

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій
Кафедра інтелектуальних технологій

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему:

«Програмний модуль підсистеми підтримки рішень для
малого сільського господарства»

Галузь знань **12 «Інформаційні технології»**

Спеціальність **122 «Комп'ютерні науки»**

Освітня програма **«Комп'ютерні науки»**

Освітній рівень: бакалавр

Виконав: студент 4 курсу, групи КН – 42

Мухін Владислав Євгенійович
(прізвище та ініціали)



Керівник Іларіонов О.Є.
(прізвище та ініціали)

Кандидат технічних наук, доцент
(науковий ступінь, звання)

Випускна кваліфікаційна робота бакалавра допущена до захисту
рішенням кафедри *інтелектуальних технологій*
Протокол № 11 від 06.06.2022р.
зав. кафедри _____ доц. Іларіонов О.Є.

Київ - 2022

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА


Факультет інформаційних технологій
Кафедра інтелектуальних технологій
Спеціальність 122 «Комп'ютерні науки»

ЗАТВЕРДЖУЮ
Завідувач _____ кафедри
інтелектуальних технологій
Іларіонов О.Є.

“ _____ ” _____ 2022 р.

ЗАВДАННЯ
НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ


Мухіну Владиславу Євгенійовичу

1. Тема проекту (роботи) . Програмний модуль підсистеми підтримки рішень для малого сільського господарства затверджена протоколом засідання кафедри від «23 » грудня 2021 р. № 4
2. Термін здачі студентом закінченого проекту (роботи) 29 травня 2022 року
3. Вихідні дані до проекту (роботи)
Системні рекомендації щодо прийняття рішень, с/г рекомендації з тваринництва, дані про користувачів
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
Аналіз процесу надання оптимального рішення закупівлі комбікорму
Розробка архітектури програмного модуля підтримки прийняття рішень з питань кормовиробництва фермерського господарства
Програмне забезпечення програмного модуля підтримки прийняття рішень з питань кормовиробництва фермерського господарства
6. Перелік презентаційного матеріалу (з точним зазначенням обов'язкових презентацій) Об'єкт, предмет, мета та завдання дослідження (1 слайд), порівняльний аналіз існуючих рішень (3 слайди), проектування інформаційної системи (2 слайди), інформаційне забезпечення (1 слайд), структура програмного забезпечення (1 слайд), огляд процесу _____ тестування (1 слайд), _____ висновок (1 слайд)
7. Дата видачі завдання 15 лютого 2022 року
Керівник _____ / Іларіонов О.Є.
Завдання прийняв до виконання  / Мухін В.Є. /

КАЛЕНДАРНИЙ ПЛАН

По р. №	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів випускної кваліфікаційної роботи	Примітка
1.	Аналіз літературних джерел, аналіз існуючих методів, аналіз основних процесів предметного середовища, постановка задачі	20.02.2022 – 07.03.2022	Виконано
2.	Проектування web-додатку для програмного модуля підсистеми підтримки рішень для малого сільського господарства	10.03.2022 – 10.04.2022	Виконано
3.	Розробка та тестування кроссбраузерного додатку програмного модуля підсистеми підтримки рішень для малого сільського господарства	12.04.2022 – 02.05.2022	Виконано
4.	Оформлення пояснювальної записки, підготовка презентації	05.05.2022 – 15.05.2022	Виконано

Студент-дипломник

 / Іхін В.Є. /

Керівник випускної кваліфікаційної роботи _____ / Іларіонов О.Є. /

ЗМІСТ

ВСТУП	5
РОЗДІЛ 1. АНАЛІЗ ПРОЦЕСУ НАДАННЯ ОПТИМАЛЬНОГО РІШЕННЯ ЗАКУПІВЛІ КОМБІКОРМУ	6
1.1 Аналітичний огляд СППР для фермерського господарства	6
1.2 Функціональність СДПР та загальна архітектура	7
1.3 Аналіз існуючих інформаційних систем допомоги прийняття рішень з питань кормовиробництва для фермерського господарства	8
РОЗДІЛ 2. РОЗРОБКА АРХІТЕКТУРИ ІНФОРМАЦІЙНОЇ СДПР З ПИТАНЬ КОРМОВИРОБНИЦТВА ДЛЯ ФЕРМЕРСЬКОГО ГОСПОДАРСТВА	16
2.1 Розробка клієнт-серверної архітектури	16
2.2 Проектування архітектури програмного модуля	17
2.2.1 Архітектура СДПР	17
2.2.2 Розробка алгоритму програмного модуля	20
2.2.3 Функціональний аналіз	21
2.2.4 IDEF0 процесу видачі списку необхідного комбікорму завдяки програмному модулю підсистеми підтримки рішень для малого сільського господарства	25
2.2.5 Архітектура програмного модуля підсистеми підтримки рішень для малого сільського господарства	27
2.3 Інформаційне забезпечення програмного модуля підсистеми підтримки рішень для малого сільського господарства	28
2.3.1 Розробка інформаційного забезпечення	28
2.4 Висновки до другого розділу	31
	3

РОЗДІЛ 3. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ПРОГРАМНОГО МОДУЛЯ ПІДСИСТЕМИ ПІДТРИМКИ РІШЕНЬ ДЛЯ МАЛОГО СІЛЬСЬКОГО ГОСОПДАРСТВА	32
3.1 Обґрунтування вибору програмних засобів	32
3.3 Структура програмного забезпечення	33
3.4 Керівництво користувача	34
3.5 Результат тестування	43
3.6 Висновки до третього розділу	44
Висновок	45
Список використаних джерел	46
Додаток А	48

ВСТУП

Сучасний світ все більше віддає перевагу комп'ютеру та його обчислювальними здібностями для перегляду та виявлення оптимальних рішень для отримання особистого результату.

Однією з таких галузей є фермерське господарство, в якому є безліч моментів які необхідно передати комп'ютеру, розрахунок зерна для посіву, кількість закупівлі харчових добавків, підрахунок та виявлення оптимального рішення для закупівлі сухого корму для тварин і безліч інших підрахунків.

Об'єктом дослідження даної роботи є процеси кормовиробництва для фермерського господарства.

Предметом дослідження роботи є програмний модуль підсистеми підтримки рішень для малого сільського господарства

Мета даного дипломного проекту полягає в створенні алгоритму модуля підсистеми підтримки рішень для малого сільського господарства.

Для досягнення даної мети потрібно виконати наступні завдання:

- провести аналіз існуючих технологічних рішень закупівлі корму
- сформуванати алгоритм перевірки наявності кормів
- сформуванати функціональні та нефункціональні вимоги для розроблюваного WEB-застосунку
- розглянути процеси даної предметної області
- розробити архітектуру рекомендаційної системи з питань кормовиробництва для фермерського господарства
- розробити та протестувати програмний продукт

РОЗДІЛ 1. АНАЛІЗ ПРОЦЕСУ НАДАННЯ ОПТИМАЛЬНОГО РІШЕННЯ ЗАКУПІВЛІ КОМБІКОРМУ

1.1 Аналітичний огляд СППР для фермерського господарства

Система підтримки прийняття рішень — комп'ютеризована система, яка через збирання та аналіз великого обсягу інформації може впливати на процес ухвалення керівничих рішень в бізнесі та підприємстві [2].

Фермери — категорія підприємців у сільському господарстві.

Фермери це власники чи орендарі земельних угідь, які ведуть своє господарство сім'єю та/або з використанням найманої робочої сили. Здійснюють виробництво, переробку і реалізацію сільськогосподарської продукції на основі використання майна, що знаходяться у них на праві оренди, довічного успадкованого володіння або у власності[3].

Сучасні системи підтримки рішень виникли у результаті злиття керівничих інформаційних систем і систем керування базами даних, як системи, що максимально пристосовані до розв'язання задач щоденної керівницької діяльності, і є інструментом допомоги тим, хто вирішує (робить вибір). За допомогою СППР може проводитись вибір рішень у певних неструктурованих і слабо структурованих задачах, у тому числі й тих, що мають багато критеріїв[4].

Однією з галузей, де активно впроваджуються інноваційні технології — це фермерське господарство. В цій сфері важливим компонентом успіху відгодовування тварин є якісний вироблений по всім стандартам комбінований корм. Комбікорм — це однорідна суміш очищених, змелених і спеціально підготовлених кормів і кормових джерел. Комбікорм виготовляють за науково обґрунтованими рецептами, які призначені для годуювання окремим видам, виробничим і статевим групам тварин [1].

Інформаційна система допомоги прийняття рішень з питань кормовиробництва для фермерського господарства допоможе фермерам

приймати рішення для допомоги закупівлі необхідної кількості комбікормів для майбутніх тварин від маленького віку дорослої особи для кожного виду тварини.

Даний програмний модуль підсистеми підтримки рішень для малого сільського господарства допоможе виявляти оптимальні рішення для закупівлі чіткої кількості корму для тварин та виявити оптимальні рішення для закупівлі у різних постачальників в залежності від ціни та можливості предоставити необхідну кількість корму однією партією.

1.2 Функціональність СДПР та загальна архітектура

Основна функціональність СДПР випливає з її визначення. Головною функцією є виведення рекомендацій виходячи з вивчення історичного та поточного стану об'єкта дослідження та порівняння їх з інформацією, що зберігається в основі знань системи. Однак це ще не все.

Необхідно завжди пам'ятати, що знання у будь-якій проблемній галузі завжди є динамічними. Щось старіє, якісь гіпотези спростовуються, підтверджуються нові теорії, дослідники знаходять нові закономірності та факти. Все це має поступово вноситися до бази знань СДПР, щоб вона була актуальною. Без цього система заковстане і перестане відповідати викликам середовища, що змінюється. Ну, як це зазвичай буває. Іншими словами, система має надавати функцію з актуалізації знань.

У процесі актуалізації знань беруть участь дві особи з ролями «Експерт» та «Інженер зі знань» (когнітолог). Перший надає знання найчастіше у неструктурованому вигляді, а другий переносить в базу знань СДПР у формалізованому і повністю структурованому вигляді й у форматі, що використовується у самій системі. Після цього експерт верифікує знання вже в базі знань, тим самим своїм авторитетом підтверджує те, що система може використовуватися для підтримки прийняття рішень, і рекомендації, що видаються нею, засновані на правильних методах виведення і коректних знаннях.

Нарешті, оскільки системою користуватимуться фахівці різних рівнів майстерності, сама система повинна мати механізм пояснення тих рекомендацій, які вона видає. Це дуже важлива функція навіть для процесу верифікації знань.

Таким чином, перелік основних функцій СДПР виглядає так:

- Вилучення знань;
- Верифікація знань;
- Висновок рекомендацій;
- Пояснення рекомендацій.

Все це дозволяє намалювати саму узагальнену функціональну архітектуру СППР у такому вигляді:

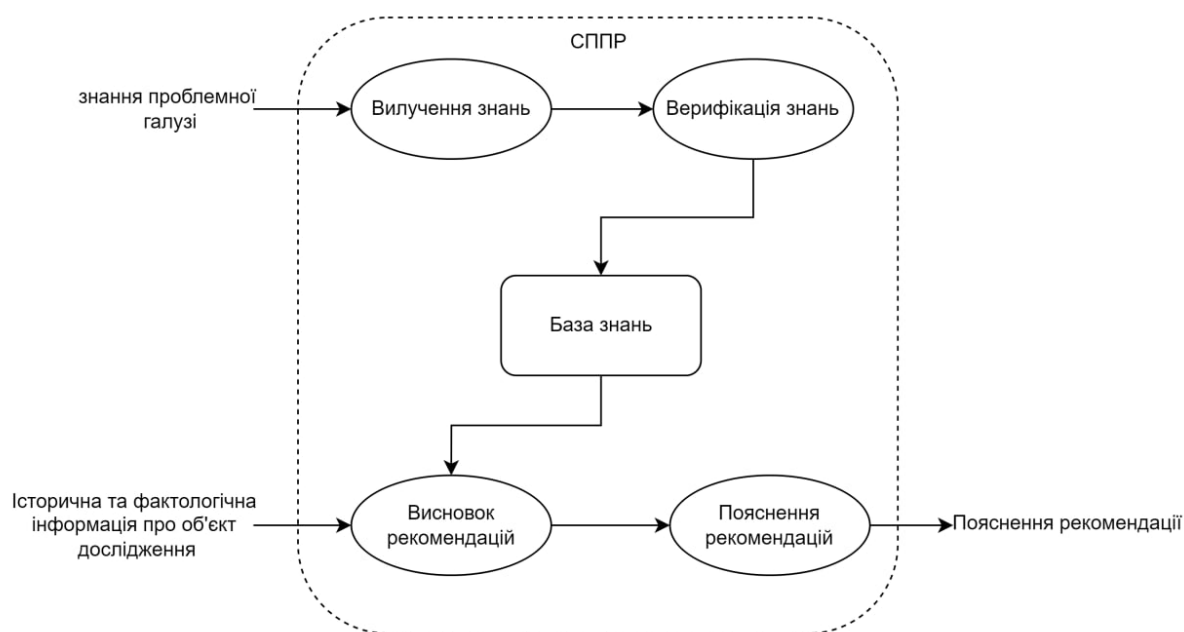


Рисунок 1 – Загальна СДПР та функціонал

1.3 Аналіз існуючих інформаційних систем допомоги прийняття рішень з питань кормовиробництва для фермерського господарства

На даний момент майже не існує можливих альтернатив систем для прийняття рішень для заготівлі кормів для майбутніх тварин.

Найпопулярніші ресурси для закупівлі комбікорму існують онлайн маркети для продажу кормів для різних видів, тому хочу виділити основні платформи для продажу комбікормів. Такі як: “Prom.ua”, “OLX” та компанія “Біла Льошка”

Найголовніше, що не вистачає всім платвормам на ринку це можливість виявити постачальника який має можливість відправити однією партією кількість корму та взагалі виявити кількість можливого комбі корму та виявити оптимальне рішення для закупівлі в окремих постачальниках для вигіднішої закупівлі.

1. Prom.ua — український маркетплейс, проєкт ІТ-компанії EVO. На його платформі підприємці самостійно створюють інтернет-магазини або розміщують свої товари в загальному каталозі. Для покупців на Промі зібрано більше 100 мільйонів товарів. Згідно з Kantar TNS CMeter станом на вересень 2019 року Prom.ua входить в ТОП-10 найбільш відвідуваних сайтів UA.net.
2. OLX (*OnLine eXchange*) — платформа онлайн-оголошень, яка об'єднує людей для покупки, продажу або обміну товарами та послугами. Станом на 2018 рік на майданчику зареєстровано 1,5 млн продавців, розміщено понад 11 млн оголошень і кожну хвилину додається близько 100 нових. Оголошення класифікуються за такими категоріями, як «Дитячий світ», «Нерухомість», «Транспорт», «Запчастини для транспорту», «Робота», «Тварини», «Дім і сад», «Електроніка», «Бізнес та послуги», «Мода і стиль», «Хобі, відпочинок і спорт», «Віддам безкоштовно» і «Обмін». OLX є найбільш відвідуваним сервісом оголошень в Україні. Кожен другий інтернет-користувач з України відвідує ресурс мінімум один раз на місяць.
3. “Біла Льошка” – онлайн підприємство для закупівлі кормів майже для всіх відомих видів тварин та птиць для сільського господарства.

На основі аналізу існуючих онлайн маркетів для продажу кормів для різних видів визначимо параметри їх порівняння:

1. Список кормів – для кожної системи необхідно мати великий список та вибір всіх існуючих кормів для тварин для вибору користувачем.
2. Підтримка виявлення оптимального рішення – виявлення оптимального рішення для закупівлі необхідної кількості корму та взагалі наявності у постачальників. Відсутня можливість виявити найкращі можливості закупівлі в окремних компаніях.
3. Ціна продукту – система повинна підрахувати замість користувача, чи вистачить в нього грошей які є на застосунку чи необхідно поповнити рахунок.
4. Виявлення найдешевшого варіанту – додаток повинен виявити найдешевші і найдорожчі варіанти закупівлі товару.
5. Можливість поділитися інформацією в соц-мережах – можливість поділитись з друзями в соціальних мережах для рекомендації якогось корму.

Проведемо порівняльний аналіз існуючих рішень за наведеними вище факторами.

Таблиця 1.1. Порівняння існуючих рішень за факторами

	Prom.ua	OLX	“Біла Льошка”
Список кормів	+	+	+
Підтримка виявлення оптимального рішення	-	-	-
Ціна продукту	+	+	+
Виявлення найдешевшого варіанту	+	+	-
Можливість поділитися інформацією в соц-мережах	+	+	-

Завдяки аналізу існуючих можливих варіантів можливо виявити основні головні мінуси кожної компанії, а саме — відсутність можливості користуватись програмою підтримки, яка допоможе виявити оптимальні рішення для закупівлі необхідної кількості комбікорму для тварин, та виявити найдешевші варіанти або отримання інформації загальної кількості комбікорму в кожній компанії та виявити яка саме компанія допоможе надати однією партією всю кількість комбікорму.

1.4 Аналіз основних процесів предметного середовища.

1.4.1 Аналіз предметної області

Не найкраща ефективність виробництва сільськогосподарської продукції вітчизняних організаціях значною мірою походить через недостатню ефективністю управління процесами на виробництві. Зокрема, не повне використання генетичного потенціалу тварин та невисокі економічні показники виробництва пов'язані, в першу чергу, з нераціональністю планування відгодовування.

Вимоги до відгодовування тварин зростають відповідно до сучасних умовах економіки та потреб, і основною мірою у зв'язку з зростанням значущості факторів якості тваринницької продукції. Так, наприклад, щоб отримати сир гарної якості потрібно молоко, що відповідає суворо тим же високим стандартам якості. В свою чергу, наука та дослідження на цю тему проголошують, що необхідні властивості молока можуть бути забезпечені лише за правильним та своєчасним підбором кормів та необхідної збалансованості компонентів харчування. Тому залишається істотно важлими створення інформаційної система допомоги прийняття рішень з питань кормовиробництва для фермерського господарства. Спочатку система повинна отримувати інформацію про необхідну тварину від користувача та кількість голів тварини. Всередині системи повинно аналізувати дані для

отримання рекомендації закувілі. Ця інформація оброблюється саме системою рекомендацій комбікорму. Серед елементів керування для даного процесу зазначено акон України про “Про державний контроль кормів, здоров’я та благополуччя тварин”. Закони та інші нормативно-правові акти, якими встановлюються вимоги до кормів на будь-яких стадіях їх виробництва, обігу та використання [5].

1.5 Постановка задачі на розробку програмного модуля підтримки прийняття рішень з питань кормовиробництва фермерського господарства

Основна задача для програмного модуля підсистеми підтримки рішень для малого сільського господарства – розробка програмного застосунку та алгоритму

Функціональні вимоги для інформаційної системи допомоги прийняття рішень з питань кормовиробництва для фермерського господарства:

1. Реєстрація в системі - В кожній системі для перегляду кількості нових користувачів для звітів необхідна реєстрація, в якій вводить користувач особисту інформацію для реєстрації в системі.
2. Збереження історії перегляду необхідних критеріїв - Користувач повинен мати доступ для відновлення запитів які він робив раніше та для збереження інформації в системі для перевірки актуальних запитів користувачів.
3. Пошук та парсинг комерційних пропозицій з інших платформ - Необхідно розміщувати та сортувати як свою продукцію та корми корми так корми інших компаній для великого вибору.
4. Вивід інформації про потреби закупівлі комбікормів для фермерського господарства – один з найголовніших елементів модуля це алгоритм для перевірки наявності необхідної кількості

в нашій компанії або чи взагалі існує такий корм для вибраної тварини.

Нефункціональні вимоги описують цілі і атрибути якості. Атрибути якості (quality attributes) представляють собою додатковий опис функцій продукту, виражені через опис його характеристик, важливих для користувачів або розробників. До таких характеристик відносяться:

1. Легкість і простота використання.
2. Цілісність.
3. Ефективність і стійкість до збоїв.

До нефункціональних вимог можна віднести:

1. Система прийняття рішень повинна мати зрозумілу структуру
2. Датасети повинні бути об'ємними та містити різні приклади комбікормів для якісного аналізу
3. Зручний інтерфейс - Інтерфейс повинен бути адаптивно зрозумілим на базовому рівні, кнопка з функціоналом виглядає як кнопка з функціоналом та таблиці з зручною інформацією.
4. Зручна сторінка прейскурантів - Користувачу повинно бути зрозуміло, до якої компанії відносяться ціни та кількості кормів та їх ціну.

1.6 Висновки до першого розділу

Сучасний світ вимагає все перенести на плечі систем які допоможуть виявити оптимальне рішення та пошуку необхідного матеріалу для кожного корму. Сьогодні обов'язково необхідно допомогти фермерам для прийняття рішень закупівлі комбікорму для тварин, для полегшення пошуку в одній фірмі відразу однієї партії корму або комбінувати різними видами корму для необхідної тварини.

РОЗДІЛ 2. РОЗРОБКА АРХІТЕКТУРИ ІНФОРМАЦІЙНОЇ СДПР З ПИТАНЬ КОРМОВИРОБНИЦТВА ДЛЯ ФЕРМЕРСЬКОГО ГОСПОДАРСТВА

2.1 Розробка клієнт-серверної архітектури

Початком розробки архітектури програмного модуля підсистеми підтримки рішень для малого сільського господарства починається з розробки та виявлення основного функціонування додатку. Додаток буде реалізований через web-додаток. Розглянемо детально на діаграмі розгортання (рис. 3) для опису середовища та функціонування системи прийняття рішень через серверну частину.

Діаграми розгортання поділяється на дві частини: серверна частина та клієнтська частина.

Клієнтська частина відповідає за користування користувачем саме додатку через сервер. Відображення через веб-браузер сторінку у вигляді .html файлу з яким клієнт може функціонувати та відправляти запити на сервер для отримання від серверу текстових процесів рекомендацій та табличних процесів у вигляді таблиць преїскурантів.

Серверна частина – найголовніша частина діаграми розгортання та відповідає за основні процеси функціонування додатку. Початок серверної частини починається від запиту клієнта через веб додаток. Після отримання запиту сервер відправляє SQL запити до бази даних кормів та отримує необхідну інформацію для формування рекомендацій та відправки до клієнта необхідної інформації.

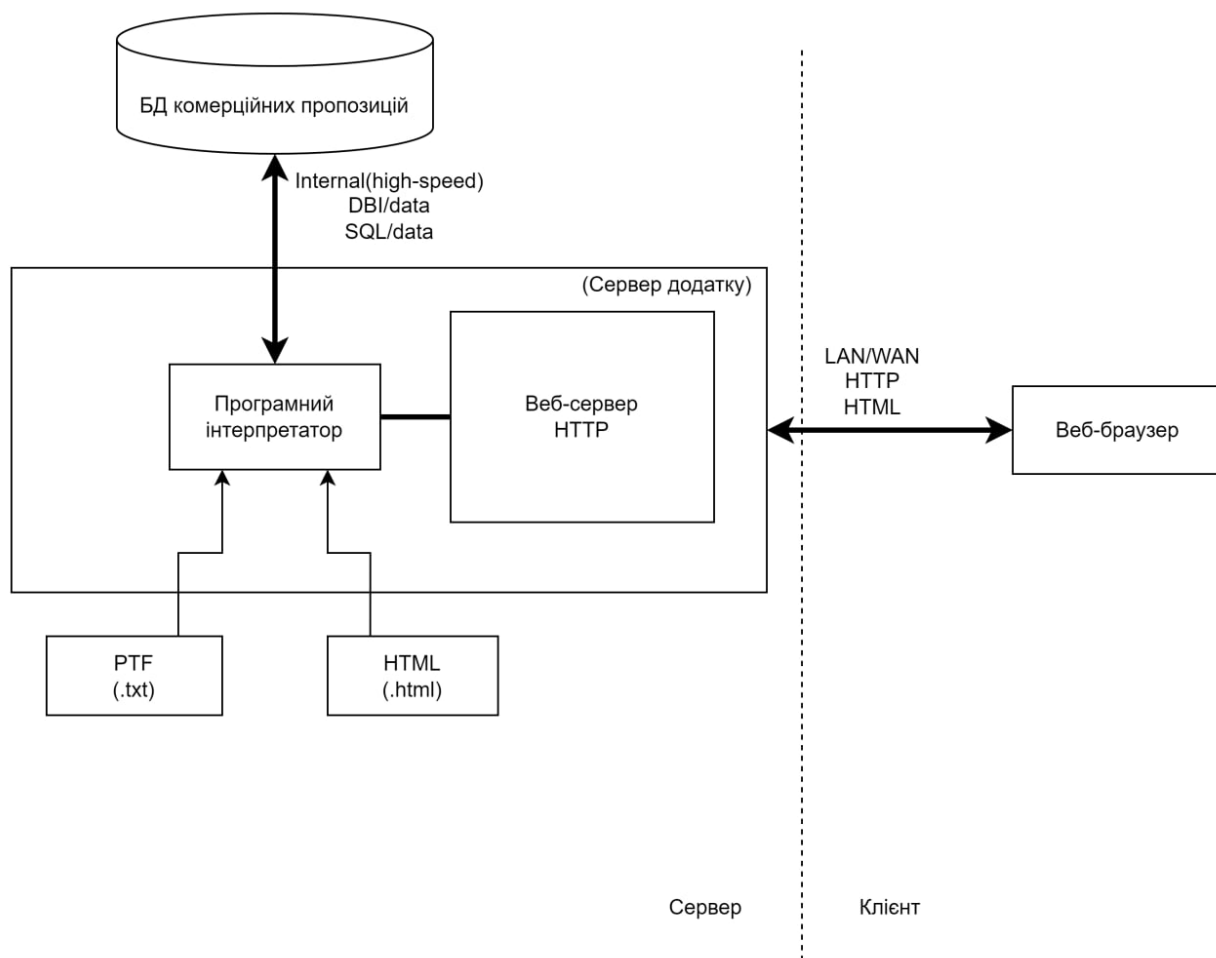


Рисунок 2 – Діаграми розгортання

2.2 Проектування архітектури програмного модуля

2.2.1 Архітектура СДПР

При проектуванні системи потрібно чітко розмежувати рівні даних, логіки та уявлення системи, що розробляється. Подібний поділ відповідає шаблону модель-подання-контролер (MVC pattern) в архітектурному плані. Архітектура системи, що розробляється, представлена на рис. 2. На рівні даних виділяються сутності (entities), необхідні для вирішення задачі діагностики (як адміністративні, так і сутності, описують ознаки, діагнози і т.д.). Сутності містять у собі виключно дані. Менеджери сутностей є реалізацію шаблону об'єкта доступу до даних (DAO pattern), та надають можливість виконання CRUD-операцій над наявними даними: здійснюють вибірку, зміну, додавання та видалення екземплярів сутностей.

Менеджери сутностей взаємодіють із рівнем логіки докладання, при цьому самі бізнеслогіки не містять; є єдиним джерелом даних для всіх послуг системи.

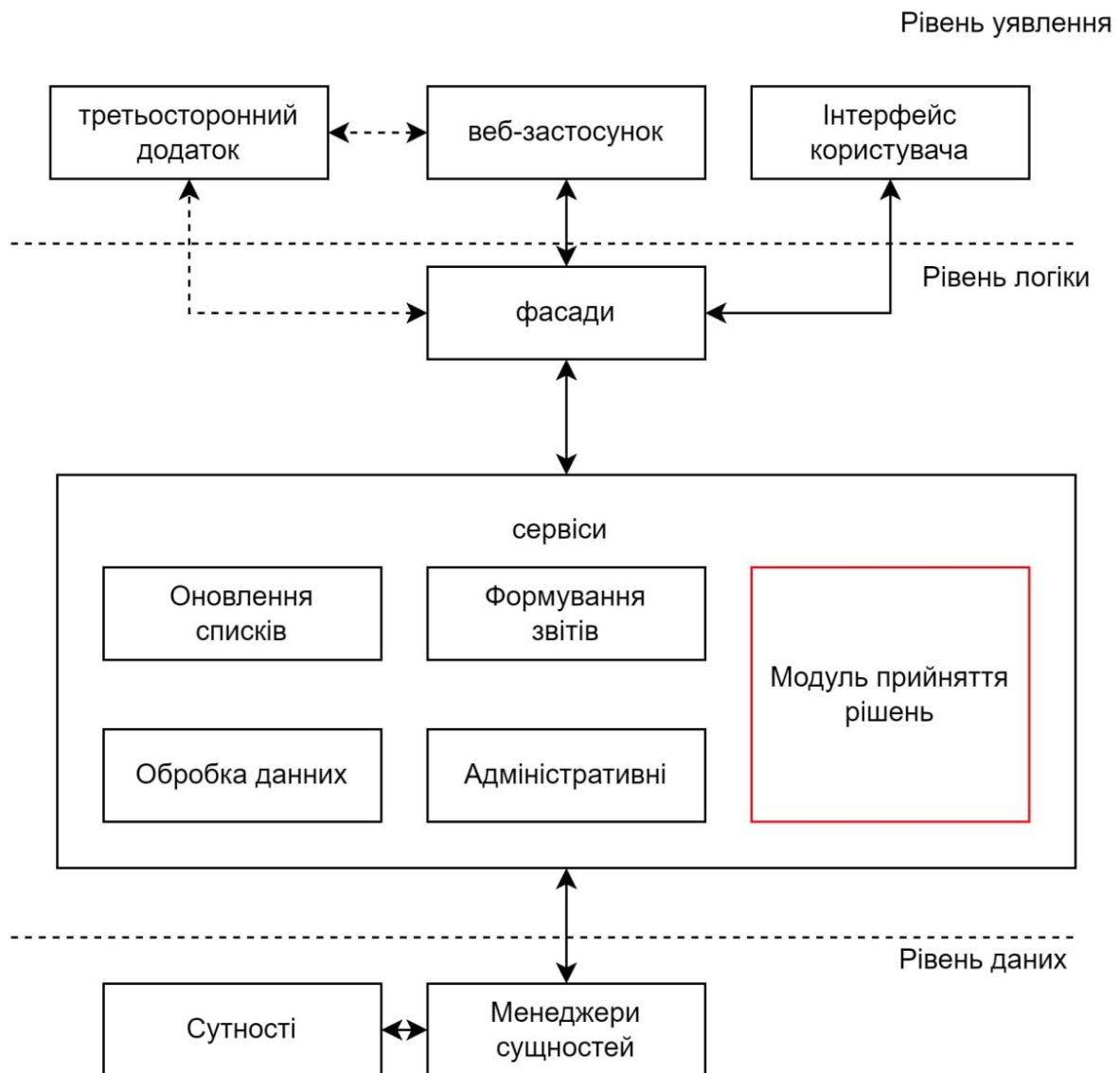


Рисунок 3 – Архітектура СДПР

Зв'язок даних БД і сутностей програми здійснюється за допомоги механізму об'єктно-реляційного зв'язування (ORM) Для цього передбачається використання бібліотеки Hibernate, яка дозволяє описати зв'язки полів сутностей з колонками таблиць БД (з допомогою анотацій, або XML-описи), при цьому зникає необхідність написання великого обсягу JDBC-коду для отримання даних із БД.

Рівень логіки включає сервіси системи, що містять всю бізнес-логіку програми. Адміністративні послуги відповідають за обробку даних про пацієнтів, співробітників, ведення відповідної медичної документації. Сервіси обчислень надають доступ до різних алгоритмів: формування діагностично значимих інтервалів ознак, генетичні алгоритми тощо. Сервіси формування знань є реалізацією шаблону будівельник (Builder pattern), здійснюють побудову фреймів знань системи, структура яких розглянуто вище. Сервіси обробки експертних даних здійснюють обробку даних, наданих експертами (функції приладдя, опис структури симптомокомплексів, ваги ознак та ін.) і готують метадані для роботи сервісів формування знань. Діагностичні сервіси, використовуючи знання системи, що здійснюють підтримку прийняття рішень: на підставі вхідних даних про результати обстеження пропонується деяке безліч діагнозів, що відповідають стану діагностованого. Фасади (Facade pattern) забезпечують представлення ядра системи поза (API для інтерфейсу користувача та третьосторонніх додатків), приховуючи від зовнішнього світу всю внутрішню реалізацію та Структуру системи.

Кожен з фасадів є обгорткою для одного, або кількох сервісів, у своїй може містити додаткову логіку (наприклад, валідацію вхідних даних). Рівень представлення включає графічний інтерфейс користувача (GUI) та веб-сервіс. GUI є самостійним додатком, реалізованим відповідно до вимог шаблону MVC. GUI використовує API ядра системи, що надаються фасадами. Взаємодія з третьосторонніми програмами може здійснюватись як безпосередньо через фасади (у тому випадку, якщо третестороння програма є Java-додатком, то можливий прямий виклик методів фасадів, використовуючи механізм RMI), так і через веб-сервіс. Веб-сервіс є обгорткою для всіх фасадів системи, надає можливість спілкуватися з ядром системи за допомогою обміну SOAP-повідомленнями[14].

2.2.2 Розробка алгоритму програмного модуля

Перший етап проектування додатку починається з формування та розробки алгоритму сортування та допомоги прийняття рішень для моєї системи комбікорму.

Початок мого алгоритму починається з вводу необхідної кількості тварин та вид тварини. Наступний крок, це перевірка необхідної кількості комбі корму яка вписана в системі, необхідно перевірити загалом кількість комбі корму від перших місяців життя тварини до дорослого віку. Опишемо загальну структуру мого алгоритму. Система на початку перевіряє взагалі наявність такого комбі корму для вибранної тварини, якщо вона існує в списку нашої системи ми перевіряємо далі, якщо не існує то система видає інформацію про відсутність комбі корму для тварини.

Наступний етап, при умові наявності тварини в нашій системі є перевірка можливих варіантів для видачі рекомендацій, чи є комбі корм в нашій компанії для вибранної тварини, чи є необхідна кількість, якщо в нашій компанії взагалі відсутній комбі корм для необхідної тварини, ми перевіряємо наявність в інших компаніях, та видаємо рекомендації для закупівлі в інших компаніях нашої системи. Якщо в нашій компанії є такий корм та є необхідна кількість – система видає рекомендації видачі комбі корму тільки закупівлі з нашої компанії. Якщо в нашій компанії є такий комбі корм, але його не вистачає система перевіряє наявність в інших компаніях, якщо в них не має такого комбі корму взагалі, система видає рекомендації закупівлі тільки в нашій компанії та рекомендації закупівлі в інших системах та перевірки на інших ринках. Якщо, в нашій та в інших компаніях нашої системи є необхідна кількість, рекомендуємо закупівлі тільки на нашому ринку з нашої системи, якщо відсутня необхідна кількість взагалі, видаємо рекомендацію системи з повідомлення про наявність в

перегляду можливих варіантів для закупівлі корму для необхідної кількості необхідної тварини в який належить перегляд історію запитів минулих пошуків кормів.

До адміністративних функцій належить два основні модулі, модуль роботи з користувачами для створення звітів по користувачам та модуль роботи з базами даних. Адміністратор може додавати нові компанії та види корму або редагувати вже відомі компанії з їх преїскурантами.

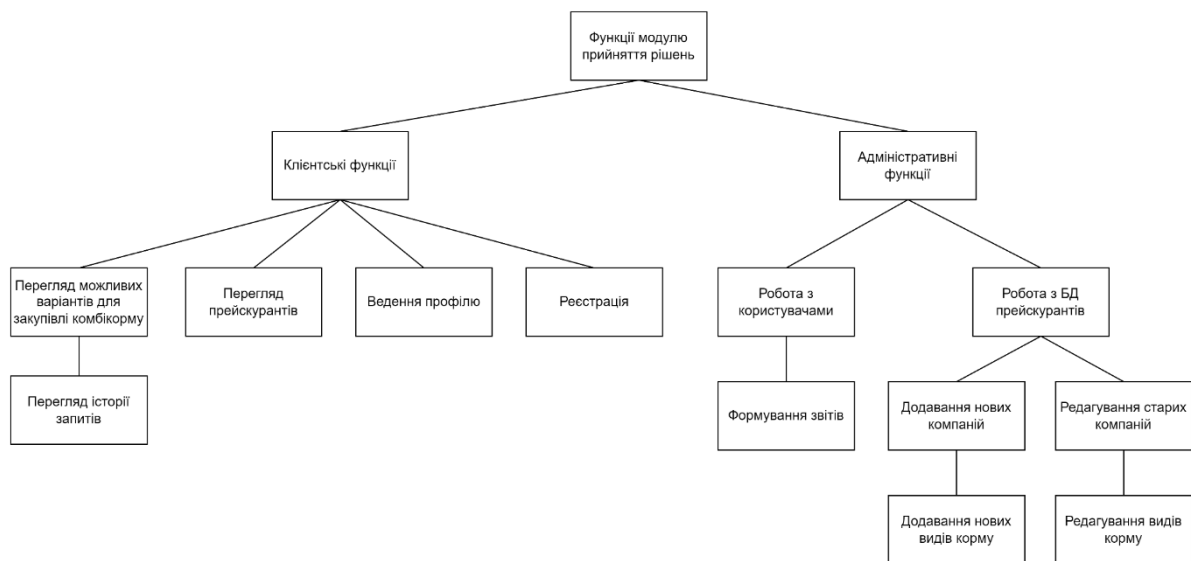


Рисунок 5 – Дерево функцій.

Наступний етапом роботи розробки модуля це опис функціонування додатку для системи прийняття рішень. Початком запуску додатку є відкритий додаток та авторизація для старих користувачів або реєстрація для нових користувачів. Після успішної авторизації користувач переходить до сторінки преїскурантів та має можливість переглянути всі можливі преїскуранти та можливість дати запит на необхідну тварину та необхідну кількість для розрахунку оптимального рішення завдяки системі

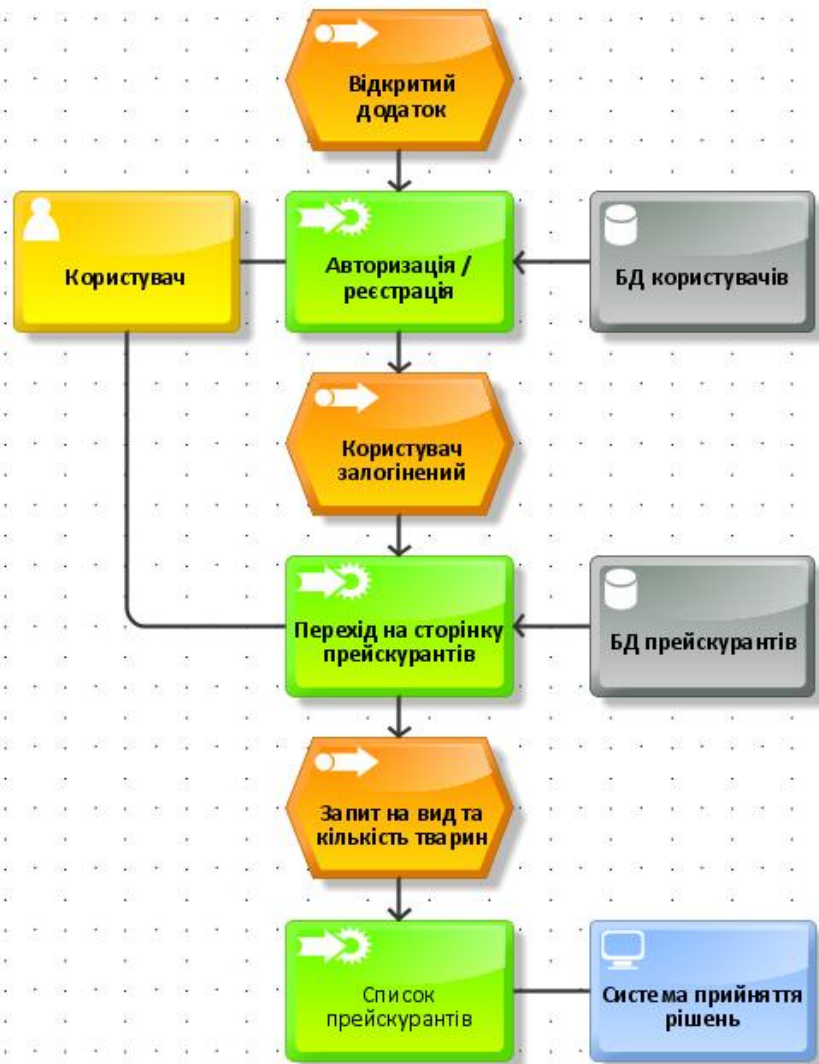


Рисунок 6 – Діаграма Event-Driven Process Chain для функціонування WEB-додатку.

Ланцюжки процесів допоможуть переглянути процеси додатку та можливості користувача в самому додатку на вершині знаходиться процес “Робота з системою прийняття рішень” яка включає в собі основні підпроцеси: “Реєстрація в системі прийняття рішень” та “Користування системою прийняття рішень”.

До процесу “Реєстрація в системі прийняття рішень” входить підпроцеси “Заповнення основних даних” в якому знаходяться підпроцеси

“Заповнення логіну”, “Заповнення паролю”, “Заповнення ПІБ” та “Заповнення email” всі підпроцеси допомагають користувачу вводити свої особисті дані для реєстрації в додатку, після заповнення всіх основних даних користувач переходить до процесу “Підтвердження реєстрації”.

До процесу “Користування системою прийняття рішень” користувач переходить до основних підпроцесів системи, а саме: підпроцес “Вибір тварини та її кількість” який переходить до підпроцесу системи “Список преїскурантів які можуть надати необхідну кількість корму”, наступний підпроцес процесу “Користування системою прийняття рішень” – “Історія запитів” для перегляду історії всіх минулих пошуків кормів. Останній підпроцес це – “Перегляд преїскурантів” який допоможе користувачу отримати весь список всіх компаній.

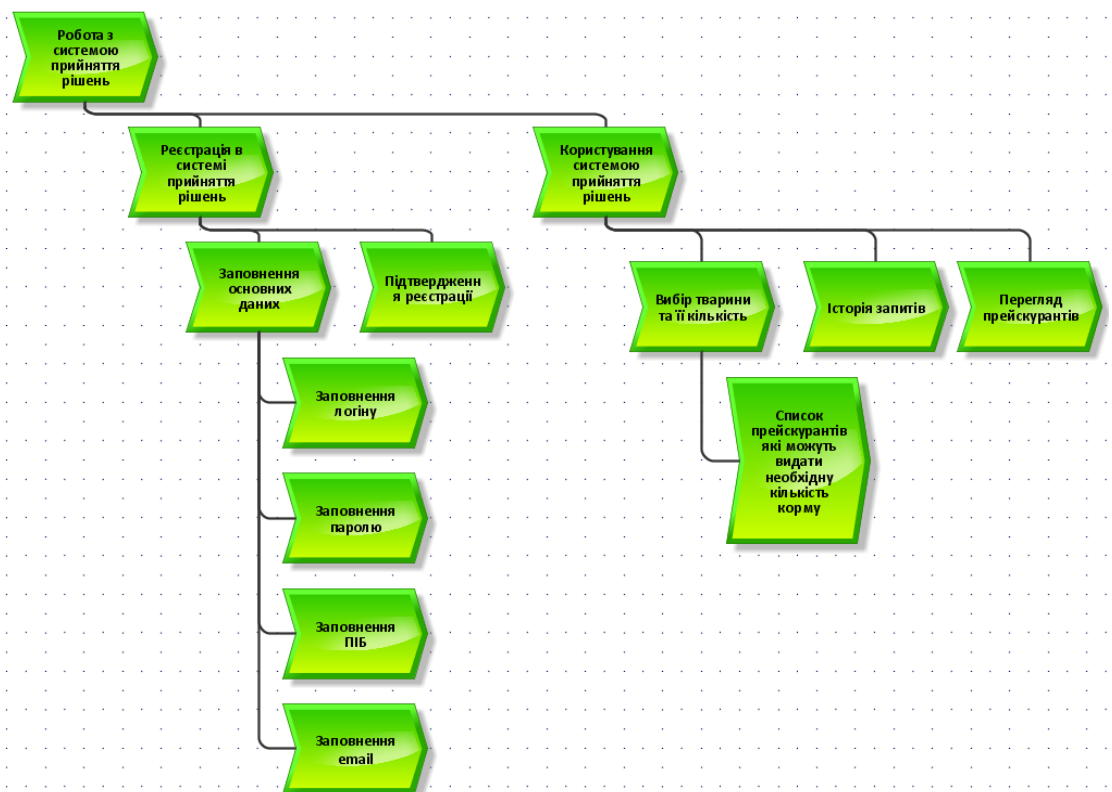


Рисунок 7 – Діаграма Value Added Chain Diagram для функціонування мобільного додатку.

2.2.4 IDEF0 процесу видачі списку необхідного комбікорму завдяки програмному модулю підсистеми підтримки рішень для малого сільського господарства

Діаграма “ЯК БУДЕ” у нотації IDEF0 допоможе представити виконувані процеси. А саме вже з використанням браузерного додатку інформаційній СДПР з питань кормовиробництва для фермерського господарства.

IDEF0 — Function Modeling — методологія функціонального моделювання та графічного опису процесів, призначена для формалізації та опису бізнес-процесів. Особливістю IDEF0 є її акцент на ієрархічному представленні об'єктів, що значно полегшує розуміння предметної області. У IDEF0 розглядаються логічні зв'язки між роботами, а не послідовність їх виконання у годині (WorkFlow).

Також відображаються всі сигнали управління. Така модель є однією з найпрогресивніших моделей і використовується в організації бізнес-проектів і проектів, що базуються на моделюванні всіх процесів як адміністративних, так і організаційних [15].

Розглянемо процес вибору комбікорму завдяки системі прийняття рішень закупівлі комбікорму. На вході система приймає дані користувача для перевірки авторизованих користувачів або відсорткування нових користувачів для створення нового акаунту. Дані вводить користувач в додаток СДПР з питань кормовиробництва для фермерського господарства на виході користувач отримує список необхідних комбікормів.

Серед елементів керування для даного процесу зазначено довідники відгодовування тварин та норми виробництва комбікорму для усунення випадків захворювань у тварин та забезпечення повним списком поживних речовин.

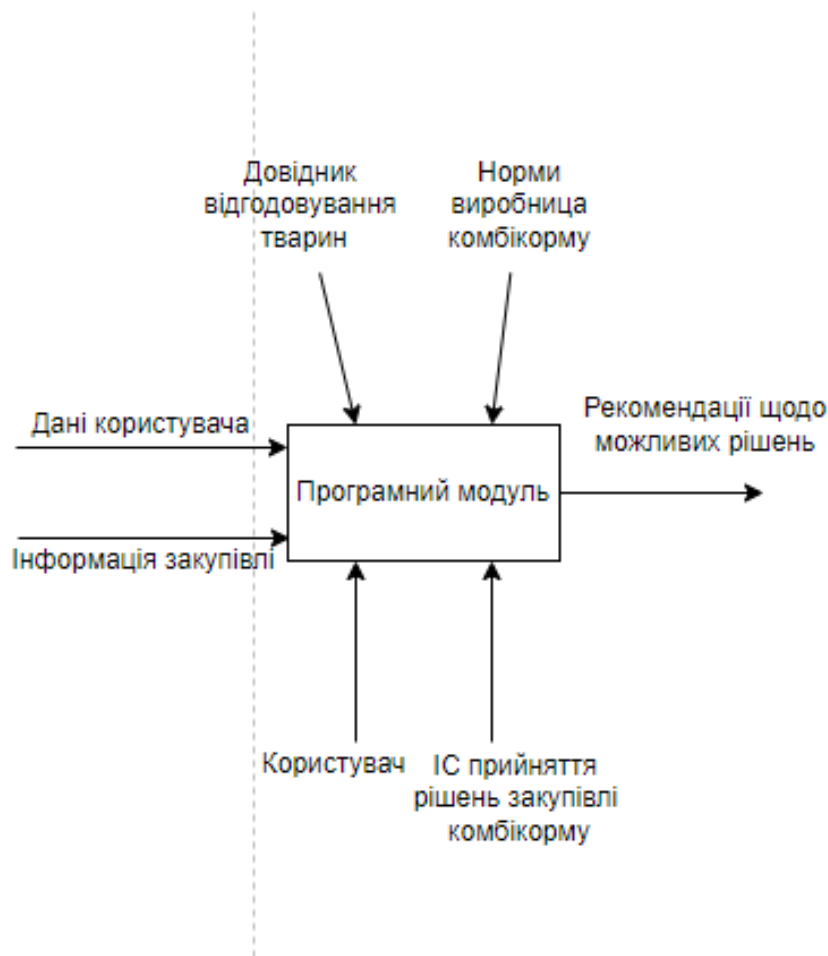


Рисунок 8 – Контекстна діаграма ЯК БУДЕ.

Розглянемо процес вибору комбікорму завдяки системі прийняття рішень на більш детальній діаграмі декомпозиції основних процесів “ЯК БУДЕ”. На вході до системи отримуємо особисті дані користувача для відсортування нових та старих користувачів, після авторизації система отримує інформацію вже авторизованого користувача та пропонує вибрати необхідну кількість необхідної тварини та її кількість. Після вводу користувачем необхідної тварини система переглядає існуючі корми в нашій компанії та переглядає потім в інших компаніях можливі варіанти та підраховує всі витрати в нашій компанії та в інших і користувач отримує список комбікормів.

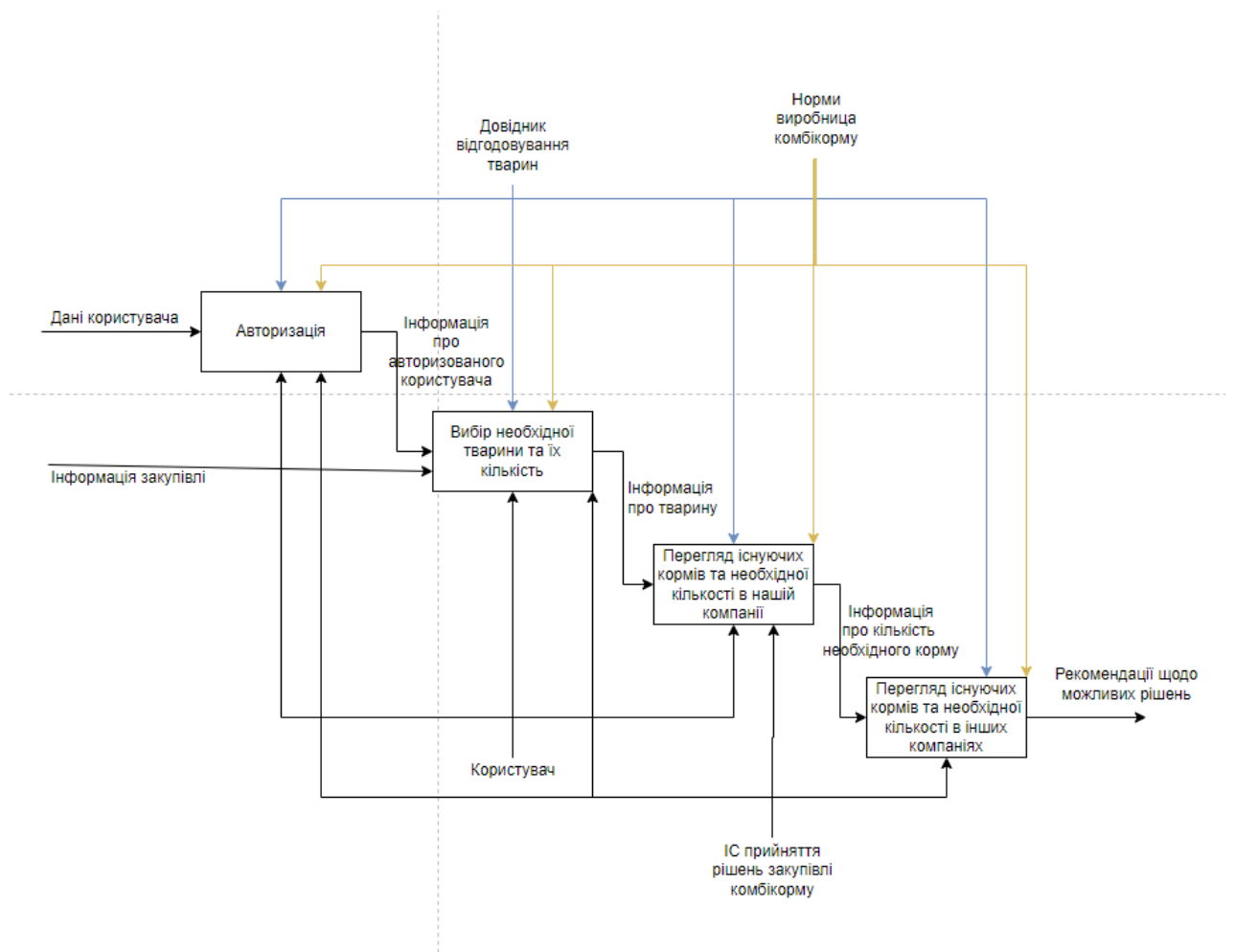


Рисунок 9 – Декомпозиція основних процесів діаграми ЯК БУДЕ.

2.2.5 Архітектура програмного модуля підсистеми підтримки рішень для малого сільського господарства

Архітектура розроблюваної інформаційної системи - новий етап побудови нової діаграми. Дана схема відображає модель, структуру, виконувані функції й взаємозв'язок компонентів. Дана система ділиться на модулі роботи з клієнтами та адміністративний, відповідно до дерева функцій.

Модуль роботи з клієнтами вміщує в собі модуль роботи з профілем який дозволяє редагувати особисту інформацію акаунту, модуль реєстрації допомагає новим користувачам увійти до системи та останній модуль роботи з пошуком комбікорму – допомагає переглядати прейскуранти нашої та чужих компаній. Та найголовніший клієнтський модуль роботи з системою

прийняття рішень яка допоможе виявити оптимальне рішення закупівлі комбікорму від постачальників.

Адміністративний модуль вміщує в собі модуль звітності користувачів, їх активність та популярні запити пошуку користувачів. Модуль підсистеми роботи з бд включає в собі модули роботи з базою даних користувачів та базою даних кормів.

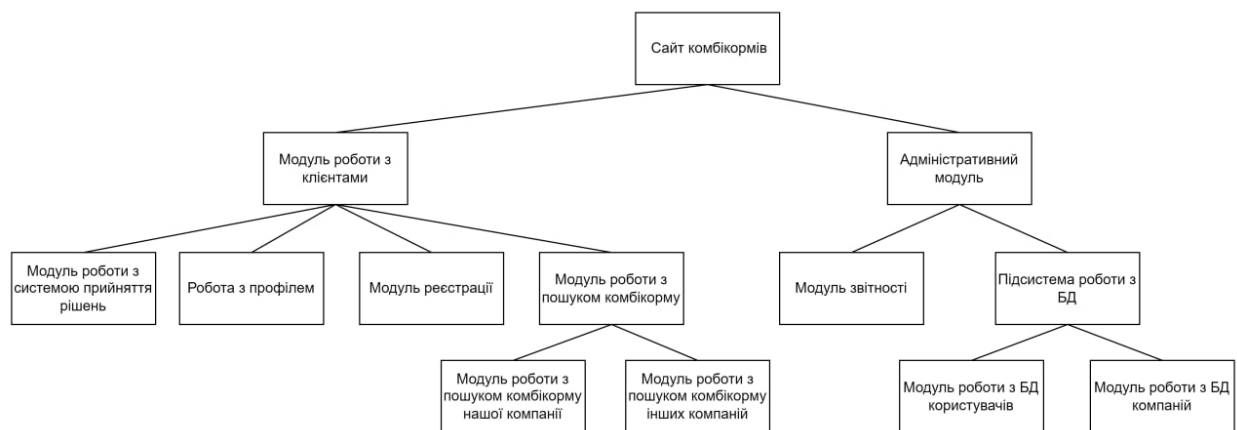


Рисунок 10 – Архітектура системи.

2.3 Інформаційне забезпечення програмного модуля підсистеми підтримки рішень для малого сільського господарства

2.3.1 Розробка інформаційного забезпечення

Як вже було зазначено раніше, дана навчальна система для правильної роботи потребує правильно розподілені дані.

Спочатку розробимо концептуальну модель бази даних (рис. 13). На ній для заданої предметної області відображено наступні сутності:

- Користувач
- Компанія
- Корм

Таблиця 2. - Зв'язки між сутностями

№	Сутності, що утворюють зв'язок	Тип зв'язку	Пояснення
1	Користувач – Компанія	Один до багатьох	Один користувач отримує багато компаній. Декілька компаній для одного користувача.
2	Компанія – Корм	Один до багатьох	Одна компанія має декілька видів корму. Кожен корм для окремої компанії

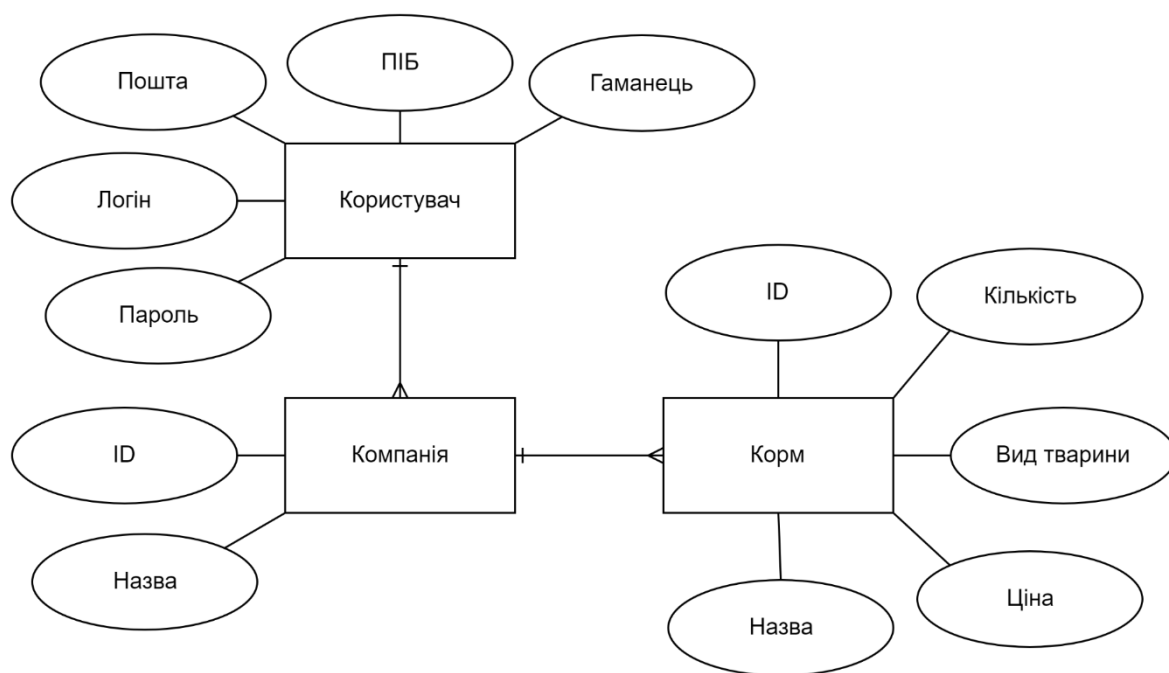


Рисунок 11 – Концептуальна модель бази даних.

Таблиця 2.3. - Склад та характеристика атрибутів таблиць

№ п/п	Назва елемента даних	Тип елемента даних	Обов'язков е значення	Обмеженн я	Ключ
Таблиця User «Користувач»					
1	Login	Символьний	так		ПК

2	name	Символьний	так		
3	password	Символьний	так		
4	email	Символьний	так		
5	money	Ціле число			
6	id Companies	Ціле число	так		ЗК
Таблиця Companies «Компанія»					
1	id Companies	Ціле число			ПК
2	name	Символьний			
3	id feed	Ціле число			ЗК
Таблиця feed «Корм»					
1	id feed	Ціле число	так		ПК
2	name	Символьний			
3	price	Ціле число			
4	amount	Ціле число			
5	kind	Символьний			

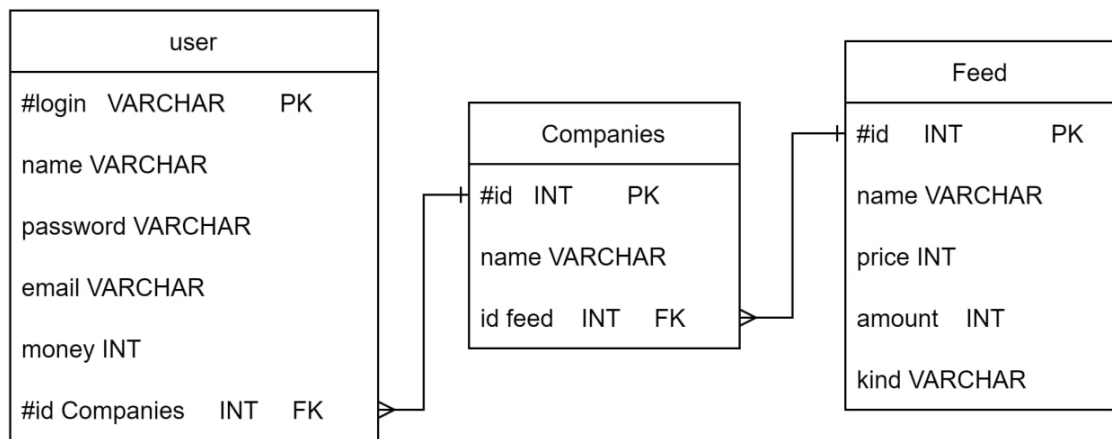


Рисунок 12 – Фізична модель бази даних.

Останньою є фізична модель бази даних (рис. 14), розроблена на основі даталогічної. Вона вже відображає, як дані насправді зберігаються в базі даних та обмеження кожного з полів.

2.4 Висновки до другого розділу

В результаті проведення етапу розробки програмного модуля підсистеми підтримки рішень для малого сільського господарства

Було проведено функціональний аналіз та побудовано дерево функцій, яке відображає бізнес-процеси інформаційної системи. Також було розроблено діаграми Event-Driven Process Chain та Value Added Chain Diagram для функціонування мобільного додатку. На даних схемах було зображено та описано логіку роботи програми.

Також було представлено діаграми IDEF0 "ЯК БУДЕ", які показують процеси видачі рекомендацій, виконувані з використанням даної туристичної системи.

Було сформовано архітектуру системи для відображення її внутрішніх компонентів та їх взаємозв'язок.

РОЗДІЛ 3. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ПРОГРАМНОГО МОДУЛЯ ПІДСИСТЕМИ ПІДТРИМКИ РІШЕНЬ ДЛЯ МАЛОГО СІЛЬСЬКОГО ГОСПОДАРСТВА

3.1 Обґрунтування вибору програмних засобів

Для виконання даної роботи, а саме програмний модуль підсистеми підтримки рішень для малого сільського господарстварозробити WEB-додаток для реалізації завдання.

Для розробки застосунку було використано ряд інструментів та методів, за допомогою яких створено програмний модуль підсистеми підтримки рішень для малого сільського господарства.

Основна мова програмування JavaScript ES6 з використанням HTML5 та CSS3.

Було прописано повністю DOM-дерево на базі JavaScript ES6 без використання інших готових бібліотек, завдяки цьому на сайті було спроектовано адаптивний інтерфейс та блоки преїскурантів в залежності від бази даних.

Основна мова програмування JavaScript ES6 з використанням HTML5 та CSS3. JavaScript створювався як скриптова мова для браузерів Netscape. Компанія Microsoft також визнала його потенціал і включила під ім'ям JScript в Internet Explorer 3, забезпечивши часткову підтримку стандартів мови, що призвело до безладу зі стандартами і версіями JavaScript. Тому Netscape, Microsoft та інші зацікавлені компанії звернулися до організації ECMA (Європейська асоціація виробників комп'ютерів), де було схвалено першу специфікацію мови ECMA-262. У зв'язку з тим, що назва «JavaScript» була зареєстрованим товарним знаком, для нового стандарту було вирішено використовувати ECMAScript (або скорочено ES). ECMAScript спочатку був розроблений для використання як мова сценаріїв, але пізніше став широко використовуватися як мова програмування загального призначення[7].

HTML-документ - це звичайний текстовий документ, може бути створений як у звичайному текстовому редакторі (Блокнот), так і в спеціалізованому з підсвічуванням коду (Notepad++, Visual Studio Code і т.п.). HTML-документ має розширення “.html”.

HTML-документ складається з дерева HTML-елементів та тексту. Кожен елемент позначається у вихідному документі початковим (відкриває) і кінцевим (закриває) тегом (за рідкісним винятком)[8].

CSS (Cascading Style Sheets) — мова таблиць стилів, яка дозволяє прикріплювати стиль (наприклад, шрифти та колір) до структурованих документів (наприклад, документів HTML та додатків XML).

Зазвичай CSS-стилі використовуються для створення та зміни стилю елементів веб-сторінок та інтерфейсів користувача, написаних на мовах HTML і XHTML, але також можуть бути застосовані до будь-якого виду XML-документа, у тому числі XML, SVG і XUL[9].

3.3 Структура програмного забезпечення

Сторінка index.html “Головна сторінка” яка відображає всі необхідні елементи та включає в собі основні чотири файли на мові програмування JavaScript – script.js який включає в собі головні функції: function login() функція перевірки користувача, функція реєстрації function reg(), та велика функція яка складається з маленьких частин для перевірки кількості кормів в нашій та інших компаніях.

Таблиця 4. -Специфікація програмних модулів

Модуль	Опис
Головна сторінка	Початковий модуль, на якому відображається весь процес відображення всієї інформації. - Вихідна інформація: Блок авторизації, блок преїскурантів та блоки вибору найкращих кормів.

script.js	Головний файл який вміщує в собі всю логіку сторінки.
function login()	Функція для авторизації старих користувачів.
function reg()	Функція для створення нових користувачів.
функція СДПР	Головна функція яка складається з інших функцій для перевірки наявності взагалі в нашій компанії продукту, якщо він відсутній шукає в інших компанія, для перевірки необхідної кількості продуктів в нашій компанії, вистачить взагалі у нас чи необхідно пропонувати докупити або відразу все купити в інших компанія, також функція відображає та створює головні блоки через DOM-дерево на сторінці зі всіма рекомендаціями.

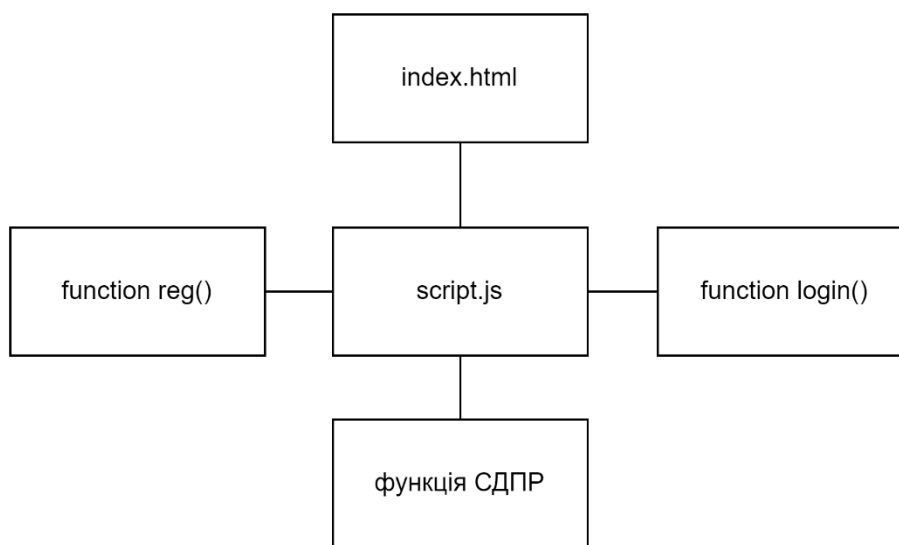


Рисунок 13 – Структурна схема програмних модулів

3.4 Керівництво користувача

Для початку користувача зустрічає поле(рис. 13) для вводу логіну та паролю для старих користувачів або можливість зареєструватися для нових користувачів.

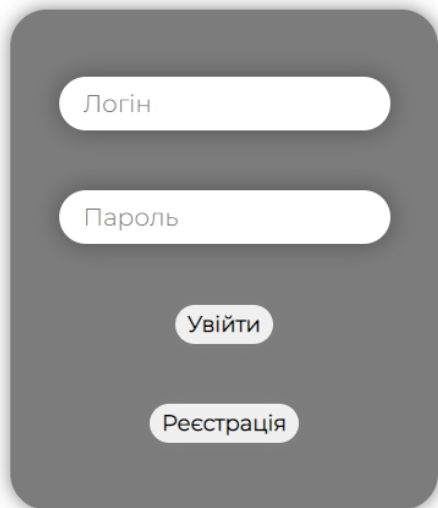
A dark gray rounded rectangle containing two white input fields. The top field is labeled 'Логін' and the bottom field is labeled 'Пароль'. Below the fields are two white buttons: 'Увійти' and 'Реєстрація'.

Рисунок 14 – Блок для авторизації користувача

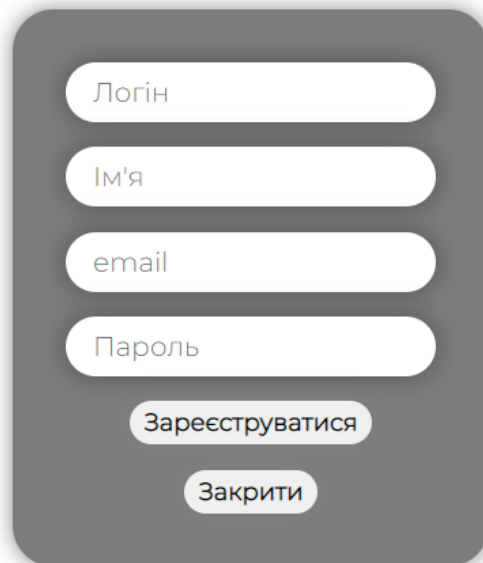
A dark gray rounded rectangle containing four white input fields. From top to bottom, they are labeled 'Логін', 'Ім'я', 'email', and 'Пароль'. Below the fields are two white buttons: 'Зареєструватися' and 'Закрити'.

Рисунок 15– Блок для реєстрації користувача

Для нових користувачів можлива функція реєстрації(рис. 15) яка відкриває блок для вводу інформації про нового користувача.

Після реєстрації користувача відправляє до минулого блоку(рис.16) для авторизації користувача. Після успішної авторизації користувачу відкривається головний блок з преїскурантами(рис.18).

Комбікорм власного виробництва

Наша компанія

№	Назва корму:	Вид тварини:	Кількість:	Ціна:
1.	Великий індик	<u>Індик</u>	300 кг	26.3 грн/кг
2.	Індик №123	<u>Індик</u>	600 кг	23.3 грн/кг
3.	Свиня 2000	<u>Свиня</u>	1000 кг	30.57 грн/кг
4.	Золота курка	<u>Курка</u>	400 кг	16.42 грн/кг
5.	Коровка МУ	<u>Корова</u>	1000 кг	31.13 грн/кг

ПКомбікорм іншого виробництва

Компанія 1

№	Назва корму:	Вид тварини:	Кількість:	Ціна:
1.	Корм для корови	<u>Корова</u>	5000 кг	40.12 грн/кг
2.	Гарна вівця №12	<u>Вівця</u>	3000 кг	36.33 грн/кг
3.	Індик 100кг №7	<u>Індик</u>	5000 кг	34.18 грн/кг
4.	Свиня як корова	<u>Свиня</u>	1300 кг	37.99 грн/кг

Компанія 2

№	Назва корму:	Вид тварини:	Кількість:	Ціна:
---	--------------	--------------	------------	-------

Рисунок 16 – Блок прейскурантів нашої та інших компаній

На головній сторінці прейскурантів відкривається можливість до двох головних функцій сторінки(рис. 19), а саме розрахунок майбутнього поголів'я та історія пошуку.

Розрахунок майбутнього поголів'я

Історія пошуку

Комбікорм власного виробництва

Наша компанія

№	Назва корму:	Вид тварини:	Кількість:	Ціна:
1.	Великий індик	<u>Індик</u>	300 кг	26.3 грн/кг
2.	Індик №123	<u>Індик</u>	600 кг	23.3 грн/кг
3.	Свиня 2000	<u>Свиня</u>	1000 кг	30.57 грн/кг
4.	Золота курка	<u>Курка</u>	400 кг	16.42 грн/кг
5.	Коровка МУ	<u>Корова</u>	1000 кг	31.13 грн/кг

ПКомбікорм іншого виробництва

Компанія 1

№	Назва корму:	Вид тварини:	Кількість:	Ціна:
1.	Корм для корови	<u>Корова</u>	5000 кг	40.12 грн/кг
2.	Гарна вівця №12	<u>Вівця</u>	3000 кг	36.33 грн/кг

Рисунок 17 – Історія пошуку

Якщо користувач відкриває сторію без минулих запитів тоді він отримує наступне повідомлення про те що історія відсутня(рис. 20).

Історія пошуку пуста!

Рисунок 20 – Повідомлення про відсутню історію пошуку

Якщо користувач раніше робив пошук то він отримає ну пусте повідомлення про історію, а отримає повідомлення про те що саме він шукав раніше та можливість відновити запит або закрити вікно історії (рис. 21).

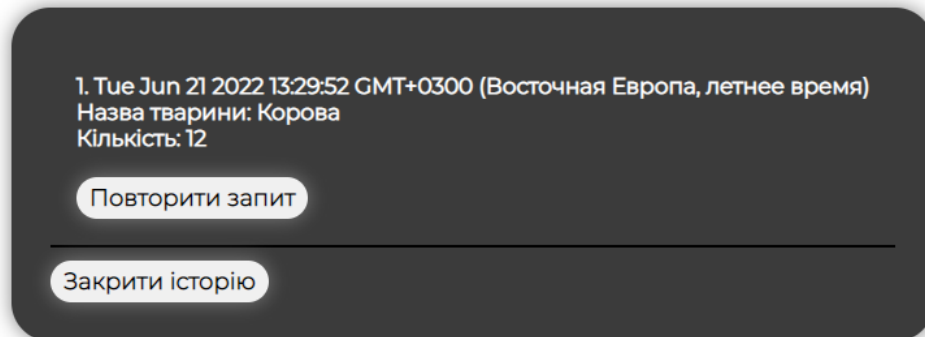


Рисунок 18 – Повідомлення про історію пошуку

Якщо користувач натисне кнопку для алгоритму для нього відкриється меню з можливим вводом інформації про тварину та кількість, почати алгоритм або закрити це вікно.(рис. 21)

<p>Рисунок 19 – Вікно для вводу інформації</p>	<p>Рисунок 20 – Ввод не вірної інформації</p>

Якщо користувач вводить невідому назву тварини та почне пошук, він буде повідомлено що такої тварини не існує в нашій базі даних і нічого не отримує крім повідомлення.

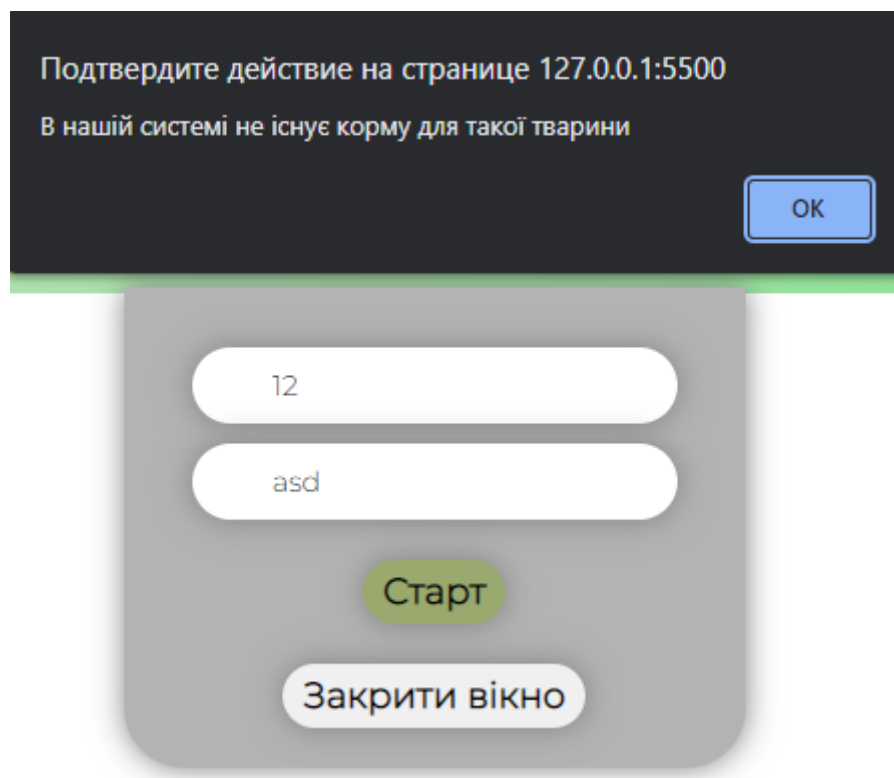


Рисунок 21 – повідомлення про неіснуючу тварину

Якщо користувач введе вірну тварину та її кількість він отримає список можливих варіантів для закупівлі в нашій компанії, або в інших, можливе повідомлення про те що у нас неможливо отримати одним замовленням для тварини і можливо купити частину 1 корму та частину іншого, або взагалі скупити весь корм у нас та докупити в інших компанія, або можливий варіант що користувачу необхідно стільки корму, що у нас навіть не має і отримає повідомлення про те що необхідно шукати на інших сайтах таку кількість корму.

Необхідний корм для користувача в кількості: 4200кг для 'Корова' - 12 голів

Компанія 1				
1.	Корм для корови	Корова	5000 кг	40.12 грн/кг
Вам буде коштувати: 168504 грн. у вас не вистачає 163826 грн.				

Компанія 4				
2.	Корм для корови	Корова	50000 кг	42.12 грн/кг
Вам буде коштувати: 176904 грн. у вас не вистачає 172226 грн.				

Рисунок 22 – повідомлення про те що неможливо у нашій компанії закупити відразу корм однією партією

Коли користувач введе чітку кількість корму, існуючу тварину і система зрозуміє, що в нас є необхідна кількість корму але ми не можемо її видати одним видом корму – система повідомляє скільки необхідно корму, скільки загалом у нас є різними кормами та видає повідомлення про комбінування різних видів кормів та видає список компаній які можуть видати комбікорм однією партією(рис. 25).

У нас є такий корм, але не можемо віддати тільки 1 вид корму, можемо пропонувати різні види корму

Необхідно користувачу 640кг для "Індик" -8 голів
Загалом у нас є 900кг

НАША КОМПАНІЯ				
Повна ціна за корм:13980 грн.				
1.	Великий індик №123	Індик	600 кг	23.3 грн/кг

НАША КОМПАНІЯ				
Повна ціна за корм:7890 грн.				
2.	Великий індик	Індик	300 кг	26.3 грн/кг

Необхідний корм для користувача в кількості: 640кг для 'Індик' - 8 голів

Компанія 2				
1.	ТОВ великдень	Індик	2000 кг	30.44 грн/кг
Вам буде коштувати: 19481.6 грн. у вас не вистачає 14803.6 грн.				

Компанія 4				
2.	Великий індик №123123	Індик	60000 кг	33.3 грн/кг
Вам буде коштувати: 21312 грн. у вас не вистачає 16634 грн.				

Компанія 1				
3.	Індик 100кг	Індик	5000 кг	34.18 грн/кг
Вам буде коштувати: 21875.2 грн. у вас не вистачає 17197.2 грн.				

Рисунок 23 – повідомлення про те що можливо у нашій компанії купити корм різними партіями, або закупити однією в інших компаніях

Коли користувач вводить кількість тварин та її вид, система аналізує наступне – чи можемо ми відправити всю партію, чи в нас є лише частина? Користувач отримує повідомлення про те, що в нас є такий корм але не має необхідної кількості – тільки частина. Також отримує список всіх компаній які можуть видати необхідну кількість комбікорму однією партією.

У нас є такий корм, але його не вистачає. Необхідно 2640кг для 'Індик' - 33 голів

У нас є лише: 900кг, не вистачає 1740кг, це хватає лише для 11.25 голів, можливо взяти в інших компаніях корму для 21.75 голів

Повна ціна за всі корми(2 штук) нашої компанії:21870грн.

НАША КОМПАНІЯ
Повна ціна за корм:13980 грн.

1.	Великий індик №123	Індик	600 кг	23.3 грн/кг
----	--------------------	-------	--------	-------------

НАША КОМПАНІЯ
Повна ціна за корм:7890 грн.

2.	Великий індик	Індик	300 кг	26.3 грн/кг
----	---------------	-------	--------	-------------

Необхідний корм для користувача в кількості: 2640кг для 'Індик' - 33 голів

Компанія 4

1.	Великий індик №123123	Індик	60000 кг	33.3 грн/кг
----	-----------------------	-------	----------	-------------

Вам буде коштувати: 87912 грн. у вас не вистачає 83234 грн.

Компанія 1

2.	Індик 100кг	Індик	5000 кг	34.18 грн/кг
----	-------------	-------	---------	--------------

Вам буде коштувати: 90235.2 грн. у вас не вистачає 85557.2 грн.

Рисунок 24 – повідомлення про те що у нас немає такого обсягом, але є такий продукт для необхідної тварини та рекомендація докупити в інших, або відразу купити в інших.

Коли користувач введе таку кількість тварин, що в нас не має такої кількості комбі корму, користувач отримує інформацію скільки взагалі в нашій системі знайдено такого виду комбікорму та видає інформацію про те скільки необхідно буде користувачу ще для закупівлі.

У нас є такий корм, але його не вистачає. Необхідно 9849840кг для 'Індик' - 123123 голів

У нас є лише: 900кг, не вистачає 9848940кг, це хватає лише для 11.25 голів, можливо взяти в інших компаніях корму для 123111.75 голів

Повна ціна за всі корми(2 штук) нашої компанії: 21870грн.

НАША КОМПАНІЯ
Повна ціна за корм: 13980 грн.

1.	Великий індик №123	Індик	600 кг	23.3 грн/кг
----	--------------------	-------	--------	-------------

НАША КОМПАНІЯ
Повна ціна за корм: 7890 грн.

2.	Великий індик	Індик	300 кг	26.3 грн/кг
----	---------------	-------	--------	-------------

Всього в інших компаніях є 67000кг корму, це буде коштувати 2229780грн!
Вам необхідно знайти ще 9782840кг корму (Підрахунок без нашого корму)

Необхідний корм для користувача в кількості: 9849840кг для 'Індик' - 123123 голів

Компанія 2

1.	ТОВ великдень	Індик	2000 кг	30.44 грн/кг
----	---------------	-------	---------	--------------

Вам необхідно скупити весь корм!
60880 грн.

Компанія 4

2.	Великий індик №123123	Індик	60000 кг	33.3 грн/кг
----	-----------------------	-------	----------	-------------

Рисунок 25 – повідомлення про відсутність усієї кількості ні в нашій компанії ні в інших всього корму

3.5 Результат тестування

Останнім кроком оглянемо процес тестування інформаційної системи допомоги прийняття рішень з питань кормовиробництва для фермерського господарства.(табл. 5).

Таблиця 3.5 – Перелік основних тест-кейсів

id	Пріоритет	Модуль	Кроки	Очікувані результати
1	найвищий	login()	Можливість увійти в свій акаунт Кроки: 1. Ввести правильний логін 2. Ввести правильний пароль 3. Натиснути кнопку «Увійти»	1. Логін відображено у полі логіну 2. Пароль відображено як крапки у полі паролю 3. Користувач переходить на сторінку з прејскурантами 4. Відображає інформацію про користувача
2	високий	login()	Перевірка невірно заповненого паролю для входу Кроки: 1. Ввести правильний логін 2. Ввести неправильний пароль 3. Натиснути кнопку «Увійти»	1. Логін відображено у полі логіну 2. Пароль відображено як крапки у полі паролю 3. Користувач залишається на головній сторінці
3	високий	reg()	Перевірка успішної реєстрації Кроки: 1. Ввести новий логін який відсутній в нашій системі. 2. Ввести новий пароль, ПІБ та email.	1. Логін відображено у полі логіну 2. ПІБ, email, та пароль відображено 3. Користувач повертається до сторінки авторизації

			3. Натиснути кнопку «Зареєструватись»	
4	високий	reg()	Перевірка не успішної реєстрації Кроки: 1. Ввести існуючий логін який вже є в нашій системі. 2. Ввести новий пароль, ПІБ та email. 3. Натиснути кнопку «Зареєструватись»	1. Логін відображено у полі логіну. 2. ПІБ, email, та пароль відображено. 3. Користувач отримує повідомлення про все існуючий логін в системі.

3.6 Висновки до третього розділу

Для безпосередньої реалізації програмного модуля підсистеми підтримки рішень для малого сільського господарства на початку шляху став вибір мови програмування та застосунків реалізації матеріалу.

Було визначено структуру програмного застосунку, побудовано її у вигляді схеми та описано специфікацію програмних модулів у вигляді таблиці.

Заключним етапом був огляд процесу тестування створеного додатку інформаційної системи допомоги прийняття рішень з питань кормовиробництва для фермерського господарства, а саме – було створено ряд тест кейсів для перевірки функціоналу.

Висновок

Програмний модуль підсистеми підтримки рішень для малого сільського господарства як і в цілому СППР допоможе полегшити спосіб перевірки наявності необхідного комбікорму для фермерів та допоможе прийняти рішення для закупівлі в нашій компанії корму або в інших компаніях та кількість всього корму та ціну в кожній компанії.

В результаті реалізації вищеописаного підмодуля проведено роботу з аналізу існуючих технологічних рішень закупівлі корму. Сформовано алгоритм перевірки наявності кормів та функціональні та нефункціональні вимоги для розроблюваного WEB-застосунку. Розглянуто процеси даної предметної області. Розроблено архітектуру програмного модуля підсистеми підтримки рішень для малого сільського господарства. Розроблено та протестовано програмний продукт.

В результаті реалізації було створено програмний модуль підсистеми підтримки рішень для малого сільського господарства з використанням таких інструментів, як JavaScript ES6, HTML5 та CSS3. Також додаток було протестовано відповідно до описаних тест кейсів.

Розроблений додаток можна покращувати та доповнювати новими функціями у майбутньому.

Список використаних джерел

1. Подобєд, Л. І. Комбікорми і кормосуміші для молодняка сільськогосподарських тварин. — К. : Урожай, 1994. — 144 с.
2. Ларичев О. И., Петровский А. В. Системы поддержки принятия решений. Современное состояние и перспективы их развития. // Итоги науки и техники. Сер. Техническая кибернетика. — Т.21. М.: ВИНТИ, 1987.
3. Фермер [Архівовано 21 листопада 2016 у Wayback Machine.] // Юридична енциклопедія : [у 6 т.] / ред. кол.: Ю. С. Шемшученко (відп. ред.) [та ін.]. — К. : Українська енциклопедія ім. М. П. Бажана, 2004. — Т. 6 : Т — Я. — 768 с.
4. Ларичев О. И., Петровський А. В. Системы підтримки прийняття рішень. Сучасний стан та перспективи їх розвитку. // Підсумки науки та техніки. Сер. Технічна кибернетика. - Т.21. М.: ВИНТИ, 1987,
5. Закон України “Про державний контроль за дотриманням законодавства про харчові продукти, корми, побічні продукти тваринного походження, здоров’я та благополуччя тварин” від 18.05.2017 р. № 2042-VIII - <https://document.vobu.ua/doc/3653>
6. Система підтримки прийняття рішень щодо управління товарними запасами - <http://www.swsys.ru/index.php?page=article&id=2535>
7. *Nicholas C. Zakas. Next-Generation JavaScript Engines // High Performance JavaScript / Editor: Mary E. Treseler. — 1st ed. — 1005 Gravenstein Highway North, Sebastopol, CA 95472: O’Reilly Media, 2010. — P. xiii.*
8. *Connolly D., Masinter L. The 'text/html' Media Type — IETF, 2000. — 8 p.*
9. Дэйв Ши, Молли Е. Хольцшлаг. Философия CSS-дизайна. / Пер. с англ. А. Слинкина. — М.: НТ Пресс, 2005. — 312 с.: ил.

10. *Alexei White*. Major JavaScript Engines // JavaScript Programmer's Reference. — Indianapolis, IN 46256: Wiley Publishing, Inc., 2009. — P. 12—13. — (Programmer's Reference).
11. Alter S. L. Decision support systems: current practice and continuing challenges. Reading, Mass.: Addison-Wesley Pub., 1980.
12. ГОСТ Р ИСО МЭК ТО 10032-2007: Еталонна модель управління даними (ідентичний ISO/IEC TR 10032:2003 Information technology - Reference model of data management)
13. Коннолли Т., Бегг К., 2003.
14. Архітектура системи підтримки прийняття рішень у медицині, що ґрунтується на комбінованому вирішальному правилі - Бурцев, Максим Володимирович
15. Слюсар, В.І. (2018). Методологія ідентифікації критичних вимог до ОБТ.

ДОДАТКИ

Додаток А

```
//проверка есть ли у нашей компании
function algoritm(info) {

    for (let i = 0; i < priseListCompanies[0].priceList.length; i++) {
        //есть ли в нашей компании это животное
        if (info.animal == priseListCompanies[0].priceList[i].animal) {
            foodList(priseListCompanies[0].priceList, info, "my")
            break
        }
        else if (i == priseListCompanies[0].priceList.length - 1) {
            searchOtherCompaines(info)
        }
    }

}

//проверка есть ли у других компаний
function searchOtherCompaines(info) {

    var allPrice = []
    for (let i = 0; i < priseListOtherCompanies.length; i++) {

        //создаем список со всеми другими компаниями для такого животного
        for (let j = 0; j < priseListOtherCompanies[i].priceList.length; j++) {

            allPrice[allPrice.length] = {
                nameCompanies:           priseListOtherCompanies[i].nameCompanies,
                ...priseListOtherCompanies[i].priceList[j]
            }

        }

    }

    for (let i = 0; i < allPrice.length; i++) {
        if (info.animal == allPrice[i].animal) {
            foodList(allPrice, info, "other")
            break
        }
        else if (i == allPrice.length - 1) {
            alert("В нашій системі не існує корму для такої тварини")
        }
    }

}

//сортировка списка продуктов
function foodList(arr, userInfo, type) {

    var newListArr = []
    for (let i = 0; i < arr.length; i++) {
        if (arr[i].animal == userInfo.animal) {
```

```

        newListArr[newListArr.length] = arr[i]
    }
}

newListArr.sort((a, b) => a.price - b.price)

amountFood(newListArr, userInfo, type)

}

// перевірка хватит ли у нас вообще такого количества
function amountFood(newListArr, userInfo, type) {

    var amountAnimalFoor

    for (let i = 0; i < animal.length; i++) {
        if (animal[i].nameUkr == userInfo.animal) {
            amountAnimalFoor = userInfo.amount * animal[i].food
            break
        }
    }
    if (type == "my") {
        for (let i = 0; i < newListArr.length; i++) {
            if (newListArr[i].amount >= amountAnimalFoor) {
                searchAmountWhoCan(newListArr, type, amountAnimalFoor,
userInfo.animal, userInfo.amount)

                break
            } else {
                searchAmountWhoCan(newListArr, amountAnimalFoor, userInfo.animal, userInfo.amount, false)
                searchOtherCompaines(userInfo)
                break
            }
        }
    } else {

        for (let i = 0; i < newListArr.length; i++) {
            if (newListArr[i].amount >= amountAnimalFoor) {
                searchAmountWhoCan(newListArr, type, amountAnimalFoor,
userInfo.animal, userInfo.amount)
                break
            }
            else if (i == newListArr.length - 1) {
                alert(`Hi в нашій, ні в інших компаніях не має такої кількості
кормів! Необхідно: ${amountAnimalFoor} кг для ${userInfo.animal}`)

                for (let k = 0; k < newListArr.length; k++) {
                    if (newListArr[k].amount < amountAnimalFoor) {
                        searchAmountWhoCan(newListArr, type, amountAnimalFoor,
userInfo.animal, userInfo.amount, "typeOverAmount")
                        break
                    }
                }

                break
            }
        }
    }
}
}

```

```

}

// оставляем список той компании и того продукта которой хватит для нас корма
function searchAmountWhoCan(arr, type, amountAnimalFoor, animal, amount,
typeOverAmount) {
    var newFinalArr = []

    if (typeOverAmount == "typeOverAmount") {
        for (let i = 0; i < arr.length; i++) {
            if (arr[i].amount < amountAnimalFoor) {
                newFinalArr[newFinalArr.length] = arr[i]
            }
        }
        divNewListPrice(newFinalArr, type, amountAnimalFoor, animal, amount,
typeOverAmount)
    }
    else {
        for (let i = 0; i < arr.length; i++) {
            if (arr[i].amount >= amountAnimalFoor) {
                newFinalArr[newFinalArr.length] = arr[i]
            }
        }
        if (type == "thereIsButLittle") {
            divThereIsButLittle(newFinalArr, amountAnimalFoor, animal, amount)
        }
        else divNewListPrice(newFinalArr, type, amountAnimalFoor, animal,
amount)
    }
}
}

```