

Olio-ohjelmointi

**Projekti**

# **Finnkino app**

LUT Yliopisto

Tietotekniikka

Miko Mattila, Teemu Viitanen & Otto Oikarinen

Olio-ohjelmointi CT60A2411

Kevät 2022

# Kuvaus ohjelmasta

## Sovelluksen

## repository

Linkki projektin Github repositoryyn: [https://github.com/tviitanen/OOP\\_project\\_repo](https://github.com/tviitanen/OOP_project_repo)

## Demovideo

Demovideolla esitetään koko sovelluksen toiminnallisuus ja käyttö.

Linkki videoon: <https://streamable.com/apolc1>

## Elokuvahaku Finnino rajapinnasta

Finninon rajapinnasta tiedonhaku on toteutettu suoraan kurssin opetusvideoilla esitetyllä tavalla. Aluksi rajapinnasta haetaan XML-muotoisena tietona kaikki saatavilla olevat teatterit ja kun käyttäjä valitsee jonkin teatterin, haetaan rajapinnasta kyseisen päivän kaikki esitykset. Käyttäjällä on myös mahdollisuus asettaa lisähakukriteerejä, jotka ovat tietyn päivän ja tietyn kellonajan mukaiset haut jolloin, esitettävästä tiedosta suodatetaan sopimattomat tiedot pois. Päivämäärän ja kellonajan valintaan käyttäjälle tarjotaan myös mahdollisuus syöttää tiedot graafisten käyttöliittymäkomponenttien: 'calendar' ja 'timepicker' avulla. Tiedot esitetään käyttäjälle 'recyclerview' näkymän avulla, josta käyttäjä näkee elokuvan nimen ja alkamisajankohdan. Sen lisäksi käyttäjä voi valita näkymästä yksittäisen elokuvan ja katsoa sen tarkempia tietoja haettuna IMDB-rajapinnasta.

## Suosikit ja tallennus XML-tiedostoon

Sovelluksessa on mahdollisuus tallettaa omat elokuvasuosikit paikallisesti puhelimelle siten, että tiedot ovat talletettuna XML-tiedostoon, jossa ne säilyvät vaikka sovelluksen sulkisi. Käyttäjän avatessa suosikit näkymän, esitetään käyttäjälle suoraan omat sen hetkiset suosikit. Sivulla on myös painike, josta suosikkeja pääsee vaihtamaan. Vaihdettaessa suosikkeja, sivulle syötetään oma käyttäjänimi ja suosikkielokuvat, jotka talletetaan tallenna-painikkeella.

## Asetukset ja tallennus XML-tiedostoon

Asetus-sivu, joka aukeaa sovelluksen oikeasta yläkulmasta, toimii periaatteeltaan hyvin samalla tavalla kuin suosikit. Asetukset-sivulta käyttäjä voi muokata asetuksia, jotka niin ikään talletetaan omaan tiedostoonsa talteen myös seuraavaa käynnistyskertaa varten. Tällä hetkellä asetukset sivulla ainut toimiva asetus on 'Darkmode' -asetus, joka muuttaa sovelluksen teeman Androidin oletus tummaksi teemaksi. Sovellukseen syötetyt asetusvalinnat näkyvät myös asetussivulla oikeina valintoina. Sen lisäksi muuttamalla Android-järjestelmän kielivalintaan Suomi/Englanti, voidaan valita sovellukseen eri kieli. Itse asetussivulla oleva 'languages'-pudotusvalikko ei ole oikeasti vaihda kieltä vaan siinä näkyy ainoastaan toiminnallisuus.

## Elokuvan tietojenhaku IMDB rajapinnasta

Kun elokuvat on haettu, voi käyttäjä tarkastella elokuvia halutessaan tarkemmin. Tämä tapahtuu hakusivulla klikkaamalla haluamaansa elokuvaa. Sovellus hakee sitten IMDB:n API:sta tarvittavat lisätiedot Finnkinon XML-hausta saadun elokuvan alkuperäisen nimen perusteella. IMDB:n API:n haku tapahtuu seuraavasti: ensin sovellus tekee haun IMDB:stä löytyviin elokuviin siihen tarkoitettuun erilliseen osoitteeseen. Tämän tarkoitus on saada elokuvan ID, joka poikkeaa Finnkinon ID:stä. Kun tämä ID on haettu, voidaan sen avulla toteuttaa tarkempi kysely juuri siihen elokuvaan, myös erilliseen osoitteeseen. Tämän haun tuloksena saamme JSON-tiedoston, jota lukemalla saamme haluamamme tarkemmat tiedot elokuvasta. Nämä tiedot tallennetaan sitten Movie-olioon, joka kopioidaan uuteen aktiviteettiin, josta ne voidaan sitten siirtää elementteihin, niin, että ne näkyvät näytöllä.

## Käyttäjän rekisteröinti ja autentikointi Firebase tietokannalla

Sovellus sisältää käyttäjän sisäänkirjautumisen fragmentissa, josta sisäänkirjaututtua avautuu käyttäjän tiedot aktiviteettiin. Aktiviteetin tiedot haetaan firebase tietokannasta, jonne ne on tallennettu käyttäjän rekisteröitymisen yhteydessä. Käyttäjän rekisteröinti on implementoitu Googlen Firebasen tarjoamalla autentikoinnilla. Firebase salaa käyttäjän salasanan SCRYPTillä ja tallentaa sen salattuna tietokantaan. Tietokantaan tallennetaan myös käyttäjätunnus (sähköposti), käyttäjän nimi ja salasana.

Käyttäjän antamat syötteet tarkastetaan rekisteröinnin yhteydessä. Kaikki tiedot tulee olla syötetty, ja jos ei ole niin ohjelma ilmoittaa puuttuvista tiedoista/virheistä syötteessä. Sähköpostin tulee olla muotoa @domain.xyz. Salasanat tulee sisältää vähintään 12 merkkiä pitkä ja sisältää ainakin yksi iso ja pieni kirjain, numero ja erikoismerkki. Salasanat tarkistus on suoritettu .matches metodilla ja regexillä. Kaikista puutteista salasanassa tulee erillinen ilmoitus, eli käyttäjälle kerrotaan mitä puuttuu. Käyttäjän syöttämä syntymäpäivä yritetään parsia päivämääräksi ja ilmoitetaan käyttäjälle, mikäli syöte on virheellinen/väärässä muodossa. Päivämäärän syöttökenttää tuplaklikkaamalla avautuu kalenteri ikkuna josta voi valita päivämäärän.

Sisäänkirjaututtaessa käyttäjän syötteet tarkastetaan Firebase tietokannassa ja käyttäjälle ilmoitetaan toastilla, onnistuiko kirjautuminen vai ei. Onnistuneen sisäänkirjautumisen jälkeen avautuu uusi ikkuna, jossa käyttäjää tervehditään ja näytetään hänen tietonsa (nimi, sähköposti ja syntymäpäivä). Käyttäjälle ilmoitetaan, jos tietojen haku tietokannasta ei onnistunut. Käyttäjä voi kirjautua ulos painikkeella, jolloin palataan takaisin sisäänkirjautumisfragmenttiin ja käyttäjälle ilmoitetaan hänen kirjautuneen ulos.'

## **Palautteen lähetys**

Painamalla fragmenttien flotatingActionButtonia avaa sähköposti applikaation uutena intenttinä parametreinaan sähköposti ja aihe. Automaattisesti sähköposti ikkuna avautuu sähköpostiosoitteeseen "support@finnkino.fi" ja aiheena on "Feedback".

## **Useampi kieli**

Sovellusta on mahdollista käyttää useammalla kielellä. Tämä toteutettiin käyttämällä Androidin valmiita ominaisuuksia lokalisaatioon. Käytännössä kaikki sovelluksesta löytyvä teksti laitettiin strings.xml tiedostoon, jolle tehtiin suomenkielinen variantti. Nämä tekstit käännettiin sen jälkeen suomeksi. XML-elementteihin ne korvattiin käyttäen tyyliä "@string/tämäElementti" ja koodin sekaan tyyleillä "R.string.tämäElementti" tai

“getText(R.string.tämäElementti)”, jolloin sovellus hakee siihen automaattisesti oikean tekstin sen perusteella, mikä on puhelimen oletuskieli.

## **Tekijät**

### **Miko Mattila**

Sovelluksen lähtökohtana toimi kurssin viikkotehtävissä tekemäni Finnkinso sovellus, jota kuitenkin täytyi parannella huomattavasti nykyiseen nähden. Finnkinso haussa parannuksia olivat hakukriteerien toteuttaminen sekä toiminnallisemman ja visuaalisemman käyttöliittymän toteuttaminen. Kaiken kaikkiaan sovelluksen näkymistä loin asetukset, Finnkinso-haku sivun ja suosikit. Sovelluksen asetukset ja suosikit ovat toiminataperiaatteeltaan hyvin samanlaisia ja melko kevyesti toteutettuja näkymiä. Kummassakin tiedot talletetaan ja luetaan omasta XML-tiedostostaan ja esitetään oman aktiviteettinsa puolella. Läheskään kaikkea en tietysti tehnyt yksin vaan myös muut ryhmäläiset tekivät esimerkiksi käyttöliittymiin parannuksia ja paikoittain kierrätettiin jo toisaalla toimivaksi todettua koodia.

### **Otto Oikarinen**

Implementoin haun IMDB:n API:sta. Aktiviteetin toteuttaminen IMDB-tuloksille. Lokalisatio sovellukselle, sisältäen tekstien kääntämisen, niiden korvaamisen tiedostoissa sekä elementeissä. Lokalisation myötä tein myös hiomista käyttöliittymissä, jotta tekstit näkyvät järkevästi kaikilla kielillä.

### **Teemu Viitanen**

Käyttäjän rekisteröityminen/luominen tietokantaan, sisäänkirjautuminen ja käyttäjän tietojen näyttäminen aktiviteetissa. Firebase tietokannan implementointi ohjelmaan. Salasanan vahvuuden tarkistus ja loput näihin liittyvät ominaisuudet. Palautteen lähetys. Kello ja kalenteri aikaa ja päivämäärää valitessa elokuva haussa sekä päivämäärän valinta kalenterista rekisteröityessä syntymäpäivää valitessa.

## Ohjelman toteutus

Luokkakaavio on piirretty Dia-ohjelmistolla. Yhteydenpitoon ryhmän jäsenten välillä on käytetty Discordin viestikanavaa ja puhelutoimintoa. Dokumentaatio on luotu .docx muodossa Microsoft Wordilla. Lähdekoodin versionhallintaan käytimme Github-versionhallintapalvelua. Luonnollisesti ohjelmistokehityksessä käytimme Android Studiota.

## Käytettyjä lähteitä:

- Finnkino API: <https://www.finnkino.fi/xml/>
- IMDB API: <https://imdb-api.com/>
- Androidin dokumentaatio: <https://developer.android.com/docs>
- Firebase-tietokanta: <https://firebase.google.com/docs>

## Kirjastot

Javan ja Androidin kirjastojen lisäksi käytimme alla olevia:

```
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
```

## Luokkakaavio

Luokkakaavio on sen verran iso, että sen löytää alla olevasta linkistä:

[https://lut-my.sharepoint.com/:i:/g/personal/otto\\_oikarinen\\_student\\_lut\\_fi/EQMRiTsL4BOmTWXs\\_sEGbEBIZL2nPGhEYHBCwFX---Xg?e=oQkoLK](https://lut-my.sharepoint.com/:i:/g/personal/otto_oikarinen_student_lut_fi/EQMRiTsL4BOmTWXs_sEGbEBIZL2nPGhEYHBCwFX---Xg?e=oQkoLK)

## Toteutetut ominaisuudet

Ominaisuus	Perustelut	Pisteet
Finnkino elokuvahaku	Pakollinen	
Tietojen (suosikit) tallennus XML-tiedostoon	Pakollinen tietojen talletus tiedostoon + suosikit toiminnallisuus	3
Sisäänkirjautuminen	Sisäänkirjautuminen Firebase autentikaatiolla	3
Sovelluksella voi olla useampi käyttäjä (+ luominen ja tietojen talletus)	Käyttäjien luominen Firebase tietokantaan ja tietojen talletus	3
Salasanan talletus salattuna	Salasana salataan Firebase tietokannassa SCRYPTillä	2
Salasanan vahvuuden tarkistus	Vähintään 12 merkkiä pitkä, sisältää vähintään yhden ison ja pienen kirjaimen, numeron sekä erikoismerkin	2
Palautteen lähetyk	Avaa nappulasta sähköpostin (gmail) osoitteeseen "support@finnkino.fi" ja aiheena "Feedback"	2
Käyttäjän tiedottaminen	Toastit, errorit EditText kentissä	2

Käyttäjän perustietojen talletus ja käyttö	Käyttäjän tietojen (nimi, ikä, sähköposti) tallennus tietokantaan ja näyttäminen ruudulla	2
Kalenterin ja kellon käyttö ajan/päivämäärän valintaan	Käyttäjälle avautuu kalenteri ikkuna päivämäärää valitessa (tuplaklikkaus), yhdellä klikkauksella manuaalinen syöttö. Kello avautuu kellonaikaa syötettäessä (tuplaklikkaus).	5
Fragmenttien hyödyntäminen aktiviteettien sijasta käyttöliittymiä rakennettaessa	Ohjelmassa käytetään sekä aktiviteetteja että fragmentteja, fragmenteissa liikkuminen sivupalkin avulla.	2
Elokuva arvostelut ja elokuvan tiedot IMDB rajapinnasta	Ohjelma hakee IMDB API:sta elokuvan saama arvostelun, ohjaajan nimen sekä elokuvan juonen. Pistemääräksi ehdotettu viittä, sillä vaati käytännössä huomattavasti enemmän työtä, kuin jotkin muut ominaisuudet, joista saa saman verran pisteitä.	5
Firebase tietokannan käyttö	Firebase-tietokannan käyttö teki sovelluksesta huomattavan paljon realistisemmän verrattuna kurssilla läpikäytyihin metodeihin.	5
Hyvä UI	Ohjelman käyttöliittymä on visuaalisesti kiva, noudattaa nykyaikaisia malleja ja menetelmiä sekä on helppo käyttää.	5
Responsiivinen	Ohjelma toimii erikokoisilla näytöillä sulavasti	2
Asetukset	Asetuksissa 'darkmode'-teema kytken ja asetukset talletetaan XML-tiedostoon seuraavaa käynnistyskertaa varten.	2
Kielet	Sovellus toimii suomeksi ja englanniksi	2
Summa		45



## Työmäärät

Tekijä	Tehtävät	Tunnit
Miko Mattila	Suosikit, Finnkino-haku, Asetukset	30
Otto Oikarinen	IMDB API, lokalisatio, luokkakaavio	28
Teemu Viitanen	Käyttäjän rekisteröityminen, sisäänkirjautuminen, tiedot, Firebase, kalenterivalinta, kellovalinta, palautteenlähetyksen toiminto	21
Summa		79

## Mitä opin harjoitustyöstä?

### Miko Mattila

Opin harjoitustyöstä laajalti lisää olio-ohjelmoinnista ja erityisesti Android Studion, Githubin ja Javan käyttämisestä. Erityisen tärkeänä oppimanani asiana pidän ensikosketusta suuremman ohjelmointiprojektin toteuttamisesta ja ylipäättään millaista on luoda suurempi sovellus kokonaisuus yksittäisen näkymän/toiminnallisuuden sijaan. Opin myös paljon itse olio-ohjelmointia, sillä opin toteuttamaan paljon uusia toiminnallisuuksia, kuten XML-tiedoston lukemista ja kirjoittamista, sekä fragmenttien ja aktiviteettien toiminnallisuuksista.

### Otto Oikarinen

Opin harjoitustyöstä erityisesti API:n kanssa työskentelystä sekä JSON-tietojen hakemisesta ja lukemisesta. Tämän lisäksi opin myös sovellusten lokalisoinnista, eli kuinka sovellusten kieli saadaan vaihdettua puhelimen asetuksien perusteella. Kuten Miko sanoikin, tästä tuli myös hyvä kuva isommasta projektista ja sen koordinoinnista, sekä GitHubin käyttämisestä. Näiden kaikkien lisäksi opin tietysti paljon myös aktiviteeteista, olio-ohjelmoinnista ja Android Studion ominaisuuksista.

### Teemu Viitanen

Firestore-tietokannan käyttäminen ja implementointi Android studioon. Regex-osaamisen syventyminen käyttäjän syötteen tarkistuksessa (sekä käyttö oikeassa ohjelmassa). Fragmenttien ja aktiviteettien käyttö ”oikeasti”.

## **Palaute harjoitustyöstä (vapaaehtoinen)**

### **Miko Mattila**

Harjoitustyö oli kokonaisuudessaan hyvin opettavainen ja laajuudeltaan mielestäni sopivan laaja. Työssä oli sopivasti valinnanvaraa siitä, mitä haluaa/täytyy tehdä ja samalla se antaa mahdollisuuden keskittyä omiin mielenkiinnonkohteisiin. Tietysti työssä eniten syvältä oli aina kun jokin ‘palikka’ ei halunnut millään toimia, mutta sitäpä ohjelmointi itsessään onkin. Omalta kohdaltani vaikeinta oli toteuttaa XML-tiedoston kirjoittaminen. Ensivuodelle toivoisin enemmän opetusta Githubin käytöstä ja mielestäni hyödyllinen palikka olisi myös Firestore-tietokanta, jota voisi halutessaan hyödyntää työssään.

### **Otto Oikarinen**

Sovelluksen lokalisatio oli suhteellisen helppo toteuttaa. Sen toteuttaminen olisi vain pitänyt tarkistaa aivan kehitystyön alussa, niin sen olisi saanut rakennettua yhtä aikaa muiden ominaisuuksien kanssa, eikä kaikkien tekstien etsiminen ja korvaaminen myöhemmin olisi ollut niin aikaa vievä prosessi. Itselleni hankaluuksia tuottivat aktiviteetit sekä JSON-tietojen hakeminen ja lukeminen, joka ei meinannut aluksi toimia millään. Android Studioon olisi kaivannut kattavampaa perehdytystä. Oli kuitenkin mukava nähdä, ettei Android-sovelluksen kehittäminen kuitenkaan aivan liian työlästä ole, vaan se onnistuu suhteellisen nopeallakin perehdytyksellä Javaan. Jakaisin itse ehkä kurssin kahteen osaan: 3op Java-kurssi, jonka jälkeen 4-6op Android-kehityskurssi jatkoksi, jonka pääsisältönä olisi Android Studio, sen erilaiset toiminnallisuudet ja sovelluskehitys Javalla ylipäänsä.

## Teemu Viitanen

Firestore tietokannan käyttö oli erittäin opettavaista ja mielenkiintoista. Sovelluksesta tuli enemmän oikean sovelluksen tapainen, kun käyttäjän autentikoinnin voi hoitaa niin, että se toimii kaikilla laitteilla, eikä vain vertailla käyttäjän syötteitä lokaaliin tiedostoon tai muuhun vastaavaan paikkaan tallennettuihin tietoihin. Firebasen käyttö osoittautui aika käyttäjystävälliseksi ja dokumentaatiota ja tietoa oli helposti saatavilla. Kokonaisuutena opin harjoitustyöstä erittäin paljon, ja kurssin viikkotehtävissä kohdattu turhautuminen/epävarmuus hälveni javan/android studion suhteen. Fragmentit aiheuttivat hetkittäin harmaita hiuksia, sillä välillä tuntui, että ihan sama mitä tekee, lopputuloksena on aina NullPointerException. Netistä löytyvät vinkit + androidin dokumentaatiot käyttävät myös eri menetelmiä kuin kurssilla läpi käytävät.