

Calo4U

Vaatimusmäärittely

Bunyisa Jianjumnong

Elmeri Uitto

Ilari Melin

Tuomas Kilpeläinen

Sisällysluettelo

1. Johdanto	4
1.1 Tarkoitus ja kattavuus.....	4
1.2 Tuote ja ympäristö	4
1.3 Määritelmät, termit ja lyhenteet	4
1.4 Viitteet.....	5
1.5 Yleiskatsaus dokumenttiin	5
2. Yleiskuvaus	6
2.1 Ympäristö	6
2.2 Toiminta.....	6
2.3 Käyttäjät	7
2.4 Yleiset rajoitteet	7
2.5 Oletukset ja riippuvuudet	7
3. Tiedot ja tietokanta	8
3.1 Tietosisältö	8
3.2 Käyttöintensiteetti.....	8
3.3 Kapasiteettivaatimukset.....	8
3.4 Tiedostot ja asetustiedostot.....	9
4. Toiminnot	10
4.1 Lisää resepti.....	11
4.2 Katso reseptit	12
4.3 Kalorintarvelaskuri	13
4.4 Reseptin valmistus	14
4.5 Reseptin lisäys etusivulle	15
4.6 Syö/poista	15
4.7 Kaloriseuranta	16
5 Ulkoiset liittymät	17

6. Muut ominaisuudet.....	18
6.1 Suorituskyky ja vasteajat.....	18
6.2 Saavutettavuus (availability), toipuminen, turvallisuus, suojaukset	18
6.3 Ylläpidettävyys	18
6.4 Siirrettävyys ja yhteensopivuus.....	18
6.5 Operointi	19
6.6 Käytettävyys (Usability), käytön tehokkuus, käyttäjien tyytyväisyys.....	19
7.Suunnittelurajoitteet	20
7.1 Standardit	20
7.2 Laitteistorajoitteet	21
7.3 Ohjelmistorajoitteet	21
7.4 Muut rajoitteet.....	22
8. Hylätyt ratkaisuvaihtoehdot.....	23
9. Jatkokehityssajatuksia	24
Lähteet	25

1. Johdanto

Tässä kappaleessa esittelemme Calo4U-sovelluksemme yleispiirteisesti, keskittyen sen käyttötarkoitukseen, käyttöympäristöön ja mahdollisiin viitteisiin.

1.1 Tarkoitus ja kattavuus

Calo4U on luotu helpottamaan käyttäjän kaloreiden seurantaan. Sen päätarkoitus on olla helppo reseptikirjasto, johon käyttäjä voi lisätä omia reseptejä ja seurata viikkotasolla omaa kalorisaantia. Calo4u myös tallentaa raaka-aineiden kalorimäärät muistiin, joka nopeuttaa uusien reseptien luontia.

1.2 Tuote ja ympäristö

Calo4u on reseptisovellus, jota käytetään ruuanlaiton yhteydessä. Se helpottaa omien kaloreiden seurantaan siihen rakennetulla kalorin seuranta toiminolla. Sovellusta käytetään tietokoneella.

1.3 Määritelmät, termit ja lyhenteet

JSON (JavaScript Object Notation) on tiedon tallennukseen ja välitykseen luotu yksinkertainen ja kevyt avoimen standardin tiedostomuoto. JSON-tiedostoihin on helppo kirjoittaa tietoa ja tieto on ihmisille helppolukuista.

WPF (Windows Presentation Foundation) on Microsoftin kehittämä ilmainen avoimen lähdekoodin käyttöliittymän pohja, joka toimii grafiikan, tekstin, animaatioiden ja videon piirron rajapintana Windows-käyttöjärjestelmillä.

XAML (Extensible Application Markup Language) on merkintäkieli, jota käytetään pääasiassa Microsoftin kehittämissä ohjelmointikehyksissä, kuten WPF:ssä graafiseen suunnitteluun ja määrittelyyn.

C# on Microsoftin kehittämä korkean tason ohjelmointikieli. C# on suosittu ja monipuolinen ohjelmointikieli, jota voi käyttää laajasti erilaisiin ohjelmistoihin, kuten WPF-ohjelmien logiikan tekemiseen.

.NET Framework on Microsoftin kehittämä ohjelmistokehys, jota Microsoft Visual Studiossa tehdyt ohjelmistot käyttävät. Näihin kuuluu esimerkiksi WPF-ohjelmistot.

Kaloritarve on suositeltu kalorimäärä, jonka tulisi saavuttaa tietyn ajan kuluessa onnistuakseen omassa tavoitteessa.

Resepti sisältää yksityiskohtaisen kuvauksen ruoanlaittoon tarvittavista raaka-aineista, annosmäärästä ja valmistusmenetelmästä.

Annos on yhteen ruokailukertaan tarjottavan ruoan määrään.

Ohje sisältää tarkat vaiheet ja toimenpiteet, joita tulisi noudattaa ruoan valmistuksessa.

Aktiivisuus viittaa fyysisen toiminnan ja liikunnan määrään

1.4 Viitteet

Sovelluksen tekemisessä on käytetty hyödyksi seuraavia lähteitä

Expert Pt. (n.d.). BMR Formula. Haettu 19.2.2024 osoitteesta <https://expertpt.co.uk/news/how-many-calories-should-i-eat>

W3 Schools. (n.d.). Haettu 1.2.2024 osoitteesta <https://www.w3schools.com/>

Stack Over Flow (n.d.). Haettu 1.2.2024 osoitteesta <https://stackoverflow.com/>

Live Charts. (n.d.). Haettu 26.2.2024 osoitteesta <https://livecharts.dev/>

Microsoft. (5.13.2022). C# Compiler Errors. <https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/compiler-messages/>

1.5 Yleiskatsaus dokumenttiin

Tässä dokumentissa esitellään Calo4U-sovelluksen käyttötarkoitus, sen sisältämät ominaisuudet sekä käyttöympäristö. Pääasiallinen kohdeyleisö ovat Calo4U-sovelluksen käyttäjät, jotka etsivät tietoa ja ohjeistusta sovelluksen käyttöön.

Dokumentti sisältää myös tietoa Calo4U-sovelluksen tietokannasta, toiminnoista, ulkoisista liittymistä, muista ominaisuuksista, suunnittelurajoitteista, hylätyistä ratkaisuvaihtoehdoista, jatkokehitysideoista ja liitteistä

2. Yleiskuvaus

Tässä kappaleessa esittelemme sovelluksen ympäristöä, toimintaa, yleisiä rajoitteita sekä oletukset ja riippuvuudet

2.1 Ympäristö

Sovellusta hyödynnetään pääasiallisesti ruoanvalmistuksen aikana ja aterian nauttimisen jälkeen. Käyttäjät voivat seurata sovelluksen antamia ohjeita samalla kun he loihtivat herkullisia ruokalajejaan. Lisäksi he voivat jälkikäteen tallentaa nautitun aterian kalorimäärän tavoitteensa saavuttamiseksi ja seurata edistymistään kohti asetettua kaloritavoitetta.

2.2 Toiminta

Calo4U -sovelluksella käyttäjä voi kysyä sovellukselta omaa päivittäistä kalorintarvetta käyttämällä sovelluksen tarjoamaa kalorilaskuria. Jos käyttäjä on tietoinen omasta päivittäisestä kalorintarpeesta, hän voi syöttää myös oman arvonsa.

Tällä sovelluksella käyttäjä voi myös lisätä oman reseptinsä syöttämällä raaka-aineiden määrät ja niiden kalorimäärät sekä ohjeet, minkä jälkeen resepti tallennetaan sovellukseen. Myöhemmin kun käyttäjä haluaa valmistaa ruoka-annoksen, hän voi hakea kyseisen reseptin reseptiluettelosta ja lisää sen valmistettujen ruokien listalle.

Kun käyttäjä on syönyt ruoka-annoksen, kyseisen ruoka-annoksen kalorimäärät päivittyvät etusivulla olevaan ympyräkaavioon, joka näyttää viikon syödyt kalorimäärät sekä viikon jäljellä olevat kalorimäärät.

Calo4U -sovellusta avattaessa käyttäjälle aukeaa etusivu, jossa näkyy navigointipalkki, lista valmisteuista ruoista ja viikon kalorig seuranta, joka on kuvattu ympyräkaaviona.

- Navigointipalkki on sovelluksen jokaisella sivulla, jonka avulla käyttäjä pääsee:
 - Etusivulle
 - Lisää uusi resepti -sivulle
 - Katso reseptit -sivulle
 - Kalorintarve -sivulle
- Valmistettujen ruokien listalle käyttäjä voi lisätä omat valmistamansa ruoat, joko Katso resepti -sivun kautta tai etusivulla olevan manuaalisen syötön kautta, jonka jälkeen käyttäjän syötyä, hän painaa syö -painiketta, minkä jälkeen sen ruoan kalorimäärä päivittyy ympyräkaavioon.

- Ympyräkaavio tarjoaa käyttäjälle tietoa viikon kaloriseurannasta, jossa näkyy viikon syöty kalorimäärä sekä viikon jäljellä oleva kalorimäärä. Ympyräkaavion alareunassa esiintyy myös viikon numero, viikon kaloritavoite, saavutettu kalorimäärä ja kuinka paljon kaloreita on jäljellä tällä viikolla.

2.3 Käyttäjät

Sovelluksen käyttäjät muodostuvat monipuolisesta käyttäjäkunnasta, johon kuuluvat henkilöt, jotka ovat aktiivisesti kiinnostuneita oman kalorikäytön seuraamisesta ja terveellisestä ravitsemuksesta. Lisäksi käyttäjien joukossa on niitä, jotka ovat kiinnostuneet vain ruoanlaitosta ja haluavat tehdä ja tallentaa omia reseptejään reseptikirjastoon.

2.4 Yleiset rajoitteet

Jotta käyttäjä voisi hyödyntää sovelluksen kalorilaskuria, hänen tulee tuntee ruoanvalmistuksessa käytettyjen raaka-aineiden kalorimäärät. Sovellus on vain suomen kielellä, joten kielen osaaminen on suositeltavaa.

2.5 Oletukset ja riippuvuudet

Sovellukseen liittyvä oletus on, että sovelluksen käyttäjä pitää terveelliset elämäntavat tärkeänä osoittaen suurta kiinnostusta ravitsemukseen ja käyttäjä haluaa ylläpitää tasapainoista kalorinsaatia.

Tämä sovellus on suunniteltu käyttäjäystävällisyyteen, mikä mahdollistaa hyödyllisen käytön ilman liiallista keskittymistä käytön aikana ja esim. jos raaka-aineen määrä on väärä, sovelluksessa löytyy poisto- ja muokkauspainikkeet, joiden avulla käyttäjä voi muokata tai poistaa ei toivottua raaka-ainetta tai määrää haluamallaan tavalla.

Myös tallennettu resepti auttaa käyttäjää säästämään aikaa, koska sitä voi aina muokata ja se voidaan helposti valmistaa eri annoksen määrällä.

Sovelluksen maksimaalinen käyttö riippuu todella paljon käyttäjän aktiivisuudesta, kuinka aktiivisesti käyttäjä lisää uusia reseptejä, raaka-aineita ja kaloritetoja. Jos käyttäjä on vähemmän taipuvainen tarjoamaan aktiivisesti mainittuja tietoja, sovelluksen maksimaalinen osaaminen ei välttämättä realisoidu. Lisäksi tehokkaan hyödyn saaminen sovelluksesta edellyttää, että käyttäjä on tietoinen omasta pituudestaan, painostaan ja iästään kalorilaskuria käytettäessä.

3. Tiedot ja tietokanta

Kappaleessa käsitellään Calo4U:n tiedonhallintaa, tietojen tallennusta ja tietokantaa.

3.1 Tietosisältö

Sovellus tallentaa kaikki tiedot paikallisesti neljään eri JSON-tiedostoon. Käyttäjille, viikko tiedoille, resepteille ja raaka-aineiden kaloreille sovellus luo omat tiedostonsa ja Tallentaja-luokka hoitaa tähän liittyvät toiminnot. Tietokantoihin lisätään käyttäjän lisäämät reseptit, raaka-aineet sekä niiden kalorit ja käyttäjän omat tiedot, viikkojen syödyt ja tavoite kalorit, sekä kalorin laskennan aloitus päivämäärä.

3.2 Käyttöintensiteetti

Reseptejä tallennetaan aluksi suurempia määriä, mutta kun sovellusta on pidempään käyttänyt, jää reseptin tallennus vähemmälle. Sovellus joutuu kuitenkin hakemaan aina valmiiden reseptien katselemisessa kaikki reseptit uudestaan tietokannoista, joten lataamisintensiteetti voi olla hieman suurempi, riippuen resepti kirjaston koosta. Lisäksi sovellus tallentaa aina käyttäjän syödyt kalorit käyttäjä kirjastoon, kun käyttäjä syö tai lisää kaloreita.

3.3 Kapasiteettivaatimukset

Sovellus vaatii alle 50 megatavua tilaa ja suoritusvaatimukset ovat vähäiset. Sovelluksen vaatima tila kasvaa hiukan, mitä enemmän reseptitietoa sisällä oleviin JSON-tiedostoihin lisätään, mutta tämä määrä ei tule ylittämään edes yhtä megatavua normaalilla käytöllä. Sovellus ei vaadi verkkoyhteyttä, joten käyttäjien määrä ei vaikuta suorituskykyyn.

3.4 Tiedostot ja asetustiedostot

Kaikki tallennukseen ja lataukseen liittyvät toiminnot ovat sijoitettu Tallentaja-luokkaan. Tallentaja muodostaa neljä JSON-tiedostoa;

reseptiKirjasto:

- Reseptin nimi
- Reseptin ohjeet
- Reseptin annos määrän
- Reseptin raaka-aineet
- Raaka-aineen nimi
- Raaka-aineen määrä grammoina

kayttajaKirjasto:

- Käyttäjän Nimi (Tässä versiossa vain yksi käyttäjä nimeltä "Käyttäjä")
- Päivän kalori tarve
- Valmiit ruuat lista
- Syödyt kalorit (viikossa)
- Viikko
- Aloitus päivämäärä

kaloriKirjasto:

- Raaka-aineen nimi
- Raaka-aineen kalorit sadassa grammassa

viikkoKalorit:

- Viikkonumero
- Kyseisen viikon päivän kalorit
- Kyseisen viikon syödyt kalorit

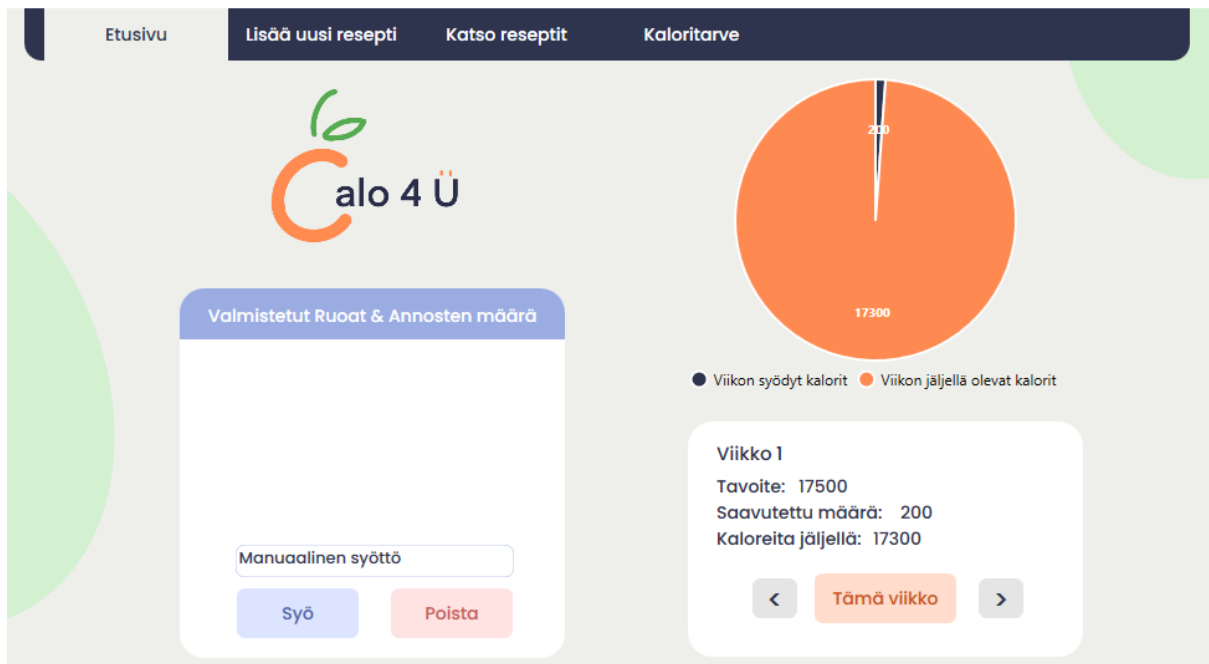
Näille jokaiselle löytyy kaksi funktiota:

- LataaKaikki(): Lataa koko JSON-tiedoston ja purkaa sen omaan luokkaan muotoonsa ja palauttaa kirjaston lista muodossa.
- Tallenna(): Lataa ensin koko tietokannan ja tämän jälkeen lisää siihen tallennettavan tiedon, jonka jälkeen muuttaa tiedot JSON-muotoon ja kirjoittaa tiedot omaan JSON-tiedostoonsa.

Näiden lisäksi Tallentajasta löytyy resepteille omat funktiot tietyn reseptin lataamiseen ja poistamiseen. Nämä funktiot toimivat samalla periaatteella kuin LaataKaikki(), mutta ne palauttavat vain haetun reseptin tai poistavat sen kokonaan. Tallennus toimii paikallisesti, joten se vaatii ainoastaan tietokoneelta tilaa toimiakseen.

4. Toiminnot

Calo4U-sovellukseen voi lisätä reseptejä ja tarkastella niitä. Sovelluksella voi myös laskea viikoittaisen kalorikulutuksen ja sen perusteella seurata viikoittaista kalorinkulutusta. Kappaleessa kuvaillaan näitä ominaisuuksia tarkemmin.



Kuva 1. Sovelluksen etusivu

4.1 Lisää resepti

Lisää resepti toiminnolla voi luoda reseptin syöttämällä Raaka-aineita, niiden grammapainon ja kilokalorimäärän. Raaka-aineiden syötön jälkeen voit nimetä reseptin, syöttää tehtyjen annosten määrän ja reseptin ohjeet. Jos raaka-ainetta syöttäessä raaka-aine löytyy kaloriKirjasto JSON-tiedostosta, ohjelma täyttää automaattisesti raaka-aineelle aikaisemmin syötetyn kilokaloriarvon.

Etusivu Lisää uusi resepti Katso reseptit Kaloritarve

Raaka-aine

Määrä (g)

Kalorit (cal/100g)

Lisää raaka-aine Poista

kana määrä: 100g kalorit: 200kc/100g

Reseptin nimi:

Annosten määrä:

Ohjeet:

Tallenna resepti

Kuva 2. Lisää uusi resepti sivu

4.2 Katso reseptit

Katso reseptit toiminto näyttää tehtyjen reseptien nimet, raaka-aineet, raaka-aineiden kalorit ja reseptin ohjeet. Reseptin voi etsiä hakukentän avulla. Näet myös reseptiä luodessa annettujen annoksien määrän. Annoksien määrää voi myös muokata. Tässä näkee myös yksittäisen annoksen kalorimäärän ja koko valmistetun reseptin kalorimäärät. Lopuksi voit lisätä ruoan pääsivun ruokalistalle.

Etusivu

Lisää uusi resepti

Katso reseptit

Kaloritarve

Reseptin nimi:

Poista

Ohjeet:

Muokkaa

Resepti

Annoksien määrä:

Kalorit (cal):

Annoksien määrä:

Valmista ruoan

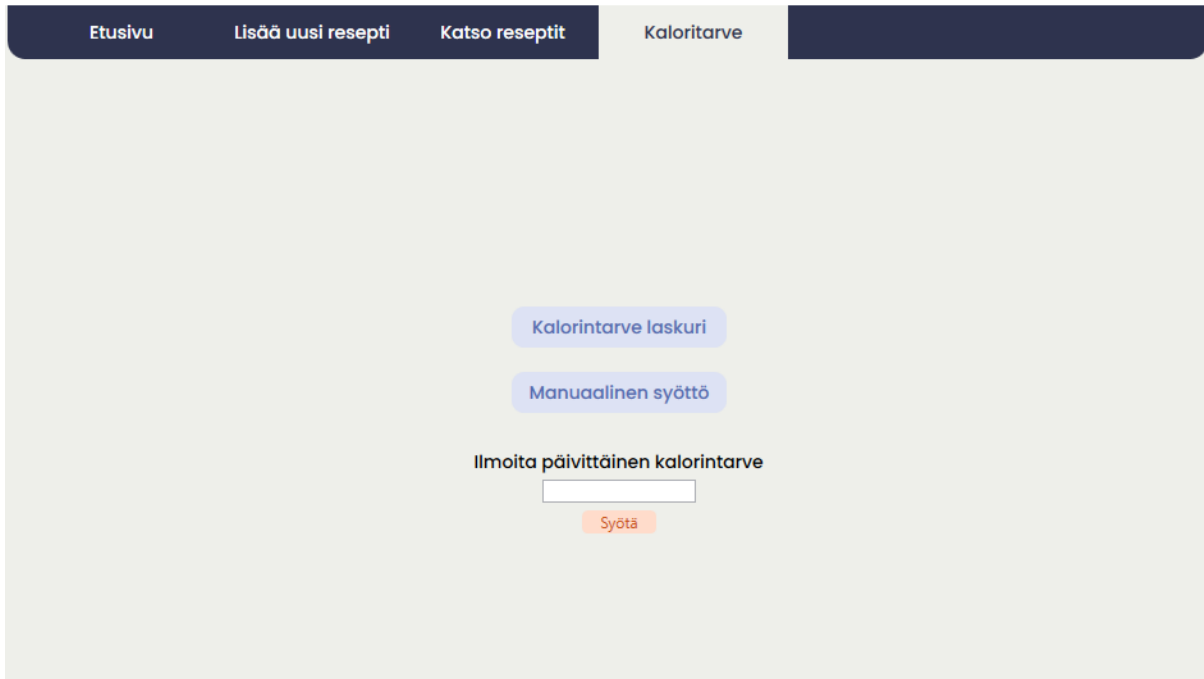
Lisää ruokalistalle

+

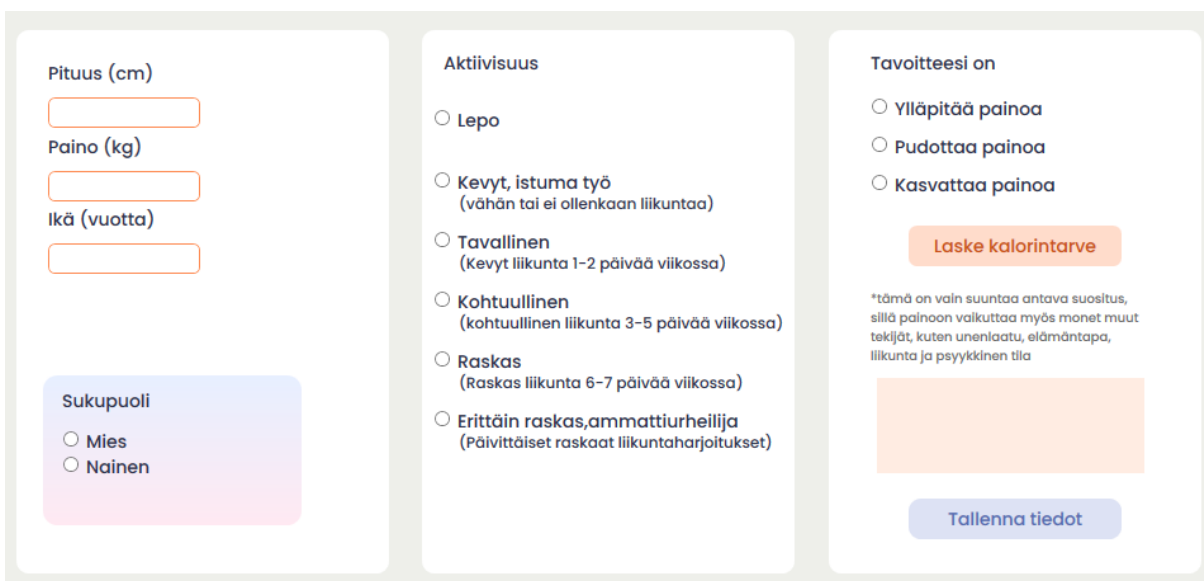
Kuva 3. Katso reseptit sivu

4.3 Kalorintarvelaskuri

Kalorintarvelaskurissa on kaksi valintaa, manuaalinen syöttö ja laskuri. Manuaaliseen syöttöön voi kirjata päivittäisen kalorimäärän manuaalisesti. Kalorintarvelaskuriin syötetään pituus, paino, ikä, sukupuoli, aktiivisuustaso ja painotavoite. Laskuri laskee näistä tiedoista suositellun kilokalorimäärän tavoitetta varten. Tämän arvon voi tallentaa, joka menee etusivun tavoitekenttään.



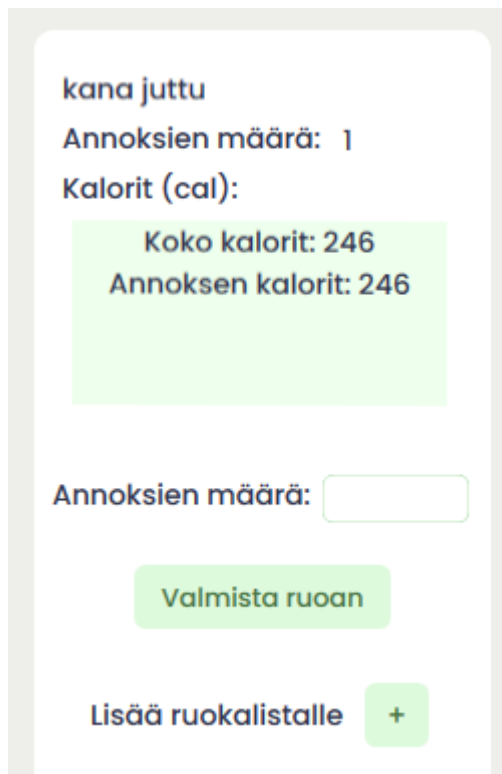
Kuva 4. Kalorilaskuri ja manuaalinen valinta sivu



Kuva 5. Kalorilaskuri.

4.4 Reseptin valmistus

Reseptin valmistus hakee resepti kirjastosta valitun reseptin ja muokkaa raaka-aineiden määrät käyttäjän syöttämän annosmäärän mukaan. Jos annosmäärää ei ole syötetty, sovellus ottaa annosmääräksi reseptin alkuperäisen annosmäärän. Tämän jälkeen sovellus tulostaa näytölle reseptin ohjeet, raaka-aineet, annokset, nimet, kalorimäärät ja samalla tallentaa sen Resepti muuttuunaan valmiiksi reseptin lisäystä etusivulle.



kana juttu

Annoksien määrä: 1

Kalorit (cal):

Koko kalorit: 246
Annoksen kalorit: 246

Annoksien määrä:

Valmista ruoan

Lisää ruokalistalle

Kuva 6. Lisää ruokalistalle.

4.5 Reseptin lisäys etusivulle

Reseptin lisäys etusivulle toimii painamalla Katso reseptit sivun Lisää ruokalistalle nappia. Tämä hakee reseptin reseptiKirjasto JSON-tiedostosta ja lisää sen etusivun valmistettujen ruokien kenttään. Jos ruoka on jo listalla, pelkästään annosten määrä lisääntyy.

Lisää ruokalistalle 

Kuva 7. Reseptin etusivulle lisäys nappi

4.6 Syö/poista

Etusivun valmistettujen ruokien kentässä on annoksen syömistä ja poistamista varten napit. Syö nappi lisää annoksen kalorimäärän käyttäjäKirjasto JSON-tiedostoon ja viikon saavutettu määrä näkyy kaloriseurannassa. Kentässä on myös manuaalinen syöttö mahdollisuus, jolla kaloreita pystyy lisätä ja poistaa manuaalisesti viikon saavutetusta kalorimäärästä.



Valmistetut Ruokat & Annosten määrä

kana juttu, Annokset: 1

Manuaalinen syöttö

Syö Poista

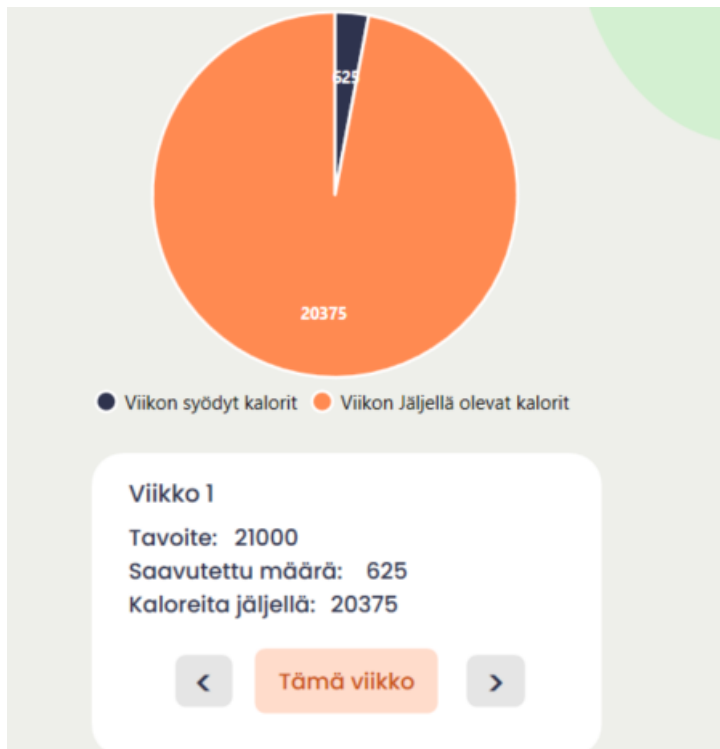
Kuva 8. Valmistetut ruokat ja syö/poista napit.

4.7 Kaloriseuranta

Etusivun kaloriseurantaan kuuluu kaksi osaa. Ensimmäiseksi sivulla on LiveCharts kirjastoa käyttävä ympyrädiagrammi, joka näyttää viikon syödyt kalorit ja viikon jäljellä olevat kalorit.

Toiseksi on kenttä, jossa lukee viikon numero, viikon kaloritavoite, viikon saavutettu kalorimäärä ja viikon kalorit jäljellä. Kentässä on myös napit edelliselle, nykyiselle ja seuraavalle viikolle.

Nykyinen nappi näyttää sen hetkisen viikon arvot `kayttajaKirjasto.json` sisällä. Edellinen ja seuraava viikko napit vaihtavat viikkonumeroa ja tämän perusteella etsii `viikkoKalorit.json` tiedostosta sen viikon kyseiset kaloriarvot.



Kuva 9. Kuva ympyrädiagrammista ja viikkoseurannasta.

5 Ulkoiset liittymät

Calo4U ei käytä ulkoisia liittymiä tässä versiossa ollenkaan, joten emme kerro niistä mitään.

6. Muut ominaisuudet

Tässä kappaleessa keromme sovelluksen muista ominaisuuksista, jotka eivät ole sovelluksen ydintoimintoja, mutta silti tärkeitä sovelluksen toimivuuden kannalta.

6.1 Suorituskyky ja vasteajat

Sovelluksen käyttötarkoitus on helpottaa ja nopeuttaa ruuan laittoa, joten sen tulee pyrkiä mahdollisimman nopeaan vasteaikaan. Vasteaika ei saisi ylittää yhden sekunnin rajapyykkiä, jotta sovelluksen käyttö pysyy miellyttävänä. Ohjelmiston vasteajat tulevat kasvamaan mitä enemmän sovelluksessa on reseptejä, joten tämä pitää ottaa huomioon jo kehitys vaiheessa.

6.2 Saavutettavuus (availability), toipuminen, turvallisuus, suojaukset

Sovellus on täysin paikallinen, joten sen tulee olla aina käytettävissä. Sovellus ei vaadi käyttäjältä tietoja, jotka voisivat olla tietoturvariskejä. Jos sovellusta päivitetään ja uusissa päivityksissä tulee käyttöön henkilökohtaisia tietoja, täytyy sovelluksen turvallisuutta kasvattaa.

Kun sovelluksessa ilmenee kriittisiä ongelmia, kuten bugeja, jotka kaatavat koko ohjelman, tai poistavat kaiken tallennettavan datan. Tulee se korjata mahdollisimman nopeasti, kuitenkin viimeistään kolmen päivän sisällä.

6.3 Ylläpidettävyys

Kehittäjien on asetettava säännölliset ja järjestelmälliset toimenpiteet etusijalle sovelluksen kestävyys ja luotettavuuden varmistamiseksi. Käyttäjän tietojen katoamisriskiä voidaan vähentää varmuuskopioinnilla.

Myös säännölliset ja johdonmukaiset päivitykset ovat elintärkeitä sovelluksen suorituskyvyn parantamiseksi, mahdollisten virheiden korjaamiseksi ja ominaisuuksien parannusten tarjoamiseksi käyttäjille. Nämä toimenpiteet edistävät sovelluksen yleistä vakautta ja turvallisuutta, jotka korostavat kehittäjien sitoutumista luotettavan ja ajan tasaisen käyttökokemuksen tarjoamiseen.

6.4 Siirrettävyys ja yhteensopivuus

Sovellus käyttää WPF:ää käyttöliittymän pohjana ja toiminnot ovat tehty C#-ohjelmointikielellä. Visuaaliset elementit sovelluksessa luodaan XAML kuvauskielellä. Sovellus vaatii .NET Frameworkin tietokoneella, joka on valmiiksi asennettu Windows Vistaan ja tätä uudempiin Windows käyttöjärjestelmiin. WPF-sovellukset toimivat vain Windows käyttöjärjestelmillä.

6.5 Operointi

Käyttäjän tyytyväisyyden kannalta, sovelluksen on oltava käyttäjäystävällinen ja käyttäjän odotusten mukainen. Sovelluksen operointi edellyttää tiukkoja toimenpiteitä käyttäjien henkilökohtaisten tietojen suojaamiseksi ja arkaluonteisten tietojen luvaton pääsyn tai vaarantumisen estämiseksi.

6.6 Käytettävyys (Usability), käytön tehokkuus, käyttäjien tyytyväisyys

Käytettävyys on sovelluksessa keskitytty paljon, sillä sovellusta on tarkoitus käyttää päivittäin. Tämän vuoksi sen pitää olla nopea- ja helppokäyttöinen. Yleisimmät toiminnot ovat helposti saatavilla etusivulla ja vähemmän käytetyt toiminnot ovat eri välilehdillä.

Käyttäjien tyytyväisyyttä varten palautteen antaminen pitää olla mahdollista ja tätä palautteita täytyy seurata aktiivisesti.

7.Suunnittelurajoitteet

Tässä kappaleessa pohditaan sovelluskehityksen aikana ilmeneviä suunnittelurajoitteita, joita tulee varmistaa sovelluksen laadun, toiminnallisuuden ja helppokäyttöisyyden varmistamiseksi sekä käyttäjien tarpeiden täyttämiseksi.

7.1 Standardit

Projektin tavoitteena on kehittää Windows-tietokoneelle sovellus. Tällöin projektin toteuttamisessa käytetään WPF:ää eli Windows Presentation Foundationia rakentamaan graafisen käyttöliittymän, joka mahdollistaa interaktiivisen eli vuorovaikutteisen viestinnän sovelluksen ja käyttäjän välillä.

WPF tarjoaa sovelluksen kehittämiseen tarvittavia työkaluja mm. XAML eli Extensible Application Markup Language. XAML-merkintäkielien käyttö mahdollistaa graafisen käyttöliittymän toteuttamisen. XAML-merkintäkielellä voi luoda sovellukseen eri käyttötarpeiden mukaan erilaisia sivuja, valintaikkunoita ja painikkeita, joita käyttäjä voi muokata ja määritellä sovelluksen käyttäytyminen halutulla tavalla.

MVVM(Model-View-ViewModel)-arkkitehtuuria käytetään yleisesti WPF-sovelluksen järjestämiseen ja jäsentämiseen. Jokaisella osalla on oma tehtävänsä.

Model on paikka, jossa sovelluksen toiminnot määrittävät koodit sijaitsevat. Model - puoli sisältää erilaisia sovelluksen logiikkaan perustuvia koodeja ja luokkia sekä tietovarastoja, jotka vastaavat sovelluksen toimintojen suorittamisesta.

View-ikkuna puolestaan näyttää XAML-koodit, jotka antavat visuaalisen esityksen sovelluksen käyttöliittymästä ja esittelee sen eri toimintoja.

View-model-ikkunassa on näkymä sovelluksen kokonaisuudesta, sisältäen sekä graafisen käyttöliittymän että kaikki toiminnot saumattomasti. Sovellusta kehittäessä tällä ikkunalla voi testata ja arvioida sovelluksen toiminnallisuutta ja tunnistaa mahdolliset käytön ongelmat.

WPF:n tarjoamia ominaisuuksia hyödyntämällä voidaan varmistaa, että sovellus täyttää kohderyhmän asettamat toiminallisuus- ja laatustandardit. Lisäksi sovelluksen testaaminen mahdollistaa käyttäjäystävällisyyden arvioinnin ja varmistaa, että sovellus on riittävän selkeä kohderyhmän käyttöön.

7.2 Laitteistorajoitteet

WPF-sovellus vaatii tietyt laitteistot optimaalisen suorituskyvyn saavuttamiseksi:

- Hajasaantimuisti eli random-access memory(RAM) tulee olla vähintään 512 MB tai suositus on vähintään 1 GB.
- Kiintolevyn muistitila on oltava vähintään 3GB.
- Prosessorin tehon on oltava joko x86 tai x64. Tämä varmistaa, että tietokoneen prosessori on riittävän tehokas WPF-sovelluksen onnistumiseen käynnistämiseen.
- Renderöinti on yksi tärkeimmistä tekijöistä WPF:n suorituskyvyn määrittämisessä. Mitä enemmän pikseleitä sovelluksessa on, sitä enemmän näytönohjaimen tehoa se vaatii. Eli jos käyttäjän näytönohjin ei ole riittävän tehokas, WPF-sovelluksen käynnistäminen kestää kauemmin, mikä haittaa käytävyyttä.
- WPF-sovellus toimii ainoastaan Windows-tietokoneella, mikä rajoittaa mahdollisuutta käyttää sovellusta muilla vaihtoehtoisilla tietokoneilla, kuten Applen tietokoneilla tai muilla ei-Windows-tietokoneilla.
- Näytön resoluutio on olennainen tekijä WPF-sovelluksen esittämisessä. Korkea resoluutioisilla näytöillä graafiset käyttöliittymäelementit ja teksti eivät välttämättä näy niin tarkasti kuin pitäisi, mikä voi johtaa tyytymättömään käyttökokemukseen.

7.3 Ohjelmistorajoitteet

Käyttäjän tietokoneen käyttöjärjestelmän tulee olla Windows Operating System ja vähintään Windows XP tai tätä uudempi. Muilla Vanhemmilla -Windows käyttöjärjestelmillä ja muilla käyttöjärjestelmillä WPF-sovellusta ei välttämättä käynnisty.

WPF on suunniteltu Windows-käyttöjärjestelmälle, minkä seurauksena kehittäjät eivät pysty kehittämään monialustaista sovellusta eli cross-platform-sovellusta. Tässä tapauksessa kehittäjän on käytettävä muita ohjelmistokieliä kuten Javascript ja HTML5, jotta ne voivat saavuttaa eri alustojen välillä yhteensopivuuden.

Lisäksi WPF:n dokumentaatiot eivät ole ajan tasalla ja vaikka ohjelman peruseräpäteet eivät ole muuttuneet. Vanhentuneen tai puutteellisen dokumentaation vuoksi on haasteellista löytää ratkaisuja kaikenlaisiin ongelmiin sovelluksen kehittämisessä.

WPF-sovelluksen kehittäminen on oltava tietyssä kehitys ympäristössä, joka riippuu hyvin paljon tietokoneelle asennettavasta .NET Frameworkista ja Microsoft Visual Studio -ohjelmasta seuraavasti:

- Microsoft Visual Studio
 - 2010/ 2012/ 2013/ 2015/ 2017/ 2019/ 2022
- .NET Framework
 - 4.0/ 4.5/ 4.5.1/ 4.6 / 4.7 / 4.8
- .NET6.0/ .NET 7.0/ .NET 8.0

7.4 Muut rajoitteet

Projektille ei ole varattu tarpeeksi aikaa toteuttaa kaikkea ryhmän ideoita, koska sovelluksessa on monia tärkeitä osia ja ominaisuuksia, jotka ovat vaikuttavat toisiinsa ja ovat myös toisiinsa yhdistetty. Jos jokin ominaisuus puuttuu, sovellus ei välttämättä noudata sovelluksen ydinideoita.

Tällä moduulilla on myös muita oppiaineita, joihin ryhmäläisten on keskityttävä. Tästä syystä oli vaikeaa saada tämän projektin viikoittaiset tehtävät tehtyä.

Versiohallintajärjestelmänä käytetään Githubia, joka auttaa meitä tekemään yhteistyötä koodauksessa, tunnistamaan, kuka tarkalleen on tehnyt muutoksia, ja myös seuraamaan muutoksiamme. Mutta "pushaaminen" eli muutosten lähettäminen paikallisesta tietokoneesta etävarastoon git bashin kautta ei aina onnistunut.

Visual Studiossa saattaa välillä esiintyä omia luontaisia bugeja, jossa virheet(errors) voivat odottamatta ilmetä kehitys prosessin aikana.

8. Hylätyt ratkaisuvaihtoehdot

Alkuperäisessä suunnittelussa resepteille piti tulla tag-ominaisuus, joka olisi toiminut reseptien haussa yhtenä elementtinä. Tämä kuitenkin osoittautui hankalaksi toteuttaa, joten poistimme sen nykyisestä versiosta, mutta tämän ominaisuuden koodi on vielä koodissa.

Lisää uusi resepti -sivulla oli ongelma, että kaikki reseptin raaka-aineet eivät näkyneet textblockissa. Tarkoituksena oli lisätä niille raaka-aineille erillinen välilehti, mutta aika oli rajoitettu. Tämä ratkaisuvaihtoehto hylättiin ja textblock oli muutettava textbox muotoon, johon voi lisätä scrollaus palkin.

Sovellusta kehittäessä havaittiin ongelman sovelluksen näyttöikkunan koosta. Kun näyttöikkuna venytetään tai pienennetään, kaikki käyttöliittymän elementit liikkuvat ja menevät päällekkäin. Ratkaisu tähän ongelmaan oli responsiivisuuden säätö, mutta aika loppui kesken, joten tämä ratkaisuvaihtoehto hylättiin ja otettiin käyttöön toinen ratkaisuvaihtoehto, jossa näyttöikkuna pysyy yhdenkokoisena, eikä sitä saa venyttää tai pienentää.

9. Jatkokehitysajatuksia

Käyttäjien tyytyväisyyden parantamiseen edellyttää jatkokehitys. Tämä edellytys ei edellytä vain käyttäjien odotusten mukaistamista vaan myös tehokkaiden toimenpiteiden toteuttamista käyttäjien henkilökohtaisen tietojen suojaamiseksi.

Jatkokehitys sisältää seuraavat toimenpiteet:

- Käyttäjä voi luoda järjestelmään oman profiilin, jossa kaikki tiedot ovat tallennettu turvallisesti, mikä suojaa mahdolliselta katoamiselta ja luvattomalta käytöltä.
- Käyttöliittymää voitaisiin parantaa esimerkiksi sijoittamalla painikkeeseen ikoni tai kuvake käytön helpottamiseksi.
- Tagien lisääminen reseptin tallentamisen yhteydessä helpottaa reseptin etsimisessä.
- Mobiilisovelluksen kehitys. Mobiilisovelluksen kehittäminen johtaa laajempaan ja vakaampaan käyttäjäryhmään sekä tarjoaa nopean, helpon ja kätevän pääsyn sovelluksen palveluihin. Käyttäjä voi missä tahansa käyttää sovellusta, mikä edistää käyttömukavuutta.
- Tiedot tallennetaan tällä hetkellä paikallisesti JSON-tiedostoihin, mutta SQL pohjainen tietokanta voisi olla tähän parempi ratkaisu muokkaamisen helppouden ja nopeuden vuoksi. Tietokanta olisi myös luotettavampi ja turvallisempi.
- Käyttäjäpalautetta varten olisi hyvä olla helppo tapa antaa palautetta, jatkokehitysajatuksena olisi palautteen antamis- ja vastaanottojärjestelmä.
- Kielivalinta mahdollisuudet.

Lähteet

Bost, K. (n.d.). Master the Basics of MVVM for Building WPF Applications. Haettu 6.3.2024 osoitteesta <https://intellitect.com/blog/getting-started-model-view-viewmodel-mvvm-pattern-using-windows-presentation-framework-wpf/>

Microsoft. (5.4.2023). A tour of the C# language. <https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>

Microsoft. (6.2.2023). Desktop Guide (WPF .NET) <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/overview/?view=netdesktop-8.0>

Microsoft. (6.2.2023). Optimizing Performance: Taking Advantage of Hardware. <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/advanced/optimizing-performance-taking-advantage-of-hardware?view=netframeworkdesktop-4.8>

Microsoft. (n.d.). NET landing page. Haettu 9.3.2024. <https://dotnet.microsoft.com/en-us/>

Microsoft. (6.2.2023). XAML overview (WPF .NET). <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/xaml/?view=netdesktop-8.0>

Syncfusion Documentation. (21.11.2023) System Requirement in Syncfusion Essential WPF. <https://help.syncfusion.com/wpf/system-requirements>

Thatti, S. (13.11.2023) Advantages and Disadvantages of WPF. <https://www.software-developer-india.com/advantages-and-disadvantages-of-wpf/>

W3Schools. (n.d.). What is JSON? Haettu 9.3.2024 https://www.w3schools.com/whatis/whatis_json.asp