

CT60A2411_07.01.2020 Olio-ohjelmointi

Harjoitustyö, Pankkiapplikaatio

Tekijät:

Taru Haimi 0565878

Leevi Kämäräinen 0568396

Joona Ylijoki 0522866

Kuvaus ohjelmasta

Teimme verkkopankkisovellusta jäljittelevän sovelluksen, jossa on myös mahdollisuus simuloida korttimaksua ja rahannostoautomaatin toimintaa.

Kirjautumisnäkyelmästä voidaan joko luoda uusi käyttäjä, Admin-käyttäjä (vain 1 kpl mahdollista), poistaa kaikki käyttäjät tai yksinkertaisesti kirjautua sisään. Main Menusta päästään 1) muokkaamaan käyttäjän henkilötietoja ja salasanaa 2) tarkastelemaan käyttäjän tilejä ja kortteja sekä lisäämään niitä 3) kirjautumaan ulos ja 4) simuloimaan korttimaksua ja rahannostoa.

Molemmissa simulaatiossa valitaan kortti, jota halutaan käyttää; rahamäärä, joka halutaan nostaa/tuhlata; maa-alue, jossa korttia halutaan käyttää; ja pinkoodi, jotta maksu/nosto saadaan suoritettua. Korttimaksussa on lisäksi mahdollista maksaa lähimaksulla ilman pinkoodia, jos kortissa on kyseinen ominaisuus ja tuhlattava summa on riittävän alhainen.

Tilejä luodessa luodaan vain tili ja tilityyppi, korttia luodessa valitaan myös, mihin tiliin kortti linkitetään, onko lähimaksua, ja asetetaan pinkoodi. Tilietietojen näkyelmästä voidaan muokata joitakin tilin ominaisuuksia, tehdä tilisiirto omalle, toisen käyttäjän olemassa olevalle tai mille tahansa tilille, tarkastella tilitapahtumia ja poistaa tili. Samasta näkyelmästä päästään myös kyseisen tilien kortteihin, josta sitten päästään korttien omiin asetuksiin. Sieltä voidaan muokata joitakin korttien ominaisuuksia ja poistaa kortti. Ainoastaan Admin-käyttäjänä voi lisätä rahaa kaikille tileille.

Kuka teki ja mitä

Taru: Suunnittelin ja tein suurimman osan UI-pohjasta, eli hahmottelin mitä kaikkia näkymiä ensinnäkin tarvitsimme ja mistä näkyelmästä on tarkoitus mennä mihinkäkin. Koska työhömme tuli paljon erillisiä käyttöliittymiä, oli kätevää ensin luoda mahdollisimman paljon ei mitään -tekeviä näkymiä, jotta koko sovelluksen runko hahmottuu selkeästi, ja siihen päälle sitten jälkikäteen koodata toiminnallisuuksia. Itse koodauksessa myöhemmässä vaiheessa huolehdimme aika paljon näkymien widgettien toimivuudesta, ja päävastuullani olivat kirjautuminen, rekisteröityminen ja simulaatiot. Toki tein myös yksittäisiä toiminnallisuksia ja virheiden korjailua sieltä täältä. Eli lyhyesti tein pohjatyön ja koodailin sitten vähän sieltä täältä kaikkea mitä tarvitsi ja puuttui.

Leevi: Hoidin aika pitkälti luokkien (User, Account, Card) toimivuudesta. Tein meidän työllemme tiedon tallentamisen Android studioon Gson-kirjaston avulla SharedPreferencesiin jonka avulla saimme sovelluskohtaisesti tiedon tallentumaan joko puhelimeen tai emulaattoriin. Rakensin myös meille rakenteen millä voimme tallentaa olioita toistensa sisällä oleviin listaan ja päätasolla tallentamaan Userit omaan listaansa (ArrayList). Olin myös vastuussa tilien ja korttien monista ominaisuuksista kuten tilien/korttien luomisesta ja niillä maksamisesta. Lyhyesti sanottuna rakensin olioille arraylist tietorakenteen, olin mukana olioiden/niiden metodien tekemisessä sekä olin vastuussa sovelluksen tiedon tallentumisesta SharedPreferencesiin.

Joona: Loin pääasiassa toimintoja korteille ja tileille. Tein kortin tietoja ja asetuksia käsitteleviä toimintoja ja näkymiä (CardInfo ja EditCardSettings). Loin myös tilitapahtumien tiedostot ja näkymät (TransActionInfo ja TransActionInfo) sekä työstin tilitapahtumien luontia. Tarkoituksemme oli saada ensin tilitapahtumat toimimaan csv-tiedoston kautta ja csv:n kanssa menikin minulla aika kauan aikaa. Lähinnä ongelmien ratkaisua. Emme kuitenkaan valitettavasti saaneet sitä lopulta onnistumaan vaan päädyimme sitten luomaan tilitapahtumat ArrayListina, jonka luonnissa olin mukana. Korjailin myös muita virheitä mitä tuli vastaan testiajojen aikana.

Yksityiskohtaisemmat kuvaukset siitä, mitä kukakin teki, löytyvät GitHubin repositorysta tiedostosta "harkkatyo_changelog.txt".

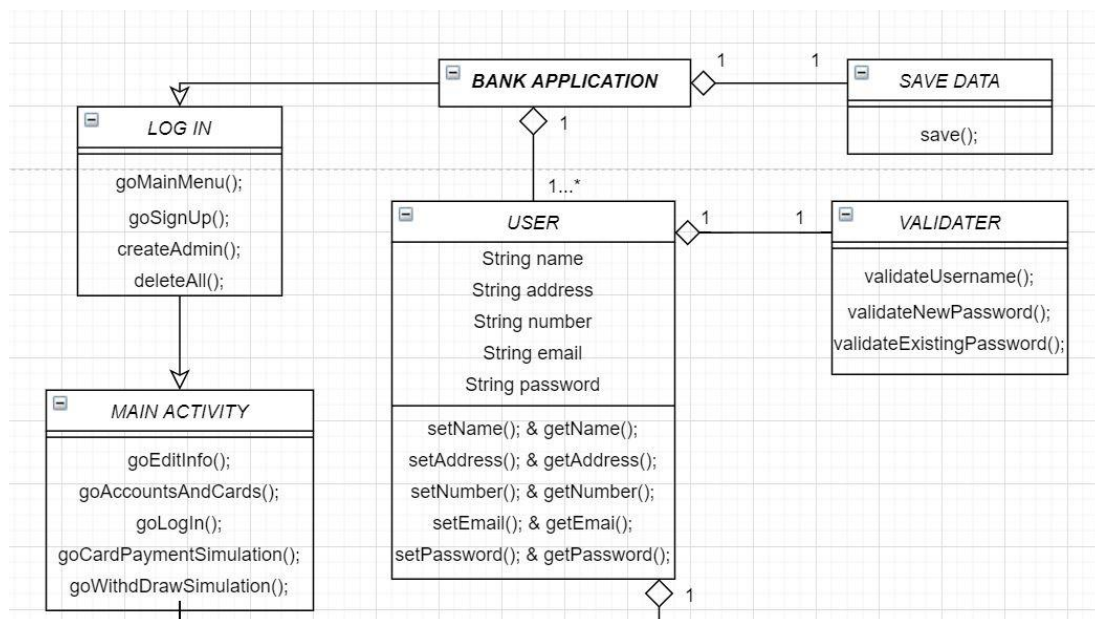
Ohjelman suunnittelu ja toteutus

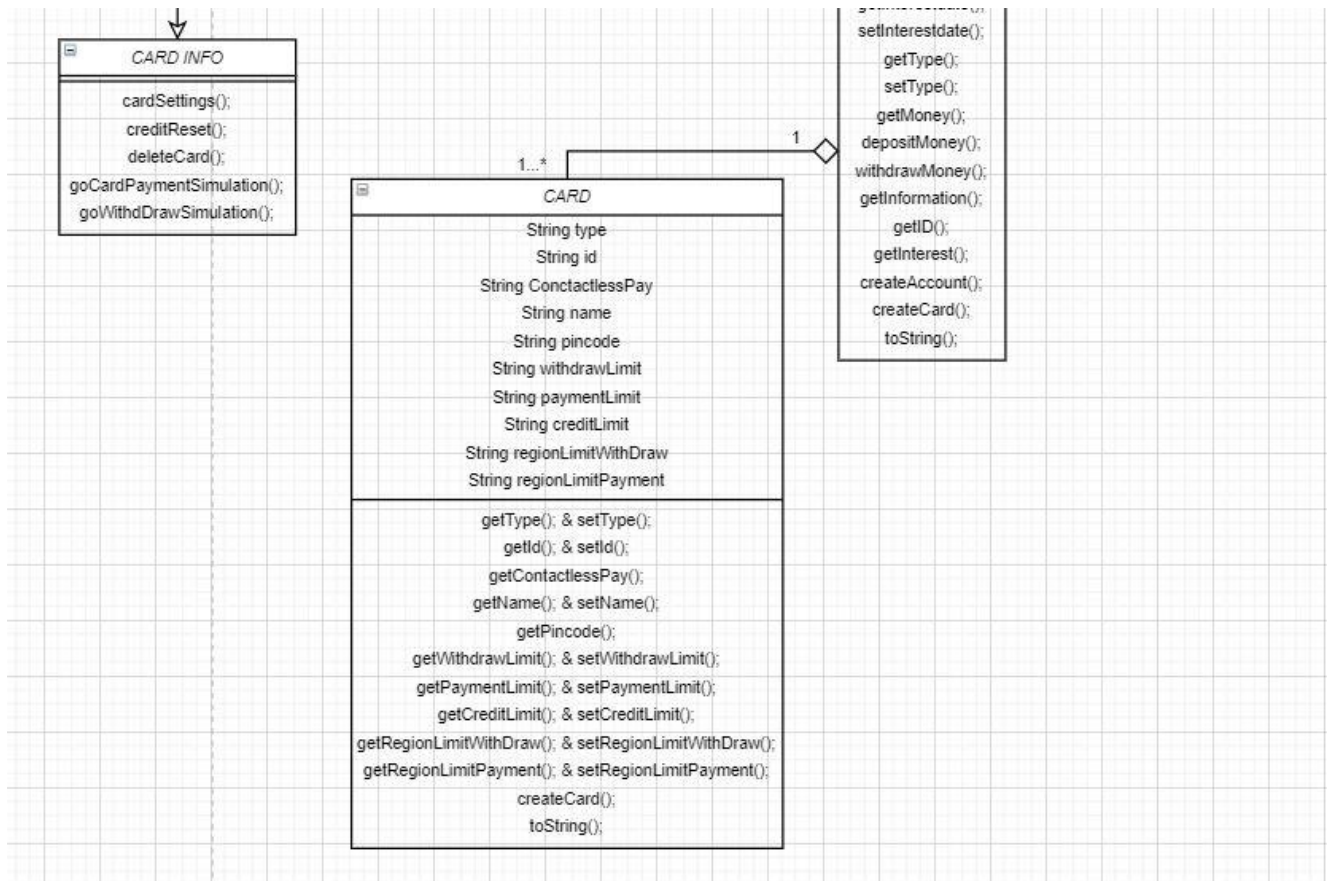
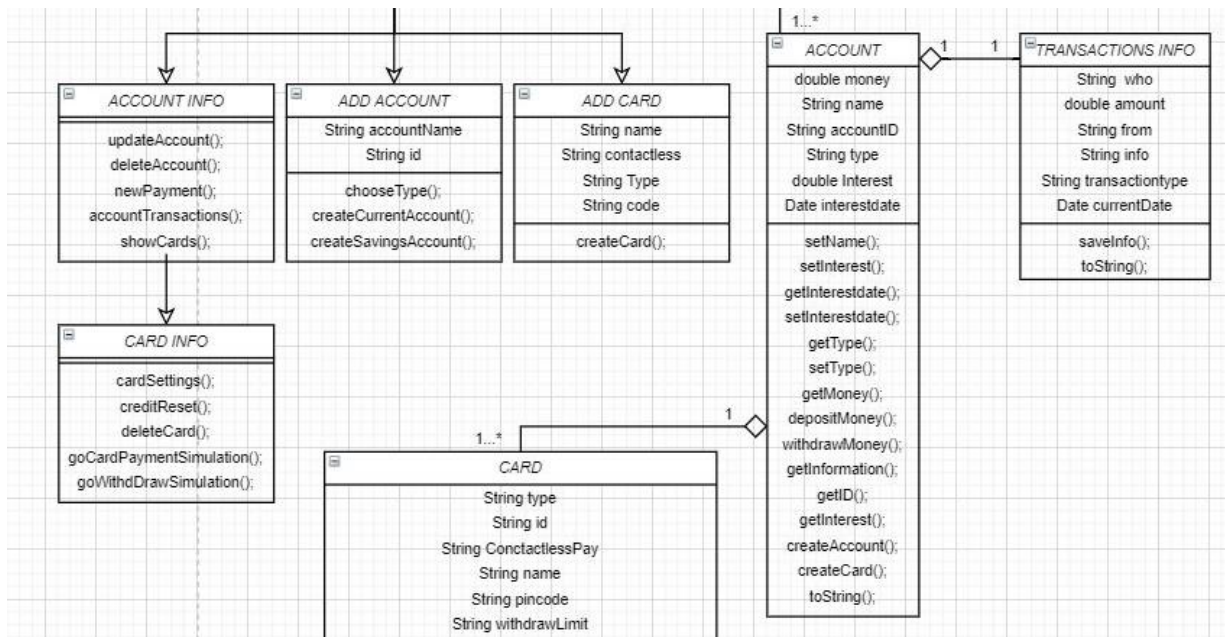
Teimme melko kattavan suunnitelman jo harkkatyön suunnitelman palautuksessa siitä, mitä meillä olisi aikomuksena ja haluna tehdä. Itse ohjelmaa lähdimme toteuttamaan aika pitkälle paperille piirrettyjen näkymähahmottelujen perusteella; mitä kaikkia ikkunoita tarvitaan ja mistä ne avataan, aloittaen MainMenusta ja siitä sitten vähitellen syvemmälle sovellukseen uusi käyttöliittymä kerrallaan. Monet käyttöliittymät luotiin ensin vain olemaan olemassa, ja toiminnallisuudet lisättiin vasta myöhemmin, kun samaa ominaisuutta tarvittiin muuallakin. Ainoat isommat poikkeamat suunnitelmaamme tapahtuivat tili- ja korttiluokkia luodessa, muuten pysyimme aika hyvin suunnitelmassamme.

Käytännöntoteutuksessa koodasimme aika paljon yhdessä (paikanpäällä/Discord; neuvoimme toisiamme/koodasimme yhdellä koneella), joten repositoryssa ei myöskään näy konkreettisesti kaikkien ryhmän jäsenten tekemä työmäärä. Työnjakoa toteutettiin aina päivä kerrallaan; sovittiin kuka lähtee keskittymään minkäkin tekemiseen, mitä tarvitsee saada kuntoon tai korjata, jne. Aika sujuvasti sumplittiin ja heiteltiin palloa toiselle tilanteen mukaan.

Luokkakaavio

Teimme luokkakaavioon näkyviin oliot/luokat ja mielestämme tärkeimmät käyttöliittymät:





Toteutetut ominaisuudet ja toiminnot

Pakolliset ominaisuudet:	Lisätoiminnot ja ominaisuudet:	Muuta kivaa:
Olio-ohjelmoitu	Käytännölliset UI-komponentit 2p	Erikseen käyttö- ja säästötili, säästötilillä 2% toimiva korko
5 luokkaa & oliota	Admin-käyttäjä 3p	Käyttäjien, tilien ja korttien poisto (erikseen ja yhdessä)
Datan kirjoitus tiedostoon (JSON)	Mahdollisuus useisiin käyttäjiin ja niiden tietojen tallennus 3p	Pankkikorteilla todelliset pinkoodi, luotto- ja lähimaksuominaisuus, joita käytetään simulaatioissa.
Käyttäjän tietojen muokkaus	Applikaatioon kirjautuminen 3p	Todellisuutta aika hyvin mallintavat ja vaadittua monipuolisemmat simulaatiot
Tilien luominen	Hyviä salasanan sääntöjä noudattava salasana ja "avainlukukoodi" 3p	Oma logo
Tilien ominaisuuksien muokkaus	Pankkikorttien ominaisuuksien muokkaus 1p	
Pankkikorttien luominen tileihin	Rahan siirto omille tileille 1p	
Tilitapahtumien tarkastelu	Rahan nosto ja korttimaksu rajoituksineen 1p	
Rahan lisäys tilille	Tilisiirto ulkopuoliselle tilille ja toiselle käyttäjälle 1p	
Harjoitustyön suunnitelma 3p	Korteissa maarajoitukset, voidaan testata simulaatioissa 2p	
3 hlö:n ryhmätyö -4p		
12p	19p	2-3p
Yhteensä: 33-34p		

Arvioitu ajankäyttö

Taru Haimi: 7pv x 5-10h + muu apu = n. 65h

Leevi Kämäräinen: 8pv x 5-10h + muu apu = n. 70h

Joona Ylijoki: 7pv x 3-10h = n. 40h

Mitä opimme harjoitustyöstä?

Taru Haimi: Harjoitustyö ja koko kurssi itsessään opetti minulle näin fuksina ja keltanokkana koodarina mitä koodaus käytännössä on; trial & errorilla pääsee pitkälle, ja jos et osaa, älä valita opintomonisteen puutteesta vaan etsi googlesta (Stack overflow on nykyään paras ystäväni). Opin käyttämään Android Studiota melko monipuolisesti ja hahmottamaan mitä kaikenlaisia ominaisuuksia sillä on mahdollista sovellukseen halutessaan tehdä; nyt kun käyttää omaa puhelinta niin rupean välillä ajattelemaan, miten tämä on koodattu/miten itse saisin tehtyä samanlaisen näkymän. Opin myös paljon ryhmätyöskentelystä ja itsestäni ryhmätyöskentelijänä; teen mielelläni oman tahtoni ja visioni mukaan asioita eikä aina ole helppo "alistua" muiden mielipiteisiin tai edes välttämättä neuvotella vaihtoehtoisista toimintatavoista/ toteutuksista ilman pientä kinastelua, mutta tässä parin kuukauden aikana oma asenteeni ryhmässä

työskentelyyn on muuttunut ja tämä on oikeastaan aika kivaa, eikä kinaan johtavia erimielisyyksiä enää oikeastaan tule. Kaiken kaikkiaan oli siis aika opettava ja kasvattava kokemus tämä harkkatyö. Oli muutenkin mielekästä työstää tällaista harkkatyötä, jossa pääsee tekemään käytännönläheistä sovellusta ja koko ajan näkemään oman työnsä jäljen.

Leevi Kämäräinen: Opin erittäin paljon siitä kuinka koodataan Javalla olio-ohjelmointia sekä siitä, kuinka Android pohjaista sovellusta rakennetaan. Opin paljon siitä, kuinka todella monella erilaisella tavalla voi koodatessa ratkaista saman ongelman. Osaan paremmin nykyään googlaamaan koodiini liittyviä ongelmia sekä pystyn aika pitkälle itsenäisesti koodaamaan, vaikka se ei aina olisi sitä parhainta koodaustyyliä. Opin ryhmätyöskentelystä sitä, että pitää antaa muillekin mahdollisuus ideoida ja tehdä töitä, eikä itse yrittää hamuilla kaikkia hommia/ideointia itselle. Opin hahmottamaan sovelluksien kokonaisuutta paremmin, sekä näkemään jo aika pitkälle sen että mitkä ideat ovat helposti toteutettavissa ja mitkä vaativat enemmän töitä. Jälkeenpäin ajateltuna koodista olisi saanut selkeämpää /lyhyempiä jos olisi tallentanut pidempiä koodin pätkiä muuttujiin ja käyttänyt näitä muuttujia sitten uudelleen tarvittaessa koodissa.

Joona Ylijoki: Opin harjoitustyötä koodatessa monta sellaista asiaa, jotka tuntuivat alkuun Eclipsellä koodatessa menevän vähän ohi ja ihmetyttävän, että miksi tämä nyt näin toimii. Muun muassa ArrayListin käyttö valkeni paremmin harjoitustyön aikana. Oli hetkiä, kun en tajunnut jotain mitä toinen oli koodannut, mutta kysyessä sain selityksen ja ymmärsin sitten, että miten ja miksi näin oli. Harjoitustyötä oli myös mukava tehdä Android Studiolla, vaikkei sekään ollut alkuun kovin hyvin hanskassa, koska siinä pystyi suoraan havaitsemaan kirjoittamansa koodin vaikutuksen ja toiminnan käytännön kautta. Visuaalisten näkymien teko oli myös opettavaista ja mielenkiintoista puuhaa. Ryhmätyöskentelyssä päädyttiin keskustelun kautta aina hyvin kompromissiin eikä erimielisyyksiä liialti tullut.

Helpot ja vaikeat ominaisuudet ja toiminnot

Helppoja ominaisuuksia/toimintoja

- Kokonaisuudessaan ei ollut mitään ylitsepääsemättömiä ongelmia, jotka eivät olisi ratkenneet riittävällä pään seinään hakkaamisella.

Vaikeita ominaisuuksia/toimintoja

- Tiedostojen käsittelyssä tuli ongelmia, kun viikkoharjoituksissa ei käyty läpi kuin .txt-tiedoston kirjoitus ja luku sekä xml-tiedoston luku. (esim. csv-tiedoston käsittely, yritimme melko pitkään saada sitä toimimaan, mutta jouduimme lopulta luopumaan siitä).
- Tietokannan toteutus ja tiedon hallinta; näin fukseina ei meillä tosiaan ollut mitään käsitystä, miten tietokannan kautta olisi kannattanut yrittää käsitellä/tallentaa tietoja, joten nyt vaikka kaikki data on tallessa (olioissa ja oliot listoissa) ja sovelluksemme toimii kätevästi ja ”oikeaoppisesti”, ei tällainen ratkaisu pidemmän päälle / oikeassa elämässä olisi millään tavalla järkevä.
- Lähdimme tekemään sovellusta aluksi vain yhdelle käyttäjälle ja käytimme tietojen säilymiseen shared preferenssiä. Myöhemmin päädyimmekin moniin käyttäjiin, ja vaihtaminen tietojen tallentamiseen gson-kirjaston avulla oli melko työlästä ja aikaa vievää, ja kyseisen kirjaston käyttäminen tuotti jotain ongelmia myöhemmissä vaiheissa.

Muuta:

Mielestämme harkkatyön ohjeistus ja pisteytys olivat sopivat: ohjeistuksessa tulee selkeästi ilmi vaaditut asiat ja monipuolisesti ideoita mitä kaikkia ominaisuuksia voisi tehdä, ja pisteitä sai aika sopivasti per ominaisuus, muutama huomio:

- Hyvän salasanan vaatiminen ja avainkoodin tekeminen tuntui yllättävän helpolta suhteessa siitä saataviin pisteisiin (verrattuna joihinkin muihin toimintoihin).
- Simulaatioista tuleva 1p tuntui taas vähän pieneltä siihen nähden, miten monipuoliset itse satuimme niistä tekemään (vaikka helpommalla nekin olisi saanut halutessaan tehtyä, jolloin tuo 1p on ymmärrettävä)