

Tietokannat kirjallinen raportti

1. Tehtävänä on luoda tietokanta mobiili pankki-aplikaatioon, jota käytetään moninaisten tietojen käsittelyyn. Tietokantaan otetaan yhteyttä pankin asiakkaiden tietojen, asiakkaan tilin, tiliin liitettyjen tapahtumien ja korttien käsittelyyn. Tämän lisäksi sisäänkirjautuminen sovellukseen tehdään myös tietokannan kautta. Pankin työntekijälläkin on oman osansa tietokannasta josta saadaan hänen käsiteltävänä olevat tilit. Työssä tulee ainakin olla 5. kyselyä tietokantaan liittyen ja tämän lisäksi kaikki tietojen muutos tulee näkyä tietokannassa eli tietoja tulee päivittää UPDATE komennolla. Työssä tulee käyttää myös JOIN komentoa joilla tauluja liitetään yhteen, sekä CASCADE komentoa jolla tietoja voi poistaa helposti paljon turhaksi käyneitä tietoja kerralla.

2. Käsitellöllä pyrin kuvaamaan tarkasti tietokantaan tulevien osien yhteyksiä ja niiden eheyssääntöjä. Käsitellöön laitetaan seitsemän kohdetyyppeä, joita tarvitaan kuvaamaan tulevan tietokannan tarvitsemia osia. Kuvaan seuraavaksi ne:

Bank; joka kuvaa pankkia, jossa tilit ovat ja jossa asiakkaat käyttävät palveluita. Tietokannan käytettävyyden kannalta pankin attribuutteina on vain pankin nimi, osoite ja Bankin erittelevä BankID, jota käytetään myös pääavaimena joka ei saa olla nolla. Bank yhteystyyppinä on Admin(1:K) ja Customer(1:M). Vaikka tilit(accounts) ja kortit(bank cards) kohdetyypit kuuluvat myös pankkiin suoranaisesti, niin niiden hallinta pankin puolesta tehdään admin luokan kautta.

Customer: kuvaa asiakkaan tietoja pankin järjestelmässä ja sen attribuutteina on Name, Email, address, age(Jonka eheyssääntönä on ettei se voi olla <18), sekä customerID. Customer yhteystyyppinä on Bank(M:1) ja Account(1:N).

Admin: On toinen sana pankin työntekijöiden käyttäjille joilla he hallinovat asiakkaiden tietoja ja sen attribuutteina on Name ja WorkerID(joka on myös pääavain). Admin:in yhteystyyppinä on Account(K:R, eli adminilla voi olla monia tilejä ja tileillä voi olla monia adminia) ja Bank (K:1).

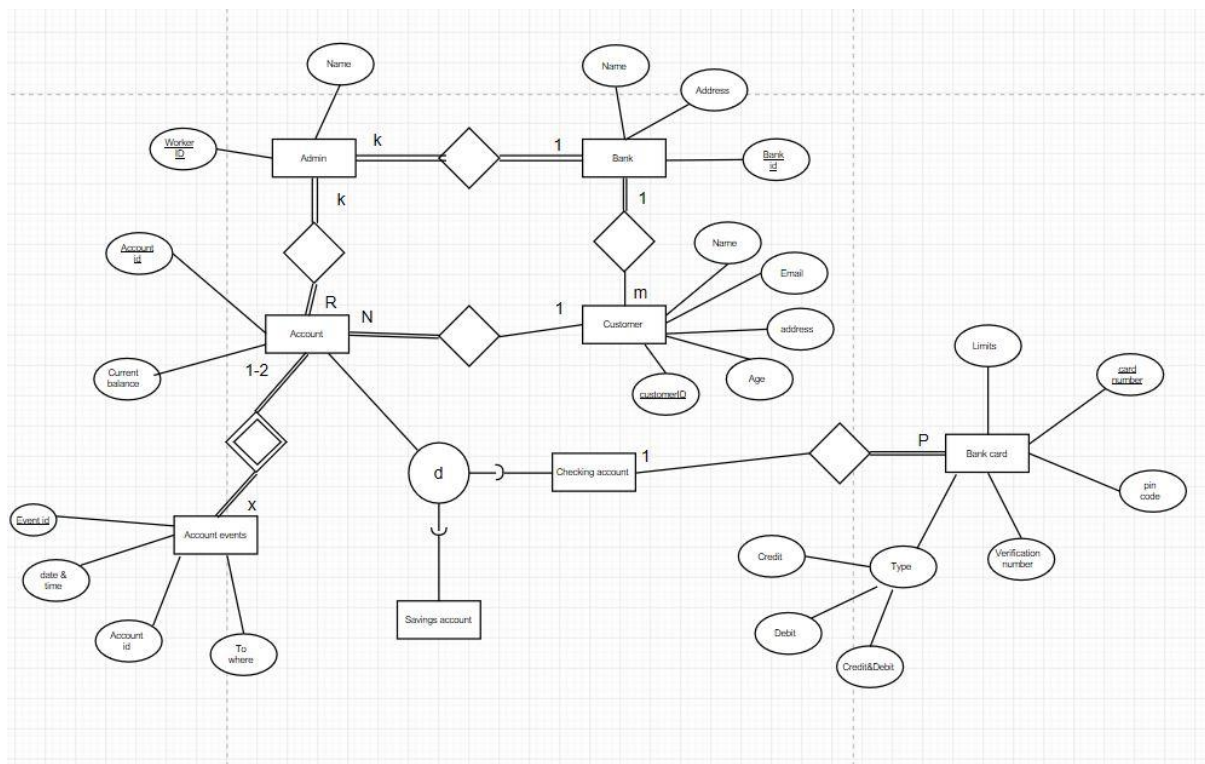
Account: Accountti eli tili pitää sisällään tilin tiedot. Sen attribuutteina on accountID ja currentBalance. Sen yhteystyyppinä on Admin(R:K),

Customer(N:1), Account event(1:x) ja Bank Card(1:P). Tämän lisäksi se kuuluu yhteen kahdesta alikohdetyypistä: Checkipng account(1:1) joka on käyttötili tai

savings account(1:1). Checkings accountin kautta tili liittyy myös yhteystyyppiin Bank Card.

Account event: on heikko kohdetyyppi, joka tunnistetaan kohdetyyppi Account:in avulla. Sen attribuutteja on Eventid, date&time, Accountid, To Where(joka kertoo saavan paikan joko toinen accountid/jokin muu paikka). Sen ainoa kohdetyyppi on sen määrittelevä Account.

Bank Card: Pitää sisällään tileihin liitettyjen pankkikorttien tiedot, ja sen attribuutteja on cardNumber, pinCode, verificationNumber, onlineLimit, cashLimit, checkingLimit. tämän yhteystyyppinä oli siis vain account(P:1) johon se liitetään.

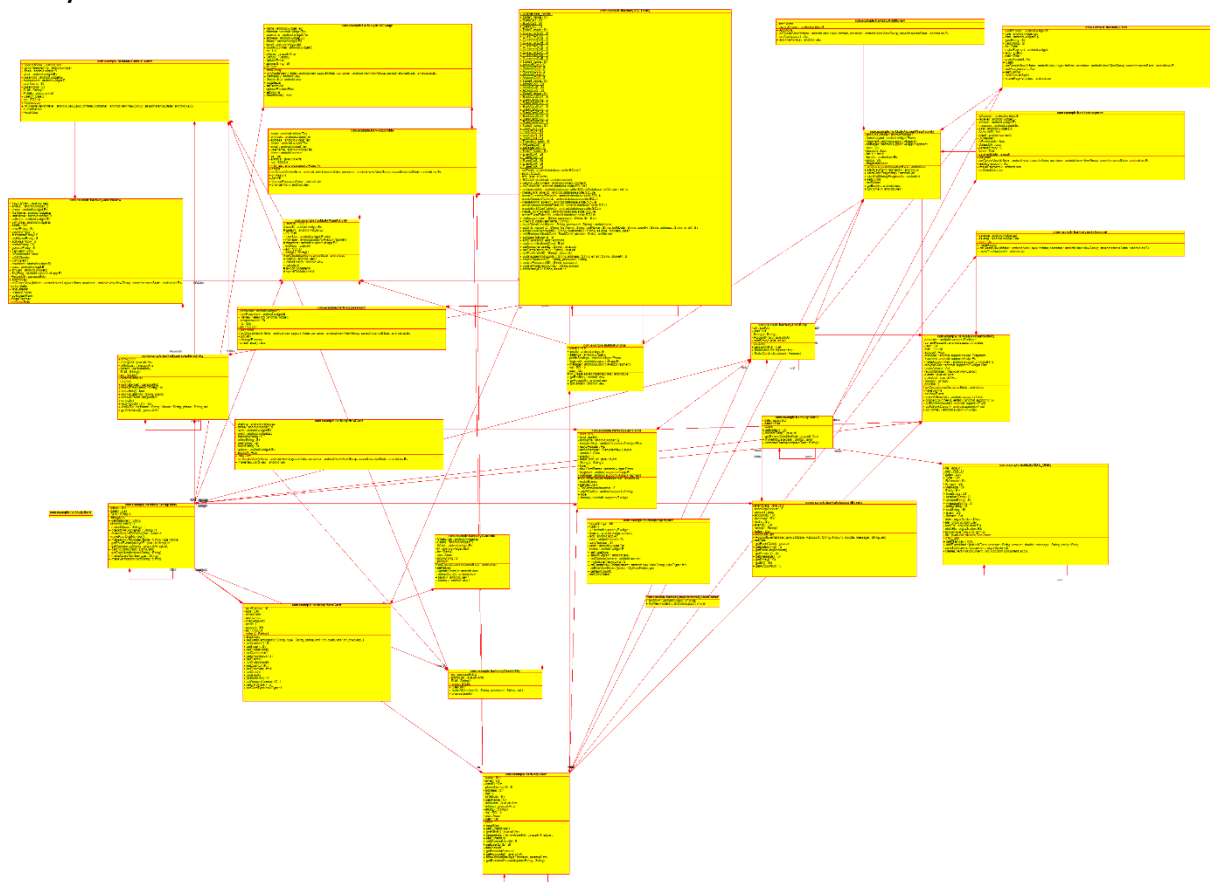


3. Ohjelman kuvaus

Ohjelma on siis pankin mobiili applikaatio, jolla käyttäjä voi hallinnoida tilejensä ja niihin liittyviä tapahtumia. Applikaation tulee pystyä Siirtämään rahaa tililtä toiselle, nostamaan rahaa, muuttamaan käyttäjän tietoja sekä sisään kirjautumisen tiedot ja käyttäjän persoonalliset tiedot. Näiden lisäksi sen applikaation kautta hallinnoidaan käyttäjän tileihin liittyneitä pankkikortteja niiden rajoja sekä tilitapahtumia. Applikaatiolla voisi olla myös mahdollisuus kirjautua admin tiliin jolla voisi hallinnoida kaikkia tilejä, mutta tätä ominaisuutta ei toteutettu vaikka siihen liittyvä tietorakenne tietokantaan onkin toteutettu.

4. Tietokanta toteutetaan applikaatioon javan sisäisellä SQL toiminnallisuudella. Sinne siis lisätään omaan luokkaansa manuaalisesti taulukot ja niiden attribuuttien tiedot käyttäen SQLHleper class:iä. Tänne on laitettu useita eri komentoja, jotka auttavat ohjelman ajossa hakemalla tietoja tietokannasta. Nämä haut tehdään getWritableDatabase avulla jonka rawQuary metodia käytetään haussa. RawQuary vastaa normaalia SQL komentoriville laitettua lausetta ja se palauttaa kursorin jossa on riveissä tarvittavat tiedot. Kursorista nämä tiedot otetaan tietokannan taulujen kolumni indeksien perusteella, jotka vastaavat taulujen eri attribuutteja

5 täydellinen luokkakaavio



6 Yhteenveto

Kokonaisuudessaan tietokannan toteutus ohjelmassa oli onnistunut, mutta teon yhteydessä löytyi useita mielenkiintoisia kohtia. Esimerkiksi kun tein tileihin liittyvien tapahtumien tietorakennetta oli ajatuksenani että tapahtumassa voi olla joko 1 tai 2 eri tiliä riippuen siitä onko tapahtuma tilisiirto vai käteisnosto. Päädyin kuitenkin niin tietokannassa että sitä kuvaavissa diagrammeissa että sen voi suoraan vain laittaa olevan yhden tilin tapahtuma ja jos toinen tili on tapahtumassa mukana niin tapahtuma vaan tallennetaan myös sen tilin tietoihin. Toinen asia mikä tuli esille oli heikossa relaatiossa, joka on tilin ja sen tapahtumien välillä. Siinä tilanteessa tilitapahtumalla tulee olla oma ID:nsä vaikka kyseistä avainattribuuttia ei käytettäisikään mihinkään sillä muuten tapahtuman tietoja ei voi järkevästi tallentaa tietokantaan, sillä mikään toisten attribuuttien liittäminen yhteen ei luo vahvaa/uniikkia avainta.

Tarkemmin tietokannan ohjelmallinen toteutus tehtiin kuten tehtävän annossa pyydettiin ja tästä enemmän olio-ohjelmoinnin osiosta.