### RĪGAS 64. VIDUSSKOLA

**Veselīgas ēšanas nedēļas plāna veidošanas sistēma**

### Programmatūras prasību specifikācija

Autors: **Aleksandrs Baranovskis**

Darba vadītājs: Programmēšana II sk. **Edvards Bukovskis**

### RĪGA 2023

# Saturs

[Saturs 2](#_TOC_250023)

1. [Ievads… 3](#_TOC_250022)
   1. Nolūks 3
   2. [Mērķis 3](#_TOC_250021)
   3. [Darbības sfēra 3](#_TOC_250020)
   4. [Pārskats 3](#_TOC_250019)
2. [Problēmas izpēte un analīze 4](#_TOC_250018)
   1. [Problēmas apraksts 4](#_TOC_250017)
   2. [Automatizācijas risinājumi 4](#_TOC_250016)
   3. [Izpētes metodes 4](#_TOC_250015)
   4. [Izpētes process 5](#_TOC_250014)
3. [Programmatūras prasību specifikācija 6](#_TOC_250013)
   1. [Programmatūras produkta mērķauditorija 6](#_TOC_250012)
   2. [Mērķauditorijas vajadzību raksturojums 6](#_TOC_250011)
   3. [Programmatūras produkta apraksts 6](#_TOC_250010)
   4. [Programmatūras produkta funkciju apraksts 7](#_TOC_250009)
   5. Programmas produkta skice 9
4. Programmatūras izstrādes plāns 10
   1. [Metodes izvēle 10](#_TOC_250008)
   2. [Izstrādes plāns 10](#_TOC_250007)
5. [Atkļūdošanas un akceptēšanas pārskats 11](#_TOC_250006)
   1. [Testēšanas plāns un izpilde 11](#_TOC_250005)
   2. Akceptēšanas plāns 13
6. [Lietotāja ceļvedis 14](#_TOC_250004)
   1. Lietotāja ceļvedis 14
7. [Piemērotās licences pamatojums 15](#_TOC_250003)
   1. [Produkta licence 15](#_TOC_250002)

[Izmantotā literatūra 16](#_TOC_250001)

[Pielikumi… 17](#_TOC_250000)

* 1. **Nolūks**

# Ievads

Dokuments ir sagatavots Programmēšana II mācību priekšmeta eksāmena pielaides darba ietvaros. Dokumentā tiek aprakstīta Veselīgās ēšanas nedēļas plāna veidošanas sistēmas (turpmāk, VĒNPVS) struktūra un funkcionalitāte, kā arī, dokuments sevī ietver autora gatavā produkta redzējumu un tā pirmatnējo tehnisko izpildījumu.

## Mērķis

Darba mērķis ir izveidot sistēmu, kas ievadot nepieciešamo dienas kaloriju daudzumu, diētas, alerģijas u.c. parametrus automātiski izveido nedēļas ēšanas plānu ar 3 maltītēm dienā (brokastis, pusdienas, vakariņas.) Kopā ar šo plānu arī būs pilns saraksts ar nepieciešamajām sastāvdaļām, lai izveidotu visas dotās maltītes, kā arī pie katras maltītes būs pieejama saite uz tīmekļa vietni, kurā būs rakstīts maltītes īss apraksts ar galveno informāciju, nepieciešamo sastāvdaļu un virtuves piederumu saraksts un instrukcijas maltītes pagatavošanai.

## Darbības sfēra

VĒNPVS ir paredzēts ikdienišķai lietošanai, lai atvieglotu dienas un nedēļas plāna

veidošanu un vienlaicīgi uzturot veselīgu dzīves veidu.

## Pārskats

Pirmajā daļā ir problēmas izpēte un analīze, kas ietver izpētes metodes izvēli un pamatojumu, izpētes procesa aprakstu, izpētes datu apkopojumu.

Otrā daļa iekļauj programmatūras prasību specifikāciju, kas iekļauj mērķauditorijas izvēli un tās raksturojumu, programmatūras produkta un tā funkcijas aprakstu, programmatūras skici.

Trešā daļa iekļauj programmatūras izstrādes plānu.

Ceturtā daļa iekļauj atkļūdošanas un akceptēšanas pārskatu. Piektā daļa iekļauj lietotāja ceļvedi.

Sestā daļa iekļauj piemērotās licences pamatojumu.

# Problēmas izpēte un analīze

## Problēmas apraksts

Diemžēl mūsdienu pasaule ir pielāgojusies tādai pārtikas produktu patēriņa sistēmai, kurai ir vairākas negatīvas sekas uz cilvēka veselību. Dzīvesveida izmaiņas ir tik ļoti piespiedušas mūs, ka cilvēkam ir tik maz laika, lai patiešām padomātu, vai tas, ko mēs ēdam, ir veselīgs uzturs! Globalizācija ir nopietni ietekmējusi cilvēka ēšanas paradumus un piespieda daudzus cilvēkus ēst greznus un kaloriju bagātus ātrās ēdināšanas produktus, ko tautā dēvē par ātro pārtiku. Pētījumi par iespējamo kaitīgumu veselībai, kas saistīts ar šādu ļoti kaloriju pārtikas produktu lietošanu uzturā, ir devuši ieskatu, kā no tiem izvairīties, bet diemžēl veiktie pasākumi nav tik efektīvi, kā būtu nepieciešams. Tādas saslimšanas kā aptaukošanās, saindēšanās ar pārtiku, dehidratācija, sirdsdarbības traucējumi, cukura diabēts un artrīts jaunattīstības valstīs ir ievērojami palielinājušās, un šādu neveselīgu un pārstrādātu pārtiku, kā arī augstu kaloriju patēriņu, kas satur daudz tauku, ir ievērojami veicinājuši.[1]

## Automatizācijas risinājumi

Viens no automatizācijas risinājums ir izveidot programmu, kas automātiski izveido veselīgu nedēļas ēšanas plānu, kurš tiek izveidots atkarībā no padotajiem parametriem, piemēram, kaloriju daudzums un alerģijas. Lai palīdzētu cilvēkiem arvien vairāk izvairīties no ātrā ēdiena, tad vēl var klāt pievienot instrukcijas maltītes pagatavošanai, kas motivētu cilvēkus pašiem taisīt savas maltītes, un tās ņemt līdzi tā vietā, lai pasūtītu ātro pārtiku.

Vēl viens automatizācijas risinājums ir programma, kas atrod neveselīgai ātrajai pārtikai alternatīvas, kuras būtu veselīgas un vielu bagātākas vienlaicīgi nezaudējot ātrās pagatavošanas un patērēšanas aspektus. Programmas varētu pasniegt atbildi atkarībā no lietotāja ievadītās pārtikas un atrastu tam līdzīgu bet veselīgāku alternatīvu.

Šī darba izveidei tika izvēlēta nedēļas ēšanas plānas izveides risinājums, jo pēc darba autora domām, tā ne tikai atrisinātu aprakstīto problēmu, bet arī veicinātu veselīgas dzīves veida uzturēšanu, kā arī doto iespēju lietotājam apgūt ēdiena gatavošanas principus.

## Izpētes metodes

Priekš izpētes tika izmantota novērojuma metode. Šī metode tika izmantota, jo šī problēma ir masveida, bet tā arī vispārīga un cilvēki, kas cieš no šīs problēmas neseko tai līdzi, kas nozīmē, ka intervijas metode nebūtu noderīga, jo aptaujājamie nevarētu sniegt specifiskas atbildes un problēmas jautājumiem. Tā kā šī problēma ir dziļi saistīta ar medicīnas nozari, kas ir augsta līmeņa nozare, tad priekšizpēte būtu sarežģīta. Problēmas ar šo metodi, ir ka daļa ir balstīta uz spekulācijām, bet ar izvēlēto risinājumu, tā nebūs problēma.

## Izpētes process

Pirmajā solī tika veikta sagatavošanās, kurā tika atrasti nepieciešamie informācijas avoti, no kuriem tiks veikti novērojumi.

Otrais solis bija norise, kuras laikā tika veikti novērojumi par galvenajām problēmām, faktiem un viedokļiem, kas bija saskatāmi atrastajos informācijas avotos.

Trešais solis bija rezultātu apkopošana, kurā iegūtie rezultāti tika sadalīti apakšgrupās

ietverot katru problēmas daļu, kuru varētu uzlabot ar programmas palīdzību.

Ceturtais solis ir izvērtēšana, kurā tika izskatīta katras apakšgrupas risināšanas svarīgums, un kā tos varētu risināt ar programmas palīdzību.

Pēdējais solis bija secinājumu izdarīšana, kur tika veikti secinājuma par problēmu, to apakšnodaļu atrisinājuma veidiem un kādas funkcionalitātes programma varētu veikt, lai risinātu problēmu.

# Programmatūras prasību specifikācija

## Programmatūras produkta mērķauditorija

Viena no programmatūras mērķauditorijām būtu studenti vecumā no 20 līdz 23, jo studentiem bieži ir jāapvieno mācības kopā ar darbu, kā arī iespējams ar sporta nodarbībām un citām aktivitātēm, kas bieži noved pie tā, ka netiek izveidots stabils dienas vai nedēļas plāns, kas parasti nozīme, ka vai nu tiek izlaistas ēdienreizes, vai nu maltītēs tiek izmantota ātrā pārtika.

Otra no mērķauditorijām galvenokārt būtu visi strādājošie cilvēki ar lielu darba daudzumu, bet precīzāk tie būtu darbinieki vecumā no 25 līdz 35 gadi, jo cilvēkiem šajos gados parasti ir jauni savās darba vietās, kas nozīmē, ka viņiem parasti ir liela slodze, jo ir daudz jāstrādā, lai varētu tikt augstākās darba pozīcijās. Tas atkal rada iepriekš minēto problēmu, kur cilvēks vai nu neēd vai izmanto ātro pārtiku.

## Mērķauditorijas vajadzību raksturojums

Galvenā mērķauditorijas vajadzība būtu programma, kas varētu automātiski izveidot pilnvērtīgu nedēļas ēšanas plānu, lai viņiem tas nebūtu jādara manuāli, kas var būt laikietilpīgi. Lietotājiem ari vajag iespēju no maltītēm izslēgt kādas sastāvdaļas pret kurām lietotājam varētu būt alerģija, kā arī jau dot visu nepieciešamo sastāvdaļu sarakstu, lai lietotājam nebūtu jātērē laiks veidojot sastāvdaļu sarakstu un dotu iespēju nopirkt visas sastāvdaļas vienā reizē nevis vairākās, kā arī likvidētu vajadzību aprēķināt izmaksas. Kā arī programmai vajag spēt iekļauties noteiktā budžetā, jo studentiem ari var būt limitēts naudas daudzums, kas varētu neļaut tiem atļauties visas iespējamās maltīšu opcijas, ko programma piedāvās. Tā kā daudzas no piedāvātajām maltītēm var būt lietotājam jaunas, tad arī būs nepieciešamība priekš pagatavošanas instrukcijas un maltītes aprakstam.

## Programmatūras produkta apraksts

VĒNPVS ir programma, kurā lietotājam vajadzēs izveidot savu profilu un tā atkarībā no lietotāja ievadītajiem datiem un filtriem izveido nedēļas ēšanas plānu, kas sevī iekļaus 3 maltītes dienā. Dati, kas lietotājam būs jāievada ir kaloriju daudzums dienā, budžets, pēc kuras diētas ēšanas plāns tiks veidots, ja tāda ir un pret kurām sastāvdaļām lietotājam ir alerģija.

Lietotājs arī spēs redzēt sastāvdaļu daudzumu, kas būs nepieciešams visai nedēļai, kā arī kopējo cenu. Tajā pašā vietā varēs arī atzīmēt, kuras sastāvdaļas un cik liels daudzums jau ir pieejams.

Pie katras maltītes arī būs pieejama mājaslapa, kurā varēs izlasīt īsu maltītes raksturojumu, pie kādām diētām tas skaitās, cena par maltīti, pagatavošanas ilgumu, sastāvdaļu sarakstu, pagatavošanas instrukciju, cenas sadalījumu un informāciju par uzturvērtību.

Šīs programmatūras izveidei ir nepieciešams dators uz kura var kodēt. Programmatūras izveidei tiks izmantota Python programmēšanas valoda. Python valodā tiks izmantotas: requests, os, json, getpass un bcrypt bibliotēkas. Priekš recepšu iegūšanas un plānas veidošanas tiks izmantots Spoonacular API.

## Programmatūras produkta funkciju apraksts

#### Profila izveides funkcijas apraksts

*Tabula 3.1.*



|  |  |
| --- | --- |
| **Mērķis:** |  |
| Ļaut lietotājam izveidot profilu, kur viņa nedēļas plāns un pārējā informācija tiks saglabāta. |
| **Ievaddati:** |
| Profila lietotājvārds, parole un parole atkārtoti. |
| **Apstrāde:** |
| Tiek pārbaudīts, vai paroles ir vienādas, datu bāze tiek saglabāts lietotājvārds un šifrēta  parole. |
| **Izvaddati:** |
| Ja paroles ir vienādas, tad tiek izvadīts teksts “Profile creation successful!” un lietotāja informācija tiek saglabāta datubāzē.  Ja paroles nav vienādas, tad tiek izvadīts teksts “Passwords don’t match! Try again!” un  lietotājam tiek dota iespēja atkal ievadīt paroli. |
| **Dienas kaloriju daudzuma ievades funkcijas apraksts** *Tabula 3.2.* | |
| **Mērķis:** |  |
| Ļaut lietotājam izvēlēties, vidējo kaloriju daudzumu dienā, kāds būs nedēļas plāna maltītēm. |
| **Ievaddati:** |
| Kaloriju daudzums dienā. |
| **Apstrāde:** |
| Pie Spoonacular API parametriem tiks pievienots kaloriju dadzums un atbilstoši tabula tiks  veidota pēc dotā dadzuma. |
| **Izvaddati:** |
| Ja tiek ievadīts skaitlis, tad plānā dienas kaloriju daudzums būs tuvs ievadītajam skaitlim. Ja netiek ievadīts skaitlis, tad tiek izvadīts teksts “Invalid input! Try again!” un lietotājs  ievada daudzumu vēlreiz. |

#### Diētas izvēles funkcijas apraksts



*Tabula 3.3.*

|  |
| --- |
| **Mērķis:** |
| Ļaut lietotājam izvēlēties, kādai diētai sekos plāns. |
| **Ievaddati:** |
| Vai nu, ka nesekos diētai, vai nu diētas nosaukums. |
| **Apstrāde:** |
| Pie Spoonacular API parametriem tiks pievienots diēta un atbilstoši plāns tiks veidota pēc  dotās diētas. |
| **Izvaddati:** |
| Ja tiek ievadīts, ka plāns ies pēc diētas, tad:   * ja tiek ievadīta diēta no saraksta, tad tiek izvadīts teksts “Selected diet: (izvēlētā diēta)” un plāns tiks veidots pēc noteiktās diētas; * ja tiek ievadīts kaut kas, kas nav no dotajām diētām, tad tiek izvadīts teksts “Not an option! Try again!” un lietotājs izvēlas diētu vēlreiz.   Ja tiek ievadīts, ka plāns neies pēc diētas, tad tiek izvadīts teksts “No diets selected!” un plāns nesekos diētām.  Ja netiek ievadīts ne viens, ne otrs, tad tiek izvadīts teksts “Invalid answer!” un lietotājam  atkal jāizvēlas, vai plāns sekos diētai, vai nē. |

#### Alerģiju izvēles funkcijas apraksts



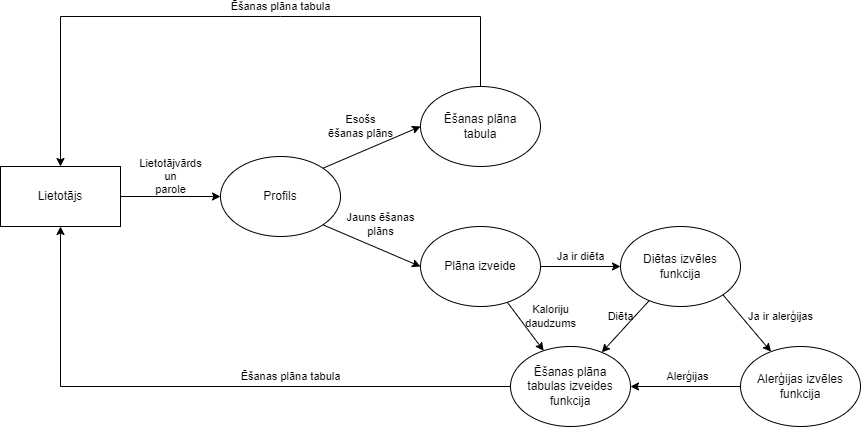
*Tabula 3.4.*

|  |
| --- |
| **Mērķis:** |
| Ļaut lietotājam izvēlēties, vai kādas sastāvdaļas netiks izmantotas. |
| **Ievaddati:** |
| Vai nu, ka alerģiju nav, vai nu, ka alerģiju ir un kādas sastāvdaļas neiekļaut. |
| **Apstrāde:** |
| Pie Spoonacular API parametriem tiks pievienotas kādas sastāvdaļas izslēgt no maltītēm un  atbilstoši plāns tiks veidots bez tām sastāvdaļām |
| **Izvaddati:** |
| Ja tiek ievadīts, ka ir alerģijas, tad lietotājam prasīs kādas sastāvdaļas izslēgt un, kad tas ir ievadīts, tad tiks izvadīts teksts “Excluded ingredients: (sastāvdaļas)” un programma tās sastāvdaļas izslēgs no plāna.  Ja tiek ievadīts, ka alerģiju nav, tad tiek izvadīts teksts “No allegies selected!” un plāns  neizslēgs sastāvdaļas. |

|  |  |
| --- | --- |
| Ja netiek ievadīts ne viens, ne otrs, tad tiek izvadīts teksts “Invalid answer!” un lietotājam  atkal jāizvēlas, vai ir alerģijas, vai nē. |  |
| **Plāna izveides funkcijas apraksts** *Tabula 3.5.* | |
| **Mērķis:** |  |
| Ļaut lietotājam izveidot nedēļas ēšanas plānu. |
| **Ievaddati:** |
| Obligātais: kaloriju daudzums dienā. Izvēles: diēta un alerģijas |
| **Apstrāde:** |
| Pie Spoonacular API parametriem tiks pievienoti ievaddati un API atslēga. |
| **Izvaddati:** |
| Nedēļas ēšanas plāna tabula, kurā ir maltītes atkarībā no ievaddatiem, nepieciešamās  sastāvdaļas utt. |



* 1. **Programmatūras produkta skice**



1. **Programmatūras izstrādes plāns**

## Metodes izvēle

Tika izvēlēta spirālveida metode, jo bija zināms, ka būs nepieciešamība aiziet uz kādu no iepriekšējiem izstrādes soļiem, lai vai nu uzlabotu, vai nu mainītu lietas, lai programma būtu efektīvāka.

## Izstrādes plāns

Prasību analīzē tika noskaidrotas pasūtītāja prasības pret programmatūras produktu, galvenās no tām ir sistēmas funkcija, ierobežojumi un raksturojuma apraksti.

Projektēšanas punktā tika izveidots programmatūras modelis balstoties uz prasību analīzes veiktajiem secinājumiem. Galvenais punkta uzdevums bija izveidot uzskatāmu modeli, kas sekoja visām pasūtītāja prasībām.

Programmēšanas cikla galvenais uzdevums ir uzprogrammēt programmatūru, kura sekoja līdzi programmatūras modelim un labas programmēšanas principiem, kā arī izveidot tā, lai būtu pēc iespējas mazāk kļūdu un problēmu.

Testēšanas cikla galvenais uzdevums ir iztestēt visas funkcijas, lai tās strādātu pēc noteiktā plāna un nebūtu kļūdu un nerastos problēmas ar programmatūru.

Dokumentēšanas cikla galvenais uzdevums ir nodokumentēt katru programmas funkciju un izstrādi, lai, gan lietotājiem, gan pasūtītājam būtu zināms par programmas funkcijām un kā programma strādā.

Ieviešanas cikla galvenais uzdevums ir programmu veiksmīgi ieviest tīmeklī, lai lietotāji to varētu izmantot.

# Atkļūdošanas un akceptēšanas pārskats

## Testēšanas plāns un izpilde

Testēšana notiks ievadot ievaddatus un pārbaudot vai tiks izvadīti gaidāmie izvaddati.

#### Profila izveides funkcijas testēšana

|  |  |  |  |
| --- | --- | --- | --- |
| **Ievaddati** | **Sagaidāmaos rezultāts** | **Reālais**  **rezultāts** | **Rezultāta atspoguļojums** |
| Parole un atkārtotā parole, kuri ir vienādi | Programma atgriež “Profile creation successful!” | Izpildās |  |
| Parole un atkārtota parole, kuri nav vienādi | Programma atgriež “Passwords don’t match!  Try again!” | Izpildās |  |

#### Dienas kaloriju daudzuma ievades funkcijas testēšana

|  |  |  |  |
| --- | --- | --- | --- |
| **Ievaddati** | **Sagaidāmais rezultāts** | **Reālais rezultāts** | **Rezultāta atspoguļojums** |
| Skaitlis (šajā gadījumā:  2678) | Programma atgriež “Daily calories:  2678” | Izpildās |  |
| Jebkas, kas nav skaitlis | Programma atgriež “Invalid input! Try again!” | Izpildās |  |

#### Diētas izvēles funkcijas testēšana

*Tabula 5.1.*

*Tabula 5.2.*

*Tabula 5.3.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Ievaddati** | **Sagaidāmais rezultāts** | **Reālais rezultāts** | **Rezultāta atspoguļojums** |
| Atbilde “no” vai “n” | Programma atgriež “No diets selected!” | Izpildās |  |
| Atbilde, kas nav “yes”, “y”, “no” vai “n” | Programma atgriež “Invalid answer!” | Izpildās |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Atbilde “yes” vai “y”, un tad kāda no dotajām izvēlēm | Programma atgriež “Diet selected; (izvēle)” | Izpildās | Text  Description automatically generated |
| Atbilde “yes” vai “y”, un tad izvēle, kas nav dota | Programma atgriež “Not an option, try again.” | Izpildās |  |

#### Alerģiju izvēles funkcijas testēšana

*Tabula 5.4.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Ievaddati** | **Sagaidāmais rezultāts** | **Reālais rezultāts** | **Rezultāta atspoguļojums** |
| Atbilde “no” vai “n” | Programma atgriež “No allergies selected!” | Izpildās |  |
| Atbilde, kas nav “yes”, “y”, “no” vai “n” | Programma atgriež “Invalid anwer!” | Izpildās |  |
| Atbilde “yes” vai “y” un kāda sastāvdaļa | Programma atgriež “Excluded ingredients: (sastāvdaļas)” | Izpildās |  |

#### Plāna izveides funkcijas testēšana

*Tabula 5.5.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Ievaddati** | **Sagaidāmais rezultāts** | **Reālais rezultāts** | **Rezultāta atspoguļojums** |
| Kaloriju daudzums, diēta, alerģijas (piemēram: 2678,  Primal, olives) | Programma izveido JSON tabulu, kurā maltītes seko ievaddatiem | Izpildās | **Bilde pārāk liela** |

* 1. **Akceptēšanas pārskats**

Lietotājs pēc testēšanas bija apmierināts un uzlabojumi nebija vajadzīgi.

* 1. **Lietotāja ceļvedis**

# Lietotāja ceļvedis

* Pirmais solis ir profila izveidošana. Lai to izdarītu Jums ir jāievada lietotājvārds, kuru Jūs tālāk izmantosiet, lai ieietu profilā un paroli.
* Pēc profila izveides Jums būs iespēja izveidot savu ēšanas plāna tabulu. No sākuma

ievadiet dienas kaloriju daudzumu.

* Pēc tam, ja vēlaties, lai plāns seko kādai diētai, tad atbildiet ar “yes” vai “y” pie jautājamu par diētām un izvēlaties kādu no piedāvātajām diētām.
* Tagad atbildat vai Jums ir kāda alerģija, un ja ir, tad uzrakstiet kādas sastāvdaļas neizmantot plāna veidošanā.
* Apsveicu, Jūs esat izveidojuši Jūsu plānu, tagad par papildus funkcijām.
* Ja Jūs uzspiedīsiet uz pogu “Sastāvdaļas”, tad Jūs redzēsiet visas nepieciešamās sastāvdaļas un to atsevišķo un kopējo cenu. Jūs arī varat noņemt no tā saraksta sastāvdaļas, ja Jums jau tās ir vai arī kad Jūs nopērkat.
* Pie katras maltītes ir bilde un poga, kas ved uz mājaslapu, kur ir rakstīts tai maltītei nepieciešamās sastāvdaļas, pagatavošanas instrukcija, apraksts un cita īsa informācija.
* Tās arī būs visas programmas funkcijas, lai Jums veselīga ēšana un jautra gatavošana.

# Piemērotās licences pamatojums

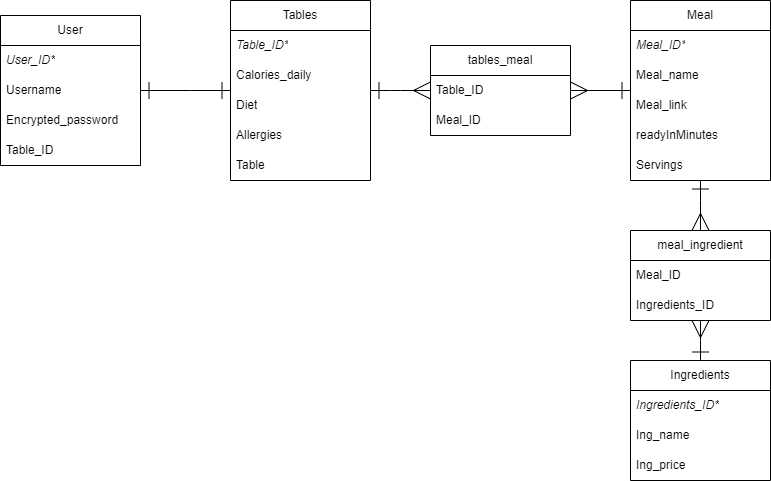
## Produkta licence

Tiktu izmantota Creative Commons licence, jo kamēr autori uzskata, ka šādai programmai vajadzētu būt atvērti pieejamai un galvenajām funkcijām jābūt brīvām, ja tā tiktu uzlikta par atvērta pirmkoda licenci, tad lielākām kompānijām būtu iespēja izmantot un pelnīt naudu no kāda cita idejas un darba. Tāpēc tika izvēlēta šī licence, lai veidotājiem būtu kontrole pār to, kurš un kā šo programmu izmanto, atļaujiet parastiem cilvēkiem izmantot un pat veikt savus uzlabojumus programmai priekš sevis, un vienlaicīgi kontrolētu, lai kompānijas neizmanto šo programmu komerciāli.

# Izmantotā literatūra

* + 1. Modern Diet and its Impact on Human Health. Pieejamība: [https://www.longdom.org/open-access/modern-diet-and-its-impact-on-human-health-](https://www.longdom.org/open-access/modern-diet-and-its-impact-on-human-health-35026.html#%3A~%3Atext%3DOver%20nutrition%2C%20especially%20absorption%20and%2Cobesity%2C%20hypertension%2C%20and%20hyperlipidemia) [35026.html#:~:text=Over%20nutrition%2C%20especially%20absorption%20and,obes](https://www.longdom.org/open-access/modern-diet-and-its-impact-on-human-health-35026.html#%3A~%3Atext%3DOver%20nutrition%2C%20especially%20absorption%20and%2Cobesity%2C%20hypertension%2C%20and%20hyperlipidemia) [ity%2C%20hypertension%2C%20and%20hyperlipidemia](https://www.longdom.org/open-access/modern-diet-and-its-impact-on-human-health-35026.html#%3A~%3Atext%3DOver%20nutrition%2C%20especially%20absorption%20and%2Cobesity%2C%20hypertension%2C%20and%20hyperlipidemia)

# Pielikumi



*Pielikums nr.1. Datu bāzes modelis*

import requests import os import json import getpass import bcrypt

with open("api\_key.json", "r") as openfile: api\_key = json.load(openfile)

api\_key = api\_key["api\_key"]

def sign\_up():

global username, hashedpass

username = input("Enter your username: ") while True:

password = getpass.getpass(prompt = "Enter your password: ") check\_pass = getpass.getpass(prompt = "Re-enter your password: ") if password == check\_pass:

print("Profile creation successful!") break

else:

print("Passwords don't match! Try again!") password = password.encode("utf-8")

hashedpass = bcrypt.hashpw(password, bcrypt.gensalt())

def diets():

global diet

diets = {"Gluten Free":"glutenfree", "Ketogenic":"ketogenic", "Vegetarian":"vegetarian",

"Lacto-Vegetarian":"lactovegetarian", "Vegan":"vegan", "Pescetarian":"escetarian", "Paleo":"paleo",

"Primal": "primal",

"Low FODMAP":"lowfodmap", "Whole30":"whole30"

}

x = json.dumps(diets) x = json.loads(x)

#Diet picker

diet\_options = ("Gluten Free", "Ketogenic", "Vegetarian", "Lacto-Vegetarian", "Vegan", "Pescetarian", "Paleo", "Primal", "Low FODMAP", "Whole30")

input\_message = "Pick an option:\n"

for index, item in enumerate(diet\_options): input\_message += f'{index+1}) {item}\n'

input\_message += 'Your choice: ' while True:

diet = input(input\_message) if diet in diet\_options:

print(f"Selected diet: {diet}") return diet

else:

print("Not an option, try again.")

def allergies():

exclude = input("Write which ingredients to exclude (example of input: olives,shellfish): ")

print(f"Excluded incredients: {exclude}") return exclude

def week\_planner():

url = "https://api.spoonacular.com/mealplanner/generate"

#Calories while True:

calories = input("Enter the amount of calories per day: ") try:

calories1 = float(calories) except:

print("Invalid input! Try again!") else:

calories = str(calories) print(f"Daily calories: {calories}") break

#Diet

while True:

ans = input("Is there a diet you want to follow? (y/n) ") if ans == "y" or ans == "yes":

diet = diets() break

elif ans == "n" or ans == "no": diet = "none"

print("No diets selected!") break

else:

print("Invalid answer!")

#Allergies while True:

ans = input("Do you have any allergies? ") if ans == "y" or ans == "yes":

exclude = allergies() break

elif ans == "n" or ans == "no": exclude = "none"

print("No allergies selected!") break

else:

print("Invalid answer!")

#Website API scrapping query\_params = {

"apiKey": api\_key, "timeFrame": "Week", "targetCalories": calories, "exclude": exclude, "diet": diet

}

spoon\_resp = requests.get(url, params = query\_params) meal\_plan = json.dumps(spoon\_resp.json(), indent=4) meal\_plan = json.loads(meal\_plan)

user\_data = {

f"{username}":{ "user\_data":{

"username":f"{username}", "encrypted\_password":f"{hashedpass}"

},

"plan": {

\*\*meal\_plan

}

}

}

user\_data = json.dumps(user\_data, indent = 4) #print(user\_data)

with open("users.json", "w") as outfile: outfile.write(user\_data)

sign\_up() week\_planner()

*Pielikums nr.2. Programmas kods*