



VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS INSTITUTAS
KOMPIUTERINIO IR DUOMENŲ MODELIAVIMO KATEDRA

Baigiamasis bakalauro darbas

Optimalios dietos parinkimo sistema

Atliko:

Marius Jonikas

parašas

Vadovas:

dr. Joana Katina

Vilnius
2019

Turinys

Santrauka	4
Summary	5
Ivyadas	6
1. Panašių darbų analizė	8
1.1. Android programėlė „HealthyDiet“	8
1.1.1. „HealthyDiet“ palyginimas su šio darbo praktine dalimi	8
1.2. Android programėlė „Lifesum“	9
1.2.1. „Lifesum“ palyginimas su šio darbo praktine dalimi	9
1.3. Internetinė svetainė www.eatthismuch.com	9
1.3.1. Svetainės www.eatthismuch.com palyginimas su šio darbo praktine dalimi	9
1.4. Internetinė svetainė www.kilo.lt	10
1.4.1. Svetainės www.kilo.lt palyginimas su šio darbo praktine dalimi	10
1.5. Internetinė svetainė www.kalorijos.lt	10
1.5.1. Svetainės www.kalorijos.lt palyginimas su šio darbo praktine dalimi	10
1.6. Analizuotų sistemų palyginimas tarpusavyje	11
2. Teorinis sistemos modelis	12
2.1. Sistemos vizija ir funkciniai reikalavimai	12
2.2. UML diagramos	12
2.2.1. Klasių diagrama	12
2.2.2. Panaudos atvejų diagrama	13
2.2.3. Duomenų bazės schema	14
3. Praktinis sistemos įgyvendinimas	16
3.1. Naudojami įrankiai ir technologijos	16
3.1.1. Java programavimo kalba	16
3.1.2. NetBeans platforma	16
3.1.3. Swing karkasas	16
3.1.4. SwingX plėtinys	16
3.1.5. AWT karkasas	16
3.1.6. JFreeChart biblioteka	17
3.1.7. SQLite duomenų bazė	17
3.2. Programinės darbo dalies struktūra	17
3.3. Grafinės vartotojo sąsajos kūrimas	17
3.4. Dietos sudarymas	19
3.4.1. Automatinis dietos generavimas	20
3.4.2. Savarankiškas dietos generavimas	24
3.4.3. Automatinio ir savarankiško dietų generavimo palyginimas	25
3.5. Kitos sistemos funkcijos	25
3.5.1. Išsaugotos dietos	25
3.5.2. Dietų kalendorius	25
3.5.3. Maisto produktų duomenų bazė	26

3.5.4. Svorio kitimo istorijos išsaugojimas	26
3.5.5. Kūno masės indekso skaičiavimas	26
Išvados ir rekomendacijos	27
Ateities tyrimų planas	28
Literatūros šaltiniai	29
Priedai	30
A. Optimalios dietos parinkimo sistemos pagrindinis langas	30
B. Automatiškai sugeneruota dieta	31
C. Savarankiškas dietos generavimas	32
D. Išsaugotų dietų peržiūros langas	33
E. Dietų kalendorius	34
F. Nepageidaujamų maisto produktų kategorijų atsisakymas	35
G. Konkrečių nepageidaujamų maisto produktų atsisakymas	36
H. Maisto produktų duomenų bazės peržiūros ir valdymo langas	37
I. Naujo maisto produkto įtraukimo į duomenų bazę langas	38
J. Išsaugoti anksčiau naudotojo įvesti duomenys	39
K. Svorio kitimo grafikas	40
L. Kūno masės indekso skaičiavimas	41

Santrauka

Šio darbo tikslas - sukurti optimalios dietos parinkimo programą, pritaikytą asmeniniam kompiuteriui. Tikslui pasiekti buvo nagrinėjamos panašaus tikslo siekiančios sistemos, naudojama informacija iš įvairių šaltinių, užtrikrinant dietų kūrime naudojamų formulių tikslumą. Dietos sudaromos individualiai, atsižvelgiant į naudotojo duomenis, iš lanksčioje duomenų bazėje saugomų maisto produktų. Naudotojui leidžiama lengvai ištrinti duomenų bazėje esančius produktus bei įtraukti naujus. Valgiaraštį galima generuoti automatiškai, pasirenkant nepageidaujamus produktus, valgių per dieną skaičių bei kitą informaciją, arba savarankiškai, viską renkantis pačiam, stebint nuolat atnaujinamas rekomendacijas dėl kilokalorijų bei maistinių medžiagų dietoje trūkumo arba pertekliaus. Sukurta sistema turi ir daugiau funkcijų: dietų išsaugojimą, peržiūrėjimą ir ištrynimą, mitybos kalendorių, svorio pokyčių istorijos išsaugojimą, grafikų braižymą ir kt.

Summary

Building an Optimal Diet System

The main goal of this work is to create a desktop application for a personal computer which would help user to achieve an optimal diet fulfilling caloric intake as well as nutritional needs. Nutritional needs include distribution of proteins, carbohydrates and fats, as well as recommended daily intake of calcium, cholesterol and fiber. Recommended norms are based on user characteristics and goals. To achieve this, systems with similar functionality were analyzed and information from various sources was used, thus ensuring precision of formulas applied while generating a diet. Diet is generated from products stored in a flexible database either automatically or user can create a diet plan by oneself using provided recommendations and a special graphical user interface. If diet is generated automatically, user chooses a time period for a diet (1 day, 3 days or a week) as well as a number of meals per day (3 or 5). Once diet creation is completed, user can save it and/or include it into a diet calendar. In addition to a dietary plan selection and diet calendar, another functions provided by this application include a flexible food database, where user can search for products, sort them based on a various criteria, delete them or include new ones, possibility to save weight change history (of the last year) and draw line charts based on it, also a Body Mass Index calculator.

Iyadas

Kiekvienas gyvas organizmas, tiesiogine to žodžio prasme, yra sudarytas iš to, ką valgo. Dėl šios priežasties, mityba yra vienas iš svarbiausių gyvenimo aspektų, smarkiai įtakančių gyvenimo kokybę ir trukmę. Tiesa, modernioje visuomenėje, kurioje visi skuba atlikti kuo įmanoma daugiau darbų, vartojamo maisto kokybė dažnai yra pamirštama. Žmonės valgo bet ką, ką randa parduotuvėje, ypač, jei maisto produktas nereikalauja jokio paruošimo, nepaisant to, kad tokie produktai dažnai ir būna mažiausiai naudingi organizmui. Ypač klesti greitojo maisto restoranai, nes žmonės neturi laiko gamintis maistą patys, sveikesnį maistą gaminančiuose restoranuose reikia ilgiau laukti, taigi, prarandama laiko, be to, greitojo maisto restoranuose ir kainos yra mažesnės. To pasėkmė - nutukimas ir diabetas yra didesnės problemos, nei bet kada iki šiol, o dažniausia mirties priežastis pasaulyje (ir dar dažnesnė išsivysčiusiose šalyse) yra širdies bei kraujagyslių ligos, labiausiai siejamos su netinkama mityba ir per dideliu kūno svoriu. Dėl šių priežasčių šiame darbe kuriama aplikacija, kuria naudojantis žmonėms bus lengviau susidaryti optimalią dietą, gyventi ilgesnį bei kokybiškesnį gyvenimą.

Panašių sistemų netrūksta. Tiesa, dalis jų yra orientuotos į vartojamų kalorijų skaičių, neatsižvelgiant arba nepilnai atsižvelgiant į pasirinkto maisto sudėtį. Daugybė sistemų yra mokamos arba iš dalies mokamos. Taip pat nėra nei vienos viešai prieinamos mobiliosios arba asmeniniam kompiuteriui skirtos dietos sudarymo programėlės lietuvių kalba - yra tik keletas internetinių puslapių, kuriuose būtina registracija ir dažnai dalis funkcijų yra mokamos.

Šio darbo tikslas - sukurti sistemą, padedančią naudotojui susikurti asmeninę optimalią dietą ir skatinančią tinkamus mitybos įpročius.

Uždaviniai:

1. Išanalizuoti rinkoje esančius, panašaus tikslo siekiančius produktus.
2. Išnagrinėti dietos sudarymą analizuojančius šaltinius ir formules.
3. Sukurti teorinę sistemos modelį.
4. Pritaikyti teorinę informaciją praktiškai, sukuriant Java aplikaciją, pritaikytą asmeniniam kompiuteriui, nepriklausomai nuo to kompiuterio operacinės sistemos.
5. Darbo procesą dokumentuoti šiame rašto darbe.

Darbo metu sukurta sveikesnei mitybai pasiekti naudinga sistema. Naudotojui siūlomos įvairios funkcijos, viena iš svarbiausių - automatizuotas optimalios dietos sudarymas, atsižvelgiant į asmeninius naudotojo duomenis. Naudotojui pateikiamas optimalus kilokalorijų skaičius, baltymų, angliavandenių, riebalų, kalcio ir skaidulinių medžiagų kiekiai, kuriuos reikėtų suvartoti per parą ir pavyzdinė dieta, sudaryta iš duomenų bazėje esančių produktų, naudojantis teorinėje darbo dalyje nagrinėjamomis formulėmis, kurios yra orientuotos ne tik į suvartojamų kalorijų skaičių, bet ir į maistinių medžiagų pasiskirstymą. Naudotojas gali rinktis, kiek kartų per dieną nori valgyti (3 arba 5), kokiam laikotarpiui dieta turėtų būti generuojama (1 dienai, 3 dienom arba savaitei), gali ignoruoti tam tikrus dietos parinkimo kriterijus, atsisakyti nepageidaujamų produktų kategorijų arba konkrečių produktų. Ne ką mažiau svarbi sistemos funkcija yra savarankiškas dietos sudarymas, kuomet naudotojas pats į valgiaraštį įtraukia maisto produktus, laisvai pasirinkdamas jų kiekius, arba pašalina iš valgiaraščio jau įtrauktus produktus. Po kiekvieno produkto įtraukimo arba pašalinimo yra perskaičiuojami ir naudotojui pateikiami kilokalorijų bei maistinių medžiagų dietoje trūkumai arba pertekliai. Automatiškai arba savarankiškai sugeneruotas dietas galima išsaugoti

ir/arba priskirti konkrečioms dienoms kalendoriuje. Įsijungęs kalendoriaus langą sistemoje, naudotojas gali peržiūrėti anksčiau įtrauktas dietas, atlaisvinti kai kurias dienas (arba visą mėnesį iš karto) arba priskirti laisvoms dienoms dietas iš išsaugotų dietų sąrašo. Kita svarbi sistemos dalis - maisto produktų duomenų bazė. Maisto produktų duomenų bazės lange lentelės pavidalu yra pateikiami visi išsaugoti produktai bei informacija apie juos. Produktus galima rūšiuoti abėcėlės tvarka arba pagal bet kokį kitą kriterijų didėjimo/mažėjimo tvarka. Taip pat galima nesudėtingai produktus iš duomenų bazės ištrinti arba įtraukti į ją naujus. Sistema naudotojui suteikia ir galimybę sekti savo paskutinių metų laikotarpio svorio pokyčių istoriją bei atvaizduoti pokyčius grafiškai. Dar viena, mažiau reikšminga sistemos funkcija - kūno masės indekso skaičiuoklė.

1. Panašių darbų analizė

Prieš kuriant naują sistemą, būtina išanalizuoti rinkoje esančius, tam pačiam tikslui skirtus produktus. Būtina įvertinti, ar kuriama sistema turės bent vieną pranašumą prieš jau egzistuojančias ir nebus per daug į jas panaši. Kitu atveju, darbas gali būti beprasmiškas. Šiuo atveju analizuojamos penkios sistemos, turinčios panašius tikslus: Android programėlės „HealthyDiet“ ir „Lifesum“ bei internetinės svetainės www.eatthismuch.com, www.kilo.lt ir www.kalorijos.lt. „Lifesum“ yra pati populiariausia dietos sudarymo programa Google Play parduotuvėje, parsisiųsta dešimtis milijonų kartų. „HealthyDiet“ parsisiuntimų skaičius yra gerokai mažesnis, tačiau ši programa analizuojama, nes yra paprastesnė, t.y. panašesnė į šiame darbe implementuojamą sistemą bei jos tikslas yra labai panašus į šio darbo temą. Internetinės svetainės, žinoma, skiriasi nuo Android programėlių. Priešingai nei kitos keturios nagrinėjamos panašios sistemos, www.eatthismuch.com sudaro naudotojams dietą automatiškai. Svetainė www.kilo.lt pasirinkta todėl, jog yra vienas labiausiai išbaigtų dietą sudarančių įrankių lietuvių kalba. Ji nuo Android programėlių skiriasi ne tik savo kalba ir platforma, tačiau ir dietos sudarymo metodu - vietoje matematinių skaičiavimų yra naudojami žmogiškieji ištekliai. Kita lietuviška interneto svetainė, nagrinėjama šiame darbe, yra www.kalorijos.lt. Pagrindinė šios sistemos funkcija - kilokalorijų ir maistinių medžiagų (baltymų, angliavandenių ir riebalų) skaičiuoklė.

1.1. Android programėlė „HealthyDiet“

(Nuoroda į sistemą)

Programos tikslas - padėti naudotojui sveikai maitintis bei pasiekti norimą fizinę formą.

Programos funkcijos:

1. Dietos sudarymas ir patarimai kaip numesti svorio ir/arba sveikiau maitintis.
2. Kalorijų ir maistinių medžiagų skaičiuoklė.
3. Patiekalų planavimas, sveikų patiekalų receptai.
4. Didelė maisto patiekalų duomenų bazė.
5. Asmeninis kalorijų suvartojimo planas, atsižvelgiant į naudotojo duomenis.
6. Svorio pokyčių statistika.
7. Vandens suvartojimo statistika.
8. Nueitų žingsnių skaičiuoklė.

1.1.1. „HealthyDiet“ palyginimas su šio darbo praktine dalimi

Programėlė „HealthyDiet“ yra išbaigta, pasižymi ne tik funkcionalumu, bet ir dizainu. Į tokio lygio programų kūrimą yra investuojama daug pinigų ir laiko, todėl nieko keisto, kad programa lenkia šio darbo metu kuriamą aplikaciją pagal dizainą ir nemažą dalį funkcionalumo. Tiesa, galima rasti ir „HealthyDiet“ programos trūkumų, lyginant su šio darbo praktine dalimi:

1. Nėra automatinio dietos generavimo, tik savarankiškas produktų įtraukimas į dietą.
2. Programoje naudojama tik anglų kalba, todėl ne visi žmonės gali ja naudotis. Šiame darbe kuriama aplikacija naudoja lietuvių kalbą ir gali būti geresnis pasirinkimas lietuviams, neturintiems gerų anglų kalbos žinių.
3. Dalis funkcijų yra mokamos. Ne kiekvienas žmogus gali arba nori mokėti pinigus už pagalbą sudarant dietos planą, todėl mažiau išbaigto dizaino, tačiau nemokama programa gali būti tinkamas pasirinkimas.

1.2. Android programėlė „Lifesum“

(Nuoroda į sistemą)

Programos tikslas - atrasti tinkamiausią dietą, atsižvelgiant į naudotojo fizinę formą ir sveikatą.

Programos funkcijos:

1. Dietos plano sudarymas, patarimai, kaip sveikai maitintis ir numesti svorio.
2. Kalorijų skaičiuoklė, įvedant maisto produkto duomenis arba nuskenuojant brūkšninį kodą.
3. Kalorijų ir maistinių medžiagų suvartojimo statistika.
4. Sveiko maisto receptai.
5. Pagalba renkantis maisto produktus.

1.2.1. „Lifesum“ palyginimas su šio darbo praktine dalimi

Ši programėlė yra panaši į šiame darbe jau analizuotą „HealthyDiet“, taigi, programos tikslas taip pat panašus į šio darbo praktinės dalies tikslą. Dizainas ir funkcionalumas yra puikiai įgyvendinti, dėl to programa parsisiųsta dešimtis milijonų kartų. Visgi, dauguma sistemų turi trūkumų, net ir lyginant su paprastesnėmis sistemomis, tokiomis, kaip šiame darbe kuriama programa. „Lifesum“ trūkumai:

1. Nėra automatinio dietos generavimo.
2. Dalis funkcijų yra mokamos.
3. Reikalinga registracija.
4. Nėra išversta į lietuvių kalbą.

1.3. Internetinė svetainė www.eatthismuch.com

(Nuoroda į sistemą)

Svetainės tikslas - automatiškai generuoti sveikos mitybos planus.

Svetainės funkcijos:

1. Automatinis dietos generavimas.
2. Pagalba nusistatant mitybos tikslą.
3. Sveikos mitybos receptų pateikimas.
4. Reguliarus elektroninių laiškų su receptais ir priminimais siuntimas naudotojams.
5. Svorio pokyčių ir mitybos istorijos saugojimas.

1.3.1. Svetainės www.eatthismuch.com palyginimas su šio darbo praktine dalimi

Sistemos yra skirtingos - šiame darbe kuriama asmeniniam kompiuteriui tinkama aplikacija, o www.eatthismuch.com yra internetinė svetainė. Kaip rašoma svetainėje, www.eatthismuch.com yra 2011 metais pradėtas projektas, kurį vysto šeši profesionalūs programuotojai, o naudotojų skaičius siekia šimtus tūkstančių, todėl nieko keisto, jog svetainė pasižymi išbaigtu dizainu bei pažangiais automatinio dietų generavimo algoritmais. Visgi, tobulų sistemų nebūna, todėl galima išvelgti ir www.eatthismuch.com trūkumų, lyginant su šio darbo praktine dalimi:

1. Priešingai negu šiame darbe kuriamoje sistemoje, svetainėje nėra savarankiško dietos sudarymo funkcionalumo.
2. Dauguma svetainės funkcijų yra prieinamos tik registruotiems naudotojams.

3. Nors svetainės duomenų bazė yra didelė bei detali, ji nėra lanksti, priešingai nei šio darbo duomenų bazė, iš kurios naudotojas gali laisvai išimti produktus bei įdėti į ją naujus.
4. Svetainė nėra išversta į lietuvių kalbą.

1.4. Internetinė svetainė www.kilo.lt

(Nuoroda į sistemą)

Svetainės tikslas - padėti naudotojams pasiekti norimą kūno svorį ir išvaizdą.

Svetainės funkcijos:

1. Dietologo konsultacijos.
2. Valgiaraščio sudarymas.
3. Trenerio konsultacijos.
4. Treniruočių grafiko sudarymas.
5. Forumas diskusijoms su žmonėmis, siekiančiais panašių tikslų.

1.4.1. Svetainės www.kilo.lt palyginimas su šio darbo praktine dalimi

Sistemų tikslai yra tik iš dalies panašūs - šiame darbe siekiama sukurti optimalios dietos sudarymo programą, padėti naudotojams sveikai maitintis ir/arba numesti/priaugti svorio, tuo tarpu www.kilo.lt yra orientuota ne tik į mitybą, bet ir į sportą. Taip pat svetainė suteikia naudotojams forumą, tai yra, bendruomenę, tačiau kaip didžiausią skirtumą galima įvardyti pačio tikslo realizavimo būdą - šiame darbe siekiama sudaryti optimalią dietą, remiantis matematinėmis, moksliniais patikrintomis formulėmis, tuo tarpu www.kilo.lt naudoja žmogiškuosius išteklius, kiekvienam naudotojui asmeniškai dietą ir treniruočių programą sudaro specialistai. Taigi, www.kilo.lt turi trūkumų, lyginant su šio darbo praktine dalimi:

1. Svetainės paslaugos yra mokamos.
2. Kadangi svetainėje dietą sudaro specialistai individualiai kiekvienam naudotojui, tai užtrunka šiek tiek laiko, tuo tarpu šiame darbe kuriama programėlė dietą sudaro iš karto.
3. Šiame darbe kuriama programėlė yra ne tik nemokama, bet ir nereikalauja jokios registracijos ar prisijungimo, priešingai negu www.kilo.lt.

1.5. Internetinė svetainė www.kalorijos.lt

(Nuoroda į sistemą)

Svetainės tikslas - padėti naudotojui savarankiškai susidaryti dietą.

Svetainės funkcijos:

1. Kilokalorijų ir maistinių medžiagų skaičiuoklė.
2. Savarankiškas dietos plano sudarymas.
3. Sudarytų dietų išsaugojimas ir redagavimas.

1.5.1. Svetainės www.kalorijos.lt palyginimas su šio darbo praktine dalimi

Nors šio darbo praktinė dalis yra asmeniniam kompiuteriui skirta programėlė, o www.kalorijos.lt yra interneto svetainė, sistemos turi gana panašaus funkcionalumo. Pagrindinė svetainės funkcija - kilokalorijų ir maistinių medžiagų skaičiuoklė, įtraukiant maisto produktus, yra panaši į šio

darbo praktinėje dalyje įgyvendintos sistemos savarankiško dietos sudarymo funkcionalumą. Taip pat svetainės www.kalorijos.lt kilokalorijų ir maistinių medžiagų skaičiuoklė leidžia rūšiuoti produktus pagal tam tikrus kriterijus, panašiai, kaip šiame darbe sukurtos programos duomenų bazėje saugomų produktų peržiūros lange. Dar du panašumai - sudarytų dietų išsaugojimas ir kalorijų poreikio apskaičiavimas pagal asmeninius duomenis. Žinoma, sistemos turi ir skirtumų. Kaip jau buvo minėta, sistemos įgyvendintos skirtingose platformose. Dauguma www.kalorijos.lt funkcijų yra prieinamos tik prisijungusiems naudotojams, o šio darbo praktinėje dalyje įgyvendintoje sistemoje naudotojų registracija ir prisijungimas neegzistuoja. Svetainėje naudojama plati duomenų bazė, tačiau naudotojas neturi galimybės į ją įdėti arba iš jos išimti produktų. Šio darbo metu sukurtoje programoje naudotojui pateikiama nedidelė standartinė duomenų bazė, tačiau suteikiamas funkcionalumas nesudėtingai ištrinti joje saugomus produktus arba įtraukti naujus. Apibendrinant, galima įvardyti šiuos svetainės www.kalorijos.lt trūkumus:

1. Nėra automatinio dietos generavimo.
2. Nelanksti duomenų bazė.
3. Dauguma funkcijų prieinamos tik registruotiems naudotojams.

1.6. Analizuotų sistemų palyginimas tarpusavyje

Analizės metu buvo nagrinėjamos penkios sistemos, siekiančios padėti naudotojams pasiekti/palaikyti norimą kūno svorį bei sveikatą. Analizuotų sistemų privalumai ir trūkumai yra apibendrintai pateikti (1. lentelė) lentelėje.

	HealthyDiet	Lifesum	www.eatthismuch.com	www.kilo.lt	www.kalorijos.lt
Automatinis dietos generavimas	-	-	+	-	-
Savarankiškas dietos generavimas	+	+	-	-	+
Kalorijų poreikio skaičiuoklė	+	+	+	-	+
Svorio pokyčių istorija	+	+	+	+	-
Kūno masės indekso skaičiavimas	+	+	+	+	+
Sporto treniruočių sudarymas	-	-	-	+	-
Specialistų konsultacijos	-	-	-	+	-
Forumai	-	-	-	+	-

1 lentelė. Panašių sistemų palyginimas tarpusavyje.

2. Teorinis sistemos modelis

2.1. Sistemos vizija ir funkciniai reikalavimai

Optimalios dietos parinkimo sistema yra nekomercinė, paprasta naudoti, skirta automatiniam arba savarankiškam mitybos plano sudarymui, atsižvelgiant į individualius duomenis ir panaudojant mokslškai patikrintas formules. Sistema leidžia naudotojui pasirinkti norimą tikslą - numesti svorio, palaikyti esamą svorį arba priaugti svorio, atsisakyti nepageidaujamų maisto produktų arba produktų kategorijų, pavyzdžiui, mėsos, pieno ir kitų produktų, išskyrus vaisius bei daržoves, kurie šiame darbe yra laikomi neatsiejama sveikos dietos dalimi, taip pat, jei dieta generuojama automatiškai, galima pasirinkti, kiek kartų per dieną planuojama valgyti ir kokiam laikotarpiui dietą norima sudaryti. Programa naudotojui suteikia ir kitas funkcijas: kūno masės indekso skaičiuoklę, maisto produktų duomenų bazę, kurioje galima peržiūrėti produktų maistines sudėtis ir surūšiuoti produktus pagal skirtingus kriterijus, ištrinti produktus iš duomenų bazės arba įtraukti naujus, sudarytų dietų išsaugojimą ir trynimą, dietų kalendorių ir svorio pokyčių istorijos išsaugojimą. Sistemai keliami šie funkciniai reikalavimai:

1. Naudotojo įvestų fizinių duomenų apdorojimas.
2. Automatinis visavertės dietos generavimas.
3. Pagalba naudotojui susidaryti dietą pačiam.
4. Nepageidaujamų valgyti produktų neįtraukimas į generuojamą dietą.
5. Dietos tikslo pasirinkimas.
6. Anksčiau naudotų duomenų išsaugojimas ir trynimas.
7. Kūno masės indekso skaičiuoklė.
8. Duomenų bazės peržiūrėjimas, produktų rūšiavimas pagal skirtingus kriterijus, produktų įtraukimas į duomenų bazę ir ištrynimas iš jos.
9. Sudarytų dietų išsaugojimas, peržiūrėjimas bei ištrynimas.
10. Dietų kalendorius, dietų įtraukimas į jį ir išėmimas iš jo.
11. Galimybė išsaugoti svorio pokyčių istoriją ir nubrėžti grafiką.

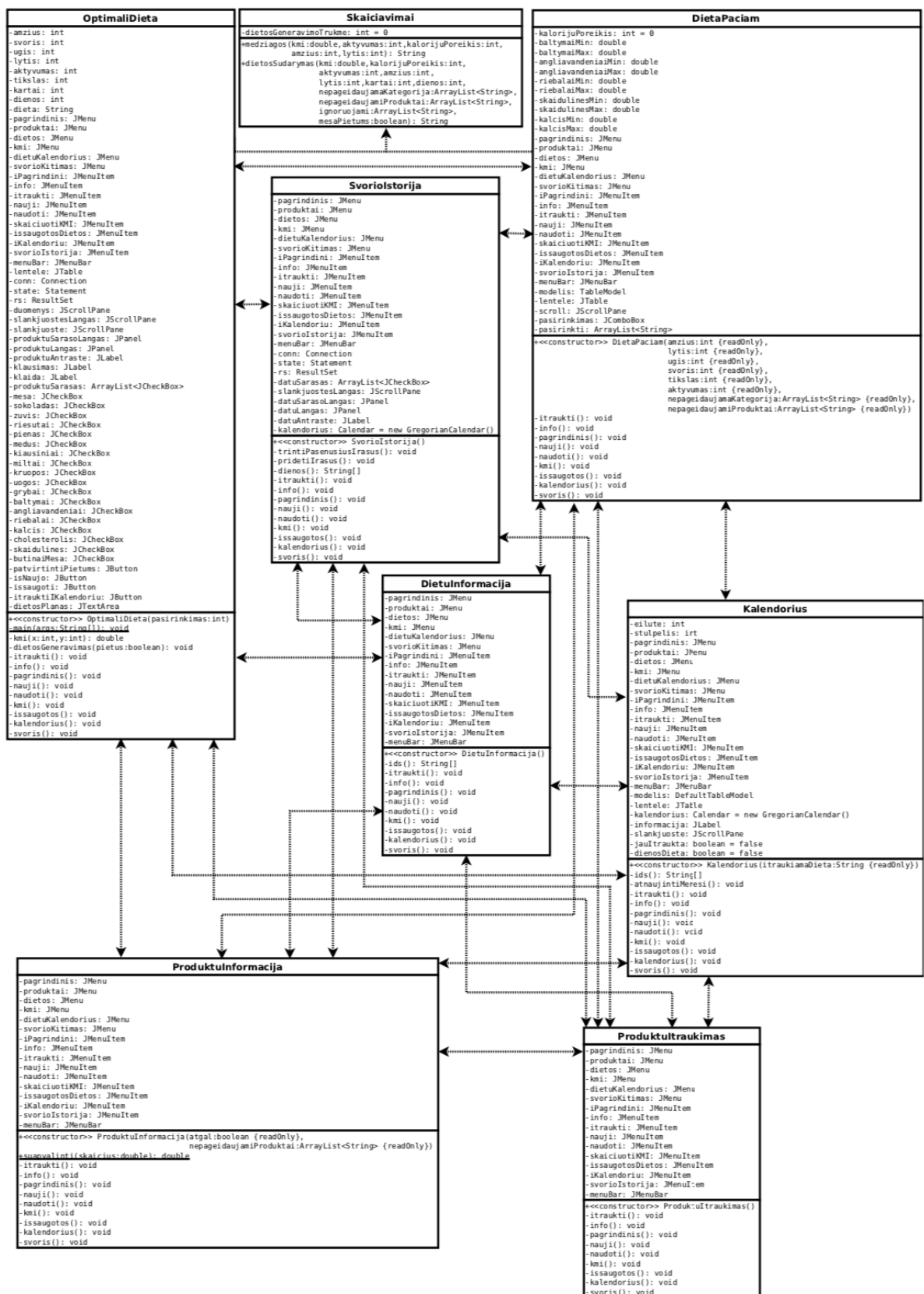
Nefunkciniai sistemos reikalavimai:

1. Sklandus veikimas, sistema neturėtų strigti.
2. Sistema turėtų veikti bet kokioje asmeniniams kompiuteriams skirtoje operacinėje sistemoje.

2.2. UML diagramos

2.2.1. Klasių diagrama

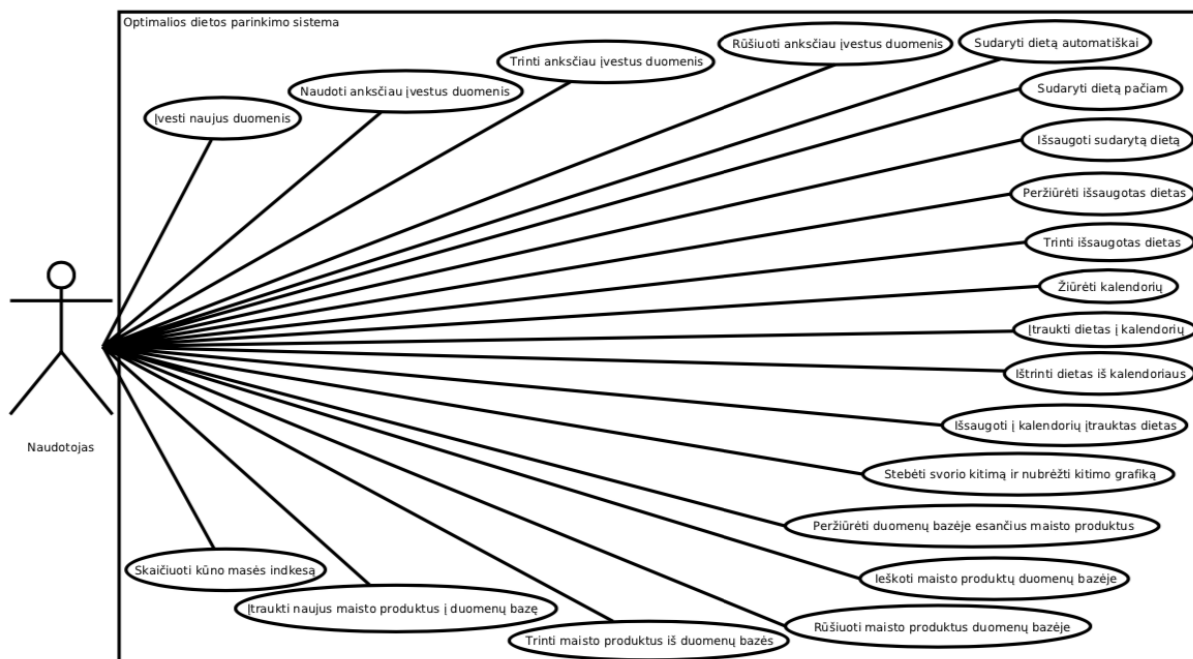
Iliustracijoje (žr. 1 pav.) pateikta sistemos klasių diagrama. Klasės *OptimaliDieta*, *DietaPaciam*, *DietuInformacija*, *Kalendorius*, *ProduktuInformacija*, *ProduktuItraukimas* ir *SvorioIstorija* visos turi abipuses priklausomybes tarpusavyje. Abipusė priklausomybė reiškia, jog tokį ryšį turinčios klasės abi gali sukurti viena kitos objektą. Šios klasės turi tokį ryšį, nes jos visos sukuria grafines naudotojo sąsajas, į kurias, naudotojo patogumui, yra įtrauktas meniu, kurį panaudojant galima iš karto patekti iš vieno lango į bet kurį kitą. Klasės *OptimaliDieta* ir *DietaPaciam* papildomai turi vienpuses priklausomybes nuo klasės *Skaiciavimai*. Vienpusė priklausomybė reiškia, jog *OptimaliDieta* ir *DietaPaciam* klasės sukuria *Skaiciavimai* klasės objektą, tačiau *Skaiciavimai* klasėje jokių kitų klasių objektai nėra kuriami. Ši klasė yra vienintelė, kuri nesuteikia grafinės naudotojo sąsajos ir yra naudojama tik dietos generavimo metu.



1 pav. Sistemos klasių diagrama

2.2.2. Panaudos atvejų diagrama

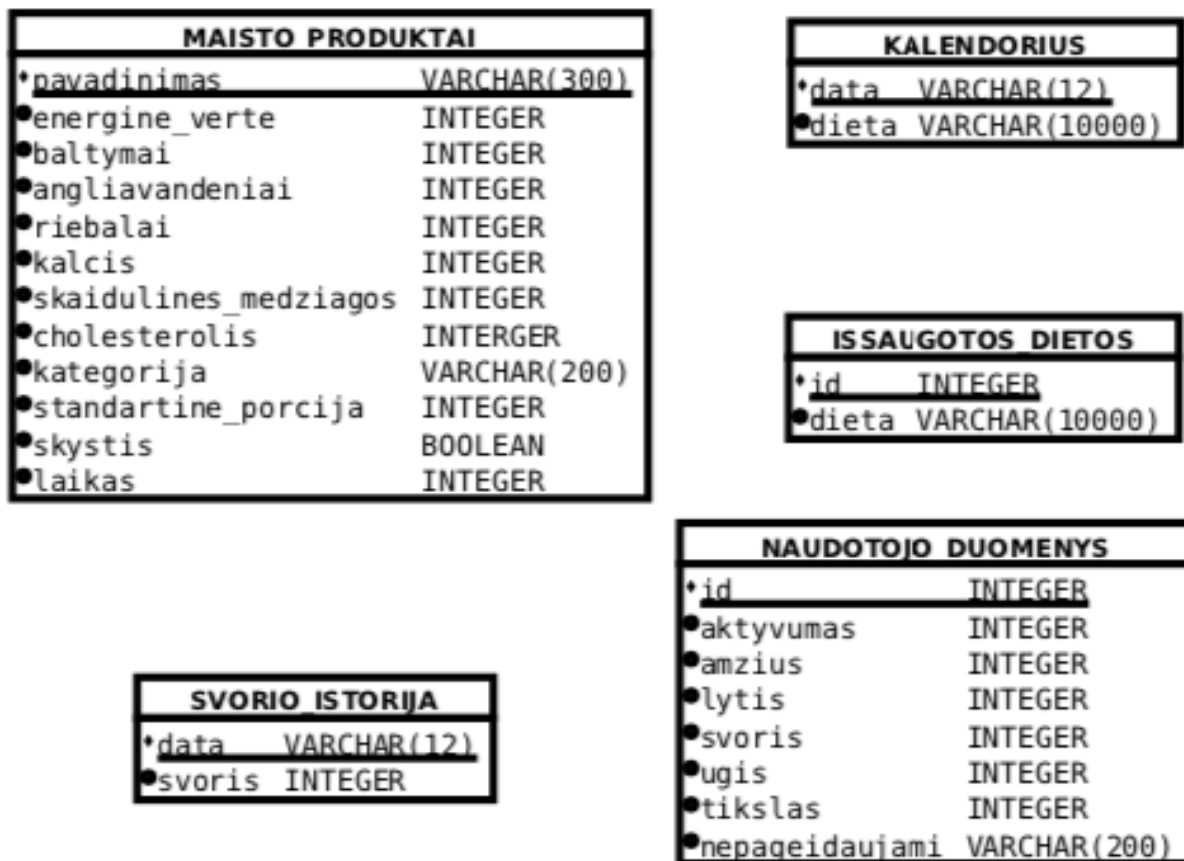
Diagrama (žr. 2 pav.) vaizduoja galimus sistemos panaudojimo atvejus. Panaudos atvejų diagramoje aiškiai matosi, jog sistemą naudoja tik vieno tipo aktorius - naudotojas.



2 pav. Sistemos panaudos atvejų diagrama

2.2.3. Duomenų bazės schema

Duomenų bazės įgyvendinimui pasirinkta SQLite reliacinė duomenų bazių valdymo sistema. Duomenų bazėje saugomos 5 lentelės: *MAISTO_PRODUKTAI*, *NAUDOTOJO_DUOMENYS*, *ISSAUGOTOS_DIETOS*, *KALENDORIUS* ir *SVORIO_ISTORIJA*. Visos lentelės yra dinamiškos, naudotojas gali iš jų įrašus ištrinti arba įtraukti į jas naujus. *MAISTO_PRODUKTAI* yra, ko gero, pati svarbiausia lentelė, joje saugoma visų maisto produktų informacija, naudojama dietos sudarymui. *NAUDOTOJO_DUOMENYS* lentelėje yra išsaugomi prieš dietos generavimą sistemos naudotojo įvesti duomenys tam, kad kitą kartą nereikėtų informacijos įvedinėti rankiniu būdu iš naujo. Kadangi kai kurie duomenys, pavyzdžiui, svoris ar nepageidaujami produktai, gali keistis, naudotojo įvestų įrašų skaičius nėra ribojamas, nebeaktualius įrašus galima nesudėtingai ištrinti. Lentelė *ISSAUGOTOS_DIETOS* yra sugeneruotų ir naudotojo pasirinktų išsaugotų dietų saugykla. Šioje lentelėje dietos yra numeruojamos pagal tai, kuri buvo išsaugota anksčiau, naudotojoas dietas gali peržiūrėti, ištrinti arba įtraukti į dietų kalendorių. Dietų kalendoriaus informacija yra saugoma lentelėje *KALENDORIUS*. Šioje lentelėje, kaip ir *ISSAUGOTOS_DIETOS* lentelėje, yra saugomos sugeneruotos dietos, o pirminis šios lentelės raktas yra dienos, kuriai dieta buvo priskirta, data. Dar viena lentelė, *SVORIO_ISTORIJA*, taip pat turi pirminį raktą, kuris yra data. Kiekvienoje šios lentelės eilutėje yra saugoma vienos dienos data ir skaitinė vertė, reprezentuojanti naudotojo kūno svorį tą dieną. Duomenų bazės lentelės yra pavaizduotos (žr. 3 pav.) iliustracijoje.



3 pav. Sistemos duomenų bazės schema

Schemoje aiškiai matosi, jog visos lentelės turi pirminius raktus, tačiau neturi išorinių raktų, o neturint išorinių raktų, neegzistuoja ryšiai, kuriuos būtų galima atvaizduoti duomenų bazės schemoje. Tai nereiškia, jog lentelės nėra susijusios. Pavyzdžiui, jau buvo minėta, jog išsaugotas dietas galima įtraukti į kalendorių, t.y. iš *ISSAUGOTOS_DIETOS* lentelės galima nukopijuoti ir į *KALENDORIUS* lentelę įtraukti dietos tekstą. Lygiai taip pat ir į kalendorių įtrauktas dietas galima nukopijuoti į išsaugotų dietų sąrašą, jeigu tame sąrašė dar nėra identiškos dietos. Tuo tarpu pačios dietos yra generuojamos iš maisto produktų, panaudojant naudotojo duomenis, t.y. informaciją iš *MAISTO_PRODUKTAI* ir *NAUDOTOJO_DUOMENYS* lentelių. Visi šie loginiai ryšiai yra aiškiai apibrėžti programiniame kode ir jaučiami naudojant sistemą.

Pirminiai raktai sistemoje panaudoti tam, jog būtų užtikrintas duomenų vientisumas ir sklandus programos veikimas. Pavyzdžiui, du produktai negali turėti identiškų pavadinimų, nes tai sukeltų neaiškumų generuojant dietą, kalendoriuje viena dieta negali būti priskirta daugiau negu vienai dienai ir t.t. Išorinių raktų nuspręsta nenaudoti, nes jų integravimas ne tik nepadėtų numatytam sistemos funkcionalumui, tačiau netgi sukurtų papildomų nereikalingų apribojimų.

3. Praktinis sistemos įgyvendinimas

3.1. Naudojami įrankiai ir technologijos

Optimalios dietos parinkimo sistema buvo kuriama NetBeans integruotoje programavimo aplinkoje, naudojant Java programavimo kalbą. Naudotojo duomenų įvesčiai ir išvesčiai naudojama grafinė naudotojo sąsaja, įgyvendinta panaudojant Swing ir AWT karkasus, savarankiško dietos sudarymo palengvinimui integruojant automatinio užpildymo funkciją panaudotas Swing karkaso plėtinys SwingX, svorio grafikų braižymui papildomai panaudota JFreeChart biblioteka. Maisto produktų informacija, naudotojo įvesti duomenys, išsaugotos dietos bei kita informacija yra saugoma SQLite integruotoje duomenų bazėje.

3.1.1. Java programavimo kalba

Java yra bendrosios paskirties, objektiškai orientuota programavimo kalba. Vienas iš pagrindinių šios programavimo kalbos privalumų ir priežastis, kodėl ji buvo pasirinkta šiam darbui - nepriklausomybė nuo kompiuterio architektūros. Java aplikacijos yra kompiliuojamos į baitkodą, kuris gali būti paleidžiamas Java virtualiojoje mašinoje, instaliuotoje į bet kokią įrenginį.

3.1.2. NetBeans platforma

Platforma NetBeans pasirinkta todėl, nes ji yra inovatyvi, nemokama ir patogi naudoti kuriant Java aplikacijas. Kaip rašoma NetBeans internetinėje svetainėje [8], ši platforma dažniausiai yra įvardijama kaip nemokama Java integruota programavimo aplinka. Tiesa, ši atviro kodo platforma gali būti naudojama ir kitoms programavimo kalboms bei karkasams, tačiau šis darbas buvo vykdomas būtent pačia populiariausia tarp įrankio naudotojų Java programavimo kalba.

3.1.3. Swing karkasas

Swing karkasas yra programavimo sąsaja, skirta suteikti grafinę naudotojo sąsają Java aplikacijoms. Šiuo karkasu sukurta grafinė naudotojo sąsaja veikia vienodai skirtinguose įrenginiuose, nepriklausomai nuo jų operacinės sistemos. Dėl šios priežasties dauguma grafinių elementų (mygtukų, lentelių ir kt.) buvo sukurta panaudojant Swing karkasą.

3.1.4. SwingX plėtinys

Šis plėtinys yra sukurtas palengvinti Java Swing aplikacijų funkcionalumo įgyvendinimą. Šiame darbe plėtinys yra panaudotas tik vieną kartą, savarankiško dietos sudarymo lange, suteikiant naudotojui galimybę pasinaudoti automatinio užpildymo funkcija, tokiu būdu greičiau pasirenkant maisto produktų pavadinimus iš išskleidžiamąjo sąrašo.

3.1.5. AWT karkasas

AWT (Abstract Window Toolkit), kaip ir Swing, yra karkasas, padedantis sukurti grafinę naudotojo sąsają Java aplikacijoms. Šis karkasas yra senesnis, parūpina „sunkesnius“ grafinius elementus negu Swing karkasas, ne visada yra nepriklausomas nuo operacinės sistemos, dėl to grafiniai elementai šiame darbe šiuo karkasu nėra kuriami. Visgi, yra funkcionalumo, kurio įgyven-

dinti vien Swing karkaso neužtenka, pavyzdžiui, pakeisti fono spalvą ar suprogramuoti mygtuko paspaudimo įvykį.

3.1.6. JFreeChart biblioteka

Kaip rašoma JFreeChart internetinėje svetainėje [4], ši Java biblioteka yra nemokama ir padeda programuotojams integruoti grafikus į jų kuriamas aplikacijas. JFreeChart pasirinkta šiam darbui, nes yra lanksti ir paprasta naudoti, grafikus galima nubrėžti panaudojant kur kas mažiau kodo, nei reikėtų naudojantis kitomis bibliotekomis arba ypač nenaudojant jokių papildomų bibliotekų, tik Swing ir AWT karkasus.

3.1.7. SQLite duomenų bazė

SQLite yra reliacinė duomenų bazių valdymo sistema, kuri, kaip rašoma šaltinyje [7], priešingai nei dauguma kitų duomenų bazių valdymo sistemų, neveikia kliento-serverio principu, o yra integruota į pačią programą. SQLite yra populiarus įrankis Java darbatalio aplikacijų kūrimui, nes nereikia rūpintis serveriu, konfigūracijomis ar papildomų aplikacijų instaliavimu.

3.2. Programinės darbo dalies struktūra

Darbą sudaro 8 Java klasės: *OptimaliDieta*, *Skaiciavimai*, *DietaPaciam*, *DietuInformacija*, *Kalendorius*, *ProduktuInformacija*, *ProduktuItraukimas* ir *SvorioIstorija*. *OptimaliDieta* yra pagrindinė, didžiausia klasė, joje yra „main“ metodas. Šioje klasėje yra sukuriamą pagrindinė naudotojo sąsaja, įjungiamą įsijungus programai. *Skaiciavimai* klasė gauna reikiamus parametrus iš pagrindinės klasės ir juos apdoroja, sudarant dietą automatiškai. Jeigu naudotojas dietą nusprendžia sudaryti savarankiškai, *OptimaliDieta* klasėje surinkti naudotojo duomenys yra perduodami ne į *Skaiciavimai* klasę, o į *DietaPaciam*. Ši klasė suteikia reikiamą informaciją ir grafinę naudotojo sąsaja savarankiškam dietos sudarymui. Automatiškai arba savarankiškai parinktas dietas galima išsaugoti duomenų bazėje. Išsaugotas dietas galima peržiūrėti arba ištrinti pasinaudojant *DietuInformacija* klasėje sukurta grafinę naudotojo sąsaja. Taip pat sugeneruotas arba išsaugotas dietas galima įtraukti į dietų kalendorių, kuris yra įgyvendintas klasėje *Kalendorius*. Klasė *SvorioIstorija* leidžia naudotojui įvesti/ištrinti savo svorio pokyčių informaciją, taip pat šią informaciją galima atvaizduoti grafiškai, nubrėžiant linijinį svorio kitimo grafiką. Likusios dvi klasės: *ProduktuInformacija* ir *ProduktuItraukimas* yra naudojamos maisto produktų duomenų bazės peržiūrėjimui ir valdymui.

3.3. Grafinės vartotojo sąsajos kūrimas

Kaip jau buvo minėta šiame darbe, grafinė naudotojo sąsaja kuriama naudojant Swing karkasą, papildant funkcionalumą AWT karkaso, SwingX plėtinio ir JFreeChart bibliotekos funkcijomis *OptimaliDieta*, *ProduktuInformacija*, *ProduktuItraukimas*, *SvorioIstorija*, *DietaPaciam*, *DietuInformacija* ir *Kalendorius* klasėse. Pirmasis žingsnis yra sukurti pagrindinį sistemos grafinį elementą, langą, kuriame bus talpinami visi kiti grafiniai elementai. Naudojant Swing karkasą, tai padaryti yra labai paprasta, užtenka vienos eilutės `SetSize(x,y)`, kur `x` yra lango plotis, o `y` yra lango aukštis pikseliais. Toliau apibrėžiami visi kiti darbe naudojami grafiniai elementai: mygtukai (`JButton`), akutės (`JRadioButton`), teksto laukeliai (`JLabel` ir `JTextArea`), teksto įvesties laukeliai (`JTextField`), žymimieji langeliai (`JCheckBox`), slankjuostės (`JScrollPane`), išskleidžiamieji sąrašai

(JComboBox), lentelės (JTable) ir meniu (JMenu, JMenuItem ir JMenuBar). Pastarasis elementas (meniu) yra identiškas visose klasėse, sukurtas naudotojo patogumui, kad būtų galima greičiau ir paprasčiau pereiti nuo vienos sistemos funkcijos prie kitos.

Duomenų bazės turiniui (maisto produktams, anksčiau naudotiems naudotojo duomenims ir kt.) atvaizdavimui dažniausiai naudojamos lentelės. Toks duomenų atvaizdavimo būdas yra tvarkingas ir patogus naudotojui, kuris gali rūšiuoti lentelės turinį pagal bet kurią iš stulpelių didėjimo arba mažėjimo tvarką, taip pat gali lengvai pasirinkti vieną arba daugiau lentelės eilučių, bei įrašus ištrinti iš duomenų bazės, arba įrašo informaciją panaudoti naujo dietos plano sudarymui (anksčiau įvestų naudotojo duomenų lentelės atveju). Lentelė naudojama ir savarankiško dietos sudarymo metu, tiesa, automatiškai sudarytos dietos, arba išsaugotos dietos (nesvarbu, ar sugeneruotos automatiškai, ar savarankiškai) yra atvaizduojamos teksto pavidalu tekstiniam laukelyje su slankjuoste. Kalendoriaus grafinėje sąsajoje yra naudojamas šių dviejų atvaizdavimo metodų (lentelės ir tekstinio laukelio) mišinys - pats kalendorius yra atvaizduojamas lentelės pavidalu, tačiau, pasirinkus dieną (lentelės langelį), kuriai yra priskirta dieta, ši dieta atvaizduojama tekstiniam laukelyje.

Dar vienas duomenų atvaizdavimo būdas - grafikas. Grafikai labiausiai tinka atvaizduoti su laiku kintančius duomenis, nes jie suteikia galimybę įvertinti pokyčius vizualiai, o ne tik „sausais“ skaičiais. Šiame darbe nėra daugybės pokyčių išsaugojimo ir stebėjimo funkcijų, todėl grafikas yra braižomas tik vizualiam svorio kitimo istorijos atvaizdavimui.

Duomenų valdymui ar įtraukimui dažniausiai naudojami mygtukai, papildant jų funkcionalumą išskleidžiamaisiais sąrašais, kuomet reikia pasirinkti atvaizduojamo/įtraukiamo elemento individualų pavadinimą arba identifikacinį numerį. Jeigu reikalingas patvirtinimas, pavyzdžiui, prieš ištrinant įrašą/(-us) iš duomenų bazės, naudojamos akutės („taip“ arba „ne“). Nepageidaujamiems produktams arba į duomenų bazę įtraukiamų naujų produktų kategorijoms nurodyti naudojami žymimieji laukeliai.

Šiame darbe jau buvo minėta, jog Java programavimo kalba parašytos aplikacijos yra vykdomos Java virtualiojoje mašinoje, todėl ta pati aplikacija gali sėkmingai veikti įrenginiuose su skirtingomis operacinėmis sistemomis (Windows, Linux, macOS ir kt.). Tiesa, ne viskas yra taip paprasta. Nors ta pati aplikacija ir pasižymi identišku funkcionalumu skirtingose operacinėse sistemose, ji nebūtinai atrodo vienodai. Skirtingos operacinės sistemos skirtingai atvaizduoja Java grafinius elementus, pakeičia jų išsidėstymą. Šiai problemai išspręsti Java Swing karkasas parūpina klasę *UIManager*. Šios klasės metodas „setLookAndFeel“ yra iškviečiamas iš karto įsijungus programai, „main“ metode, dar prieš sukuriant grafinius sistemos elementus. Metodo paskirtis yra nustatyti universalią sistemos išvaizdą ir elgseną, kuri turėtų būti identiška kiekvienoje operacinėje sistemoje. Tiesa, programos testavimo metu pastebėta, jog minimalių skirtumų atsiranda, nepaisant šio metodo panaudojimo. Pavyzdžiui, kartais Windows operacinėje sistemoje sukuriamas šiek tiek mažesnis langas ir į jį sunkiau telpa visi grafiniai elementai, negu macOS operacinėje sistemoje. Kadangi Windows operacinė sistema yra populiariausia tarp asmeninių kompiuterių, be to, yra geriau turėti šiek tiek per didelį, negu šiek tiek per mažą langą, galutinė sistemos versija yra orientuota į įrenginius su Windows programine įranga. Žinoma, Linux ir macOS operacinėse sistemose programa veikia lygiai taip pat, kaip Windows, o minimalūs estetiniai skirtumai naudotojui įtakos daryti neturėtų.

3.4. Dietos sudarymas

Pagrindinė sistemos paskirtis, kaip galima nesunkiai nuspėti iš darbo temos, yra optimalios dietos sudarymas. Sistema suteikia galimybę sudaryti dietą dviem būdais: automatiškai arba savarankiškai. Bet kuriuo atveju, prieš pradedant dietos sudarymą, reikia nustatyti kilokalorijų ir maistinių medžiagų (baltymų, angliavandenių, riebalų, skaidulinių medžiagų, kalcio ir cholesterolio) normas. Šios normos yra apskaičiuojamos individualiai kiekvienam naudotojui pagal jo asmeninius duomenis: amžių, svorį, ūgį, lytį, fizinio aktyvumo lygį bei tikslą (palaikyti svorį, numesti svorio arba jo priaugti). Naudodamas programą pirmą kartą, naudotojas turi šiuos duomenis suvesti pasinaudodamas *OptimaliDieta* klasėje sukurta grafine naudotojo sąsaja. Pasirinkus automatinę arba savarankišką dietos generavimą, suvesti duomenys yra išsaugomi duomenų bazėje, bei sekantį kartą generuodamas dietą naudotojas gali įsijungti jau panaudotų duomenų lentelę ir rinktis vieną iš įrašų greitesniam bei patogesniai dietos sudarymui. Be jau paminėtų asmeninių parametrų, naudotojas taip pat gali pasirinkti nepageidaujamus maisto produktus pagal kategorijas arba pasirinkti individualius produktus iš duomenų bazės, kurių nenori įtraukti į tuo metu sudaromą dietą. Pasirinkus automatinę dietos sudarymą, taip pat prašoma pasirinkti, kiek kartų per dieną naudotojas pageidauja valgyti (3 arba 5) ir laikotarpį, kuriam dieta bus sudaroma (1 diena, 3 dienas arba savaitei), taip pat naudotojas gali pasirinkti vieną arba kelis dietos parinkimo kriterijus (maistinių medžiagų normas) kaip ignoruojamus, pavyzdžiui, sudarant dietą ignoruoti baltymų kiekį dietoje ir pan. Galiausiai, jeigu naudotojas nepažymėjo mėsinių produktų kategorijos, kaip nepageidaujamos, suteikiama galimybė būtinai įtraukti mėsiską produktą į sudaromos dietos pietus. Gavus visus reikiamus duomenis, pradedamas savarankiškas arba automatinis dietos generavimas. Kaip jau buvo minėta, optimali dieta yra tokia, kuri patenkina reikiamas kilokalorijų ir maistinių medžiagų normas. Pirmiausia, pagal Hario Benedikto formulę, minimą šaltinyje [6], yra apskaičiuojamas kilokalorijų poreikis:

Vyrams: $Kilokalorijų\ Poreikis = (655 + (9.6 \times W) + (1.8 \times H) - (4.7 \times A)) \times K$,

Moterims: $Kilokalorijų\ Poreikis = (65.4 + (13.7 \times W) + (5 \times H) - (6.8 \times A)) \times K$,

čia W yra svoris kilogramais, H - ūgis centimetrais, A - amžius metais, bei K - aktyvumo koeficientas. Fiziškai neaktyviems žmonėms aktyvumo koeficientas yra 1.5, vidutiniškai aktyviems žmonėms - 1.7, o gyvenantiems aktyvų fizinį gyvenimą - 2.

Gautas kilokalorijų poreikis yra optimalus žmonėms, siekiantiems sveikai maitintis ir palaikyti esamą svorį. Jei naudotojas pasirenka tikslą numesti svorio, iš kilokalorijų poreikio yra atimama 500kcal. Jei tikslas yra svorio priaugimas, prie kilokalorijų poreikio pridedama 500kcal. Tuomet yra skaičiuojamas kūno masės indeksas (KMI) pagal standartinę formulę, taip pat minimą šaltinyje [6]:

$$KMI = W/(H \times H),$$

čia W yra svoris kilogramais, o H - ūgis metrais.

Jeigu dieta generuojama automatiškai, gautos reikšmės, kartu su aktyvumu, amžiumi, lytimi, valgių per dieną skaičiumi, dietos sudarymo laikotarpiu, nepageidaujamų produktų ir ignoruojamų dietos sudarymo kriterijų sąrašais bei kintamuoju, nurodančiu, ar pietums privaloma priskirti mėsos produktą, ar ne, yra perduodamos *Skaiciavimai* klasėje esančiam dietos sudarymo metodui, jeigu dieta sudaroma savarankiškai, tuomet reikiami duomenys yra perduodami *DietaPaciam*

klasės konstruktoriui.

3.4.1. Automatinis dietos generavimas

Skaiciavimai klasėje esantis automatinio dietos generavimo metodas iš pradžių sukuria 12 sąrašų, saugančių visą dietos sudarymui reikalingą informaciją apie produktus. Sąrašai yra kuriami lygiagrečiai, todėl yra vienodo ilgio bei to paties indekso skirtingų sąrašų elementai saugo to paties maisto produkto informaciją. Sąrašuose saugoma ši informacija: kilokalorijų, baltymų, angliavandenių, riebalų, skaidulinių medžiagų, kalcio ir cholesterolio kiekiai 100 gramų (arba mililitrų) produkto, taip pat standartinės produkto porcijos dydis gramais (arba mililitrais), produkto pavadinimas, valgis, kurio metu produktas yra tinkamiausias (pusryčiai, pietūs, vakarienė arba užkandžiai (priešpiečiai ir pavakariai)), produkto kategorija ir požymis, nusakantis, ar produktas yra skystas. Kilokalorijos skaičiuojamos paprastais vienetais (jei parašyta, kad produktas turi X kilokalorijų, reiškia, 100g produkto yra X kilokalorijų), tuo tarpu išvardytų medžiagų kiekiai skaičiuojami mikrogramais (arba nanolitrais). Sąrašai suformuojami while cikle, lygiagrečiai į sąrašus įtraukiant visų maisto produktų informaciją, išskyrus produktus, kuriuos naudotojas pasirinko kaip nepageidaujamus. Informacija apie produktus gaunama iš duomenų bazėje saugomos lentelės *MAISTO_PRODUKTAI*. Sekantis žingsnis yra įvertinti skaidulinių medžiagų poreikį. Tiek per didelį, tiek per mažą skaidulinių medžiagų suvartojimą yra žalingas sveikatai. Šaltinyje [3] teigiama, jog optimalus skaidulinių medžiagų kiekis yra 14g/1000kcal, taigi, optimalų kiekį galima apskaičiuoti pagal formulę:

$$\text{Skaidulinių Medžiagų Poreikis} = (\text{Kilokalorijų Poreikis} \times 0.014)g.$$

Žinoma, patenkinti kiekvienos medžiagos poreikio idealiai, be paklaidos, yra beveik neįmanoma. Dėl to šiame darbe naudojamos minimalaus ir maksimalaus medžiagų kiekio reikšmės. Šiuo atveju:

$$\begin{aligned} \text{Minimalus Skaidulinių Medžiagų Kiekis} &= \text{Skaidulinių Medžiagų Poreikis} - 5g, \\ \text{Maksimalus Skaidulinių Medžiagų Kiekis} &= \text{Skaidulinių Medžiagų Poreikis} + 5g. \end{aligned}$$

Apskaičiavus skaidulinių medžiagų poreikį, skaičiuojamas baltymų, angliavandenių ir riebalų poreikis. Šių medžiagų sveiko suvartojimo kiekio intervalai apskaičiuojami pagal šaltinyje [6] pateiktas proporcijas:

Neturintiems atsvario neaktyviems žmonėms: *Baltymų poreikis* = 10 - 20 %; *Angliavandenių poreikis* = 55 - 75 %; *Riebalų poreikis* = 10 - 15 %,

Neturintiems atsvario aktyviems žmonėms: *Baltymų poreikis* = 15 - 20 %; *Angliavandenių poreikis* = 55 - 60 %; *Riebalų poreikis* = 15 - 20 %,

Turintiems atsvario žmonėms: *Baltymų poreikis* = 25 - 35 %; *Angliavandenių poreikis* = 45 - 55 %; *Riebalų poreikis* = 15 - 25 %,

čia procentaliai nurodoma, kokią dalį valgiaraštyje turėtų sudaryti atitinkamos maistinės medžiagos. Šiame darbe, kaip ir šaltinyje [6], turinčiais atsvario yra laikomi žmonės, kurių KMI > 25.5.

Gautos proporcijos iš karto yra padauginamos iš kilokalorijų poreikio, taip nustatant, koks kalorijų skaičius turėtų būti suvartojamas kaip baltymai, koks - kaip angliavandeniai ir koks - kaip riebalai. Šaltinyje [6] teigiama, jog 1g baltymų, taip pat ir 1g angliavandenių turi po 4kcal, tuo tarpu 1g riebalų turi 9kcal. Įvertinus šių maistinių medžiagų normas, nustatoma, kiek naudotojas turėtų suvartoti kalcio. Optimalus kalcio suvartojimo kiekis, priklausomai nuo žmogaus amžiaus ir kai kurioms amžiaus grupėms - nuo lyties, yra pateikiamas šaltinyje [5]. Sudarant optimalią dietą, kalcio kiekiui taikoma 0.01g paklaida. Atradus reikiamą kalcio kiekį, pradedama dietos paieška. Reikia nustatyti, kiek kalorijų turėtų būti sunaudojama per kiekvieną iš valgių. Šaltinyje [6] teigiama, jog maitinantis 3 kartus per dieną, pusryčiams turėtų būti sunaudojama 35% paros energijos normos, pietums - 40%, o vakarienei - 25%. Jeigu maitinamasi 5 kartus per dieną, pusryčiams turėtų būti suvartojama 30% paros kilokalorijų kiekio, priešpiečiams ir pavakariams - po 10%, pietums - 30% ir vakarienei - 20%.

Apibrėžus kilokalorijų normas kiekvienam iš valgių, pradedama rinkti produktus į dietą. Pusryčiai parenkami gana paprastu būdu - tiesiog paimamas vienas produktas, tinkantis pusryčiams, o šio produkto kiekis parenkamas pagal tai, kiek reikia, kad būtų idealiai patenkintas pusryčių kilokalorijų poreikis. Tuomet renkami valgiaraščiai likusiems dviem arba keturiem valgiams. Tiek užkandžiams, tiek ir vakarienei mitybos planas sudaromas pagal identiškus algoritmus, tiesiog produktai renkami iš atitinkamam valgiui tinkamų produktų aibės. Užkandžių bei vakarienės parinkimas vyksta pagal šiuos žingsnius:

1. Atsitiktine tvarka parenkamas vienas produktas iš visų dietos sudarymui potencialiai naudojamų produktų aibės.
2. Tikrinama, ar parinktas produktas yra tinkamas šiam valgiui, ar dar nėra produkto, atrinkto šiam valgiui, priklausančio lygiai tai pačiai kategorijai, taip pat tikrinama, ar produktas nėra skystas, o jeigu yra, ar dar nėra šiam valgiui priskirta kito skysto produkto.
3. Jeigu visos 2 žingsnio sąlygos yra patenkinamos, į valgį yra įtraukiama standartinė produkto porcija, prie valgiaraščio generavimo metu sukaupto kilokalorijų kiekio yra pridedama pasirinkto produkto standartinės porcijos energinė vertė. Jeigu produktas yra skystas, pažymima, jog vienas skystas produktas į šio valgio valgiaraštį jau yra įtrauktas, t.y. kito skysto produkto parinkti nebebus galima.
4. Jeigu prie sukaupų kilokalorijų kiekio pridėjus 30 kilokalorijų gaunamas didesnis skaičius, negu kilokalorijų poreikis šiam valgiui, paieška yra nutraukiama.
5. Jeigu paieška yra tęsiama, patikrinama, ar skaičius, kelintą kartą algoritmas yra kartojamas, yra dalus iš 10 ir ar į šio valgio valgiaraštį yra įtrauktas daugiau negu vienas produktas.
6. Jeigu bent viena iš sąlygų nėra patenkinama, algoritmas grįžta į 1 žingsnį. Jeigu abi 5 žingsnio sąlygos yra patenkinamos, iš šiam valgiui jau parinktų produktų yra parenkamas vienas atsitiktine tvarka.
7. Patikrinama, ar 6 žingsnyje parinkto produkto kiekis sudaromame valgiaraštyje neviršija dviejų standartinių produkto porcijų ir ar produktas nėra skystas. Jeigu šios sąlygos yra patenkinamos, prie dietos yra pridedama pusė standartinės produkto porcijos, o algoritmas grįžta į pirmąjį žingsnį.
8. Jeigu 7 žingsnio sąlygos nėra patenkinamos, kintamasis „papildomas“ yra padidinamas vienetu (+1). Jeigu šis kintamasis yra dalus iš 10 ir 6 žingsnyje parinktas produktas nėra skystas, prie dietos yra pridedama pusė šio produkto standartinės porcijos, nebepriklausomai nuo to, koks produkto kiekis jau yra įtrauktas į dietą.
9. Grįžtama į 1 žingsnį. Ciklas yra kartojamas tol, kol 4 žingsnio sąlyga nėra patenkinta.

Kaip galima suprasti išsigilinus į algortimą, optimaliausiu atveju sudarytas valgiaraštis patenkintų šias sąlygas:

1. Neturėtų būti daugiau negu vieno tos pačios kategorijos produkto.
2. Neturėtų būti daugiau negu vieno skysto produkto.
3. Neturėtų būti viršijamos duomenų bazėje nurodytos standartinės produktų porcijos.

Jeigu potencialiai į valgiaraštį įtraukti tinkamų produktų aibė nėra labai maža, o naudotojo kilokalorijų poreikis nėra didelis, šios sąlygos yra patenkinamos. Kitu atveju, tenka daryti išimtis. Šios išimtys yra apibrėžtos užkandžių ir vakarienės valgiaraščių sudarymo algoritmo 5 - 8 žingsniuose. Nuspręsta, jog tinkamiausias būdas pasiekti reikiamą kilokalorijų kiekį yra viršyti standartinės produktų porcijas. Šis variantas yra patogesnis ir patrauklesnis naudotojui, negu įtraukiant daugiau negu vieną tos pačios kategorijos produktą, pavyzdžiui, yra kur kas patogiau pasigaminti ir suvalgyti didesnę porciją jautienos, negu kartu su jautiena gamintis dar ir kiaulienos ar pan. Taip pat algoritme nurodyta, jog skystų produktų porcijos nėra didinamos jokiais aplinkybėmis, nes per didelis skysčio kiekis vieno valgio metu taip pat nebūtų patogus naudotojui.

Pietūs yra sudaromi pagal labai panašų algoritmą, kaip ir užkandžiai bei vakarienė, tiesiog pietų parinkimo algoritmas yra papildytas dviem papildomais patikrinimais. Vienas iš patikrinimų yra tai, jog į valgiaraštį pietums negali būti įtrauktas ne tik daugiau negu vienas skystas produktas, bet ir daugiau nei vienas daržovių kategorijai priklausantis produktas (pavyzdžiui, vienos salotos gali priklausyti tik daržovių kategorijai, kitos - ir daržovių, ir pieno produktų, todėl, neturint šio patikrinimo, abi salotos galėtų būti įtrauktos į dietą, taip sudarydamos ne itin patogų ir akiai patrauklų valgiaraštį). Kitas patikrinimas yra reikalingas tik tuo atveju, jeigu naudotojas nurodė būtinai pietums priskirti mėsiską produktą. Jeigu naudotojas pasinaudojo šia sistemos funkcija, tuomet, sudarinėjant valgiaraštį pietums ir pasiekus reikiamą kilokalorijų kiekį yra tikrinama, ar buvo parinktas mėsos produktų kategorijai priklausantis produktas. Jeigu mėsisko produkto valgiaraštyje nėra, visos produktų parinkime naudojamos reikšmės (atrinktų produktų ir jų kiekių sąrašai, kintamasis „papildomas“ ir kt.) yra anuliuojamos ir algoritmas grįžta į pirmąjį žingsnį.

Tiesa, atlikus paiešką kiekvienam iš valgių, dietos sudarymas dar nėra baigtas. Kol kas tėra sukurti valgiaraščiai, idealiai patenkinantys kilokalorijų poreikį kiekvienam iš 3 arba 5 paros valgių, tačiau nėra aišku, ar šie valgiaraščiai pilnavertiškai patenkina maistinių medžiagų dietoje normas. Dėl šios priežasties yra suskaičiuojama, kiek atrinktuose produktuose yra baltymų, angliavandenių, riebalų, kalcio, skaidulinių medžiagų ir cholesterolio. Apskaičiuavus medžiagų kiekius, vykdomas patikrinimas, ar kiekis atitinka rekomenduojamas normas. Kaip teigiama šaltinyje [2], cholesterolio per parą neturėtų būti suvartojama daugiau negu 0.3g, tuo tarpu likusių medžiagų normų nustatymo tvarka jau buvo aprašyta šiame darbe. Kaip jau buvo minėta, naudotojas, generuodamas dietą automatiškai, turi galimybę ignoruoti tam tikras arba net visas maistines medžiagas. Jeigu kai kurios maistinės medžiagos buvo pasirinktos, kaip ignoruojamos, į jų kiekį dietoje, žinoma, nėra atsižvelgiama. Jeigu patikrinimo metu nustatyta, kad medžiagų kiekiai dietoje atitinka reikiamas normas, paieška yra nutraukiama. Kitu atveju, grįžtama į pradžią, valgiaraščiai kiekvienam iš valgių yra sudaromi iš naujo, vėl yra vykdomas patikrinimas ir t.t., kol sudaryta dieta atitinka reikiamas normas. Jeigu dieta generuojama ne 1 dienai, o 3 dienom arba savaitei, normas atitinkančios vienos dienos dietos yra išsaugomos, o paieška kartojama tol, kol sėkmingai sudarytų dietų skaičius atitinka reikiamą dienų skaičių.

Deja, dažniausiai atsitiktiniu būdu sudaryta dieta maistinių medžiagų normų idealiai nepatenkina ir ciklą tenka kartoti tūkstančius kartų. Kai kuriais atvejais, visų normų idealiai patenkinti netgi nėra įmanoma, dėl per mažos dietos sudarymui naudojamos produktų aibės ir/arba nestan-

dartinių naudotojo fizinių duomenų. Tokiu atveju, tenka daryti išimtis. Pirmuosius 10 000 kartų algoritmas bando visas normas patenkinti idealiai. Jeigu to padaryti visgi nepavyksta, pradedama mažinti minimalias skaidulinių medžiagų ir kalcio reikšmes, bei didinti maksimalias šių dviejų medžiagų, taip pat ir cholesterolio, normas. Šie veiksmai vyksta kartą per 1000 iteracijų, 1% sumažinant minimalias ir 1% padidinant maksimalias reikšmes. Baltymų, angliavandenių bei riebalų normų stengiamasi nekoreguoti, nes jos yra laikomos pačia svarbiausia subalansuotos dietos dalimi. Visgi, kartais ir šių normų patenkinti idealiai nepavyksta. Pagrindinė to priežastis - per mažas maisto produktų, tinkamų dietai, aibė. Ypač problemų kyla, kai naudotojas pasirenka kelias maisto produktų kategorijas, kaip nepageidaujamas, pavyzdžiui, atsisakius žuvies ir mėsos, baltymų kiekis dietoje dažniausiai gaunasi gerokai mažesnis, negu reikia. Dėl šios priežasties, nesudarius dietos per 40 000 bandymų (30 000 papildomų bandymų nuo to karto, kai pradėta didinti tinkamus kalcio, skaidulinių medžiagų ir cholesterolio intervalus), pradedama kas 1000 iteracijų 1% mažinti minimalias ir 1% didinti maksimalias baltymų, angliavandenių ir riebalų reikšmes.

Verta paminėti, jog žmogui, neturinčiam labai išskirtinių fizinių duomenų (labai didelio arba mažo kūno svorio ir pan.), taip pat neatsisakančiam jokių maisto produktų, dietą dažniausiai pavyksta sudaryti dar nepradėjus koreguoti pagrindinių maistinių medžiagų (baltymų, angliavandenių ir riebalų) normų. Labai išskirtiniais atvejais, pavyzdžiui, jeigu naudotojo kūno masė yra daugiau negu 120kg, tikslas - priaugti svorio, bei dauguma maisto produktų pažymėti, kaip nepageidaujami ir/arba duomenų bazėje tiesiog saugomas labai mažas maisto produktų kiekis, optimalios dietos sudaryti gali būti tiesiog neįmanoma. Žinoma, nuolatos mažinant minimalias ir didinant maksimalias dietos apribojimo reikšmes, dietą sudaryti anksčiau ar vėliau pavyksta, tačiau, jeigu tos normos labai smarkiai skiriasi nuo pradinių optimalios mitybos rekomendacijų, tokios dietos vadinti optimalia negalima, todėl, kuriant sistemą, būtina numatyti ir išskirtinius atvejus, kuomet paieška būtų nutraukta. Šiame darbe sprendimas nutraukti dietos paiešką yra priimamas panaudojant laikmatį ir algoritmo ciklo pakartojimo skaičiaus patikrinimą. Paieška nutraukiama, kai yra patenkinamos abi šios sąlygos:

1. Algoritmas nerado nė vienos tinkamos dietos per 1 minutę.
2. Algoritmas nerado nė vienos tinkamos dietos per 50 000 bandymų.

Jeigu abi šios sąlygos yra patenkinamos, naudotojas yra informuojamas, jog dietos sudaryti neįmanoma ir vienintelis būdas sėkmingai panaudoti šį sistemos funkcionalumą yra padidinti algoritmui pateikiamų maisto produktų aibę t.y. mažesnę produktų dalį pažymėti kaip nepageidaujamą ir/arba įtraukti naujų maisto produktų į duomenų bazę. Verta pakartoti, jog tai yra tik teorinė situacija, kuri neturėtų pasitaikyti praktikoje, nebent naudotojas specialiai įvedinės neracionalius duomenis, tikrindamas sistemos galimybes ir pan. Kitas atvejis, kuomet naudotojas gauna panašų pasiūlymą padidinti dietos generezimui naudojamų maisto produktų aibę yra kai bent vienam iš valgių tinkamų neskystų produktų poaibis yra tuscias. Sistema dar prieš pradėdama dietos generavimo ciklą atlieka patikrinimą, ar bent teoriškai yra įmanoma sudaryti dietą ir algoritmo vykdymą pradeda tik tuo atveju, jeigu ši sąlyga yra patenkinama.

Automatinio dietos parinkimo algoritmas nebuvo kurtas konkrečiam žmogui arba konkrečiai maisto produktų duomenų bazei - siekta sukurti algoritmą, kuris būtų kuo įmanoma universalesnis. Taip pat siekta, jog sugeneruota dieta ne tik būtų tinkama matematiškai, bet ir atrodytų racionaliai, primintų žmogaus sudarytą valgiaraštį, o ne tik matematiškai išspręsta uždavinį. Šio tikslo buvo siekiama bandymų būdu, naudojantis ne tik logika, bet ir žmogiškąja intuicija. Dėl to pusryčiams nuspręsta nenaudoti algoritmo, kuris yra naudojamas kitiems keturiems valgiams, o į pietų parinkimo

algoritmą įtrauktos papildomos sąlygos. Kitu atveju, kai kurios sugeneruotos dietos būtų šiek tiek mažiau patrauklios naudotojams, nei yra dabar. Žinoma, sugeneruotos dietos nėra tobulos. Visų pirma, tobulų dietų nebūna. Visų antra, kas tinka vienam žmogui, nebūtinai tinka kitam, pavyzdžiui, vienas žmogus sutiktų vakarienei vartoti žuvies produktus kartu su pieno produktais, o kitas žmogus - ne. Dėl to naudotojui suteikta galimybė vienu mygtuko paspaudimu generuoti dietą iš naujo, koreguoti maisto produktų duomenų bazę, nebūtinai įtraukti visus duomenų bazėje esančius produktus į tuo metu generuojama dietą ir kt. Tobulų sistemų nebūna, tačiau šiame darbe siekta sukurti kuo įmanoma universalesnį, greitesnį ir intuityvesnį įrankį, turint ribotus laiko išteklius bei darbą vykdant vienam žmogui.

3.4.2. Savarankiškas dietos generavimas

Savarankiško dietos generavimo funkciją pasirinkęs naudotojas yra nukreipiamas į *DietaPaciam* klasės grafinę naudotojo sąsają. Šios klasės konstruktorius gauna visus naudotojo asmeninius duomenis: amžių, lytį, svorį, ūgį, tikslą, fizinio aktyvumo lygį bei nepageidaujamus produktus. Būtų galima iš karto perduoti ir kūno masės indeksą bei kilokalorijų poreikį, tačiau šias reikšmes apskaičiuoti yra paprasta panaudojant vos kelias eilutes kodo, todėl nuspręsta geriau turėti mažiau kintamųjų konstruktoriuje, juolab, kad visos reikšmės, panaudojamos šių dydžių apskaičiavimui, yra ir taip reikalingos kitiems skaičiavimams bei dietos išsaugojimui. Sukūrus *DietaPaciam* objektą, pirmiausia, pagal ankstesniuose poskyriuose jau minėtas formules ir šaltinius, yra apskaičiuojamos kilokalorijų ir maistinių medžiagų normos. Šios normos, kaip rekomendacijos, yra pateikiamos naudotojui savarankiško dietos sudarymo grafinės naudotojo sąsajos lango viršuje. Po šiomis rekomendacijomis yra pateikiamas išskleidžiamasis sąrašas, sudarytas iš visų duomenų bazėje saugomų produktų pavadinimų, išskyrus produktus, kuriuos naudotojas pasirinko kaip nepageidaujamus. Šio sąrašo paskirtis yra į valgiaraštį įtraukiamo produkto pasirinkimas. Pavadinimai sąraše yra surūšiuoti abėcėlės tvarka, todėl galima tiesiog pasirinkti produktą iš sąrašo, arba galima įrašyti tikslų pavadinimą pačiam. Naudotojo patogumui maisto produktų išskleidžiamajame sąraše yra integruota jau minėta automatinio užpildymo funkcija. Taip pat, virš šio sąrašo yra mygtukas, kurį paspaudus galima automatinio užpildymo funkciją išjungti, jeigu naudotojas nėra patenkintas šiuo funkcionalumu. Jeigu naudotojas pajunta, kad visgi automatinio užpildymo funkcija buvo naudinga įtraukiant produktus į dietą, vieno mygtuko paspaudimu ją galima sugrąžinti. Į sąrašą nėra įtraukti nepageidaujami produktai, tačiau, jeigu naudotojas visgi nuspręstų įrašyti vieno iš nepageidaujamų, duomenų bazėje saugomų produktų pavadinimą, programa leistų įtraukti produktą į dietą. Pasirinkus produktą, reikia nurodyti jo kiekį gramais arba mililitrais. Visi produktai, kuriuos naudotojas jau įtraukė į dietą, yra atvaizduojami lentelėje. Lentelėje atvaizduojami produktų pavadinimai, kiekiai, kilokalorijos bei visų maistinių medžiagų kiekiai. Lentelės eilutes (produktų pavadinimus ir jų informaciją) galima rūšiuoti pagal bet kurią iš kriterijų (stulpelių) didėjimo arba mažėjimo tvarka, taip pat galima bet kurią į dietą jau įtrauktą produktą ištrinti. Kiekvieną kartą įtraukus arba ištrinus produktą iš dietos, savarankiško dietos sudarymo lango apačioje yra atnaujinama informacija apie kilokalorijų bei maistinių medžiagų dietoje trūkumus bei perteklius. Savarankiškai dieta yra generuojama tik vienai dienai ir valgiaraštis nėra skirstomas į valgius, tiesiog padedama naudotojui susidaryti produktų sąrašą ir pamatyti, kiek tokia valgiaraštyje yra kilokalorijų, kokių medžiagų trūksta bei kokių medžiagų yra per daug, lyginant su optimaliausiomis sveikos mitybos rekomendacijomis.

3.4.3. Automatinio ir savarankiško dietų generavimo palyginimas

Šio darbo tikslas - padėti naudotojui pasiekti tinkamą mitybą. Pagrindinis uždavinys, siekiant šio tikslo, yra optimalaus valgiaraščio generavimas. Valgiaraštis generuojamas pagal moksliniais tyrimais paremtas matematinės formules, tačiau, šiame darbe sukurtos sistemos pagalba, dietą galima generuoti dviem skirtingais būdais: automatiškai arba savarankiškai. Abu būdai turi privalumų bei trūkumų. Automatinio dietos generavimo privalumai:

1. Dieta sugeneruojama greitai.
2. Visas darbas atliekamas automatiškai, be naudotojo pastangų.
3. Galima generuoti dietą nebūtinai 1 dienai, o ir 3 dienom arba savaitei.
4. Vienu mygtuko paspaudimu galima generuoti dietą kiek norima kartų pagal tuos pačius parametrus.

Savarankiško dietos generavimo privalumai:

1. Yra dalykų, kuriuos žmogui suprasti yra labai paprasta, tačiau labai sunku apibrėžti programiniu kodu. Vietoje to, kad bandyti sudaryti idealias maisto produktų parinkimo į dietą taisykles, galima norimą valgiaraštį susidaryti pačiam, pasinaudojant elementaria žmogiškąja intuicija.
2. Algoritmas, kuris sudaro optimaliai vieno žmogaus poreikius atitinkančią dietą, nebūtinai yra optimalus kitam žmogui. Be to, nėra lengva sukurti algoritmą, vienodai gerai dirbantį su visiškai skirtingomis maisto produktų duomenų bazėmis. Dėl to naudotojui yra naudinga turėti galimybę susidaryti valgiaraštį turint visišką pasirinkimo laisvę, atsižvelgiant į matematiškai apskaičiuotas rekomendacijas arba jas ignoruojant.

3.5. Kitos sistemos funkcijos

3.5.1. Išsaugotos dietos

Šiame darbe jau ne kartą buvo minėta, jog sugeneruotas dietas galima išsaugoti. Dietų išsaugojimui naudojama duomenų bazės lentelė *ISSAUGOTOS_DIETOS*. Išsaugotas dietas galima peržiūrėti arba trinti *DietuInformacija* klasėje įgyvendintos grafinės naudotojo sąsajos pagalba. Išsaugotų dietų valdymo lange yra išskleidžiamasis sąrašas dietos pasirinkimui, saugantis visų duomenų bazėje išsaugotų dietų identifikacinius numerius ir mygtukas, skirtas ištrinti pasirinktą dietą. Pasirinktos dietos yra atvaizduojamos tekstiname laukelyje su slankjuoste. Pasirinkus dietą ištrinti, dar paprašoma papildomai patvirtinti veismą.

3.5.2. Dietų kalendorius

Ką tik sugeneruotas ir/arba išsaugotas dietas galima įtraukti į mitybos kalendorių, kuris yra sukurtas klasėje *Kalendorius*. Kalendorius yra atvaizduotas Java lentelės pavidalu. Dienos, kurioms yra priskirtos dietos (lentelės langeliai) yra nuspalvintos kitokia spalva (tamsiai pilka, o ne balta). Pasirinkus dieną, kuri jau yra užimta, galima peržiūrėti tai dienai priskirtą dietą. Dieta atvaizduojama tekstiname laukelyje su slankjuoste, analogiškame išsaugotų dietų peržiūros lange esančiam laukeliui. Užimtas dienas galima atlaisvinti ištrinant dietas iš duomenų bazės. Kalendoriui priskirtos dietos yra saugomos *KALENDORIUS* lentelėje duomenų bazėje. Dienos nebūtinai yra atlaisvinamos po vieną, sistema suteikia galimybę atlaisvinti visas pasirinkto mėnesio dienas

iš karto. Dienoms, kurios dar neturi priskirtų dietų, dietas galima priskirti iš išsaugotų dietų sąrašo. Jeigu kalendoriuje yra priskirtų dietų, kurios nėra saugomos *ISSAUGOTOS_DIETOS* lentelėje, šias dietas galima įtraukti į išsaugotų dietų sąrašą ir vėliau priskirti kitoms dienoms. Jeigu į kalendorių bandoma priskirti 3 arba 7 dienoms sugeneruotą dietą, pasirinktai dienai yra priskiriama pirmosios dienos dieta, sekančiai dienai priskiriama antrosios dienos dieta ir t.t. Žinoma, tokiu atveju, ne tik pasirinkta diena, bet ir sekančios 2 arba 6 dienos turi būti laisvos, nes vienai dienai gali būti priskirta tik viena dieta.

3.5.3. Maisto produktų duomenų bazė

Klasėje *ProduktuInformacija* yra parūpinta grafinė naudotojo sąsaja, skirta duomenų bazėje esančių produktų peržiūrai, rūšiavimui bei trynimui. Maisto produktų sąrašas ir kiekvieno iš jų informacija (kilokalorijų 100g skaičius ir maistinė sudėtis) yra atvaizduota lentelėje, duomenis gavus iš *MAISTO_PRODUKTAI* lentelės duomenų bazėje. Instaliuodamas programą naudotojas yra parūpinamas standartine, nelabai didele duomenų baze, kurią sudaro produktai, atrinkti iš šaltinio [1], tačiau ši duomenų bazė yra lanksti, naudotojas gali atlikti joje bet kokius pakeitimus. Produktus galima rūšiuoti pagal bet kurį iš kriterijų didėjimo arba mažėjimo tvarka, taip pat galima pasirinkti bet kokių produktų skaičių lentelėje vienu metu ir produktus ištrinti. Naujų produktų įtraukimo į duomenų bazę funkcionalumas ir grafinė naudotojo sąsaja yra parūpinta *ProduktuItraukimas* klasėje.

3.5.4. Svorio kitimo istorijos išsaugojimas

Žodis „dieta“ daugeliui asocijuojasi su svorio metimu. Kitą vertus, dalis žmonių, pavyzdžiui, aktoriai, besiruošdami tam tikram vaidmeniui, arba sportininkai, kartais laikosi dietų turėdami priešingą tikslą - priaugti svorio. Bet kuriuo atveju, ar norima priaugti, ar numesti svorio, o galbūt palaikyti esamą svorį, tačiau daryti tai maitinantis pagal sveikos mitybos rekomendacijas, galimybė stebėti svorio pokyčius yra naudinga. Dėl šios priežasties sistema naudotojui suteikia galimybę išsaugoti paskutinių metų laikotarpio svorio kitimo istoriją (nuspręsta, jog senesnė nei metų informacija nėra labai aktuali). Naudotojas gali įvesti savo svorį bei priskirti įrašui vieną iš pastarųjų metų laikotarpio datų, taip pat stebėti išsaugotus įrašus tiek paprastu tekstu, tiek ir nubrėžiant grafiką. Grafikai žmonėms dažnai būna vaizdingesnis ir intuityviai suprantamesnis pokyčių atvaizdavimo būdas už tiesiog tekstą.

3.5.5. Kūno masės indekso skaičiavimas

Viena iš paprasčiausių sistemos funkcijų yra kūno masės indekso skaičiuoklė. KMI, kaip jau buvo minėta šiame darbe, yra naudojamas dietos sudarymo metu, paskirstant baltymų, angliavandenių ir riebalų suvartojimo proporcijas, tačiau, dietos generavimo metu, naudotojui nėra pateikiamas. Naudodamas šią skaičiuoklę, naudotojas tiesiog įveda savo ūgį bei svorį ir sužino savo KMI. Sistemą būtų galima papildyti ir kitomis skaičiuoklėmis, pavyzdžiui, kilokalorijų poreikio arba maistinių medžiagų pasiskirstymo, tačiau šios reikšmės, priešingai negu KMI, yra pateikiamos naudotojui dietos generavimo metu, todėl nuspręsta neužpildyti darbo pasikartojančiu funkcionalumu.

Išvados ir rekomendacijos

Darbo tikslas buvo sukurti sistemą, padedančią naudotojui susikurti asmeninę optimalią dietą ir skatinančią tinkamus mitybos įpročius. Kad įgyvendinti šį tikslą, buvo atlikta panašaus tikslo siekiančių sistemų analizė, susikurtas teorinis sistemos modelis bei sukurta Java aplikacija, siekianti teorinę informaciją pritaikyti praktiškai. Pagrindinis suskurtos sistemos funkcionalumas - optimalios dietos generavimas. Šį funkcionalumą galima išskirti į dvi dalis - automatinį ir sava-rankišką dietos generavimą. Sugeneruotas dietas galima išsisaugoti bei įtraukti į dietų kalendorių. Išsaugotas ir į kalendorių įtrauktas dietas galima peržiūrėti ir ištrinti. Dietos yra generuojamos iš duomenų bazėje saugomų produktų. Duomenų bazė yra lanksti, produktus galima iš jos ištrinti bei įtraukti į ją naujus. Dietų generavimo metu yra atsižvelgiama į naudotojo poreikius, kurie ap-skaiciuojami pagal naudotojo pateiktą informaciją. Naudotojo pateikta informacija yra išsaugoma, kad kitą kartą dietą būtų galima generuoti greitai ir patogiai pagal tą pačią informaciją. Sistema naudotojui papildomai suteikia svorio kitimo istorijos išsaugojimą ir galimybę avaizduoti šią is-toriją grafike, taip pat ir kūno masės indekso skaičiuoklę. Darbo metu stengtasi ne tik išspręsti matematines dietos parinkimo užduotis, bet ir parūpinti naudotojui pakankamai patrauklę grafinę naudotojo sąsają. Naudotojo patogumui sukurtas meniu, padedantis greitai ir patogiai pereiti nuo vienos sistemos funkcijos prie kitos, duomenų atvaizdavimui panaudotos Java lentelės (JTable), kurios ne tik pateikia duomenis akiai patraukliu formatu, bet ir palengvina duomenų rūšiavimą bei pasirinkimą. Taip pat naudotojas turi galimybę panaudoti paieškos ir automatinio užpildymo funkcijas, padedančias atlikti darbą greičiau. Pagrindinės išvados, priimtose darbo metu:

1. Sudarant optimalią dietą yra svarbūs ne tik matematiniai skaičiavimai, bet ir žmogiškoji intuicija.
2. Daugelis šio darbo kūrimo laikų (2019) dietos parinkimo sistemų vis dar yra orientuotos į pagalbą naudotojui pačiam susidarant valgiaraštį, o ne visišką šio proceso automatizavimą.
3. Svarbus yra ne tik programos funkcionalumas, bet ir išvaizda, grafinė naudotojo sąsaja.

Nors nepriklausomybė nuo interneto ryšio yra privalumas, vystant šią sistemą toliau, būtų ga-lima parūpinti prisijungimą prie interneto naudojančių funkcijų. Interneto ryšys galėtų būti ypač naudingas praplėčiant maisto produktų duomenų bazės funkcionalumą. Reikėtų sukurti nutolusią duomenų bazę, kurios struktūra sutaptų su šiame darbe naudojama maisto produktų duomenų baze ir į kurią būtų įtraukta kur kas daugiau maisto produktų. Tuomet naudotojas galėtų įkelti norimus maisto produktus į savo lokalią duomenų bazę vos keliais mygtukų paspaudimais, vietoje to, kad visą informaciją rinktų rankiniu būdu, kaip kad tenka daryti dabar. Būtų galima sukurti ir daugiau mažesnių duomenų bazių, pavyzdžiui, duomenų bazė, skirta išskirtinai veganams, kurie galėtų iš karto parsisiųsti šią duomenų bazę ir nenaudoti laiko bei pastangų susidarydami duomenų bazę patys.

Ateities tyrimų planas

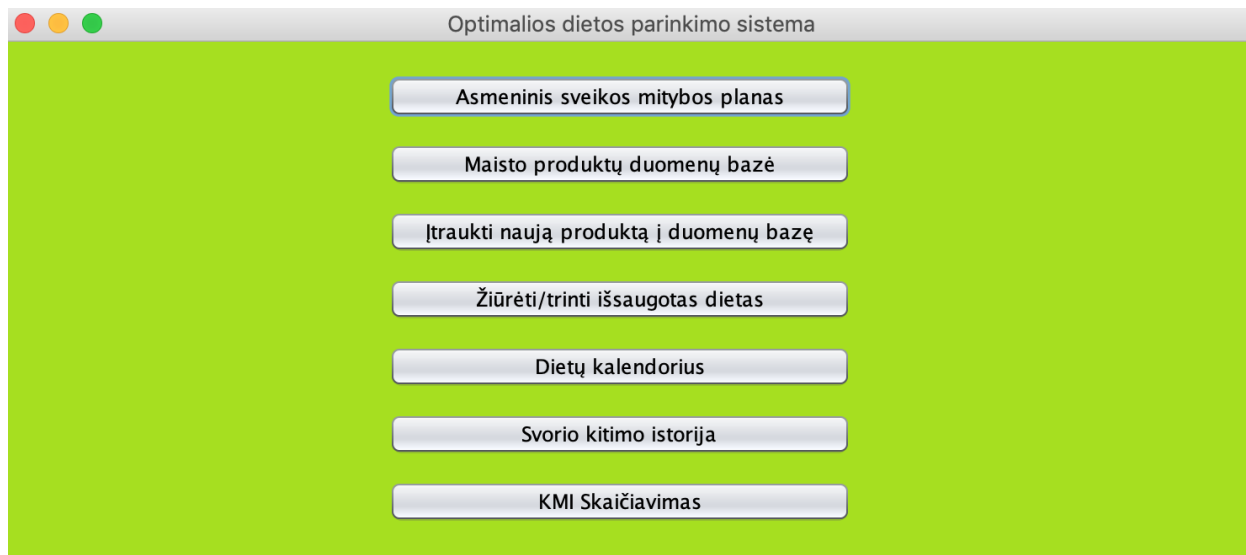
Nors konkrečių sistemos tobulinimo darbų šiuo metu nėra numatyta, galimybė tęsti pradėtą darbą nėra atmetama. Jeigu darbą visgi būtų nuspręsta pratęsti, reikėtų pagalvoti apie „Išvados ir rekomendacijos“ skyriuje paminėtą sistemos prijungimą prie interneto ir didelės, nutolusios duomenų bazės integraciją. Taip pat duomenų bazę galima padaryti detalesnę, t.y. įtraukti į ją daugiau mineralų ir vitaminų, nes kol kas duomenų bazėje saugoma tik pagrindinė, tačiau ne visa maisto produktų informacija. Mažiau detali duomenų bazė yra iš dalies palankesnė naudotojui, kuris įvedinėja naujus maisto produktus rankiniu būdu, tačiau integravus į sistemą interneto ryšį ir nutolusią duomenų bazę, šis patogumas taptų mažiau aktualus.

Literatūros šaltiniai

- [1] Maisto bazė, 2019.
<http://foodbase.azurewebsites.net>.
- [2] ACCESSDATA.FDA.GOV. Cholesterol, 2019.
https://www.accessdata.fda.gov/scripts/InteractiveNutritionFactsLabel/cholesterol.html?fbclid=IwAR3SeenG_6Hcxfr1t8v3bL9DCbHY14pFsTwpuMf087THa6BedpANHtyE8M.
- [3] FIBERFACTS.ORG. Recommended Daily Intake, 2017.
<https://fiberfacts.org/consumer-recommended-daily-intake/?fbclid=IwAR0keOusp2z5TeJ4GP328zu4nZaozgycaAmhHPYgBLQKJTrbNYRk7wyUug>.
- [4] JFreeChart, 2017.
<http://www.jfree.org/jfreechart/>.
- [5] National Institutes of Health Office of Dietary Supplements. Calcium, 2018.
https://ods.od.nih.gov/factsheets/Calcium-HealthProfessional/?fbclid=IwAR0T5PN7RzED9tuPHexv8AyNDDVePSXPt9p14tJiXzLpESp6P5NqcS_ew7Y.
- [6] Alina Kožajeva. PAROS ENERGIJOS (MAISTINIŲ MEDŽIAGŲ) POREIKIO SKAIČIAVIMAS, 2008.
https://maistassportui.lt/straipsniai/ivairus/paros-energijos-maistiniu-medziagu/?fbclid=IwAR2GBU-yUZc4rQWf9Bii3Qc2TjbynVt_ToFgkf5rLgqFYvx1x1jblYlhvSQ.
- [7] Wikipedia. SQLite, 2019.
<https://en.wikipedia.org/wiki/SQLite>.
- [8] NETBEANS.ORG. An Introduction to NetBeans, 2019.
<https://netbeans.org/about/index.html>.

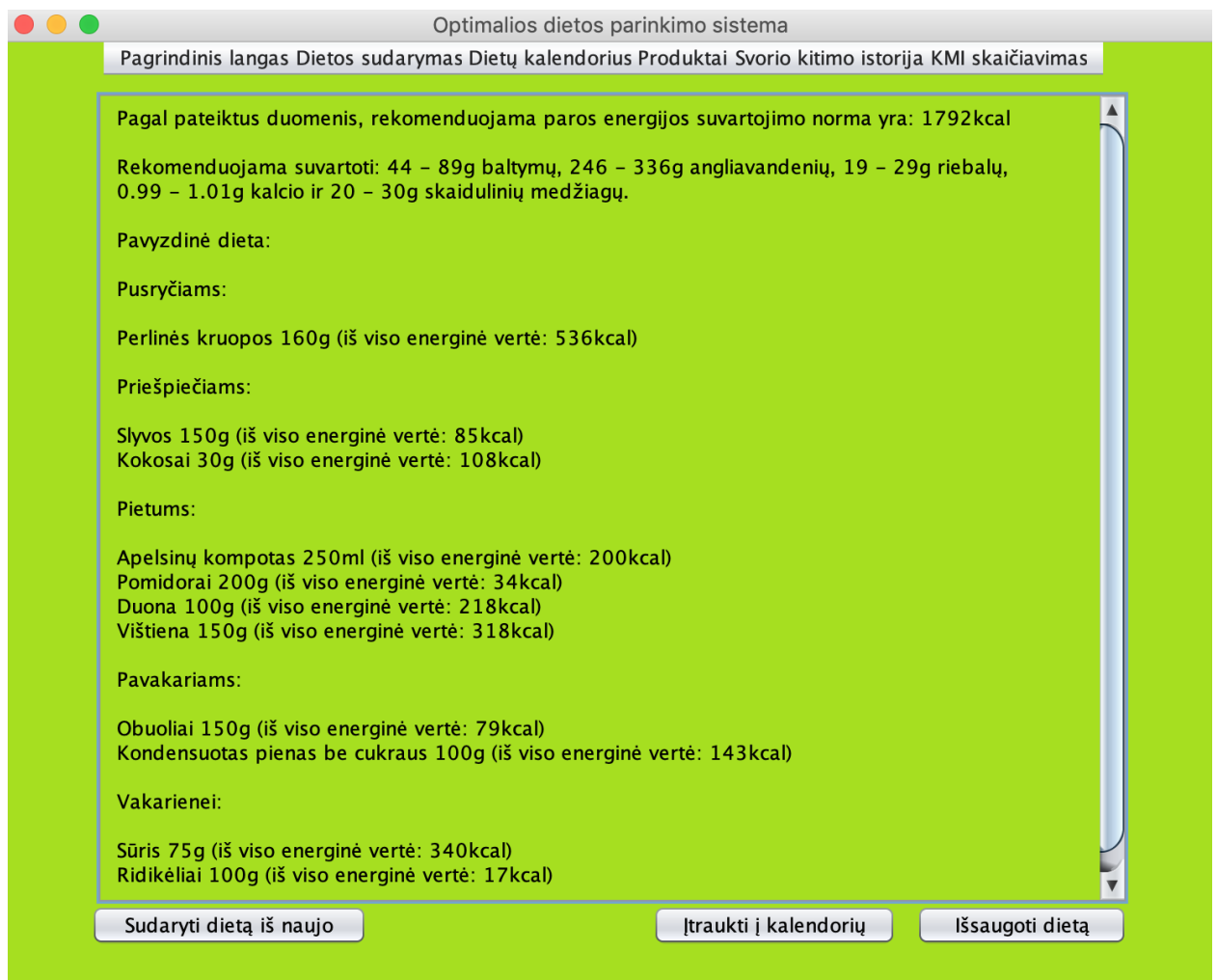
Priedai

A. Optimalios dietos parinkimo sistemos pagrindinis langas



4 pav.

B. Automatiškai sugeneruota dieta



5 pav.

C. Savarankiškas dietos generavimas

Savarankiškas dietos generavimas

Pagrindinis langas Dietos sudarymas Dietų kalendorius Produktai Svorio kitimo istorija KMI skaičiavimas

Pagal pateiktus duomenis, rekomenduojama paros energijos suvartojimo norma yra: 2162kcal

Rekomenduojama suvartoti: 54 – 108g baltymų, 297 – 405g angliavandenių, 24 – 36g riebalų, 0.99 – 1.01g kalcio ir 25 – 35g skaidulinių medžiagų.

Išjungti pavadinimo spėjimą

Įveskite naujo produkto pavadinimą: Produktų informacija

Produkto kiekis: g arba ml

Sudaroma dieta:

Pavadinimas	Kiekis (g arba ml)	Energinė vertė (kcal)	Baltymai (g)	Angliavandeniai (g)	Riebalai (g)	Cholesterolis (g)	Kalcis (g)	Skaidulinės medži...
Manų kruopos	150	540	15.45	112.2	1.5	0	0.02	3.6
Bananai	120	116	1.44	27.72	0.36	0	0.01	2.52
Ananasų sultys	250	125	0.75	28.5	0.25	0	0.05	0.25
Duona	100	218	7.7	52.5	1.4	0	0.05	10
Lapinių salotų ir ag...	150	90	2.19	4.08	7.76	0.02	0.08	1.38
Jautiena	150	253	49.64	1.55	5.49	0.14	0.04	0.33
Obuolių sultys	250	110	0.25	25	0.25	0	0.02	0
Razinos	30	84	0.81	21.39	0.18	0	0.02	2.91
Kefyras	250	150	8.5	12.25	6.25	0.03	0.3	0
Ryžių košė	270	380	8.21	70.96	8.24	0.02	0.05	2.08
Sūris	25	113	10.38	0.03	8	0.02	0.32	0

Rekomenduojama kilokalorijų norma viršyta: 17kcal

Baltymų norma patenkinta idealiai.

Angliavandenių norma patenkinta idealiai.

Rekomenduojama riebalų norma viršyta: 3.68g

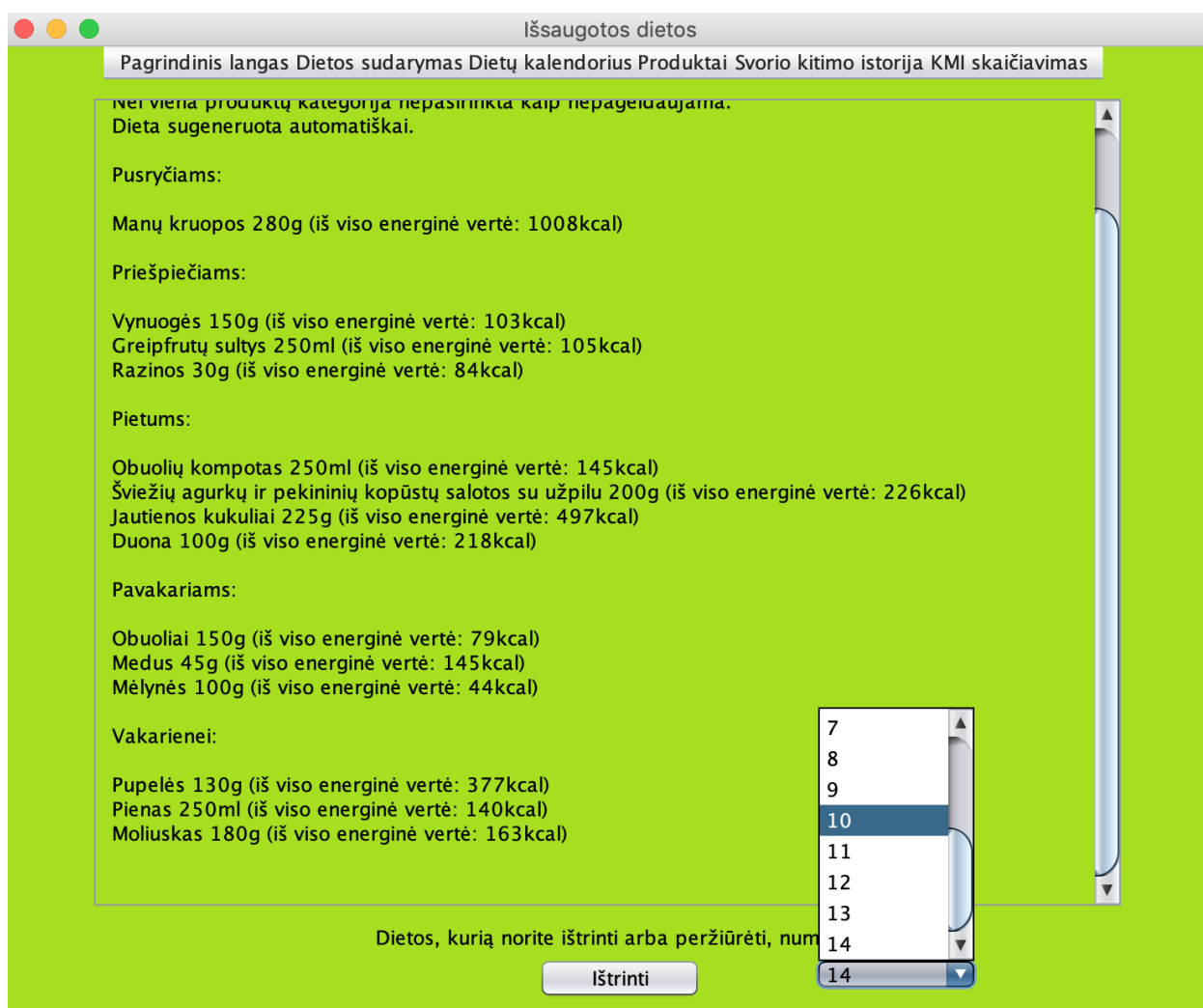
Iki rekomenduojamos kalcio normos trūksta: 0.03g

Iki rekomenduojamos skaidulinių medžiagų normos trūksta: 1.93g

Cholesterolio norma neviršyta (iki pertekliaus trūksta: 0.07g)

6 pav.

D. Išsaugotų dietų peržiūros langas



7 pav.

E. Dietų kalendorius

Dietų kalendorius

Pagrindinis langas Dietos sudarymas Dietų kalendorius Produktai Svorio kitimo istorija KMI skaičiavimas

2019, Birželis

Se	Pr	An	Tr	Kt	Pn	Št
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

Atlaisvinti visą mėnesį

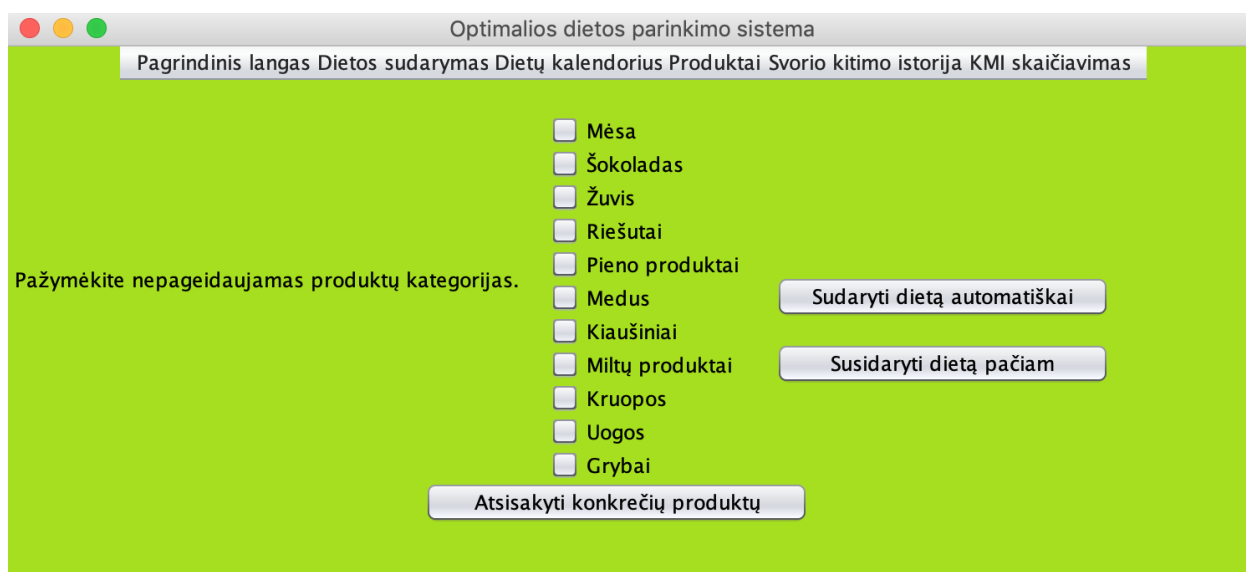
Pasirinkta diena: 2019, Birželis, 7d.

Žiūrėti dietą

Atlaisvinti šią dieną

8 pav.

F. Nepageidaujamų maisto produktų kategorijų atsisakymas



Optimalios dietos parinkimo sistema

Pagrindinis langas Dietos sudarymas Dietų kalendorius Produktai Svorio kitimo istorija KMI skaičiavimas

Pažymėkite nepageidaujamas produktų kategorijas.

- ☐ Mėsa
- ☐ Šokoladas
- ☐ Žuvis
- ☐ Riešutai
- ☐ Pieno produktai
- ☐ Medus
- ☐ Kiaušiniai
- ☐ Miltų produktai
- ☐ Kruopos
- ☐ Uogos
- ☐ Grybai

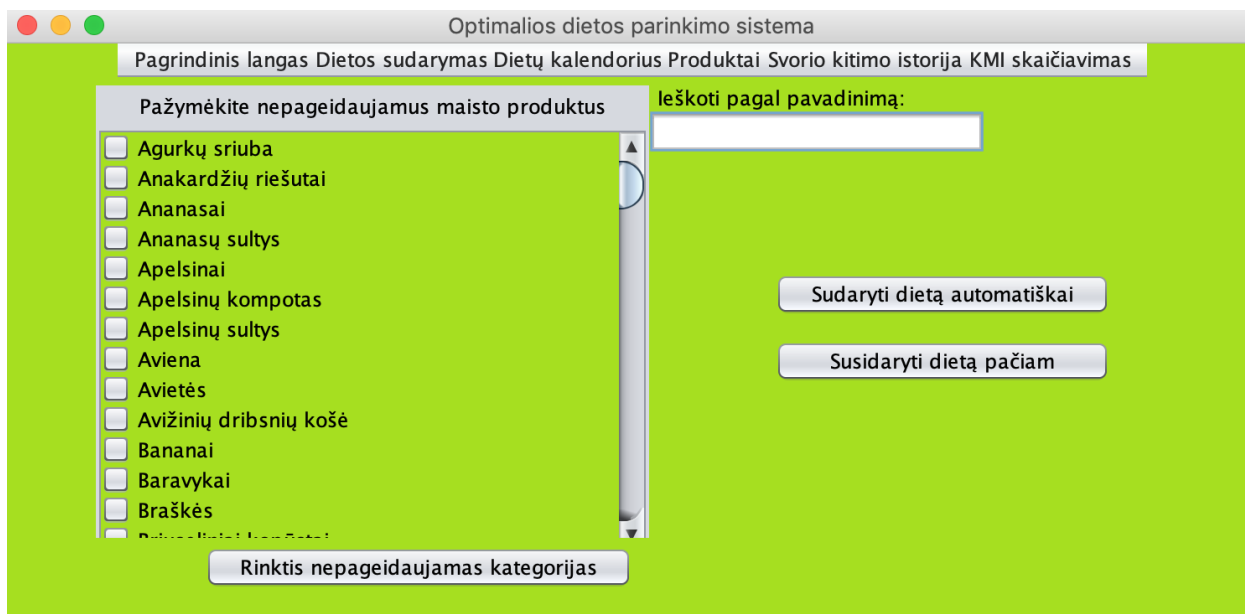
Sudaryti dietą automatiškai

Susidaryti dietą pačiam

Atsisakyti konkrečių produktų

9 pav.

G. Konkrečių nepageidaujamų maisto produktų atsisakymas



10 pav.

H. Maisto produktų duomenų bazės peržiūros ir valdymo langas

Informacija apie maisto produktus

Pagrindinis langas Dietos sudarymas Dietų kalendorius Produktai Svorio kitimo istorija KMI skaičiavimas

Ieškoti:

Pavadinimas	Kilokalorijos (100g)	Baltymai (%)	Angliavandeniai (%)	Riebalai (%)	Cholesterolis (%)	Kalcis (%)	Skaidulinės medžiago...
Juodasis šokoladas	570	6.7	56.6	34.3	0	0.04	1.7
Pieninis šokoladas	565	9.8	54.7	32.8	0.03	0.29	0.7
Ikrai (raudonieji ir juo...	267	24.6	1.9	17.9	0.59	0.28	0
Daržovių salotos (agu...	84	0.88	4.23	7.59	0	0.02	1.21
Šviežių agurkų ir peki...	113	1.43	4.25	10.2	0	0.05	0.45
Šviežių kopūstų salot...	71	1.38	6.88	5.18	0	0.05	2.11
Šviežių kopūstų salot...	89	1.29	7.99	6.67	0.01	0.04	1.9
Šviežių pomidorų sal...	75	1.36	4.88	6.2	0.02	0.03	1.51
Šviežių pomidorų sal...	110	1.09	4.97	10.22	0	0.02	1.48
Morkų salotos su Čes...	91	1.11	8.95	6.69	0.01	0.04	2.45
Morkų ir salierų salotos	78	0.96	9.46	5.13	0.01	0.04	2.36
Lapinių salotų ir agur...	60	1.46	2.72	5.17	0.02	0.05	0.92
Ridikėlių salotos su g...	87	1.69	4.12	7.61	0.02	0.07	1.2
Žiedinių kopūstų salo...	138	1.6	5.25	12.95	0	0.03	1.57
Ryžių ir kumpio salotos	153	7.23	12.57	8.41	0.03	0.02	0.42
Ryžių salotos su krab...	141	7.93	11.8	7.56	0.03	0.02	1.42
Ridikai	16	1	4.1	0.1	0	0.05	1.6
Ridikėliai	17	1.1	3.9	0.1	0	0.05	1.6
Migdolai	580	20	20.5	52	0	0.28	12.9
Žuvų maltinukai	221	17.27	11.4	12.39	0.1	0.04	1.32
Jautienos kukuliai	221	15.64	14.3	11.81	0.04	0.03	1.35
Makaronų apkepas	167	4.4	27.39	4.94	0.03	0.03	1.01
Varškės apkepas	251	16.67	20.18	11.78	0.06	0.09	0.42
Varškės ir ryžių apke...	160	5.13	23.68	5.62	0.04	0.04	1.31
Greipfrutų sultys	42	0.5	9.2	0.1	0	0.01	0.1
Morkų sultys	45	0.4	11.3	0.1	0	0.01	1.3
Obuolių sultys	44	0.1	10	0.1	0	0.01	0
Pomidorų sultys	14	0.8	2.6	0.1	0	0.01	0.4
Vynuogių sultys	65	0.6	15	0	0	0.02	0.5
Medus	324	0.3	81.7	0	0	0.01	0
Margasis upėtakis	111	18.2	0.1	4.2	0.06	0.01	0
Vaivorykštinis upėtakis	161	18.6	0	9.6	0.07	0.02	0
Apelsinų kompotas	80	0.24	19.56	0.06	0	0.02	0.54
Mandarinų kompotas	49	0.2	12.57	0.08	0	0.02	0.52

11 pav.

I. Naujo maisto produkto įtraukimo į duomenų bazę langas

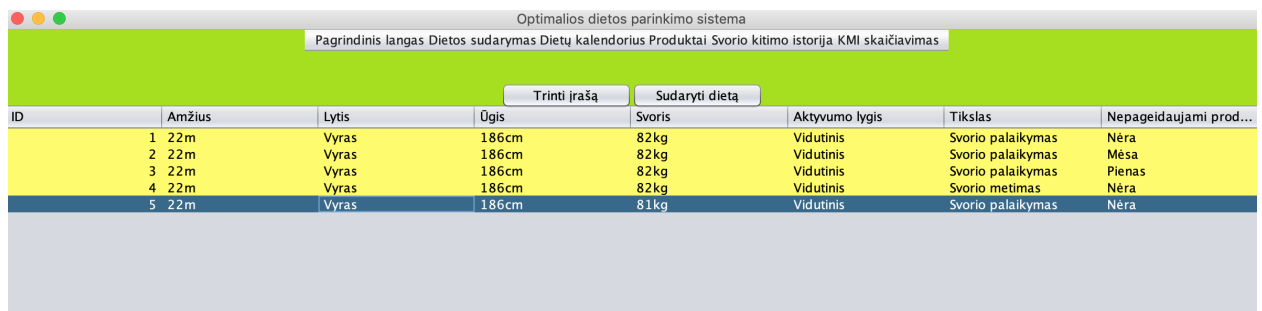
Įveskite kilokalorijų ir maistinių medžiagų kiekius 100g produkto

Pagrindinis langas Dietos sudarymas Dietų kalendorius Produktai Svorio kitimo istorija KMI skaičiavimas

Produkto pavadinimas:	Kilokalorijos (kcal):	Baltymai (g):	Angliavandeniai (g):	Riebalai (g):	Kalcis (g):	Skaidulinės medžiagos (g):
Cholesterolio kiekis (g):	Stand. porcija (g arba ml):	Ar produktas yra skystas?	Paspauskite kategorijai pasirinkti		Paspauskite valgio metui pasirinkti	
		☐ Taip ☐ Ne	Produkto kategorija		Tinkamas metas valgyti	

12 pav.

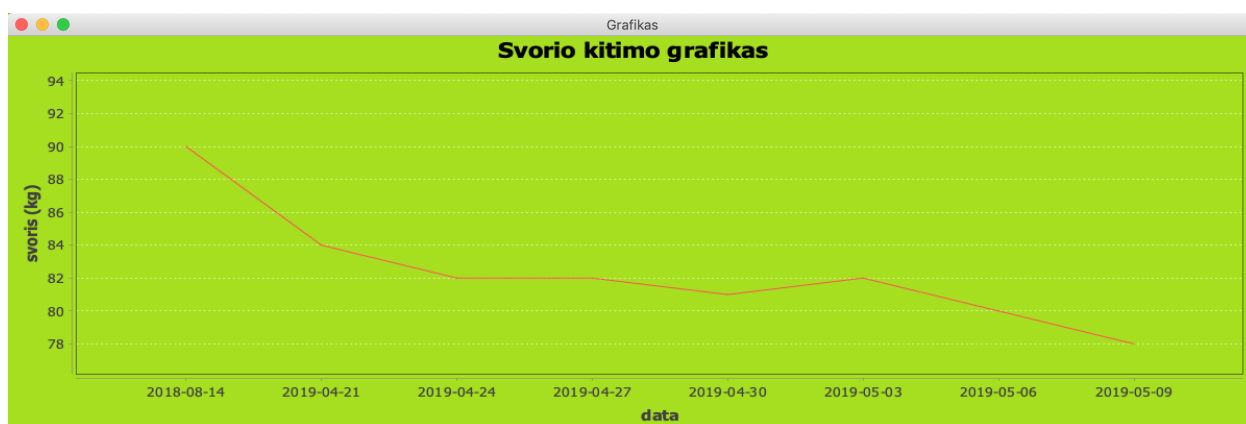
J. Išsaugoti anksčiau naudotojo įvesti duomenys



ID	Amžius	Lytis	Ūgis	Svoris	Aktyvumo lygis	Tikslas	Nepageidaujami prod...
1	22m	Vyras	186cm	82kg	Vidutinis	Svorio palaikymas	Nėra
2	22m	Vyras	186cm	82kg	Vidutinis	Svorio palaikymas	Mėsa
3	22m	Vyras	186cm	82kg	Vidutinis	Svorio palaikymas	Pienas
4	22m	Vyras	186cm	82kg	Vidutinis	Svorio metimas	Nėra
5	22m	Vyras	186cm	81kg	Vidutinis	Svorio palaikymas	Nėra

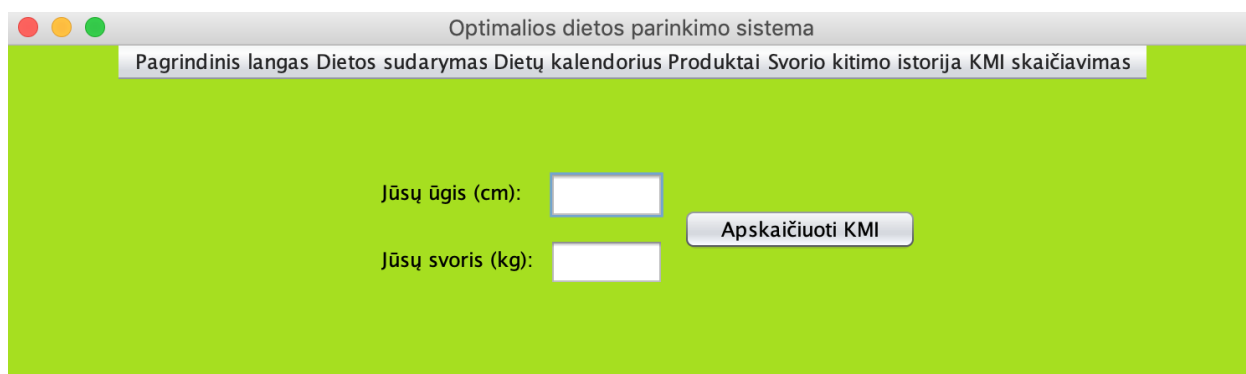
13 pav.

K. Svorio kitimo grafikas



14 pav.

L. Kūno masės indekso skaičiavimas



Optimalios dietos parinkimo sistema

Pagrindinis langas Dietos sudarymas Dietų kalendorius Produktai Svorio kitimo istorija KMI skaičiavimas

Jūsų ūgis (cm):

Jūsų svoris (kg):

Apskaičiuoti KMI

15 pav.