

Tietokantojen perusteet

**Harjoitustyö – Android sovellus**

Lappeenrannan teknillinen yliopisto  
Tietotekniikan koulutusohjelma

Tietokantojen perusteet  
Kevät 2019

Nico Aalto  
Väinö Marttinen  
Joonas Maljanen  
[nico.aalto@student.lut.fi](mailto:nico.aalto@student.lut.fi)  
[vaino.marttinen@student.lut.fi](mailto:vaino.marttinen@student.lut.fi)  
[joonas.maljanen@student.lut.fi](mailto:joonas.maljanen@student.lut.fi)

## SISÄLLYSLUETTELO

SISÄLLYSLUETTELO .....	1
1 TEHTÄVÄNANNON KUVAUS JA TYÖN RAJAUKSET .....	2
2 TIETOKANNAN KÄSITEMALLI JA EHEYSSÄÄNNÖT .....	3
2.1 Käsittemalli .....	3
2.2 Relaatiomalli .....	4
3 OHJELMAN KUVAUS .....	5
4 OHJELMALLINEN TOTEUTUS JA TOIMINNALLISUUDET .....	7
5 OHJELMAN TÄYDELLINEN LUOKKAKAAVIO .....	8
6 YHTEENVETO .....	9

## 1 TEHTÄVÄNANNON KUVAUS JA TYÖN RAJAUKSET

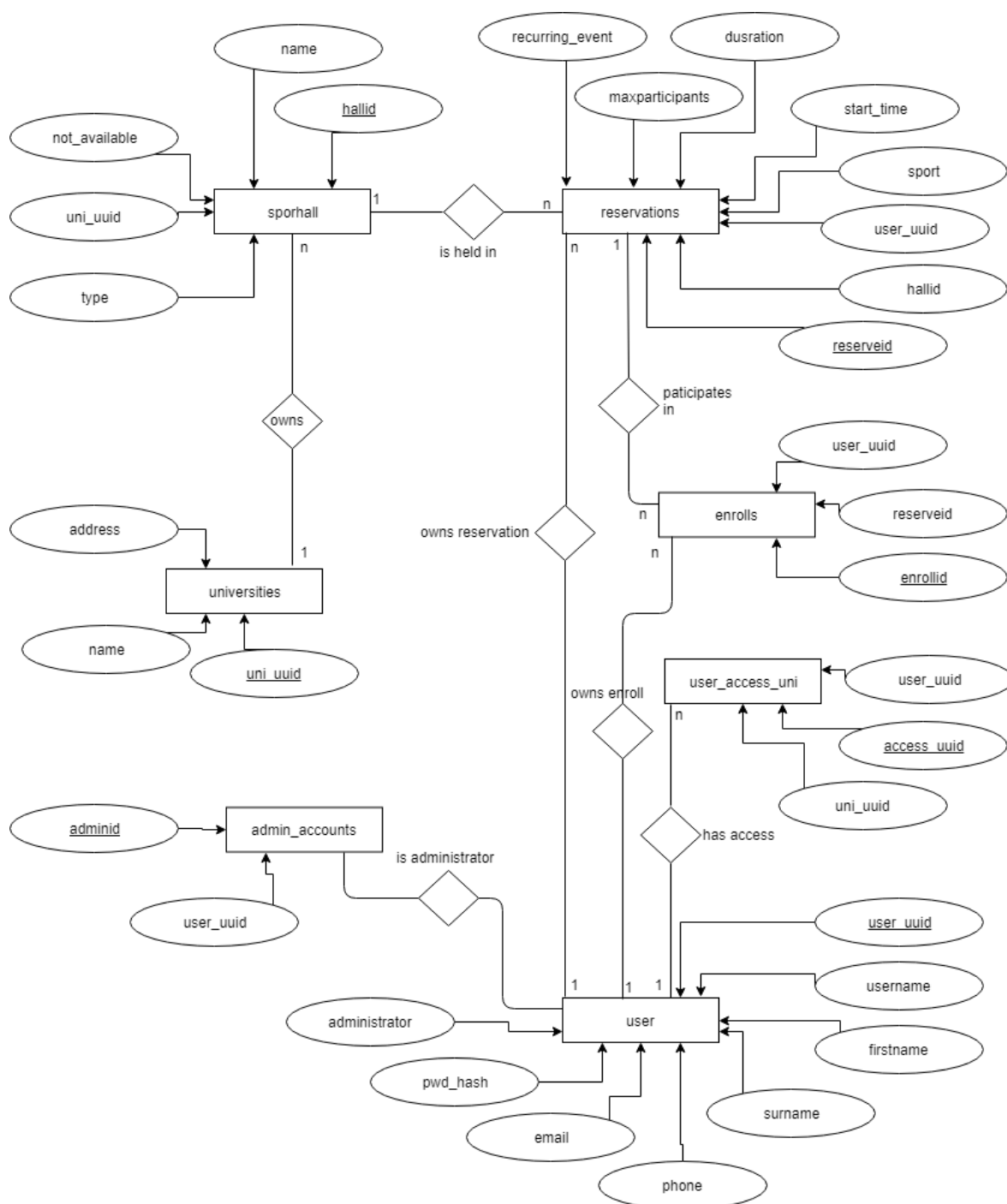
Tehtävänantona oli tehdä Android sovellus, joka hyödyntää kaiken tiedon tallentamisessa ja varastoinnissa tietokantaa ja samalla käyttää olio-ohjelmoinnin perusteita. Valitsimme aiheeksemme monitoimisalin varausjärjestelmän. Sovellus oli alun perin suunniteltu yhden henkilöntyöksi, mutta laajeni nopeasti, kun ryhmän koko kasvoi ensin kahteen ja heti perään kolmeen henkilöön. Tämä aiheutti tavoitteiden arvioimista ja asettamista uudelleen vastaamaan ryhmän uutta kokoa. Varausjärjestelmän kautta tulisi pystyä varaamaan sali itselleen halumaan tarkoitukseen tietyksi ajaksi ja mahdollista liittyä muiden luomille varauksille. Varauksia pitää pystyä muokkaamaan ja kaikki ilmoittautumiset pitää nähdä keralla. Muiden varauksia selatessa näkymästä tulee löytyä tiedot salista, kestosta, varaajasta ja mitä lajia varten sali on varattu.

Sovellukseen kirjautumisen toteutimme ilman käyttäjän luontia ja se tapahtuu, kun ylläpitäjä on kirjautunut sisään ja hallitsee käyttäjiä ja pystyy luomaan uusia sitä kautta. Useita käyttäjiä pystyy olemaan samaan aikaan useampi ja niiden tiedot tallentuvat tietokantaan. Ylläpitäjä pystyy käyttäjien poistamisen ja luomisen lisäksi muokkaamaan saleja ja lisäämään ja poistamaan niitä. Ylläpitäjä näkee myös kaikki tiedot varauksista, käyttäjistä ja saleista. Korvasimme seuraavat ominaisuudet tietokantojen käyttämisellä, käyttäjien ”jäädyttäminen” ja kaikkien tietojen muokkaaminen ylläpitäjänä.

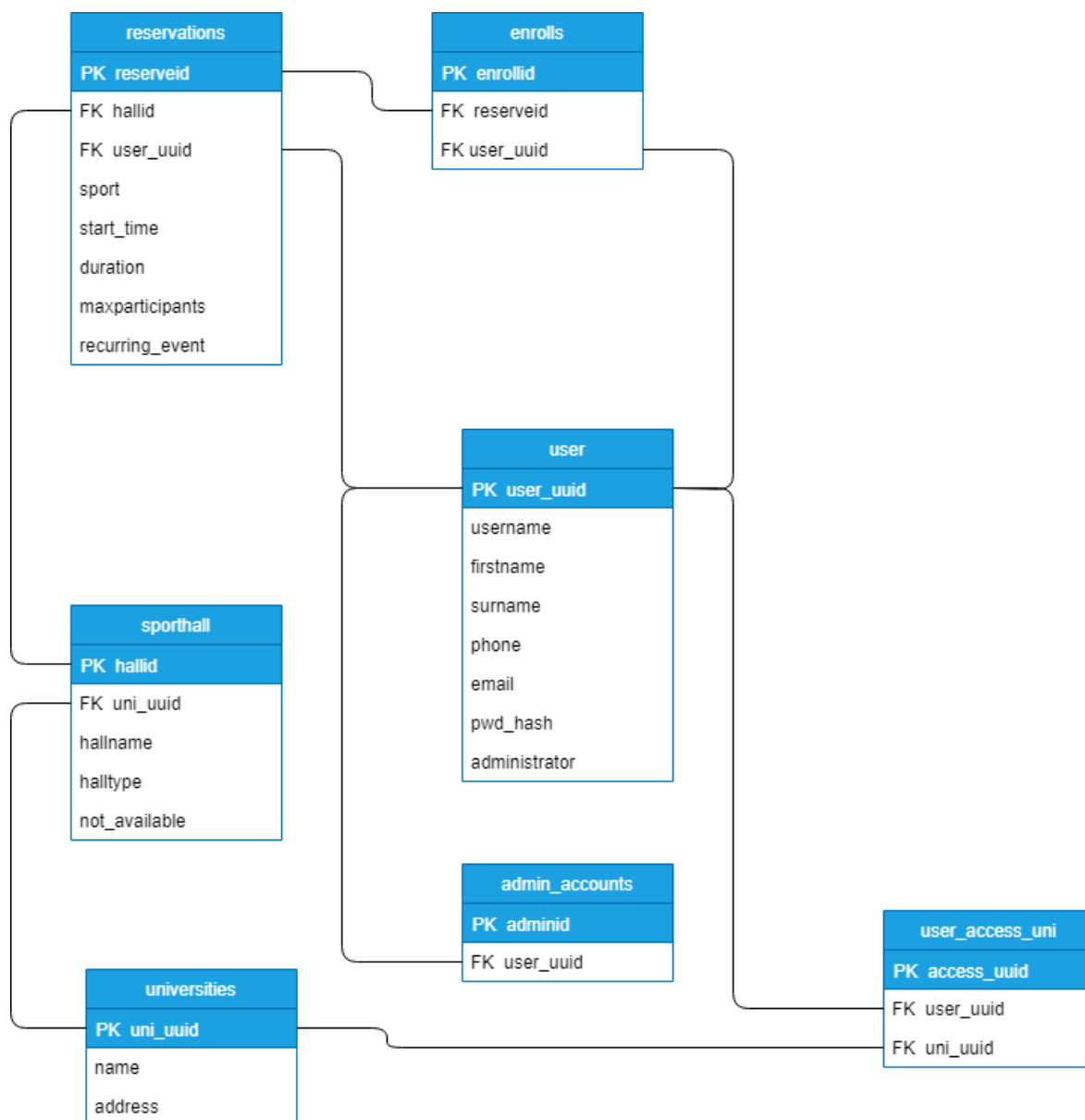
Tehtävänantoa lisäsi, kun ryhmän jäsenien määrä kasvoi lopulta kolmeen henkilöön. Vakiovuorojen varaaminen tuli ominaisuutena lisänä sekä varaustietojen ulosotto CSV-tiedostona. Salasanojen tulee noudattaa hyvän salasana sääntöjä ja ne tulee olla suolauksen kautta salattuna tietokannassa. Koko tietokannan tiedot tulee saada järjestelmästä ulos JSON-tiedostomuodossa. Järjestelmässä voi olla useampi yliopisto ja käyttäjällä tulee näkyä, minkä yliopistojen saleihin hänellä on oikeudet tehdä varaus.

## 2 TIETOKANNAN KÄSITEMALLI JA EHEYSSÄÄNNÖT

### 2.1 Käsitemalli



## 2.2 Relaatiomalli



### 3 OHJELMAN KUVAUS

Sovelluksemme käyttö alkaa kirjautumis ikkunasta ja olettaa että käyttäjälle on jo luotu käyttäjätili ja tili oikeudet ovat oikein. Kun käyttäjä on kirjautunut sovellukseen, hänelle tarjotaan tärkeimmät ominaisuudet heti ensimmäisenä näytön ylälaudassa. Nämä neljä painiketta sisältävät sovelluksen perustoiminnot ja ne ovat merkittynä käyttäjälle ”user tasks”. Sovelluksen ominaisuuksiin kuuluu salin varaaminen haluamaansa urheilulajin käyttöön, jos käyttäjä haluaa varauksen toistuvan viikoittain, voi käyttäjä valita kuinka monta viikkoa varausta toistetaan. Käyttäjä voi myös liittyä toisen henkilön varaamalle vuorolle mukaan, jos vuorolla on vielä tilaa. Käyttäjä voi myös muokata varaamiaan vuoroja ja sovellusta tarkistaa tietokannasta, että uusi varaus on mahdollinen. Käyttäjä näkee kaikki ne vuorot, joille hän on ilmoittautunut samalta näkymältä automaattisesti. Päänäkymän viimeinen ominaisuus on omien tietojen muokkaaminen ja samasta näkymästä hän voi kirjautua ulos.

Käyttäjällä ollessa hallinnolliset oikeudet, hän pystyy hallitsemaan muita käyttäjiä ja saleja. Käyttäjiä hallittaessa niitä voidaan luoda lisää tai poistaa valittu käyttäjätili. Valitun käyttäjätilin voi myös poistaa halutulta vuorolta. Saleja hallittaessa käyttäjä voi nähdä kaikki varaukset, poistaa käytöstä tai ottaa salin takaisin käyttöön, sekä poistaa salin kokonaan. Käyttäjä voi myös luoda uusia saleja

Korvasimme toiminnallisuudesta käyttäjän ”jäädätyksen” ja ylläpitäjän näkymästä kaiken tiedon muokkaamisen. Ohjelman toiminnallisuus pohjautuu pitkälti fragmenttien käyttämiseen ja niiden käytössä on muiden luokkien metodit käytössä ja tätä kaikkea hallitsee MainActivity.

### **Monitoimisalın vuoronvarausjärjestelmä (käyttäjän näkökulma)**

Työn aiheena on tietokanta, johon pystytään tallentamaan tietoa ja sieltä pystyy vastaavasti hakemaan tietoa ohjelman käytettäväksi. Tietokannan taulut tuli määritellä siten että ne näyttävät tärkeitä tietoja ja ne pystytään helposti yhdistämään. Kuitenkin montaa taulua ei tarvitsisi kutsua, kun halutaan tallentaa yhden tyyppistä tietoa, kuten uuden salin tiedot.

Tietokantaa käytetään taustalla ja sovelluksen käyttäjä ei syötä suoraan tietokantaan mitään, vaan hänelle annetaan selkeä käyttöliittymä, johon voi syöttää tietoa. Tämä tieto syötetään metodien kautta tietokantaan, jossa se säilytetään. Varsinaista käyttäjää tietokannallamme ei ole, koska sovelluksen käyttäjä ei käytä tietokantaa suoraan. Sovelluksen käyttäjä toimii välillisenä käyttäjänä tietokannalle.

Tietokannasta löytyvät käyttäjien tiedot mukaan lukien kirjautumista varten tarvittavat tiedot, kuten käyttäjänimi ja salasanan hajautusalgorimin lopputulos. Näin salasanaa ei tallenneta missään vaiheessa tietokantaan, jolloin sitä ei pysytä myöskään saamaan sieltä ulos. Käyttäjistä tallennetaan perustiedot nimi, sähköposti, puhelin numero ja lisäksi on käyttöoikeus taso. Tietokantaan tallennetaan salien tiedot ja niiden varaukset, näihin varauksiin liitetään osallistujat erillisen taulun avulla. Kaikki salit ja käyttäjät kuuluvat johonkin yliopistoon, joka toimii ylimpänä luokkana näille. Jos yliopistot poistetaan, kaikki käyttäjät poistuvat myös. Joten sovelluksesta on estetty viimeisen yliopiston poistaminen. Käyttäjiä, saleja ja yliopistoja voidaan luoda lisää tai poistaa. Nämä ominaisuudet ovat kuitenkin varattu vain käyttäjille, joilla on hallinta oikeus.

#### 4 OHJELMALLINEN TOTEUTUS JA TOIMINNALLISUUDET

Sovelluksen ohjelmallinen toteutus pohjautuu pitkälti MainActivity:n kautta fragmenttien hallitsemiseen. Kaikki näkymämme ovat omissa fragmenteissa ja niitä kutsutaan tarpeen mukaan alkunäkymän päälle. Jokaisella fragmentilla on oma luokkansa, jossa se on määritetty avaamaan tietty xml-tiedosto. Luokka sisältää myös tarvittavat metodit ja niiden kutsut. Fragmenteista kutsutaan muiden luokkien metodeja, jotta saadaan tietoa tallennettua tietokantaan.

Kaikki ohjelmassa käytettävä tieto löytyy tietokannasta ja tiedot päivitetään sinne. Tieto siirtyy fragmentista metodien kautta tietokantaa olioiden avulla ja sama toimii olioiden avulla toiseen suuntaan. Näin saamme yhdellä kutsulla kaikki tarvittavat tiedot tietokannasta.

Sovelluksessa on käytössä hallintaluokkia jotka hallitsevat olioita ja luokat pitävät sisällään metodeja joita kutsumalla tieto siirtyy olioissa. Näin yksikään fragmentti ei itse käytä tietokantaa vaan tieto siirretään eteenpäin.

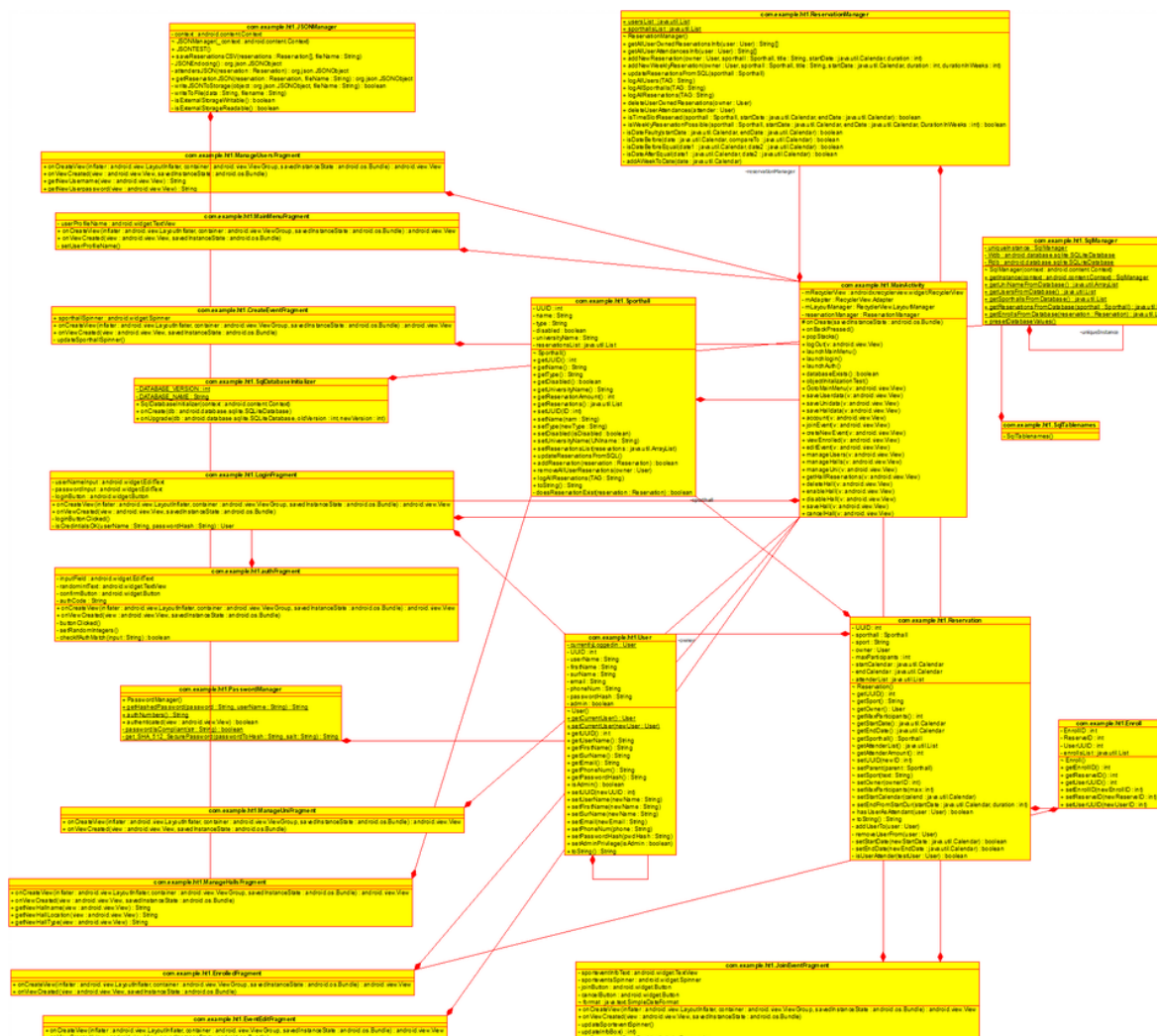
Sovelluksessa pystyy

- Varaamaan saleja
  - Tekemään niistä viikoittain uusiutuvia
- Liittymään muiden tekemiin varauksiin
- Muokkaamaan omia varauksia
- Näkemään kaikki vuoroille osallistumiset
- Saamaan varaus tiedot ulos CSV-tiedostona
- Muokkaamaan omia tietoja
  - Puhelin numero ja sähköposti
- Hallitsemaan muita käyttäjiä ylläpitäjänä
  - Lisää tai poista käyttäjä
  - Poista käyttäjien varauksia
- Hallitsemaan saleja ylläpitäjänä
  - Lisää tai poista saleja
  - Poista sali väliaikaisesti käytöstä
  - Katsomaan sen varaukset



- Asettamaan salille maksimi osallistujamäärän ja sen ollessa 0, rajaa ei ole
- Kirjautumaan sisään

## 5 OHJELMAN TÄYDELLINEN LUOKKAKAAVIO



## **6 YHTEENVETO**

Tietokannan käskyt olivat aika helppoja, kun niitä käsiteltiin suoraan, mutta siirryttäessä Java ohjelmointiin syntaksissa oli hieman opettelemista. Lopulta tietokannan syntaksi Javassakin oli suhteellisen nopeasti ymmärrettävää. Siinä kuitenkin huomasi miten taulut liittyvät toisiinsa, kun Javan kautta poistaa yliopiston niin tietokannasta poistuu kaikki siihen liittyvää salit ja käyttäjät.