

Ohjelmointi II Harjoitustyö

Henkilötietojen ylläpito sovellus

|  |
| --- |
| Sampo Kajaste ETA20SP |

SISÄLTÖ

[1 Ohjelman kuvaus 4](#_Toc58848129)

[1.1 Toiminnallisuudet 5](#_Toc58848130)

[2 Kuvitettu käyttöohje 5](#_Toc58848131)

[2.1 Toinen formi 5](#_Toc58848132)

[3 Sovelluksen tiedostot 5](#_Toc58848133)

[3.1 Json 5](#_Toc58848134)

[4 Itsearviointi 5](#_Toc58848135)

[4.1 Arvio työhön käytetystä ajasta 5](#_Toc58848136)

[4.2 Haasteet 5](#_Toc58848137)

[4.3 Saavutukset 5](#_Toc58848138)

# **Ohjelman kuvaus**

Tee sovellus, jolla voi pitää yllä yrityksen henkilöstön tietoja ja henkilön toimisuhteen tietoja. Sovelluksen täytyy pitää lokia kaikista tehdyistä toiminteista (kuka teki, mitä teki, milloin teki). Sovelluksen käyttäjän ei tarvitse kirjautua sovellukseen. Lokitiedot tallennetaan tekstitiedostoon selkokielisenä. Henkilöiden ja toimisuhteiden tiedot tallennetaan tiedostoon, jonka voi halutessaan kryptata jotenkin.

Henkilöistä tallennetaan tiedot:

etunimet

sukunimi

kutsumanimi

henkilötunnus (HETU)

kotiosoite:

katuosoite

postinumero

postitoimipaikka

Toimisuhteesta tallennetaan tiedot:

alkamispäivä

päättymispäivä (huom. voi olla myös toistaiseksi voimassa oleva, jolloin päättymispäivää ei täytetä)

nimike

yksikkö (osasto, jossa henkilö työskentelee)

Kaikkia tietoja tulee pystyä muokkaamaan sovelluksessa ja muutokset tulee tallentua tiedostoon. Tietoja täytyy pystyä lisäämään ja poistamaan. Tietoja poistettaessa täytyy käyttäjältä kysyä varmennus tietojen poistolle.

Henkilötunnuksen tulee olla oikeanlainen. Tee sovellukseen tarkastus, joka varmentaa syötetyn henkilötunnuksen.

Sovellus osaa ehdottaa postitoimipaikkaa syötetyn postinumeron perusteella. Esimerkiksi kirjoittamalla postinumeron alkuun merkit 70 ehdottaa sovellus postitoimipaikaksi Kuopio. Ehdotukset luetaan aiemmin syötetyistä tiedoista (eli ensimmäisellä kerralla sovelluksella ei ole tietoja ehdotuksiksi).

Sovellus listaa henkilöt ja listaa tulee pystyä järjestämään kutsumanimen, sukunimen sekä nimikkeen perusteella nousevaan (a - z) ja laskevaan (z - a) aakkosjärjestykseen.

## Toiminnallisuudet

Sovelluksessa on kolme tietuetta, joista kaksi on liitetty ensimmäiseen tietueeseen.

Sovelluksen kolme tietuetta ovat työntekijä, osoitetiedot ja työntekijätiedot.

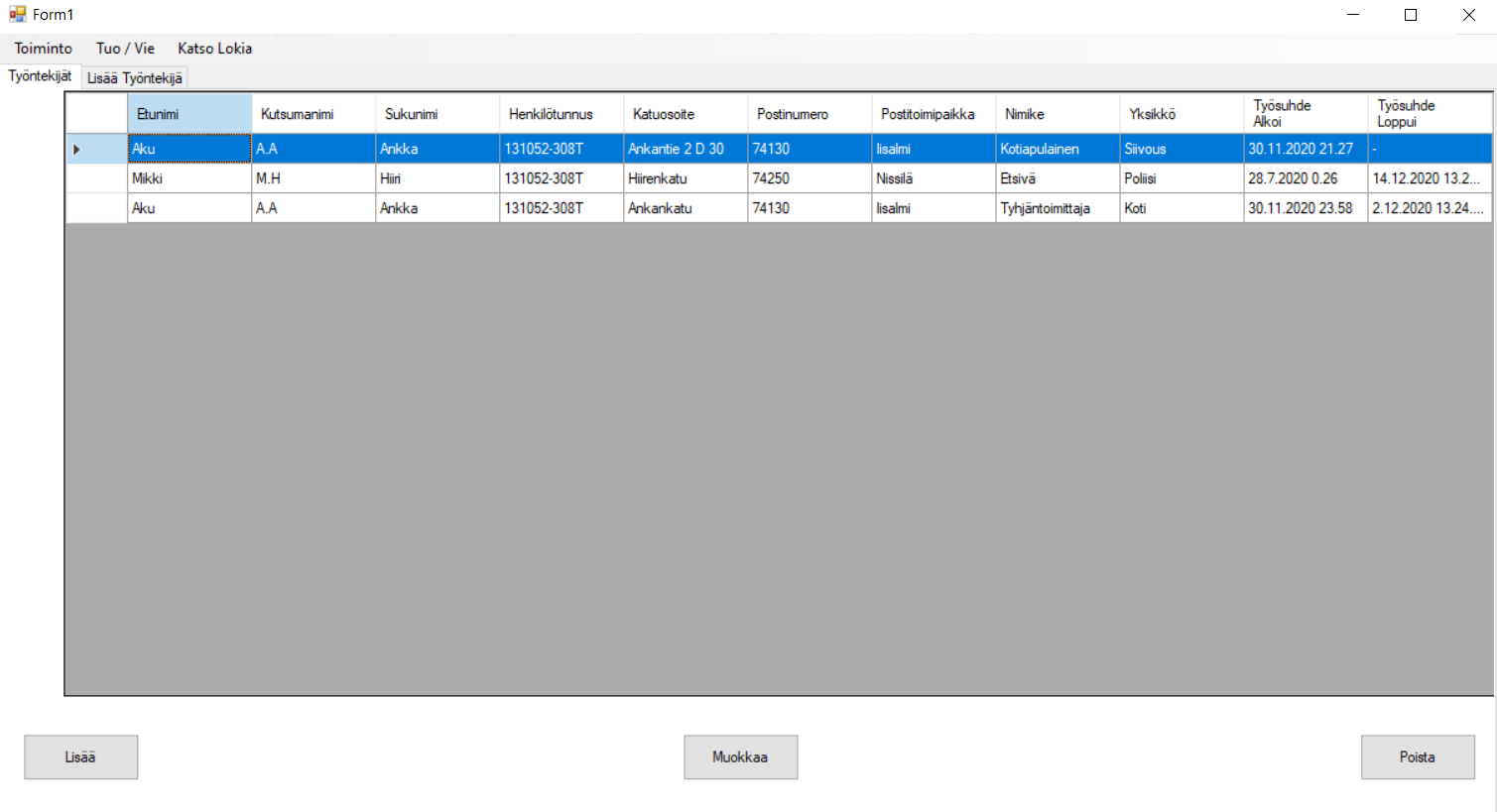
sovelluksen toiminnot ovat:

* kolme tiedoston haku tai tallennus toimintoa
* 3 validointi toimintoa
* 2 erillistä toimintoa
* 2 keypress toimintoa
* 9nappiin yhdistettyä toimintoa
* 1 päivämäärään liittyvä toiminto

# Kuvitettu käyttöohje

## Sovelluksen alkunäkymä:

Sovellus aukeaa työntekijät näkymään, jossa on listattuna työntekijät. Sovellus hakee työntekijät tiedostosta. Sovelluksella voidaan lisätä muokata ja poistaa työntekijöitä. Painamalla ”lisää” tai ”muokkaa” nappeja sovellus avaa ”Lisää työntekijä” välilehden, jossa on täytettävä lomake työntekijän tiedoista. Sovellukseen voidaan tuoda erillisestä tiedostosta työntekijä listat sekä tallentaa erilliseen listaan työntekijä tiedot. Sovellus tallentaa myös loki tiedostoon kaikki toiminnot, jotka käyttäjä tekee työntekijöiden tietoihin.



Sovelluksen alkunäkymässä on

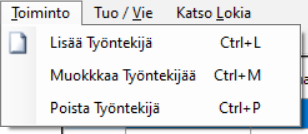
Valikot: Toiminto, Tuo / Vie ja Katso Lokia

Napit: Lisää, Muokkaa ja Poista

Sovelluksessa on kaksi välilehteä ”Työntekijät” ja ”Lisää työntekijä”, joista ensimmäinen on valittuna. Tämä välilehti näyttää työntekijöiden tiedot uudelleen järjestettävässä taulukossa.

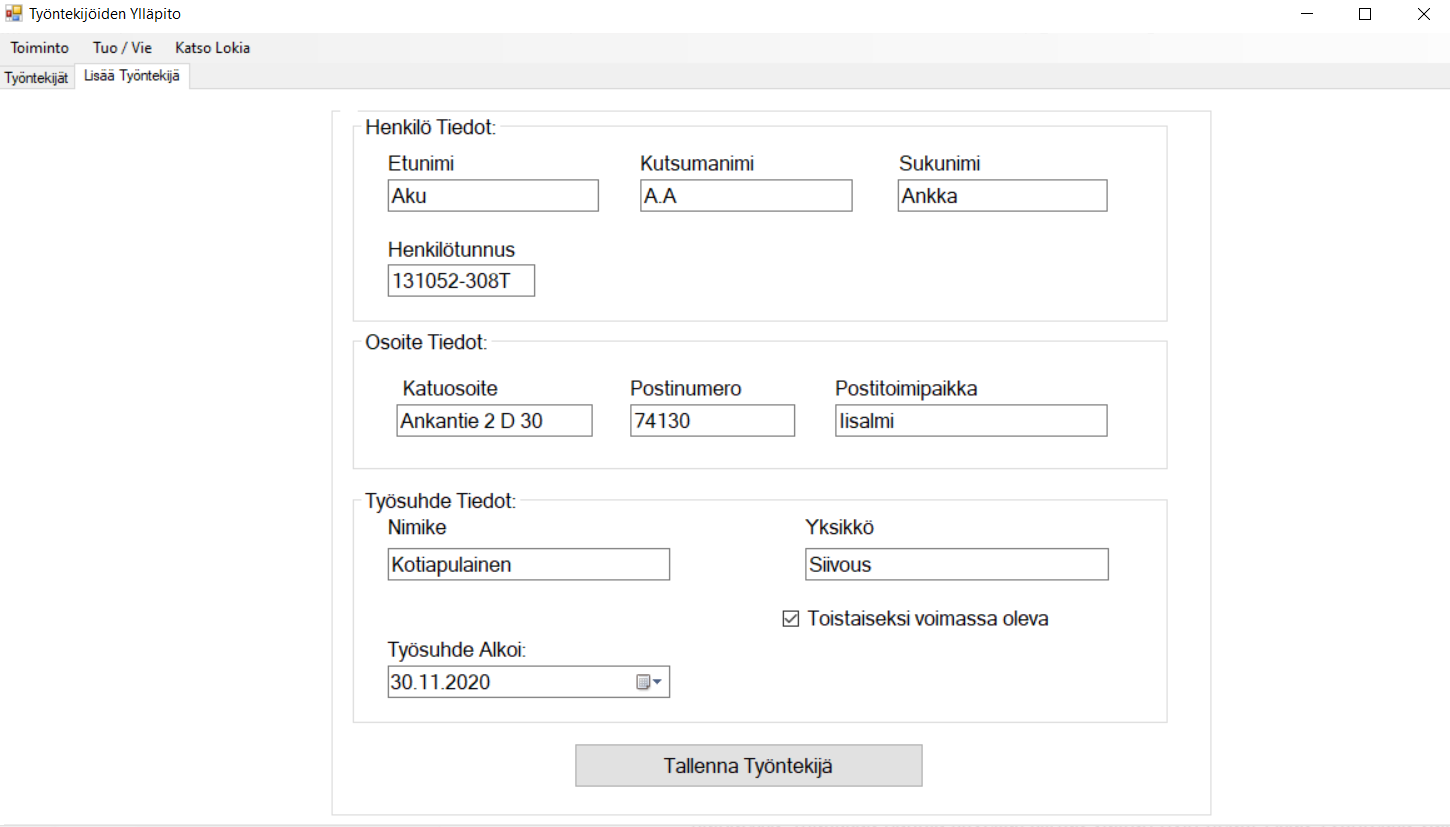
Toimintovalikon painikkeilla sekä sovelluksen alalaidassa olevien nappien toiminnot on yhdistetty samaksi.

”Lisää työntekijä Ctrl + L” nappi, ”Lisää” nappi sekä ”Lisää työntekijä” välilehden painaminen tuovat esiin ”Lisää Työntekijä” välilehden, jolla voidaan lisätä uusi työntekijä. Nappien toiminnallisuudessa on huomioitu, että ”Lisää Työntekijä” välilehdellä kaikki täytettävät kentät tyhjenevät.



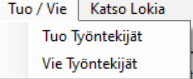
Sovelluksen taulukosta käyttäjä voi valita työntekijä rivin. Rivi maalautuu siniseksi. Käyttäjän painaessa ”muokkaa” nappia sovellus siirtää valitun rivin tiedot ”Lisää Työntekijä” välilehdelle täytettäviin kenttiin muokkaamista varten.



”Poista” nappi toimii samalla periaatteella, mutta nappi ei siirrä tietoja mihinkään vaan kysyy käyttäjältä ”oletko varma, että haluat poistaa työntekijän”. Käyttäjän hyväksyessä sovellus poistaa työntekijä tiedot taulukosta.

Henkilötietojen täytössä on huomioitava, ettei sovellus päästä käyttäjää eteenpäin, ellei jokainen kenttä ole täytetty. Henkilötunnuksen kohdalla on sovellus tarkistaa henkilötunnuksen oikeellisuuden. Postinumero kentässä sovellus tarkistaa aikaisemmin syötetyt toimipaikat ja tarjoaa aikaisemmin syötettyjen perusteella käyttäjälle valmiiksi aiemmin syötettyä postitoimipaikkaa. Postinumerotunnistus toimii vain suomalaisissa viiden numeron postinumeroissa. ”Työsuhde päättyi” kenttä ilmestyy lomakkeelle ”Toistaiseksi voimassa oleva” ruksin pois ottaessa. Lomakkeella on huolehdittu, ettei työsuhteen loppu päivämäärää voi laittaa aikaisemmaksi kuin työsuhteen alku päivämäärä.

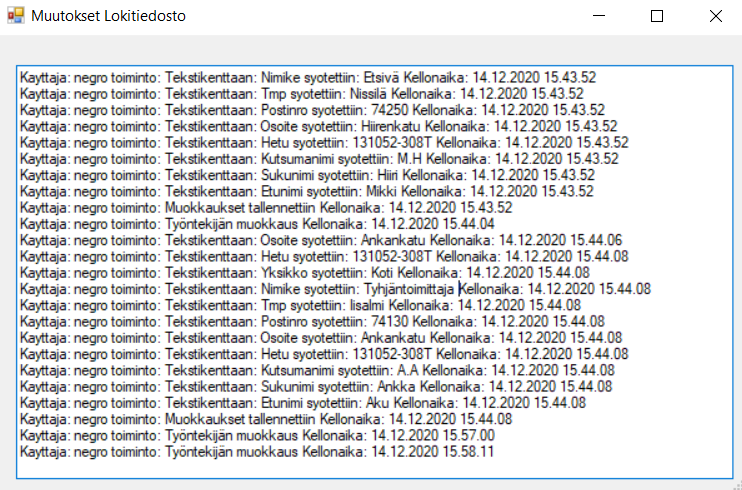
## Sovelluksen valikot:



Sovelluksen valikossa on Tuo / Vie valikko, josta käyttäjä voi tuoda erillisiä työntekijätiedostoja tai tallentaa erilliseen tiedostoon työntekijä tiedot. Tiedot tallentuvat .Json tiedostomuodossa.

Sovellus tallentaa jokaisen napin painalluksen sekä tekstimuutoksen loki tiedostoon. Sovellukseen on tehty erillinen Katso Lokia nappi tiedoston helpompaan tarkasteluun.

## Toinen sovellus ikkuna



Sovelluksen valikko nappia ”Katso Lokia” painamalla sovellus avaa uuden ikkunan, jossa on käyttäjien tekemät muutokset, jotka tuodaan log.txt tiedostosta käyttäjän nähtäväksi.

# Sovelluksen tiedostot

## .Json

Sovellus tallentaa työntekijöiden tiedot tietueet .Json tiedostoon

## .txt

sovellus tallentaa käyttäjien tekemät toiminnot log.txt tiedostoon

# Itsearviointi

## Arvio työhön käytetystä ajasta

Työntekemiseen kulunutta aikaa on vaikea arvioida tarkasti. Olen tehnyt päivässä sovellukseen ne toiminnallisuudet, jotka eivät ole vaatineet selvittelyä tai opiskelua. Sovelluksessa on ollut useampi toiminnallisuus, joita en ole suoriltaan saanut toimimaan. Arvioisin että jokaiseen toiminnallisuuteen, jota olen joutunut miettimään tai opettelemaan, ovat kukin vieneet noin 6 tuntia internetin selailua. Opeteltavia toiminnallisuuksia on ollut neljä. Sovelluksen tekemiseen on siis kulunut noin 4 päivää. Lisäksi olen hionut ja testannut sovellusta ongelma tilanteiden varalta sekä korjannut näitä vihreitä. Virheiden korjauksiin on kulunut noin puolen päivän verran. Dokumentaation olen tehnyt päivässä. Kaiken kaikkiaan sovellukseen on mennyt 5 päivää.

## Haasteet

Todellisia haasteita sovelluksen toiminnan kannalta on ollut neljä eri toimintoa. Suurin ja ehkä hankalin vaihe sovelluksen suunnittelussa on ollut tehtävän annon ymmärtäminen ja toimintojen sovittaminen toisiinsa. Kun tehtäväksi annossa ei selkeästi määritetä nappeja ja niiden toimintoja on oma hommansa suunnitella sovellukseen nappien paikat ja toiminnot. Ei varsinaisesti ohjelmointi ongelma, mutta ongelma kuitenkin. Tämä ei ehkä ole suurin vahvuuteni ohjelmoinnissa. Näkisin enemmän koodin ymmärtämisen ja yhteensovittamisen omana vahvuutenani. Henkilötunnuksen validointia jouduin huikan enemmän miettimään ja tutkimaan. Kaksi seuraavaa ongelmaa olivat liitännäisiä gridviewin rivivalintaan. En saanut muokkaus ja poisto nappeja toimimaan, koska Data Table yhdistettynä listaan sekä gridviewiin. Näistä piti selvittää mikä komponenteista oikeastaan valitsee rivin ja mihinkä se tulisi tallentaa, jotta tiedot saadaan siirrettyä suoraan muokkausta varten työntekijää varten tarvittaviin teksti kenttiin. Viimeisenä koodaus ongelmana oli tekstitiedostoon tallentaminen.

## Saavutukset

Saavutuksen tunteet sovellusta tehdessä ovat olleet moninaiset. Suurimman tyydytyksen olen saanut henkilötunnuksen validointia tehdessä. En vain saanut mistään alkuun kiinni miksi osa henkilötunnuksista toimi ja miksi osa ei. Jossain vaiheessa sovellus lisäsi numeroita henkilötunnuksen tarkiste numeroon, niin ettei tarkiste tunnus enää toiminut oikealla henkilötunnuksella. Vika lopulta löytyi. Vika ei ollut suorananisesti omassa koodissani vaan siinä, etten tiennyt ettei henkilötunnuksissa käytetä I, G, O, Q tai Z kirjaimia. Tämän selvitettyäni oli helppo rakentaa if- rakenne, jolla varmenne toimi oikein. Tässä olisi toki voinut käyttää Switch:ä, mutta näin if- else rakenteen paremmaksi toteutukseksi. Gridvievin Currentrow toimintoa jouduin selvittelemään laajemmin. Kysyin jopa opettajalta neuvoa, kun ei jostain syystä tämä auennut. Löysin internetistä useammankin tavan valita tuo rivi ja tuoda se listaan, mutta jokaisessa mallissa huomasin olevan jotain outoja komentoja, joita en ymmärtänyt. Näitä kun lähdin selvittämään, putosin sellaiseen kanin koloon, ettei loppua näkynyt. Sain toimimaankin tuon koodin internetistä kopioidulla koodipätkällä. Poistin sen, koska en ymmärtänyt mitä koodi teki. Lopulta löysin vastauksen, jota olin etsinyt. Vastaus löytyi tuntiharjoituksista. Selvittelyn määrään nähden uskallan sanoa, että ymmärrän huomattavasti paremmin tuon gridviewin toiminnoista. Tämä on oppimiseni kannalta varmaan se pääasia. Koodissa tuli vastaa Nullrefrence kaksi kertaa. En osaa sanoa korjasinko tuon ongelman oikeaoppisesti, mutta järjestelmä ei enää ainakaan heitä ongelmana tätä.

## Mitä opin

Harjoitustyön tekeminen opetti useammaltakin osin minua ohjelmoinnista. Suurin osa sovelluksen toiminnasta ja toiminnoista olen jo oppinut oppitunneilla, mutta harjoitustyö vahvisti tätä osaamista ja itseäni haastamalla jouduin oppimaan uutta. Hyvänä esimerkkinä on struktien linkittäminen. Vaikka sovelluksen olisi voinut rakentaa yhden struktin avulla tein sovelluksen kolmella struktilla opettelun vuoksi. Login katselu ominaisuus oli myös oma ideani. Tämän avulla sain opetella useamman formin linkittämistä, Vaikka vain vähäisillä tiedoilla. Samalla Loki tietojen katselu nappi toimitti testauksen avustetta. Ei suuri lisätty toiminnallisuus mutta lisätty toiminnallisuus kuitenkin.