



LUT School of Engineering Science
CT60A2411 Olio-ohjelmointi

Jiri Musto

Harjoitustyö - Android sovellus

30.4.2020

Shan Shan Yang 0546392

Sisällysluettelo

1. Tehtävänannon kuvaus ja työn rajaukset.....	1
2. Ohjelman kuvaus	1
3. Ohjelman suunnittelu ja toteutus.....	2
3.1 Toteutetut ominaisuudet ja toiminnot.....	2
4. Ohjelman luokkakaavio	4
5. Oma reflektio	4

1. Tehtävänannon kuvaus ja työn rajaukset

Harjoitustyön tavoitteena oli toteuttaa Olio-ohjelmoinnin -kurssille Android applikaatio, jossa osoitetaan osaamista oliopohjaisten ohjelmointimenetelmien käytöstä tyypillisten ohjelmointiongelmien ratkaisemiseksi ja pystyy soveltamaan Java-kielen tavanomaisia piirteitä ohjelmoinnissa. Toteutin tämän yhden henkilön työnä ja aiheekseni valitsin monitoimisalin varausjärjestelmän.

Työn tavoitteena oli laatia helposti käyttöön otettava varausjärjestelmä, jonka kautta käyttäjä pystyy varaamaan vapaana olevan monitoimisalin itselleen haluamalleen päivälle ja tietyksi ajaksi. Lisäksi käyttäjä pystyy liittymään muiden luomille varauksille. Selatessaan muiden käyttäjien varauksia, näkymästä tulee löytyä tarvittavat tiedot salista, kestosta ja salin käyttötarkoitus.

Tässä työssä keskitytään asioihin, jotka ovat tyypillisen varausjärjestelmän kannalta olennaisia. Ohjelma rajattiin niin, että peruselementit löytyvät kuten varauksen luominen, siihen liittyminen, muuttaminen ja poistaminen. Helppokäyttöisyys on tärkeä ominaisuus, jota työssä lähdin tavoittelemaan. Lisäominaisuudet kuten varauskohtaisen kuvan ja useamman monitoimihallin mahdollisuuden lisääminen jätettiin pois, koska työ haluttiin pitää yksinkertaisena. Tietokannassa on mahdollisuus useammalle yliopistolle mutta toiminnallisuutta ei kuitenkaan toteutettu ajanpuutteen vuoksi.

2. Ohjelman kuvaus

Sovellus on suunniteltu varausjärjestelmäksi yliopiston liikuntatiloja varten. Kohderyhmänä on LUT Groupin opiskelijat ja henkilökunta. Varausjärjestelmän avulla opiskelijat voivat varata tarvitsemansa tilan käyttöönsä. Kirjaututtuaan sisään, käyttäjä pystyy ”create event”-painikkeesta luomaan tapahtuman, johon hän pystyy määrittämään päivän, ajan, keston, salin ja rajaamaan osallistujamäärän.

Kun käyttäjä valitsee päivämäärän niin näytölle ilmestyy kalenteri, josta pystyy klikkaamaan haluamansa päivän. Käyttäjä pääsee ”enroll”-painikkeesta näkymään, jossa hän voi liittyä tapahtumaan osallistujaksi. Päivämäärä kenttään voi kirjoittaa haluamansa päivämäärän, jolloin spinner-lista päivittää sen päivän saatavilla olevat tapahtumat. Käyttäjä voi myös mennä suoraan spinneriin valitsemaan haluamansa tapahtuman. Kaikissa toiminnoissa on ”save”-painike, joka tallentaa käyttäjän tekemät muutokset. Käyttäjä pääsee myös muokkaamaan tapahtumiaan ”edit event”-näkyvän kautta, joka avautuu päävalikosta kyseistä painiketta painamalla. ”View own

events”- painikkeella käyttäjä pääsee tarkastelemaan omia tapahtumiaan, että mihin on liittynyt tai ilmoittautunut.

Vain admin käyttäjä voi luoda uusia käyttäjiä. Videodemon takia tietokantaan on valmiiksi lisätty tapahtumia ja käyttäjiä, jotta toiminnallisuuden esittäminen on helpompaa.

Android toastit ovat pieniä yksinkertaisia infoja meneillään olevasta operaatiosta pienen pop upin muodossa. Esimerkiksi kun sali on varattu, niin käyttäjälle ilmoitetaan *Time is taken!* eikä tilaa voi näin ollen varata, koska se on jo varattu.

Käyttöliittymä on selkeä ja helppolukuinen. Mukavat ja tarkkaan valitut värit ovat silmälle miellyttäviä eivätkä värit riitele keskenään.

3. Ohjelman suunnittelu ja toteutus

Ohjelman suunnittelussa pohditaan yksityiskohtaisesti asioiden sommittelua ja asettelua kuvaruudulle. Sovellukselle hain esteettistä kokonaisuutta unohtamatta toimivuutta. Hyvään suunnitteluun kuuluu mielestäni myös se, että kaikki toiminnot sovelluksessa ovat yhdenmukaiset. Yhdenmukaisuuden pitäisi näkyä painikkeissa ja sijainnissa.

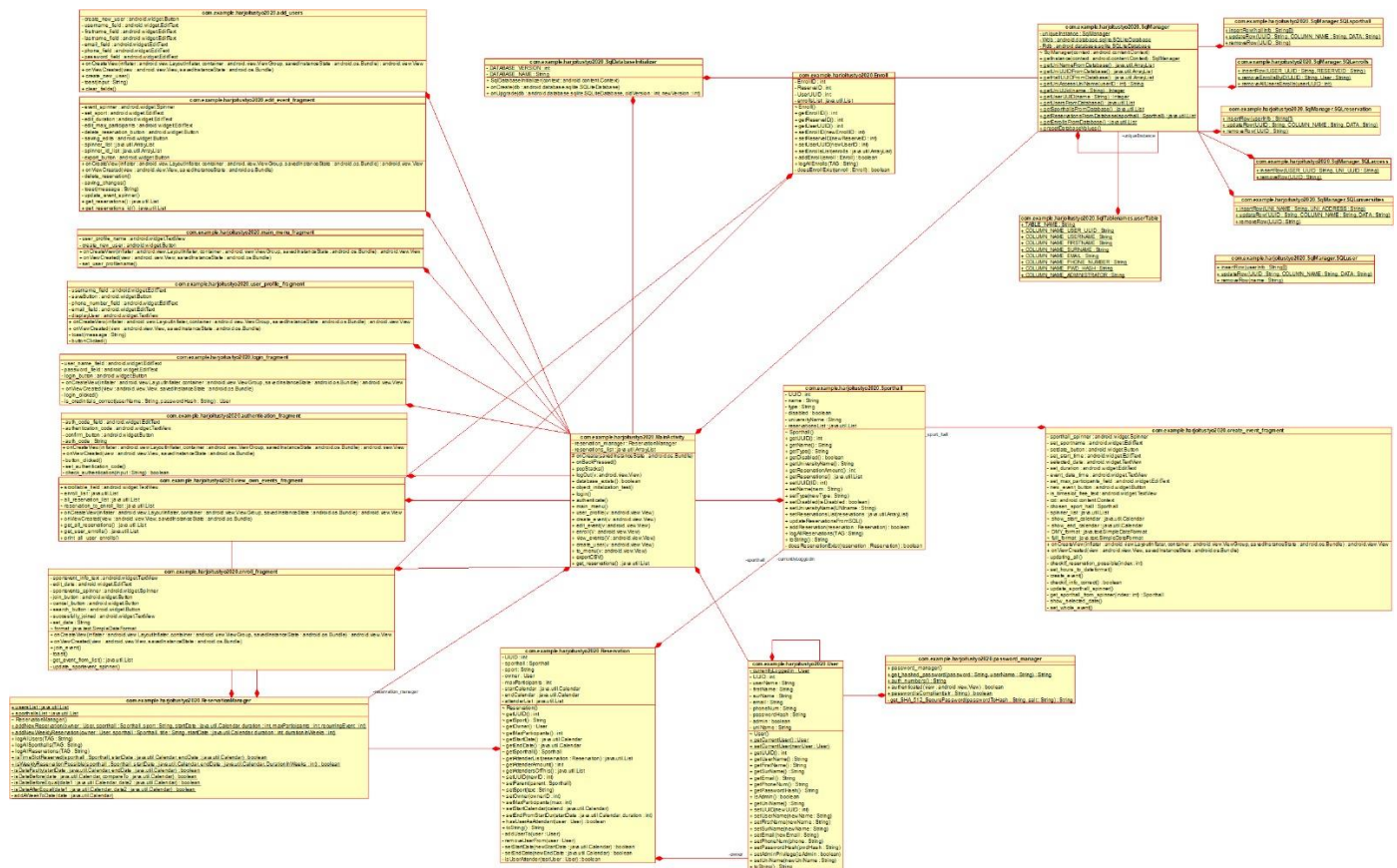
Tiedon tallentamiseen käytettiin tietokantaa, että sieltä löytyy varaukset, käyttäjät ja salit. Näkymien välillä liikkumiseen käytettiin fragmentteja, jolloin vältettiin tiedonsiirtoa aktiviteettien välillä. Tämän koin hankalammaksi kuin yhdessä aktiviteetissa pysymisen ja näkymien vaihtamiseen.

3.1 Toteutetut ominaisuudet ja toiminnot

Pistemäärä	Toiminto
Pakollinen	Olio-ohjelmoitu
Pakollinen	Vähintään viisi erilaista luokkaa & oliota (käyttöliittymäluokkia ei lasketa)
Pakollinen	Tärkeiden tietojen kirjoitus tiedostoon (XML, JSON tai CSV)
Pakollinen	Aihe spesifit perustoiminnot (löytyvät jokaisen aiheen alta listattuna)
Pakollinen	Voi muokata käyttäjän tietoja (nimi, yhteystiedot, jne.)

Pakollinen	Voi tehdä varauksia (voitte päättää järkevän rajauksen mihin asti tulevaisuuteen varauksia voi tehdä) 1. Sisältää tiedon salista 2. Sisältää tiedon kellonajasta ja päivästä 3. Sisältää tiedon kuka on varannut 4. Sisältää kuvauksen mitä varten sali on varattu 5. Toisen varauksen päälle ei voi varata
Pakollinen	Voi muokata varauksia ennen varausten alkua
Pakollinen	Voi tarkastella varauksia päivittäin / viikoittain eri salien kohdalla (suunnitelkaa itse). Vapaat ja varatut ajat näkyvät käyttäjälle
Pakollinen	Voi tarkastella omia varauksia
2 pistettä	Ohjelma on rakennettu hyvin suunnitelluista UI-komponenteista
5 pistettä	Tietokanta
3 pistettä	Admin-käyttäjä
3 pistettä	Useampi käyttäjä (ja niiden luominen), tietojen tallennus järkevästi jonnekin
3 pistettä	Kirjautuminen applikaatioon
3 pistettä	Kirjautumisen salasana noudattaa hyvän salasanan sääntöjä (sisältää vähintään yhden numeron, erikoismerkin, ison ja pienen kirjaimen, on vähintään 12 merkkiä pitkä). JA Kirjautumisen jälkeen, ruudulle tulee myös satunnainen 6-numeroinen luku, joka käyttäjän pitää syöttää (simuloi avainlukukoodia).
3 pistettä	Salasanatallennus käyttää jonkinlaista hash-menetelmää ja suolausta (esim SHA-512 + salt)
1 piste	Voi lisätä uusia lajeja, jos niitä ei valmiiksi löydy
2 pistettä	Voi ilmoittautua / osallistua varattuihin tapahtumiin (esim. sali varattu koripalloa varten -> max 20 ihmistä voi ilmoittautua osallistuvansa)

4. Ohjelman luokkakaavio



5. Oma reflektio

Kokonaisuudessaan harjoitustyöhön meni aikaa 4 viikkoa. Joskus meni 6 h päivässä ja toisinaan koko päivä uppoutuen työhön. Kokonaisuudessaan aika tuli käytettyä noin 300-350 tuntia.

Harjoitustyön avulla tuli hyvin kerrattua kurssisisältö. Harjoitustyö myös kiteytti kaikki tärkeimmät toiminnot Javassa. Lisäksi pitkäjänteisyyttä tuli harjoiteltua, sillä harjoitustyöhön meni minulta paljon aikaa. Vaikeutta lisäsi se, että tietoa piti aika lailla itse osata hakea verkosta ja välillä ei tiennyt mitä edes tarvitsee hakea. Harjoitustyön aikana korostui mm. stackoverflow ja muiden sivujen merkitys.

Olio ylipäättään oli vaikeaa, koska tämä ei ollut minulle ennestään tuttu ja eroaa proseduuriseen ohjelmointiin kuten python. Asiaa joutui selvittämään itse todella paljon. Olioiden luominen oli aluksi hankalaa, koska se tuntui sekavalta. Helppoa oli esim. asettaa muuttujien nimet ja alustaa ne. Kun rutiinia alkoi syntyymään, niin asiat etenivät sulavammin. Konsolin tulostaminen oli myös helppoa sekä UI suunnittelu ei ollut hankalaa, sillä se oli jo ennestään tuttu.