

## Työtunnit

Nimi	Pvm	Tunnit	Mitä tein
Leevi Raassina	10.4	5h	Githubin käyttöönotto ja testailu, drawer menun teko
Miska Hämäläinen	10.4	5h	Githubin käyttöönotto ja testailu, drawer menun teko
Leevi Raassina	12.4	3.5h	Login screen + SQLite database
Miska Hämäläinen	12.4	2h	Tämän dokumentin työstäminen
Leevi Raassina	14.4	5h	Hyvän säännön salasana + avainlukukoodi
Miska Hämäläinen	14.4	5h	SHA-512 + salt + login verify
Miska Hämäläinen	15.4	3h	SHA-512 + salt + login verify VALMIS
Miska Hämäläinen	18.4	2h	Tämän dokumentin työstäminen
Miska Hämäläinen	28.4	3h	Admin tunnus
Leevi Raassina	28.4	3h	Pankkitilien epätoivoista säätöä (Ei toimi)
Leevi Raassina	29.5	4h	Pankkitilit + Json
Leevi Raassina	30.5	3h	Asetukset
Miska Hämäläinen	31.5	3h	Kaikenlaisia pikku korjauksia
Leevi Raassina	31.5	3h	SE TOIMII
Leevi Raassina	4.6	2h	Korttien teko pohjustusta
Leevi Raassina	5.6	3h	Lisää kortti juttuja
Leevi Raassina	2.8	6h	Kortti juttuja & bugi korjauksia
Leevi Raassina	3.8	3h	Bugi fixejä
Leevi Raassina	4.8	5h	Viimeistelyjä + koodin siivousta

Huom. Tunnit ovat todella suuntaa antavia, ja sieltä puuttuu monta päivää.

## Suunnitelma ja toteutus

<b>Pakolliset yleiset ominaisuudet</b>	
Oljo-ohjelmoitu <b>VALMIS</b>	
Vähintään viisi erilaista luokkaa & oliota (käyttöliittymäluokkia ei lasketa) <b>VALMIS</b>	
Tärkeiden tietojen kirjoitus tiedostoon (XML, JSON tai CSV) <b>VALMIS</b>	
Aihe spesifit perustoiminnot (löytyvät jokaisen aiheen alta listattuna) <b>VALMIS</b>	
<b>Yleiset ominaisuudet</b>	<b>Pisteet</b>
Tietokanta SQLite <b>VALMIS</b>	5
Useampi käyttäjä <b>VALMIS</b>	3
Kirjautuminen applikaatioon <b>VALMIS</b>	3
Salasan hash & salt <b>VALMIS</b>	3
Rakennettu hyvin suunnitelluista UI-komponenteista <b>VALMIS</b>	2
Kirjautumisen salasana noudattaa hyvän salasan sääntöjä <b>VALMIS</b>	3
	<b>19</b>
<b>Spesifit ominaisuudet</b>	<b>Pisteet</b>
Voi siirtää rahaa omalta tililtä toiselle <b>VALMIS</b>	1
	<b>1</b>
<b>Pakolliset spesifit ominaisuudet</b>	
Voi muokata käyttäjän tietoja (nimi, yhteystiedot, jne.) <b>VALMIS</b>	
Voi luoda käyttäjille tilejä (oikeasti pitäisi tehdä pyyntö pankille, tässä kuitenkin voi tehdä suoraan) <b>VALMIS</b>	
Voi muokata tilien ominaisuuksia (rahan siirto tileiltä toisille) <b>VALMIS</b>	
Voi luoda tileihin pankkikortteja (oikeasti pitäisi tehdä pyyntö pankille, tässä kuitenkin voi tehdä suoraan) <b>VALMIS</b>	
Voi tarkastella tilitapahtumia (siirrot, nostot, talletukset, maksut, jne.) <b>VALMIS</b>	
Voi lisätä rahaa tilille <b>VALMIS</b>	
	<b>Pisteet</b>
<b>Yleiset + spesifit</b>	<b>20</b>
<b>Yleiset + spesifit + pakolliset</b>	<b>33</b>

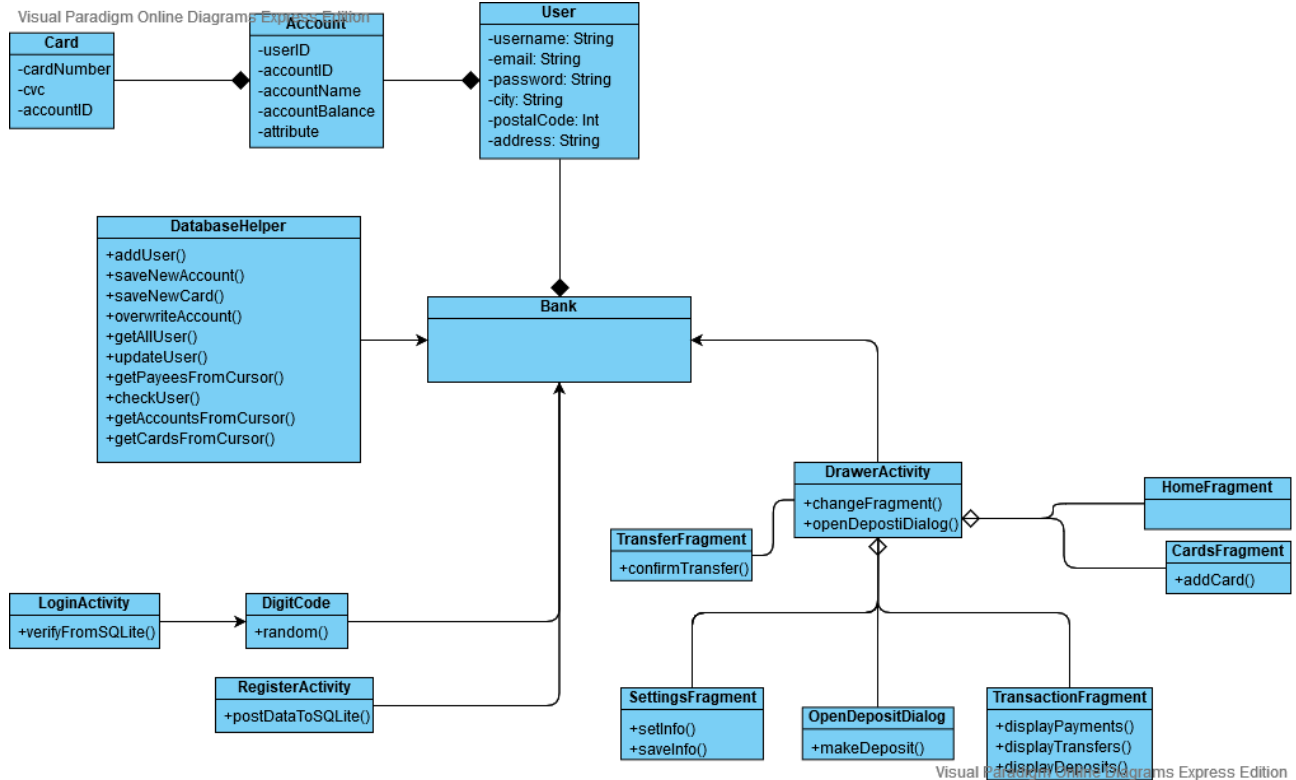
OLIO Harkkatyö - Pankkiapplikaatio  
Miska Hämäläinen  
Leevi Raassina  
Pankkiapplikaatio

Lähdimme liikkeelle päivittämällä luokkakaavion kommenttien perusteella, jotka saimme harjoitustyösuunnitelman palautukseen, jonka jälkeen lähdimme suoraan ohjelmoimaan. Varsinaista järjestystä missä lähdimme tekemään ominaisuuksia ei ollut, valitsimme sellaisen mikä kuulosti helpolta, eli 'Navigation drawer' valikon, joka ei isompia ongelmia tuottanut. Seuraavana lisäsimme sisäänkirjautumis- ja rekisteröinti ikkunan ja niiden toiminnallisuudet.

Otimme käyttöön SQLite tietokannan, johon tallennamme käyttäjätiedot eli juokseva id, nimi, salasana, sähköposti, tilintyyppi (user/admin) ja käyttäjäkohtaisen suolauksen. Tietokannan rakenteen valmistuttua teimme toiminnallisuudet, jossa rekisteröitymisen jälkeen käyttäjä tallennetaan tietokantaan. Salasana salataan SHA-512 salauksella ja suolataan käyttäen Javan tarjoamaa satunnaisgeneraattoria SecureRandomia. Sisäänkirjautumisen validointi meni täysin rikki tässä vaiheessa, koska tietokantaan tallennettu String joka piti muuntaa Byte[] muotoon MessageDigestin updatea varten aiheutti ongelmia, tämä korjattiin käyttämällä ISO 8859-1 merkistöä joka ei hävitä merkkejä String -> Byte -> String muunnoksen aikana. Todennäköisesti tämän olisi voinut tehdä helpommin mutta tämä on nyt tehty omalla tavalla.

Loimme checkAdmin methodin jolla tarkistetaan onko käyttäjä admin hakemalla tietokannasta käyttäjän user\_type- sarake johon rekisteröinti vaiheessa merkataan joko user tai admin. Lopulta tosin päätimme jättää admin-toiminnot tekemättä, joten tästä tuli turha metodi.

Käytämme JSONia tallentamaan entisen käyttäjän tiedot ja sen avulla haemme tiedot tililistaan, korttilistaan ja tilitapahtumiin.



## Mitä opin harjoitustyöstä?

Miska Hämäläinen

- Ei kannata lähteä soitellen sotaan
- Eri character set enkoodaukset aiheuttaa päänvaivaa, merkit eivät olekaan sitä mitä luulet
- Tietokannat ovat käteviä tietojensäilytykseen vrt. tiedostot
- Itsevalittavat ominaisuudet ovat paljon hausکمپia tehdä kuin pakolliset
- Vielä parempi suunnittelu olisi helpottanut koko projektia.

Leevi Raassina

Harkkatyöstä opin sen että tee se jumalauta ajallaan. Deadlinen lähestyessä alkoikin olemaan jo niin paljon muita deadlinejä ja intensiivi kursseja, että harkan tekeminen vain jäi. Opin myöskin sen, että älä turhaudu jos jokin ei toimi. Monta kertaa kävi niin että jaksoi koodata tunnin ja meni motivaatio, kun mikään ei yksinkertaisesti toiminut. Ja lopuksi opin myöskin sen että google on todella hyvä ystävä koodatessa.