

# Elokuvan arvostelusovellus - Movisio

Tekijät Riku Mattila, Asta Lehtovaara ja Anni Kesäniemi

## Kuvaus ohjelmasta

Harjoitustyön tarkoituksena oli luoda android pohjainen sovellus. Loimme elokuvien arvostelusovelluksen, Movision. Käyttäjä pystyy luomaan sovellukseen oman käyttäjätunnuksen ja sovellukseen pystyy kirjautumaan useampi käyttäjä. Kirjautumis ominaisuus on tehty Google Firebasen avulla, johon käyttäjän tiedot tallentuvat. Google Firebase mahdollistaa nopean ja turvallisen kirjautumisen. Sovelluksen etusivulla näkyy lista elokuvista, jotka on implementoitu Finnkinon XML:stä. Etusivulla käyttäjä pystyy hakemaan elokuvaa nimen, genren tai näyttelijän nimen perusteella. Kun käyttäjä klikkaa valitsemaansa elokuvaa, aukeaa arvostelusivu, jossa käyttäjä pystyy arvostelemaan elokuvan numeerisesti tähdillä sekä sanallisesti. Kun käyttäjä on arvostellut elokuvan, elokuvan arvostelut tallentuvat käyttäjän tietoihin ja käyttäjä voi katsella arvostelujaan myöhemmin.

## Tekijät

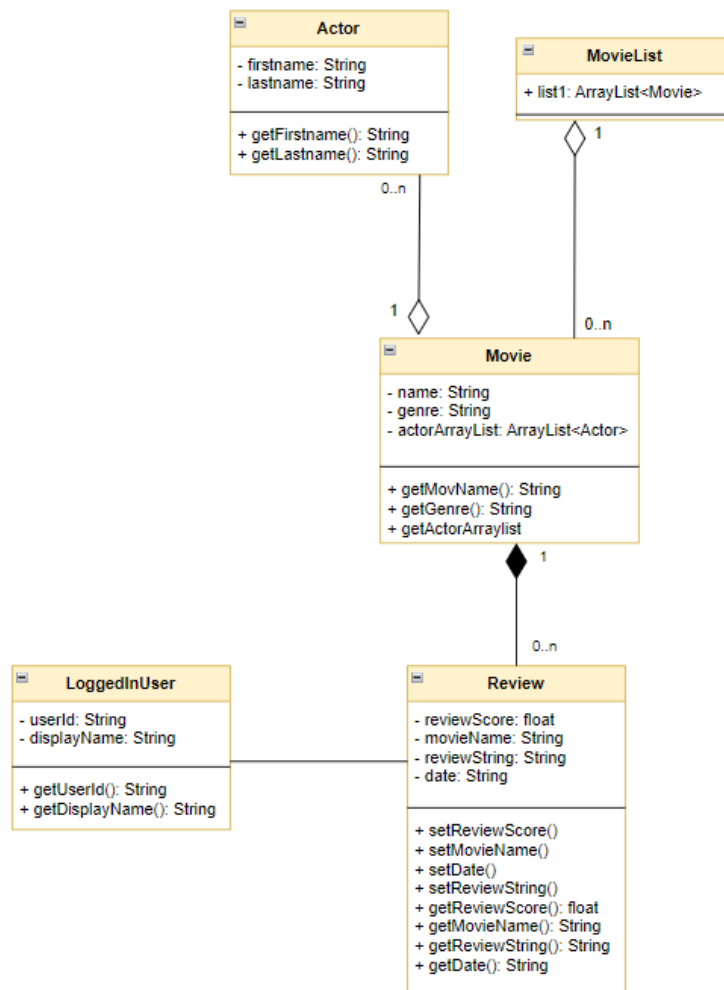
Riku teki käyttäjän kirjautumisen, tietojen tallennuksen sekä elokuvan arviointi toiminnot. Anni ja Asta tekivät osansa yhdessä, koska Astan Android Studio ei pyörittänyt ohjelmaa. Anni ja Asta implementoivat Finnkinon API:n Movielistiin sekä suunnittelivat UI:n. Ryhmän keskinäinen kommunikointi oli hyvää ja tapasimme viikoittain. Riku hyödynsi omaa osaamistaan haastavimmissa ominaisuuksissa ja oli muutenkin ryhmän most valuable player.

## Ohjelman toteutus

Ohjelma toimii Android 11 tai uudemmalla versiolla.

Työkalu	Käyttökohde
GitHub	Versionhallinta
WhatsApp	Yhteydenpito
Google Docs	Dokumentaation kirjoittaminen
Draw.io	UML-kaavioiden tekeminen
Android Studio	Ohjelmistokehitys, Testaus
Google Firebase	Ohjelmistokehitys

## Luokkakaavio



## Toteutetut ominaisuudet

Ominaisuus	Toteutettu / Perustelut	Pisteet
Olio-ohjelmoitu	Toteutettu	Pakollinen
Vähintään viisi erilaista luokkaa & oliota (käyttöliittymäluokkia ei lasketa)	Toteutettu	Pakollinen
Vähintään yhden API:n käyttö, esim. Ilmastodieetti: <a href="https://ilmastodieetti.ymparisto.fi/il">https://ilmastodieetti.ymparisto.fi/il</a>	Toteutettu	Pakollinen

<a href="http://mastodieetti/swagger/ui/index">mastodieetti/swagger/ui/index</a> , Järvirajapinta: <a href="http://rajapinnat.ymparisto.fi/api/jarvirajapinta/1.0/">http://rajapinnat.ymparisto.fi/api/jarvirajapinta/1.0/</a> , THL: <a href="https://thl.fi/fi/tilastot-ja-data/aineistot-ja-palvelut/avoin-data">https://thl.fi/fi/tilastot-ja-data/aineistot-ja-palvelut/avoin-data</a> tai Finnkino: <a href="https://www.finnkino.fi/xml/">https://www.finnkino.fi/xml/</a>		
Sovellus tallentaa käyttäjän toiminnan (käyttäjän syöttämät arvot / tulokset) johonkin talleen (JSON, XML jne. formaatit ovat käteviä)	Toteutettu	Pakollinen
Käyttäjä voi katsoa tallentamiaaan tietoja myöhemmin (esim. listata katsomansa elokuva paremmuusjärjestyksessä)	Toteutettu	Pakollinen
Ohjelma on rakennettu hyvin suunnitelluista UI-komponenteista	Toteutettu kaikki komponentit ovat toimivia ja käytettäviä	1 – 5 pistettä
Kirjautuminen applikaatioon	Toteutettu FirebaseAuth avulla	3 pistettä
Sovelluksella voi olla useampi käyttäjä (ja niiden luominen), tietojen tallennus järkevästi jonnekin	Toteutettu FirebaseAuth avulla	3 pistettä
Kirjautumisen salasana noudattaa hyvän salasanan sääntöjä (sisältää vähintään yhden numeron, erikoismerkin, ison ja pienen kirjaimen, on vähintään 12 merkkiä pitkä)	Toteutettu FirebaseAuth avulla ja käyttäjän luominen vaatii salasanan joka noudattaa hyvän salasanan sääntöjä.	2 pistettä
Salasanan tallennus käyttää jonkinlaista hash-menetelmää ja suolausta (esim SHA-512 + salt)	Toteutettu Firebase Authin avulla admin voi lähettää uuden salasanan sähköpostiin	2 pistettä
Elokuva-appi antaa mahdollisuuden syöttää elokuvan näyttelijät / ohjaajan ja katsomiaan elokuvia voi hakea näyttelijän / ohjaajan pohjalta	Toteutettu, Actor-luokassa. Pystyy hakemaan näyttelijän etunimen, sukunimen tai koko nimen perusteella elokuvia.	4 pistettä
Scoped storagen käyttäminen tiedon tallennuksessa (ei vaadi käyttäjän myöntämiä oikeuksia laitteen massamuistiin, vaan toimii omassa "hiekkalaatikossaan")	Toteutettu firebaseen avulla ja tietoja ei tallenneta välimuistiin vaan reaaliaikaisesti REALTIME STORAGE:en	2 pistettä
Jokin oma hieno ominaisuus tai toiminto (tai useampi)		Max 5 pistettä per ominaisuus

<b>Huomaa, että yllä listat ominaisuudet olivat vain esimerkkejä. Kannattaa ehdottomasti luoda omia (animoinnit, haitarielementit, päivittyvät hakukentät, singletonit...) ja pisteyttää niitä!</b>		
Summa		34

## Työmäärät

Tekijä	Tehtävät	Tunnit
Riku Mattila	Suunnittelu, Kirjautuminen, Tietojen tallentaminen, Testaus	60
Asta Lehtovaara	Suunnittelu, Finnkinon XML:n implementointi, UML-kaavio, Dokumentoinnin viimeistely, Testaus	50
Anni Kesäniemi	Suunnittelu, Finnkinon XML:n implementointi, UML-kaavio, Dokumentoinnin viimeistely, Testaus	50
Summa		160

## Mitä opin harjoitustyöstä?

**Riku Mattila:** Ryhmätyön erityinen anti oli ohjelmistoprojektin tekeminen ryhmässä. Jokaisen tyyli ohjelmoida erosi hieman ja muilta myös oppi paljon ryhmätyön aikana. Erityisen hienoa oli saada tukea vaikeina hetkinä muilta ryhmän jäseniltä. Firebasen käyttäminen ja opettelu oli hienoa ja toi uutta ajateltavaa ohjelmointiin. Opin myös paljon laajemmin eri ohjelman osien yhteensovittamisesta ja esimerkiksi tiedonvälityksestä eri aktiviteettien/fragmenttejen välillä.

**Asta Lehtovaara:** Ryhmätyössä kommunikoinnin tärkeys korostui, jotta kaikilla oli sama käsitys, minkälaisen ohjelman teemme. Opin käyttämään GitHubia sekä ohjelmoimaan ryhmässä. Opin myös käsittelemään sekä lukemaan laajempaa koodia ja ohjelmaa. Riku ohjelmoi ohjelmaamme kirjautumisominaisuuden Firebasea käyttäen, opin itse myös samalla Firebasen toiminnoista ja miten sitä pystyy hyödyntämään koodissa. Harjoitustyö myös kokosi viikkotehtävissä olleet tehtävät. Esimerkiksi pääsin hyödyntämään XML:n lukemista harjoitustyössä ja liittämään sen laajempaan kokonaisuuteen. Harjoitustyössä konkretisoitui myös eri aktiviteettien ja rajapintojen välinen yhteys.

**Anni Kesäniemi:** Opin sietämään epävarmuutta projektityössä. Lisäksi opin etsimään tietoa laajemmin sekä hyödyntämään GitHubia yhdessä ohjelmoidessa. Harjoitustyö oli vaikea ja aikaa vievä, mutta lopussa kova työ palkittiin, kun koodi alkoi näyttämään oikealta ohjelmalta. Kurssin aikana tuntui, että en osannut soveltaa oppimaani niin hyvin tai ymmärtänyt olioiden hyödyntämistä täysin. Harjoitustyössä pääsi kunnolla paneutumaan olioihin ja niiden ominaisuuksiin. Koen, että harjoitustyössä opin ymmärtämään rajapintojen välistä kommunikointia ja tiedon välitystä. Opin myös, että ohjelmoinnin projektityössä kommunikointi on tärkeää, jotta välttyttäisiin epäselvyyksiltä ja koodin ristiriitaisuudelta. Suunnitteluvaiheeseen olisi voinut vielä enemmän käyttää aikaa, koska suunnittelemani lopullisesta UI:sta muodostui vasta työn loppuvaiheessa.

## Palaute harjoitustyöstä (vapaaehtoinen)

- Mitkä ominaisuudet / toiminnot olivat helppoja / vaikeita toteuttaa?  
Vaikeinta oli toteuttaa mm. ActorList eli saada Finnkinon XML:stä childnode lisättyä listaan.
- Oliko jokin asia aivan syvältä?  
Harjoitustyö oli melko vaikea tehdä vain viikkotehtävien perusteella ja materiaalia harjoitustyön tekemiseen oli melko vähän.
- Oliko jokin asia todella hyvää tässä työssä?  
Oli kiva, että pakollisia ominaisuuksia ei ollut niin paljon ja sai ryhmässä päättää mitä ominaisuuksia sovellukseen halusi implementoida.
- Mitä toivoisit ensi vuoden harjoitustyöhön?  
Ensi vuoden harjoitustyöhön toivottaisiin enemmän opetusmateriaalia, joka tukisi harjoitustyön tekoa. Nyt melkein kaikki harjoitustyöhön vaaditut ominaisuudet eivät löytyneet materiaaleista vaan kaikki piti etsiä netistä.