

**LUT School of Engineering Science**

CT60A2411 Olio-ohjelmointi

23.7.2019

Dokumentaatio

Janetta Huoponen

Kesä 2019

# **Pankkiapplikaation määrittelydokumentti**

23.7.2019

Janetta Huoponen

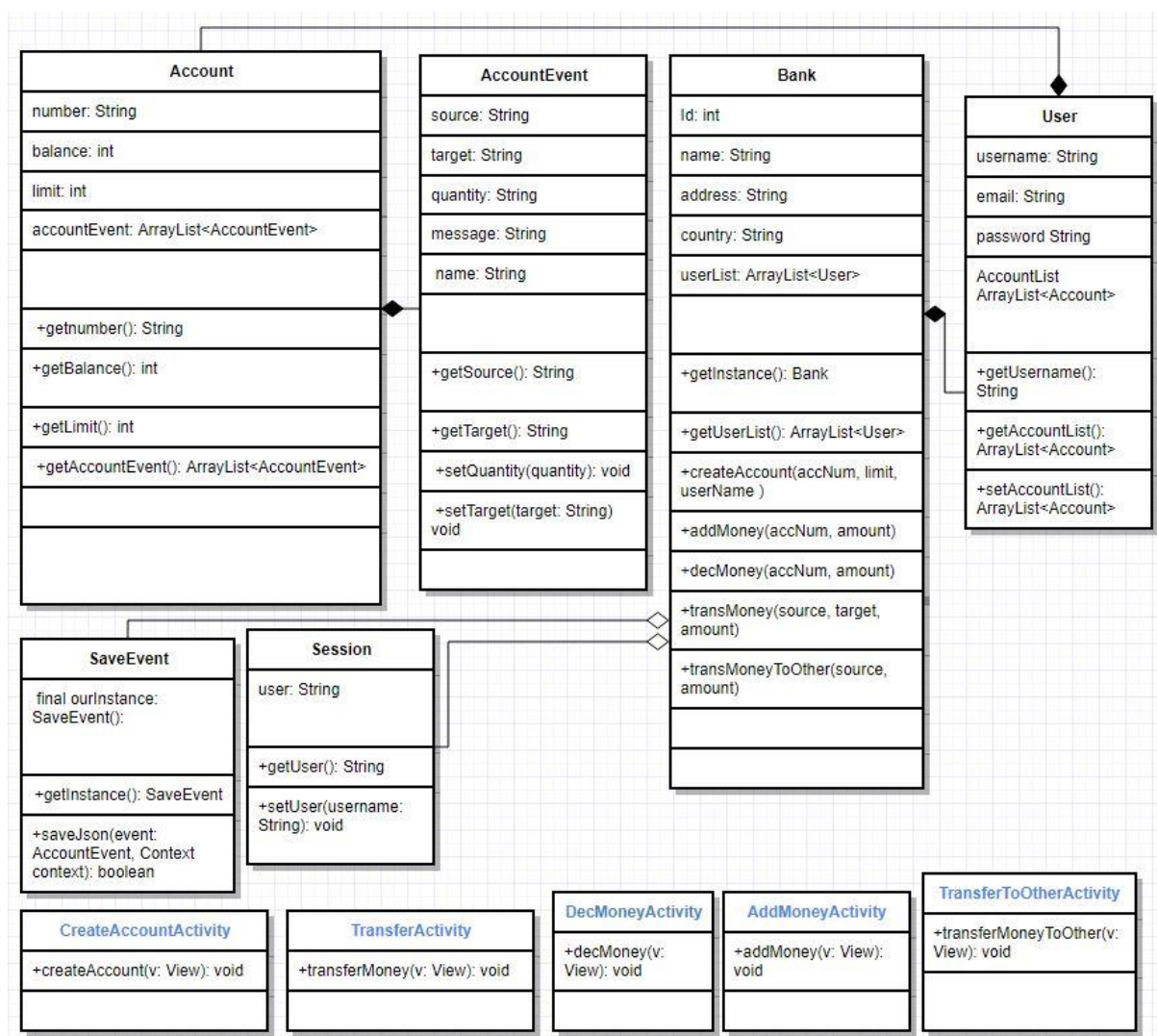
Tietotekniikka

# 1 KUVAUS, SUUNNITTELU JA TOTEUTUS

Tavoitteena oli toteuttaa pankkiapplikaatio, joka mahdollistaa rahan siirron omalta tililtä toiselle, rahan lisäämisen tilille, rahan nostamisen tililtä ja tilisiirtojen tekemisen ulkopuoliselle tilille.

Toteutin sovelluksen yksilötyönä ja toteutus eteni säännöllisesti kolmen viikon ajan käyttäen päivässä noin kolme tuntia harjoitustyön tekemiseen.

Pankkiapplikaation suunnittelussa hyödynsin luokkakaaviota (Kuva 1), johon visualisoin sovelluksessa käytettäviä luokkia ja niiden attribuutteja. Lisäksi esittelin luokkakaaviossa oleville luokille kuuluvia toimintoja eli metodeja. Luokkakaaviossa esittelen luokkien välisiä suhteita luokkakaavioon liittyvillä viivoilla.



Kuva 1. Pankkiapplikaation luokkakaavio

Pankkiapplikaatiossa luokkien välille on rakennettu kompositio ja kooste suhteita. Kärki on laitettu osoittamaan siihen luokkaan, joka on vastuultaan tärkeämpi ja koostuu applikaation muista olioista. Mustalla kärjellä korostetaan luokkien välisten suhteiden tärkeyttä.

Käytin pankkiapplikaation toteutuksessa Android-käyttöjärjestelmän virallista ohjelmointiympäristöä Android Studioita. Android Studiosta käytin versiota 3.4.1. Pankkisovelluksen toteutuksessa hyödynsin Android Studion SDK-työkalua, joka sisältää muun muassa sovelluskehitykseen liittyviä kirjastoja, debuggaustyökalun ja Android Emulatorin. Pankkisovelluksessa käytettiin hyödyksi erilaisia riippuvuuksia, joita hallinnoitiin Android Studion Gradle-työkalulla. Gradle vastaa sovelluksen riippuvuuksista muihin kirjastoihin, koodin kääntämisestä ja tiedostojen pakkaamisesta APK-tiedostoiksi.

Pankkiapplikaatio toteutettiin hyödyntäen aktiviteettejä, jotka tarjoavat käyttäjälle erilaisia toimintoja ja niihin liittyviä näkymiä. Näkymä on yleisesti XML-tiedosto. Tiedostossa oleviin elementteihin sidotaan aktiviteetin erilaisia toimintoja. Tästä voidaan mainita esimerkiksi sovellukseen toteutettu `CreateAccountActivity`, johon oli luotu julkinen `createAccount` – niminen metodi, joka sidottiin `CreateAccountActivity`n XML-tiedostossa olevaan painikkeeseen, jossa loogisesti luodaan uusi tili. Pankkiapplikaation `MainActivity` – aktiviteetti on määritelty sovelluksen `AndroidManifest.xml` -tiedostossa pääaktiviteetiksi, joka käynnistetään aina kun sovellus avataan. `MainActivity` –komponentissa on otettu käyttöön työkalupalkki, johon on lisätty vasempaan yläreunaan valikko, joka mahdollistaa käyttäjälle nopean navigoinnin eri käyttöliittymien välillä.

Pankkiapplikaatiossa käyttäjällä on mahdollisuus siirtää rahaa omalta tililtä toiselle, lisätä rahaa tilille, tehdä rahanostoja ja tilisiirtoja ulkopuolisille tileille.

Harjoitustyö kattaa kaikki perusvaatimukset ja niiden lisäksi laajennettujen vaatimusten ensimmäisen vaatimuksen eli pankkiapplikaatio on rakennettu hyvin suunnitelluista UI-komponenteista. Näiden perustelujen pohjalta tavoittelen harjoitustyöstä 16 pistettä.

Harjoitustyön aikana opin sovelluskehityksen suunnittelun merkityksestä. Luokkakaavion suunnitteluvaiheessa oli tarkoin pohdittavia tarvittavia luokkia, luokkien välisiä suhteita ja millainen vastuu yksittäisellä luokalla on sovelluksessa. Toiminnallisuuksien suunnittelussa oli

huomioitava käyttäjä ja käyttötilanne. Käyttöliittymä tuli toteuttaa niin, että se on helppokäyttöinen ja käyttäjälle mieluista. Lisäksi opin MVC-arkkitehtuurin toimintaperiaatetta, jonka tarkoituksena on erottaa käyttöliittymä sovellusalueesta. Esimerkiksi harjoitustyötä tehdessä ymmärsin hallintaluokan merkityksen ja kuinka sovellus jaetaan loogisesti osiin.