

Rahat raiteille Kulunseurantasovellus

Vaatimusmäärittely

Mikael Vihelä

Fei Raita

Lumi Sainomaa

Leevi Hamari

Sisällysluettelo

1. Johdanto	3
1.1 Tarkoitus ja kattavuus.....	3
1.2 Tuote ja ympäristö	3
1.3 Määritelmät, termit ja lyhenteet.....	3
1.4 Viitteet	4
1.5 Yleiskatsaus dokumenttiin	4
2. Yleiskuvaus.....	5
2.1 Ympäristö	5
2.2 Toiminta	5
2.3 Käyttäjät.....	5
2.4 Yleiset rajoitteet.....	5
2.5 Oletukset ja riippuvuudet	5
3. Tiedot ja tietokanta	5
3.1 Tietosisältö	6
3.2 Käyttöintensiteetti	6
3.3 Kapasiteettivaatimukset	6
3.4 Tiedostot ja asetustiedostot	6
4. Toiminnot.....	6
4.1 Etusivu.....	7
4.2 Lisää kategoria	8
4.3 Lisää kirjaus	8
5. Ulkoiset liittymät.....	9
6. Muut ominaisuudet	9
6.1 Suorituskyky ja vasteajat	9
6.2 Saavutettavuus (availability), toipuminen, turvallisuus, suojaukset	9
6.3 Ylläpidettävyys	10
6.4 Siirrettävyys ja yhteensopivuus	10
6.5 Operointi	10
6.6 Käytettävyys (Usability), käytön tehokkuus, käyttäjien tyytyväisyys	10
7. Suunnittelurajoitteet	10
7.1 Standardit	10
7.2 Laitteistorajoitteet	10

7.3 Ohjelmistorajoitteet	11
7.4 Muut rajoitteet	11
8. Hylätyt ratkaisuvaihtoehdot	12
9. Jatkokehitysajatuksia	12
Liitteet.....	12

1. Johdanto

Tässä osiossa kuvataan RahatRaitelle sovelluksen keskeisiä ominaisuuksia ja käyttötarkoituksia.

1.1 Tarkoitus ja kattavuus

RahatRaiteille sovelluksen tarkoitus on seurata käyttäjän taloudellisia menoja ja auttamaan säästämään menoissa. RahatRaiteille käyttäjät kirjaavat menonsa, jonka avulla käyttäjät näkevät rahan kulutuksen. Sovellus seuraa aikaisempia merkintöjä ja sen avulla kertoo käyttäjälle missä ollaan säästetty ja missä ollaan tuhlatu enempi viime kuluihin verrattuna. Sovellus on kehitetty, jotta käyttäjä pystyy katsomaan kaikki menonsa yhdestä samasta paikasta.

1.2 Tuote ja ympäristö

RahatRaiteilla on perusarjessa auttava tilinseuranta sovellus, josta näkee kaikki omat taloudelliset menot ja tulot yhdestä paikasta näppärästi. Sovellus on tarkoitettu käytettävän missä ja milloin vaan, Windows pohjasella laitteella.

1.3 Määritelmät, termit ja lyhenteet

Alla on lyhyesti selitettynä määritelmiä, termejä ja lyhenteitä, joita sovelluksessa käytetään. Näiden ymmärtäminen on käyttäjäkokemuksen kannalta tärkeää.

Lisää kategoria = Käyttäjä lisää oman haluamansa kategorian, nimeten sen haluamallaan tavalla.

Lisää kirjaus = Käyttäkää kirjaa oman menonsa, nimeten sen, valitsemalla kategorian, värin ja määrittämällä hinnan.

Väri = Käyttäjä voi määrittää kategorialle oman värin.

Poista = Käyttäjä voi poistaa luodun kategorian/kirjauksen listalta.

1.4 Viitteet

Dokumentin sisällön tukena käytettiin seuraavia lähteitä.

Moodle Hamk (n.d.) Vaatimusmäärittely esimerkki.

https://learn.hamk.fi/pluginfile.php/1255979/mod_forum/intro/Vaatimusm%C3%A4%C3%A4rittely%20-%20esimerkki.pdf?time=1705927085914#page=22&zoom=100,109,76

Stack Overflow (n.d.) Mitä tähän laitetaan. Haettu 2.2.2024.

<https://stackoverflow.com/questions>

Microsoft. (n.d.) Microsoft Learn. Haettu 12.2.2024.

<https://learn.microsoft.com/en-us/>

Ann Vega. (27.3.2023). Retrieving values from a key-value pair using a List in C#. Haettu 28.2.2024.

https://copyprogramming.com/howto/getting-sum-from-key-value-pair-using-list-in-c#google_vignette

DeDub.to. (27.11.2023) How to delete item from JsonFile using C#. Haettu 27.2.2024.

<https://debug.to/6461/how-to-delete-item-from-jsonfile-using-c>

1.5 Yleiskatsaus dokumenttiin

Tämä dokumentti kuvaa RahatRaiteille-nimisen rahankulutuksen seurantasovelluksen yleisiä ominaisuuksia, toimintaa ja käyttöympäristöä, ja se on suunnattu RahatRaiteille sovelluksen käyttäjille ja kehittäjille, jotka tarvitsevat tietoa sovelluksen toiminnasta ja käytöstä. Dokumentti sisältää kuvauksen sovelluksen käyttöympäristöstä ja toiminnasta, tietojen ja tietokannan käsittelystä, sovelluksen toiminnoista, ulkoisista liittymistä sekä muista ominaisuuksista, suunnittelurajoitteista, hylätyistä ratkaisuvaihtoehdoista ja jatkokehitysideoista.

2. Yleiskuvaus

Tässä osiossa kuvataan sovelluksen käyttöympäristöä, toimintaa ja käyttäjää, sekä yleiset rajoitteet ja riippuvuudet.

2.1 Ympäristö

Sovelluksen käyttöympäristönä toimii sovellus, johon käyttäjä voi omien tietojensa perusteella kirjata kulujaan. Tavoitteena on antaa käyttäjälle yksinkertainen ja toimiva tapa seurata arkisia rahankulutuksen menoja ja tarjota samalla säästövinkkejä.

2.2 Toiminta

Käyttäjän ensimmäisellä sivulla on diagrammi, josta näkee kulutuksen seurantaan helposti ja nopeasti. Etusivulta käyttäjällä on kolme vaihtoehtoa: lisää kirjaus, luo kategoria ja seuraa kulutusta. Lisää kategoria-sivulla käyttäjä luo eri rahankulutus kohteita, joihin lisätään esimerkiksi "Laskut" tai "Kauppa". Kategorioihin kirjataan kulunut rahamäärä. Etusivun diagrammista pystyy seuraamaan kulutusta kirjattujen menojen perusteella. Tarkastele menoja sivulla on kirjattuna kaikki käyttäjän tekemät kirjaukset.

2.3 Käyttäjät

Sovellus on tarkoitettu jokaiselle, joka haluaa seurata oman rahansa kulutusta. Ikärajoitusta ei ole ja käyttöliittymä on yksinkertainen, eli soveltuu jokaiselle.

2.4 Yleiset rajoitteet

Tässä osiossa kuvataan kaikki tuotteen suorituskriteerit ja rajoitteet mm. mitä se vaatii toimiakseen tai mihin se ei pysty.

2.5 Oletukset ja riippuvuudet

Käyttäjän oletetaan tietävän omat tulonsa ja menonsa. Ilman näitä tietoja kulutuksen seuranta on mahdotonta.

Sovelluksen maksimaalisen hyödyn perusteella käyttäjän tulisi kirjata aktiivisesti rahan kulutustaan.

3. Tiedot ja tietokanta

Tässä osiossa käsitellään RahatRaiteille sovelluksen tiedonhallintaa ja tietokantaan liittyviä ominaisuuksia.

3.1 Tietosisältö

Käyttäjän omat luodut kategoriat ja tulot/menot tallenetaan json muodossa. Tallenteita käytetään, jotta käyttäjän ei tarvitse joka kerta kirjata tuloja/menoja ja omia luotuja kategorioita uudestaan.

3.2 Käyttöintensiteetti

Käyttäjä syöttää kuukausi tulonsa ja menonsa ja sovellus näyttää mihin kaikkialle rahaa on mennyt. Sovelluksessa voi verrata eri kuukausien menoja ja tuloja. Sovellus kertoo, missä on säästetty tai tuhlatu aikaisempiin kuukausiin verrattuna. Sovelluksessa on pop-up toiminto, joka kertoo erilaisia säästövinkkejä käyttäjille.

3.3 Kapasiteettivaatimukset

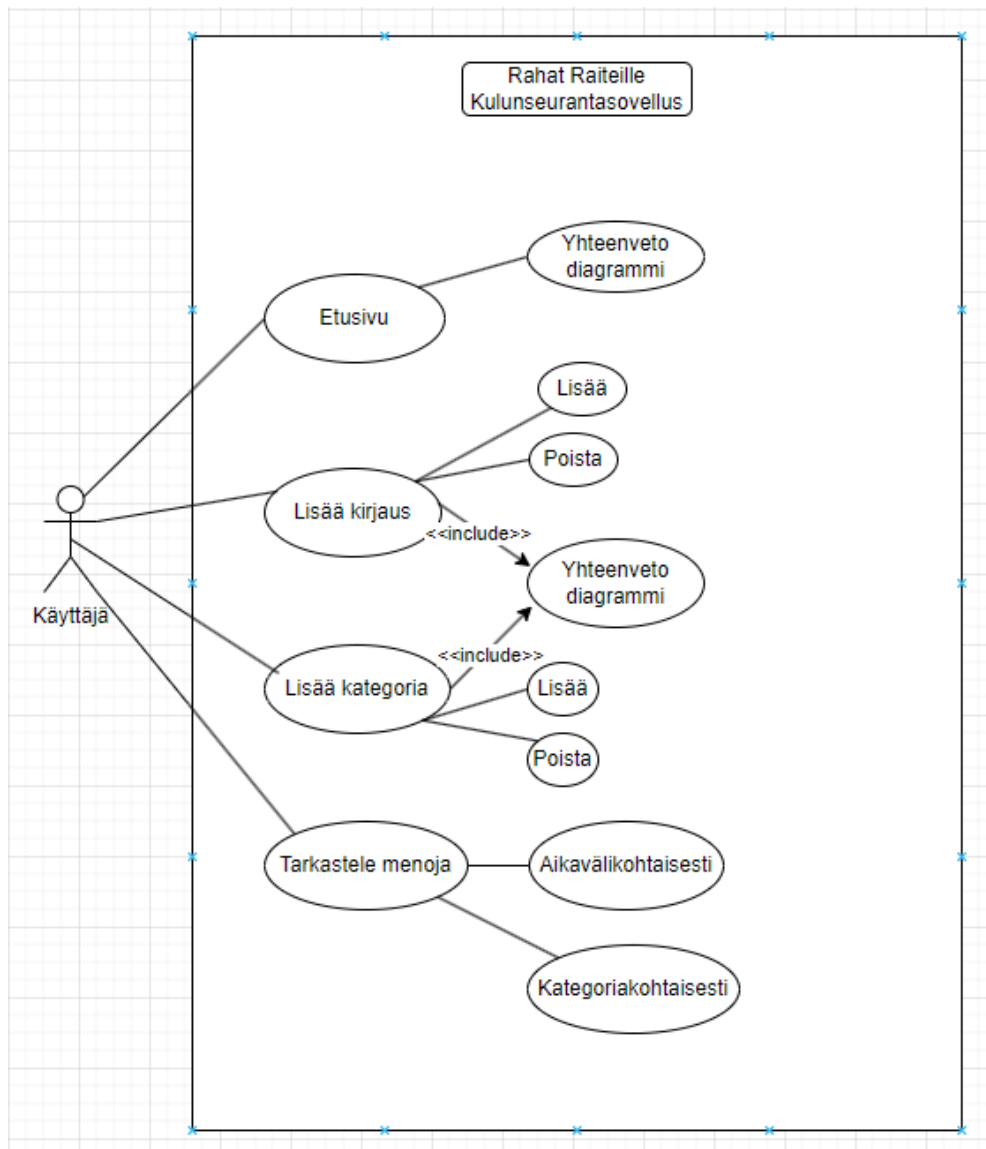
Sovellus on kevyt, joten laitteelta ei vaadita korkeaa suorituskykyä.

3.4 Tiedostot ja asetustiedostot

Käyttäjän luodut kategoriat ja kirjaukset tallennetaan json muodossa, minkä ansiosta käyttäjän ei tarvitse merkata niitä joka kerta kun avaa sovelluksen.

4. Toiminnot

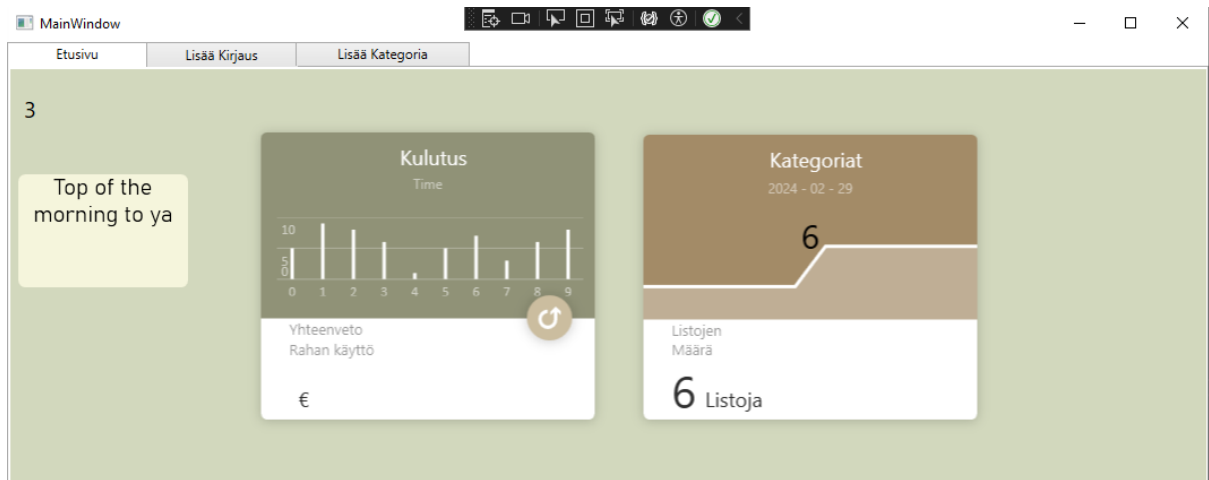
RahatRaitelle sovelluksessa voit kirjata omia kulujasi ja seurata, mihin käytät rahaa ja kuinka paljon. Sovellus antaa myös sinulle säästövinkkejä ja diagrammin, josta voit seurata kulutusta. Alla olevassa kaaviossa on yksinkertaisesti esitetty sovelluksen käyttöliittymä.



Kuva 1. UML-kaavio tulevasta projektista.

4.1 Etusivu

Etusivu, jossa käyttäjä näkee rahan kulutuksen ja luodut kategoriat. Kuvassa näkyy myös pop-up, jonka tarkoitus on kertoa rahan säästö vinkkejä.



Kuva 2. Etusivu näkymä.

4.2 Lisää kategoria

Käyttäjä luo omia kategorioita kirjoittamalla nimen ja värin. Kun kategoria on luotu, ilmestyy se listaan. Kun käyttäjä haluaa poistaa kategorian kirjoittaa hän poistettavan kategorian nimen ja varmistaa poiston, jolloin kategoria katoaa listalta.

5

Luodut Kategoriat:

- qqq -
- htht -
- qsqs -
- Rouka - Keltanen
- Matka - Punanen
- Qqqq - Vihree

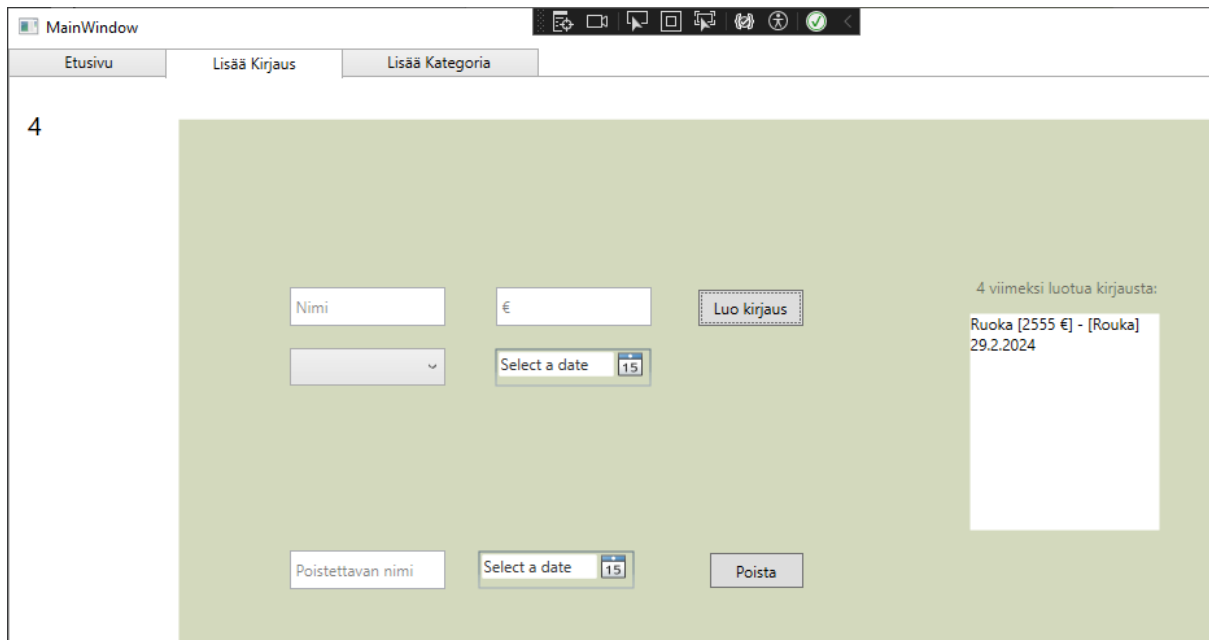
Nimi Väri Lisää

Poistettavan nimi Poista

Kuva 3. Käyttäjä pystyy lisäämään kategorioita.

4.3 Lisää kirjaus

Käyttäjä luo kirjauksen omista menoistaan nimeemällä sen ja määrittelemällä hinnan, kategorian ja päivämäärän. Luodut kirjaukset näkyvät listassa. Käyttäjä onnistuu poistamaan kirjauksen kirjoittamalla poistettavan kirjauksen nimen ja päivämäärän, jolloin kirjaus katoaa listasta.



Kuva 4. Käyttäjä pystyy luomaan/poistamaan kirjauksen omista menoistaan.

5. Ulkoiset liittymät

WPF-sovellus hyödyntää pääosin web-palveluita, jonka ansiosta sovelluksen päivittäminen onnistuu etäpalvelimelta.

6. Muut ominaisuudet

Tässä kappaleessa käsitellään sovelluksen erilaisia ominaisuuksia, kuten suorituskykyä, turvallisuutta, ylläpidettävyyttä ja käytettävyyttä. Nämä ominaisuudet ovat avainasemassa sovelluksen käytettävyyden ja käyttäjäkokemuksen kannalta.

6.1 Suorituskyky ja vasteajat

Tavoite vasteaika on 1ms. Sovellus pystyy tukemaan montaa erikäyttäjää samanaikaisesti ja sovellus on kevyt joten laitteelta ei vaadita paljoa.

6.2 Saavutettavuus (availability), toipuminen, turvallisuus, suojaukset

Sovelluksen tarkoitus on olla vuorokauden ympäri ylhäällä. Ennen huoltokatkota ilmoitetaan päivä ennen huoltokatkoa että sovellus on kiinni. Toipumiseen käytetään niin paljon aikaa kun siihen vaaditaan, mutta tavoite on pitää se lyhyenä.

6.3 Ylläpidettävyys

Sovellus tulee varmuuskopioida säännöllisesti, jotta minimoidaan mahdollisuus tietojen häviämiseen. Sovelluksen kehittäjät vastaavat sen säännöllisestä päivittämisestä.

6.4 Siirrettävyys ja yhteensopivuus

Ohjelmistossa on käytössä kielenä C# WPF. Ainoi yhteensopiva käyttöjärjestelmä on Windows mutta saatetaan laajentaa tulevaisuudessa muillekin alustoille.

6.5 Operointi

Sovelluksen tulee olla yksinkertainen ja hyvin toimiva käyttäjäkokemuksen parantamisen perusteella. Käyttäjän antamat tiedot ovat suojattuna, jotta henkilökohtaiset taloustiedot eivät ole muiden saatavilla ja turvataan yksityisyyttä.

6.6 Käytettävyys (Usability), käytön tehokkuus, käyttäjien tyytyväisyys

Kirjaa ominaisuudet, joita järjestelmän loppukäyttäjät asettavat järjestelmälle, jotta se toimisi tarkoituksensa mukaisesti ja tehokkaasti sekä olisi mahdollisimman käyttäjäystävällinen. Esimerkiksi käyttöliittymän selkeys, käytön oppimisen helppous, vikasietoisuus jne.

7. Suunnittelurajoitteet

Tässä kappaleessa käydään läpi sovelluksen suunnitteluun vaikuttaneista rajoitteista. Rajoitteiden huomiointi on tärkeää, jotta

7.1 Standardit

Sovellukselta vaaditaan kykyä tallentaa käyttäjän tietoa ja antaa käyttäjälle yksinkertainen sovellus rahankulutuksen seurantaan varten. Käyttäjä voi nappia painamalla operoida sivujen välillä, mikä tekee käyttöliittymästä todella yksinkertaisen.

7.2 Laitteistorajoitteet

Sovellus on WPF-sovellus, jolla on tiettyjä laitteistovaatimuksia. Nämä listatut vaatimukset on otettava huomioon sovellusta kehittäessä.

Käyttöjärjestelmäversio = WPF-sovellusten tulee toimia Windows-käyttöjärjestelmässä, ja niiden toiminta voi vaihdella eri Windows-versioissa.

Prossessoriteho = Monimutkaiset tai raskaat WPF-sovellukset voivat vaatia enemmän prosessoritehoa toimiakseen sulavasti. Vanhemmat tai vähemmän tehokkaat

prosessorit saattavat kokea suorituskykyongelmia erityisesti, jos sovellus käyttää paljon animaatioita tai grafiikkaa.

Muisti = WPF-sovellukset voivat käyttää muistia, erityisesti jos ne käsittelevät suuria tietomääriä tai grafiikkaelementtejä. Riittämätön muistimäärä voi johtaa sovelluksen hidastumiseen tai jopa kaatumiseen.

Näytönohjain = Graafisesti vaativat WPF-sovellukset voivat vaatia tehokkaan näytönohjaimen ja hyvän grafiikkatuen toimiakseen optimaalisesti. Joissain tapauksissa vanhemmat tai integroidut näytönohjaimet eivät tue kaikkia WPF:n ominaisuuksia, kuten 3D-grafiikkaa tai kiihdytettyä videon toistoa.

Resoluutio = Korkean resoluution näytöt voivat aiheuttaa ongelmia WPF-sovelluksille, jotka eivät ole skaalautuvia. Sovellukset voivat näyttää epätarkoilta tai niiden käyttäily voi olla hankala käyttää, jos ne eivät tue korkeaa resoluutiota.

Tallennustila = Riittämätön tallennustila voi vaikuttaa sovelluksen suorituskykyyn varsinkin, jos se käsittelee suuria tietomääriä tai tallentaa paljon väliaikaista tietoa.

WPF-versio = Uudemmat WPF-versiot voivat sisältää optimointeja ja parannuksia suorituskykyyn verrattuna vanhempiin versioihin. Siksi vanhemmat WPF-versiot saattavat toimia hitaammin tai kohdata enemmän ongelmia tietyissä tilanteissa.

7.3 Ohjelmistorajoitteet

Kerrotaan mahdollisista rajoitteista, jotka voivat johtua käytetystä kielestä, tekniikan ongelmasta tai jokin muu rajoite. On tärkeää kertoa rajoitteista, jotta lukija tietää, onko esimerkiksi sovellus tai toiminto hänen laitteellaan toimiva.

Käyttöjärjestelmäriippuvuus = WPF-sovellukset toimivat vain Windows-käyttöjärjestelmissä. Tämä tarkoittaa sitä, että ne eivät ole suoraan käytettävissä muilla käyttöjärjestelmillä, kuten Linuxilla tai macOS:lla. Tämä voi olla rajoite, jos haluat kehittää monialustaisen sovelluksen.

.NET Framework -versioriippuvuus: WPF-sovellukset perustuvat .NET Frameworkiin, joten niiden suorittaminen edellyttää asianmukaista .NET Framework -versiota, joka on asennettu käyttöjärjestelmään. RahatRaiteille sovelluksessa on käytetty .NET Framework 8.0-versiota.

7.4 Muut rajoitteet

Projektimme suorittamiseen annettiin viiden viikon aikataulu. Aikataulu todettiin todella tiukaksi, minkä johdosta suunnittelimme tuotteen, joka toimii ja antaa käyttäjälle hyvän kokemuksen. Ideoita oli paljon, joista vaativimmat siirrettiin osaksi sovelluksen jatkokehitystä.

8. Hylätyt ratkaisuvaihtoehdot

Tarkastelu sivulla piti olla piechart, joka saisi datan json:sta. Aika oli lopussa ja vaikea toteuttaa niin jätimme sen pois ja siirryimme helpompaan ja yksinkertaisempaan ideaan.

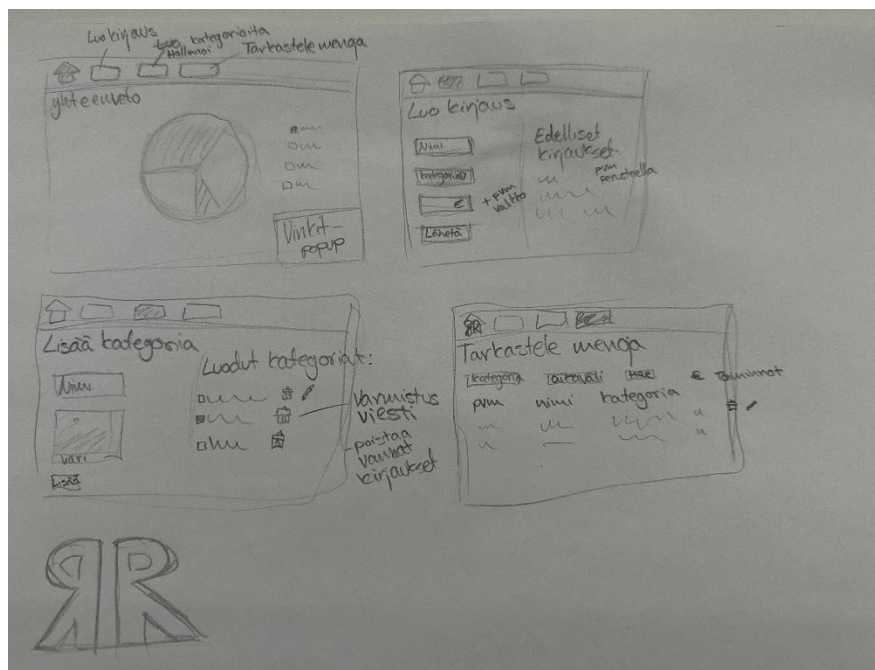
9. Jatkokehitysjatuksia

Tulevaisuudessa tarkoitus on laajentaa sovellus muillekin käyttöjärjestelmille, jotta useampi ihminen pääsisi käyttämään sovellusta käyttöjärjestelmästä huolimatta.

Sovelluksessa olisi hyvä olla päiväkohtainen budjettilaskuri, joka laskee rahamäärän ja kuukaudessa olevien päivien avulla, kuinka paljon rahaa voi päivässä käyttää. Myös etusivulla olevan statistiikka olisi hyvä saada toimimaan. Nämä ratkaisut hylättiin aikataulun seurauksesta.

Liitteet

Liite 1: RahatRaiteille ulkonäkö piirrettynä



Kuva 5. Piirretty versio tulevasta ulkonäöstä ja logosta.

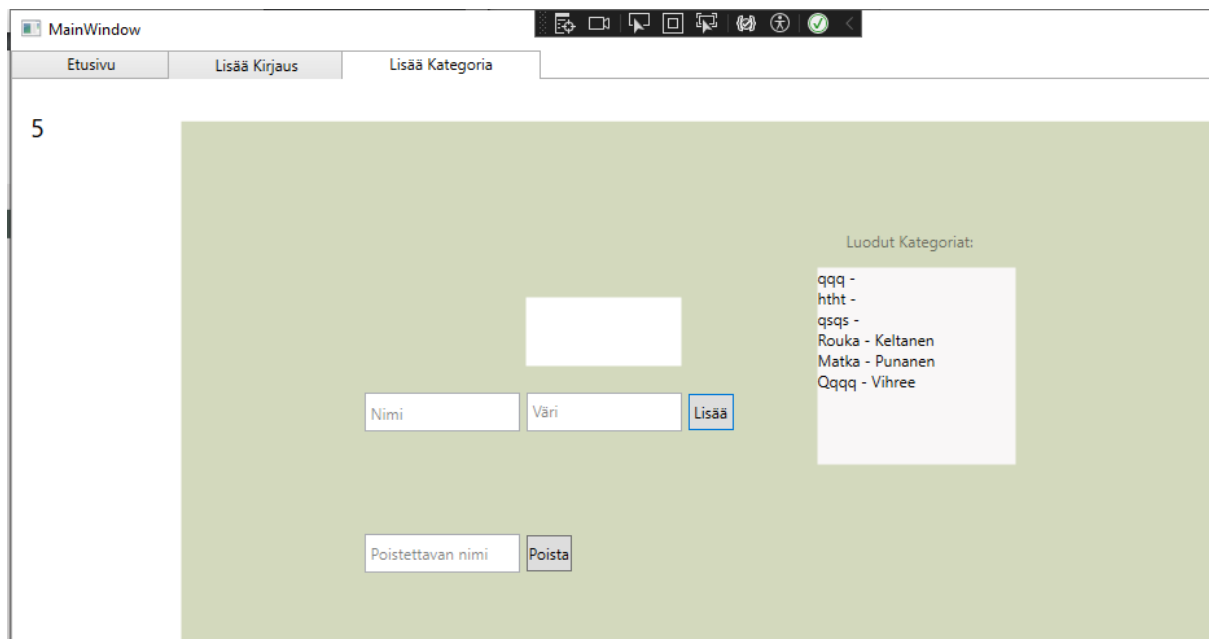


Kuva 6. Lopullinen versio logosta

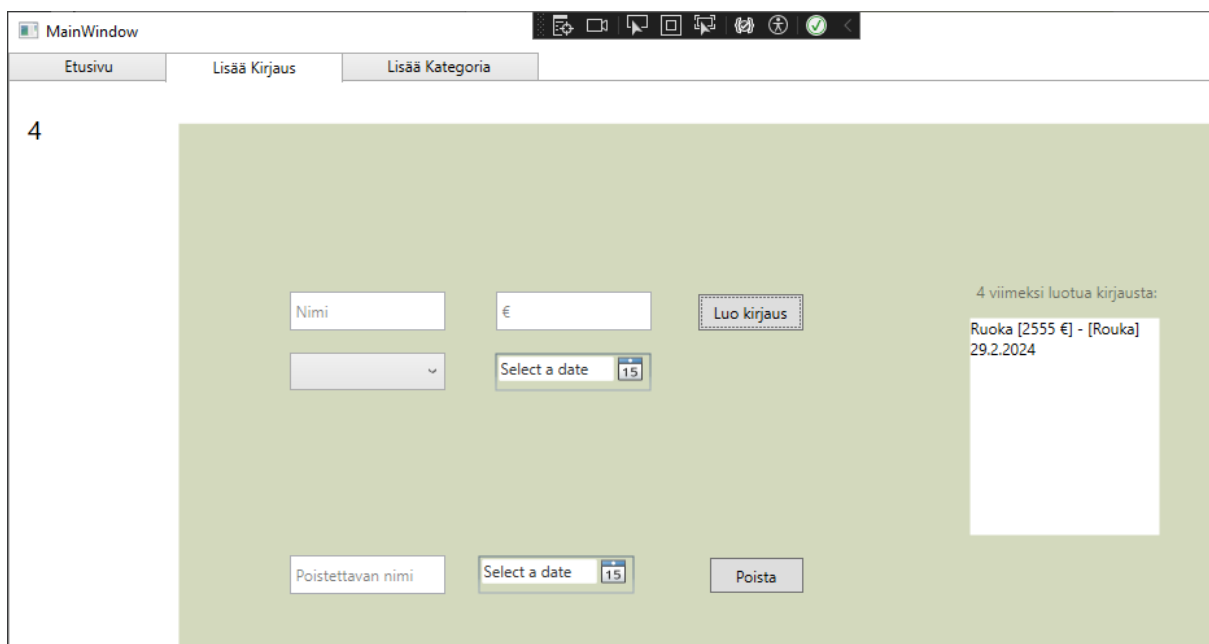
Liite 2: RahatRaiteille käyttäminen



Kuva 7. Etusivu, josta näkee rahan käytön ja listojen määrän.



Kuva 8. Lisää Kategoria sivu, jossa käyttäjä voi luoda/poistaa oman kategorian. Luodut kategoriat näkyvät "Luodut Kategoriat" listassa.



Kuva 9. Lisää Kirjaus sivu, jossa käyttäjä voi lisätä menonsa nimen, summan, kategorian ja päivämäärän. Luodut kirjaukset näkyvät listalla. Käyttäjä pystyy myös poistamaan kirjauksen listalta.