

Yritys / Opintolaitos: Oulun Yliopisto / Tietojenkäsittelytieteiden laitos		Päivitetty: 24.9.2016
Projektin / kurssin nimi: Käyttöliittymäohjelmointi		Projektin / kurssin tunniste: 811375A
Title: Salary Administration GUI –ohjelmiston rakenne		
Tiivistelmä: <p>Tämä dokumentti kuvaa Salary Administration GUI –käyttöliittymän liittyvät vaatimukset, ohjelmiston rakenteen sekä kääntämiseen tarvittavat työkalut- ja ohjeet. Ohelmisto kehitettiin kesäkurssina vuoden 2016 aikana Oulun Yliopistolla tietojenkäsittelytieteiden laitoksella.</p>		
Tekijät Aapo Keskimölä Elisa Kortela	Tarkastajat Jouni Lappalainen	Levitys Jouni Lappalainen Muut Käyttöliittymäohjelmointi-kurssin pitäjät
Tekijä : Aapo Keskimölä Sivumäärä : 44 Yhteensopivuus : Microsoft Word 2003-2009		Template: Aldapro_proj_fin.odt
Tämän dokumentin julkinen levittäminen ei ole sallittu ilman dokumentin <i>tekijöiden</i> kirjallista suostumusta.		

TIIVISTELMÄ

Harjoitustyön aiheena on suunnitella helppokäyttöinen palkanlaskentaohjelma, jolla voi käyttäjä voi helposti laskea erityyppisten työntekijöiden palkat kuukausittain. Käyttöliittymä mahdollistaa työntekijöiden tietojen syöttämisen, muuttamisen, tallentamisen ja lataamisen.

Harjoitustyön tarkoituksena on oppia käyttöliittymäsuunnitteluprosessi lähtien vaatimuksista, evaluointiin ja toteutukseen. Tässä kurssityössä on olennaisesti keskitytty käyttöliittymäsuunnittelun ohella ohjelmasuunnitteluun ja toteutukseen. Ohjelman toteutuksessa käytettiin Qt-ohjelmointiympäristöä ja C++ kieltä. Nämä valinnat tehtiin siksi, koska molemmat työkalut ovat laajalti käytössä maailmalla.

Ohjelman aihe osoittautui hyvin haastavaksi ja harjoituksen aikana pääsi yhdistämään erilaisia ohjelmointitekniikoita, joiden toteutus ei ollut aina suoraviivaista. Tästä syystä ohjelman rakenne on kuvattu hyvin kattavasti, sisältäen arkkitehtuurin UML-diagrammit ja suunnitteluratkaisut, jotka osaltaan auttavat ymmärtämään paremmin tämän ohjelman toiminallisuutta.

Ohjelmiston kompleksisuudesta johtuen, tämä harjoitustyö on suoritettu soveltuvien osien käyttäen Test-Driven Development –periaatteita. Tästä on kerrottu lisää kappaleessa 7.

SISÄLLYSLUETTELO

TIIVISTELMÄ	2
1 VAATIMUSMÄÄRITTELY	6
1.1 TOIMINNALLISUUDEN MÄÄRITTELY	6
1.2 KÄYTTÄJÄRYHMIEN IDENTIFIOINTI	7
1.2.1 ESIMERKKIKÄYTTÄJÄ 1: PALKANLASKIJA	7
1.2.2 ESIMERKKIKÄYTTÄJÄ 2: PROJEKTIPÄÄLLIKKÖ	8
1.2.3 ESIMERKKIKÄYTTÄJÄ 3: TOIMITUSJOHTAJA	8
1.3 KÄYTÖN KONTEKSTIN MÄÄRITTELY	8
1.3.1 FYYSINEN KONTEKSTI	8
1.3.2 SOSIAALINEN KONTEKSTI	8
1.3.3 ORGANISATORINEN KONTEKSTI	8
1.3.4 TOIMINNALLINEN KONTEKSTI	8
1.3.5 TEKINEN KONTEKSTI	8
2 KÄYTTÖLIITTYMÄKONSEPTIT	9
2.1 KONSEPTI #1	9
2.2 KONSEPTI #2	10
3 KÄYTTÖLIITTYMÄN PROTOTYPOINTI.....	11
3.1 PERUSNÄKYMÄ.....	11
3.2 TYÖNTEKIJÄTYYPIT	12
3.2.1 TUNTITYÖNTEKIJÄ.....	12
3.2.2 KUUKAUSITYÖNTEKIJÄ.....	13
3.2.3 MYYNTITYÖNTEKIJÄ.....	14
3.3 TYÖNTEKIJÄN LISÄÄMINEN	15
3.4 TYÖNTEKIJÄN POISTAMINEN	17
3.5 PÄÄVALIKKO.....	19
4 KÄYTTÖLIITTYMÄN EVALUOINTI.....	21
4.1 HEURISTINEN EVALUOINTI.....	21
4.1.1 TUOTTEEN TILAN NÄKYVYYS.....	21
4.1.2 TUOTTEEN VASTAAVUUS TOSIELÄMÄN KANSSA	21
4.1.3 KÄYTTÄJÄN KONTROLLI JA VAPAAUS.....	21
4.1.4 YHTÄPITÄVYYS JA STANDARDIT	21
4.1.5 VIRHEIDEN ESTÄMINEN	21
4.1.6 TUNNISTAMINEN MIELUMMIN KUIN MUISTAMINEN	21
4.1.7 KÄYTÖN JOUSTAVUUS JA TEHOKKUUS	21
4.1.8 ESTEETTINEN JA MINIMALISTINEN TAITTO	22
4.1.9 VIRHETILOJEN TUNNISTAMINEN, ILMOITTAAMINEN JA KORJAAMINEN	22
4.1.10 OPASTUS JA OHJEISTUS	22
4.2 SKENAARIOPOHJAINEN EVALUOINTI.....	22
4.2.1 PALKANLASKIJA.....	22
4.2.2 PROJEKTIPÄÄLLIKKÖ	22
4.2.3 TOIMITUSJOHTAJA	22
4.3 EVALUOINTIEN LÖYDÖKSET JA VAIKUTUKSET	23
5 VIIMEISTELTY SUUNNITELMA KÄYTTÖLIITTYMÄSTÄ.....	24
5.1 PERUSNÄKYMÄ.....	24
5.2 TYÖNTEKIJÄTYYPIT	25
5.2.1 TUNTITYÖNTEKIJÄ.....	25
5.2.2 KUUKAUSITYÖNTEKIJÄ.....	26
5.2.3 MYYNTITYÖNTEKIJÄ.....	27
5.3 TYÖNTEKIJÄN LISÄÄMINEN	28
5.4 TYÖNTEKIJÄN POISTAMINEN	31
5.5 PÄÄVALIKKO.....	32
5.5.1 UUDEN LISTAN LUOMINEN	33
5.5.2 LISTAN TALLENTAMINEN NIMELLÄ.....	34
5.5.3 LISTAN LATAAMINEN TIEDOSTOSTA.....	35

5.5.4	OHJELMAN SULKEMINEN.....	36
6	KÄYTTÖOHJE.....	37
6.1	KÄYNNISTYSOHJEET	37
6.2	KÄYTTÖLIITTYMÄN KÄYTTÖ.....	37
	POIKKEAMAT SUUNNITELMASTA	39
7	OHJELMISTON RAKENNE.....	40
7.1	OHJELMISTON HAKEMISTORAKENNE	40
7.2	TEST-DRIVEN DEVELOPMENT (TDD).....	40
7.3	MVC-ARKKITEHTUURI	40
7.3.1	NÄKYMÄ (VIEW).....	41
7.3.2	MALLI (MODEL).....	41
7.3.3	KÄSITTELIJÄ (CONTROLLER)	41
7.4	TYÖNTEKIJÄTYYPPIEN LUONTI	41
7.5	TIETOJEN TALLENNUS	41
7.6	LUOKKADIAGRAMMI (YKSINKERTAISTETTU)	42
8	HARJOITUSTYÖN TEKIJÖIDEN PALAUTUSTIEDOT	43
LIITE A	LUOKKADIAGRAMMI	44

KUVAT

Kuva 1:	Ensimmäinen käyttöliittymäkonsepti	9
Kuva 2:	Toinen käyttöliittymäkonsepti	10
Kuva 3:	Käyttöliittymäprototyyppi perusnäkökulmasta	11
Kuva 4:	Käyttöliittymäprototyyppi tuntityöntekijästä	12
Kuva 5:	Käyttöliittymäprototyyppi kuukausityöntekijästä	13
Kuva 6:	Käyttöliittymäprototyyppi myyntityöntekijästä	14
Kuva 7:	Uuden työntekijän luominen käyttöliittymäprototyypissä	15
Kuva 8:	Käyttäjän tietojen tallentuminen käyttöliittymäprototyypissä	16
Kuva 9:	Työntekijän poistaminen käyttöliittymäprototyypissä	17
Kuva 10:	Työntekijän poistamisen vahvistus käyttöliittymäprototyypissä	18
Kuva 11:	Käyttöliittymäprototyypin päävalikko	19
Kuva 12:	Käyttöliittymästä poistuminen	20
Kuva 13:	Viimeistely suunnitelma perusnäkökulmasta	24
Kuva 14:	Viimeistely suunnitelma tuntityöntekijälomakkeesta	25
Kuva 15:	Viimeistely suunnitelma kuukausityöntekijälomakkeesta	26
Kuva 16:	Viimeistely suunnitelma myyntityöntekijälomakkeesta	27
Kuva 17:	Viimeistely suunnitelma työntekijän lisäämisestä, vaihe 1	28
Kuva 18:	Viimeistely suunnitelma työntekijän lisäämisestä, vaihe 2	29
Kuva 19:	Dialogi-ikkuna kysyy, halutaanko olemassa oleva olio korvata	29
Kuva 20:	Viesti työntekijätyypin puuttumisesta	30
Kuva 21:	Viesti onnistuneesta tallennuksesta	30
Kuva 22:	Viimeistely suunnitelma työntekijän poistamisesta	31
Kuva 23:	Dialogi-ikkuna pyytää varmistusta työntekijän poistamisesta	31
Kuva 24:	Dialogi-ikkuna ilmoittaa työntekijäolion onnistuneesta poistamisesta	32
Kuva 25:	Viimeistely suunnitelma päävalikosta	32
Kuva 26:	Viimeistely suunnitelma uuden listan luomisesta	33
Kuva 27:	Dialogi-ikkuna kysyy nykyisen listan tallentamisesta	33
Kuva 28:	Viimeistely suunnitelma listan tallentamisesta	34
Kuva 29:	Viimeistely suunnitelma listan lataamisesta tiedostosta	35
Kuva 30:	Dialogi-ikkuna pyytää varmistusta listan lataamisesta	35
Kuva 31:	Viimeistely suunnitelma ohjelmasta poistumisesta	36
Kuva 32:	Dialogi-ikkuna pyytää varmistusta sulkemisesta	36
Kuva 33:	Ruutukaappaus valmiista käyttöliittymästä	37

Kuva 34. Projektin hakemistorakenne.	40
Kuva 35. MVC-malliarkkitehtuuri.	41
Kuva 35. SalaryAdministrationGui ohjelmiston UML-luokkadiagrammi.	42

TAULUKOT

Taulukko 1. Työntekijätyypit ja palkkatyypit.....	6
Taulukko 2. Työntekijätyyppien ominaisuudet.....	6
Taulukko 3. Järjestelmän käyttötarkoitus.	6
Taulukko 4. Järjestelmän toimintovaatimukset.....	7
Taulukko 5. Käyttöliittymän muut vaatimukset.	7
Taulukko 6. Hakemistojen yksityiskohtaisempi kuvaus.....	40
Taulukko 7. Harjoitustyön tekijöiden palautustiedot	43

1 VAATIMUSMÄÄRITTELY

SalaryAdministrationGui, on standalone käyttöliittymäsovellus, jolla voidaan laskea erilaisten työntekijöiden palkat syöttämällä tiedot tekstikenttiin ja jolloin ohjelma laskee automaattisesti työntekijän palkan ja tulostaa sen ruudulle. Yleistettävyyden vuoksi, oletetaan, että yrityksessä on kolmenlaisia työntekijöitä, joiden palkanlaskenta ja ominaisuudet poikkeavat olennaisesti toisistaan

Tyyppi	Palkantyyppi
Kuukausityöntekijä	Kuukausipalkka
Tuntityöntekijä	Tuntipalkka
Myyntimies	Kuukausipalkka + Bonus

Taulukko 1. Työntekijätyypit ja palkkatyypit.

Kaikki työntekijöillä on lisäksi yhteisiä muuttujia, koska he ovat kaikki henkilöitä. Työntekijöiden attribuutit on eritelty allaolevassa taulukossa.

Tyyppi	Yhteiset	Omat
Kuukausityöntekijä	Nimi, Sosiaaliturvatunnus, Palkka	Kuukausipalkka
Tuntityöntekijä		Tuntipalkka, tehdyt tunnit
Myyntimies		Kuukausipalkka, bonus-%

Taulukko 2. Työntekijätyyppien ominaisuudet.

Minimivaatimuksina ohjelmalla on se, että sillä voidaan lisätä, muokata ja tallentaa useiden työntekijöiden tietoja, jotka voidaan myös tallentaa. Tämän lisäksi, sovelluksella on oma tietojärjestelmä, johon voi tallentaa useimpien työntekijöiden tiedot, jotka ohjelma myös muistaa jälkikäteen.

1.1 TOIMINNALLISUUDEN MÄÄRITTELY

Taulukko 3. Järjestelmän käyttötarkoitus.

Käyttö	Kuvaus
Palkanlaskenta	Ohjelman todennäköisin pääkäyttäjryhmä on palkanlaskijat, jotka voivat käyttää ohjelmaa arvioimaan kuinka paljon palkkaa kullekin työntekijälle maksetaan kuukausittain
Projektisuunnittelu	Käyttöliittymää voidaan käyttää myös projektien kokonaisbudjetin arvioinnissa
Yritysjohtaminen	Yritysten hallintahenkilöt voivat myös käyttää käyttöliittymää työntekijöiden hallintajärjestelmänä, johon tallennetaan yrityksessä työskentelevien henkilöiden henkilötiedot

Taulukko 4. Järjestelmän toimintovaatimukset.

Toiminto	Kuvaus
Laske palkka	Järjestelmän tulee näyttää valitun työntekijän kuukausipalkka ja laskea se
Luo uusi työntekijä	Työntekijän lisääminen tietokantaan
Poista työntekijä	Työntekijän poistaminen tietokannasta <i>Dialogi: Kysyy käyttäjältä varmistuksen</i>
Tallenna työntekijä	Editoitavat kentät mahdollistavat työntekijän tietojen muokkaamisen <i>Dialogi: Kysyy käyttäjältä varmistuksen, jos yritetään kirjoittaa olemassaolevan työntekijän tietojen päälle</i>
Tyhjennä lomake	Nappi, jolla voidaan poistaa nopeasti valmiiksi täytetyt työntekijätiedot lomakkeesta
Tallenna työntekijälista	Tallentaa kaikki käyttöliittymän työntekijät tiedostoon <i>Dialogi: Kysyy käyttäjältä tiedostonimen</i> <i>Automaattinen tallennus: Käyttöliittymän tila säilytetään levyllä, josta se voidaan aina palauttaa</i>
Luo uusi työntekijälista	Tyhjentää käyttöliittymässä näkyvät työntekijät <i>Dialogi: Kysyy käyttäjältä varmistuksen</i>
Lataa työntekijät	Lataa työntekijälistan levyltä <i>Dialogi 1: Kysyy käyttäjältä varmistuksen</i> <i>Dialogi 2: Kysyy käyttäjältä tiedostonimen, josta työntekijät ladataan</i>
Lisätietoja	Näyttää käyttäjälle tietoja ohjelmasta sekä tekijöiden nimet ja vastualueet
Lopeta ohjelma	Valinta, jolla voidaan sulkea ohjelma <i>Dialogi: Kysyy käyttäjältä varmistuksen</i>

Taulukko 5. Käyttöliittymän muut vaatimukset.

Muu vaatimus	Kuvaus
Korvaavuus	Käyttöliittymäosan pitää olla helposti korvattavissa
Käytettävyys	Ohjelman tulee olla helppokäyttöinen, ymmärrettävä ja mahdollisimman intuitiivinen
Asennettavuus	Ohjelman pitää olla helposti asennettavissa (standalone)
Laajennettavuus	Ohjelmiston pitää tukea useita alustoja
Pysyvyys / Persistenssi	Sovellus sisältää oman tietokannan, jolla se voi palauttaa tilansa käyttöjärjestelmän kaatuessa
Best practice	Ohjelmiston rakenteen tulee käyttää parhaita tiedossa olevia tekniikoita
Virheet	Järjestelmän tulee pystyä käsittelemään erilaisia virheitä kaatumatta ja ilmoittamaan niistä käyttäjälle
Lokitus	Käyttöliittymä tallentaa ajon aikana lokit lokaaliin tiedostoon, josta ne voidaan hakea myöhemmin

1.2 KÄYTTÄJÄRYHMIEN IDENTIFIOINTI

Ohjelma sopii kaikkien palkanmaksusta vastaavien ja kiinnostuneiden henkilöiden käytettäväksi. Tällaisia ovat esimerkiksi projektipäälliköt, toimitusjohtajat ja palkanlaskijat. Projektipäällikkö saattaa olla kiinnostunut ennakoimaan kuluja, toimitusjohtaja hahmottamaan minkälaista henkilökuntaa yrityksessä on töissä ja ovatko heidän palkkansa linjassa toistensa kanssa, kun taas palkanlaskussa oleellista voi olla vain tietää paljonko palkkaa maksetaan ja kenelle. Näistä ryhmistä keskeisin ja suurilukuisin on palkanlaskijat.

1.2.1 ESIMERKKIKÄYTTÄJÄ 1: PALKANLASKIJA

Palkanlaskija tietää, minkä tyyppisiä työntekijöitä yrityksessä on töissä ja mikä heidän palkkatasonsa on. Hän haluaa ohjelman avulla laskea heidän palkkansa työtuntien tai muiden palkkasummaan vaikuttavien asioiden avulla. Palkanlaskija voi syöttää jokaisen työntekijän tiedot Salary Administration GUI:hin, joka näiden perusteella laskee palkat automaattisesti ja pitää yllä työntekijätietokantaa.

1.2.2 ESIMERKKIKÄYTTÄJÄ 2: PROJEKTIPÄÄLLIKKÖ

Projektipäällikkö haluaa ennakoida projektin kuluja. Hän voi Salary Administration GUI:hin koota projektin mahdollisen työntekijäporukan tiedot, jolloin ohjelma voi antaa arvioitujen työtuntien perusteella projektille mahdolliset henkilökulut.

1.2.3 ESIMERKKIKÄYTTÄJÄ 3: TOIMITUSJOHTAJA

Toimitusjohtaja ei ole niinkään kiinnostunut määrittelemään yksittäisten henkilöiden palkkoja vaan tarkastelemaan kokonaiskuvaa. Hän haluaa Salary Administration GUI:n avulla nähdä, miten paljon minkäkin tyyppisiä käyttäjiä yrityksessä on töissä ja ovatko heidän palkkauksensa linjassa toistensa kanssa.

1.3 KÄYTÖN KONTEKSTIN MÄÄRITTELY

1.3.1 FYYSINEN KONTEKSTI

Käyttöjärjestelmää käytetään oletettavimmin työpaikoilla toimisto-olosuhteissa. Muita mahdollisia käyttöpaikkoja ovat majoitustilat työmatkalla. Käyttö tapahtuu sisätiloissa normaalissa huoneenlämpötilassa. Melutaso on oletettavasti matala.

1.3.2 SOSIAALINEN KONTEKSTI

Käyttöjärjestelmää käytetään työyhteisön keskellä tai työmatkoilla. Paikalla voi olla muita yrityksen työntekijöitä.

1.3.3 ORGANISATORINEN KONTEKSTI

Käyttöjärjestelmä on suunniteltu toimimaan Windows-käyttöjärjestelmällisillä koneilla, sillä se on selvästi yleisin.

1.3.4 TOIMINNALLINEN KONTEKSTI

Käyttö tapahtuu oletettavasti paikallaan istuessa. Käyttäjä ei tee muita asioita käyttöliittymää käyttäessään. Käyttö saattaa häiriintyä muiden työntekijöiden tarpeiden johdosta.

1.3.5 TEKNINEN KONTEKSTI

Käyttöjärjestelmää käytetään Windows-alustaisella pöytäkoneella tai kannettavalla. Käyttö ei vaadi verkkoyhteyttä tai muita ohjelmia.

2 KÄYTTÖLIITTYMÄKONSEPTIT

2.1 KONSEPTI #1

The wireframe shows a software interface for managing employees. At the top, there are four filter buttons: 'All', 'Monthly', 'Hourly', and 'Sales', followed by a close button 'X'. Below the filters, the interface is divided into two main sections. The left section contains input fields for 'Employee type' (a dropdown menu) and a 'Clear' button. Below this, there are fields for 'Employee' information: 'Name' (with a '(string)' label) and 'SSN' (with a '(string)' label). Further down, there are fields for 'Hourly' information: 'Hours done' (with a '(int)' label) and 'Salary' (with a '(int)' label). At the bottom of the left section are three buttons: 'New', 'Save', and 'Delete'. The right section features a list of employee names: 'Employee 1', 'Employee 2', and 'Employee n', with a vertical scrollbar on the right. Below the list, there are two columns labeled 'Name' and 'SSN', and a row labeled 'Employee type'.

Kuva 1: Ensimmäinen käyttöliittymäkonsepti

Konseptissa on oikealla laidalla lista työntekijöistä ja sen alla yksityiskohtaisempia tietoja valitusta työntekijästä. Vasemmalle avautuvat työntekijän kaikki tarkat tiedot, joita voi poistaa ja tallentaa tai luoda uusi työntekijä alalaidan nappien avulla. Yläreunassa olevien nappien avulla voidaan rajoittaa listan näkymä tietyn tyyppisiin työntekijöihin.

Hyviä puolia: Tarvitaan vain yksi näkymä. Taitto on selkeä ja pystyy vain järjestelmän näkemällä olettamaan miten sitä käytetään.

Huonoja puolia: Ylälaidan napit ehkä hämmentäviä, kuten myös "New" -nappi työntekijän tietojen alla.

Tämä konsepti valikoitui selkeytensä ja helpomman ymmärrettävyytensä tähden prototypoitavaksi.

2.2 KONSEPTI #2

A hand-drawn user interface concept on a grid background. The interface is divided into three main sections within a rectangular frame. On the left is a list box containing the text 'Employee 1', 'Employee 2', and 'Employee n'. In the center is a vertical column of four buttons labeled 'Modify', 'Delete', 'Save', and 'New'. On the right is a form with five input fields, each preceded by a label: 'Name:', 'SSN:', 'Emp. type:', 'Hours done:', and 'Salary:'.

Kuva 2: Toinen käyttöliittymäkonsepti

Vasemmallalla on lista työntekijöistä. Yksityiskohtaisemmat tiedot tulevat näkyviin oikealle puolelle. Keskellä on nappeja, joiden avulla valitun työntekijän tietoja voidaan manipuloida tai luoda uusi työntekijä.

Hyviä puolia: Yksinkertainen käyttöliittymä, vaatii vain yhden näkymän.

Huonoja puolia: Näkymä on vähän hämmentävä, etenkin keskellä olevien nappien suhteen. Nyt ei ole selvää, mihin ne vaikuttavat.

3 KÄYTTÖLIITTYMÄN PROTOTYPOINTI

3.1 PERUSNÄKYMÄ

File

Last name:

First name(s):

SSN:

Pay type

Last name	First name(s)
Aalto	Erkki
Nieminen	Marja-Liisa
Virtanen	Riitta Anneli

Save Delete

Clear form

Kuva 3: Käyttöliittymäprototyyppi perusnäköymästä

Oikealla nähdään lista jo olemassa olevista työntekijäolioista. Vasemmalla on tyhjä lomake uusien työntekijätietojen syöttämistä varten, joka sisältää syöttökentät sukunimelle ("Last name"), etunimille ("First name(s)") ja sosiaaliturvatunnukselle ("SSN"). "Pay type" on alasvetovalikko työntekijätyypin valintaa varten. Lomakkeen alla ovat napit työntekijän tallentamista ("Save") ja poistamista ("Delete") sekä lomakkeen tyhjentämistä ("Clear form") varten. Ylälaidassa nähdään valikkopainike "File". Yksityiskohtaiset tiedot näytävästä näköymästä listan alla luovuttiin turhana.

3.2 TYÖNTEKIJÄTYYPIT

3.2.1 TUNTITYÖNTEKIJÄ

File

Last name:

First name(s):

SSN:

Pay type:

Hourly salary:

Hours done:

Calculated salary: XXX

Save Delete

Clear form

Last name	First name(s)
Aalto	Erkki
Nieminen	Marja-Liisa
Virtanen	Riitta Auneli

Kuva 4: Käytölliittymäprototyyppi tuntityöntekijästä

Kun "Pay type" -alasvetovalikosta valitaan tuntityöntekijä ("Hourly"), muuttuu lomake siten, että se näyttää tälle työntekijätyypille oleelliset lisäkentät tuntipalkalle ("Hourly salary") ja työtunneille ("Hours done"). Lisäksi lomakkeen alle tulee ohjelman laskema näistä kertyvä työntekijän kokonaispalkka ("Calculated salary").

3.2.2 KUUKAUSITYÖNTEKIJÄ

File

Last name:	<input type="text"/>	Last name	First name(s)
First name(s):	<input type="text"/>	Aalto	Erkki
SSN:	<input type="text"/>	Nieminen	Marja-Liisa
Pay type:	<input type="text" value="Monthly"/>	Virtanen	Riitta Anneli
Monthly salary:	<input type="text"/>		

Save Delete

Clear form

Kuva 5: Käytölliittymäprototyyppi kuukausityöntekijästä

Kun vastaavasti valitaan kuukausityöntekijä ("Monthly"), ilmestyy lomakkeeseen lisäkenttä kuukausipalkkaa ("Monthly salary") varten.

3.2.3 MYYNTITYÖNTEKIJÄ

File

Last name:	<input type="text"/>	Last name	First name(s)
First name(s):	<input type="text"/>	Aalto	Erkki
SSN:	<input type="text"/>	Nieminen	Marja-Liisa
Pay type:	<input type="text" value="Salesman"/>	Virtanen	Riitta Auneli
Monthly salary:	<input type="text"/>		
Outcome claim:	<input type="checkbox"/>		
Bonus(%):	<input type="text"/>		
Calculated salary:	<input type="text"/>		

Save

Delete

Clear form

Kuva 6: Käyttöliittymäprototyyppi myyntityöntekijästä

Myyntityöntekijän ("Salesman") valitseminen lisää lomakkeeseen kentät kuukausipalkasta ja bonuksesta ("Bonus (%)"). Lisäksi näkyviin tulee tarkistusruutu odotetun tuloksen toteutumisesta ("Outcome claim") ja rivi ohjelman laskemalle kokonaispalkalle.

3.3 TYÖNTEKIJÄN LISÄÄMINEN

File

Last name:	Yrjänä
First name(s):	Pertti
SSN:	010166-1234
Pay type:	Monthly
Monthly salary:	3500

Last name	First name(s)
Aalto	Erkki
Nieminen	Marja-Liisa
Virtanen	Riitta Ayneli

Save Delete

Clear form

Kuva 7: Uuden työntekijän luominen käyttöliittymäprototyypissä

Kuvassa käyttöliittymän lomakeosoioon on täytetty tiedot uudesta työntekijästä, jonka jälkeen siirrytään painamaan tallennusnappia.

File

Last name:	Yrjänä	Last name	First name(s)
First name(s):	Pertti	Aalto	Erkki
SSN:	010166-1234	Nieminen	Marja-Liisa
Pay type	Monthly <input type="radio"/>	Virtanen	Riitta Ayneli
Monthly salary:	3500	Yrjänä	Pertti

Save Delete

Clear form

Kuva 8: Käyttäjän tietojen tallentuminen käyttöliittymäprototyypissä

Tallennetun työntekijän nimi ilmestyy käyttöliittymän työntekijälistaan, josta kaksoisklikkaamalla työntekijän tiedot ilmestyvät näkyville lomakkeeseen.

3.4 TYÖNTEKIJÄN POISTAMINEN

The image shows a hand-drawn prototype of a user interface for deleting an employee. It is titled "File" in a box at the top left. On the left side, there are four input fields labeled "Last name:", "First name(s):", "SSN:", and "Pay type:". To the right of these fields is a table with two columns: "Last name" and "First name(s)". The table contains three rows of data: "Aalto" and "Erkki", "Marja-Liisa", and "Riitta Anneli". In the center, there is a dialog box titled "Employee deletion" with the text "Are you sure you want to delete the employee?". Below this text are two buttons labeled "Yes" and "No". At the bottom of the main form, there are three buttons: "Save", "Delete", and "Clear form". A large arrow points from the "Delete" button towards the "Employee deletion" dialog box.

Last name	First name(s)
Aalto	Erkki
	Marja-Liisa
	Riitta Anneli

Employee deletion

Are you sure you want to delete the employee?

Yes No

Save Delete Clear form

Kuva 9: Työntekijän poistaminen käyttöliittymäprototyypissä

Lähtöoletus on, että jonkun työntekijän tiedot on valittu näkyviin työntekijälistasta. Tämän jälkeen käyttäjä on painanut poistamisnappia. Näkyviin ilmestyy dialogi, jossa varmistetaan että toimenpide halutaan suorittaa.

File

Last name:		Last name	First name(s)
First name(s):		Aalto	Erkki
SSN:		Nieminen	Marja-Liisa
Pay type		Virtanen	Riitta Auneli

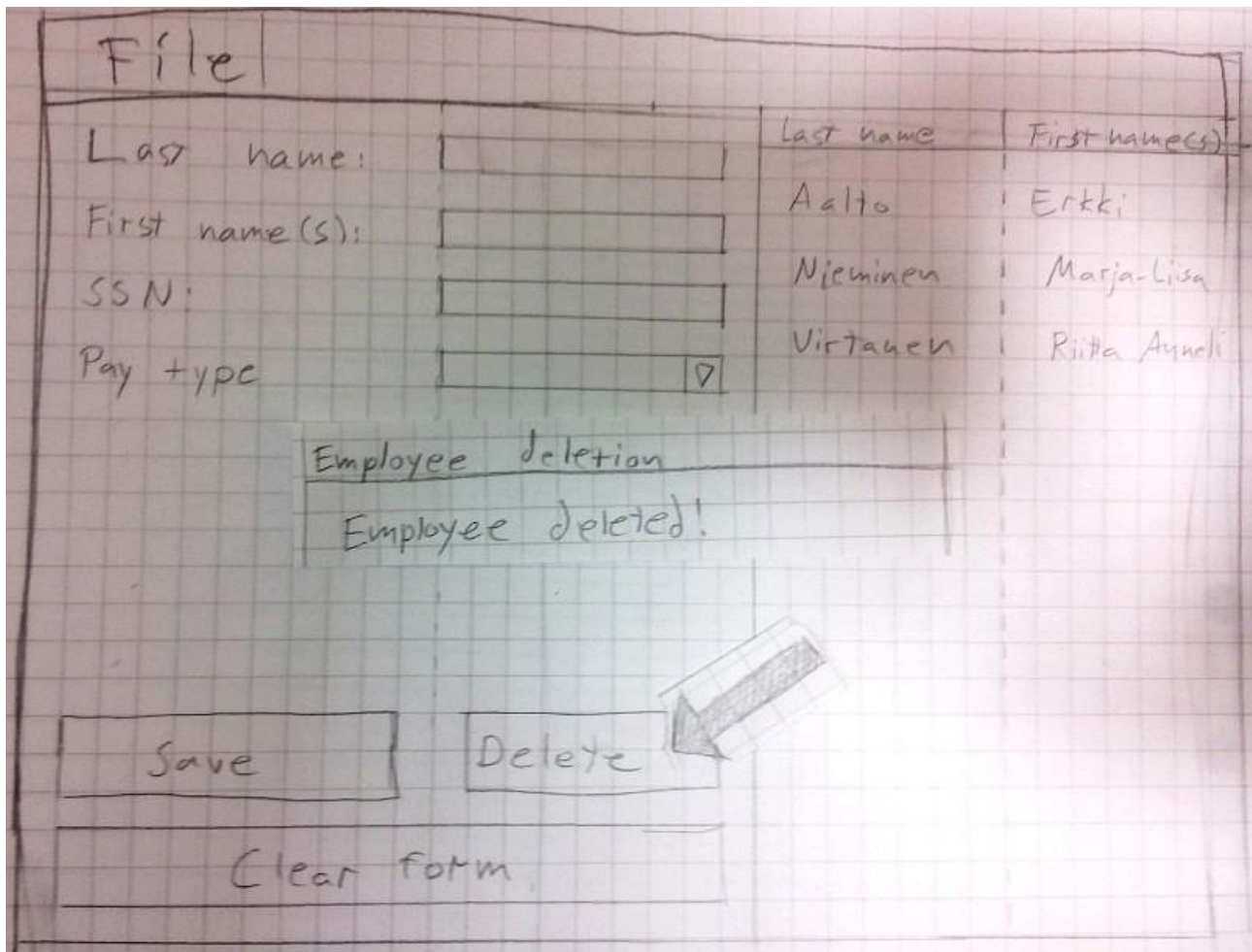
Employee deletion

Employee deleted!

Save

Delete

Clear form



Kuva 10: Työntekijän poistamisen vahvistus käyttöliittymäprototyypissä

Kun käyttäjä hyväksyy poisto-operaation, ilmestyy näkyviin dialogi-ikkuna, joka vahvistaa toimenpiteen tulleen suoritetuksi.

3.5 PÄÄVALIKKO

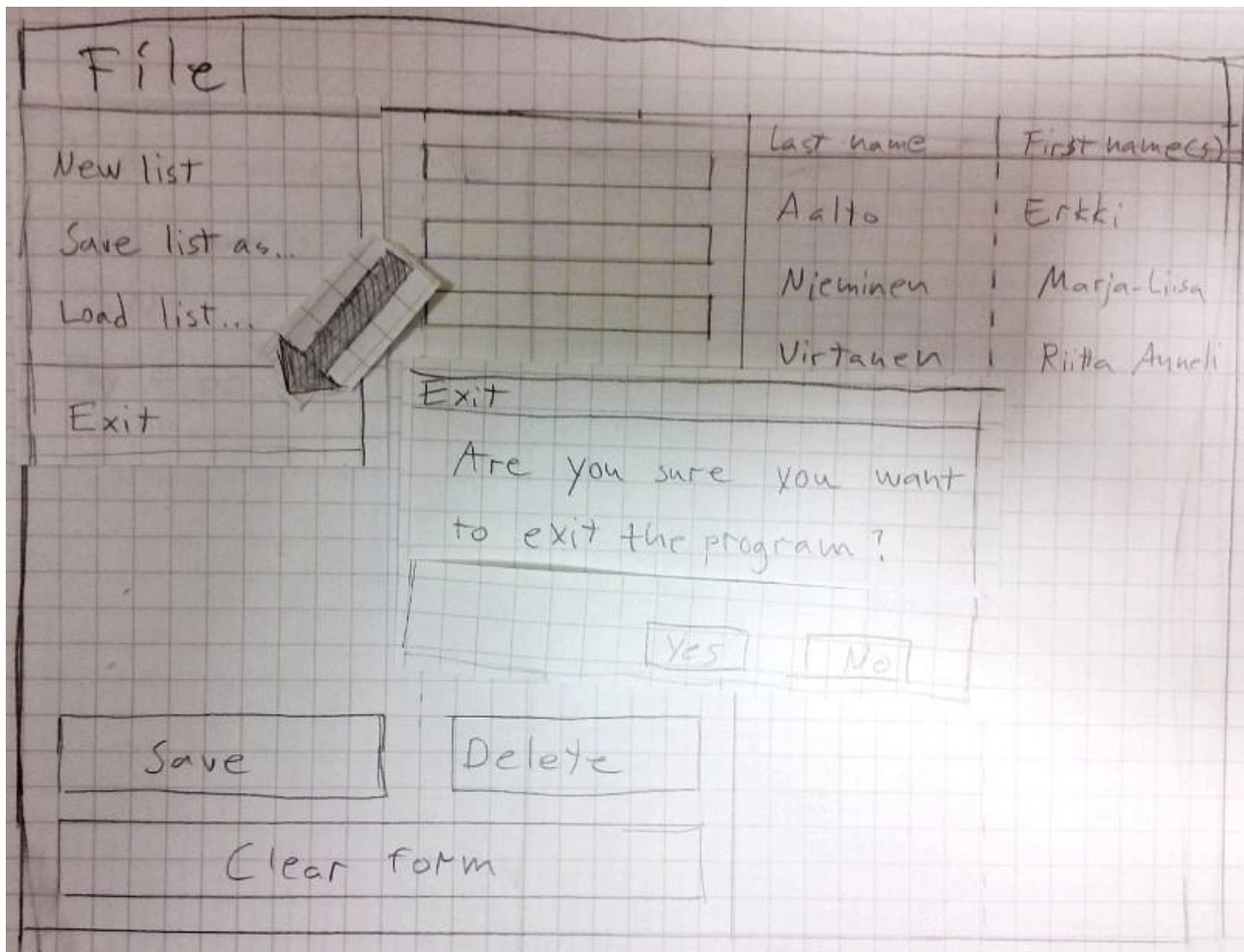
File	
New list	
Save list as...	
Load list...	
Exit	

Last name	First name(s)
Aalto	Erkki
Niemi	Marja-Liisa
Virtanen	Riitta Ayneli

Save	Delete
Clear form	

Kuva 11: Käyttöliittymäprototyypin päävalikko

Ylälaidan palkista löytyvä "File"-painike avaa näkyviin päävalikon. Tätä kautta on mahdollista luoda uusi työntekijälista ("New list"), tallentaa nykyinen lista "Save list as...") ja ladata tiedostosta aikaisemmin tallennettu lista ("Load list..."). Uuden listan valitseminen tyhjentää nykyisen näkymän oikeassa reunassa olevan työntekijälistan. Listan tallentaminen ja lataaminen tuovat näkyviin standardien mukaiset dialogi-ikkunat tiedoston hakua varten.



Kuva 12: Käyttöliittymästä poistuminen

Kun valikosta valitaan "Exit", avautuu dialogi-ikkuna joka varmistaa toimenpiteen olevan haluttu. Jos käyttäjä vahvistaa toimenpiteen painamalla "Yes", ohjelma sulkeutuu.

4 KÄYTTÖLIITTYMÄN EVALUOINTI

4.1 HEURISTINEN EVALUOINTI

Heuristisessa arvioinnissa sovellettiin Jakob Nielsenin käyttöliittymäsuunnittelun heuristiikkoja vuodelta 1995 ([10 Usability Heuristics for User Interface Design](#)).

4.1.1 TUOTTEEN TILAN NÄKYVYYS

Perusnäkömästä katselemalla nähdään, että ohjelma käsittelee jonkunlaista listaa johon liittyy lomakkeen täyttämistä. Tallennettaessa täytetty lomake päivittyy listaan uusi nimi, jolloin toimenpiteen voidaan tulkita onnistuneen. Pitäisikö tässä kuitenkin olla jokin dialogi-ilmoitus? Jos työntekijälista on pitkä, käyttäjä ei välttämättä havaitse uuden työntekijän lisäämistä ja saattaa yrittää tallentaa samaa työntekijää toisen kerran.

4.1.2 TUOTTEEN VASTAAVUUS TOSIELÄMÄN KANSSA

Käyttöliittymän käyttämä kieli vastaa tosielämän kieltä. Termistön olettaisi olevan käyttäjän ymmärrettävissä. "Outcome claim" –tarkistusruutu saattaa kuitenkin olla tarkoituksensa puolesta hämmentävä.

4.1.3 KÄYTTÄJÄN KONTROLLI JA VAPAUS

Käyttöliittymässä ei ole "tee uudelleen" ja "kumoa" –toimintoja. Kumoaminen korvaantuu osittain käyttöliittymän poistamistoiminnoilla, tosin tällä tavalla ei voida poistaa useaa viimeksi lisättyä työntekijää kerralla. Kerran poistettua työntekijää ei pystytä palauttamaan. Ohjelma kuitenkin varmistaa dialogi-ikkunalla halutaanko poisto suorittaa. Käyttäjä ei pysty ryhmittelemään työntekijöitä tyyppin mukaan eikä näkemään koontaa palkkatiedoista. Suunnittelussa on tältä käytön osalta puutteita.

4.1.4 YHTÄPITÄVYYS JA STANDARDIT

Valikon ja nappien toiminnot vastaavat nimiltään ja funktioiltaan tavanomaisia ohjelmistokäytäntöjä. Listassa näkyvät otsikot "Last name" ja "First name(s)" vastaavat lomakkeen kenttien otsikoita, jolloin niiden voidaan olettaa liittyvän toisiinsa.

4.1.5 VIRHEIDEN ESTÄMINEN

Käyttöliittymän suunnittelusta puuttuvat dialogi-ikkunat vaillinaisten käyttäjätietojen syöttämisen käsittelyyn. Se ei myöskään estä tallentamasta samaa henkilöä tai sosiaaliturvatunnusta kahdesti.

4.1.6 TUNNISTAMINEN MIELUMMIN KUIN MUISTAMINEN

Koska käyttöliittymässä on vain yksi näkymä, ei käyttäjän tarvitse muistella mitään aiemmin suoritettuja toimintoja. Käyttöliittymän taitto on yksinkertainen.

4.1.7 KÄYTÖN JOUSTAVUUS JA TEHOKKUUS

Käyttö ei ole joustavaa sikäli että on olemassa vain yhdenlainen näkymä, joka ei ole muokattavissa. On siis erityisen tärkeää suunnitella se niin, että se palvelee erityyppisiä käyttäjiä, mikä ei tällä hetkellä toteudu (ks. kohta 4.1.3). Muuten käyttö on suoraviivaista.

4.1.8 ESTEETTINEN JA MINIMALISTINEN TAITTO

Näkymä ei ole ruuhkainen, tosin ei pelkistettykään. Mitään siitä ei voi oikein karsia haittaamatta käytettävyyttä. On varmistettava, että lomakkeen alla olevat painikkeet ymmärretään siihen liittyviksi. Onnistuisiko tämä esimerkiksi kehyksillä?

4.1.9 VIRHETILOJEN TUNNISTAMINEN, ILMOITTAAMINEN JA KORJAAMINEN

Tällä hetkellä käyttöliittymään ei ole suunniteltu minkäänlaisia virheilmoituksia. Näitä on lisättävä ainakin kohdassa 4.1.5 mainittuihin tilanteisiin.

4.1.10 OPASTUS JA OHJEISTUS

Käyttöohje löytyy kohdasta 6. Tarkoituksenmukaista olisi kuitenkin, että käyttö ei sellaista vaadi.

4.2 SKENAARIOPOHJAINEN EVALUOINTI

4.2.1 PALKANLASKIJA

Tehtäväskenaario: Käyttäjä haluaa laskea palkkatiedot myyntihenkilölle, joka on toteuttanut oletetun myyntituloksen.

Käyttöskenaario: Jos lomakkeessa on ennestään tietoja, käyttäjä painaa "Clear form" –painiketta. Käyttäjä syöttää työntekijän nimi- ja sotutiedot ja valitsee alasvetovalikosta työntekijätyypin "Salesman". Lomakkeeseen ilmestyy lisää syöttökenttiä. Käyttäjä syöttää tiedot myyntihenkilön kuukausipalkasta ja tälle sovitusta bonusprosentista. Koska oletettu myyntitulo on toteutunut, käyttäjä laittaa ruksin tarkistusruutuun.

Ongelmakohta: Missä vaiheessa käyttöliittymä laskee myyntihenkilön palkan? Pitääkö käyttäjän tallentaa ensin työntekijän tiedot listaan ja avata tämän tiedot uudelleen näkyviin? Nykyisessä käyttöliittymässä ei ole mitään painiketta tai muuta viitettä siihen, miten tämä tapahtuu.

4.2.2 PROJEKTIPÄÄLLIKKÖ

Tehtäväskenaario: Käyttäjä haluaa laskea, minkä suuruiset suunnitellun projektin työntekijäkulut voidaan arvioida olevan, kun työntekijäkaarti on tiedossa.

Käyttöskenaario: Käyttäjä syöttää jokaisen projektin työntekijän tiedot Salary Administration GUI:hin yksitellen. Tuntityöntekijöille syötetään arvio työtuntimäärästä. Syötettyään kaikkien tiedot, käyttäjä voi käydä listaa läpi ja nähdä ohjelman jokaiselle työntekijälle laskemat palkat.

Ongelmakohta: Käyttäjä ei näe mistään palkkojen kokonaissummaa. Etenkin suurempihenkilöstöisten projektien kohdalla henkilöstökulujen laskeminen on vaivalloista.

4.2.3 TOIMITUSJOHTAJA

Tehtäväskenaario: Käyttäjä haluaa tarkastella minkä tyyppisiä työntekijöitä yrityksessä on ja vastaavatko palkkatasot toisiaan.

Käyttöskenaario: Käyttäjä avaa tiedostosta esille valmiin työntekijälistan. Hän avaa kaksoisklikkauksella listasta esille tietoja eri työntekijöistä.

Ongelmakohta: Käyttäjä ei pysty järjestämään listaa työntekijätyypin mukaan, jolloin vertailu on hankalaa.

4.3 EVALUOINTIEN LÖYDÖKSET JA VAIKUTUKSET

Käyttöliittymästä löytyi käytettävyyssongelmia. Työntekijäolioiden käsittely ryhmänä on puutteellista ja dialogi-ikkunoita tarvitaan lisää. Käyttöliittymän on ilmoitettava käyttäjälle virhekäytöstä työntekijän tietoja tallennettaessa ja opastettava käyttäjää oikeaan käyttöön. Palkanlaskentaominaisuuksia on selkeytettävä ja lisättävä tieto kokonaispalkasta kaikille listan työntekijöille.

Nykyisenlaisena käyttöliittymä ei palvele käyttäjiä halutulla tavalla, vaan toiminnot ovat työläitä ja vaativat käyttäjältä erillistä laskemista. Käyttäjä saattaa myös käyttää ohjelmaa virheellisesti saamatta siitä huomausta, jolloin käyttökokemus voi olla hämmentävä.

Suunnitelman tehtäviä parannuksia:

- työntekijän palkan laskemisen selkeyttäminen
- ilmoitus onnistuneesta tallennuksesta
- ilmoitus virheellisestä työntekijän tallennusyrityksestä
- mahdollisuus lajitella työntekijälistaa tyyppin mukaan

5 VIIMEISTELTY SUUNNITELMA KÄYTTÖLIITTYMÄSTÄ

5.1 PERUSNÄKYMÄ

File				
Last name		<u>Last name</u>	First nam..	Pay type
First name(s)		Aalto	Keijo	Hourly
SSN		Ahonen	Marja Riik..	Monthly
Employee type		Koskela	Antti	Salesman
		Metsälä	Kaisa	Hourly
		Virtanen	Liisa	Salesman
Save	Delete			
Clear form		Total salary:		22400

Kuva 13: Viimeistelty suunnitelma perusnäköymästä

Vasemmassa yläkulmassa on valikkopainike ("File"). Vasemmalla puolella on työntekijän tietojen syöttämistä ja tarkastelemista varten lomake, jossa ovat näkyvissä kentät sukunimelle ("Last name"), etunimille ("First name(s)") ja sosiaaliturvatunnukselle ("SSN"). Näiden alapuolella on työntekijätyypin valitsemista varten alasvetovalikko ("Employee type"). Lomakkeen alla on napit työntekijäolion tallentamista ("Save") ja poistamista ("Delete") sekä lomakkeen tyhjentämistä ("Clear form") varten. Oikealla nähdään lista jo luoduista työntekijäolioista. Lista voidaan lajitella sukunimen, etunimien tai työntekijätyypin mukaan. Oikeaan alakulmassa on esillä työntekijöiden yhteenlaskettu palkka ("Total salary").

5.2 TYÖNTEKIJÄTYYPIT

5.2.1 TUNTITYÖNTEKIJÄ

File				
Last name		<u>Last name</u>	First nam..	Pay type
First name(s)		Aalto	Keijo	Hourly
SSN		Ahonen	Marja Riik..	Monthly
Employee type	Hourly	Koskela	Antti	Salesman
Hours done		Metsälä	Kaisa	Hourly
Hourly salary		Virtanen	Liisa	Salesman
Calculated salary				
Save	Delete			
Clear form		Total salary:		22400

Kuva 14: Viimeistelty suunnitelma tuntityöntekijälomakkeesta

Valittaessa alasvetovalikosta tuntityöntekijä ("Hourly"), ilmestyvät lisäkentät tehdystä työtuntimäärästä ("Hours done") ja tuntipalkasta ("Hourly salary"). Näiden alle ohjelma laskee työntekijäolion tallentamisen yhteydessä työntekijän palkan ("Calculated salary").

5.2.2 KUUKAUSITYÖNTEKIJÄ

File				
Last name		<u>Last name</u>	First nam..	Pay type
First name(s)		Aalto	Keijo	Hourly
SSN		Ahonen	Marja Riik..	Monthly
Employee type	Monthly	Koskela	Antti	Salesman
Monthly salary		Metsälä	Kaisa	Hourly
		Virtanen	Liisa	Salesman
Save	Delete			
Clear form		Total salary:		22400

Kuva 15: Viimeisteltty suunnitelma kuukausityöntekijälomakkeesta

Kuukausityöntekijälle ("Monthly") ilmestyy vain lisäkenttä kuukausipalkasta ("Monthly salary").

5.2.3 MYYNTITYÖNTEKIJÄ

File				
Last name		<u>Last name</u>	First nam..	Pay type
First name(s)		Aalto	Keijo	Hourly
SSN		Ahonen	Marja Riik..	Monthly
Employee type	Salesman	Koskela	Antti	Salesman
Monthly salary		Metsälä	Kaisa	Hourly
Outcome claim	<input type="checkbox"/>	Virtanen	Liisa	Salesman
Bonus (%)				
Calculated salary				
Save	Delete			
Clear form		Total salary:		22400

Kuva 16: Viimeistelty suunnitelma myyntityöntekijälomakkeesta

Valittaessa alasvetovalikosta myyntityöntekijä ("Salesman"), ilmestyvät lisäkentät kuukausipalkalle ("Monthly salary") ja bonukselle ("Bonus (%)") sekä tarkistusruutu odotetun tuloksen toteutumisesta ("Outcome claim") varten. Lisäksi, kuten tuntityöntekijälle, ohjelma laskee työntekijäolion tallentamisen yhteydessä työntekijän palkan ("Calculated salary").

5.3 TYÖNTEKIJÄN LISÄÄMINEN

File				
Last name	Peltonen	<u>Last name</u>	First nam..	Pay type
First name(s)	Kalle Veikko	Aalto	Keijo	Hourly
SSN	010166-4321	Ahonen	Marja Riik..	Monthly
Employee type	Monthly	Koskela	Antti	Salesman
Monthly salary	3400	Metsälä	Kaisa	Hourly
		Virtanen	Liisa	Salesman
<div> <div>Save</div> <div>Delete</div> </div>				
Clear form		Total salary:		22400

Kuva 17: Viimeisteltty suunnitelma työntekijän lisäämisestä, vaihe 1

Lomakkeen kentät täytetään työntekijän tiedoilla, jonka jälkeen painetaan "Save"-nappulaa.

File				
Last name	Peltonen	<u>Last name</u>	First nam..	Pay type
First name(s)	Kalle Veikko	Aalto	Keijo	Hourly
SSN	010166-4321	Ahonen	Marja Riik..	Monthly
Employee type	Monthly	Koskela	Antti	Salesman
Monthly salary	3400	Metsälä	Kaisa	Hourly
		Peltonen	Kalle Veik..	Monthly
		Virtanen	Liisa	Salesman
Save		Delete		
Clear form		Total salary:		25800

Kuva 18: Viimeistelty suunnitelma työntekijän lisäämisestä, vaihe 2

Työntekijäolio ilmestyy työntekijälistaan ja työntekijöiden yhteenlaskettu palkka ("Total salary") päivittyy.

Save employee	
Employee already exists. Would you like to rewrite employee data?	
Yes	No

Kuva 19: Dialogi-ikkuna kysyy, halutaanko olemassa oleva olio korvata

Jos lomakkeeseen on syötetty jo olemassa oleva sosiaaliturvatunnus ("SSN"), ilmestyy dialogi-ikkuna varmistamaan, halutaanko jo luotu olio korvata uudella tiedolla.

Save employee	
Please select pay type.	
	Ok

Kuva 20: Viesti työntekijätyypin puuttumisesta

Jos lomakkeesta ei ole työntekijälle valittu tyyppiä ("Pay type"), huomauttaa ohjelma tästä eikä työntekijäolion luominen onnistu.


Save employee	
Employee saved	
	Ok

Kuva 21: Viesti onnistuneesta tallennuksesta

Kun työntekijäolion tiedot on onnistuneesti tallennettu, ohjelma ilmoittaa asiasta.

5.4 TYÖNTEKIJÄN POISTAMINEN

File				
Last name	Peltonen	<u>Last name</u>	First nam..	Pay type
First name(s)	Kalle Veikko	Aalto	Keijo	Hourly
SSN	010166-4321	Ahonen	Marja Riik..	Monthly
Employee type	Monthly	Koskela	Antti	Salesman
Monthly salary	3400	Metsälä	Kaisa	Hourly
		Peltonen	Kalle Veik..	Monthly
		Virtanen	Liisa	Salesman
Save		Delete		
Clear form		Total salary:		25800



Kuva 22: Viimeistelty suunnitelma työntekijän poistamisesta

Työntekijän tiedot on avattu näkyviin lomakkeeseen ja käyttäjä painaa "Delete"-nappulaa, jolloin työntekijäolio poistetaan listasta ja yhteenlasketusta palkasta ("Total salary").

Employee deletion	
Are you sure you want to delete the employee?	
Yes	No

Kuva 23: Dialogi-ikkuna pyytää varmistusta työntekijän poistamisesta

Työntekijää ei poisteta, jollei käyttäjä varmista toimenpidettä dialogi-ikkunan kautta.

Employee deletion	
Employee deleted.	
Ok	

Kuva 24: Dialogi-ikkuna ilmoittaa työntekijäolion onnistuneesta poistamisesta


5.5 PÄÄVALIKKO

File				
New list		<u>Last name</u>	First nam..	Pay type
Save list as...		Aalto	Keijo	Hourly
Load list		Ahonen	Marja Riik..	Monthly
Exit		Koskela	Antti	Salesman
		Metsälä	Kaisa	Hourly
		Virtanen	Liisa	Salesman
Save		Delete		
Clear form		Total salary:		22400

Kuva 25: Viimeistelty suunnitelma päävalikosta

Päävalikosta ("File") löytyvät toiminnot uuden työntekijälistan luomiselle ("New list"), listan nimellä tallentamiselle ("Save list as...") , olemassa olevan listan lataamiselle ("Load list") ja ohjelman sulkemiselle ("Exit").

5.5.1 UUDEN LISTAN LUOMINEN

File				
New list		<u>Last name</u>	First nam..	Pay type
Save list as...		Aalto	Keijo	Hourly
Load list		Ahonen	Marja Riik..	Monthly
Exit		Koskela	Antti	Salesman
		Metsälä	Kaisa	Hourly
		Virtanen	Liisa	Salesman
Save		Delete		
Clear form		Total salary:		22400

Kuva 26: Viimeistely suunnitelma uuden listan luomisesta


Käyttäjä valitsee päävalikosta "New list" –vaihtoehdon. Toiminto tyhjentää käytössä olevan listan.

Create new list			
Would you like to save the current list?			
		Yes	No
		Cancel	

Kuva 27: Dialogi-ikkuna kysyy nykyisen listan tallentamisesta

Jos nykyisessä listassa on jo olemassa työntekijäolioita, ohjelma kysyy, haluaako käyttäjä tallentaa sen ennen uuden listan luomista. Jos vastataan kyllä, pyydetään tallennussijaintia. Käyttäjä voi myöntämisen ja kieltämisen lisäksi perua koko toiminnon ("Cancel").


5.5.2 LISTAN TALLENTAMINEN NIMELLÄ

File				
New list		<u>Last name</u>	First nam..	Pay type
Save list as...		Aalto	Keijo	Hourly
Load list		Ahonen	Marja Riik..	Monthly
Exit		Koskela	Antti	Salesman
		Metsälä	Kaisa	Hourly
		Virtanen	Liisa	Salesman
Save		Delete		
Clear form		Total salary:		22400

Kuva 28: Viimeistelty suunnitelma listan tallentamisesta

Käyttäjä valitsee päävalikosta "Save list as...". Ohjelma pyytää tämän jälkeen käyttäjältä sijaintia, johon työntekijälista tallennetaan

5.5.3 LISTAN LATAAMINEN TIEDOSTOSTA

File				
New list		<u>Last name</u>	First nam..	Pay type
Save list as...		Aalto	Keijo	Hourly
Load list		Ahonen	Marja Riik..	Monthly
Exit		Koskela	Antti	Salesman
		Metsälä	Kaisa	Hourly
		Virtanen	Liisa	Salesman
Save		Delete		
Clear form		Total salary:		22400

Kuva 29: Viimeistelty suunnitelma listan lataamisesta tiedostosta


Käyttäjä valitsee päävalikosta "Load list", jonka jälkeen ohjelma pyytää käyttäjältä listatiedoston sijaintia.

Load list			
This operation will clear the existing list. Are you sure you want to continue?			
	<table><tr><td>Yes</td><td>No</td></tr></table>	Yes	No
Yes	No		

Kuva 30: Dialogi-ikkuna pyytää varmistusta listan lataamisesta

Jos listassa on jo työntekijäolioita, ohjelma pyytää varmistusta toisen listan lataamisesta, sillä operaatio hävittää nykyisen listan tallentamattomat tiedot.

5.5.4 OHJELMAN SULKEMINEN

File				
New list		<u>Last name</u>	First nam..	Pay type
Save list as...		Aalto	Keijo	Hourly
Load list		Ahonen	Marja Riik..	Monthly
Exit		Koskela	Antti	Salesman
		Metsälä	Kaisa	Hourly
		Virtanen	Liisa	Salesman
Save		Delete		
Clear form		Total salary:		22400

Kuva 31: Viimeistelty suunnitelma ohjelmasta poistumisesta

Käyttäjä valitsee päävalikosta "Exit".

Exit	
Are you sure you want to exit the program?	
Yes	No

Kuva 32: Dialogi-ikkuna pyytää varmistusta sulkemisesta

Ohjelma kysyy, haluaako käyttäjä varmasti sulkea ohjelman. Sulkemisen yhteydessä esillä olevassa työntekijälistassa olevat tiedot tallennetaan niin, että ne tulevat automaattisesti näkyville ohjelman seuraavan kerran avattaessa.

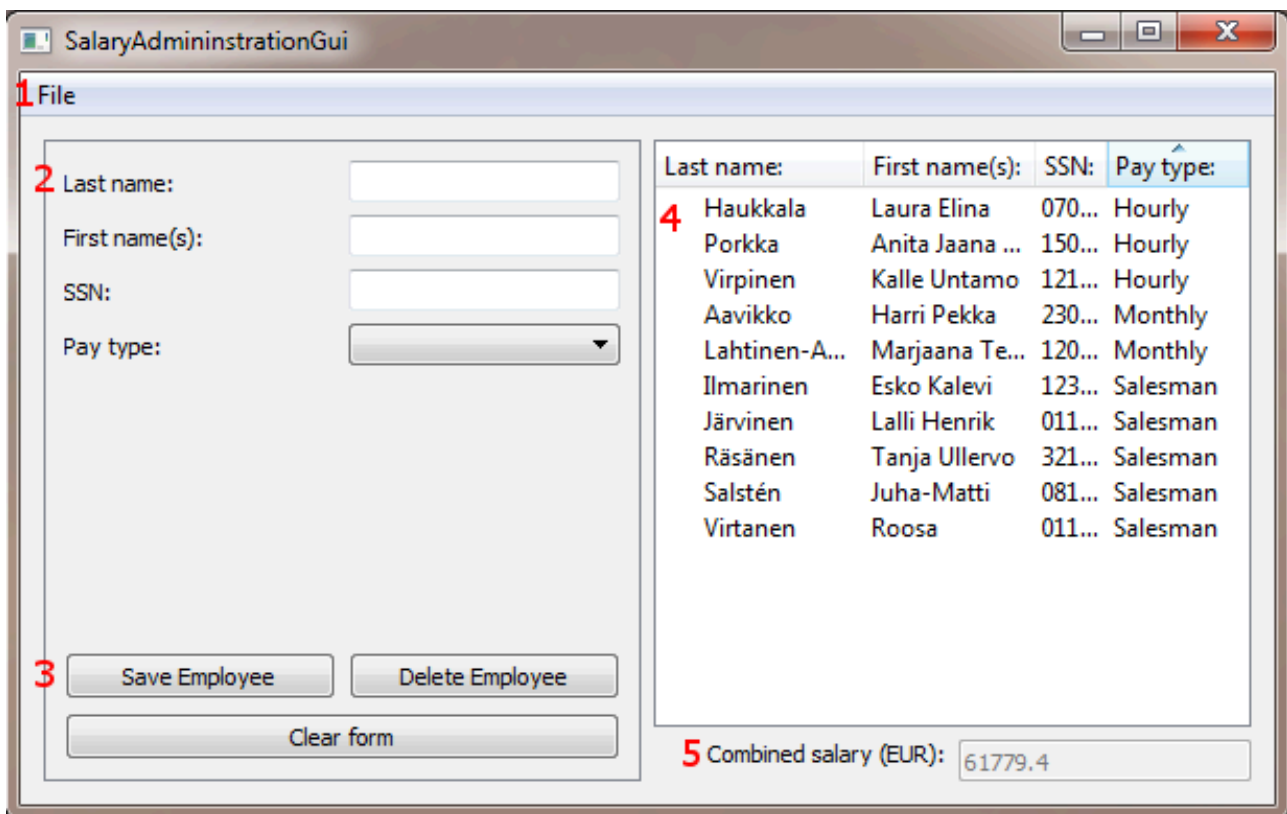
6 KÄYTTÖOHJE

6.1 KÄYNNISTYSOHJEET

Komentoriviltä (Linux/Windows)

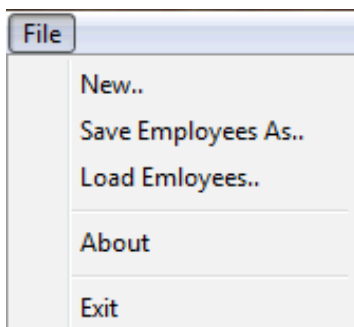
```
>> git clone http://github.com/ekortela/salary-administration-gui.git <hakemisto>  
>> cd <hakemisto>/static  
>> SalaryAdministrationGui
```

6.2 KÄYTTÖLIITTYMÄN KÄYTTÖ



Kuva 33: Ruutukaappaus valmiista käyttöliittymästä

SalaryAdministrationGui on työntekijätietojen hallintaan ja palkanlaskentaan tarkoitettu työkalu. Ohjelma ylläpitää tietokantaa työntekijöistä, jota käyttäjä pystyy hallinnoimaan. Käyttöliittymä koostuu päävalikosta (1), lomakeosioista (2), lomakkeen hallintapainikkeista (3), työntekijälistasta (4) ja työntekijöiden yhteenlasketusta palkasta (5).



Päävalikko (1)

- "New..." tyhjentää nykyisen työntekijälistan (4).
- "Save Employees As..." tallentaa nykyisen työntekijälistan csv-tiedostoon.
- "Load Employees..." lisää nykyiseen työntekijälistaan työntekijöitä tiedostosta.
- "About" avaa ikkunan, jossa on tietoja SalaryAdministrationGui-ohjelmasta.
- "Exit" sulkee ohjelman. Käytössä oleva työntekijälista tallentuu ohjelman muistiin siten, että ohjelman seuraavan kerran avattaessa voidaan jatkaa saman listan käsittelyä ilman, että sitä tarvitsee erikseen ladata tiedostosta.

Hox! Valmiita työntekijälistoja käytettäväksi löytyy kansioista "static".

Lomakeosio (2)

- Yksittäisen työntekijän tietojen kuvaamista ja käsittelyä varten.
- Käyttäjä syöttää työntekijän tiedot lomakkeen kenttiin.
- "Pay type" –alasvetovalikko muuttaa lomaketta lisäämällä sen perään työntekijätyyppikohtaisia kenttiä.

Lomakkeen hallintapainikkeet (3)

- "Save Employee" tallentaa lomakkeessa olevat tiedot olioksi työntekijälistaan (4). Listan työntekijöiden yhteenlaskettua palkkaa (5) kuvaava kenttä päivittyy. Jos tallennettava työntekijä on tyyppiä "Hourly" tai "Salesman", tallennuksen yhteydessä myös lasketaan tälle yksittäiselle työntekijälle palkka, joka tulee näkyviin lomakkeen loppuun.
- "Delete Employee" poistaa lomakkeessa näkyvän työntekijän työntekijälistasta.
- "Clear form" tyhjentää lomakkeen kaikki syöttökentät.

Työntekijälista (4)

- Listassa olevat työntekijäoliot voi lajitella sukunimen, etunimien, sosiaaliturvatunnuksen tai työntekijätyypin mukaan.
- Listan työntekijöiden yhteenlaskettu palkka (5) näkyy listan alla.

POIKKEAMAT SUUNNITELMASTA

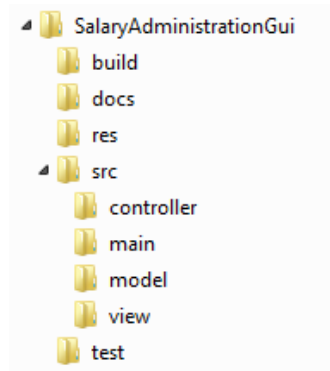
Päävalikon "Load Employees" ei avaa uutta listaa, vaan täydentää tällä hetkellä käytössä olevaa listaa. Nyt on mahdollista siis koota useampi lista yhteen, joka vanhalla suunnitelmalla olisi vaatinut jokaisen työntekijän yksittäistä syöttämistä uudelleen. Vanhanmallinen toiminto on edelleen mahdollista toteuttaa tyhjentämällä ensin käytössä oleva työntekijälista "New..." -toiminnolla ja lataamalla vasta sitten työntekijän "Load Employees" -toiminnon kautta.

Päävalikon "Exit" ei enää kysy käyttäjältä vahvistusta toiminnolle. Koska ohjelman tila tallentuu automaattisesti, ei ohjelmaa suljettaessa häviä tietoja eikä työtä mene hukkaan, joten vahvistaminen oli turhaa.

7 OHJELMISTON RAKENNE

Käyttöliittymän ohjelmointityö osoittautui haastavaksi ja tästä syystä päätettiin ohjelmointisuunnittelun prosessin mukainen tekninen kuvaus, jossa on käyty läpi ohjelmiston suunnitteluratkaisut, luokkadiagrammi, toteutus ja testaus. Tekninen dokumentointi helpottaa myös järjestelmän ylläpitoa, kehitystä ja se auttaa ymmärtämään ohjelmiston rakennetta kokonaisuutena.

7.1 OHJELMISTON HAKEMISTORAKENNE



Kuva 34. Projektin hakemistorakenne.

Taulukko 6. Hakemistojen yksityiskohtaisempi kuvaus.

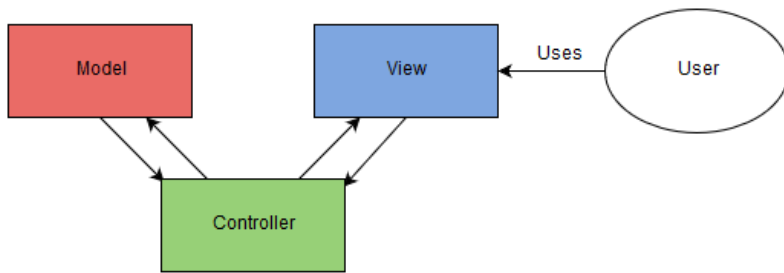
Hakemisto	Kuvaus
SalaryAdministrationGui	Projektin pääkansio ja Qt:n qmake:n vaatimat tiedostot (.pro, .pro.user)
build/	Kääntäjän ja ohjelman ajonaikana luodut tiedostot (obj, .exe, .moc, .log ...)
docs/	Dokumentit ja UML diagrammit (.doc, .jpg, .png, ...)
res/	Ohjelman tarvitsemat resurssitiedostot (.xml ...)
src/	Pääohjelma ja varsinaisen ohjelmiston lähdetiedostot (.h, .cpp, .pro)
tst/	Testausautomaatioon tarvittavat tiedostot (.h, .cpp, .pro)

7.2 TEST-DRIVEN DEVELOPMENT (TDD)

Tässä harjoitustyössä on soveltuvin osin noudatettu TDD-ohjelmistokehityksen periaatteita. Tämä onnistui käyttäen hyväksi Qt-ympäristön sisäänrakennettua *QtTest*-luokkaa, joka sisältää tarvittavan funktionaalisuuden metodien sisään- ja ulostulojen testaamiseen. Yksikkötestit sisältyvät *EmployeeControllerTest* -luokkaan, jolla samanaikaisesti testataan *EmployeeController* sekä *Employee*-tyyppien set- ja get-metodien toimivuus.

7.3 MVC-ARKKITEHTUURI

Yhtenä ohjelmistorakenteen vaatimuksista on se, että käyttöliittymä on helposti korvattavissa, kts. Taulukko 5. Käyttöliittymän muut vaatimukset. Tästä syystä ohjelmistosuunnittelussa hyödynnettiin MVC-arkkitehtuuria (Model-View-Controller), joka on suosittu käyttöliittymäsuunnittelussa, koska kyseinen arkkitehtuuri vähentää ohjelmistokomponenttien välisiä riippuvuuksia, helpottaa koodin ylläpitoa ja järjestelmän siirrettävyyttä.



Kuva 35. MVC-malliarkkitehtuuri.

MVC arkkitehtuurin malli, joka sisältää tietorakenteen, voidaan suunnitella ja testata riippumatta ohjelman muista osista kuten käyttöliittymästä. Tämä parantaa ohjelmiston testattavuutta ja myöhempiä toimintojen lisäämistä.

Kuva 36 esittää *SalaryAdministrationGui* ohjelmiston yksinkertaistetun UML-diagrammin.

7.3.1 NÄKYMÄ (VIEW)

EmployeeView, eli näkymä, määrittää ohjelmiston ulkoasun ja vastaa graafista käyttöliittymää. Tässä harjoitustyössä, käyttöliittymä vastaa virhetilojen raportoinnista käyttäjälle, käyttäjän käskyjen välittämisestä käsittelijälle ja tietokannan tiedon graafisesta esittämisestä. Kaikki kommunikointi tapahtuu käsittelijän kautta, joten mallin ja näkymän välillä ei ole suoraa riippuvuutta, joka tekee käyttöliittymästä helposti korvattavan ilman.

7.3.2 MALLI (MODEL)

Employee, *SalesmanEmployee*, *MonthlyPaidEmployee*, *HourlyPaidEmployee*, eli mallin, sisältävät työntekijöiden luokkahierarkian, eri työntekijätyyppien rakenteet sekä määrittelyt miten jokaisen työntekijätyypin tiedot tallennetaan ja luetaan levyltä. Eri työntekijätyypit perivät pääluokasta *Employee*, joka sisältää kaikille työntekijöille yhteiset attribuutit. Alaluokat sisältävät niille ainutlaatuiset attribuutit sekä datan sarjoitusmenetelmän (eng. serialization), joka määrittää missä järjestyksessä tietorakennetta kirjoitetaan ja luetaan.

7.3.3 KÄSITTELIJÄ (CONTROLLER)

EmployeeController, eli käsittelijä, on rajapinta joka mahdollistaa kommunikoinnin mallin ja näkymän välillä. Tämä kerros välittää näkymän kutsut mallin parametrien päivitykseen ja näkymän pyynnöt mallin tämänhetkisen tilan näyttämiseen. *EmployeeController* sisältää myös mallin tämänhetkisen tilan, joka on vektori johon on tallennettu jokainen luotu työntekijätyypin instanssi (objekti).

7.4 TYÖNTEKIJÄTYYPPIEN LUONTI

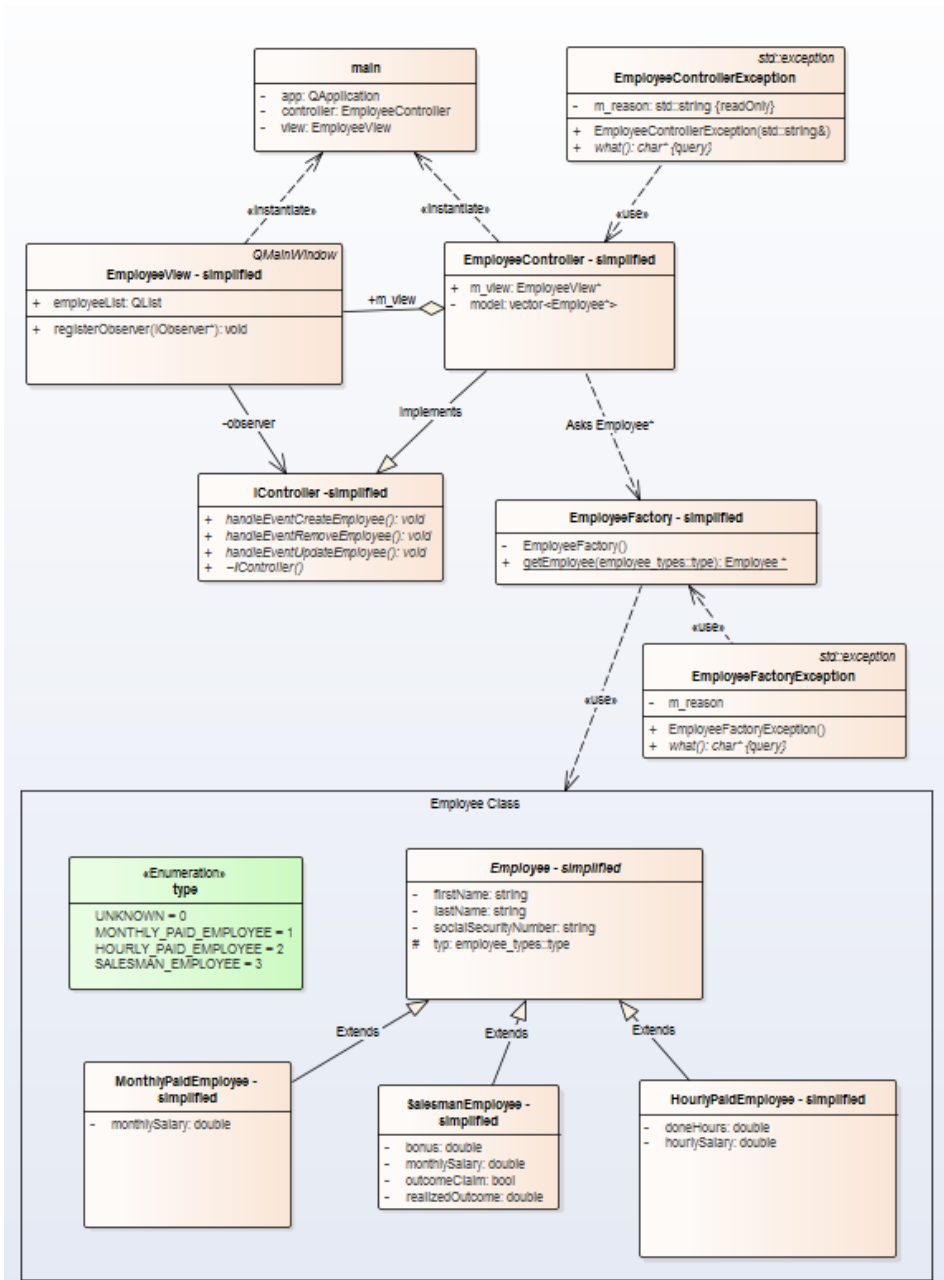
EmployeeFactory sisältää staattisen metodin *Employee* getEmployee(params)*, jolla voidaan luoda uusi *Employee** tyyppin olio. Palautettu olio on *Employee*-tyypin instanssi, joka on valmiiksi alustettu metodille syötetyillä parametreilla.

7.5 TIETOJEN TALLENNUS

Tietojen tallennus ja lukeminen tapahtuu sarjallistamisella. Tämä tarkoittaa, että jokainen työntekijätyypin instanssi muutetaan formaattiin, joka voidaan tallentaa levyille. Tähän kätevin tapa C++ ohjelmoinnissa on `>>` -operaattorin ylikuormitus. Jokainen työntekijätyyppi vaatii oman

implementaation operaattorin ylikuormittamisella, koska jokaisen työntekijätyypin luokalla on erilaiset parametrit. Tiedot tallennetaan tekstitiedostoon, joka voidaan helposti lukea ja tallentaa ohjelman käynnistyksessä, ajon aikana tai lopetuksessa.

7.6 LUOKKADIAGRAMMI (YKSINKERTAISTETTU)



Kuva 36. SalaryAdministrationGui ohjelmiston UML-luokkadiagrammi.

LIITE A Luokkadiagrammi sisältää yksityiskohtaisen kuvauksen ohjelman komponenteista.

8 HARJOITUSTYÖN TEKIJÖIDEN PALAUTUSTIEDOT

Käyttö	Tekijä 1	Tekijä 2
Nimi	Aapo Keskimölä	Elisa Kortela
Opiskelijanumero	1832390	2113762
Tiedekunta / Laitos	LUTK / Matematiikka	TST / Tietojenkäsittelytiede
Aloituvuosi	2004 (valm. 2011 pros.tekn. DI)	2009
Sähköpostiosoite	aapokesk@gmail.com	elisa.kortela@gmail.com

Taulukko 7. Harjoitustyön tekijöiden palautustiedot

LIITE A

LUOKKADIAGRAMMI

