

Kirjasovellus

Itsearviointi

Aku Laurila

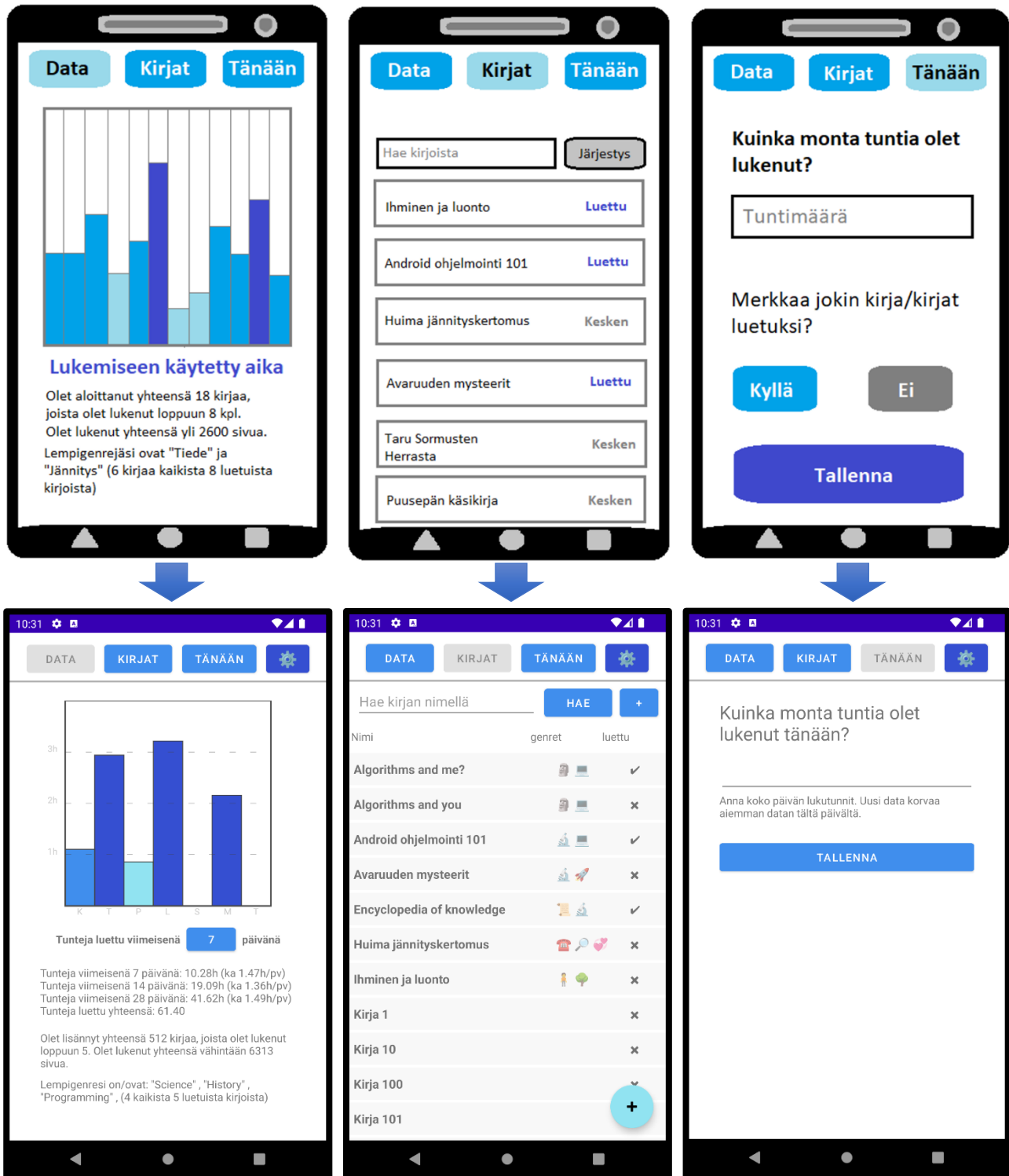
akulau@student.uef.fi

Sisällys

1. Arviointi suunnitelman toteutumisesta.....	3
1.1 Vertailua luonnoksiin.....	3
1.2 Onnistumiset	4
1.3 Ongelmat	5
1.3 Haasteet.....	5
2. Selvitys ohjelman tekemiseen käytetystä ajasta	6
3. Mitä opin?.....	7
3.1 Mitä opin käytännössä	7
3.2 Mitä opin teoriassa.....	7

1. Arviointi suunnitelman toteutumisesta

1.1 Vertailua luonnoksiin



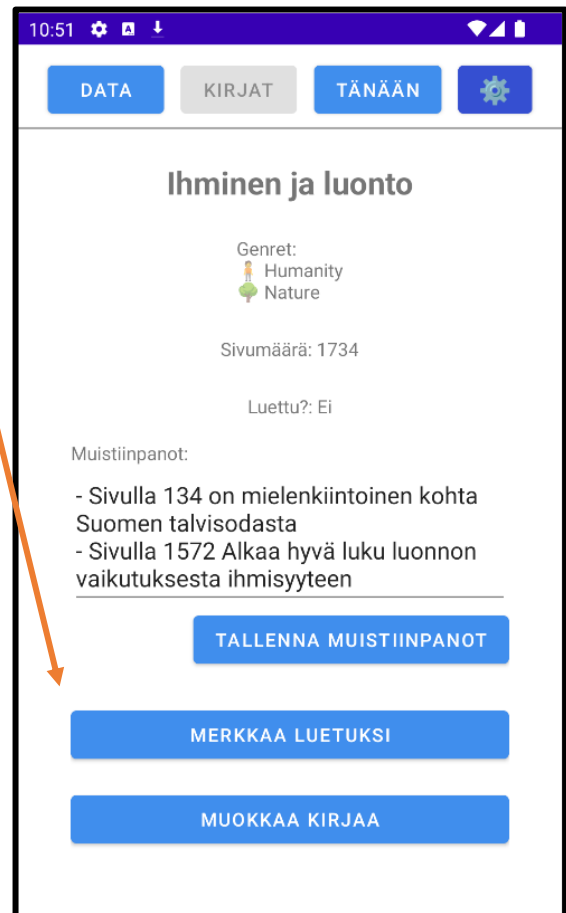
Nyt kun laitoin luonnokset vierekkäin lopullisen sovelluksen kanssa, niin yllätyin, kuinka lähelle alkuperäistä suunnitelmaa sovelluksen käyttöliittymä päätyi. Muutoksia tuli toki tehtyä, ja pitikin tehdä, jotta sovelluksesta tulisi käyttäjäystävällisempi.

Suurin ero suunnittelun- ja lopullisen sovelluksen välillä on tänään -näymästä kadonnut ”Merkaa kirja/kirjat luetuiksi”-kohta. Koin tämän kuuluvan enemmän kirjan omalle sivulle, jossa voi alkuperäisestä suunnitelmasta poiketen lisätä kirjalle myös muistiinpanoja.

Kirjat -välilehdelle ilmestyi oikeaan alalaitaan nappi, jota en ollut luonnoksiin huomannut suunnitella sinne. Napin kautta pääsee lisäämään kirjan, lisäämään genren ja muokkaamaan genreä.

Ylävalikkoon ilmestyi uusi nappi asetuksille, jonka kautta käyttäjä voi vaihtaa sovelluksen kieltä, generoida testidataa ja pyyhkiä sovelluksen tietokannan tyhjäksi.

Data -välilehti pysyi käytännössä samana. Olen todella tyytyväinen siihen, miten pylväskaavio onnistui.



1.2 Onnistumiset

Onnistuin mielestäni tekemään sovelluksen ominaisuudet suunnittelun mukaisiksi. Minulla on usein tapana alkaa toteuttamaan ominaisuuksia, jotka eivät kuulu alkuperäisen suunnitelman rajoihin. Tällä kertaa onnistuin kuitenkin pysymään projektin rajoissa. Vaikka suunnittellessa ei tullut otettua kaikkea huomioon, kuten genrejen hallintaa ja muokkausta, sain ne silti toteutettua ilman, että ne monimutkaistivat projektia liikaa.

Onnistuin mielestäni todella hyvin myös pylväsdiagrammin toteutuksessa. Olin valmistautunut taistelemaan sen parissa useamman illan, mutta taisinkin saada sen valmiiksi muutamassa tunnissa. Pidän erityisesti siitä, miten se skaalautuu datan mukaan.

Skaalautuva ja näytönkoko mukautuva käyttöliittymä onnistui mielestäni myös mainiosti. Käyttöliittymä ei hajoa näyttöä käännettäessä ja tarvittaessa, voi näkymiä aina vierittää pystysuunnassa.

Onnistuin myös pitämään projektin suurin piirtein suunnitellussa aikataulussa. Suunnitteludokumentissa arvioin neljä viikkoa ja GitHubin committien perusteella aikaa olisi mennyt noin neljä ja puoli viikkoa.

1.3 Ongelmat

Ongelmia ilmeni monessa asiassa. Suurimmat projektin etenemistä hidastavat ongelmat olivat kuitenkin harvassa ja niitä ilmeni muun muassa JavaDoc:in generoinnissa, emojiien tunnistuksessa, sekä lokalisaation ja näytön kääntämisen kanssa. Näistä mainintoja on jo kehittämisoppaassa, mutta loppujen lopuksi ongelmiin löytyi edes jokin ratkaisu, jonka turvin projekti jatkoi rullaamista eteenpäin.

1.3 Haasteet

En osaa sanoa, oliko haasteita enemmän, vai vähemmän kuin odotin. Projekti oli omalla tavallaan haastava jo sen vuoksi, etten ole ennen tehnyt ns. ”kokonaista” Android-sovellusta. Java oli kuitenkin jo ennestään tuttua ja kunhan ymmärsin sen, miten esimerkiksi resursseja kannattaa käyttää, oli ohjelmointi yllättävän helppoa.

Haastetta tuli kyllä siitä, kun päätin projektin loppupuolella, että kirjat- välilehden kirjalista pitäisi hakea asynkronisesti tietokannasta. Sain sen loppujen lopuksi toimimaan, mutta kyllä siihen pari iltaa taisi mennä.

2. Selvitys ohjelman tekemiseen käytetystä ajasta

Projektin suunniteltu aikataulu oli neljä viikkoa. GitHubin commit-historiasta tarkistettuani, huomasin aloittaneeni projektin noin neljä ja puoli viikkoa sitten. Sanoisin, että pysyin aikataulussa yllättävän hyvin, mutta yllätyin silti työmäärästä. Alkuperäinen aika-arvioni oli tehty yläkanttiin, enkä oikeasti uskonut projektissa kestävän näin kauan.

Alkuperäisessä suunnitelmassani olin jakanut sovelluksen osia eri viikoille, mutta todellisuudessa sovelluksesta tuli toteutettua aina se osa, mikä sillä hetkellä tuntui luonnistuvan parhaiten. Aloitin esimerkiksi käyttöliittymän rakentamisen jo ensimmäisellä viikolla, vaikka suunnitelmassa se oli merkattu vasta myöhemmäksi.

Projektin suunnitteluun kului ehkä tunti. Minulla oli ennestään jo idea valmiina, joten suunnittelun aika kului lähinnä näkymien luonnosteluun ja toiminnallisuuden tarkentamiseen. Varsinaiseen ohjelmointiin arvioisin vierähtäneen n. 60 – 70 tuntia. Tiedonhakuun ja opiskeluun varmaan hieman vähemmän, sillä kurssin aiemmista tehtävistä tuli opittua jo suurin osa perusteista. Arvioisin tiedonhakuun kuluneen n. 30 tuntia. Kokonaisuutena projektiin kului varmaan n. 91 – 101 tuntia.

Tehtävä	arvioitu käytetty aika (tuntia)
Suunnittelu	1
Ohjelmointi	60-70
Opiskelu/Tiedonhaku	30
	yhteensä n. 91 - 101

3. Mitä opin?

3.1 Mitä opin käytännössä

Opin, kuinka Android-sovelluksen lokalisaatiota voidaan vaihtaa ja tallentaa. Opin myös enemmän `Canvas`-luokasta ja siihen piirtämisestä ohjelmallisesti. Tietokantakutsut säikeistettyinä olivat mielenkiintoisia ja uskon oppineeni siitä ainakin `Observer`-luokan käyttöä.

Opin hyödyntämään GitHubia paljon monipuolisemmin projektissa. Käytin GitHubin Issue- ja Milestone -ominaisuuksia ([Linkki](#)) laajasti ja pidin siitä, miten niiden avulla pystyi seuraaman projektin kulkua. Committeja projektiin on kirjoittamisen aikana tullut tehtyä 114 kappaletta ja Issue:ita on luotu 48. Milestoneja oli 5 kappaletta, joista jokainen tuli toteutettua loppuun asti. Issue-label:it helpottivat tulevien ominaisuuksien ehdotusta, bugien raportointia ja kategorisointia.

3.2 Mitä opin teoriassa

Opin navigoimaan Android studion projektirakenteessa paremmin ja nopeammin. Enää ei tarvitse muistella, missä esim. `colors.xml` sijaitsee. Opin myös Android-sovelluksen rakenteesta ja ”järjestyksestä” enemmän. Järjestyksellä tarkoitan sitä, missä järjestyksessä komponenttien ohjelmakoodia ajetaan. Esimerkiksi `Activity:n onCreate()` -metodia kutsutaan myös silloin, kun näyttöä käännetään. GitHubissa pidin yllä mielenkiintoista keskustelua, ennen kuin keksin tämän: <https://github.com/UEF-CS-Courses/harjoitustyo---course-project-akulau/issues/24>

Opin myös datan yhdistelyä, käsittelyä ja analysointia. Tietokannan taulujen yhdistely oli jo ennestään tuttua muilta kursseilta, mutta kertaus teki hyvää.

Käyttäjän syötteen tarkistaminen on tärkeää, sillä väärässä muodossa annettu syöte voi kaataa sovelluksen. Tämä tuli opittua jo aivan projektin alkuvaiheilla.