



Adnan Avni, Ilona Juvonen, Perttu Harvala, Roope Kylli

Expense Tracker

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintätekniikan tutkinto-ohjelma

Tekninen dokumentaatio

3.5.2023

Sisällysluettelo

1	Johdanto	1
2	Sovelluksen visio	2
3	Sovelluksen toiminnallisuus	3
3.1	Tietokanta	4
3.2	Käyttöliittymä	5
3.3	Budjetti	6
3.4	Menot	6
3.5	Tulot	7
3.6	Lokalisointi	7
4	Ohjelmointitekkinen toteutus	8
4.1	Käytetyt ohjelmointikielet ja kirjastot	8
4.2	Arkkitehtuuri	8
4.3	Käyttöliittymä	9
4.4	Sisäinen logiikka	9
4.5	Testaus	10
4.6	Kehitysprosessi	10
5	Ohjelmiston toiminta	11
6	Sovelluksen käyttö	14
7	Sovelluksen jatkokehitys	22
8	Yhteenveto	22

1 Johdanto

Tämä dokumentti sisältää sovelluksemme tärkeimmät tiedot, pääpiirteet ja toiminnan. Dokumentin tarkoitus on kuvantaa sovelluksen toimintaa tekstimuodossa. Dokumentti on suunnattu niille, jotka haluavat tietää yksityiskohtaisemmin, miten sovellus toimii ja miten se on kehitetty sekä myös mikä on ollut sovelluksen kehitysprosessi.

Dokumentti aloitetaan esittelemällä sovelluksen visio. Tämän jälkeen tarkastellaan sovelluksen toiminnallisuuksia yleisesti. Näiden jälkeen dokumentissa tarkastellaan sovelluksen ohjelmointitekniistä toteutusta erilaisista näkökulmista. Seuraavaksi käsittelyssä on sovelluksen kehitysprosessi ja kehitysvaiheen tekniikoista hieman asiaa. Toiseksi viimeisin osio dokumentissa on sovelluksen jatkokehitys ja muut asiat, mitä sovellukselle tullaan vielä tekemään tulevaisuudessa. Dokumentti päättyy lyhyeen yhteenvetoon.

Expense Tracker on työpöytäsovellus, jonka avulla käyttäjä voi seurata omaa rahankulutusta. Käyttäjä voi asettaa oman kuukausibudjetin, josta käyttäjä pysyy aina vähentämään käyttämänsä summan. Näin sovelluksen käyttäjä pystyy tarkkailla kuukausisuunnitelmassa pysymistään muun muassa eri visuaalisten kaavioiden avulla. Sovellukseen pystyy myös lisäämään omat tulot ja muut tulotiedot, mitkä helpottavat vielä enemmän käyttäjää tarkastelemaan omaa rahankäyttöä.

2 Sovelluksen visio

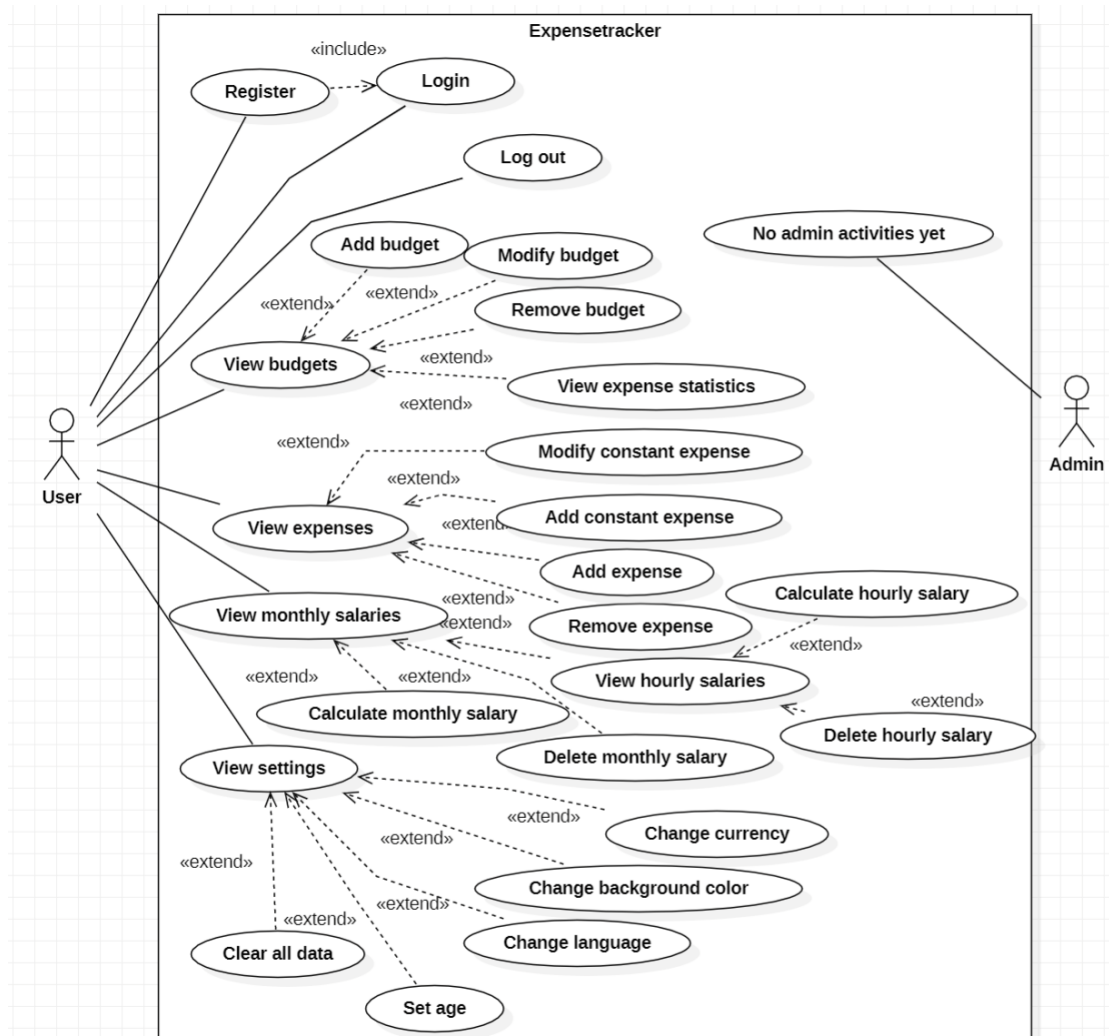
Expense tracker on suunniteltu auttamaan käyttäjiä seuraamaan heidän taloudellista tilannettansa sekä varojensa käyttöä. Suurin syy ohjelman kehitykselle on ihmisten kamppailu budjetin pitämisessä ja heidän rahankäytön seurannan tarve.

Ohjelma on kehitetty yksinkertaiseksi ja käyttäjäystävälliseksi. Käyttäjä asettaa kirjautuu tai rekisteröityy sovellukseen, jonka jälkeen hänen on asetettava aktiivinen budjetti. Budjetteja voi olla monta erilaista eri tarkoituksiin, ja käyttäjä itse määrittää budjetin nimen, sekä määrää kyseisen budjettitavoitteen. Käyttäjä voi sitten vähentää menojaan erilaisten kategorioiden avulla aktiivisesta budjetistaan. Käyttäjällä on myös mahdollisuus asettaa jatkuvia menoja, kuten esimerkiksi vuokra tai vesimaksu, jotka tallentuvat ja ovat jatkuvasti käytettävissä ja vähennettävissä valitusta budjetista. Käyttäjä voi myös seurata rahankäyttöä erilaisista graafeista. Tulot käyttäjä voi asettaa niille kuuluville välilehdellä. Tuloja voi joko laittaa vakiona kuukausipalkkana tai tuntipalkan asettaminen päivittäin on myös mahdollista. Sovelluksessa on myös ominaisuus lisätä veroprosentti ja pakolliset verot kuten työeläkemaksu ja työttömyysvakuusmaksut, jotta palkan asettaminen olisi mahdollisimman realistinen. Ikä on huomioitu, mikäli käyttäjä on asettanut sen asetukissa.

Sovelluksen toiminnalliset vaatimukset ovat seuraavanlaiset: Käyttäjän on pystyttävä helposti asettamaan budjettinsa, tulonsa ja menonsa. Tämän lisäksi on tärkeää, että sovellus laskee näiden väliset laskut huolellisesti ja ilman virheitä. Käyttäjällä pitää olla myös oikeus seurata menojaan erilaisilla graafeilla ja kaavioilla.

Sovelluksen laadulliset vaatimukset ovat sen helppokäyttöisyys ja käyttäjäystävällisyys, turvallisuus, tehokkuus. Laskujen on myös oltava luotettavia ja tarkasti laskettuja.

3 Sovelluksen toiminnallisuus



Kuva 1. Käyttötapauskaavio sovelluksen käyttötapauksista.

Sovelluksessa on kaksi käyttäjäroolia: tavallinen käyttäjä ja ylläpitäjä. Tavallinen käyttäjä voi kirjautua sisään, asettaa budjetin ja lisätä tuloja ja menoja. Ylläpitäjä on vastuussa sovelluksen ylläpidosta ja hallinnasta. Hänellä ei toistaiseksi ole erillisiä toimintoja.

3.1 Tietokanta

Sovelluksen tietokanta on SQL-pohjainen. Tietomalli koostuu viidestä taulusta:

UserInfo- tauluun tallentuu käyttäjien rekisteröintitiedot, mukaan lukien heidän rekisteröintitunnuksensa, käyttäjätunnuksensa, salasanssa, ensisijaisen valuutan ja teeman värin.

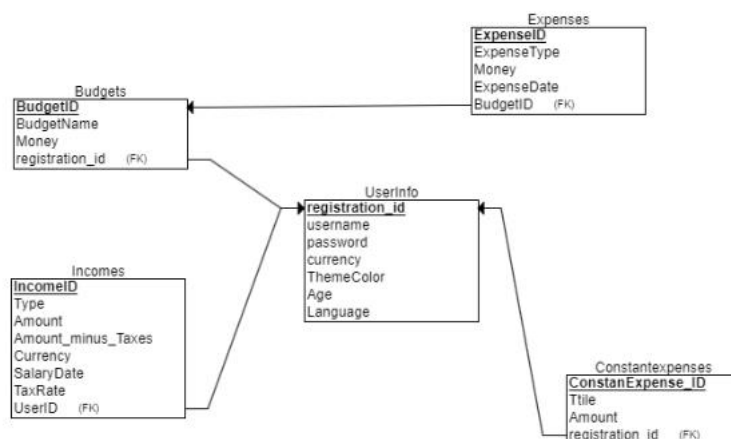
Incomes- tauluun tallentuu käyttäjien tulotiedot, mukaan lukien tulotunnus, tulon tyyppi (kuukausi tai päivä), summa, summa verojen jälkeen, valuutta, palkka-päivä, verokanta ja tulon saaneen henkilön käyttäjätunnus.

Constantexpenses- tauluun tallentuu tiedot käyttäjien pysyvistä kuluista, mukaan lukien kulun aiheuttaneen käyttäjän tunnus, kulun otsikko, summa ja rekisteröintitunnus.

Budgets- tauluun tallentuu käyttäjien budjettitiedot, mukaan lukien budjetin tunnuksen, budjetin nimen, varatun rahan ja budjetin luoneen käyttäjän rekisteröintitunnuksen.

Expense- tauluun tallentuu käyttäjien kulutiedot, mukaan lukien kulutunnuksen, kulutyyppin, käytetyn rahan määrän, kulupäivämäärän ja sen budjetin budjetin, johon kulu on liitetty.

Tämän tietomallin avulla käyttäjät voivat seurata tulojaan, menojaan ja budjettejaan samalla kun tietonsa pysyvät turvassa rekisteröintijärjestelmän avulla. Mallin avulla kulut on helppo yhdistää budjetteihin ja tulot käyttäjille.



Kuva 2. Relaatiotietokanta malli

3.2 Käyttöliittymä

Käyttöliittymä on pyritty pitämään mahdollisimman helppokäyttöisenä. Sovellukseen kirjautuminen ja rekisteröityminen on tehty helpoksi ja samanlaiseksi kuin monessa muussa sovelluksessa. Kirjautumisen jälkeen käyttäjä pääsee heti pääsivulle. Pääsivulla huomaa heti Budjettien yhteissumman mitä on jäljellä, rahansäästö vinkit, jotka vaihtuvat 30 sekunnin välein sekä budjetteihin, menoihin, tuloihin ja asetuksiin johtavat selkeät napit. Alanurkassa on mahdollisuus myös kirjautua ulos.

Budjettinäköymässä huomaa heti aktiivisen budjetin ja sitä voi helposti muokata, tai sen voi halutessaan poistaa kokonaan. Sen voi myös vaihtaa valikosta. Siellä pystyy myös asettamaan jatkuvat menot, kuten vuokran ja vesilaskun. Ruudun alareunassa näkyy pylväsdiagrammi kyseisen budjetin menoista, mutta sitä painamalla se vaihtuu piirakkadiagrammiksi.

Menot-näköymässä on helppo lukea aktiivinen budjetti sekä asettaa meno siihen budjettiin asettamalla menolle määrän, kategorian ja päivämäärän. Budjetista voi myös vähentää jatkuvat menot tästä näköymästä. Näköymän alakulmasta voi nähdä listana listatut menot ja jatkuvat menot mitkä ovat lähteneet budjetista.

Tulot-näkymässä on mahdollista asettaa joko kuukausipalkka tai tuntipalkka. Kuukausipalkkaan asetetaan palkka, veroprosentti ja päivämäärä. Tuntipalkkaan asetetaan tuntipalkka, tuntienmäärä, veroprosentti ja päivämäärä. Molemmissa näkymissä asetetut palkat tulevat listaan näkyviin näkymän alareunaan ja sieltä on myös mahdollista valita kuukausi, josta sovellus kertoo paljonko palkkaa sinä kuukautena, on asetettu.

Asetus-näkymässä käyttäjä voi vaihtaa teemaa, valuuttaa, kieltä ja asettaa itselleen iän. Asetuksissa on myös nappi mistä voi poistaa kaikki käyttäjän sisältämän datan. Asetuksissa asetetut muutokset tallennetaan käyttäjään, joten aina kun käyttäjä kirjautuu nii asetukset astuvat voimaan.

3.3 Budjetti

Budjetteihin tallennetaan lista siihen kuuluvista menoista, siihen asetettu rahamäärä, nimi ja generoitu ID. Sisään kirjautuneen käyttäjän budjetteja ylläpitää Variables-luokka, johon tallennetaan käyttäjän budjetit listana ja muuttujana käyttäjän valitsema aktiivinen budjetti. Käyttäjä voi luoda uusia budjetteja, muokata budjettien rahamääriä ja nimiä, poistaa budjetteja, sekä lisätä menoja aktiiviseen budjettiin.

3.4 Menot

Menot ovat budjetteihin luotuja rahankäyttökertoja. Menoihin tallennetaan käytetty rahamäärä, kategoria, päivämäärä ja generoitu ID. Kaikki menot kuuluvat johonkin budjettiin ja luodut menot menevät aina luontihetkellä olevaan aktiiviseen budjettiin.

3.5 Tulot

Tuloihin tallennetaan palkka, veroprosentti, tyyppi (tunti- tai kuukausipalkka), päivämäärä ja generoitu ID. Sisään kirjautuneen käyttäjän tuloja ylläpitää SalarySingle-luokka. SalarySingle-luokassa lasketaan tulot ja lisätään ne kuukausipalkka ja tuntipalkka listoihin.

3.6 Lokalisointi

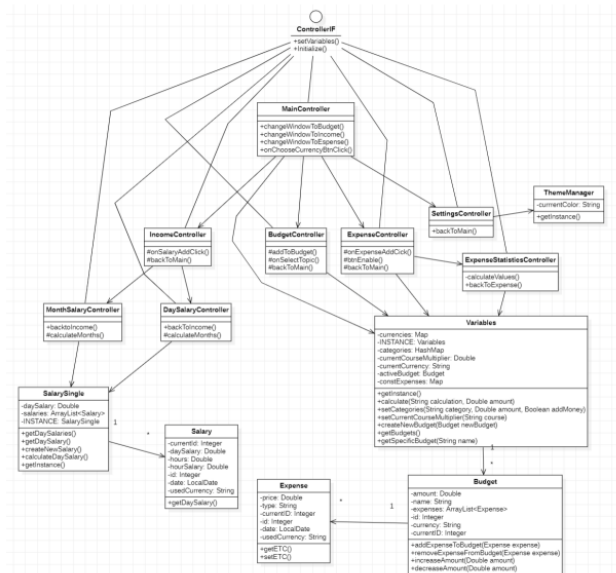
Sovellus on lokalisoitu resource bundle -tiedostojen avulla. Sovelluksessa on toistaiseksi 3 kielivaihtoehtoa, jotka ovat Suomi, Englanti ja Islanti, mutta niiden lisääminen on tehty kehittäjille helpoksi. Sovelluksessa kaikki tekstit vaihtavat kieltä, sekä myös mahdolliset sovelluksen sisäiset virheviestit näkyvät erikielellä.

4 Ohjelmointitekkinen toteutus

4.1 Käytetyt ohjelmointikielet ja kirjastot

Ohjelmisto on kokonaan kirjoitettu Java-ohjelmointikielellä. Vanhin kehitysvaiheessa käytetty version on ollut JDK 16. Käyttöliittymään on käytetty erilaisia JavaFX kirjastoja. Testaamiseen JUnit-5 kirjastoa. Tietokannan yhteydenluontiin on käytetty mysqlConnector kirjastoa. Valuutta API:iin on hyödynnetty JSON kirjastoa.

4.2 Arkkitehtuuri



Kuva 3. Projektin luokkakaavio

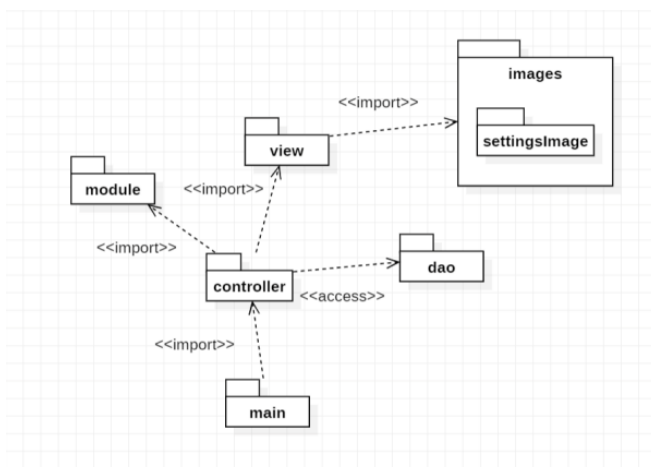
Projekti noudattaa siis MVC arkkitehtuuria. Kontrollerit ohjaavat tietoa visuaalisesta puolesta modeliin, eli itse pääkoodiin. Modelissa osiot ovat jaettu luokiksi eli erilaiset oliot tallennetaan Variables singleton luokkaan, jonka kautta kontrollerit saavat tarvittavat tiedot näkyville ja asettavat tarvittavat muuttujat.

4.3 Käyttöliittymä

Käyttöliittymä on toteutettu JavaFX:llä. Kaikki näkymät ovat tehty Scenebuilderilla FXML tiedostoja muuttaen. Jokaisella näkymällä on oma kontrolleri, joten näkymien vaihtamisen yhteydessä vaihtuu myös käytössä oleva kontrolleri.

4.4 Sisäinen logiikka

Sovellus on rakennettu MVC mallilla, joten kaikki käyttäjän syöttämät tiedot kulkeutuvat kontrollerin kautta model-tasolle. Meidän sovelluksessamme modeltasolla on Variables singleton luokka, joka huolehtii muuttujien arvoista ja sisältää tarvittavat budjetti, menot ja jatkuvat menot -oliot. Tuloihin liittyvät oliot ovat erillään SalarySingleton -luokassa, sillä ne ei tarvitse yhteyttä budjettiin liittyvien olioiden tietoihin. Tietokantaan yhteys muodostetaan kontrollereiden ja daojen kautta, sillä kontrollereista lähetetään daoille tarvittavat tiedot tietokannan muokkaamiseen.



Kuva 4. Projektin pakkauskaavio

Pakkauskaaviosta huomaa myös vahvasti MVC arkkitehtuurin, mutta tässä myös huomaa sen, kuinka tietokannan data kulkee sovelluksesta ja sovellukseen. Sovelluksen kirjautuessa tietokannasta haetaan kyseisen käyttäjän

tarvittavat tiedot. Sovelluksessa käytetään myös kuvia, kuten logoa, jotka löytyvät images pakkauksesta.

4.5 Testaus

Testaus toteutettiin JUnit-5 kirjastolla. Testit testasivat sovelluksen sisäisiä toiminnallisuuksia, kuten palkkaa, menoja ja erilaisia muuttujien muutoksia ja laskuja. Testit ajettiin myös Jenkinsin kautta Metropolian EduCloud palvelimella. Jenkins piti huolen, että testit menevät läpi jokaisen julkaisun jälkeen.

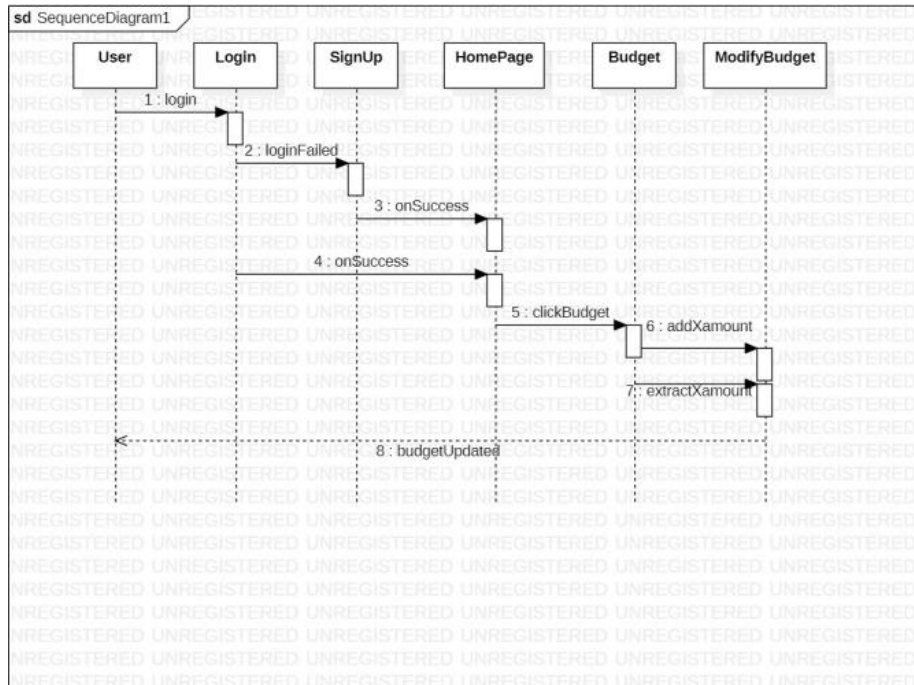
Sovelluksen käyttöliittymän ja tietokannan toimintaa ja niiden yhteyttä testasimme manuaalisesti jokainen erikseen. Testien kirjoittaminen niistä olisi vienyt liikaa resurssejamme, joten päätimme jättää ne tekemättä ja kokeilla kaiken manuaalisesti toimintoja koodatessamme.

4.6 Kehitysprosessi

Sovelluksen toteutus jaettiin kolmeen sprinttiin, joiden kesto oli yhteensä kahdeksan viikkoa. Projektin aluksi kirjoitettiin käyttäjätarinoita, jotka jaettiin tärkeysjärjestyksen mukaan eri sprinteille. Sprinteille suunnitellut käyttäjätarinat tuli suorittaa kyseisen sprintin aikana. Käyttäjätarinat jaettiin pienempiin tehtäviin, joille jaettiin tehtävästä vastuussa olevat henkilöt. Tässä apuna käytimme Nektion-nettisovellusta. Vähintään viikoittain pidimme palaverin, jossa käytiin läpi projektin etenemistä. Versionhallinnassa käytimme Githubia, jossa jokaisella oli omat branchit ja jokainen kehitti omaa tahtia omaan branchiin ominaisuuksia. Kun toiminnallisuus oli kehitetty, se puskettiin mainiin pull requestin ja muiden antaman arvostelun avulla.

5 Ohjelmiston toiminta

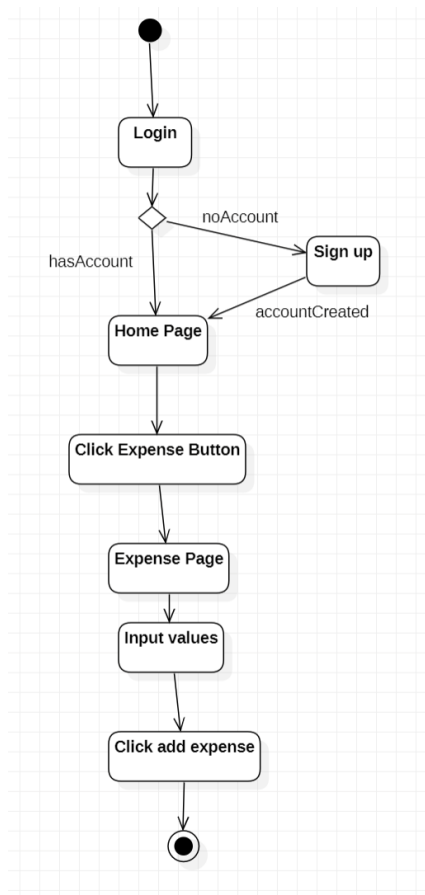
Käyttötapaus 1. Budjetin muokkaus.



Kuva 5. Sekvenssikaavio budjetin muokkauksesta sovelluksessa.

Aluksi käyttäjä kirjautuu sisään tai luo itselleen uuden käyttäjätilin ja siitä sitten siirtyy sovelluksen kotinäkymään. Kotinäkymässä käyttäjä painaa Budget-nappia ja siirtyy Budjetti-näkymään. Budjetti-näkymässä käyttäjä valitsee haluamansa budjetin, syöttää muutokset tekstikenttiin ja painaa Modify-nappia.

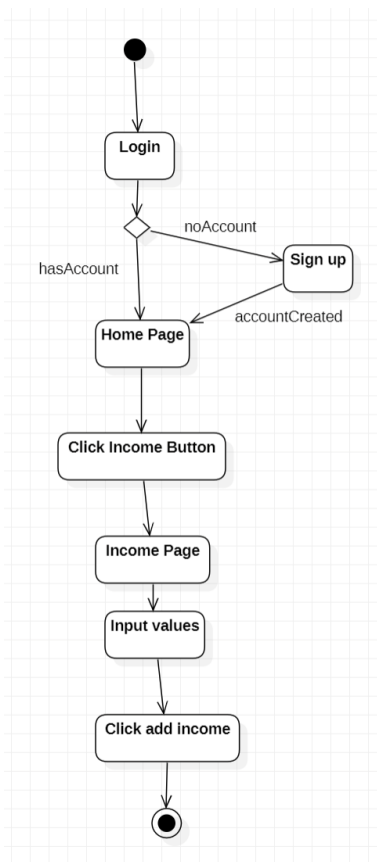
Käyttötapaus 2. Menon lisääminen.



Kuva 6. Aktiviteettikaavio menon lisäämisestä sovelluksessa.

Ensin käyttäjän on kirjauduttava sisään tai luotava uusi käyttäjätili. Kun on valmista, niin käyttäjä siirtyy kotinäkömään, jossa hän painaa Expense-nappia ja siirtyy menonäkymään. menonäkymässä käyttäjä syöttää menon rahamäärän ja valitsee menon tyypin (esim. vaatteet, ruokaostokset tai harrastukset), sekä joko valitsee tai on valitsematta menon päivämäärää (jos jättää valitsematta, niin päivämääräksi asettuu sen hetkinen päivämäärä). Sitten kun tarvittavat syötteet ollaan annettu, niin käyttäjä painaa Add-nappia.

Käyttötapaus 3. Kuukausitulon lisäys.



Kuva 7. Aktiviteettikaavio tulon lisäämisestä sovelluksessa.

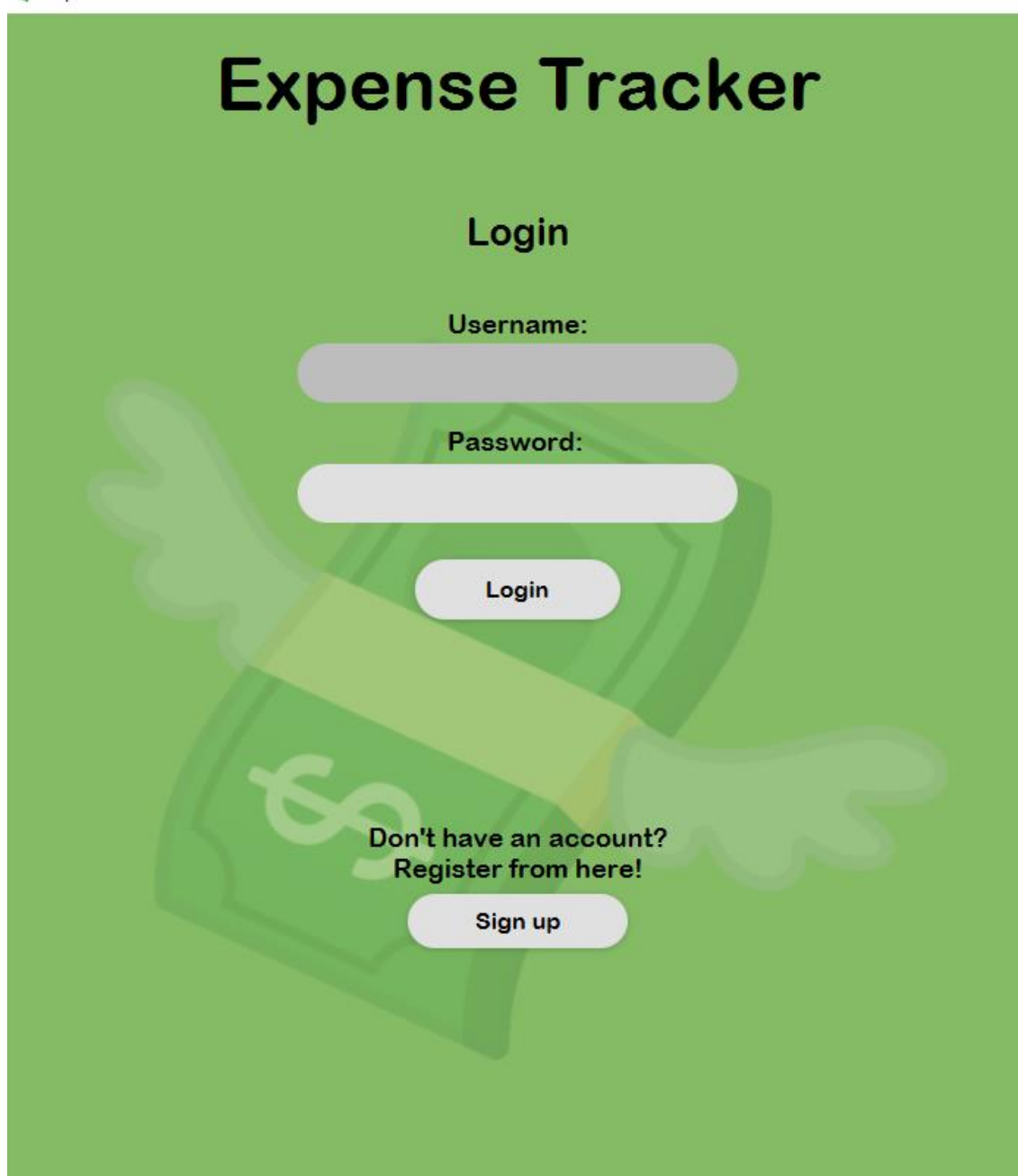
Käyttäjän on aluksi kirjauduttava sisään tai luotava uusi käyttäjätili ja kun on jommankumman tehnyt, niin hän siirtyy kotinäkymään. Kotinäkymässä käyttäjä painaa Income-nappia ja siirtyy tulonäkymään. Tulonäkymässä käyttäjä syöttää kuukausipalkkansa, veroprosenttinsa ja joko valitsee päivämäärän tai on valitsematta (jos jättää valitsematta, niin päivämääräksi asettuu sen hetkinen päivämäärä). Käyttäjä voi myös valita laskuihin mukaan pakolliset verot painamalla pakolliset verot-valintaruutua. Syötettyään tarvittavat tiedot, käyttäjä painaa Add-nappia.

6 Sovelluksen käyttö

Sovelluksen käyttö alkaa kirjautumisnäköymästä, jossa käyttäjä kirjautuu sisään syöttämällä käyttäjänimensä ja salasanan. Jos käyttäjällä ei ole vielä käyttäjätiliä sovellukseen, hän voi painaa Sign up -nappia, josta hän siirtyy käyttäjätilin luomiseen. Käyttäjätilin luomisessa käyttäjä luo uuden tilin syöttämällä uuden käyttäjänimen ja salasanan. Kun käyttäjä on kirjautunut sisään tai luonut käyttäjätilin, hän siirtyy sovelluksen kotinäköymään. Sovellus on vakioasetuksena englanniksi. Eli uusien käyttäjien kieli on aina aluksi englanti ja valuutta euro.

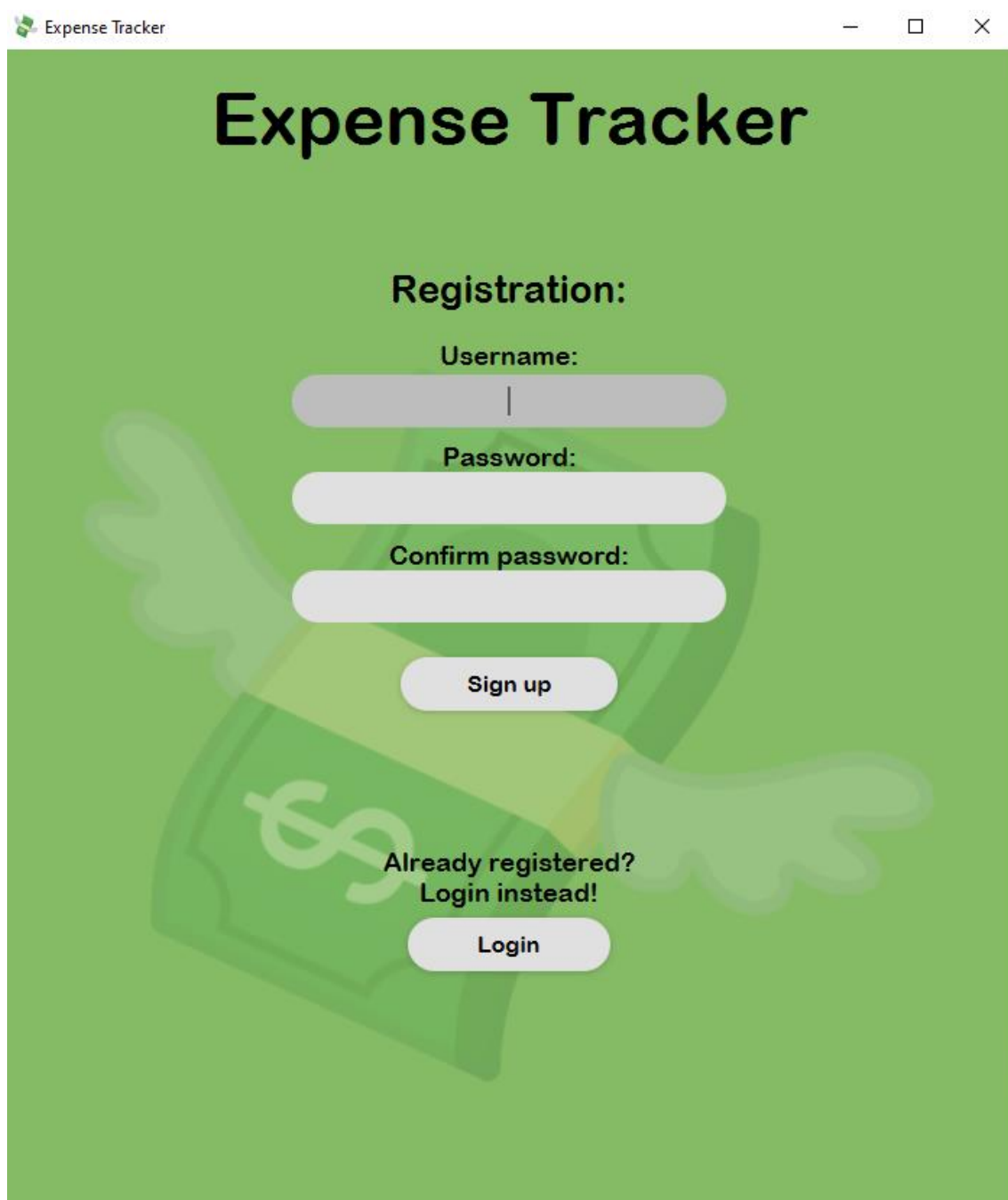
Expense Tracker

— □ ×



The image shows a web application interface for an "Expense Tracker". The background is a solid green color. At the top, the title "Expense Tracker" is displayed in a large, bold, black font. Below the title, the word "Login" is centered in a bold, black font. Underneath "Login", there are two input fields: the first is labeled "Username:" and the second is labeled "Password:". Both labels are in a bold, black font. Below the password field is a rounded rectangular button with the text "Login" in a bold, black font. At the bottom of the login section, there is a link that says "Don't have an account? Register from here!" in a bold, black font. Below this link is another rounded rectangular button with the text "Sign up" in a bold, black font. In the background, there is a faint, stylized illustration of a hand holding a stack of money, with a large dollar sign (\$) visible.

Kuva 8. Kirjautuminen.

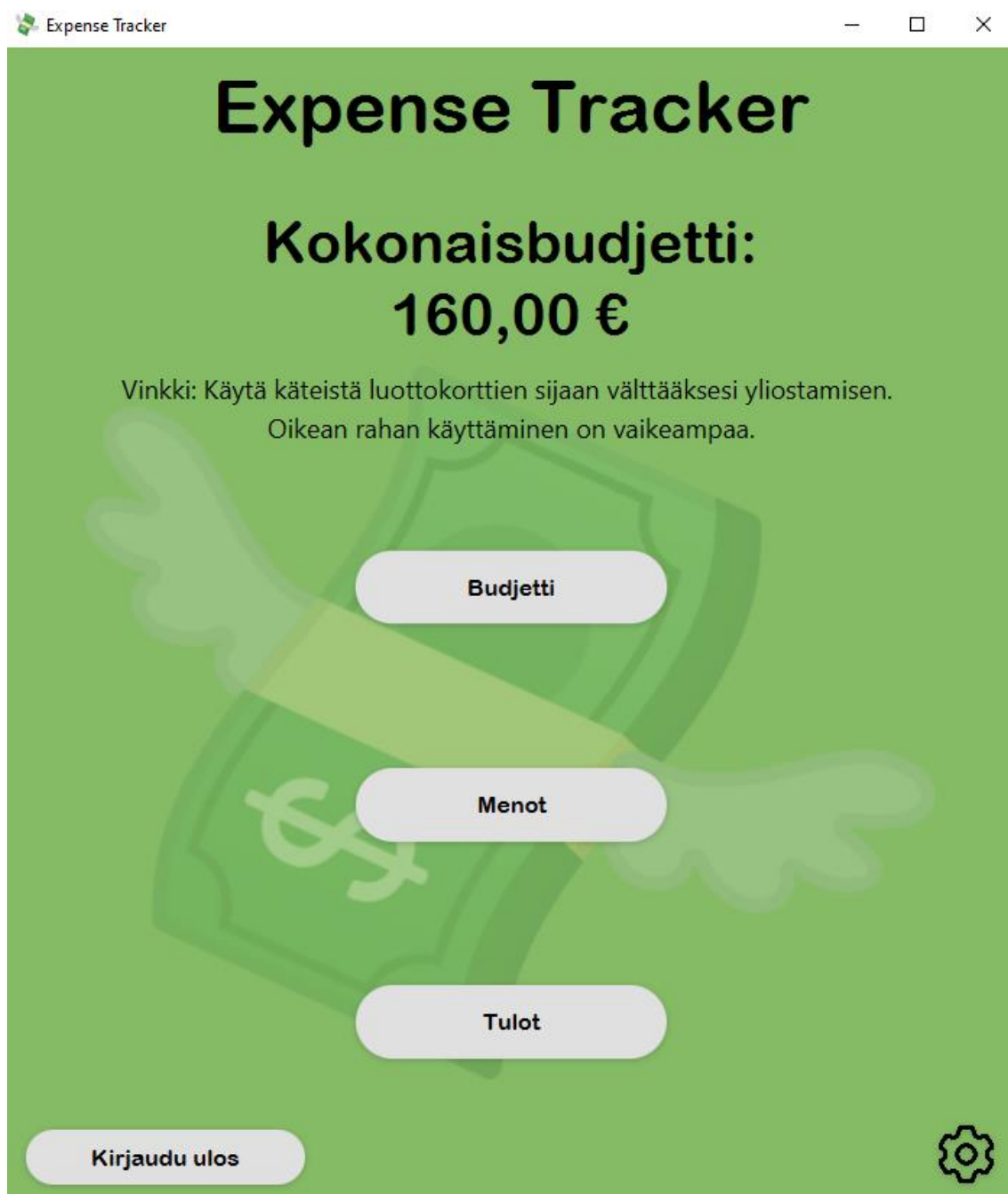


The image shows a web browser window titled "Expense Tracker". The background is a solid green color with a faint, stylized illustration of a hand holding a Euro banknote. The main heading "Expense Tracker" is centered at the top in a large, bold, black font. Below the heading, the section "Registration:" is centered. It contains three input fields: "Username:" (a light gray rounded rectangle with a vertical cursor), "Password:" (a white rounded rectangle), and "Confirm password:" (a white rounded rectangle). Below these fields is a white rounded button labeled "Sign up". Further down, the text "Already registered? Login instead!" is centered, followed by a white rounded button labeled "Login". The browser window has a standard title bar with a minus sign, a square icon, and a close 'X' icon.

Kuva 9. Rekisteröityminen.

Kotinäkymässä käyttäjä voi navigoida sovelluksen eri toimintoihin. Budjetti-toimintoihin pääsee Budjetti -napista, menotoimintoihin pääsee Menot -napista, tulotoimintoihin pääsee Tulot -napista ja asetuksiin pääsee hammasratasnapista

kotinäkymän oikeassa alakulmassa. Kotinäkylässä käyttäjä voi myös kirjautua ulos Kirjaudu ulos -napilla kotinäkylässä vasemmassa alakulmassa.



Kuva 10. Kotinäkymä.

Budjettinäkylässä käyttäjä voi luoda uusia budjetteja valitsemalla pudotusvalikosta "New" ja syöttämällä budjetille nimen ja määrän ja lopuksi painamalla Lisää -nappia. Kun budjetti on luotu, budjetin voi valita aktiiviseksi valitsemalla

sen pudotusvalikosta. Kun budjetti on valittuna, tehdyt menotiedot liittyvät siihen valittuun budjettiin. Myös kun budjetti on valittu, pystyy siihen tiettyyn budjettiin asettamaan jatkuvia menoja viereisestä valikosta. Budjettinäkömän alaosassa on tilasto valitun budjetin menoista ja sitä klikkailemalla voi valita katsooko mieluummin pylväskuviona vai piiraskuviona. Takaisin kotinäkömään pääsee Takaisin -napista.



Kuva 11. Budjettinäkömä.

Menonäkymään pääsee, kun on valittuna budjetti. Siellä käyttäjä pystyy lisäämään menoja valittuun budjettiin, sekä lisäämään manuaalisesti tiettyjä juoksevia menoja, tämän automatisointi on suuri prioriteetti jatkokehityksessä. Käyttäjä voi myös selata valitun budjetin menoja ja halutessaan poistaa niitä klikkaamalla haluamaansa menoja Menojen historia -listassa. Menonäkymästä pääsee myös tilastonäkymään, jossa näkyy valitun budjetin menojen tilastoja diagrammeina. Menonäkymästä pääsee takaisin kotinäkömään Takaisin -napilla.

Expense Tracker

Takaisin

Menot

Aktiivinen budjetti:
moi: 40,00 €

Lisää yksittäinen meno

Määrä:

Kategoria

Kiinteät menot

Poista budjetista

Lisää

Menojen historia

Harrastukset, lisätty: 2023-05-03, rahaa käytetty: 10,00 €

Harrastukset, lisätty: 2023-05-03, rahaa käytetty: 50,00 €

Kuva 12. Menonäkymä.

Tulonäkymässä käyttäjä voi laskea kuukausi- tai tuntipalkkojansa, ja tämän voi valita näkymän oikeasta yläkulmasta. Kuukausipalkan laskussa syötetään kuukausipalkan määrä, veroprosentti ja päivämäärä (tyhjä päivämäärä on automaattisesti sen hetkinen päivämäärä). Käyttäjä voi myös valita laskuun mukaan pakolliset verot klikkaamalla veroprosentin vieressä olevaa valintaruutua. Tuntipalkan lasku on muuten sama, mutta siinä syötetään tuntipalkka ja tehdyt tunnit. Palkkahistoria-listassa näkyy tehdyt palkanlaskennat (kuukausitilassa kuukausipalkat ja tunti-tilassa tuntipalkat) ja listan alta voi nähdä kokonaislaskennat eri-kuukausilta.

Expense Tracker

Takaisin

Tulot

Tuntipalkka

Lisää kuukausipalkka

€

Lisää veroaste

%

Valitse palkkapäivä

Lisää

Palkkahistoria

Tarkista kuukauden palkka:

Kuukaudet ▼

2023-05-03 Palkan määrä miinus verot 40,00% on 3000,00 €

Kuva 13. Tulonäkymä, tuntipalkanlaskenta ja palkkahistoria tuntipalkkoille.

Expense Tracker

Takaisin

Tulot

Kuukausipalkka

Lisää tuntipalkka

€

Lisää tunnit

Lisää veroaste

%

Valitse työpäivä

Lisää

Palkkahistoria

Tarkista kuukauden palkka:

Kuukaudet ▼

2023-05-03 Palkan määrä miinus verot 47,15% on 5285,00 €

Kuva 14. Tulonäkymä, kuukausipalkanlaskenta ja palkkahistoria kuukausipalkkoille.

Asetuksissa käyttäjä voi vaihtaa sovelluksen taustaväriä, valuuttaa ja sovelluksen kieltä. Käyttäjä voi myös asettaa ikänsä, joka vaikuttaa pakollisten verojen laskutapaan. Tämä ominaisuus toimii vain, jos kieleksi on valittu suomi, koska muissa maissa tätä vaikutusta ei aina ole edes olemassa. Käyttäjä voi tyhjentää käyttäjätilinsä tiedot painamalla Delete ALL data-nappia, joka löytyy asetusnäytön alimmasta napista.

Expense Tracker

Takaisin

Asetukset

Taustaväri:

Original

Valuutta:

EUR

Kieli:

Finnish

Aseta ikä:
(Vain Suomessa käytössä)

Aseta

Poista KAIKKI tiedot

Kuva 15. Asetukset.

7 Sovelluksen jatkokehitys

Sovelluksen ulkonäköä voisi yhä työstää paremmaksi, OTP2 kurssin aikana saimme siitä paljon mukavamman näköisen, mutta sen työstäminen on ikuinen projekti. Teemaväriä vaihtaessa voisi vaihtaa tekstin väriä kontrastia varten. Voisimme koittaa saada jaetun budjettiominaisuuden implementoitua niin, että se olisi yksinkertainen ja helppokäyttöinen. Jos tulee tarve lisätä erillinen sovelluksen käyttäjätyyppi järjestelmänvalvojille, esimerkiksi käyttäjien poistamista varten, niin sen voisi kehittää.

Sovelluksen testausta voisi vielä kehittää niin, että saataisiin mahdollisimman monet asiat sovelluksesta testattua. Lokalisointia voisi viedä pidemmälle niin, että se kattaisi myös esimerkiksi eri maiden pakollisia veroja ja muuttaisi tarvittaessa sovelluksen ulkonäköä (esim. kieltä luetaan oikealta vasemmalle tai kielen fontti ei sovi sovelluksen käyttöliittymään ilman muutoksia).

Hieman haastavampia kehityksiä voisivat olla esimerkiksi juoksevien menojen automatisointi niin, että tietyin aikaväleihin asetetut juoksevat menot automaattisesti vähennetään budjetista, ja mikäli luottokortin yhdistämisen sovellukseen olisi mahdollista, voisi tehdä automatisoidun menojen lisäämisen käyttäjän luottokorttiososten mukaan.

8 Yhteenveto

Projektimme on työpöytäsovellus nimeltään Expense Tracker, joka auttaa käyttäjiä tarkkailemaan heidän rahankäyttöänsä. Sovelluksen käyttöönotto on melko helppoa, tarvitset sovelluksen Github-repositorion koodin ja Metropolia tunnukset, jotta pääset käyttämään Metropolia VPN:ää, jotta sovellus toimii. Tämän lisäksi tarvitset vielä laittaa hieman konfiguraatioita, jotka kerrotaan tarkemmin Expense Tracker -repositoriossa sekä tässä dokumentissa. Dokumentissa myös kerrotaan lyhyesti sanallisesti ja kuvallisesti, miten sovellusta käytetään.

Molempien ohjelmistoprojekti kurssien lopuksi saimme mielestämme valmiin tuotteen, johon voimme olla tyytyväisiä. Kurssien aikana syvensimme osaamistaitojamme Java kielessä, ryhmätyöskentelemisessä projektin kimpussa, sekä opimme myös paljon uusia asioita erilaisista kehitysmenetelmistä, kuten esimerkiksi ketterästä kehitysmenetelmästä. Meidän loistavan dokumentoinnin takia uskomme, että sovelluksen jatkokehittäminen ei ole kovin monimutkaista.