

# CT60A2411 Olio-ohjelmointi

## Harjoitustyö - Dokumentaatio

Petteri Mäkelä

0506167

30/12/18

Tämä on kurssin CT60A2411 Olio-ohjelmointi harjoitustyön dokumentaatio.

Harjoitustyössä tuli tehdä android applikaatio valitulla, tai omavalintaisella aiheella. Valitsin annettujen aiheiden joukosta aiheekseni Nuorisotila -applikaation. Applikaatio on toimiva kokonaisuus nuorisotilojen hallintaan.

Ohjelman suunnittelu aloitettiin kynällä ja paperilla. Alustavat ideat käyttöliittymän ulkoasusta, käytetyistä luokista, sekä ohjelman eri rajapinnoista kirjattiin ylös. Ideoista viimeisteltiin mallit, joita ohjelmoinnissa lähdettiin toteuttamaan. UML-kaavioita ohjelman toiminnasta ei luotu, sillä uskoin toiminnallisuuksiin tulevan muutoksia. Ohjelman eri versioiden aikana siihen toteutettiin muutoksia ja laajennoksia, jotka eivät vielä suunnitteluvaiheessa tulleet ilmi.

Ohjelman toteutus aloitettiin käyttöliittymästä. Ohjelman päänäkömät luotiin, jonka jälkeen näkömien toiminnallisuuksia alettiin valmistamaan. Toiminnallisuudet aloitettiin eri apuluokkien, kuten tapahtumalista -luokan toteuttamisella. Tämän jälkeen yksi toiminnallisuus kerrallaan applikaatiota laajennettiin, kuitenkin pitäen toimivan kokonaisuuden aina toimivana.

Työn sekä dokumentaation tekemiseen kului aikaa noin 30 tuntia. Ajasta suurin osa kului itse tekemiseen, eli ohjelmointiin ja ongelmanratkaisuun. Ajasta vain pieni osa kului suunnitteluun ja dokumentaation toteuttamiseen.

Mitä opin harjoitustyöstä?

Harjoitustyö oli ensimmäinen olio-ohjelmoinnin projekti, joka käsitteli useita luokkia ja toiminnallisuuksia, sekä näiden rajapintoja. Sen lisäksi, että isompana projektina se opetti projektin hallintaa, se myös opetti olio-ohjelmoinnille ominaista ajattelutapaa. Käyttöliittymälle ominaisia asioita, olivat eri käyttäjien hallitseminen, android tiedostonhallinta, sekä usean käyttöliittymän käyttö ohjelmassa. Javalla työskentelylle ominaisia asioita olivat erityisesti olioiden suunnittelu-, ja toteuttamisperiaatteet. Esimerkiksi eri näkvyvyysluokkien tarkoitukset ja käyttötavat tulivat työn aikana erittäin tutuiksi.

Applikaatio käynnistyy Login -valikkoon, josta on mahdollista kirjautua eri käyttäjillä ohjelmaan, muuttaen käyttökokemusta halutuin parametrein. Ohjelmaan on toteutettu Guest, Super, sekä Admin -käyttäjät, jotka kaikki periytyvät abstraktista käyttäjä luokasta. Eri käyttäjiin kirjaudutaan käyttäjätunnuksen avulla. Kirjautumisvalikkoon on toteutettu myös salasana -ikkuna, mutta sen toiminnallisuutta ei ole toteutettu.

Ohjelman päänäköymä on tapahtumanäköymä, EventActivity. Päänäköymä näyttää käynnissä olevan tapahtuman, sekä mahdollistaa käyttöliittymässä siirtymisen. Näköymästä pääsee palautteeseen, sekä menuun. Info painike näyttää tapahtumalle osoiteut lisätiedot.

Palautteessa on tekstilaatikko, sekä arvostelutähdet. Näköymä mahdollistaa siis palautteen, ja arvosanan antamisen. Palaute annetaan käynnissä olevalle tapahtumalle.

Päänäköymän menuun on toteutettu ohjelman muut toiminnallisuudet. Guest -käyttäjälle menu mahdollistaa ulos kirjautumisen. Super -käyttäjälle on toteutettu tämän lisäksi tapahtumien hallintajärjestelmä. Admin -käyttäjälle ei ole toteutettu ylimääräisiä toiminnallisuuksia, mutta menuun on lisätty esimerkkitoiminnallisuus, joka näkyy vain Admin käyttäjälle.

Tapahtumalistauksessa on näkyvissä kaikki tallennetut tapahtumat. Näköymästä on mahdollista käynnistää tapahtuma, joka siirtää tapahtuman listan kärkeen, tehden siitä näin ”tuoreimman” tapahtuman. Näköymässä on myös mahdollista poistaa valittu tapahtuma, sekä lisätä uusi tyhjä tapahtuma. Muokkaa/näytä painikkeella pääsee tapahtumien muokkausnäköymään, josta on mahdollista nähdä kaikki tapahtuman tiedot, sekä muokata niitä. Muokkausnäköymästä näkyy myös tapahtumalle annetut palautteet.

Ohjelmaan on toteutettu XML -tiedostoon tallentamisen, sekä tiedoston lukemisen mahdollisuus. Tällä hetkellä ohjelma päivittää XML tiedostoa aina, kun käyttäjä menee päänäköymään. Tämä tekee tiedon tallentamisesta automatisoitua helpottaen käyttöä. Ohjelman tehokkuus tuskin kärsii XML-tiedostojen rakentamisesta.

### Toteutettu virheittä

Perusvaatimukset (= 13 pistettä harjoitustyöstä)

Hallintaluokka (Controller), joka kommunikoi datan ja graafisen käyttöliittymän välillä (MVC pattern). Usein hallintaluokka on MainActivity.java eli luokka, joka ajaa ohjelman ja hallitsee kaikkia toimintoja

Luokka, joka pitää kirjaa Nuorisotila tapahtuma -olioista

Nuorisotila tapahtuma -olio, joka sisältää osoitetiedot, päivämäärän, alku ja loppu kellonajat, ikähaarukan, kävijämäärän, palaute-oliot

Palaute-olio, joka pitää sisällään palautetekstin, palautteen antajan (oletuksena anonyymi), sekä tapahtuman arvion (1-5) tai (sadface - happyface)

Luokka, joka lukee ja tallentaa tiedot nuorisotila tapahtumista XML tai JSON tiedostomuodossa

Kaikkien tietojen mahdollinen muokkaus jälkikäteen

Mahdollisuus tarkastella tapahtumia

Mahdollisuus käynnistää tapahtuma. Tapahtuman käynnistyessä voi:

Muokata kävijämäärää helposti

Kirjoittaa palautteita tapahtumalle

### Toteutettu hyvin

Perusvaatimukset (= 13 pistettä harjoitustyöstä)

-

Laajennetut vaatimukset (= 26 pistettä harjoitustyöstä)

Ohjelma on rakennettu hyvin suunnitelluista UI-komponenteista (helppokäyttöinen, luonteva, toimii erilaisilla mobiililaitteilla kuten älypuhelin tai tabletti)

Kirjautuminen applikaatioon

### Toteutettu perustasolla

Perusvaatimukset (= 13 pistettä harjoitustyöstä)

-

Laajennetut vaatimukset (= 26 pistettä harjoitustyöstä)

Uusien työntekijöiden (käyttäjien) luominen (Tietojen tallennus järkevästi jonnekin)

Ohjelma on rakennettu niin, että 40 pisteen vaatimuksiin voidaan laajentaa, mutta toiminnallisuuksia ei ole toteutettu.

Pisteet 25/40. Perusvaatimukset on toteutettu hyvin, ja Laajennetut vaatimukset melko hyvin. Ohjelma, sekä dokumentaatio on pyritty toteuttamaan laadukkaasti. Harjoitustyön työmäärä yksin tehtynä on kohtalaisen suuri, joten pisteitä voisi tulla työmäärän puolesta enemmänkin.

UML Kaavio, josta käy ilmi käytetyt oliot, sekä niiden relaatiot ohjelman toiminnallisuudessa

