

**ДИПЛОМСКИ ТРУД**

**ВЕБ АПЛИКАЦИЈА - ВИЗУЕЛИЗАЦИЈА НА СОСТОЈКИ**

**ОД ТЕКСТУАЛНИ РЕЦЕПТИ**

Ментор: Изработил:

Д-р Ристе Стојанов Дејан Поповски 171245

Скопје, 2022

# **Содржина**

**[Содржина](#_Toc114861620)** [2](#_Toc114861620)

[**Апстракт** 2](#_Toc114861621)

[**Вовед** 3](#_Toc114861622)

[**Кориснички сценарија** 5](#_Toc114861623)

[Додавање на рецепт 6](#_Toc114861624)

[Прикажување на детали и пресметување на нутритивни вредности на рецепт 8](#_Toc114861625)

[Пребарување на рецепт преку дадена состојка 10](#_Toc114861626)

[Пребарување на рецепт преку ранг на нутритивни вредности 12](#_Toc114861627)

[**Архитектура на решението** 13](#_Toc114861628)

[Presentation 13](#_Toc114861629)

[Business 14](#_Toc114861630)

[Persistence 15](#_Toc114861631)

[Database 16](#_Toc114861632)

[**Користени технологии** 18](#_Toc114861633)

[**Spring boot** 18](#_Toc114861634)

[**React** 18](#_Toc114861635)

[**Semantic Web** 18](#_Toc114861636)

[**PostgreSQL** 19](#_Toc114861637)

[Користење на 3rd party технологии 19](#_Toc114861638)

[Foodviz 19](#_Toc114861639)

[USA National Agriculture Library 22](#_Toc114861640)

[Заклучок 25](#_Toc114861641)

[**Користена литература** 26](#_Toc114861642)

# **Апстракт**

*Целта на овој дипломски труд е изработка на веб апликација користејќи сервиси базирани на вештачка интелигенција и семантички веб со цел добивање на состојките искористени во даден текстуален рецепт за храна. Исто така овозможува креирање, прегледување и филтрирање на рецепти според различни критериуми. Корисниците можат да ги пребаруваат рецептите според состојките кои ги содржат или според опсег на вредноста на макронутриентите кои ги содржи рецептот. При преглед на секој рецепт освен текстот за приготвување прикажани се сите состојките и вредноста на макронутриентите (маснотии, протеини и шеќери) и калориите на секоја состојка и на целиот рецепт соодветно. Дополнително при преглед на рецептот корисниците можат да ги менуваат количините на секоја од состојките по своја жеба при што вредноста на макронутриентите и калориите соодветно ќе се ажурира.*

# **Вовед**

Храната има големо влијание врз нашето здравје. Во зависнот од нутритивните вредности кои ќе ги примиме во текот на денот преку храната, таа има различно влијание на нашето здравје. Можеме и сами да забележиме дека во зависност од типот и количината на храна што ја јадеме нашето расположение варира. Некоја храна не прави полни со енергија и може да ни го подобри расположението, а некоја да има спротивен ефект.

На пример доколку јадеме повеќе овошје и зеленчук, јатки и житарки можеме да го намалиме ризикот на ментални заболувања и депресија, а доколку јадеме храна богата со шеќери и заситени масти ризикот може да се зголеми. Затоа за нашето здравје е многу битно да имаме балансирана диета.

Често домашно готвената храна важи за поздрава. Но тешко е да утврдиме какви хранливи вредности содржи рецептот. Доколку имаме специфични барања за храната и сме во потрага на храна со точни хранливи материи важно е да можеме да ги измериме и да утврдиме која храна одговара на исхраната што ја бараме.

Во изминатите години на светско ниво е внесен голем труд за развој во областа на вештачката интелигенција, особено во областа на анализа на текст. Развиени се системи кои во висока мера разбираат напишан текст и можат да го резонираат контекстот на текстот.

Паралелно со развојот на семантичкиот веб се појавуваат голем број на бази кои содржат семантички податоци кои го определуваат значењето и поврзувањето на податоците.

Овие две технологии нудат можност за нивно комбинирање и со тоа да се направи систем кој разбира дека семантиката на еден текст е рецепт и од него да ги одвои состојките од кои е направен. Моменталниот развој на овие технологии е мотивацијата за конзумирање на ваков сервис и развивање на апликација која нуди можност да препознае кои се вредностите на нуртитивните вредности на даден рецепт.

Во оваа дипломска работа е изработена токму таква апликација која на корисниците нуди можност на креирање и пребарување на рецепти со цел добивање информација за нивните нутритивни вредности

# **Кориснички сценарија**

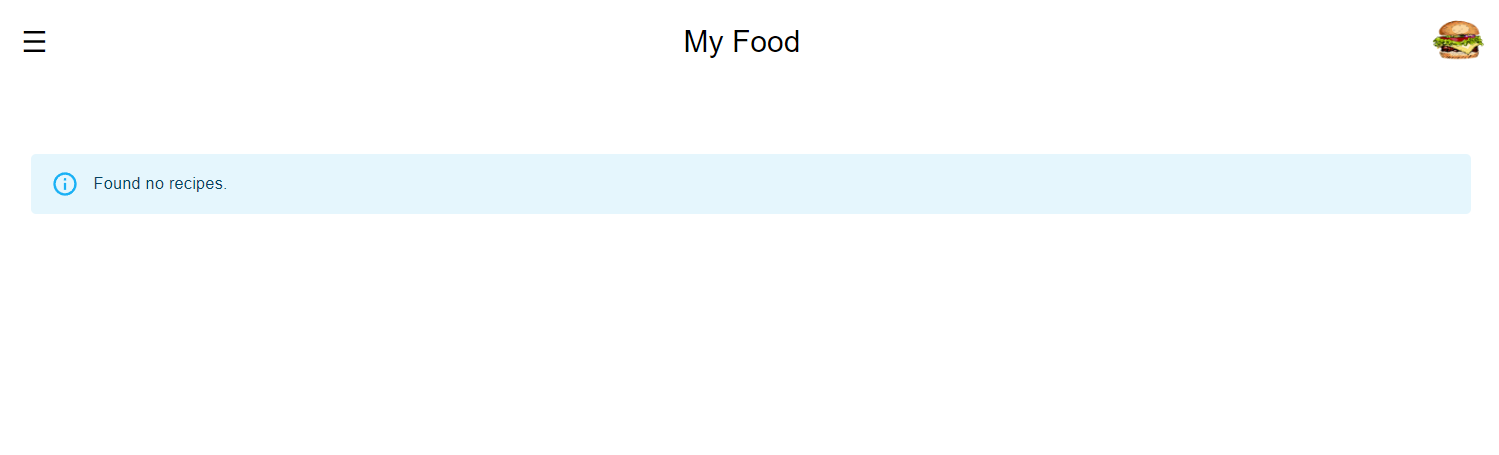
Во продолжение ќе ги разгледаме корисничките сценарија кои се имплементирани во оваа апликација.

1. Додавање на рецепт
2. Прикажување на детали и пресметување на нутритивни вредности на рецепт
3. Пребарување на рецепт преку дадена состојка
4. Пребарување на рецепт преку ранг на нутритивни вредности

При пуштање на апликацијата за прв пат ни се појавува домашната страна на која има известување дека нема пронајдени рецепти поради тоа што базата е празна.

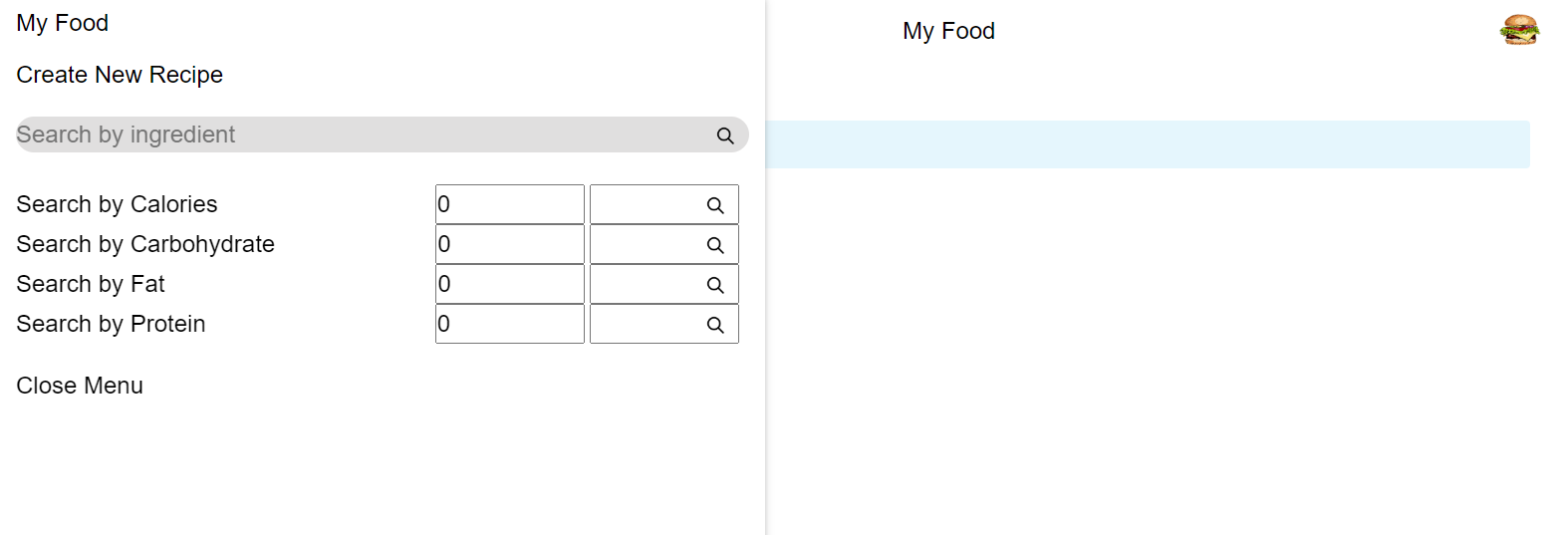
На горниот дел од екранот имаме Header на кој во левата страна се наоѓа хамбургер мени кое го отвара главното мени.

Текстот во средниот дел на header-от (My Food) е линк кој не пренасочува до почетната страна. (Слика 1)



Слика 1

При клик на хамбургер менито ни се отвара главното мени во кое има опции за навигирање до главните функционалности на апликацијата кои се наведени погоре.Слика 2



Слика 2

## Додавање на рецепт

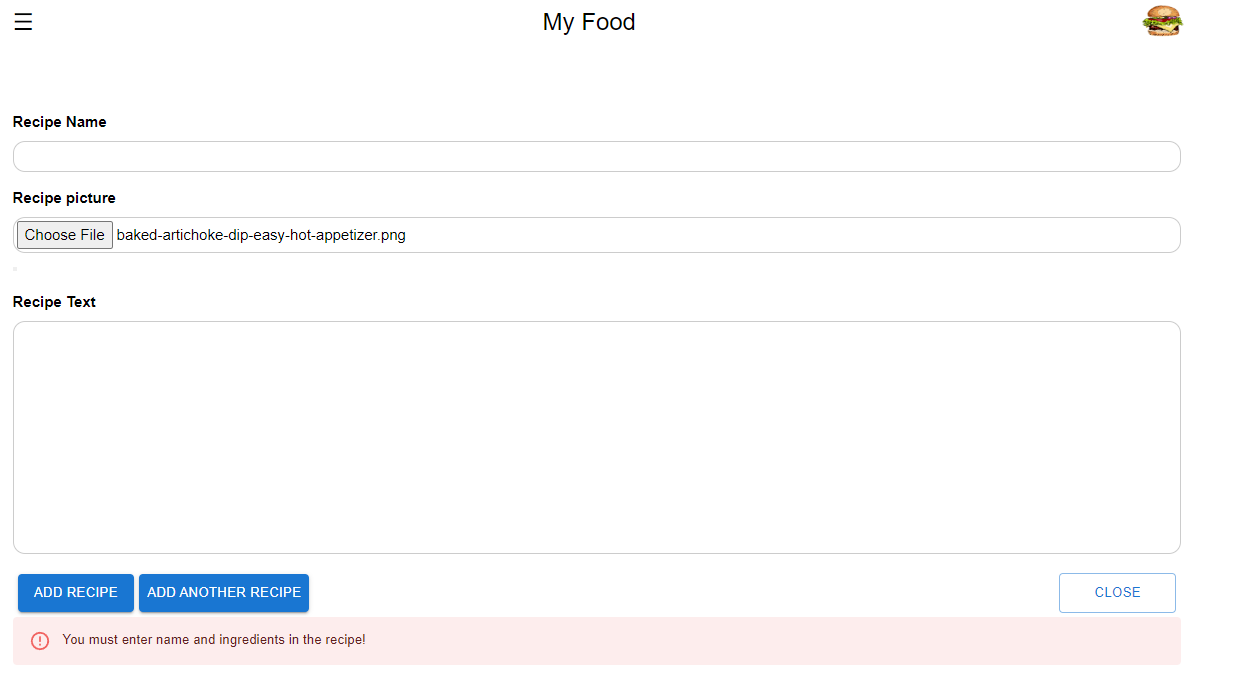
Најпрвин ќе ја разгледаме опцијата за додавање на рецепти. При клик на копчето за додавање на нов рецепт ни се појавува форма во средината на екранот која треба да ја пополниме и содржи:

* поле за име,
* поле за текст
* поле за слика на рецептот.

Под формата има 3 копчиња:

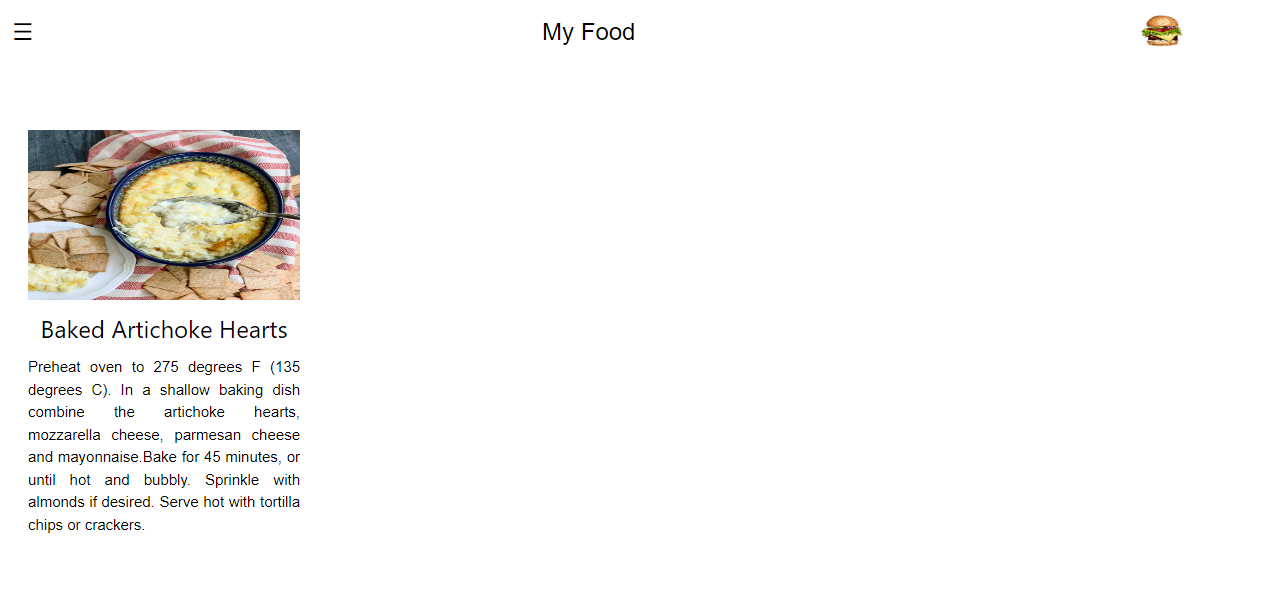
* првото копче (ADD RECIPE) ја испраќа формата за зачувување на рецепт рецептот и ја затвора формата
* второто копче (ADD ANOTHER RECIPE) има функција да ја испрати формата за креирање на нов рецепт и формата да ја подготви за креирање на друг рецепт без да ја затвори
* третото копче (Close) кое се наоѓа на најдесната страна има функција да ја затвори формата без да ја испрати.

Рецептот валидно ќе се креира само доколку сите полиња се пополнети и доколку текстот на рецептот содржи барем една состојка, во спротивно ќе добиеме порака за неправилно креиран рецепт Слика 3.



Слика 3

Доколку рецептот успешно се креира, истиот се додава во листата на рецепти на средината на екранот како што е покажано во Слика 4.



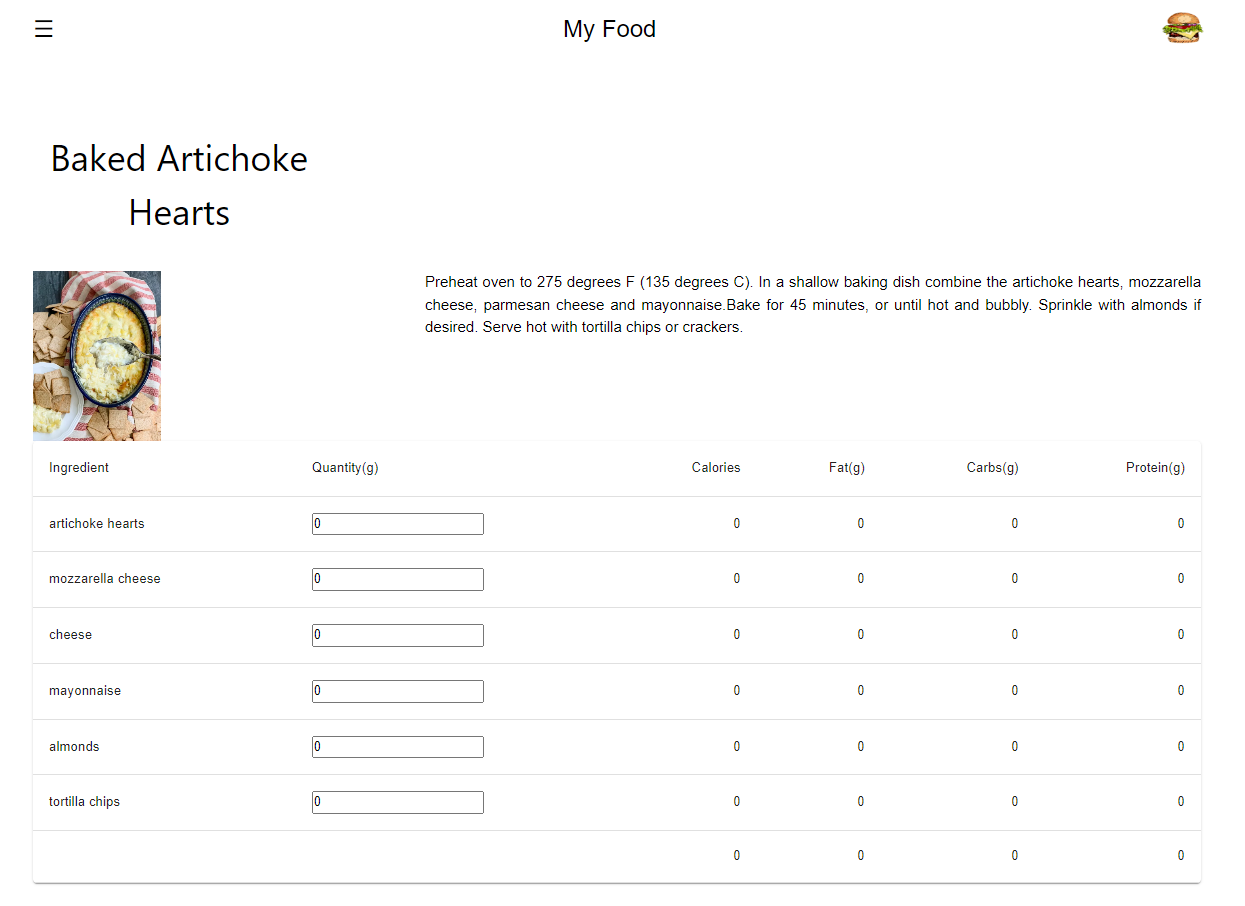
Слика 4

## Прикажување на детали и пресметување на нутритивни вредности на рецепт

При клик на рецептот се прикажува нова страна на која ги има сите информации за одбраниот рецепт. На неа може да се види името на рецептот, сликата и целиот текст. Во продолжение во табеларен приказ се излистани состојките. За секоја состојка во табелата е прикажано нејзиното име, количината на состојката која се користи во рецептот и нутритивните вредности кои ги содржи.

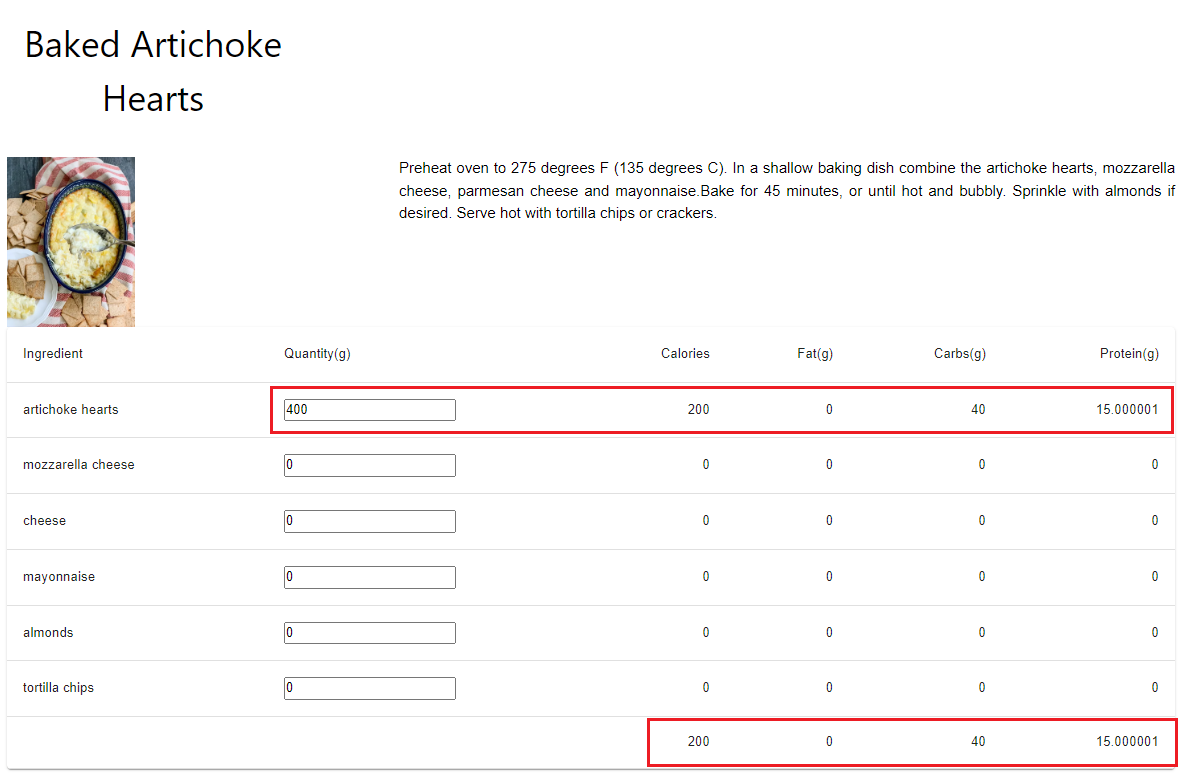
При ново креиран рецепт, нема поставеност на количините на состојките и поради тоа вредноста на нутритивните вредности е 0. Задача на креаторот на рецептот е да ги напише соодветните количини на состојките на рецептот.

Ова е прикажано на следната Слика 5.

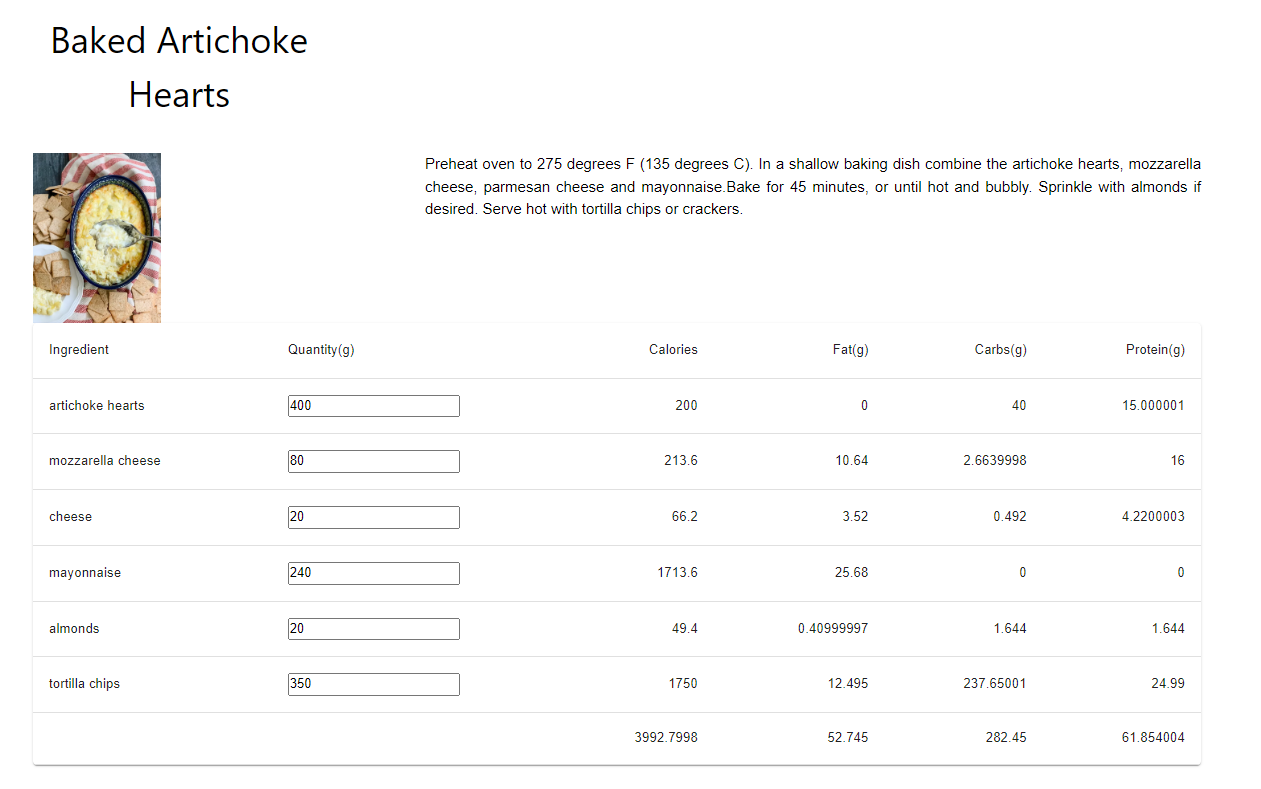


Слика 5

Последниот ред од табелата ги содржи сумирани вредностите на секоја од нутритивните вредности. При менување на количината барем на една состојка, вредноста на нутритивните вредности на рецептот соодветно се менуваат (Слика 6 и Слика 7).



Слика 6

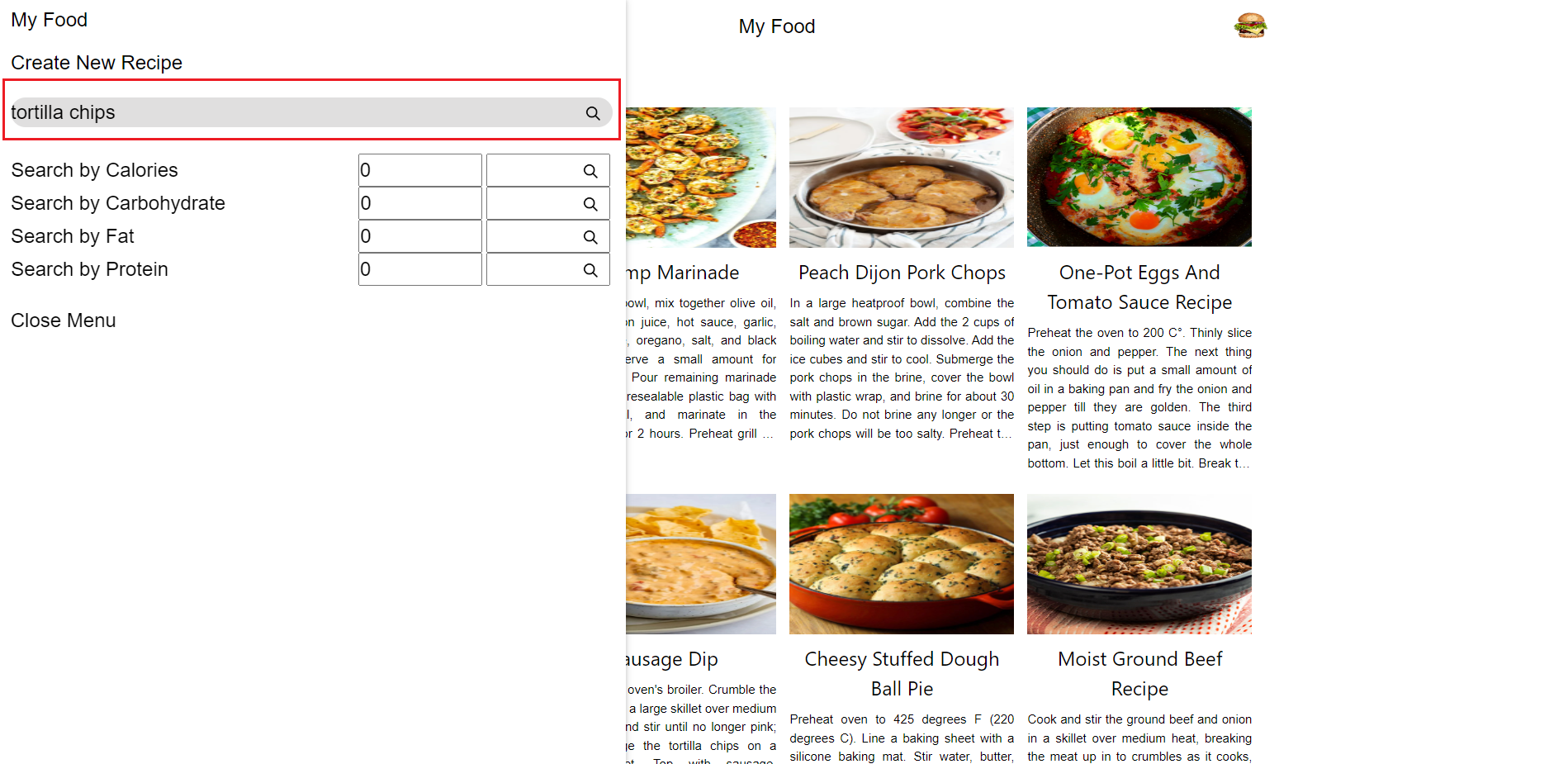


Слика 7

## Пребарување на рецепт преку дадена состојка

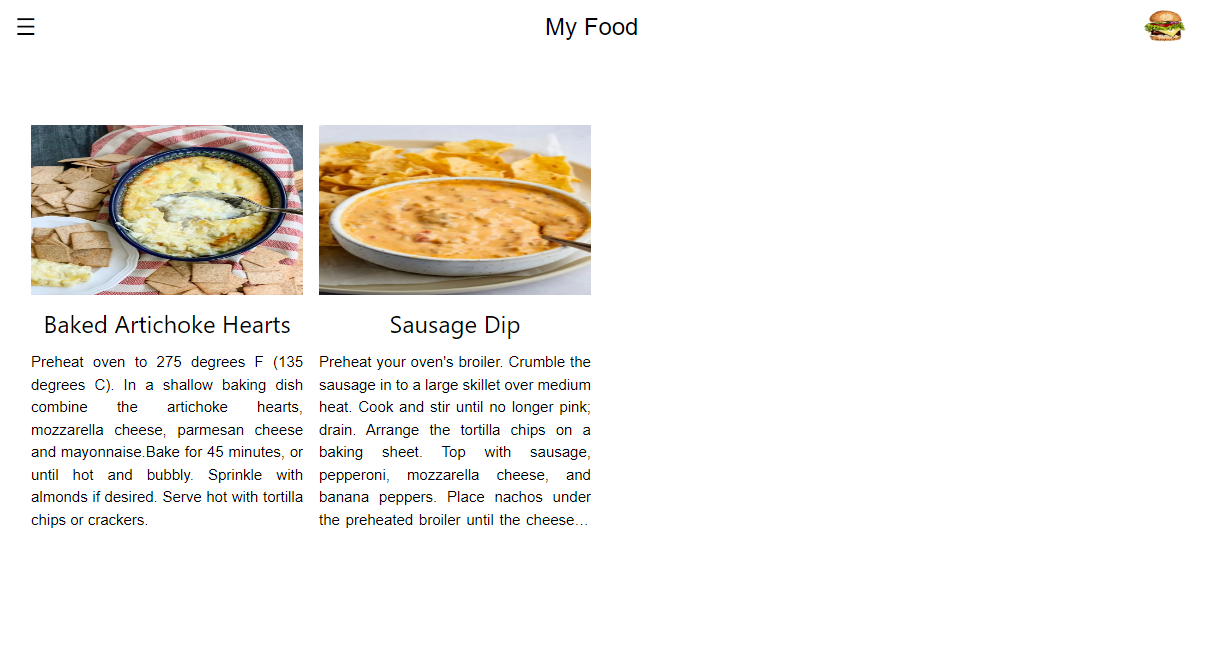
Во хамбургер менито апликацијата нуди опција за пребарување на рецепт според клучни зборови.

Оваа опција на корисникот му нуди можност за пребарување на рецепти преку дадена состојка за која корисникот е заинтересиран да ја содржат рецептите. На пример доколку не интересира кои се сите рецепти кои ја содржат состојката “tortilla chips” пребарувањето може да се направи со впишување на “tortilla chips” во полето за пребарување преку состојки означено на следната Слика 8



Слика 8

Резултатот од пребарувањето според дадената состојка ќе биде излистано на екран, како што е прикажано на Слика 9.

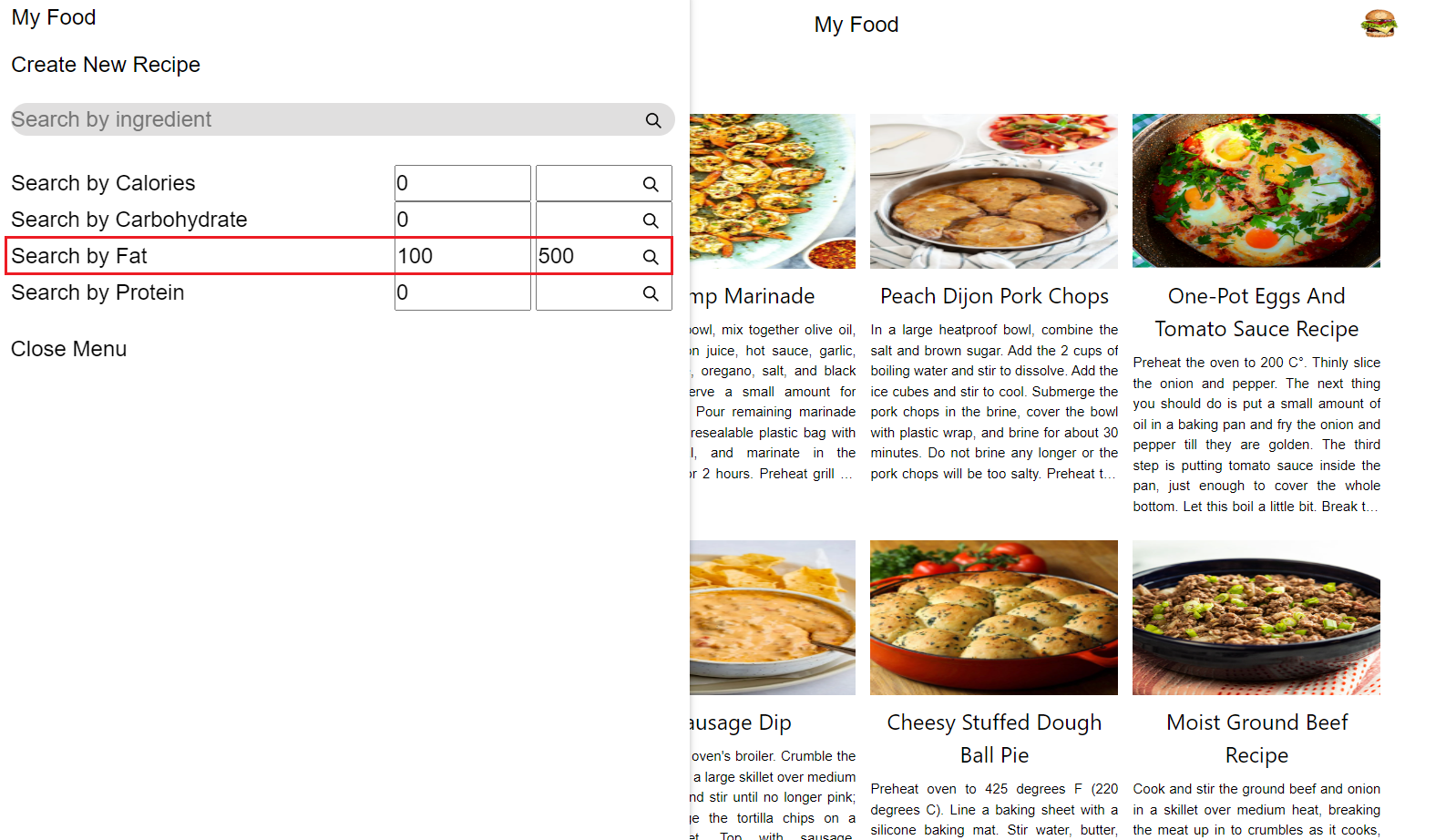


Слика 9

## Пребарување на рецепт преку ранг на нутритивни вредности

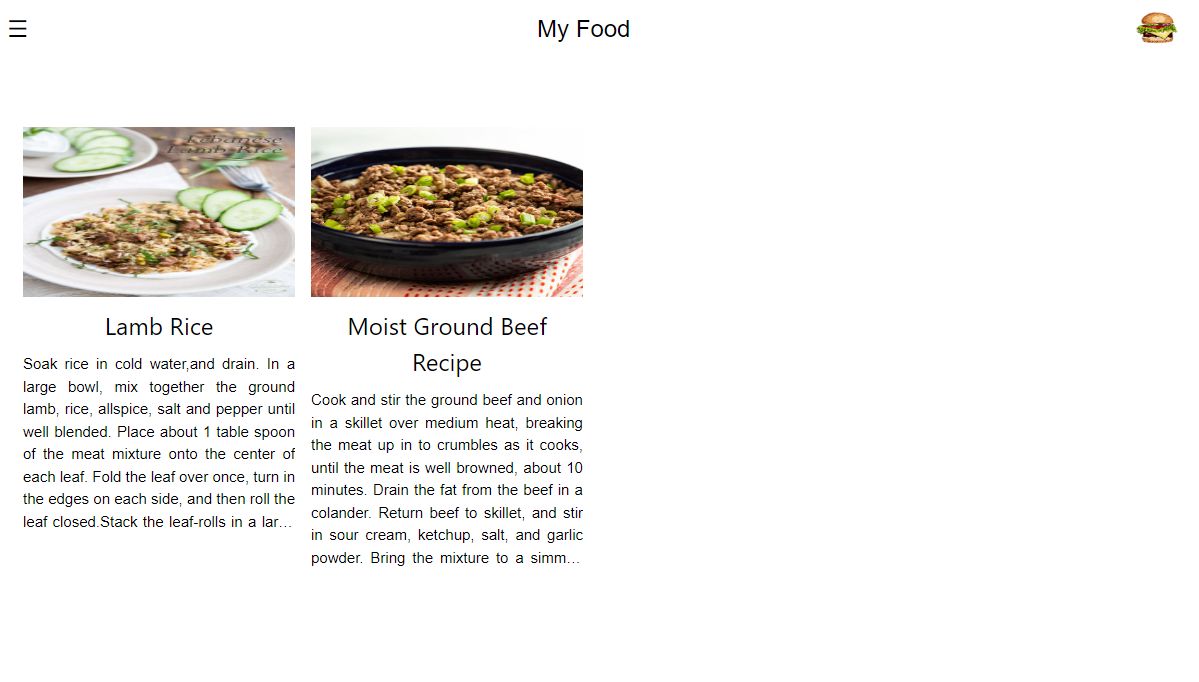
Во хамбургер менито апликацијата нуди опција за пребарување на рецепт според ранг на нутритивни вредности.

Оваа опција на корисникот му нуди можност за пребарување на рецепти кои имаат определена нутритивна вредност која е во определен ранг. На пример доколку не интересира кои се рецептите кои содржат маснотии во рангот од 100-500 единици, пребарувањето може да го направиме на начин како што е прикажан на следната слика. (Слика 10)



Слика 10

Резултатот од пребарувањето според дадената состојка ќе биде излистано на екран, како што е прикажано на Слика 11.

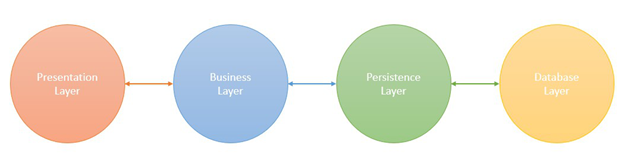


Слика 11

# **Архитектура на решението**

Архитектура на апликација [9] ги опишува техниките и шаблоните кои се користат за дизајн и изработка на апликација. Архитектурата дава правци за следење за решавање на проблеми кои се повторуваат.

Архитектурата на апликацијата го следи принципот на 4 слоеви: presentation, business, persistence и database. Слика 12



Слика 12

## Presentation

Презентацискиот слој е највисокиот слој и го претставува корисничкиот интерфејс и е комуникацискиот слој на апликацијата помеѓу корисникот и апликацијата. Работата на овој слој е едноставно прикажување на податоци на корисникот и превземање податоци од корисникот. Репрезентација на овој слој е front-end делот на апликацијата. Изработена е во Javascript библиотеката React за функционалноста и HTML и CSS за изгледот.

Презентацискиот слој има за функција да направи добро корисничко искуство при користење на апликацијата. Во оваа дипломска работа корисничкото искуство се фокусира за да му даде на корисникот едноставно пребарување и додавање на рецепти. Во преден план е додавање на рецепти и нивно пребарување по состојки кои ги содржат со што му понудува на корисникот сервис за препорака на храна која ги сочинува тие намирници.

Корисничкото искуство на апликацијата ги поддржува корисничките сценарија кои се опишани во оваа дипломска работа. Презентацискиот слој има задача да го презентира и имплементира корисничкото искуство.

## Business

Бизнис слојот е логичкиот слој на апликацијата. Бизнис слојот ја имплементира логиката зошто постои дадена апликација. Тој добива податоци од презентацискиот слој и ги трансформира користејќи ја логиката на апликацијата со цел да се постигне дадена функционалност. За да се имплементираат функционалностите на апликацијата бизнис слојот често има потреба да запише или прочита податоци. За таа цел бизнис слојот го користи persistence слојот.

Бизнис слојот ги контролира апликациските функционалности на апликацијата. Тој покрива:

* Изведување на сите потребни калкулации и валидации
* Менаџирање на текот на апликацијата
* Менаџирање на пристапот до податоците од презентацискиот слој

Физички во апликацијата овој слој е претставен преку сервис класите.

Во веб апликацијата бизнис слојот ги имплементира деталите за како се спроведуваат основните функционалности во неа. Ако ги погледнеме корисничките сценарија тие опишуваат што се случува во апликацијата. Овој слој всушност е местото каде што е имплементирана целата логика која сакаме да ја постигнеме во нашата апликација.

Да ги погледнеме повторно корисничките сценарија:

* Додавање на рецепт – бизнис слојот се осигурва дека ќе бидат детектирани состојките во рецептот и дека за да се зачува истиот ќе бидат запазени сите правила. За да се запише рецепт тој мора да има барем една состојка. Тоа е бизнис правило на оваа апликација и овој слој се грижи тоа да биде форсирано. При додавање на рецептот бизнис слојот се грижи да обезбеди за секоја од состојките информации за вредноста на нутритивната вредност. Со ова бизнис слојот ги обезбедува сите потребни детали за рецептот кои се потребни за обавување на целта на оваа апликација. Доколку некоја од информации неможе да се обезбеди во овој слој се спречува понатамошно создавање на рецепт со што се обезбедува правилно работење на апликацијата и обезбедување на интегритет на податоците.
* Пребарување на рецепт – пребарувањето во апликацијата е овозможено преку два критериума. При пребарување на рецепти бизнис слојот се грижи да го пребара точното место според бараниот критериум во базата на податоци каде што е запишан рецептот и да ги врати сите објекти кои го исполнуваат. Ова го опфаќа пребарувањето и преку состојки и преку ранг на нутритивни вредности. Задача на бизнис слојот е да знае да се справи со сите сценарија кои би можеле да излезат од пребарувањето и да даде соодветен одговор (пример: доколку пребараме рецепти според состојка која ја немаме запишано во база, како резултат испраќа празен резултат).

## Persistence

Persistence слојот има задача да го апстрахира начинот на кој апликацијата комуницира со базата или кој и да е медиум за зачувување на податоци. За имплементација на persistence слојот во оваа дипломска работа е искористен Spring data JPA.

Spring data JPA [10] е framework кој нуди имплементирање на слојот за работа со податоци. Овој framework ја олеснува работата и го скратува потребниот код кој треба да се напише да може да се имплементира интеграција со база на податоци. Убавината на Spring data JPA e дека девелоперот треба да го напише интерфејсот за репозиторито во кој декларативно ги опишува прашалниците за интеракција со базата. Spring data JPA потоа автоматски ги имплементира тие методи.

Во апликацијата имаме интеракција со два ентитети:

* Рецепт
* Состојка

Користејќи Spring data JPA се имплементирани потребните интеракции на ентитетите со соодветните табели во базата. Овие репозиторија ги содржат основните CRUD и останатите потребни методи за интеракција со податоците потребни на бизнис слојот.

## Database

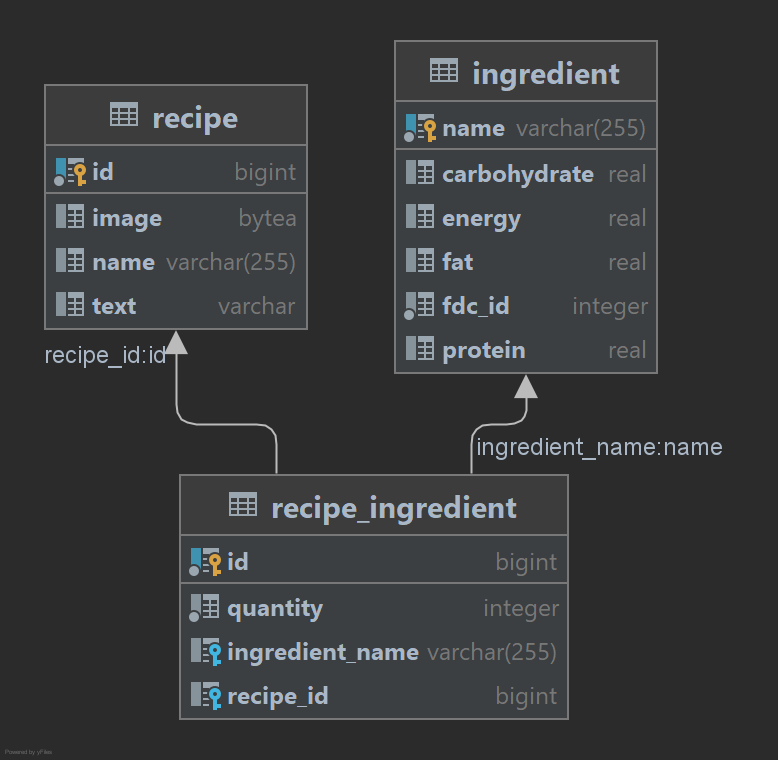
Слојот на база на податоци ја содржи вистинската имплементација на базата на податоци во која се зачувуаат податоците. Една апликација може да работи со повеќе бази на податоци. Репозитори слојот е одговорен да направи интеграција со соодветната база на податоци така што ќе и понуди на апликацијата можност за зачувување на податоците на точното место. Иако во оваа апликација тоа не е случај, не е невообичаено една апликација да работи со повеќе бази на податоци. Токму затоа репозитори слојот е воведен во архитектурата на една апликација за да се скрие овој детал од бизнис логиката на апликацијата. Бизнисот не го интересираат деталите од типот каде податоците се наоѓаат туку него само го интересира дека тој податок едноставно може да го добие од некаде.

Во оваа дипломска работа е искористена PostgreSQL. Интеракцијата со базата е овозможена со Spring data и деталите за нејзиното користење се скриени и овозможени од тој framework.

Шемата на базата на податоци е доста едноставна и се сочинува од 3 табели преку кои се моделирани интеракциите на двата ентитети кои постојат во апликацијата:

* Recipe
* Ingredient
* Recipe\_Ingredient cross-reference табела преку која е моделирана MxN релацијата која постои помеѓу овие два ентитети. Еден рецепт има повеќе состојки и една состојка е содржана во повеќе рецепти.

Шемата на базата на податоци е претставена на Слика 13.



Слика 13

# **Користени технологии**

## **Spring boot**

За изработка на back-end делот на апликацијата се користи Spring Boot[1]. Spring Boot e најпознатиот Java framework за креирање на микро сервиси и овозможува лесен начин за креирање на самостојна Spring базирана апликација која е Production-grade и може едноставно да работи. Предност на оваа технологија[2] е зголемување на продуктивноста и намалување на времето за development, бидејќи се намалува пишување на boilerplate код, анотации и XML конфигурации.

Исто така Dependency injection е фундаментален аспект на Spring framework, преку кој Spring container-от инџектира објекти во други објекти или dependencies. Односно ова дозволува loose coupling на компоненти и ја трга одговорноста од менаџирање компоненти во container-от[3].

## **React**

За изработка на front-end делот на апликацијата се користи JavaScript библиотеката React. Оваа библиотека овозможува лесен начин за креирање на интерактивни UIs базирани на компоненти. React ја следи парадигмата на декларативно програирање па за секој state на апликацијата се креираат едноставни погледи, кај кои при промена на податоците се ажурираат и рендерираат соодветните компоненти [4]. Бидејќи логиката на компонентите се пишува во JavaScript, лесно може да се испраќаат големи податоци низ апликацијата.

## **Semantic Web**

Семантички веб или исто така познат како Web 3.0 е екстензија на World Wide Web стандардите поставени од World Wide Web Consorotium (W3C). Целта на семантичкиот веб е интернет податоците да бидат разбирливи за машини.

Да се овозможи работа со семантчки податоци се користат технологии како Resource Description Framework (RDF) и Web Ontology Langugage (OWL). Овие технологии формално ја претставуваат metadata-та. На пример, онтологијата може да опише концепти, врски помеѓу ентитети и категории од работи. Оваа вградена семантика нуди значајни предности како што се расудување во однос на податоците и работење со хетерогени извори на податоци[5].

## **PostgreSQL**

PostgreSQL e open source систем за објектно релациона база на податоци[6]. Како и сите други релациони бази на податоци ги чува податоците во табели и со користње на SQL јазикот може да се испраќаат queries. Разликата на PostgeSQL од другите бази на податоци е тоа што има воведено користење на концепти од објектно ориентирано програмирање како што се user-defined објекти и table inheritence[7].

# Користење на 3rd party технологии

Како што напоменавме во описот на бизнис слојот една од основните функции на апликацијата е детектирање на состојките од текстот на рецептот и пресметување на нутритивните вредности на секоја од нив.

За да се имплементираат овие две функционалности во оваа дипломска работа се искористени два екстерни сервиси кои ги нудат токму овие услуги:

* Foodviz - <http://foodviz.env4health.finki.ukim.mk/>
* USA National Agriculture Library - <https://api.nal.usda.gov>

## Foodviz

Foodviz претставува web сервис кој нуди функционалност за детектирање на состојки од текстуален рецепт.

Развиен е на ФИНКИ и се базира на вештачка интелигенција и семантички веб.

Веб сервисот нуди повеќе rest endpoints за работа со рецепти за храна. За реализирање на потребната функционалност на бизнис слојот во дипломската работа е интегриран еден од endpoint-ите кој за задача има детектирање на збор или зборови кои претставуваат храна, односно состојка, користена во текстот на рецептот. Името на endpoint-от е predict.

Дефиницијата на повикот (request-от) до predict endpoint-от изгледа на следниот начин:

|  |  |  |
| --- | --- | --- |
| Url | Http  method | Payload |
| <http://foodviz.env4health.finki.ukim.mk/predict> | Get | * Text – текстот на рецептот кој го пребаруваме * Model – моделот за предвидување. Во нашиот случај секогаш е   bioBert-standard-model-foodon-e100-0.0005.bin ?? |

Резултатот (response-от) на овој повик е во JSON формат кој сочинува низа од токени (tokens) кои соодветствуваат на секој збор од текстот на рецептот.

Од целиот резултат од интерес се сите токени кои имаат вредност во полето otherTags различно од null. Тоа значи дека тој токен соодветствува на храна. Со користење на овие токени всушност ги добиваме состојките на рецептот кој бил пратен во requestot преку формата.

Пример, доколку го испратиме следниот рецепт на request:

|  |
| --- |
| Preheat your oven's broiler. Crumble the sausage in to a large skillet over medium heat. Cook and stir until no longer pink; drain. Arrange the tortilla chips on a baking sheet. Top with sausage, pepperoni, mozzarella cheese, and banana peppers. Place nachos under the preheated broiler until the cheese is melted, 5 to 8minutes. Serve with pizza sauce for dipping. |

Foodviz ќе ги врати следниве токени кои соодветствуваат на храна:

* sausage
* tortilla chips
* mozzarella cheese
* banana
* peppers
* cheese
* pizza sauce

Секој од овие ентитети е претставен со следнава форма во response-от од web service-от (Слика 14)



Слика 14

Добиените објкети бизнис слојот понатака ги испраќа до следното API со кое ќе ги добиеме деталите за нутритивни вредности на секоја од состојките.

## USA National Agriculture Library

USA National Agriculture Library (UAL) e национална библиотека на САД која претставува светски најголема колекција на информации поврзани со земјоделство и поврзани науки.

UAL нуди web service преку кој може да се пребарува нивната колекција и може да се добијат различни информации од нивната библиотека.

За да се комплетира функционалноста на бизнис слојот во апликацијата е имплементирана интеграција со веб сервисот на оваа библиотека со цел да се добијат нутритивните вредности на состојките кои ги извлековме со повик до Foodviz сервисот на ФИНКИ. Со овој чекор ги добиваме сите потребни информации поврзани со една состојка со што таа можеме да ја запишеме во базата на податоци.

UAL сервисот е сервис кој нуди големо количество на опции и алатки за истражување на податоци поврзани со нивниот домен. За да ја постигнеме функционалноста потребна во апликацијата ние пристапуваме до endpoint-от foods search.

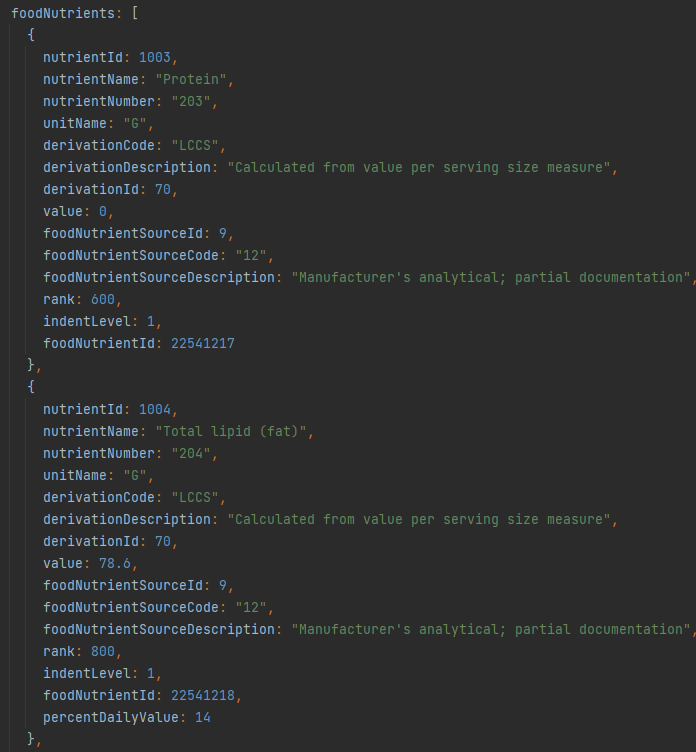
|  |  |  |
| --- | --- | --- |
| Url | Http  method | Payload |
| <https://api.nal.usda.gov/fdc/v1/foods/search> | Get | * query – името на храната која ја пребарувамe * page size – колку резултати да се вратат во одговорот * api\_key – клучот за пристап до сервисот на библиотеката |

Резултатот (response-от) на овој повик е во JSON формат кој сочинува низа од храни (tokens) кои се совпаѓаат со тоа што го пребаруваме. Во нашиот случај секогаш бараме конкретна храна и низата секогаш се сочинува повеќе производители на истата храна.

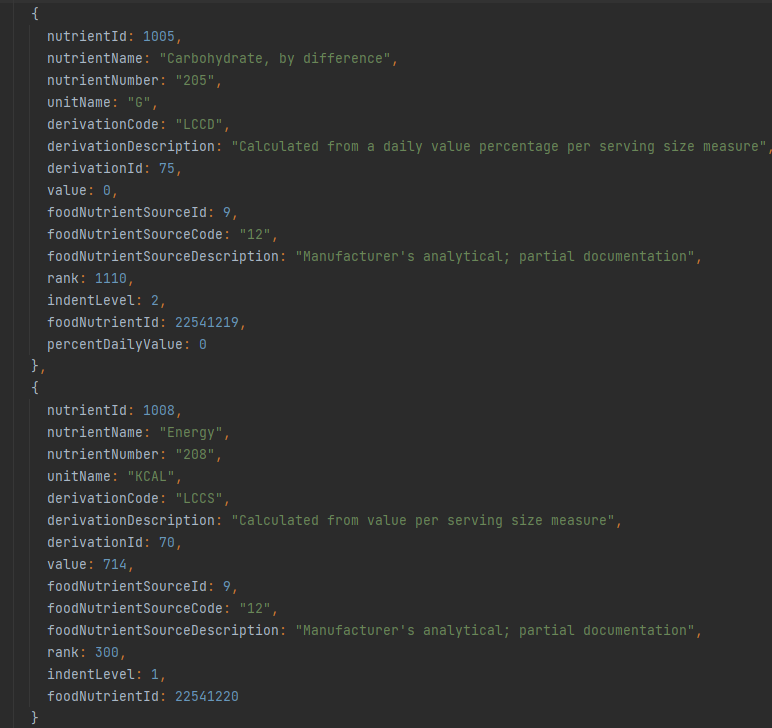
На бизнис слојот од интерес му е нутритивната вредност на храната која ја пребаруваме, така што за нашата апликација не е од круциална вредност конкретниот производител на храната, па затоа секогаш го избира првиот резултат од низата.

Од резултатот кој што е добиен од интерес ни се макронутриентите (маснотии, јагленихидрати, протеини) и калориите.

Пример, доколку на endpoint-от му испратиме барање за информации поврзани со нутритивните вредности на мајонез, тогаш ќе го добиеме следниот резултат во полето за нутритивни вредности како одговор Слика 15 и Слика 16:



Слика 15



Слика 16

Со овие информации добиени од endpoint-от на UAL ги имаме сите информации потребни за успешно зачувување на рецептот во базата на податоци. По запшувањето на рецептот во базата тој е достапен за читање, пребарување и манипулирање.

# Заклучок

Во оваа дипломска работа е претставена корисна веб апликација која користи повеќе сервиси базирани на вештачка интелигенција и семантички веб. Нивното комбинирање нуди сервис за детекција на состојки од текстуален рецепт и вредноста на нутритивните вредности на секоја од нив.

Гледајќи во иднината на оваа апликација би можело да се додаде автоматска идентификација на квантитет на состојки од текстуален рецепт со што би се намалила мануелната работа при креирање на рецепт. Исто така би можело да се додаде филтрирање на рецепти според одредена диета која ја држиме или според некои алергени.

# **Користена литература**

1. <https://spring.io/>
2. <https://www.adservio.fr/post/advantages-of-spring-boot>
3. <https://www.baeldung.com/spring-dependency-injection>
4. <https://reactjs.org/>
5. <https://en.wikipedia.org/wiki/Semantic_Web>
6. <https://www.postgresql.org/>
7. <https://www.tutorialandexample.com/postgresql-tutorial>
8. <https://spring.io/projects/spring-data-jpa>
9. <https://www.geeksforgeeks.org/spring-boot-architecture/>
10. <https://spring.io/projects/spring-data-jpa>