MovieTracker

Jaakko Pyrhönen

Kuvaus ohjelmasta

MovieTracker on Finnkinon avointa dataa hyödyntävä Android-sovellus. Sovelluksella voi:

- Katsella tuoreimpien Finnkinon teattereissa esitettävien elokuvatapahtumien tietoja ja filtteröidä niitä
 - päivämäärän ja ajan perusteella
 - teatterin perusteella
- hakea elokuvia ja filtteröidä tuloksia
 - näyttelijän ja/tai ohjaajan etu- ja/tai sukunimen perusteella
 - genrejen perusteella
 - elokuvan nimen perusteella
- Luoda useamman käyttäjätilin. Käyttäjätilille voi kirjautua ja sen voi poistaa
- Arvostella elokuvia omalla käyttäjätilillään, tallentaa näitä arvosteluja ja katsella niitä myöhemmin
- Tallentaa käyttäjäkohtaiseen kalenteriin elokuvatapahtumia ja katsella niitä

Miksi nämä on toteutettu

Yllä mainitut ominaisuudet sopivat elokuva-applikaation tyyliin ja tuovat mahdollisesti lisäpisteitä.

Miten ohjelma on suunniteltu ja toteutettu

Ohjelma on suunniteltu noudattavan MVVM-arkkitehtuurimallia. Perusidea on se, että ohjelma hakee dokumentin Finnkinon API:sta, parsii siitä erinäisiin dataolioihin alustusolioiden avulla halutut tiedot ja tallentaa säilytettävät oliokokoelmat tiedostoon, jolloin niitä ei tarvitse aina hakea netistä. Tiedot esitetään käyttäjälle fragmenteissa ja fragmentit hyödyntävät dataa saadakseen ViewModel:ja. ViewModel:it poimivat tarvittavat tiedot alustetuista dataolioista, mahdollisesti muokkaavat hieman datan esitysmuotoa ja palauttavat tulokset metodien kautta dataa pyytäneille fragmenteille. ViewModel:it eivät tiedä fragmenttien olemassaolosta mitään ja ne on luotu spesifeihin tarkoituksiin kuten UserViewModel tai TheatreViewModel.

Fragmentit esittävät ViewModel:ilta saatavia tietoja joko RecyclerViewissä tai dialogeissa. RecyclerView:t tarjoavat selattava näkymän ja näkymän erinäisiä elementtejä klikkaamalla aukeaa dialogi, joka esittää elementistä tarkempaa tietoa tai tarjoaa mahdollisesti lisävaihtoehtoja.

Tekijä

Koko projektin kehittäjä: Jaakko Pyrhönen

Toteutus

Alusta

Ohjelma toimii Androidin versiolla 5.0 ja uudemmilla. Kohdealustana on Androidin versio 12.

Kirjastot

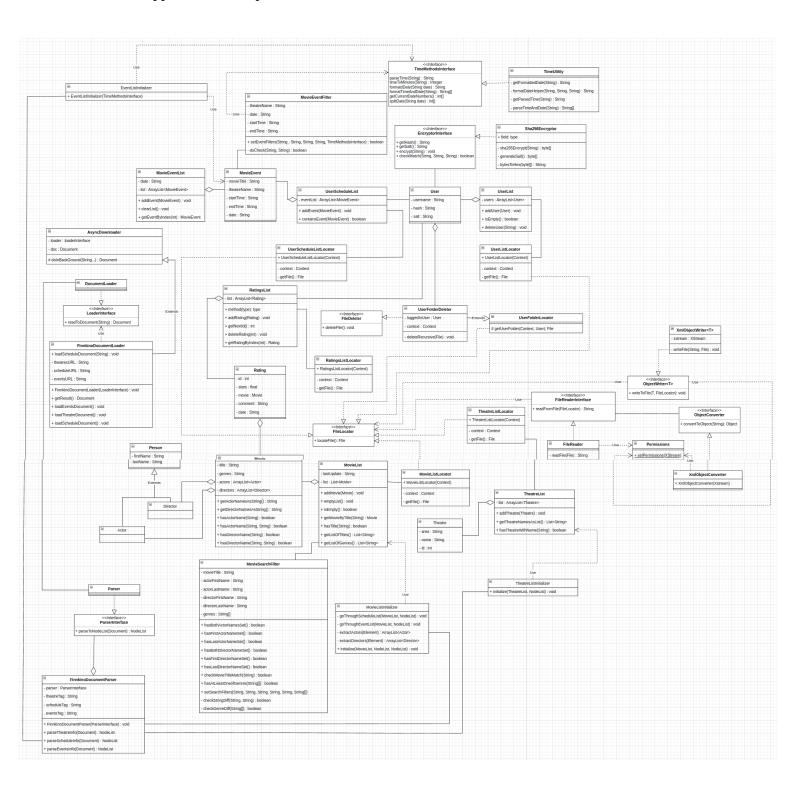
- xstream: https://x-stream.github.io/. Xstream on kirjasto, jonka avulla voidaan luoda objekteista xml-muotoinen merkkijono ja tämä merkkijono voidaan kääntää takaisin objektiksi. Tätä toiminnallisuutta on hyödynnetty projektissa tiedostoon kirjoittamisen ja sieltä lukemisen vhtevdessä.
- Android SDK:n tarjoamat oletuskirjastot

Työkalut

- Android Studio Chipmunk | 2021.2.1 patch 1
 gradle versio 7.3.3, Android plugin versio 7.2.1
 Java, kohdeyhteensopivuus java 8

Luokkakaavio

Ohessa on applikaation datapuolen luokkakaavio.



Toteutetut ominaisuudet

Ominaisuus	Perustelut	Pisteet
Olio-ohjelmoitu	Yritin parhaani mukaan noudattaa hyviä olio- ohjelmoinnin periaatteita	Pakollinen
Vähintään viisi eri luokkaa & oliotoa.	Erilaisia ei-käyttöliittymä- luokkia on 36. Monella niistä on tosin samankaltainen toiminnallisuus.	Pakollinen
Sovellus tallentaa käyttäjän toiminnan talteen	Tätä varten sovelluksessa käytetään xstream-kirjastoa, jonka avulla objektista luotu xml-merkkijono kirjoitetaan tiedostoon	Pakollinen
Käyttäjä voi katsoa tallennettuja tietoja myöhemmin	Käyttäjä voi esimerkiksi katsella omia arvostelujaan.	Pakollinen
Ohjelma on rakennettu hyvin suunnitelluista UI- komponenteista	Tästä en ehkä täyttä viittä pistettä itselleni antaisi, mutta sovelluksessa on yritetty toteuttaa MVVM-mallia jakamalla komponentit modelviewmodel- ja view-osiin. Käsittäisin tämän oikein toteutettuna hyvin suunnitelluksi sovellukseksi.	2
Kirjautuminen applikaatioon	Jotta käyttäjä voi antaa arvosteluja, hänen tulee luoda käyttäjäprofiili sovellukseen.	3
Sovelluksella voi olla useampi käyttäjä, tietojen tallentaminen järkevästi	Sovelluksessa voi luoda useita käyttäjiä. Kunkin käyttäjän omat tiedot tallennetaan heille omistettuihin hakemistoihin.	3
Kirjautumisen salasana noudattaa hyvän salasanan sääntöjä. (sisältää vähintään yhden numeron, erikoismerkin, ison ja pienen kirjaimen, on vähintään 12 merkkiä pitkä)	Tämä on toteutettu siten, että käyttäjän antama salasana tarkistetaan regular expressonien avulla ennen kuin se hyväksytään. Mikäli salasana ei kelpaa, käyttäjälle näytetään ohjedialogi.	2
Salasanan tallennus käyttää	Rekisteröitymisen yhteydessä luotuun User-olioon tallennetaan	2
Ohjelma näyttää kalenterikomponentin avulla käyttäjän tallentamia tapahtumia	Kun käyttäjä hakee elokuvatapahtumia, hän voi painaa recyclerview:ssä esitettyä tapahtumaa. Tästä aukeaa dialogi-fragmentti, josta käyttäjä voi lisätä tapahtuman	

	kalenteriinsa, joka on toteutettu CalendarView-komponentilla. Kun käyttäjä klikkaa auki tapahtumapäivän siitä näytetään dialogissa tekstimuodossa tietoja.	
Elokuva-appi antaa mahdollisuuden syöttää elokuvan näyttelijät / ohjaajan ja katsomiaan elokuvi voi hakea näyttelijän / ohjaajan pohjalta	Applikaation "Search Movies"- osiossa käyttäjä voi hakea elokuvia mm. kuvauksessa amainittujen asioiden perusteella.	
Asynkronisten HTTP-kutsujen käyttö dataa haettaessa	Tämä on vähän siinä ja siinä voiko tästä antaa pisteitä sillä FinnkinoDocumentLoader-olio kyllä hakee dokumentit asynkronisesti, mutta metodi <i>getResult()</i> odottaa joka tapauksessa haun tulosta, mikä hieman päihittää tarkoituksen.	0 tai 2
Fragmenttien hyödyntäminen aktiviteettien sijasta käyttöliittymiä rakennettaessa	Ainoa aktiviteetti, joka sovelluksesta löytyy on MainActivity. Kaikki muut näkymät ovat fragmentteja tai dialogifragmentteja	2
Scoped storagen käyttäminen tiedon tallennuksessa	Sovellus ei tarvitse massamuistiin oikeuksia. Kaikki kirjoitetaan sovellukselle omistettuun yksityiseen tilaan.	2
Summa		37-39

Työmäärät

Arviolta projektin toteuttamiseen kului kokonaisuudessaan noin 50h.

Mitä opin?

- Ymmärrän paremmin Android-järjestelmiä
- Osaan paremmin ohjelmoida Android-sovelluksia
- Kehityin Java-ohjelmoinnissa
- Kehityin huomattavasti olio-ohjelmoinnissa jo verrattuna esimerkiksi kurssin kotitehtäviin.
- Kehityin arkkitehtuurisuunnittelussa ja sen mukana kokonaisuuden "paloittelussa"
- Osaan paremmin piirtää luokkakaavioita
- Opiskelin useita tunnettuja suunnittelumalleja paremman koodin toivossa, vaikka niitä en loppujen lopuksi sen kummemmin käyttänytkään, koska ohjelma on kuitenkin melko yksinkertainen eikä malleja kannatanut väkisin pakottaa mukaan.
- Omaksuin paremmin hyvän ohjelmointityylin ajattelumalleja