

Tietokantojen perusteet ja Olio-ohjelmointi

Harjoitustyö – Dokumentaatio

Lappeenrannan teknillinen yliopisto

Tietotekniikan koulutusohjelma

Tietokantojen perusteet

Olio-ohjelmointi

Kevät 2019

Otso Weckström

Otso.weckstrom@student.lut.fi

1	MÄÄRITYS.....	3
2	MALLINNUS.....	4
3	TIETOKANTATOTEUTUS	7
4	OHJELMALLINEN TOTEUTUS.....	8
5	YHTEENVETO.....	10

1 Määrittely

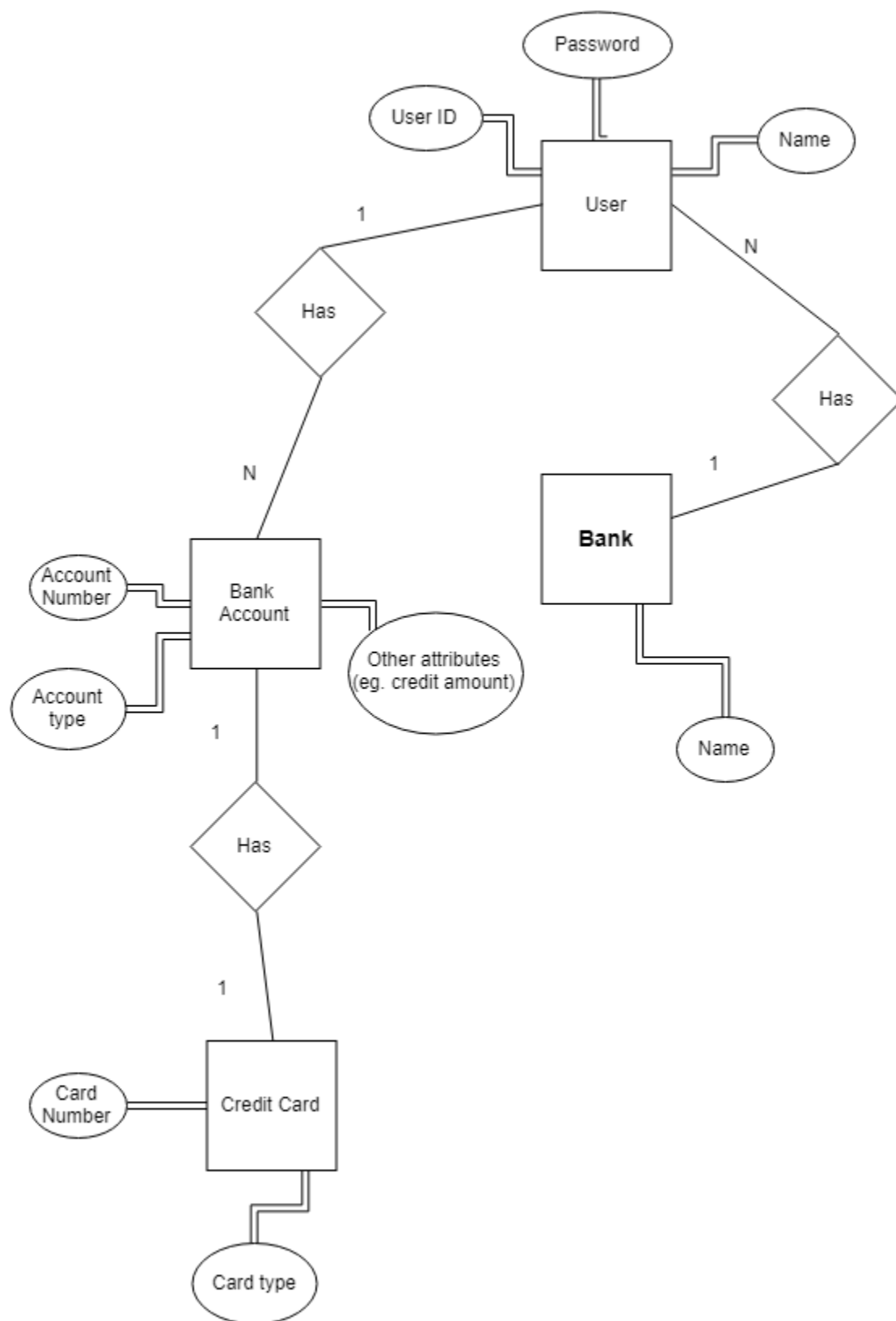
Harjoitustyöni projekti on pankkiapplikaatio. Applikaatiota varten toteutetaan yksinkertainen infrastruktuuri pankin tiedonhallintaan, eli tietokanta, johon voidaan tallettaa tärkeimmät tiedot toiminnallisuutta varten. Tietokannassa tulee voida pitää tietoja käyttäjän kirjautumistunnuksista, käyttäjän tileistä, tileillä olevista rahamääristä ja muista mahdollisista määritteistä kuten luottorajasta, sekä tilien pankki- ja luottokorteista. Käyttäjä voi luoda ja poistaa omia tilejä ja kortteja, vaihtaa kirjautumistunnuksia ja salasanaa, lisätä tileille rahaa ja tehdä korttimaksuja muille tileille. Lisäksi käyttäjä voi myös poistaa asiakastilinsä joka poistaa kaikki asiakkaan tiedot/tilit/kortit tietokannasta.

2 Mallinnus

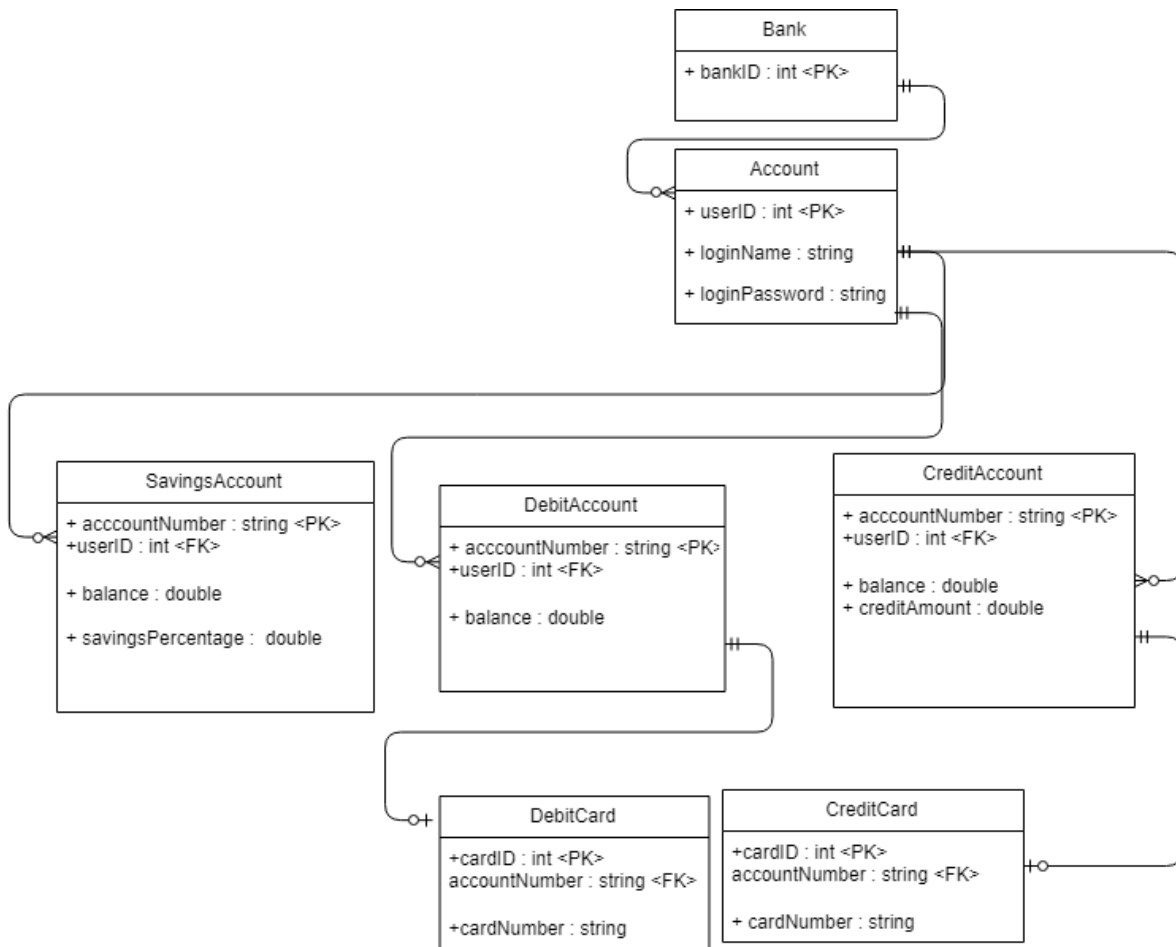
Aloitin mallintamisen luomalla karkean ER-mallin siitä mitä pankin tietokanta tulee pitämään sisällään. (kuva 1.)

Seuraavaksi muutin mallin relaatiokaavioksi. (kuva 2.)

Taulut noudattavat transformaatio sääntöjä, ja sisältävät tarpeelliset eheyssäännöt, jotka ovat näkyvissä liitteessä/tiedostossa jossa sql komennot on annettu.



Kuva 1: ER-kaavio



Kuva 2: Relaatiomalli

3 Tietokantatoteutus

Esimerkki:

Yleistä: Mallia toteuttaessa huomasin että Bank taulu on projektin kannalta täysin turha, olisin halunnut korvata taulun, jollakin hyödyllisellä kuten vaikka log taululla, johon kaikki transaktiot olisi voinut tallettaa. Harjoitustyön kiireellisen loppuunsaattamisen takia, johtuen deadlinen ”vaihtumisesta”, en ehtinyt enää korjaamaan tilannetta vaan jouduin jättämään Bank taulun työhön.

Toteutustapa: Toteutin tietokannan luomisen Android studiossa Javalla kirjoitetulla sovelluksella. Sovellus luo tietokannan ja testaamisvaiheessa lisäsi tietokantaan testidataa, jota tarkastelemalla tein hienosäätöjä eheysehtoihin, kuten salasanan CHECK määritteeseen.

4 Ohjelmallinen toteutus

Ohjelmassa on viisi olioissa tiedon tallettamista varten olevaa luokkaa: "BankCard", "CreditAccountClass", "DebitAccountClass", "SavingsAccountClass", "User".

Nämä luokat pitävät sisällään tiedon erilaisista tileistä ja pankkikorteista, user luokka puolestaan pitää sisällään tämän hetkisen käyttäjän tiedot.

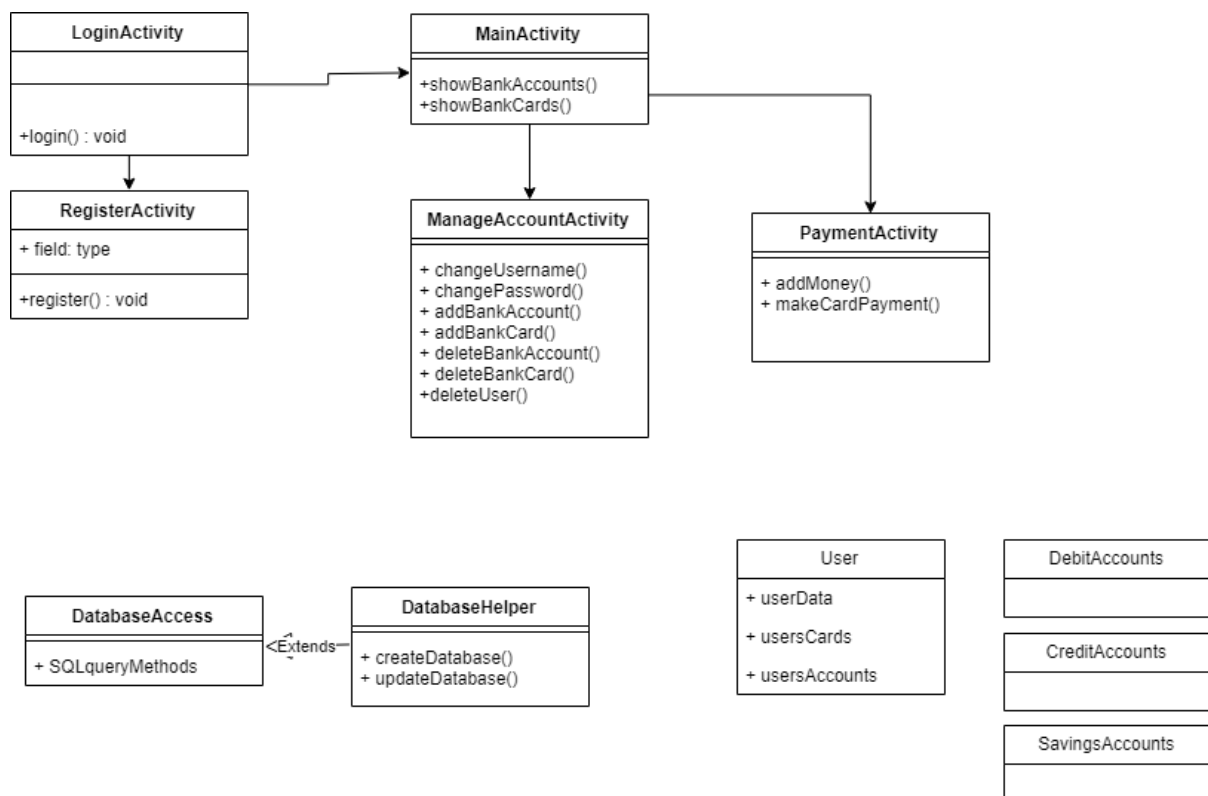
Kaksi luokkaa tietokannan kanssa toimimiseen: DatabaseHelper luo ja päivittää tietokannan, DatabaseAccess pitää sisällään kaikki SQL queryt tekevät metodit.

Lopuksi vielä useita Activityja joissa navigoimalla käyttäjä voi muokata ja selata tietoa tietokannasta.

Ohjelma sisältää tietokannan kautta toimivan kirjautumisen, rekisteröitymisen, nimen ja salasanan muutoksen, pankkitilien tarkastelun, pankkikorttien tarkastelun, tilien ja korttien luomisen sekä poistamisen, rahan lisäämisen, korttimaksujen tekemisen toiselle tilille ja käyttäjätilin poistamisen.

Pankkikorteilta voi maksaa vain sen verran mitä pankkitilillä on rahaa, ja luottokortilta katteen verran enemmän kuin tilillä on rahaa (esim. saldo 5000€ ja kate 100€ voi tililtä maksaa sillä hetkellä 5100€).

Kaikki tiedon poistaminen tapahtuu hallitusti niin että myös riippuvaisuudet poistuvat samalla, esim kun tili poistetaan myös sen tilin kortit poistuvat.



Kuvassa on näkyvissä kaikki pääluokat ja niiden toiminnallisuus, itse ohjelmassa Activityt johtavat vielä toisiin activityihin joissa metodit ovat mutta koin, että kuvasta olisi tullut turhan sekava jos

kaikki ikkunat olisi lisännyt samaan kuvaan. Ainoa ero on että osa kuvan metodeista on erillisessä ikkunassa.

5 Yhteenveto

Ohjelma on omasta mielestäni toiminnallisuudeltaan hyvä ja toimiva, toki ohjelmassa on monta asiaa, joita olisin mieluusti jatkanut ja toiminnallisuutta, joita olisin halunnut lisätä, kuten Admin näkymä ja miellyttävämpi UI. Vaikka työhön upposikin kokonaisuudessaan todella paljon aikaa, en ehtinyt käyttää raportointiin läheskään niin paljoa kuin olisin halunnut ja se varmasti näkyy laadussa.

En myöskään ehtinyt kirjoittamaan koodista, joitakin osia fiksummaksi koodiksi, joita tajusin loppusuoralla, koska aika ei vain mitenkään riitä tällä kertaa kaiken tekemiseen. Koen kehittyneeni harjoitustyön ansiosta koodaamisessa yleisesti, toki etenkin Javassa ja Android kehityksessä, mutta myös yleisesti ongelmien ratkomisen ja dynaamisen koodin saralla, ja koodin kirjoittamisessa kehittyminen olikin suurelta osin miksi kesäleirille ilmoittauduin.