

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

УДК 004.9:338.488.2:373.3/5

«ПОГОДЖЕНО» «ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ»
Декан факультету Завідувач кафедри
Інформаційних технологій Комп'ютерних наук

Глазунова О.Г., д.пед.н., професор

Голуб Б.Л., к.т.н., доцент

« » 20 р « » 20 р

МАГІСТЕРСЬКА РОБОТА

на тему: «Back-end частина системи підтримки прийняття рішень в управлінні
шкільним харчуванням»
Спеціальність: 121 – «Інженерія програмного забезпечення»

Освітня програма: «Програмне забезпечення інформаційних систем»

Орієнтація освітньої програми: освітньо-професійна
Гарант освітньої програми

К.Т.Н., доцент

(науковий ступінь та вчене звання)

_____ (підпис)

Голуб Б.Л.,

(ПБ)

Керівник магістерської роботи
К.Т.Н., доцент
(науковий ступінь та вчене звання)

_____ (підпис)

Голуб Б.Л.,

(ПБ)

Виконав

_____ (підпис)

Крупко Б.О.

(ПБ студента)

КИЇВ - 2021
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
НУБІП України

ЗАТВЕРДЖУЮ
Завідувач кафедри
комп'ютерних наук

НУБІП України

/ Голуб Б.Л., доцент, к.т.н /

підпис

20 р.

ЗАВДАННЯ

ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ РОБОТИ СТУДЕНТУ КРУПКУ БОГДАНУ ОЛЕКСАНДРОВИЧУ

НУБІП України

Спеціальність 121 – «Інженерія програмного забезпечення»

Освітня програма «Програмне забезпечення інформаційних систем»

Орієнтація освітньої програми освітньо-професійна

НУБІП України

1. Тема магістерської роботи: Back-end частина системи підтримки прийняття рішень з управління шкільним харчуванням затверджена наказом ректора НУБІП України від 29.10.2020 № 1636 “С”

2. Термін подання завершеної роботи на кафедру _____ 2021 . 11 . 30
рік, місяць, число

3. Вихідні дані до магістерської роботи: замовлення страв

НУБІП України

4. Перелік питань, що підлягають дослідженню:

1. Провести системний аналіз процесу формування шкільного меню;
2. Сформулювати технічне завдання
3. Провести аналіз існуючих рішень
4. Побудувати архітектуру системи
5. Розробити СППР з управління шкільним харчуванням

НУБІП України

Дата видачі завдання

2020 . 10 . 29
рік, місяць, число

Керівник магістерської роботи _____

підпис

/ Голуб Б.Л. /

ініціали та прізвище

НУБІП України

Завдання прийняв до виконання _____

підпис

/ Крупко Б.О. /

ініціали та прізвище

НУБІП України

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ 4

ВСТУП 5

1 СИСТЕМНИЙ АНАЛІЗ ПРОЦЕСУ ФОРМУВАННЯ ШКІЛЬНОГО МЕНЮ 7

1.1 Аналіз процесу формування шкільного меню 7

1.2 Технічне завдання 8

1.3 Моделювання процесу формування шкільного меню 13

2 ТЕОРЕТИЧНІ ПЕРЕДУМОВИ 23

2.1 Визначення ЄППР 23

2.2 Алгоритми та технології аналізу даних 25

2.3 Нормативно-правові документи 36

2.4 Норми шкільного харчування 37

3 РОЗРОБКА СИСТЕМИ 41

3.1 Архітектура системи 41

3.3 Інструменти та технології 45

3.4 Налаштування взаємодії розподіленої системи 48

3.5 Безпека даних 49

3.6 Алгоритмізація програмних модулів 51

3.7 Дані для формування звітності 57

3.8 Розгортання системи 57

4 РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ 61

4.1 Опис розробленого програмного забезпечення 61

4.2 Апаратні та програмні вимоги до системи 68

ВИСНОВКИ 69

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 71

НУБІП України

НУБІП України

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД – База даних

СУБД – Система управління базами даних

НУБІП України

ORM – Об'єктно-реляційна модель

ПЗ – Програмне забезпечення

СППР – Система підтримки прийняття рішень

ООП – Об'єктно-орієнтоване програмування

НУБІП України

SQL – Structured Query Language

UML – Unified Modeling Language

JSON – JavaScript Object Notation

HTTP – HyperText Transfer Protocol

НУБІП України

REST – Representational State Transfer

JVM – Java Virtual Machine

НУБІП України

НУБІП України

НУБІП України

НУБІП України

ВСТУП

Актуальність теми

Шкільне харчування, це важливий аспект освітнього процесу та життєдіяльності школярів. При правильно зіставленому меню та раціоні покращиться самопочуття учнів, що приведе до більшої продуктивності під час навчання, привчить до здорової їжі і у повсякденному житті, а також зменшить захворюваність.

У 2020 році за ініціативою першої леді України Олени Зеленської стартувала реформа системи шкільного харчування. Тому ця тема актуальна на рівні держави.

«Збереження здоров'я дітей є одним з найважливіших завдань держави, тому реформа шкільного харчування надзвичайно актуальна та важлива. Харчування наших дітей має і буде здоровим та збалансованим. Перекоаний, що нам це вдасться», – зазначив міністр охорони здоров'я Віктор Ляшко.

Система підтримки прийняття рішень з управління шкільним харчуванням дозволить вирішити ряд проблем:

- Формування шкільного меню відповідно до вподобань учнів
- Отримання актуальної інформації про задоволеність шкільним харчуванням
- Підвищить довіру батьків та учнів до шкільного харчування та освіти в цілому
- Надасть інформацію для контролю якості
- Отримання зворотнього зв'язку від батьків та учнів.

Об'єкт дослідження

Процес формування раціону харчування та шкільного меню в закладах освіти.

Предмет дослідження

Система підтримки прийняття рішень з управління шкільним харчуванням.

Мета дослідження

Покращити якість шкільного харчування в закладах освіти, підвищити ефективність освітнього процесу, а також надати інструменти для керування та аналізу якості шкільного харчування.

Завдання

- Провести системний аналіз процесу формування шкільного меню;
- Сформулювати технічне завдання
- Провести аналіз існуючих рішень
- Побудувати архітектуру системи
- Розробити СППР з управління шкільним харчуванням
- Описати результати дослідження

Методи дослідження

Для формування вподобань користувачів та обробку даних було застосовано технологію BigData.

Наукова новизна

Вперше було розроблено СППР з управління шкільним харчуванням для освітніх закладів України.

Апробація результатів дослідження

1. Крупко Б.О. «Огляд архітектури Back-end частини системи підтримки прийняття рішень з управління шкільним харчуванням» // Збірник наукових праць за матеріалами IV всеукраїнської науково-практичної Інтернет конференції студентів та аспірантів – Київ – 2021 – ст. 34-35 (тези доступні за посиланням:

<http://econference.nubip.edu.ua/index.php/taacsd/2021/paper/view/2410>).

НУВБІП України

1 СИСТЕМНИЙ АНАЛІЗ ПРОЦЕСУ ФОРМУВАННЯ ШКІЛЬНОГО МЕНЮ

1.1 Аналіз процесу формування шкільного меню

Процес формування шкільного меню є складним та проходить багато етапів, що ускладнює і збільшує затрати, ще сильніше.

У процесі формування шкільного меню приймає участь багато спеціалістів з суміжних і абсолютно різних галузей та напрямків:

- Дієтологи:
 - Підготовка норм харчування
 - Формування списку харчових продуктів по групах
 - Розробка принципів та зразків шкільного меню
- Технологи
 - Формування переліку страв та напоїв для різної калоричності харчування
 - Відбір продуктів для асортименту шкільних буфетів
 - Розробка технологічних карт разом з кухарами
- Кухари
 - Підготовка рецептів на відібрані страви та напої
 - Розробка технологічних карт разом з технологами
- Економісти
 - Обрахунок орієнтованої вартості сезонного меню
 - Розробка системи звітності за витрачені кошти

Після виконання цих шагів, сформоване меню повинне бути розглянуте і затверджене відповідним наказом Міністерства охорони здоров'я України та Міністерства освіти і науки. І впроваджене у закладах освіти.

Процес формування меню, який і так доволі важкий, ускладнений взаємодіями між спеціалістами і через мірною бюрократією.

Це все приводить до затримок і початок процесу формування меню розпочинається задовго до його реалізації, що приводить до додаткових затрат. Окрім цього, на процес впливає людський фактор, і при виникненні помилки потрібно починати, частину або весь процес, спочатку.

Також не варто забувати, що розгляд і затвердження меню відповідними Міністерствами може затягнутись на багато днів.

1.2 Технічне завдання

1.2.1 Опис користувачів системи. Основними користувачами системи є:

- особи, що мають відношення до освітнього процесу:

- учні;

- батьки учнів;

- працівники освітньої галузі:

- адміністратори системи;

- працівники МОН;

- працівники регіональних контролюючих установ;

- працівники освітніх закладів;

- технологи;

- кухарі;

- дієтологи;

- економісти.

1.2.2 Характеристики користувачів системи. Усі користувачі можуть реєструватися та авторизуватися в системі, керувати інформацією, введеною в особистих кабінетах, а також отримати доступ до повного набору функцій відповідно до ролі користувача в системі.

Ті, хто бере участь у освітньому процесі, використовуватимуть систему для надання зворотного зв'язку та оцінки якості шкільного харчування, а також для управління відповідними даними. Системний адміністратор матиме доступ до всіх функцій і зможе керувати ролями інших користувачів.

Співробітники Міністерства освіти і науки отримають доступ до більшості функцій системи та зможуть переглядати всі дані системи.

Співробітники регіональної контролюючої установи зможуть отримати доступ до більшості функцій системи та переглядати дані, але їхні права доступу обмежені цією зоною.

Працівники закладів освіти матимуть право використовувати функції в межах закладів, за якими вони закріплені.

Техніки, кухарі, дієтологи та економісти матимуть право використовувати функції, покладені на них посадовими обов'язками.

1.2.3 Опис основного функціоналу системи. Система повинна забезпечити функціонал:

- авторизації та автентифікації;
- надання функціоналу для формування меню:
 - формування заявки на нове меню;
 - оперування нормами шкільного харчування;
 - перегляд статусу заявки;
 - формування меню згідно норм;
 - надання альтернатив для став у меню;
 - додавання нових страв та інгредієнтів у систему;
 - адаптація меню до регіональних смаків учнів.
- надання функціоналу для зворотнього зв'язку:
 - форма зворотнього зв'язку для учнів та батьків;

форми опитування по якості харчування.
 • надання функціоналу для контролю якості:
 ○ формування звітів в розрізі учбових закладів, районів, областей, країни за весь період або проміжок часу.

1.2.4 Функціональні вимоги. Система має відповідати наступним функціональним вимогам.

- Ідентифікація та керування користувачами – ключовим компонентом системи є особистий кабінет користувача, за допомогою якого може бути організований доступ до бази даних та функцій інформаційної системи. Залежно від групи, до якої належить користувач, максимальний набір функцій буде відрізнятись. Система має передбачити модульність особових кабінетів, що дозволить користувачам, які реєструються одночасно, наприклад, як батьки та працівники навчальних закладів, перемикатися між функціями, що відповідають кожній групі. Система повинна забезпечувати реєстрацію та авторизацію користувача.

- Адміністрування користувачів у системі – система має передбачати адміністрування користувачів: зміну інформації, видалення, надання користувачьких ролей кожна з яких повинна керуватися відповідним користувачем ролі “Адміністратор”.

- Подання заяви на формування меню – система має передбачити можливість формування заяви на нове меню, як документ на базі якого будуть виконуватись дії по подальшому формуванню і реалізації шкільного меню.

- Зміна норм та правил побудови шкільного меню – оскільки норми харчування можуть змінюватись, система має передбачити можливість редагування та додавання норм харчування на основі якого буде будуватись меню.

НУБІП УКРАЇНИ

- Зміна, додання та видалення страв – система має передбачити можливість додання нових страв, видалення і редагування існуючих.
- Формування меню – система має передбачити можливість

автоматичного формування меню і надання альтернатив для кожної позиції у ньому, щоб користувач міг вибрати, та сформувати фінальну версію.

НУБІП УКРАЇНИ

- Затвердження меню – після формування та перегляду меню, має бути можливість його затвердження, після якого меню буде розповсюджено між навчальними закладами і вступлять в силу нові правила.

НУБІП УКРАЇНИ

- Перегляд технологічних карт – система має передбачити можливість перегляду кухарем технологічних карт страв у меню, на основі яких він зможе приготувати цю страву.

- Звітність – система має передбачити можливість формування звітів у розрізі країни, області, району, міста, усього закладу за весь час або певний його проміжок, задля забезпечення контролю якості за щільним харчуванням.

НУБІП УКРАЇНИ

- Зворотній зв'язок – система має передбачити форму зворотнього зв'язку і форми опитування по задоволеності меню та якістю шкільного харчування для батьків та учнів.

НУБІП УКРАЇНИ

- Замовлення – система має передбачити внесення даних про кількість приготовлених та реалізованих порцій страв.
- Адаптація під потреби учнів – система має передбачити збір інформації

з форм зворотнього зв'язку і замовлень та використовувати отримані дані під час формування наступного меню, задля максимальної адаптації під потреби учнів.

НУБІП УКРАЇНИ

1.2.5 Нефункціональні вимоги. Система має відповідати наступним нефункціональним вимогам:

НУБІП УКРАЇНИ

НУБІП України

- Надійність Системи повинна бути забезпечена за такими напрямками:
 - забезпечення працездатності компонентів програмно-технічної платформи,
 - збереження даних.

- Максимальний час відновлення працездатності ПЗ не більш ніж одна

НУБІП України

- Інформаційне забезпечення повинно відповідати таким вимогам та

можливостям:

- забезпечення фізичної та логічної цілісності даних;

НУБІП України

- мінімізація надмірності даних, що зберігаються;
- стандартизація представлення даних;
- достовірність та актуальність даних.

- ПЗ повинно мати властивості інтегрованого інформаційного

середовища:

НУБІП України

- забезпечувати зберігання даних про історію змін даних користувачами для забезпечення відповідальності за внесення змін до даних;

- забезпечувати розподіл і надання прав доступу заснованих на рольовому або іншому подібному принципі;

НУБІП України

- забезпечувати автоматичну консолідацію та інформаційну цілісність у межах географічно розподілених даних;

- передбачати за допомогою документованого API можливість інтеграції з іншими інформаційними системами.

НУБІП України

- Кожен елемент Системи має бути розроблений з урахуванням

можливості масштабування.

НУБІП України

1.3 Моделювання процесу формування шкільного меню

1.3.1 Діаграма прецедентів. Діаграма варіантів використання UML, Use Case Diagram – одне з найпростіших представлень системи. Її базові «будівельні елементи» - актори і варіанти використання. Ціль діаграми, дати найбільш загальне уявлення про функціональність системи (її компоненти), не вдаючись до деталей взаємозв'язків функцій.

На рисунку 1, зображена діаграма прецедентів на якій представленні наступні актори:

- дієтолог;
- технолог;
- кухар;
- економіст;
- Міністерство охорони здоров'я України;
- Міністерство освіти і науки України.

До дієтолога відноситься варіант використання «Підготовка основних норм харчування та принципів формування шкільного меню», що є узагальненням трьох варіантів використання: «Підготовка норм харчування», «Формування списку харчових продуктів по групах», «Розробка принципів та вразків шкільного меню». Другий варіант використання – це «Розповсюдження підготовлених норм та принципів формування шкільного меню».

До технолога відносяться два варіанти використання: «Формування переліку страв та відбір продуктів», що є узагальненням для «Формування переліку страв та напоїв для різної кратності харчування» і «Вибір продуктів для асортименту шкільних буфетів», «Розробка технологічних карт».

До кухара також відносяться два варіанти використання, це «Розробка технологічних карт» і «Підготовка рецептів на відібрані страви та напої».

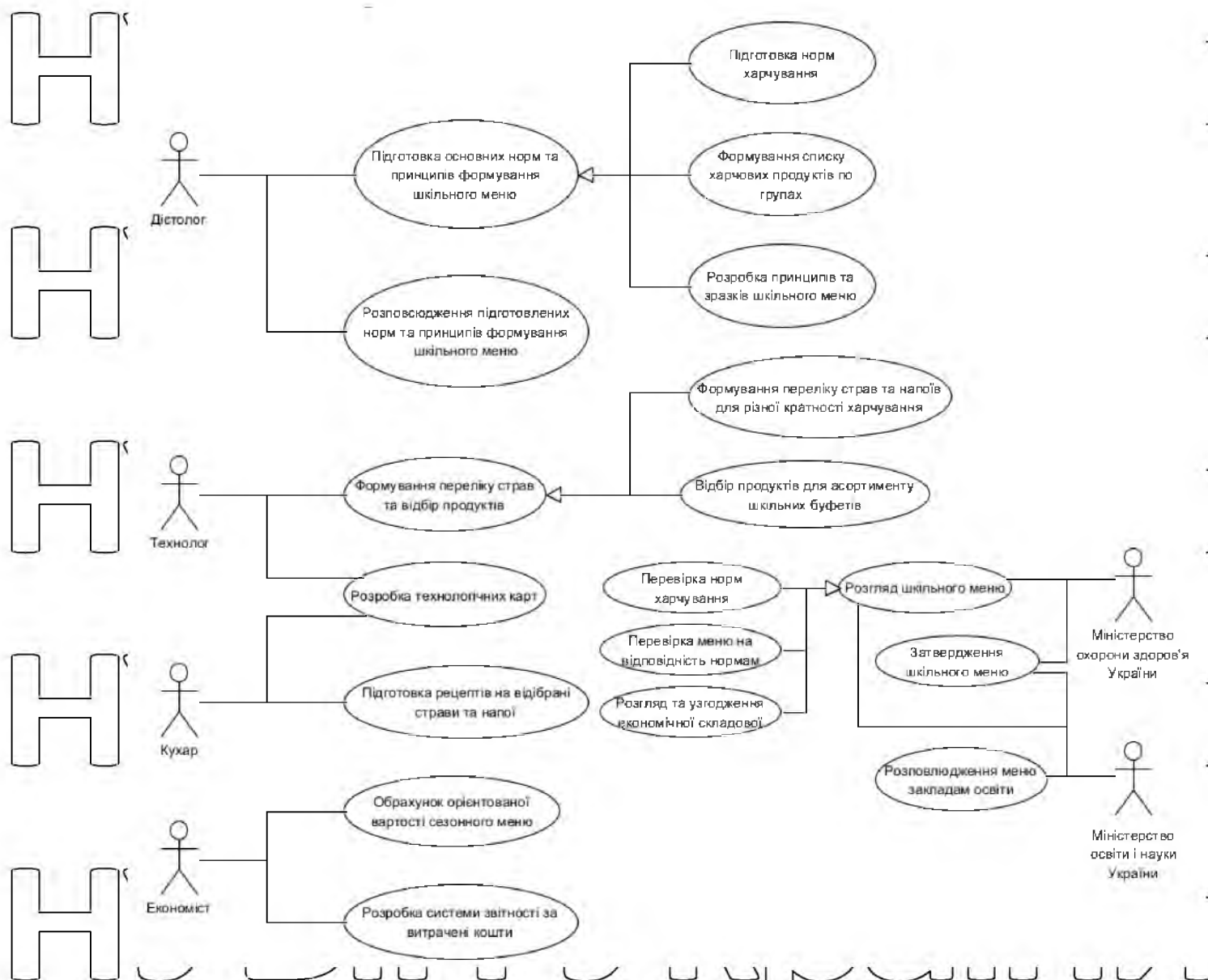


Рис. 1 Діаграма прецедентів

До економіста відноситься два варіанти використання: «Обрахунок орієнтованої вартості сезонного меню» і «Розробка системи звітності за витрачені кошти».

До Міністерства охорони здоров'я України відносяться «Затвердження шкільного меню» і «Розгляд шкільного меню», яке узагальнює «Перевірка норм харчування», «Перевірка меню на відповідність нормам» і «Розгляд та узгодження економічної складової».

До Міністерство освіти і науки України відносяться всі варіанти використання, що і до Міністерства охорони здоров'я України, а також «Розповсюдження меню закладам освіти».

Оптимістичний сценарій

1. Міністерство освіти і науки формує наказ на нове шкільне меню.
2. Наказ отримує дієтолог.
3. Дієтолог підготовлює норми харчування.
4. Дієтолог формує список харчових продуктів по групах.
5. Дієтолог розробляє принципи та зразки шкільного меню.
6. Дієтолог розповсюджує підготовлені норми та принципи формування шкільного меню Міністерствам.
7. Міністерство охорони здоров'я та Міністерство освіти і науки перевіряють норми харчування і узгоджують їх.
8. Технолог формує перелік страв та напоїв для різної кратності харчування.
9. Технолог відбирає продукти для асортименту шкільних буфетів.
10. Технолог разом із кухарем розробляють технологічні карти.
11. Кухар підготовлює рецепти на відібрані страви та напої.
12. МОЗ та МОН перевіряють меню на відповідність нормам і узгоджують його.
13. Економіст обраховує орієнтовану вартість сезонного меню.
14. Економіст розробляє систему звітності за витрачені кошти.
15. МОЗ та МОН розглядають економічну складову і погоджують її.
16. МОЗ та МОН затверджують шкільне меню.
17. Міністерство освіти і науки України розповсюджують нове меню закладам освіти.

Прагматичний сценарій

Умова 1. Норми харчування складені не правильно або не відповідають дійсності.

1. Міністерство освіти і науки формує наказ на нове шкільне меню.

2. Наказ отримує дієтолог.

3. Дієтолог підготовлює норми харчування.

4. Дієтолог формує список харчових продуктів по групах.

5. Дієтолог розробляє принципи та зразки шкільного меню.

6. Дієтолог розповсюджує підготовлені норми та принципи формування шкільного меню Міністерствам.

7. Міністерство охорони здоров'я та Міністерство освіти і науки перевіряють

норми харчування.

8. МОЗ і МОН відхиляють сформовані дієтологом норми харчування.

9. МОЗ і МОН формує лист з доробками і відправляє дієтологу.

10. Дієтолог вносить поправки до сформованих раніше норм.

11. Дієтолог відправляє норми харчування з поправками у Міністерства.

12. МОЗ і МОН повторно розглядають та перевіряють норми харчування і погоджують їх.

13. Технолог формує перелік страв та напоїв для різної кратності харчування.

14. Технолог відбирає продукти для асортименту шкільних буфетів.

15. Технолог разом із кухарем розробляють технологічні карти.

16. Кухар підготовлює рецепти на відібрані страви та напої.

17. МОЗ та МОН перевіряють меню на відповідність нормам і узгоджують

його.

18. Економіст обраховує орієнтовану вартість сезонного меню.

19. Економіст розробляє систему звітності за витрачені кошти.

20. МОЗ та МОН розглядають економічну складову і погоджують її.

21. МОЗ та МОН затверджують шкільне меню.

22. Міністерство освіти і науки України розповсюджують нове меню закладам освіти.

Умова 2. На кожному з етапів були похибки.

1. Міністерство освіти і науки формує наказ на нове шкільне меню.
2. Наказ отримує дієтолог.
3. Дієтолог підготовлює норми харчування.

4. Дієтолог формує список харчових продуктів по групах.
5. Дієтолог розробляє принципи та зразки шкільного меню.
6. Дієтолог розповсюджує підготовлені норми та принципи формування шкільного меню Міністерствам.

7. Міністерство охорони здоров'я та Міністерство освіти і науки перевіряють норми харчування.
8. МОЗ і МОН відхиляють сформовані дієтологом норми харчування.
9. МОЗ і МОН формує лист з доробками і відправляє дієтологу.

10. Дієтолог вносить поправки до сформованих раніше норм.
11. Дієтолог відправляє норми харчування з поправками у Міністерства.
12. МОЗ і МОН повторно розглядають та перевіряють норми харчування і погоджують їх.

13. Технолог формує перелік страв та напоїв для різної кратності харчування.
14. Технолог відбирає продукти для асортименту шкільних буфетів.
15. Технолог разом із кухарем розробляють технологічні карти.
16. Кухар підготовлює рецепти на відібрані страви та напої.

17. МОЗ та МОН перевіряють меню на відповідність нормам і відхиляють його.
18. МОЗ і МОН формує лист з поправками для технолога і кухаря.
19. Технолог і кухар вносять поправки представленні в листі і повторно

- відправляють меню у МОЗ та МОН.
20. МОЗ та МОН повторно перевіряють меню з поправками на відповідність нормам харчування і узгоджують його.

21. Економіст обраховує орієнтовану вартість сезонного меню.
 22. Економіст розробляє систему звітності за витрачені кошти.
 23. МОЗ та МОН розглядають економічну складову і вона не відповідає

бюджету.

24. МОН приймає рішення про меню і відправляє лист.
 25. Технолог і кухар шукають альтернативне меню.
 26. Технолог відправляє нове меню в Міністерства.

27. МОЗ та МОН повторно перевіряють меню з поправками на відповідність

нормам харчування і узгоджують його.
 28. Економіст обраховує орієнтовану вартість сезонного меню.
 29. МОЗ та МОН розглядають економічну складову і погоджують її.

30. МОЗ та МОН затверджують шкільне меню.

31. Міністерство освіти і науки України розповсюджують нове меню закладам освіти.

1.2.6 **Діаграма активності.** Діаграма активності дозволяє моделювати послідовності бізнес-процесів або дій, реалізованих методами класів. Зазначені послідовності можуть являти собою альтернативні галузей процесу обробки даних або галузям, які можуть виконуватися паралельно. Діаграми діяльності є аналогом

блок-схеми будь-якого алгоритму. Вони, як і діаграми станів та переходів, відображаються у вигляді орієнтованого графу, вершинами якого є дії, а ребрами – переходи між діями. Діаграми діяльності доцільно використовувати для аналізу:

- змісту сценаріїв застосування проектованої системи;
- взаємодії потоків робіт різних сценаріїв;
- виконання сценаріїв у багатопроцесорних обчислювальних середовищах.

НУБІП УКРАЇНИ

Ці діаграми широко використовуються в описі поведінки, що включає велику кількість паралельних процесів. Діаграма активності зображена на рис. 2 відображає процес формування шкільного меню.

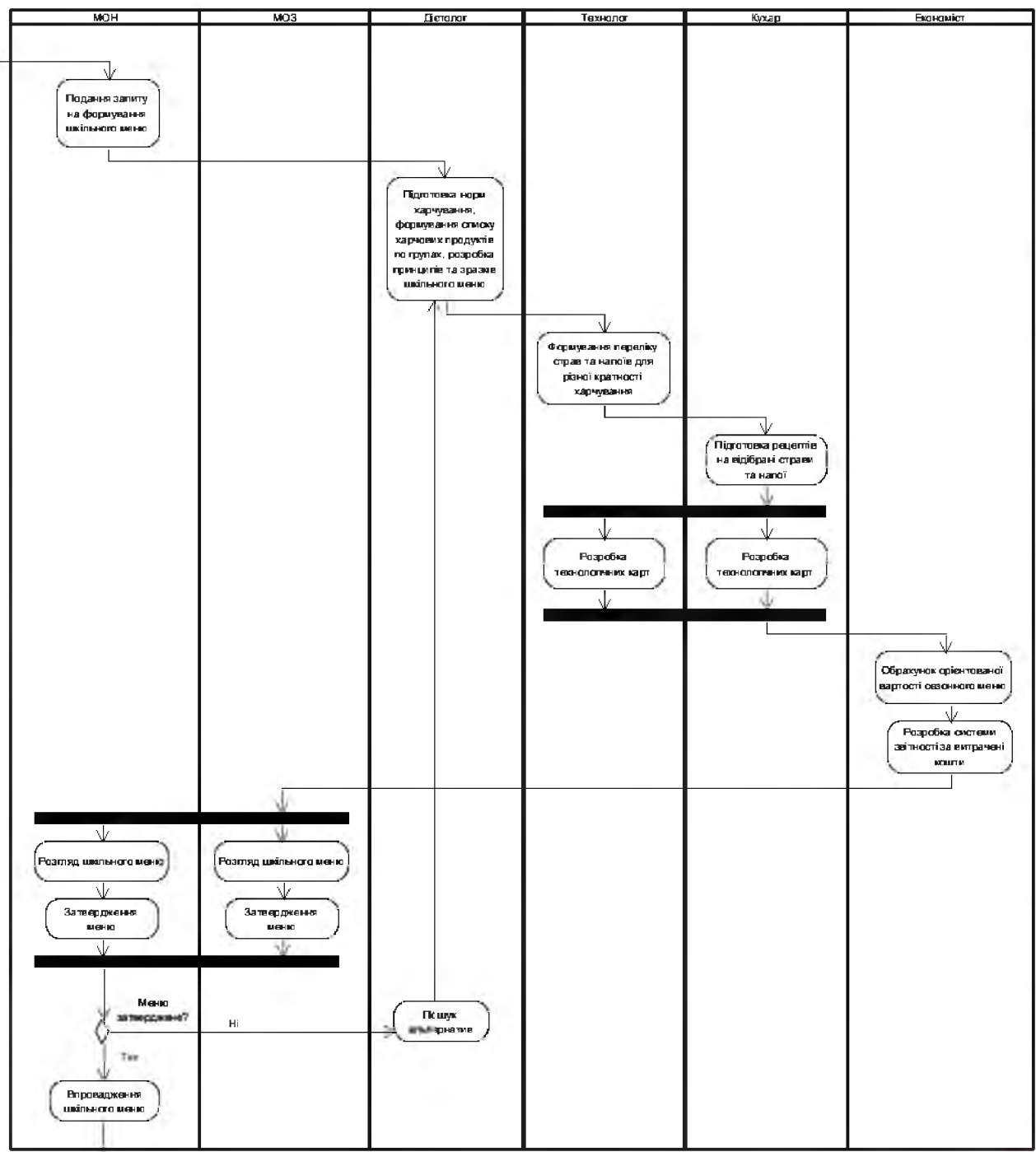
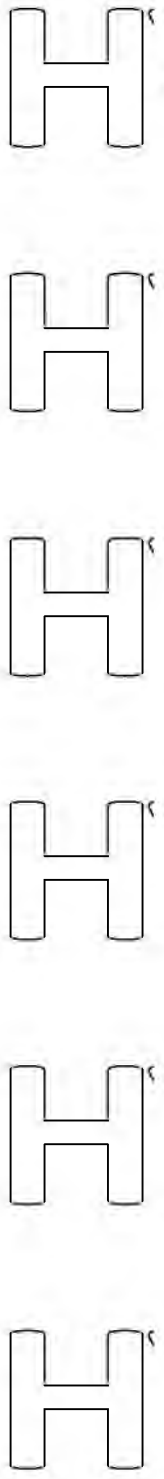
Спочатку МОН подає запит на формування шкільного меню, дієтолог розглядає його і починає роботу. Він підготовлює норми шкільного харчування, формує список харчових продуктів по групах, розробляє принципи та зразки шкільного меню. Передає свої напрацювання технологу. Технолог формує перелік страв та напоїв для різної кратності харчування і передає їх кухарю. Кухар підготовлює рецепти на відібрані страви та напої. Далі технолог і кухар, відповідно до рецептів розроблюють технологічні карти. По технологічних картах, економіст розраховує вартість мезонного меню і розробляє систему звітності за витрачені кошти.

Закінчивши роботу над меню і розрахунками вартості, готовий варіант розглядають Міністерство освіти і науки України і Міністерство охорони здоров'я, після чого меню затверджується. У разі відхилення меню хоча б одним Міністерством починається пошук альтернатив і дії повторюються знову, доки меню не буде затверджено.

Якщо меню затверджено двома Міністерствами, то МОН впроваджує це меню у закладах освіти.

НУБІП України

НУБІП України



Powered By Visual Paradigm Community Edition

Рис. 2 Діаграма активності

1.2.7 **Діаграма послідовності.** Діаграма послідовності відображає взаємодії об'єктів в порядкуванні за часом. Зокрема, такі діаграми відображають задіяні об'єкти та послідовність відправлених повідомлень.

Діаграма послідовностей відображає порядок дій та передачу повідомлень,

беручи до уваги часові особливості.

Діаграму послідовностей зображено на рис 3.

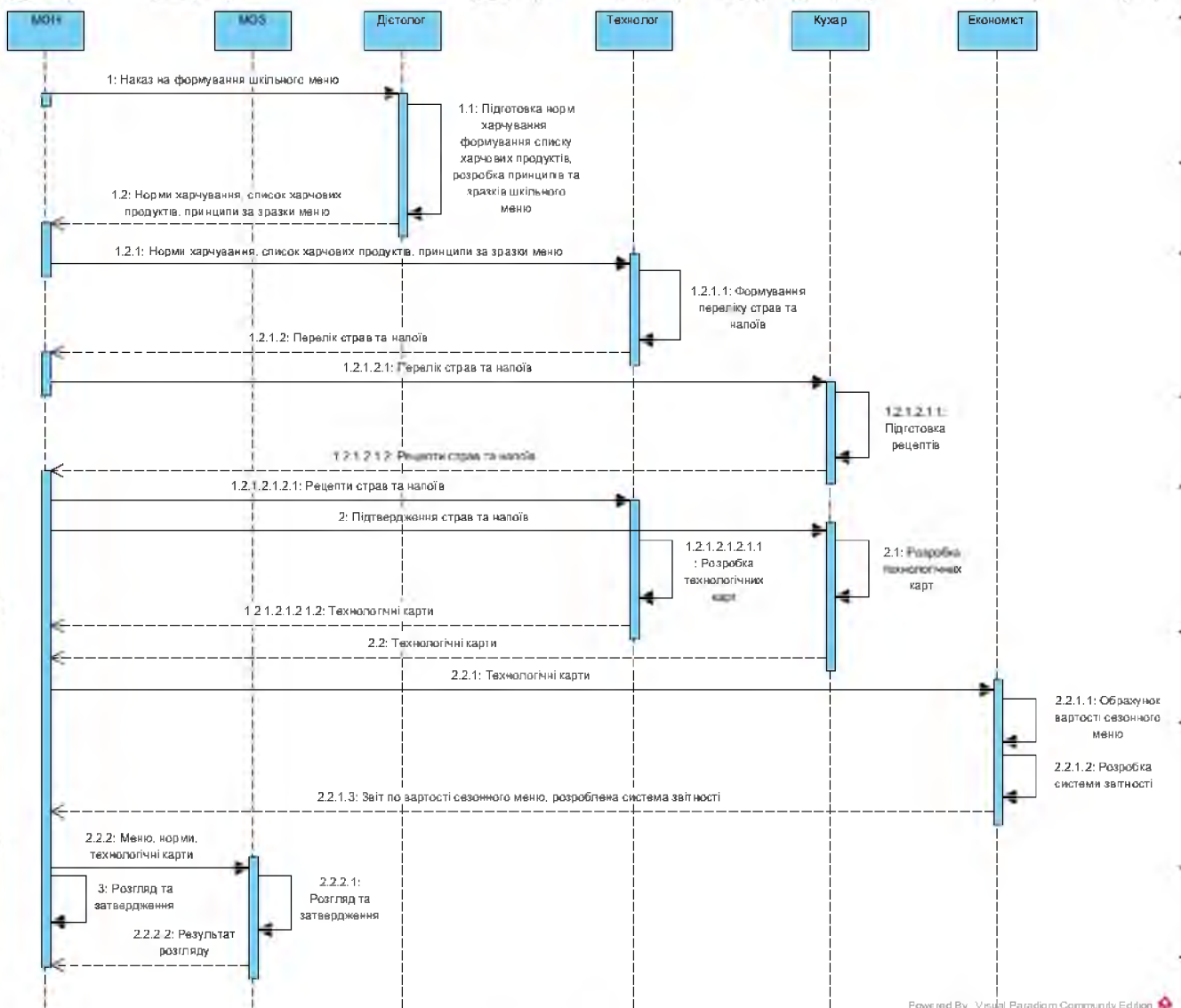


Рис. 3 Діаграма послідовностей.

Відповідно до діаграми все починається з наказу від MOH на формування шкільного меню. Дієтолог формує всі норми харчування, список харчових

продуктів та розробляє принципи і зразки шкільного меню. Після цього, відправляє це у МОН.

Міністерство освіти і науки у свою чергу отримавши дані від дієтолога відправляє їх технологу. Технолог формує перелік страв та напоїв і відправляє їх у

МОН. МОН передає отриманий перелік від технолога кухарю, який підготує список рецепти для страв та напоїв. Після підготовки рецептів кухар відправляє їх у МОН.

МОН отримавши перелік рецепти відправляє їх технологу і підтвердження кухарю, після чого вони починають розробку технологічних карт. Розробивши карти, їх відправляють назад у Міністерство.

Далі технологічні карти передають з МОН економісту, який отримавши карти, обраховує вартість сезонного меню і розробляє систему звітності за витрачені кошти. Після чого, відправляє все у МОН.

Міністерство освіти і науки отримавши звіт про вартість і розроблену систему звітності передає дані про норми харчування, меню, технологічні карти Міністерству охорони здоров'я України.

Далі МОН та МОЗ розглядають всі дані і затверджують меню. Після чого МОЗ передає у МОН результат розгляду.

НУБІП України

НУБІП України

НУБІП України

2 ТЕОРЕТИЧНІ ПЕРЕДУМОВИ

2.1 Визначення СППР

Системи підтримки прийняття рішень (СППР) — інформаційні системи, які використовують обладнання, програмне забезпечення, дані, базу моделей і роботу менеджера з метою підтримки всіх стадій прийняття рішень у процесі аналітичного моделювання. Програмні засоби включають комплекс різних алгоритмів підтримки рішень, базу моделей, базу даних, допоміжні та керівну програми, яка забезпечує процес прийняття рішень з урахуванням специфіки проблеми.

Орієнтовані на операційне управління СППР застосовуються для виконання науково-дослідних робіт, в управлінні кадрами, виробництвом тощо.

У теорії СППР на сьогодні виділяють три покоління розвитку цих систем:

перше покоління — з 1970 до 1980 р., друге — з 1980 р. до середини 90-х років, третє — з середини 90-х років ХХ ст. і донині (розроблення нових типів триває).

СППР першого покоління мають такі основні характеристики:

- управління даними — велика кількість інформації, банки даних;
- управління обчисленнями — моделі розроблялися фахівцями в галузі інформатики для спеціальних проблем;
- користувацький інтерфейс — мови програмування для великих ЕОМ, що використовуються тільки програмістами.

СППР другого покоління мають нові властивості:

- управління даними — необхідна і достатня кількість інформації відповідно до сприйняття СППР;
- управління обчисленнями — гнучкі моделі, що відповідають способові

мислення ОПР:

- інтерфейс користувача — дружній користувачеві, безпосередня робота кінцевого користувача.

До основних цілей створення СППР другого покоління належать:

- допомога ОПР у розумінні проблеми, що полягає в її структуризації, генеруванні постановок задач і формування критеріїв,
- допомога у вирішенні задач, суть якої полягає в генеруванні і/або виборі моделей і методів прийняття рішень;
- допомога в аналізі результатів, тобто проведення аналізу типу “Що буде.., якщо...”, пояснення ходу рішення, пошук і видача аналогічних рішень у минулому і їхніх наслідках.

СППР третього покоління мають ті самі ознаки, що й другого покоління, але з'явилися додаткові можливості за рахунок упровадження таких нових засобів інформаційних технологій і методів штучного інтелекту:

- сховищ і вітрин даних, що дає змогу творцям рішень аналізувати величезні обсяги даних про поточні ділові транзакції з метою вибору раціонального рішення,
- OLAP-систем, які уможливають швидке та зручне маніпулювання великими базами даних для дослідження багатьох показників бізнесової діяльності в різних ракурсах.

Вирішення задач проектування складних систем може вивчатися і розвиватися на різних рівнях узагальнення і деталізації. На найвищому рівні спільності основна увага приділяється розробці принципів організації системи та виробленню загального погляду на майбутню систему. Такі загальні аспекти побудови системи називаються архітектурою системи.

Отже, перша задача архітектора (при створенні проекту) – це визначення реальних потреб і побажань користувача. У проекті СППР виділяють три характерні рівні:

- архітектура системи – це функціональний проєкт системи з погляду користувача;

• виконання системи – це логічний опис внутрішньої структури, що робить можливим здійснення функцій, визначених архітектурою,
 • реалізація системи – це фізичне втілення виконання.

При розробці архітектури системи необхідно дотримуватися

загальноприйнятих принципів, до яких належать такі:

• погодженості – даний принцип означає, що гарна архітектура погоджена, коли часткове знання системи дозволяє передбачати й інше;

• ортогональності – цей принцип вимагає, щоб функції були незалежні одна від одної і специфіковані окремо;

• відповідності – означає, що в архітектуру варто включати тільки ті функції, що відповідають істотним вимогам до системи;

• економічності – жодна функція в описі архітектури не повинна в будь-якому вигляді дублювати іншу;

• прозорості – функції, знайдені в процесі виконання, мають бути відомі користувачеві;

• спільності – функція, що знову вводиться, повинна вводитися в такому вигляді, щоб вона відповідала якомога більшій кількості призначень;

• відкритості – користувач повинен мати можливість уточнювати специфікацію й зміст функцій системи в процесі її використання;

• повноти – специфікація функцій повинна відповідати всім вимогам і побажанням користувача.

Архітектура системи визначається функціями цієї системи та способом їх реалізації.

2.2 Алгоритми та технології аналізу даних

Аналіз даних – це процес виконання послідовних логічних дій з інтерпретації відповіді респондентів та їх перетворення у статистичні форми,

необхідні для ухвалення маркетингових та управлінських рішень. Зазвичай цей процес складається з трьох послідовних етапів (рис. 10)

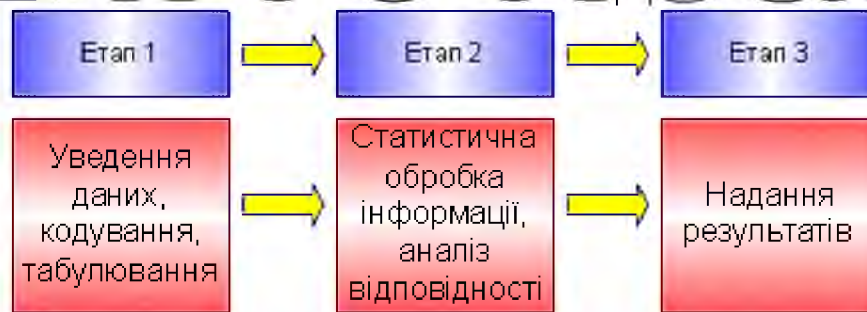


Рис. 10 Процес аналізу даних

На першому етапі дані, зібрані у процесі опитування, вводять у комп'ютер як матриці (відбувається табуляція даних). Уведення даних і їх табуляція може здійснюватися за допомогою спеціальних комп'ютерних програм, наприклад, Vortex, SPSS, Statistica, або в середовищі MS Excel чи Access. У першому, і в другому випадку уможливується сортування, форматування, відбір та інші операції оброблення даних.

Слід зауважити, що введення даних є процесом трудомістким і часто вимагає залучення спеціально підготовлених фахівців. Тому ще на етапі планування дослідження доцільно визначитися з оптимальною (а краще – мінімальною) кількістю запитань у анкеті для запобігання труднощам під час введення та аналізу отриманих даних. Проблеми загострюються, коли опрацьовувати доводиться значну кількість анкет.

На другому етапі проводять статистичний аналіз даних, пошук взаємозв'язків і відмінностей у масивах. Крім того, статистична обробка передбачає розробку висновків та гіпотез (концептуалізацію даних), одночасно проводять перевірку репрезентативності результатів, їх здатності до перенесення на всю генеральну сукупність.

Засоби аналізу даних полегшують користувачам обробку даних і маніпулювання ними, аналіз взаємозв'язків та кореляцій між наборами даних, а також допомагає виявити закономірності та тенденції інтерпретації.

Існує кілька типів методів аналізу даних, які були створені на основі бізнесу та технологій. Основними видами аналізу даних є:

- Аналіз тексту;
- Статистичний аналіз;
- Діагностичний аналіз;
- Прогностичний аналіз;
- Рекомендаційний аналіз.

Аналіз тексту. Аналіз тексту також називають Data Mining. Це один із методів аналізу даних для виявлення закономірності у великих наборах даних за допомогою баз даних або інструментів аналізу даних. Він використовувався для перетворення вихідних даних у бізнес-інформацію. На ринку присутні інструменти Business Intelligence, які використовуються для прийняття стратегічних бізнес-рішень. Загалом він пропонує спосіб вилучення та вивчення даних, виведення шаблонів і, нарешті, інтерпретації даних.

Статистичний аналіз. Статистичний аналіз показує «Що станеться?» використовуючи минулі дані у вигляді інформаційних панелей. Статистичний аналіз включає збір, аналіз, інтерпретацію, представлення та моделювання даних.

Він аналізує набір даних або вибірку даних. Існує дві категорії цього типу аналізу – описовий аналіз і аналіз висновків.

Описовий аналіз аналізує певні дані або вибірку зведених числових даних. Він показує середнє значення та відхилення для безперервних даних, тоді як відсоток і частоту для категорійних даних.

Висновок аналізує зразок з повних даних. У цьому типі аналізу ви можете знайти різні висновки з одних і тих же даних, вибравши різні зразки.

Діагностичний аналіз. Діагностичний аналіз показує «Чому це сталося?» знайшовши причину з огляду, знайденого в статистичному аналізі. Цей аналіз корисний для визначення моделей поведінки даних. Якщо у вашому бізнес-

процесі з'являється нова проблема, ви можете переглянути цей аналіз, щоб знайти подібні моделі цієї проблеми. І він може мати шанси використовувати подібні рецепти для нових проблем.

Прогнозний аналіз. Прогнозний аналіз показує, «що може статися», використовуючи попередні дані. Найпростіший приклад аналізу даних: якщо

минулого року я купив дві гри на основі своїх заощаджень, а цього року моя зарплата збільшиться вдвічі, то я можу купити чотири гри. Але, звичайно, це непросте, тому що потрібно думати про інші обставини, як-от ймовірність

зростання цін на ігри цього року або, можливо, замість ігри ви хочете купити новий телефон, або вам потрібно купити будинок.

Отже, цей Аналіз робить прогнози щодо майбутніх результатів на основі поточних або минулих даних. Прогноз – це лише оцінка. Його точність залежить від того, скільки детальної інформації у вас є і скільки ви в ній копаєтеся.

Рекомендаційний аналіз. Рекомендаційний аналіз поєднує уявлення з усіх попередніх аналізів, щоб визначити, які дії слід вжити у поточній проблемі чи рішенні. Більшість компаній, що керуються даними, використовують директивний аналіз, оскільки прогнозного та описового аналізу недостатньо для підвищення

продуктивності даних. Виходячи з поточних ситуацій і проблем, вони аналізують дані та приймають рішення.

Процес аналізу даних — це ніщо інше, як збір інформації за допомогою відповідної програми або інструменту, який дозволяє досліджувати дані та знаходити в них закономірність. На основі цієї інформації та даних ви можете приймати рішення або робити остаточні висновки.

Аналіз даних складається з наступних етапів:

НУБІП України

- збір вимог до даних;
- збір даних;
- очищення даних;

- аналіз даних;

НУБІП України

- інтерпретація даних;
- візуалізація даних;
- збір вимог до даних;

Збір вимог до даних. Перш за все, необхідно подумати навіщо робити цей

НУБІП України

аналіз даних. Необхідно вирішити, який тип аналізу даних потрібно зробити! На цьому етапі потрібно вирішити, що аналізувати і як це виміряти, зрозуміти, чому проводиться дослідження і які заходи повинні використовуватись для цього аналізу.

НУБІП України

Збір даних. Після збору вимог ви отримаєте чітке уявлення про те, які речі ви повинні виміряти і якими повинні бути ваші висновки. Тепер потрібно збирати дані на основі вимог. Коли збираються дані, слід пам'ятати, що зібрані дані мають бути оброблені або організовані для аналізу. Оскільки зібрані дані з різних джерел, необхідно вести журнал із датою збору та джерелом даних.

НУБІП України

Очищення даних. Тепер усі зібрані дані можуть бути некорисними або невідповідними для мети аналізу, тому їх слід очистити. Зібрані дані можуть містити повторювані записи, пробіли або помилки. Дані повинні бути очищені та без помилок. Цей етап необхідно виконати перед аналізом, оскільки на основі очищення даних результат аналізу буде ближче до очікуваного.

НУБІП України

Аналіз даних. Коли дані зібрано, очищено й оброблено, вони готові до аналізу. Коли маніпулюють даними, можна виявити, що є точна інформація, яка потрібна, або може знадобитися зібрати більше даних. На цьому етапі можна використовувати інструменти та програмне забезпечення аналізу даних, які допоможуть зрозуміти, інтерпретувати та робити висновки на основі вимог.

Інтерпретація даних. Після аналізу даних настав час інтерпретувати результати. Можна вибрати спосіб вираження або передачі аналізу даних або просто словами або, можливо, таблицею чи діаграмою. Потім скористатися результатами процесу аналізу даних, щоб обрати найкращий спосіб дій.

Візуалізація даних. Візуалізація даних дуже поширена у нашому повсякденному житті; вони часто представлені у вигляді діаграм і графіків. Іншими словами, дані відображаються графічно, щоб людському мозку було легше їх зрозуміти та обробити. Візуалізація даних часто використовується для виявлення невідомих фактів і тенденцій. Спостерігаючи за взаємозв'язками та порівнюючи набори даних, можна знаушу інформацію.

Висновок:

- Аналіз даних означає процес очищення, перетворення та моделювання даних для виявлення корисної інформації для прийняття бізнес-рішень
- Типи аналізу даних: текстовий, статистичний, діагностичний, прогнозний, наказовий аналіз
- Аналіз даних складається зі збору вимог до даних, збору даних, очищення даних, аналізу даних, інтерпретації даних, візуалізації даних.

Існують різноманітні методи аналізу даних, в залежності від питання, яке розглядається, типу даних та кількості даних, що зібрана. Кожен з цих методів сфокусований на стратегії вивчення нових даних, деталізації інформації для перетворення фактів та цифр у параметри прийняття рішень. Відповідно, різні методи аналізу даних можна класифікувати наступним чином:

1. Методи, засновані на математиці та статистиці:

- описовий аналіз;
- регресійний аналіз;
- факторний аналіз;
- дискримінантний аналіз;

НУБІП УКРАЇНИ

- аналіз часових рядів.
- 2. Методи, засновані на штучному інтелекті та машинному навчанні:
 - штучні нейронні мережі;

- дерева рішень;

НУБІП УКРАЇНИ

- еволюційне програмування;
- нечітка логіка.
- 3. Прийоми, засновані на візуалізації та графіках:

- Стовпчаста діаграма, гістограма;

НУБІП УКРАЇНИ

- Лінійна діаграма;
- Діаграма району;
- Кругова діаграма;

- Діаграма воронки;

НУБІП УКРАЇНИ

- Слово хмара діаграми;
- Діаграма Ганта;
- Радарна діаграма;

- Діаграма розкиду;

НУБІП УКРАЇНИ

- Діаграма міхурів;
- Датчик;
- Кадрова діаграма;

- Діаграма прямокутного дерева;

НУБІП УКРАЇНИ

• Карти

На початку двохтисячних років, через значне збільшення даних, які потребують аналізу, було введено таке поняття як «Big Data».

Big Data (великі дані) – це поєднання структурованих, напівструктурованих

та неструктурованих даних, які можуть бути видобуті для отримання інформації та використані в проєктах машинного навчання, прогнозного моделювання та інших передових програм аналітики.

НУБІП УКРАЇНИ

Системи, які обробляють і зберігають Big Data, стали загальним компонентом архітектур управління даними в великих організаціях.

Big Data часто описуються такими характеристиками (англійською – 3V):

- великий обсяг даних (Volume).

- широкий вибір типів даних, що зберігаються в системах великих даних (Variety).

- швидкість, з якою дані генеруються, збираються та обробляються (Velocity)

Нещодавно до опису того, що таке Big Data, було додано кілька нових V:

- достовірність (Veracity)

- цінність (Value)

- мінливість (Variability)

Обсяг є найбільш часто цитованою характеристикою великих даних.

Говорячи про те, що таке Big Data, слід пам'ятати, що великі дані – це сукупність даних з різних джерел, починаючи від чітко визначених і закінчуючи

слабко визначеними. Вони походять від людських або машинних джерел.

Різноманітність даних. Big Data також охоплюють широкий спектр типів даних, включаючи наступні:

- структуровані дані в базах даних та сховищах даних на основі мови структурованих запитів (SQL);

- неструктуровані дані, такі як текстові та файли документів, що зберігаються в кластерах Hadoop або системах баз даних NoSQL.

- напівструктуровані дані, такі як журнали веб-сервера або потокові дані з датчиків.

Всі різні типи даних можна зберігати разом в озері даних, яке, як правило, базується на Hadoop або службі зберігання хмарних об'єктів (що таке хмарні технології).

Крім того, програми для Big Data часто містять кілька джерел даних, які в іншому випадку не можуть бути інтегровані.

Наприклад, проект аналітики великих даних може спробувати оцінити успіх товару та майбутні продажі, співвідносячи дані про минулі продажі, дані про повернення та дані огляду онлайн-покупиць цього товару.

Швидкість відноситься до швидкості, з якою генеруються великі дані і вони повинні бути оброблені та проаналізовані.

У багатьох випадках набори великих даних оновлюються в режимі майже реального часу, замість щоденних, щотижневих або щомісячних оновлень, характерних багатьом традиційним сховищам даних.

Програми аналітики великих даних співвідносять та аналізують вхідні дані, а потім надають відповідь або результат на основі запиту. Це означає, що аналітики даних повинні детально розуміти наявні дані та мати певне розуміння того, які відповіді вони шукають, щоб переконатися, що отримана інформація є дійсною та актуальною.

Управління швидкістю передачі даних також має важливе значення, оскільки аналіз великих даних поширюється на такі сфери, як машинне навчання та штучний інтелект, де аналітичні процеси автоматично знаходять закономірності у зібраних даних та використовують їх для отримання знань.

Хоча великі дані кількісно не прирівнюються до будь-якого конкретного обсягу даних, розгортання великих даних часто включає в себе терабайти (TB), петабайти (PB) і навіть екзабайти (EB) даних.

Важливість великих даних. Компанії використовують накопичені в їх системах Big Data для поліпшення операцій, забезпечення кращого обслуговування споживачів, створення персоналізованих маркетингових кампаній на основі конкретних уподобань клієнтів і, зрештою, підвищення прибутковості.

Підприємства, які використовують великі дані, мають потенційну конкурентну перевагу перед тими, хто цього не робить. Вони можуть приймати швидші та більш обгрунтовані ділові рішення, за умови, що вони ефективно використовують дані.

Наприклад, Big Data можуть надати компаніям цінну інформацію про своїх клієнтів. Вона може бути використана для вдосконалення маркетингових кампаній з метою збільшення залучення клієнтів та коефіцієнтів конверсії.

Крім того, використання великих даних дозволяє компаніям дедалі краще орієнтуватися на споживача.

Історичні дані та дані в реальному часі можуть бути використані для оцінки миттєвих уподобань споживачів. Це дозволяє підприємствам оновлювати та вдосконалювати свої маркетингові стратегії та ставати більш чутливими до бажань та потреб клієнтів.

Великі дані також використовуються медичними дослідниками для виявлення факторів ризику захворювання та лікарями для діагностики захворювань та станів у окремих пацієнтів.

Крім того, дані, отримані з електронних медичних записів, соціальних мереж, Інтернету та інших джерел, надають організаціям охорони здоров'я та державним установам найсвіжішу інформацію про загрози інфекційних захворювань чи спалахи захворювання.

В енергетичній галузі Big Data допомагають нафтогазовим компаніям визначати потенційні місця буріння та контролювати експлуатацію трубопроводів. Так само комунальні служби використовують їх для спостереження за електричними мережами.

Фірми фінансових послуг використовують системи Big Data для управління ризиками та аналізу ринкових даних у реальному часі.

Виробники та транспортні компанії покладаються на великі дані для управління своїми ланцюгами поставок та оптимізації шляхів доставки.

Інші сфери використання включають – реагування на надзвичайні ситуації, запобігання злочинності та побудова розумних міст.

Методи Data Mining активно використовуються для пошуку раніше не відомих залежностей у великих масивах даних.

Data mining (інтелектуальний аналіз даних) це процес, який використовується компаніями для перетворення необроблених даних в корисну

інформацію. Використовуючи програмне забезпечення для пошуку закономірностей у великих пакетах даних, підприємства можуть дізнатися більше про своїх клієнтів, щоб розробити ефективніші маркетингові стратегії, збільшити продажі та зменшити витрати. Інтелектуальний аналіз даних залежить від ефективного збору, зберігання та комп'ютерної обробки даних.

Існує багато методів Data Mining, які можна використовувати, задля того, щоб перетворити необроблені дані на практичну інформацію, наприклад:

- очищення та підготовка даних;

- відстеження шаблонів;

- класифікація;

- асоціація;

- кластеризація;

- регресія;

- прогнозування;

- послідовні шаблони;

- дерева рішень;

- статистичні методи;

- візуалізація;

- нейронні мережі;

- машинне навчання та штучний інтелект.

В цій магістерській роботі для аналізу даних було обрано Big Data.

2.3 Нормативно-правові документи

Щільне харчування регламентується та контролюється, рядом нормативно-правових документів на рівні держави:

- Постанова Кабінету Міністрів України від 24 березня 2021 року № 305 «Про затвердження норм та Порядку організації харчування у закладах освіти та дитячих закладах оздоровлення та відпочинку».

<https://zakon.rada.gov.ua/laws/show/305-2021-%D0%BF#Text>

- Постанова Кабінету Міністрів України від 28 липня 2021 року № 786 «Про внесення змін до норм харчування у закладах освіти та дитячих закладах оздоровлення та відпочинку»

<https://zakon.rada.gov.ua/laws/show/786-2021-%D0%BF#n9>

- Наказ Міністерства охорони здоров'я України від 03 серпня 2017 року № 1073 «Про затвердження Норм фізіологічних потреб населення України в основних харчових речовинах і енергії»

<https://zakon.rada.gov.ua/laws/show/z1206-17#n14>

- Наказ Міністерства освіти і науки України та Міністерства охорони здоров'я від 01 червня 2005 року № 242/329 «Про затвердження Порядку організації харчування дітей у навчальних та оздоровчих закладах»

<https://zakon.rada.gov.ua/laws/show/z0661-05#Text>

- Наказ Міністерства освіти і науки України та Міністерства охорони здоров'я від 15 серпня 2006 року № 620/563 «Щодо невідкладних заходів з організації харчування дітей у дошкільних, загальноосвітніх, позашкільних навчальних закладах»

<https://zakon.rada.gov.ua/rada/show/%20%0620290-06#Text>

НУБІП УКРАЇНИ

- Постанова Кабінету Міністрів України від 19 червня 2002 року № 856 «Про організацію харчування окремих категорій учнів у загальноосвітніх навчальних закладах»

<https://zakon.rada.gov.ua/laws/show/856-2002-%D0%BF#Text>

НУБІП УКРАЇНИ

- Із змінами, внесеними згідно з Постановою Кабінету Міністрів № 850 (850-2003-п) від 04.06 2003

<https://zakon.rada.gov.ua/laws/show/850-2003-%D0%BF#Text>

- Гігієнічний стандарт: Проект Санітарного регламенту для закладів загальної середньої освіти

https://moz.gov.ua/uploads/1/6845-pro_20180828_1.pdf

Додаток 4. Рекомендації із здорового харчування дітей в загальноосвітніх навчальних закладах

https://moz.gov.ua/uploads/1/6846-pro_20180828_1_dod.pdf

НУБІП УКРАЇНИ

2.4 Норми щоденного харчування

Енергетична та поживна цінність їжі (вміст макро- і мікронутрієнтів) у

закладах дошкільної, загальної середньої освіти та інших закладах освіти, що провадять освітню діяльність на певному рівні (рівнях) повної загальної середньої освіти, дитячих закладах оздоровлення та відпочинку повинна відповідати загальним віковим потребам здобувачів освіти/дітей згідно з нормами

фізіологічних потреб в основних харчових речовинах та енергії, визначеними МОЗ.

Орієнтовна кількість калорій на сніданок, обід та вечерю, добова калорійність для різних вікових груп наведена в таблиці 1.

НУБІП УКРАЇНИ

Таблиця 1

Вікова група	Енергетична цінність сніданку, вечері, ккал	Енергетична цінність обіду, ккал	Добова енергетична цінність, ккал
1-4 роки	350-415	415-485	1385
4-6 (7) років	425-510	510-595	1700
6-11 років	525-630	630-735	2100
11-14 років	600-720	720-840	2400
14-18 років	675-810	810-945	2700

Орієнтовна кількість білків для сніданку, обіду та вечері, а також добова потреба в білку для різних вікових груп наведені у таблиці 2.

Таблиця 2

Вікова група	Кількість білку на сніданок, вечерю, грамів	Кількість білку на обід, грамів	Добова потреба у білку, грамів
1-4 роки	13-15	15-18	53
4-6 (7) років	14-17	17-20	58
6-11 років	18-21	21-25	72
11-14 років	21-25	25-29	84
14-18 років	23-27	27-32	93

Орієнтовну кількість жирів для сніданку, обіду та вечері для різних вікових груп наведено у таблиці 3.

Таблиця 3

Вікова група	Кількість жирів на сніданок, вечерю, грамів	Кількість жирів на обід, грамів	Добова потреба у жирах, грамів
1-4 роки	11-13	13-15	44
4-6 (7) років	14-16	16-19	56
6-11 років	17-21	21-24	70
11-14 років	21-25	25-29	84
14-18 років	23-27	27-32	92

Орієнтовну кількість вуглеводів для сніданку, обіду та вечері для різних вікових груп наведено в таблиці 4.

Таблиця 4

Вікова група	Кількість вуглеводів на сніданок, вечерю, грамів	Кількість вуглеводів на обід, грамів	Добова потреба у вуглеводах, грамів
1-4 роки	48-58	58-67	194
4-6 (7) років	60-72	72-84	240
6-11 років	73-88	88-103	295
11-14 років	81-98	98-114	327
14-18 років	93-112	112-131	375

Під час організації в закладах одно-, дво-, три-, чотири- або п'ятиразового харчування розрахунок калорійності за окремими прийомами їжі, кількості білків, жирів та вуглеводів проводиться відповідно до таблиці 5.

Таблиця 5

Прийоми їжі	Граничні норми добової калорійності для відповідної енергетичної та поживної (кількість білків, жирів та вуглеводів) цінності, відсотків
Сніданок	25-30
Обід	30-35
Сніданок та обід	55-65
Сніданок, обід та вечеря	80-100
Сніданок, другий сніданок, обід та вечеря	85-100
Сніданок, другий сніданок, обід, підвечірок, вечеря	90-100

НУБІП України

НУБІП України

НУБІП України

НУБІП України

3 РОЗРОБКА СИСТЕМИ

3.1 Архітектура системи

Архітектура програмного забезпечення - це структура програми або обчислювальної системи, яка включає програмні компоненти, видимі зовні властивості цих компонентів, а також відносини між ними.

Для розробки програмної системи використовувалась мікросервісна архітектура та архітектурний стиль взаємодії компонентів у розподілених мережах

REST.

Мікросервісна архітектура передбачає поділ функціоналу системи на сервіси, кожен з яких являється окремим додатком та може працювати не залежно від інших.

До основних переваг такої архітектури належать:

- відмовостійкість;
- можливість масштабування;
- гнучкість (відкритість до змін);
- легкість підтримки.

REST — це архітектурний стиль, для побудови розподілених системи, сама аббревіатура розшифровується, як **RE**presentational **S**tate **T**ransfer і перекладається як передача стану через представлення.

Використання цього стилю, дає можливість передавати стан будь-якого об'єкту. Зазвичай він передається у вигляді JSON. Таким чином можна будувати веб-сервіс, який на кожний запит буде повертати, JSON з інформацією. А перевага використання JSON, що він являється об'єктом JS і його відобразити буде не важко.

На рис. 4 зображено архітектуру програмної системи, де можна розділити систему на дві частини, це Front-end (Веб-додаток і мобільний додаток) і Back-end (набір сервісів з якими взаємодіють клієнти).

На рис. 4 зображено наступні сервіси: авторизації, профайлів, страв, словник, меню, зворотнього зв'язку, замовлень, аналітики, обробки даних. А також API Gateway, Kafka та декілька баз даних.

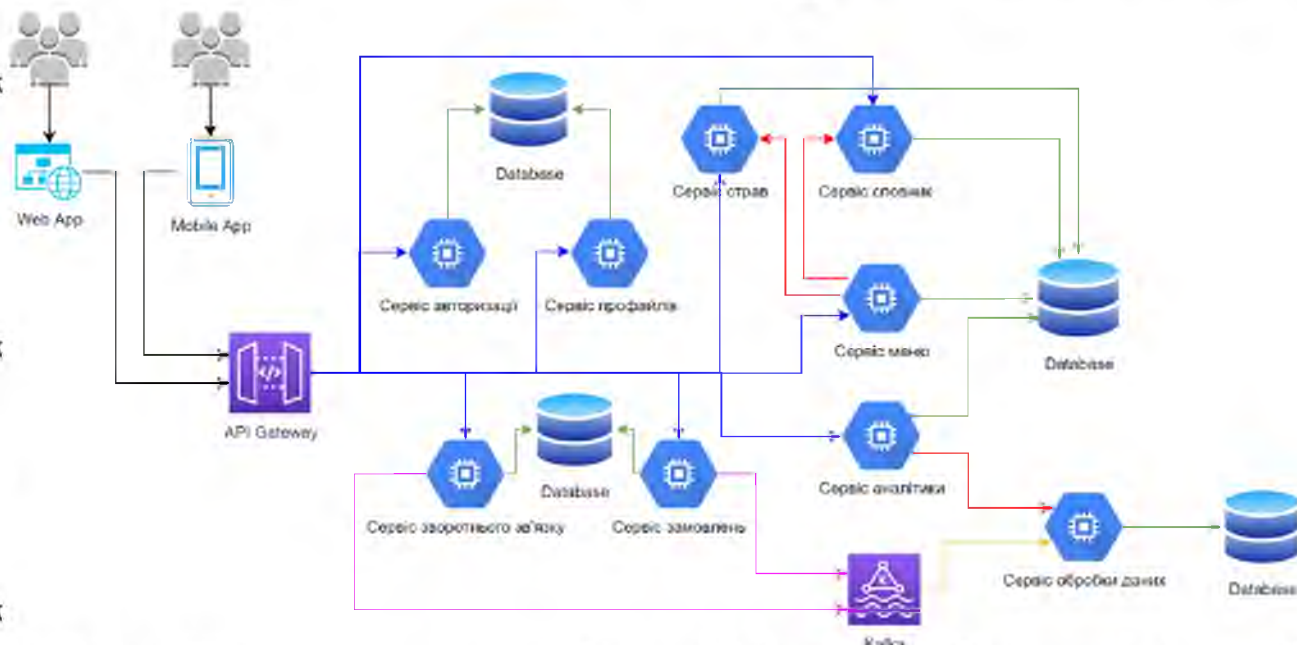


Рис. 4 Архітектура системи

API Gateway – точка входу у систему з публічної мережі, у ролі шлюзу виступає NGINX налаштований, як зворотній проксі, він надає маршрути для сервісів всередині мережі, а також перенаправляє запити з публічної мережі на відповідні сервіси. При масштабуванні виступає як балансер навантаження.

Сервіс авторизації – сервіс, що відповідає за авторизацію та реєстрацію користувачів. Працює на технологіях OAuth2 і JWT. Тобто при запиті на авторизації у системі при валідних даних користувача він формує токен у якому містить інформація про користувача, далі з цим токеном користувач може отримати доступ до інших сервісів, якщо це передбачено клієнтом, який він використовує, і його роллю в системі.

Сервіс профайлів – сервіс, що відповідає за маніпуляції з даними користувача типу ім'я, прізвище, місце роботи і т.д.

Сервіс страв – сервіс, який відповідає за маніпулювання інформацією про страви, які можуть бути у шкільному меню, склад страв і рецепт приготування.

Сервіс словник – сервіс, що містить загальну інформацію, що рідко змінюється або не змінюється взагалі: області, райони, міста, норми харчування, статті, тощо.

Сервіс меню – сервіс, що відповідає за формування і доступ до меню.

Сервіс зворотнього зв'язку – сервіс, що забезпечує зворотній зв'язок від учнів та їх батьків.

Сервіс замовлень – сервіс, що відповідає за отримання інформації про кількість приготовлених та реалізованих порцій страви з шкільного меню.

Kafka – це брокер повідомлень, через який передаються повідомлення від сервісів зворотнього зв'язку та замовлень до сервісу обробки даних.

Сервіс обробки даних – сервіс, який працює на технології BigData і відповідає, за обробку повідомлень отриманих через брокер повідомлень, виділяє аналітичні дані та зберігає їх до бази даних.

Сервіс аналітики – сервіс, який відповідає за отримання актуальної аналітики від сервісу обробки даних, та надає доступ до отриманих даних іншим сервісам та користувачам.

Одним з найважливіших етапів проектування та розробки системи, є розбиття великої системи на підсистеми. З цією метою існує діаграма пакетів. Діаграма пакетів використовується для групування класів та компонентів системи у пакети.

Групування у пакети виконується за спільними ознаками, наприклад, належність до однієї підсистеми, набір класів чи компонентів, які виконують подібні функції, коли класи чи компоненти можна узагальнити чимось спільним.

Пакет – це елемент моделі для угруповання взаємозалежних елементів. Він відповідає представленню пакета у таких мовах, як Java або .NET.

Розбиття програми на пакети, дозволяє, структурувати програми, та полегшує подальшу підтримку системи, при розбитті потрібно правильно під брати назви для пакетів, щоб не отримати зворотній ефект. При побудові великих систем, це є обов'язковий етап, адже з часом кількість файлів буде надзвичайно велика, і без групування знайти потрібний буде важко.

У файловій системі пакет представляється, як звичайний каталог, який може містити файли або ж інші каталоги.

Діаграма пакетів зображена на рис. 5.

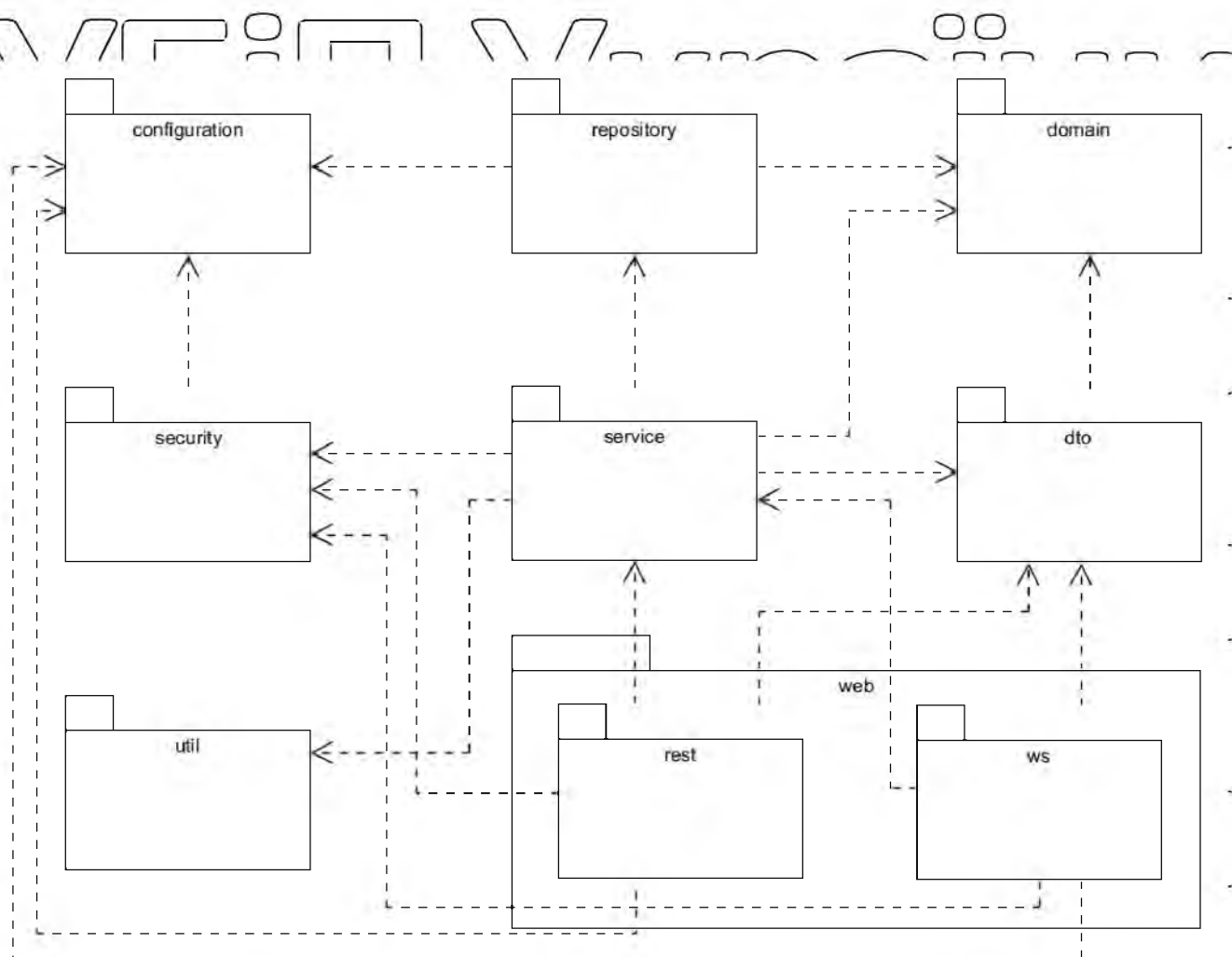


Рис. 5 Діаграма пакетів

Діаграма пакетів містить:

configuration – пакет в якому зберігаються класи, що виконують конфігурування сервісу під особливості бізнес-логіки,

repository – пакет, що містить інтерфейси та запити для взаємодії з БД.

domain – пакет, що містить класи-моделі, що є проєкціями на сутності у БД.

security – пакет, який містить класи, що відповідають за безпеку даних та доступ у сервісі.

service – містить сервіси для роботи з різними елементами системи,

реалізують бізнес-логіку системи.

util – класи без стану, для виконання нескладних задач

dto – класи, що мають спільні поля з моделями, використовуються для спілкування з клієнтами та сервісами.

web – пакет у якому зібрані класи для роботи з клієнтами та іншими сервісами містить у собі пакети rest та ws.

rest – пакет, що містить контролери для роботи з клієнтами.

ws – пакет, що містить пакети для реалізації взаємодії через сокети.

3.3 Інструменти та технології

Перед початком розробки програмного забезпечення потрібно обрати інструментарій для його створення. Інструменти та технології слід вибирати опираючись на спроектовані моделі, вимоги, архітектуру та проблему, яку вирішуватиме система.

Розроблюване ПЗ повинно функціонувати в мережі Інтернет, тому потрібно обрати технології, що забезпечать стабільну роботу при роботі з багатьма клієнтами.

При розробці програмної системи використовувалися різні інструменти та технології в залежності від призначення сервісу.

Всі сервіси було розроблено на мові програмування Java. Вибір впав на цю мову тому, що вона є крос платформною, стійкою до навантажень, стабільною у роботі та у поєднанні з фреймворком Spring дозволяє створити веб-сервіс, який буде надавати інтерфейси для роботи з ним за допомогою мережі.

Spring – це фреймворк побудований на мікро-сервісній архітектурі, який призначений для розробки різних програмних продуктів, проте зазвичай він використовується при побудові веб-рішень. Цей фреймворк містить велику кількість модулів, доступ до яких можна отримати в будь-який час та з легкістю

добавити їх до ПЗ. Spring побудований з використанням рефлексії, що надає змогу фреймворку дописувати код під час компіляції та виконання додатку. Модулі, що використовувались при розробці:

Spring Boot – каркас для побудови рішень на базі фреймворка Spring, це основний модуль, якому передається управління сервісом;

Spring Data JPA – модуль для роботи з БД, який містить в собі ORM Hibernate, JPA і набір класів та інтерфейсів для швидкого створення репозиторіїв, що працюють з сутностями БД.

Spring Web – модуль для побудови веб-рішень, в тому числі веб-сервісів та веб-сайтів, також містить версію серверу Tomcat, що дозволяє уникати труднощі пов'язані з налаштуванням веб сервера під час розгортання ПЗ.

Spring Security Starter – модуль для захисту системи та розподілу користувачів, управління доступом за допомогою прав та ролей. Цей модуль адаптує сервіс та надає базовий функціонал для забезпечення захисту даних у системі.

Spring Security OAuth2 – модуль, який дозволяє реалізувати технологію OAuth2 для захисту даних у розподілених системах, дозволяє генерувати токен, який є обов'язковим для доступу до даних. Тобто авторизація виконується без створення сесії, як у випадку з серверами, які працюють за архітектурою MVC.

Spring Actuator – модуль, що дозволяє отримати дані про стан сервісу, використовується для опитування сервісів, щоб перевірити чи вони запустилися і чи працюють.

Spring Kafka – модуль для роботи з брокером повідомлень Kafka. Надає інструменти та базові інтерфейси для під'єднання та відправлення та отримання повідомлень з брокера.

Spring Boot Test – модуль, який містить набір інструментів для тестування сервісу і перевірки його працездатності.

Spring Security Test – модуль, який дозволяє протестувати безпеку сервісу.

Spring Boot Dev tools – інструменти для розробки, що дозволяють ефективніше використовувати дебагер та вносити зміни без перезапуску всього додатку.

Для побудови та комфортної роботи з залежностями використовувався пакетний менеджер Maven. Пакетний менеджер дозволяє з легкістю встановлювати бібліотеки просто вказавши її назву в конфігураційному файлі, після чого вона буде завантажена з центрального репозиторію, під'єднана та готова до використання, а при перенесенні процесу розробки з одного комп'ютера на інший, пакетний менеджер дозволить не переносити бібліотеки, які зазвичай містять багато дрібних файлів, копіювання яких займає багато часу, а завантажити їх. Також надає можливість докейнеризації додатків, що дозволяє створювати докер-образи, які можна розгортати у контейнерах без лишніх дій.

Liquibase – бібліотека для контролю міграцій до БД. Дозволяє описувати запити у форматі XML, а після запуску вони автоматично виконуються і оновлюють БД до актуальної версії. Також зникає залежність від СУБД, адже запит генерується бібліотекою при старті відповідно до використовуваного драйверу.

NGINX – веб-сервер, який може працювати в ролі зворотнього проксі та в якості балансера навантаження, надає можливість перенаправляти та балансувати запити між сервісами для підвищення відмовостійкості.

Docker-compose – утиліта, яка дозволяє описати порядок завантаження сервісів та налаштування для них, після чого запуск всієї системи виконується однією командою.

Logback – бібліотека, що дозволяє написати конфігурацію для журналювання у системі, а також записувати логи у файл, консоль або ж відправляти у брокер повідомлень. Також дозволяє змінити і сам формат логів, наприклад, у брокер повідомлень відправляти у форматі JSON.

Також при розробці використовувалась система контролю версій Git. Вона дозволяє керувати версійністю додатку, зберігати код у репозиторії на сервері компанії, таким чином створюється резервна копія. Система контролю версій дозволяє відмінити останні зміни за необхідності та повернутись до попередніх версій, це може бути дуже важливо при виникненні критичних помилок, які дуже важко виправити.

Для написання коду використовувалось інтегроване середовище IntelliJ IDEA від компанії JetBrains. Це середовище містить інструменти для відлагодження та розробки Java-додатків, функції автодоповнення, генерування фрагментів коду, засоби для тестування REST додатків, роботи з БД та інші.

3.4 Налаштування взаємодії розподіленої системи

Про побудову розподіленої системи однією з головних проблем є налаштування взаємодії між сервісами, адже не рідко вони знаходяться не лише в різних місцях географічно, а і у різних часових поясах.

Тут варто подбати про декілька ключових моментів:

- використовувати один часовий пояс у форматі UTC, рекомендовано без зміщення, як основний для всієї системи, а при роботі з клієнтом приводити цей час у його часовий пояс;

- при взаємодії з сервісами пам'ятати про затримки мережі, надвеликі вибірки з непотрібною інформацією використовувати не потрібно, так як і безліч маленьких вибірок. Потрібно витягувати з сервісів лише ту інформацію, яка потрібна в даний момент часу;

- не варто забувати про кешування, задля збільшення швидкості роботи і зменшення кількості запитів, потрібно кешувати відповіді, які навряд зміняться у найближчий час, при цьому можна очищати кеш сервісу, наприклад, один раз в добу, або встановлювати термін дії для кешу в розмірі декількох годин після чого він біде очищатись автоматично.

Для організації взаємодії між сервісами, використовувався RestTemplate, це частина фреймворку Spring, яка працює в якості HTTP-клієнта. Він дозволяє кодувати запити в єдиному форматі (UTF-8), і працювати з іншими сервісами, а при відсутності з'єднання опрацювати виключну ситуацію і вберегти систему від падіння.

Також використовувався брокер повідомлень Kafka, для передані великої кількості інформації з сервісу до сервісу.

3.5 Безпека даних

При побудові, будь-якої, системи важливим аспектом є безпека даних, під якою розуміється збереження ціліності даних, конфіденційності, доступності та захищеність від несанкціонованого доступу, логування змін користувачами.

Набір цих базових функцій дозволить вберегти систему від проблем з неправильними обрахунками та втручанням користувачів, без доступу, у роботу системи.

Клієнт – це додаток, з якого користувач отримує доступ до ресурсів системи.
 Користувач – це особа, яка зареєстрована в системі, і хоче отримати доступ через клієнт.

Для забезпечення безпеки даних у системі існує розподіл на ролі. Тобто кожен користувач, який хоче отримати доступ до якоїсь не публічної інформації, або внести, будь-які, дані повинен пройти процес авторизації в системі. Це означає, що користувач за допомогою клієнта, який також зареєстрований в системі, має авторизуватись, за допомогою, свого логіну і паролю. Після чого в системі згенерується токен відповідно до дозволів клієнта і користувача.

Токен містить базову інформацію, таку як, логін користувача, перелік ресурсів до яких має доступ клієнт, ролі користувача, термін дії токена.

Токен, шифрується кодуванням Base64 і підписується приватним RSA-ключем.

Для доступу до, будь-якого, ресурсу користувач має надати свій персональний токен, який на сервісі буде розшифровано, пройде етап валідації публічним RSA-ключам, після чого буде перевірений на сервісі авторизації. Якщо ключ є повністю валідним, і користувач має права на доступ до ресурсу, то тоді сервіс виконає всі необхідні дії і поверне користувачу дані, при цьому буде записано, на яку точку звертався користувач, час звернення, логін користувача.

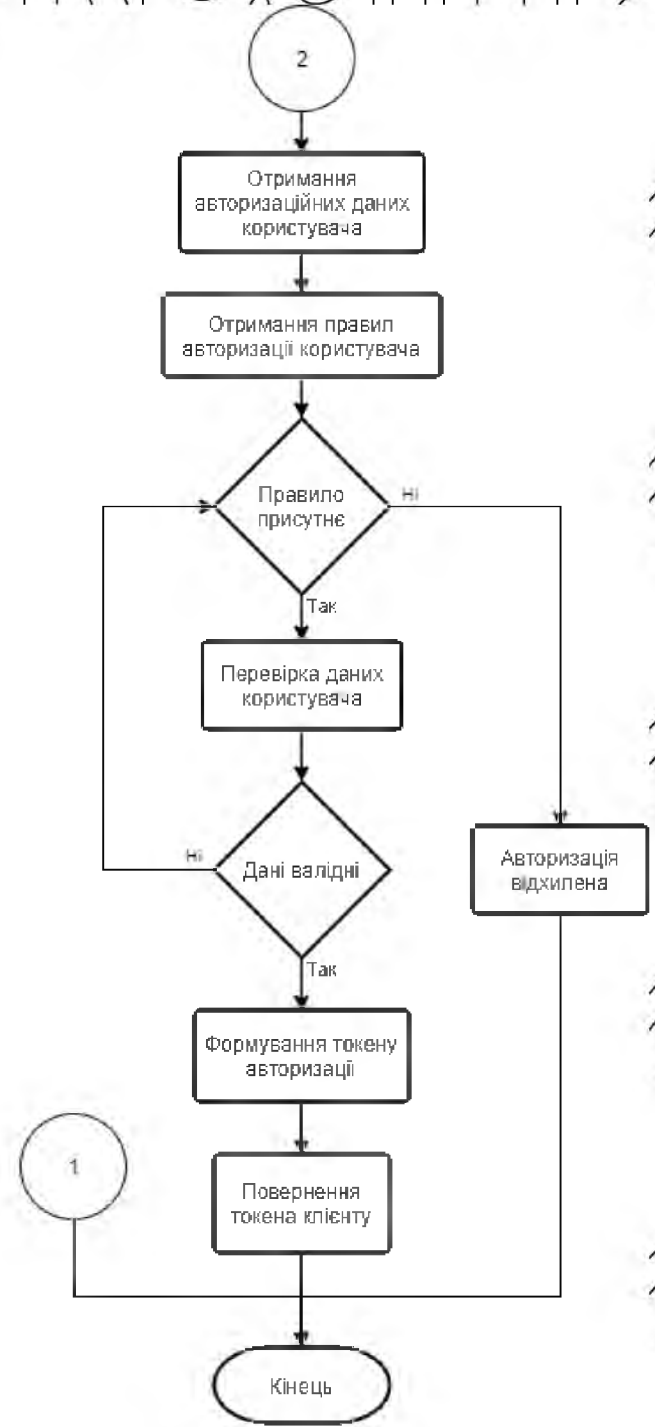
НУБІП України

НУБІП України

3.6 Алгоритмізація програмних модулів

3.6.1 Алгоритм авторизації користувача

Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition

Рис. 6 Алгоритм авторизації користувача

Алгоритм авторизації користувача (рис. 6) починається з вводу авторизаційних даних користувачем у програмі-клієнті. Після натискання кнопки

«Увійти», або іншого тригера на початок авторизації, клієнт повинен додати у запит заголовок, зі своїми закодованими авторизаційними даними, кодування Base64, а також поле з типом авторизації. Потім клієнт відправляє запит на сервіс авторизації.

Сервер авторизації, отримавши запит, спочатку витягує дані клієнта, потім знаходить всі правила авторизації для клієнта і перевіряє їх на валідність за кожним правилом, доки не буде позитивний результат, або не закінчатся правила. У негативному випадку авторизація буде відхилена, за спробою доступу через неавторизований клієнт.

У позитивному випадку сервіс авторизації, витягує дані користувача, і тип авторизації, знаходить правила для конкретного типу авторизації користувачів. Після чого перевіряє чи є такий користувач в системі і чи відповідають отримані дані, тим які є в системі, за кожним правилом. Якщо користувач присутній і дані валідні генерується токен і повертається користувачу, а якщо ні, і правила закінчилися то користувач отримує відмову.

Цей алгоритм є реалізацією технології OAuth2 для забезпечення безпеки даних та контроль доступу у розподілених системах та реалізує принцип авторизації без використання сесії. Що також дозволяє не прив'язуватись до архітектури типу MVC.

Такий підхід дозволяє реалізувати безліч способів авторизації у системі. Наприклад, стандартний спосіб авторизації з використанням БД, авторизація доменних користувачів з Active Directory через LDAP, авторизація через сторонні сервіси (Google, Facebook, LinkedIn, Twitter і т.д.), або написати, якийсь свій персональний спосіб авторизації користувачів.

Реалізація цього підходу надає гнучкість системи безпеки і захист на рівні всієї системи, а не кожного сервісу окрему.

3.6.2 Алгоритм аналізу даних. Алгоритм, що використовується для аналізу

даних представлений на рис. 7.

У системі існують сервіси, які відправляють повідомлення у брокер повідомлень, які потрібно обробляти, та зберігати корисну інформацію.

Якщо сервіс обробки даних працює, то він читає дані з брокера повідомлення, і у паралельному режимі, оброблює прочитані повідомлення, спочатку він встановлює сервіс з якого прийшло повідомлення, якщо це сервіс зворотнього зв'язку, то з повідомлення отримується відгук, а саме, оцінка якості від користувача, потім цей відгук асоціюється з учбовим закладом та стравами і

зберігається до БД. А якщо це сервіс замовлень, то отримуються дані замовлення, які також асоціюються з учбовим закладом і стравами, після чого ці дані також зберігаються. Після чого починається нова ітерація і цикл працює доки сервіс запущений.

Паралельно з цим виконується ще один цикл, який спрацьовує кожну хвилину і підсумовує дані отримані з брокера повідомлень.

Спочатку витягуються всі дані з БД, що були записані, групуються, знаходиться сума кількісних показників. Після чого з БД береться інформація по групах за день і до них добавляється нова інформація і записується у БД. Після чого вхідні дані видаляються з БД. Цей цикл також працює поки запущений сервіс обробки даних.

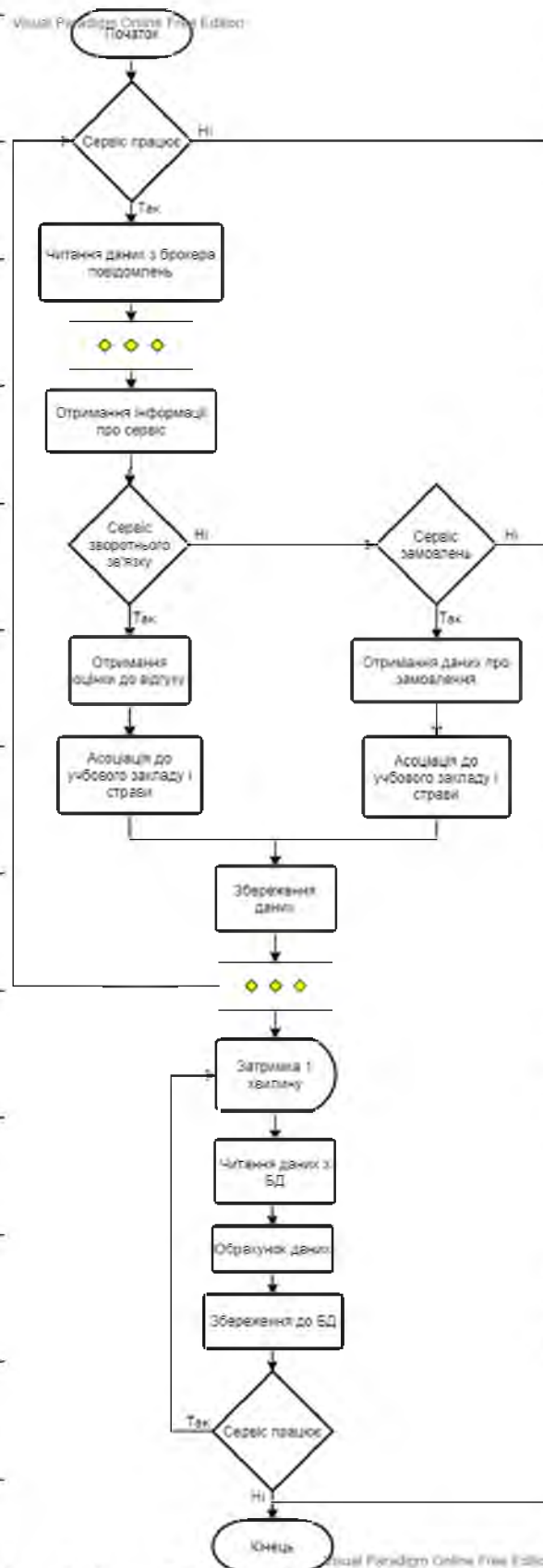


Рис. 7 Алгоритм аналізу даних

3.6.3 Алгоритм формування меню. Алгоритм формування меню представлений на рис. 8.

Після виклику користувачем функції формування нового меню, сервіс звертається до сервісу словника і отримує від нього дані по нормам харчування.

Потім він звертається до сервісу страв і отримує повний набір страв. Діх інгредієнтів, а після цього отримує дані з сервісу аналітики про вподобання учнів у конкретному регіоні.

Далі сервіс підбирає страви і формує меню максимально наближене до норм харчування. При чому страви не повинні співпадати з минулим меню.

Після формування основного меню алгоритм знаходить альтернативи для кожної страви чи напою з подібним складом і калорійністю, а також добавляє позицію, яка відрізняється за складом і повертає всі ці дані користувачу.

Користувач, ознайомившись з меню і альтернативами може замінити страви у меню з необхідності, або сформувавши повторно меню повністю або частково, якщо цей варіант йому не підійшов.

Якщо меню підходить користувачу він його затверджує після чого воно зберігається до БД.

Наступним кроком, є погодження меню працівниками МОН. Користувач відповідної ролі, переглядає меню, фінальні показники, та відповідність нормам харчування, і може його затвердити, після чого воно вступить в силу, або відмовити і додати причину відмови.

Якщо меню не погоджене, інший користувач, повинен внести правки до меню відповідно причині відмови. Після чого знову відправити на затвердження.

Ця ітерація буде повторюватись до тих пір поки працівник МОН не затвердить меню.

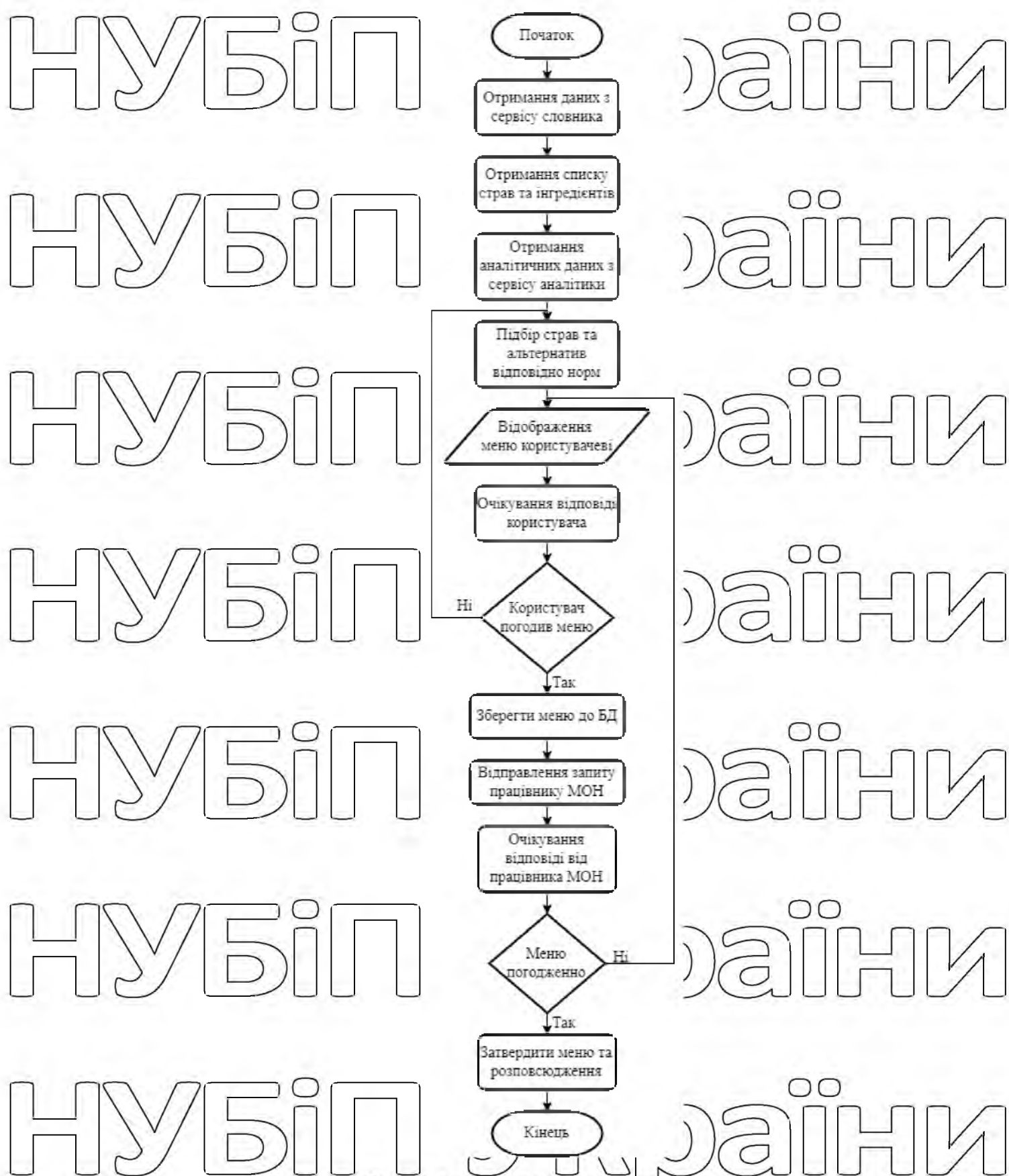


Рис. 8 Алгоритм формування меню

3.7 Дані для формування звітності

Для СІПР є важливою складовою наявність аналітичної складової, яка буде корегувати рішення. Не виключенням є СІПР з управління шкільним харчуванням.

Наявність аналітичної складової також дозволяє реалізувати контроль за якістю і системою звітності.

Для формування звітності часткові дані уже є у системі, це міста, райони, області. Частину даних вносять менеджери і адміністратори, це заклади освіти.

Також в системі мають бути норми харчування, які завантажуються при старті системи і потім можуть бути змінені.

Кількісні показники для звітів отримуються під час роботи системи з сервісу обробки даних. І на базі отриманих даних можна сформувати звіт.

Звіт формується за весь час або період, по всіх даних або в розрізі учбового закладу, міста, району або області.

Сформований звіт відправляється з системи у вигляді JSON для подальшої візуалізації на стороні клієнта.

3.8 Розгортання системи

Для відображення архітектури розгортання програмної системи використовується діаграма розгортання. Діаграма розгортання показує обчислюванні вузли під час роботи програми та компоненти, об'єкти та артефакти, що розміщуються на цих вузлах.

Вузли представляються, як прямокутні паралелепипеди з артефактами, розташованими в них, зображеними у вигляді прямокутників. Вузли можуть мати підвузли, які представляються, як вкладені прямокутні паралелепипеди. Один вузол діаграми розгортання може концептуально представляти безліч фізичних вузлів, таких, як кластер серверів баз даних.

НУБІП України

Існує два типи вузлів:

- Вузол пристрою;
- Вузол середовища виконання.

Вузли пристроїв – це фізичні обчислювальні ресурси зі своєю пам'яттю і

сервісами для виконання програмного забезпечення, такі як звичайні ПК, мобільні телефони.

Вузол середовища виконання – це програмний обчислювальний ресурс, який працює всередині зовнішнього вузла і який надає собою сервіс, який виконує інші виконувані програмні елементи.

Діаграма розгортання представлена на рис. 9. На діаграмі зображені вузли.

Client – пристрій клієнта з якого він отримує доступ до системи.

Revers proxy – вузол з NGINX та конфігураційним файлом, який налаштований, як зворотній прокеї.

SM-API network – приватна мережа системи у якій виконуються внутрішні взаємодії сервісів

Auth – вузол, на якому розміщується сервіс авторизації.

Profile – вузол, на якому розміщується сервіс профайлів.

Dictionary – вузол, на якому розміщується сервіс словник.

Feedback – вузол, на якому розміщується сервіс зворотнього зв'язку.

Order – вузол, на якому розміщується сервіс замовлень.

Message broker – вузол, на якому розміщується Apache Kafka і Zookeeper.

Processing – вузол, на якому розміщується сервіс обробки даних.

Analytic – вузол, на якому розміщується сервіс аналітики.

Food – вузол, на якому розміщується сервіс страв.

Menu – вузол, на якому розміщується сервіс меню.

RDBMS – вузол, на якому розміщується система управління реляційними базами даних PostgreSQL.

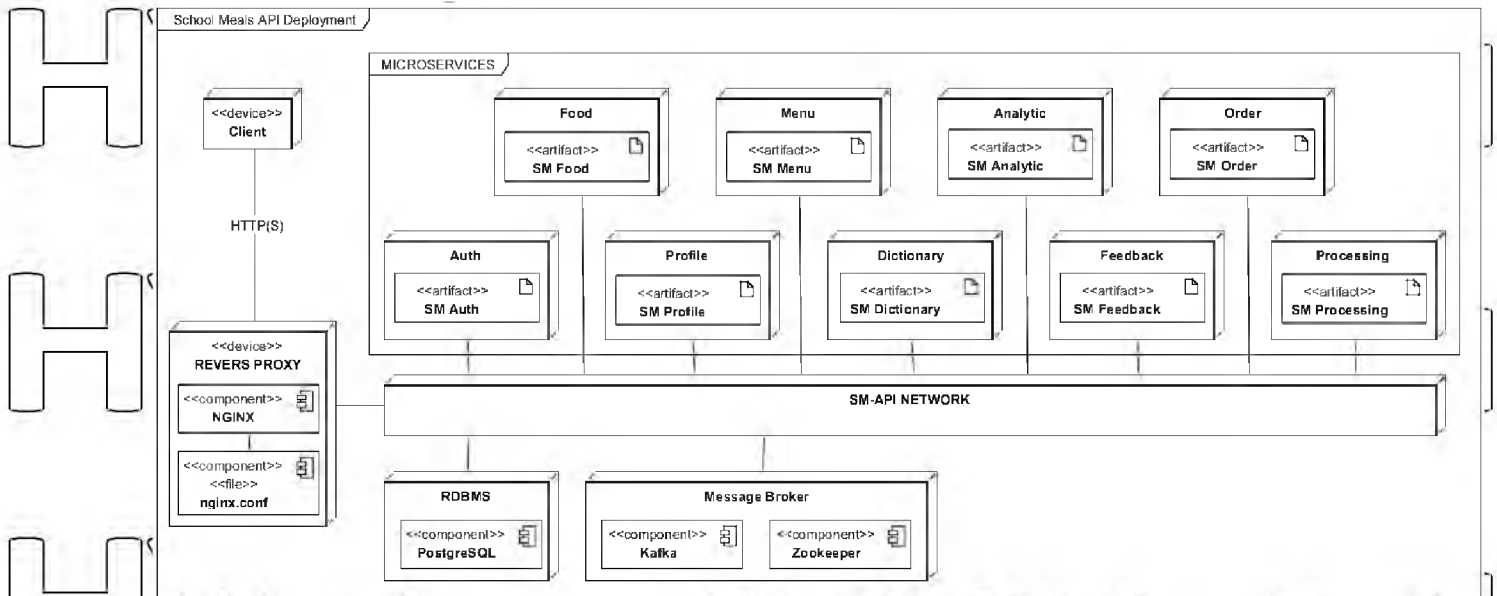


Рис. 9 Діаграма розгортання

Для повноцінної роботи системи, а також зручного бекапу даних рекомендується встановити засіб для віртуалізації Docker. Для даної програми рекомендується використання дистрибутивів ОС Linux: Debian, Ubuntu або CentOS, оскільки з ними докер працює найкраще.

Docker - це засіб який дозволяє запускати процеси в ізольованих контейнерах, при цьому використовуючи ресурси основної системи. Це дає нам можливість розгорнути проєкт на машині і не боятись, що він принесе шкоду головній системі. Docker дозволяє створювати спільні каталоги, а також перенаправляти порти. Це дасть нам можливість отримати працювати з докером зовні, а також робити бекап простим копіюванням одного каталогу, що є доволі зручно.

При використанні Docker нам не потрібно встановлювати на основну систему СУБД і решту програм, вони будуть працювати в окремих контейнерах. Такий підхід дозволить не загроможувати основну систему сміттям. І при затраті великої кількості ресурсів ми зможемо видалити контейнер однією простою командою в консолі і відновити його іншою, а завдяки спільному каталогу всі дані будуть збережені.

Для швидкого розгортання програмного продукту варто створити файл `docker-compose.yml`. Він використовується для задання параметрів одного або групи контейнерів і подальшого їх розгортання однією командою `docker-compose up`.

Використання такого способу дозволить швидко розгорнути систему та приступити до експлуатації сайту. Проте він не є обов'язковий. Можна все налаштувати самостійно, для цього потрібно встановити на комп'ютер СУБД PostgreSQL та виконати її початкові налаштування. Далі потрібно створити базу даних та схему даних.

Також потрібно встановити JVM, оскільки програма написана на мові JAVA, то віртуальні машина потрібна для запуску сервісів.

Для доступу до API через мережу Internet потрібно встановити NGINX. Це також веб-сервер, який у нашому випадку виконає функцію зворотнього проксі та балансера. Він буде отримувати запити, що надходять з мережі Internet та перенаправляти їх по приватній мережі.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

НУБІП України

4.1 Опис розробленого програмного забезпечення

В результаті дослідження та розробки було створено дев'ять мікросервісів, кожен з яких відповідає за свій набір функцій.

Перед використанням системи її потрібно розгорнути, для тестування і демонстрації чудово підійде локальна машина, а оскільки було створено `docker-compose.yml`, то розгортання відбувається однією командою `docker-compose up`.

Для перевірки стану сервісів, використовуємо утиліту `Docker Desktop for Windows`. Після розгортання стан сервісів буде виглядати, так, як на рисунку 10.

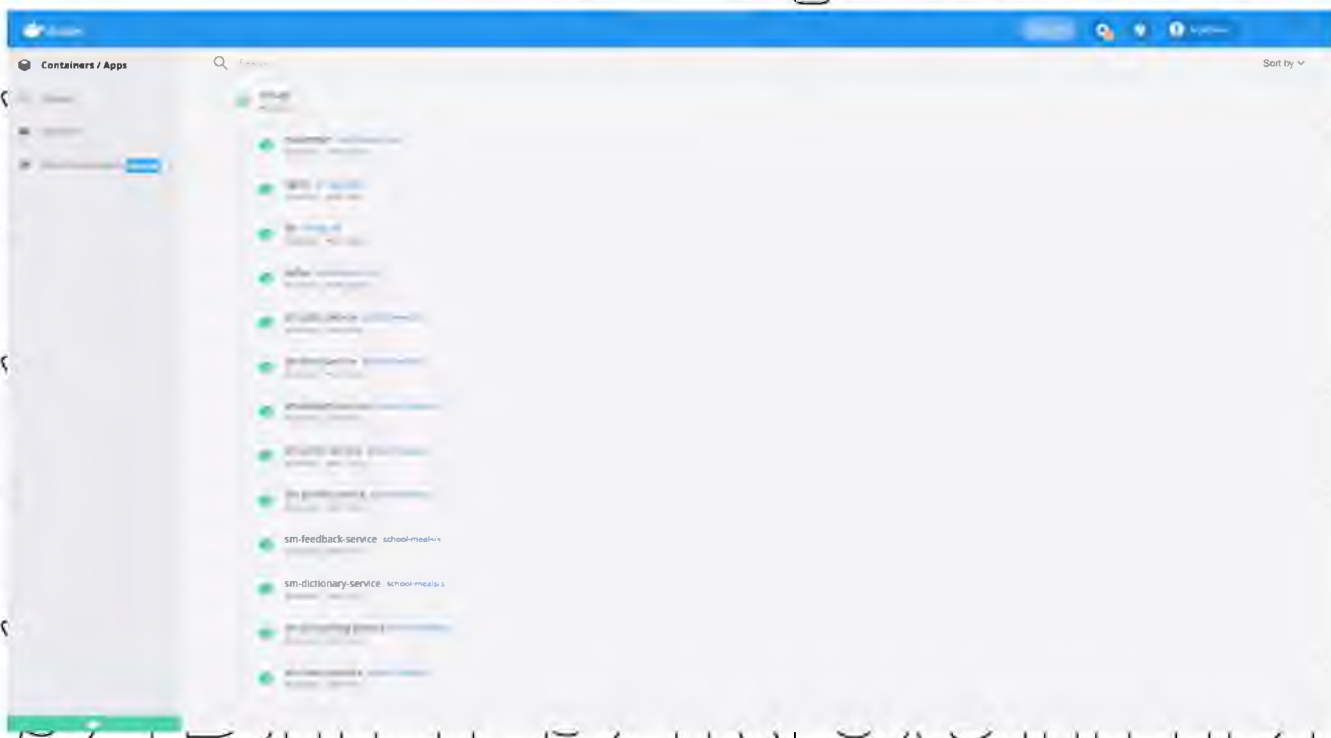


Рис. 10 Стан сервісів системи

Зелений куб перед назвою кожного з сервісів, означає що сервіс запущений, та працює. Якщо куб буде червоним, це означає, що у сервісі сталась критична помилка і він перестав працювати. А якщо, куб сірий, то сервіс був зупинений користувачем.

Якщо система розгортається з допомогою docker-compose, то всі сервіси групуються, у нашому випадку група називається sm-api. Група може перебувати у чотирьох станах, про що буде повідомляти колір іконки перед її назвою:

- Зелений – всі сервіси запущені, та працюють.
- Жовтий – щонайменше один сервіс зупинений, або в ньому сталась критична помилка і він перестав працювати.
- Червоний – у всіх сервісах сталась помилка і вони перестали працювати.
- Сірий – всі сервіси групи були зупинені користувачем.

Розглянемо роботу основних сервісів системи підтримки прийняття рішень з управління шкільним харчуванням. Для демонстрації роботи використаємо утиліту Postman, яка є інструментом для тестування REST API. Її основним завданням є

відправлення запитів з вказаними параметрами на ендпоінти системи. Також містить функціонал для створення змінних та написання автоматичних тестів і сценаріїв тестування.

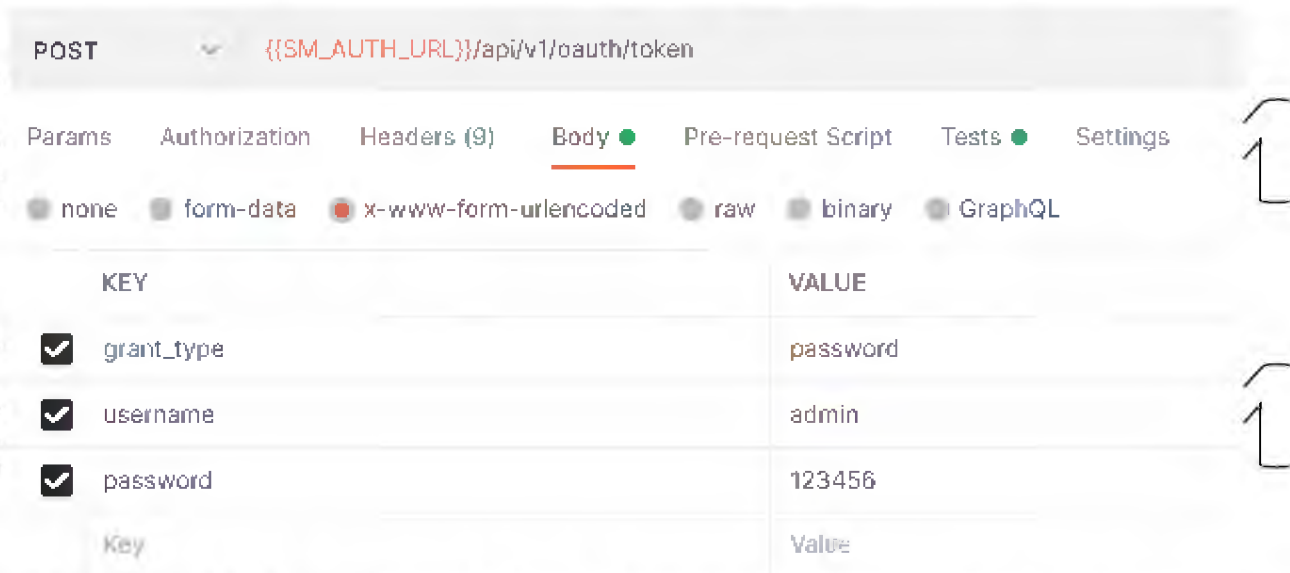
Сервіс авторизації. Він забезпечує безпеку даних та контроль доступу. Надає засоби авторизації і реєстрації користувачів у системі, та забезпечує доступ за ролями. Найчастіше використовуваною функцією буде авторизація у системі.

Сервіс підтримує три типи авторизації, які можуть бути розширені за вимогою, це:

- password – авторизація за логіном або електронною поштою і паролем користувача.
- refresh token – авторизація за токеном, використовується для оновлення токена доступу без повторного вводу паролю.
- client_credentials – авторизація за назвою клієнта і його паролем, використовується для авторизації сервісів у системі.

Для отримання доступу у систему буде використовуватись тип password. Для авторизації запит повинен містити заголовок Authorization у якому міститься ім'я і пароль клієнта, з якого хоче авторизуватись користувач, зашифрований кодуванням

Base64. Тіло запиту має бути типу `x-www-form-urlencoded` або `form-data`, які повинні містити три обов'язкові значення: `grant_type`, `username`, `password`. Тіло запиту на авторизацію зображено на рис. 11, а відповідь від сервера – на рис. 12.



POST `{{SM_AUTH_URL}}/api/v1/oauth/token`

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings

none form-data **x-www-form-urlencoded** raw binary GraphQL

KEY	VALUE
<input checked="" type="checkbox"/> grant_type	password
<input checked="" type="checkbox"/> username	admin
<input checked="" type="checkbox"/> password	123456
Key	Value

Рис. 11 Тіло запиту на авторизацію користувача

```

1
2
"access_token": "eyJhbGciOiJIUzU1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiJ1b3R5b2ZpbG1vbmFyeSIsInNtLXByb2ZpbGU1LCJzbS1mb29kIiwic29tbnVudSI6ImNtLWVudF9pZC16IndlYi1hcHAifQ.",
3
4
"token_type": "bearer",
5
"refresh_token": "eyJhbGciOiJIUzU1NiIsInR5cCI6IkpXVCJ9.eyJhdWQiOiJ1b3R5b2ZpbG1vbmFyeSIsInNtLXByb2ZpbGU1LCJzbS1mb29kIiwic29tbnVudSI6ImNtLWVudF9pZC16IndlYi1hcHAifQ.",
6
7
"expires_in": 7199,
"scope": "READ WRITE"

```

Рис. 12 Відповідь сервера на успішну авторизацію користувача

Відповідь від сервера містить два JSON Web Token, які містять інформацію про користувача та авторизацію, тип токенів, термін дії в секундах для токена доступу.

Наступним оглянемо сервіс меню, а саме метод створення меню. Для генерації меню достатньо відправити запит на відповідний ендпоінт користувачу, який має на це право. Запит і відповідь від сервера зображено на рис. 14.

SM-API / SM Menu / GENERATE MENU

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** {{SM_MENU_URL}}/api/v1/menu/generate
- Response Body (JSON):**

```

1  {
2    "tuesday": [
3      {
4        "unit": "g",
5        "name": "Салат або овочі соєві",
6        "weight": "80",
7        "calories": 38
8      },
9      {
10       "unit": "g",
11       "name": "Сіне курки в сметанному соусі",
12       "weight": "38/38",
13       "calories": 186
14     },
15     {
16       "unit": "g",
17       "name": "Хачапурі",
18       "weight": "158",
19       "calories": 288
20     },
21     {
22       "unit": "g",
23       "name": "Чаш",
24       "weight": "288",
25       "calories": 68
26     },
27     {
28       "unit": "g",
29       "name": "Хніб",
30       "weight": "26",
31       "calories": 62
32     }
33   ],
34   "wednesday": [
35     {
36       "unit": "g",
37       "name": "Хачапурі",
38       "weight": "158",
39       "calories": 288
40     }
41   ]
42 }

```

Рис. 14 Запит та відповідь від сервера на генерування меню

Сервер повертає отримане меню, яке користувач зможе змінити за необхідності. І останнім, розглянемо сервіс аналітики. У цьому сервісі можна отримати дані для формування звітності з використанням фільтрів.

Розглянемо запит на стримання аналітики за 1 день по всіх регіонах. Для цього потрібно відправити запит на сервер з вказанням дати, за яку потрібні дані. На рис. 15 зображено запит та відповідь від сервера, на отримання аналітичних даних за один день.

The screenshot displays a REST client interface for a GET request. The request URL is `{ISM_ANALYTIC_URL}/api/v1/analytic/general/day/2021-11-29`. The response is shown in JSON format, containing an array of three objects representing different food items.

```

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
31
{
  "threeStars": 2,
  "name": "Салат або овочи свіжі",
  "rating": 4.5813953488372894,
  "orders": 11345,
  "twoStars": 0,
  "oneStars": 1,
  "fourStars": 10,
  "fiveStars": 30
},
{
  "threeStars": 0,
  "name": "Плов (з рисом і свининою)",
  "rating": 4.798701298701299,
  "orders": 16213,
  "twoStars": 0,
  "oneStars": 0,
  "fourStars": 31,
  "fiveStars": 123
},
{
  "threeStars": 0,
  "name": "Компот з сушених фруктів",
  "rating": 4.978260869565218,
  "orders": 20331,
  "twoStars": 0,
  "oneStars": 0,
  "fourStars": 1,
  "fiveStars": 45
}

```

Рис. 15 Запит та відповідь від сервера на отримання аналітичних даних

Відповідь від сервера приходять у форматі JSON, який підтримує більшість додатків, та є зручним для візуалізації у різних виглядах, як таблицю, так і графіком. Прикладні візуалізації у вигляді графіків зображено на рис. 16-18. Для візуалізації використовувався сервіс <https://jsontochart.com>.

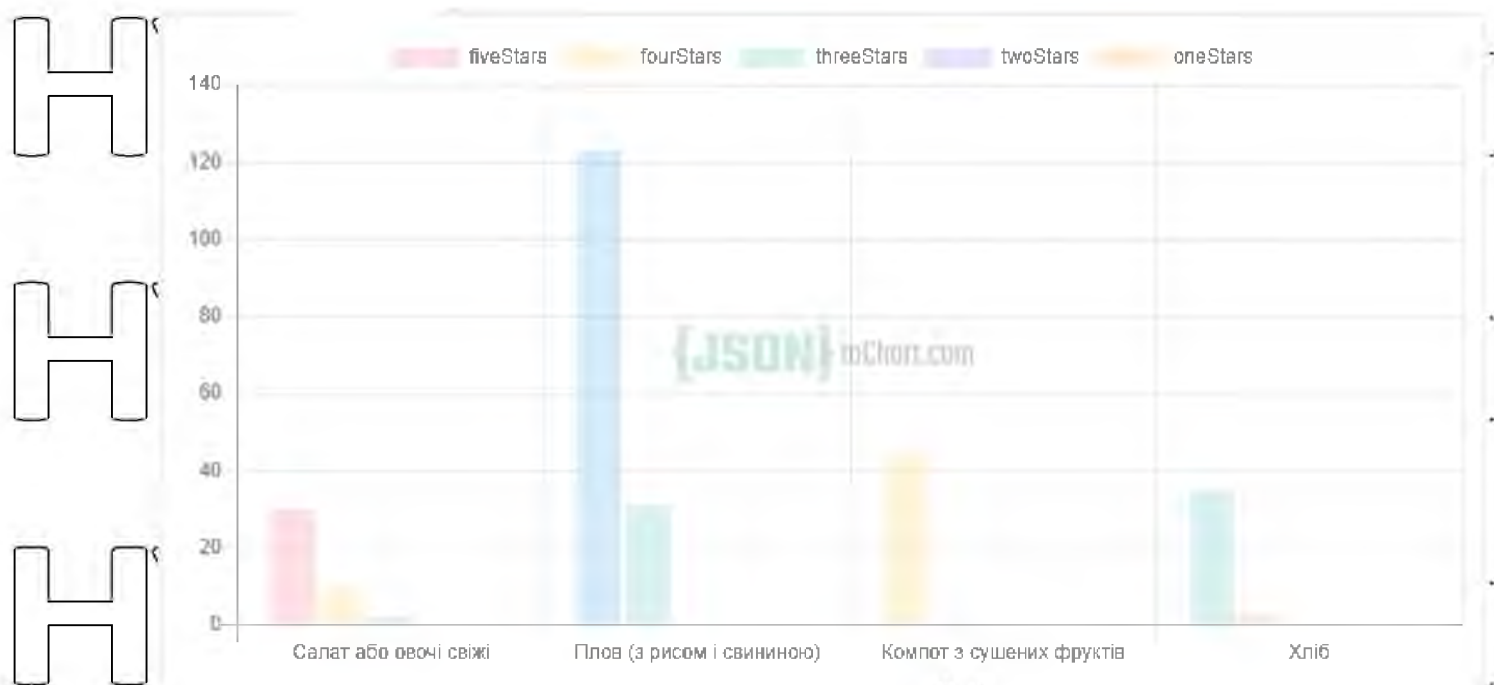


Рис. 16 Оцінки в розрізі страв за один день



Рис. 17 Рейтинг страв

НУБІП України

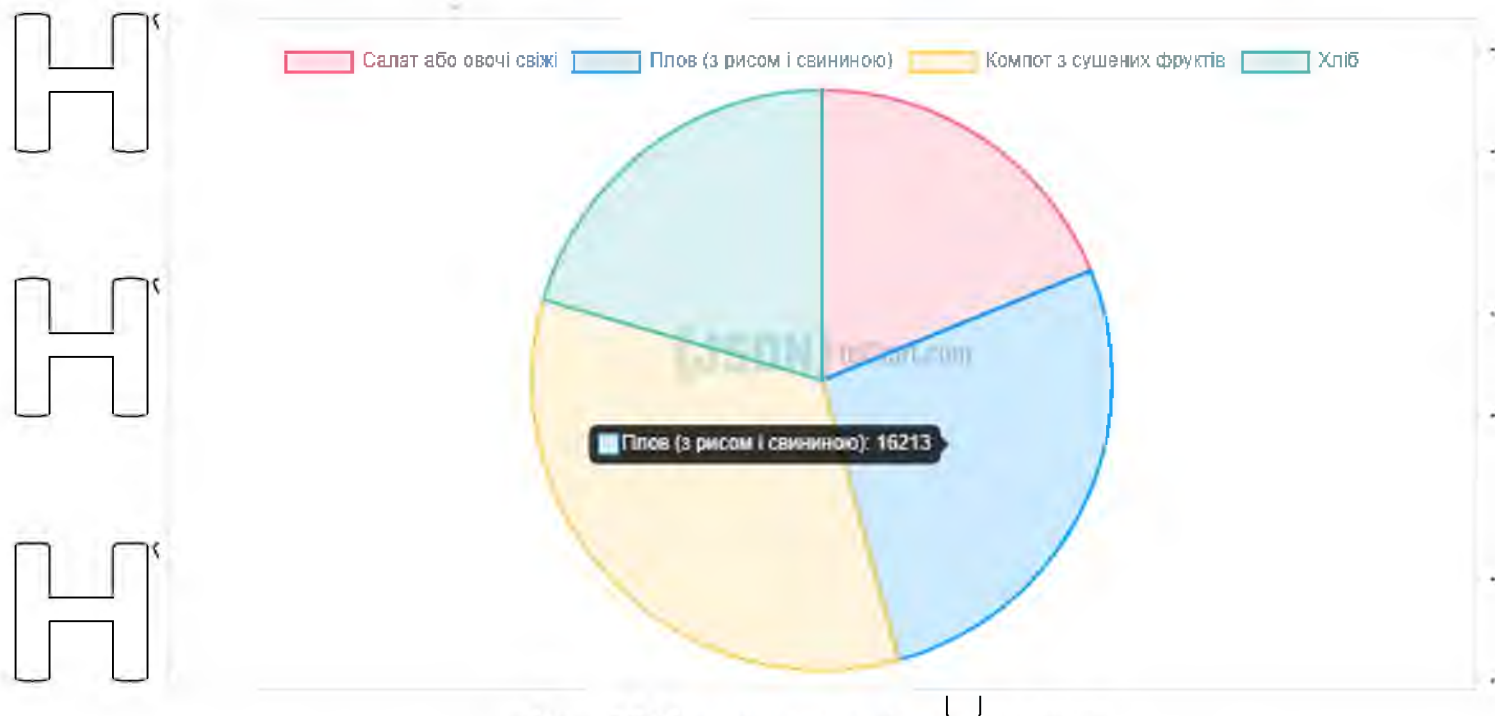


Рис. 18 Кількість реалізованих страв

4.2 Апаратні та програмні вимоги до системи

Мінімальними вимогами є:

ОЗУ: не менше 24 ГБ

Процесор: 6 логічних ядер частотою не менше 1.5 ГГц

Пам'ять пристрою: 2ТБ + пам'ять для БД і файлів.

OS: Windows 8, 8.1, 10, Unix (Debian, Fedora, CentOS, Ubuntu).

А також наявність встановленого Docker, JVM 11

ВИСНОВКИ

В результаті виконання магістерської роботи було проаналізовано предметну область у сфері управління шкільним харчуванням, формування раціону шкільного меню. Було виділено вимоги та написано технічне завдання для системи, що реалізовувалась.

Після аналізу предметної області було спроектовано, систему, проектування виконувалось засобами мови UML. На етапі проектування було побудовано діаграми, що відображають предметну область: прецедентів, активності, послідовностей. І саму систему: архітектури, пакетів. Також було розроблено діаграму розгортання, для відображення розміщення сервісів на хмарі або в контейнерах.

Після проектування розроблювалась сама система. Основними засобами для реалізації були мова програмування Java та Spring Framework. Для зберігання даних в момент розробки використовувалась СУБД PostgreSQL, також була протестована і сумісна з СУБД MS SQL Server без додаткових змін у коді.

Було складено для кожного з сервісів Dockerfile для створення образу і подальшого розгортання в контейнері, а також написано скрипт для docker-compose, який дозволяє повністю розгорнути систему в локальному середовищі, лише однією командою.

В результаті було розроблено:

- 9 мікросервісів;
- алгоритм обробки даних та формування шкільного меню;
- налаштовано взаємодію з Apache Kafka;
- налаштовано NGINX для роботи в якості зворотнього проксі;
- розроблено набір міграцій для БД, що не прив'язані до СУБД і є сумісні, як з PostgreSQL, так і з MS SQL Server.

Отже, в результаті досліджень було виділено залежності, та реалізовані в системі:

- Якість шкільного харчування напряду впливає на якість освітнього процесу.

Учні демонструють більшу зацікавленість і увагу до предметів, перебуваючи

в хорошому настрої, та при здоровому харчуванні.

- Зворотній зв'язок збільшує довіру батьків та учнів до шкільного харчування та освітньої політики.

- Корекція шкільного меню до регіональних смаків учнів, підвищив оцінку

якості шкільного харчування.

- Наявність звітів у системі та аналітики дозволяє, відкрито контролювати якість шкільного харчування та виконувати певні дії до закладів у яких оцінка

якості низька.

- Відкритість системи, має позитивний вплив на рівень довіри до освітньої системи.

- Виконання дій по формуванню меню від заявки до готового меню значно

пришвидшується завдяки швидкому обміну інформацією, та набором

довідкових даних по нормах харчування.

НУБІП України

НУБІП України

НУБІП України

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Про затвердження норм та Порядку організації харчування у закладах освіти та дитячих закладах оздоровлення та відпочинку [Електронний ресурс].

URL: <https://zakon.rada.gov.ua/rada/show/z05-2021-%D0%BF#Text>

2. Про затвердження Норм фізіологічних потреб населення України в основних харчових речовинах і енергії [Електронний ресурс].

URL: <https://zakon.rada.gov.ua/rada/show/z1206-17#n14>

3. Принципи і методи системного аналізу [Електронний ресурс].

URL: https://stud.com.ua/45001/investuvannya/printsipti_metodi_sistemnogo_analizu

4. Вигере К.И. Разработка требований до ПЗ. М.: Російська редакція Microsoft, 2004. – 575 с.

5. Фаулер Скотт До. UML в короткому викладі. Застосування стандартної мови об'єктного моделювання: Пер. з англ. – М.: Мир, 1999. – 191 с.

6. Лешек А. Мацяшек Анализ требований и проектирование систем. Разработка информационных систем с использованием UML.: Пер. с англ. – М.: Издательский дом «Вильямс». – 2002. – 432 с.

7. Ларман К. Применение UML и шаблонов проектирования / Крег Ларман. – Москва: Издательский дом "Вильямс", 2004. – 624 с.

8. Якобсон А. Унифицированный процесс разработки программного обеспечения / А. Якобсон, Г. Буч, Д. Рамбо., 2002. – 496 с.

9. Тема 8. Унифицированный язык моделирования UML [Електронний ресурс]. URL: <https://sanara.mgrip.ru/~dzhadzha/dis/15/200.htm>

10. Діаграма діяльності [Електронний ресурс]. URL: <http://5fan.ru/wievjob.php?id=21291>.

НУБІП України

11. Проектування програмного забезпечення засобами ОМД [Електронний ресурс]. URL: http://mmsa.kpi.ua/sites/default/files/disciplines/didkovska_mv_testing_lecture_3.pdf

12. Моделі і методи проектування інформаційних систем [Електронний ресурс]. URL: https://elcarning.su.mdu.edu.ua/free_content/lectured/def1c9452f2a161439391120eef364dd8ce4d8c5c/20160217112601/170352/index.html

13. Діаграма розгортання [Електронний ресурс]. URL: <https://studfiles.net/preview/5010027/page:6/>

14. Методи створення оптимальної моделі баз даних [Електронний ресурс]. URL: <https://studfile.net/preview/7144845/page:28/>

15. Аналіз даних і результати дослідження [Електронний ресурс]. URL: <https://sites.google.com/site/marketingdistance/tema-2/2-6-analiz-danih-i-rezultati-doslidzenna>

16. What is Data Analysis? Research | Types | Methods | Techniques [Електронний ресурс]. URL: <https://www.guru99.com/what-is-data-analysis.html>

17. Що таке Big Data? [Електронний ресурс]. URL: <https://futurenow.com.ua/shlo-to-take-big-data-velyk-japi/>

18. Data Mining [Електронний ресурс]. URL: <https://www.investopedia.com/terms/d/datamining.asp>

19. Роберт С. Алгоритми на Java / С. Роберт, У. Кевін., 2013. – 848 с.

20. Уоллс К. Spring в действии / Крейг Уоллс., 2013. – 752 с.

21. Хорстман К. С. Java. Библиотека профессионала, том 1. Основы / Кей С. Хорстман. – Київ: Діалектрика, 2020. – 864 с.

22. Крокфорд Д. JavaScript сильные стороны / Дуглас Крокфорд. – Москва: Питер, 2012. – 176 с.