelee näitä tilanteita ja siirtää vieritystoiminnan ylemmälle tasolle tarvittaessa.

Kolmas isompi kokonaisuus, joka vaati myös paljon tutkimustyötä, oli kohdejärjestelmän alkuperäisten toimintojen laajennus tukemaan eleiden käyttöä. Alkutilanteessa jokainen ele aiheutti tietynlaisen, joko osittain tai kokonaan virheellisen vasteen, kuten valinnan, vaikka toivottu toiminta olisi ollut vieritys. Tämä kuitenkin ratkaistiin laajentamalla komponenttien toiminnallisuudesta vastaavia luokkia niin, että voidaan tunnistaa haluttu toiminto ilman virheitä.

Jokaisen ongelmatilanteen ratkaisussa pidettiin mielessä kohdejärjestelmän eheys. Eheyden varmistasi se, että kaikki eleisiin liittyvä rajattiin tarkasti toimimaan vain silloin, kun ele-tuki on käytössä sovellusta käynnistäessä.

Kohdejärjestelmään tehtiin myös vierityspalkkien lisäksi muita graafisia muutoksia tukemaan ele-tuen tarjoamia uusia toimintoja. Näiden muutosten taustalla oli alkuperäisen kosketustuen vaatimat tilaratkaisut sormen paksuudelle sopivaksi. Muutosten jälkeen varsinaiselle sovelluksen informaatio-osuudelle jää siten enemmän tilaa.

## 8 TESTAUS

Tässä luvussa on käyty läpi testaukseen liittyvät asiat. Varsinaista virallista testiä ei opinnäytetyöprosessin aikana tehty, mutta toteutusta testattiin kuitenkin jatkuvasti työn edetessä ketterän ohjelmistokehityksen mukaisesti. Testausta varten luotiin testitapaukset ja ensimmäinen versio ele-tuesta testattiin toimistotestillä.

### 8.1 Testitapaukset ja epävirallinen testaus

Testitapaukset luotiin helpottamaan käytettävyyden testausta. Kustakin komponentista tai suuremmasta kokonaisuudesta luotiin yksi testitapaus. Testitapauksia luotaessa erityistä huomiota kiinnitettiin siihen, että testauksessa keskitytään helppokäyttöisyyteen, käyttäjän tekemien virheiden paikannukseen, alkuperäisen järjestelmän moitteettomaan toimintaan tuen ollessa poissa käytöstä sekä tuen moitteettomaan tekniiseen toimintaan. Testitapaukset dokumentoitiin testauksenhallintajärjestelmään.

Yrityksen testausprotokolla ei sallinut opinnäytetyön tekijän suorittaa virallisia testejä. Epävirallinen testaus kuitenkin suoritettiin testitapausten mukaisesti, josta raportoitiin opinnäytetyön ohjaajalle. Koska opinnäytetyötä varten hankittu monikosketusnäyttö tukee ohuiden käsineiden käyttöä, epäviralliseen testaukseen sisältyi myös tärkeimpien tukeen liittyvien toimintojen testaus käsineillä. Kaikki kolme käytettyä käsinettä olivat tehty lateksista. Yhdet olivat puuteroimattomat, yhdet puuteroidut sekä yhdet steriloidut ja puuteroimattomat. Kaikki testatut käsineet toimivat kosketusnäytöllä hyvin, mutta vain puuteroitu käsine toimi yhtä hyvin kuin paljas sormi. Optimaalisen toiminnan puolesta on myös tärkeää, että käyttäjällä on omaan käteen sopivat käsineet. Liian isoa käsinettä käytettäessä varsinkin puuteroimattoman käsineen sormenpää menee helposti ruttuun, jolloin eleiden käyttö vaikeutuu hieman.

### 8.2 Toimistotesti

Ele-tuen käytettävyyttä testattiin myös yrityksen sisäisessä toimistotestissä, jossa osalla osallistujista oli ennestään kokemusta vastaavan tekniikan käytöstä ja osalla ei lainkaan. Testi suoritettiin opinnäytetyön tekijän, ohjaajan sekä yhden henkilökunnan jäsenen valvonnassa etukäteen mietittyjen käytötapausten avulla.

Testissä käytettiin kehityksessä käytettyä kosketusnäyttöä sekä kohdejärjestelmän kehitysversiota, jossa on ele-tuki käytössä. Testistä saatiin arvokasta palautetta, sillä

ele-tuesta löytyi vielä parannettavaa ja korjattavaa. Osa testin tuloksista on esitelty liitteessä 1. Muut tulokset koskevat järjestelmän toimintaa yksityiskohtaisesti, joten niitä ei julkaista tässä raportissa.

### 8.3 Agile-testaus

Ele-tuen toimintaa ja käytettävyyttä testattiin jatkuvasti myös toteutuksen yhteydessä. Testattavia aiheita olivat eleiden toiminta, alkuperäisen järjestelmän koskemattomuus sekä helppokäyttöisyys. Käytännössä testausta suoritettiin hyvin intensiivisesti siinä mielessä, että aina kun jotain muutoksia tehtiin, se muutos testattiin joko itse kohdejärjestelmässä tai vähintään toteutuskappaleessa esiteltyllä erillisellä sovelluksella.

Agile-testaus on kehityksen ohessa tehtävää testausta. Se on osa ketterän ohjelmistokehityksen prosessia. Tässä työssä sovellettiin Scrum-viitekehystä, joka on kehitetty ohjelmistoprojektien hallintaan. Scrumissa kehitys jaetaan pyrähdyksiin, tehtäviin ja tarkistuspisteisiin. Työn aikana määriteltiin tehtäviä ja pidettiin tarkistuspisteitä, mutta ei varsinaisesti jaettu kehitystä ajallisesti useisiin yhtä pitkiin jaksoihin eikä pidetty päiväpalavereja. Jaksojen pituus vaihteli sopimuksen mukaan.

## 9 YHTEENVETO

Opinnäytetyön vaatimuksena oli kehittää kliiniseen tietojärjestelmään tuki, jonka avulla järjestelmää voidaan käyttää älylaitteista tutuilla eleillä. Tuki saatiin toteutettua alkuperäistä suunnitelmaa suppeampana, eli tuesta jäi puuttumaan monen sormen yhtäaikainen käyttö. Minimivaatimukset kuitenkin täyttyivät.

Ele-tuen lisäys oli monella tapaa haastava, varsinkin kun aluksi näytti siltä, ettei tukea voida toteuttaa saatavien ohjelmistoresurssien avulla. Toisekseen tietojärjestelmän suuruus sekä käyttöympäristön vaatimukset loivat paineita tehdä korkealaatuista työtä. Luovuutta käytettiin työn toteuttamiseksi ja siten työkalu toteutettiin itse. Itse tuen lisäys tietojärjestelmään oli kohtalaisen helppoa, kun järjestelmän oppi tuntemaan.

Kosketusnäyttötuki jatkettuna ele-tuella kliinisessä tietojärjestelmässä tuo lisäarvoa järjestelmälle, koska se helpottaa järjestelmän käyttöä sekä vähentää tai jopa poistaa hiiren käytön tarpeen. Kliinisessä ympäristössä on usein hyvin rajallinen tila, joten on hyvä, että hiiren käyttöön varattu tila saadaan muuhun käyttöön. Tuen jatkokehityksessä voitaisiin kehittää järjestelmään myös kosketusnäytöltä käytettävä näppäimistö, jolloin näppäimistön tila saataisiin muihin tarkoituksiin. Hiiren ja näppäimistön poistuminen helpottaa myös ympäristön pitämistä puhtaana. Hiiren ja näppäimistön puhdistaminen on hyvin työlästä verrattuna kosketusnäytön pintaan, joka puolestaan voidaan puhdistaa nopeasti tarkoitukseen sopivalla puhdistusaineella ja/tai mikrokiutuliinalla.

Opinnäytetyön teko syvensi Java-ohjelmointikielen taitoja, projektinhallinnan osaamista, ajankäytön suunnittelua sekä laatustandardit täyttävää työskentelyä. Projektinhallintamenetelmät, kuten CVS, tulivat uutena asiana ja niistä on paljon hyötyä alkuvalla uralla.

Jatkokehityksessä voidaan myös ottaa kantaa mahdollisiin järjestelmän resoluution vaihteluihin, koska toteutetussa tuessa ei tähän otettu perusteellisesti kantaa. Tarjolla olevilla Java-luokkakirjastoilla ei opinnäytetyön aikana keksitty keinoa toteuttaa monikosketusta, joten monikosketuksen lisääminen mahdollisuuden avautuessa voisi olla jatkokehityksen paikka.

## LÄHTEET

Cederqvist, P. Version Management with CVS [verkkosivu] [viitattu 23.5.2013]

Saatavissa: <http://www.network-theory.co.uk/docs/cvsmanual/>

Derek Robert Price & Ximbiot 2005 - 2006. [verkkosivu] [viitattu 23.4.2013]

Saatavissa: <http://cvs.nongnu.org/#introduction>

Feizabadi S. History of Java [verkkosivu] [viitattu 24.5.2013] Saatavissa:

[http://ei.cs.vt.edu/~wwwbtb/book/chap1/java\\_hist.html](http://ei.cs.vt.edu/~wwwbtb/book/chap1/java_hist.html)

GE Healthcare. healthymagination Investor Update [verkkodokumentti] [viitattu 23.5.2013] Saatavissa:

[http://www.ge.com/sites/default/files/ge\\_deutsche\\_bank\\_presentation\\_06242010\\_0.pdf](http://www.ge.com/sites/default/files/ge_deutsche_bank_presentation_06242010_0.pdf)

GE Healthcare UK. [verkkosivu] [viitattu 23.5.2013] Saatavissa:

<http://www3.gehealthcare.co.uk/>

Grec, V. Creating a simple Gesture Application in Android [verkkosivu] [viitattu

15.5.2013] Saatavissa: <http://androidresearch.wordpress.com/2012/01/10/working-with-gesture-api-in-android/>

Hamilo, M. 2010. Miten kosketusnäyttö toimii? Tiede-lehti [verkkajulkaisu] [viitattu

15.5.2013] Saatavissa: <http://www.vogella.com/articles/Eclipse/article.html>

iGesture. [verkkosivu] [viitattu 15.5.2013] Saatavissa: <http://www.igesture.org/>

Ion, F. 2013. THE PAST, PRESENT, AND FUTURE OF TOUCH. Arstechnica

[verkkootikkeli] [viitattu 15.5.2013] Saatavissa:

<http://arstechnica.com/gadgets/2013/04/from-touch-displays-to-the-surface-a-brief-history-of-touchscreen-technology/>

Johnson, C. Multi-touch added to resistive touchscreens [verkkootikkeli] [viitattu

15.5.2013] Saatavissa: <http://www.eetimes.com/electronics-news/4234048/Multi-touch-added-to-resistive-touchscreens>

Java SE API. API specification for version 6 of the Java™ Platform, Standard Edition

[verkkodokumentti] [viitattu 15.5.2013] Saatavissa:

<http://docs.oracle.com/javase/6/docs/api/overview-summary.html>

Jämsä, L. 2010. Resistiivinen kosketuspaneeli [verkkoartikkeli] [viitattu 15.5.2013]  
Saatavissa: <http://www.ruuvipenkki.fi/2010/11/02/resistiivinen-kosketuspaneeli>

Lars Vogel. Eclipse IDE Tutorial [verkkopublication] [viitattu 13.5.2013] Saatavissa:  
<http://www.vogella.com/articles/Eclipse/article.html>

Lion Precision 2012. Capacitive Sensor Operation and Optimization [verkkosivu]  
[viitattu 23.4.2013] Saatavissa: <http://www.lionprecision.com/tech-library/technotes/cap-0020-sensor-theory.html>

LukeW Ideation + Design 1996 - 2013. Touch Gesture Reference Guide [verkkosivu]  
[viitattu 23.4.2013] Saatavissa: <http://www.lukew.com/ff/entry.asp?1071>

Mobile Tuxedo 2012. Touch gesture icons [verkkosivu] [viitattu 15.5.2013]  
Saatavissa: <http://www.mobiletuxedo.com/touch-gesture-icons/>

MT4j. Wiki [verkkosivu] [viitattu 23.4.2013] Saatavissa:  
[http://www.mt4j.org/mediawiki/index.php/Main\\_Page](http://www.mt4j.org/mediawiki/index.php/Main_Page)

MT4j Core. JavaDocs. MT4j Wiki [verkkosivu] [viitattu 15.5.2013] Saatavissa:  
<http://www.mt4j.org/mediawiki/index.php/Javadocs>

Oracle, Java SE Documentation 2011. [verkkodokumentti] [viitattu 23.4.2013]  
Saatavissa: <http://docs.oracle.com/javase/6/docs/technotes/guides/language/>

Pitkäkangas, V. 2012. MOBILE MULTITOUCH. Mobiililaitteen ja monikosketusnäytön väliset vuorovaikutusmenetelmät [verkkopublication] [viitattu 15.5.2013] Saatavissa:  
[https://publications.theseus.fi/bitstream/handle/10024/41415/Pitkakangas\\_Ville.pdf?sequence=1](https://publications.theseus.fi/bitstream/handle/10024/41415/Pitkakangas_Ville.pdf?sequence=1)

PhoneArena. Touchscreen technologies in phones [verkkoartikkeli] [viitattu 15.5.2013] Saatavissa: [http://www.phonearena.com/news/Article-Touchscreen-technologies-in-phones\\_id3067/page/2](http://www.phonearena.com/news/Article-Touchscreen-technologies-in-phones_id3067/page/2)

Planar Systems. Planar PT1785P [verkkosivu] [viitattu 23.4.2013] Saatavissa:  
<http://www.planar.com/products/desktop-touch-screen-monitors/17-inch/pt1785p/>

Smardec. Mouse Gestures Java Library [verkkosivu] [viitattu 15.5.2013] Saatavissa:  
<http://www.smardec.com/products/mouse-gestures.html>

Sparsh UI [verkkosivu] [viitattu 23.4.2013] Saatavissa:

<http://code.google.com/p/sparsh-ui/>

Suomen Asiakastieto Oy 2013. [verkkosivu] [viitattu 23.4.2013]. Saatavissa:

<https://www.aarre.fi/aarre/?hakusana=18970646>

US 7479949 B2. Touch screen device, method, and graphical user interface for determining commands by applying heuristics [verkkosivu] [viitattu 15.5.2013]

Saatavissa: <http://www.google.com/patents/US7479949>

US 8046721 B2. Unlocking a device by performing gestures on an unlock image [verkkosivu] [viitattu 15.5.2013]. Saatavissa:

<http://www.google.com/patents/US8046721>





## TOIMISTOTESTIN TULOKSET

Taulukko 1. Järjestelmä on parempi tuen kanssa

Osallistujan numero	Täysin samaa mieltä	Osittain samaa mieltä	N/A	Osittain erimieltä	Täysin erimieltä
1	X				
2		X			
3		X			
4		X			
5		X			
6	X				
7		X			
8	X				

Taulukko 2. Tuki toimii yhtä sulavasti kuin älypuhelimessa tai taulutietokoneessa

Osallistujan numero	Täysin samaa mieltä	Osittain samaa mieltä	N/A	Osittain erimieltä	Täysin erimieltä
1				X	
2				X	
3			X		
4				X	
5			X		
6				X	
7		X			
8		X			

Taulukko 3. Tuki on toteutettu hienosti

Osallistujan numero	Täysin samaa mieltä	Osittain samaa mieltä	N/A	Osittain erimieltä	Täysin erimieltä
1		X			
2		X			
3		X			
4	X				
5	X				
6		X			
7	X				
8	X				

Taulukko 4. Pitkien listojen vieritys on helppoa

Osallistujan numero	Täysin samaa mieltä	Osittain samaa mieltä	N/A	Osittain erimieltä	Täysin erimieltä
1			X		
2		X			
3		X			
4	X				
5			X		
6	X				
7	X				
8	X				

Taulukko 5. Vierityspalkit ovat riittävän esillä

Osallistujan numero	Täysin samaa mieltä	Osittain samaa mieltä	N/A	Osittain erimieltä	Täysin erimieltä
1				X	
2			X		
3				X	
4		X			
5					X
6	X				
7					X
8				X	

Taulukko 6. Gesturet toimivat tarkasti

Osallistujan numero	Täysin samaa mieltä	Osittain samaa mieltä	N/A	Osittain erimieltä	Täysin erimieltä
1	X				
2		X			
3		X			
4				X	
5		X			
6		X			
7		X			
8		X			

