

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**Національний університет кораблебудування імені адмірала Макарова****Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій****"Допущений до захисту"****Завідувач кафедри ІУСТ**

к.т.н, доц. Михелєв І.Л.**" ____ " _____ 2023 р.****КВАЛІФІКАЦІЙНА РОБОТА****на здобуття ступеня вищої освіти "магістр"****на тему: "Аналіз моделей управління підприємствами та проектування
інформаційної системи для зоопарку"****Виконав: студент групи 6141 м**

Васильченко А.С.
(підпис)**Керівник роботи:**

доцент кафедри ІУСТ, к.т.н.
(посада, науковий ступінь вчене звання)

Гайдаєнко О.В.
(підпис)**м. Миколаїв – 2023 р.**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами

Кафедра інформаційних управляючих систем та технологій

Спеціальність 122 "Комп'ютерні науки"

Освітня програма "Комп'ютерні науки"

"ЗАТВЕРДЖУЮ"

Гарант освітньої програми

к.т.н, доц. Ажищев В.Ф.

" " 2023 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття ступеня вищої освіти "магістр"

Студенту Васильченку Андрію Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи: "Аналіз моделей управління підприємствами та проектування інформаційної системи для зоопарку"

Керівник роботи: Гайдаєнко Оксана Володимирівна

Затверджені наказом ректора № 1098-уч. від "18" вересня 2023 року.

2. Термін подання роботи: 15 грудня 2023 р.

3. Вихідні дані до проекту (роботи): ДСТУ щодо обробки інформації, літературні джерела, технічна документація на існуючі аналоги систем, матеріали наукового стажування

4. Перелік питань, що належать до розробки (найменування розділів): Вступ. 1. Розробка концепції інформаційної системи "Зоопарк". 2. Проектування інформаційної системи "Зоопарк". 3. Реалізація проекту інформаційної системи "Зоопарк". 4. Техніко-економічне обґрунтування розробки. 5. Охорона праці. 6. Охорона навколишнього середовища. Висновки.

5. Перелік презентаційних матеріалів: _____

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Гайдаєнко О.В., доцент		
2	Гайдаєнко О.В., доцент		
3	Гайдаєнко О.В., доцент		
4	Гайдаєнко О.В., доцент		
5	Гайдаєнко О.В., доцент		
6	Гайдаєнко О.В., доцент		

7. Дата видачі завдання 05 вересня 2022 р.**КАЛЕНДАРНИЙ ПЛАН**

№з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів роботи	Примітка
1	Розробка змісту дипломного проекту		
2	Дослідження моделей управління підприємствами		
3	Розробка концепції інформаційної системи		
4	Проектування інформаційної системи		
5	Реалізація проекту інформаційної системи		
5	Техніко-економічне обґрунтування розробки		
6	Охорона праці		
7	Охорона навколишнього середовища		
8	Вступ, висновки, список використаної літератури, додатки		
9	Презентація		

Студент

(підпис)

Васильченко А.С.

(ПІБ)

Керівник роботи

(підпис)

Гайдаєнко О.В.

(ПІБ)

АНОТАЦІЯ

Дипломний проект на тему: “Аналіз моделей управління підприємствами та проектування інформаційної системи для зоопарку” має об’єм 124 сторінок, містить розділи – 6, рисунки – 20, використані джерела – 41.

Мета проекту полягає у підвищенні ефективності управління діяльністю зоопарку за рахунок впровадження сучасних систем, які дозволять мінімізувати витрати та збільшити прибуток.

Завдання проекту полягає у дослідженні моделей управління підприємствами та розробці інформаційної системи, яка забезпечить автоматизацію процесів управління комунальною установою, покращить якість управління зоопарком та взаємодію між працівниками.

У першому розділі зроблений опис об’єкту дослідження і аналіз предметної області, сформовані вимоги, на основі яких розроблена концепція інформаційної системи.

У другому розділі розроблений технічний проект системи. Представлені загальносистемні рішення для вибору типових програмних засобів. Спроектовано базу даних. Описані алгоритми роботи проекту. Визначені програмні і технічні засоби реалізації і експлуатації проекту.

У третьому розділі розроблена робоча документація та проведена апробація проекту, яка показала коректну роботу усіх програмних модулів інформаційної системи.

У четвертому розділі наведено техніко-економічне обґрунтування розробки інформаційної системи.

У п’ятому розділі представлені правила та рекомендації щодо безпеки відвідувачів та працівників у зоопарку, а також правила безпеки за комп’ютером.

У шостому розділі представлені правила охорони навколишнього середовища.

Актуальність та новизна роботи полягає у вдосконаленні системи управління зоопарком для ефективної роботи за рахунок впровадження сучасних інформаційних технологій та комп'ютерної техніки.

Ключові слова: зоопарк, проектування, інформаційна система, дослідження.

ABSTRACT

The diploma project on the topic: "Analysis of enterprise management models and design of an information system for a zoo" has a volume of 124 pages, contains 6 chapters, 20 figures, 41 used sources.

The goal of the project is to increase efficiency through the implementation of modern systems that will minimize costs and increase profits.

The task of the project is to study enterprise management models and develop an information system that will ensure the automation of municipal institution management processes, improve the quality of zoo management and interaction between employees.

In the first chapter, the description of the research object and the analysis of the subject area are made, the requirements are formed, on the basis of which the concept of the information system is developed.

The technical design of the system is developed in the second chapter. System-wide solutions for choosing typical software tools are presented. The database is designed. Algorithms of the project are described. The software and technical means of implementation and operation of the project are defined.

In the third section, working documentation was developed and project testing was carried out, which showed the correct operation of all software modules of the information system.

The fourth chapter provides the technical and economic rationale for the development of the information system.

The fifth chapter presents rules and recommendations for the safety of visitors and workers in the zoo, as well as computer security rules.

The sixth chapter presents the rules of environmental protection.

The relevance and novelty of the work lies in the improvement of the zoo management system for effective work due to the introduction of modern information technologies and computer equipment.

Keywords: zoo, design, information system, research.

ЗМІСТ

ВСТУП.....	10
РОЗДІЛ 1. РОЗРОБКА КОНЦЕПЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ	
“ЗООПАРК”	11
1.1. Дослідження моделей управління комунальною установою	11
1.1.1. Поняття моделі управління.....	11
1.1.2. Історія становлення та розвитку менеджменту	12
1.1.3. Класифікація моделей управління	20
1.2. Опис об'єкту та аналіз предметної області.....	26
1.2.1. Історія та розвиток зоопарків	26
1.2.2. Аналіз предметної області	31
1.3. Розробка концепції інформаційної системи “зоопарк”	38
1.3.1. Вимоги до ІС зоопарку	38
1.3.2. Розробка концепції інформаційної системи	40
1.4. Постановка задачі	44
Висновки до розділу	45
РОЗДІЛ 2. ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ “ЗООПАРК”	46
2.1. Загальносистемні рішення	46
2.2. Рішення по інформаційному забезпеченню	52
2.2.1. Розробка концептуальної моделі даних	52
2.2.2. Розробка логічної моделі даних	53
2.2.3. Розробка фізичної моделі даних.....	54
2.3. Рішення з математичного забезпечення	56
2.4. Рішення з програмного забезпечення	62
2.5. Рішення з технічного забезпечення	70
Висновки до розділу	71
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОЕКТУ ІНФОРМАЦІЙНОЇ СИСТЕМИ	
“ЗООПАРК”	72

Висновки до розділу	75
РОЗДІЛ 4. ТЕХНІКО-ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ РОЗРОБКИ.....	77
Висновки до розділу	78
РОЗДІЛ 5. ОХОРОНА ПРАЦІ	79
Висновки до розділу	88
РОЗДІЛ 6. ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА.....	89
Висновки до розділу	92
ВИСНОВКИ	93
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	94
ДОДАТКИ.....	97
Додаток А – Інструкція користувача	97
Додаток Б – Текст програми	101
Додаток В – SQL код таблиць.....	117

ВСТУП

Темою дипломної роботи є розробка інформаційної системи управління зоопарком. Метою роботи – підвищення ефективності управління діяльністю зоопарком за рахунок впровадження сучасних систем, які дозволять мінімізувати витрати та збільшити прибуток.

У першому розділі проведено дослідження моделей управління підприємствами. Сформовані вимоги до інформаційної системи, на основі яких розроблена концепція інформаційної системи. Проведено обстеження підприємства, проведено аналіз інформаційних процесів, сформовано вимоги до проектованої системи, визначено цілі та завдання проекту.

У другому розділі розроблений технічний проект системи. Наведені загальносистемні рішення по вибору типових програмних засобів, спроектовано базу даних і представлені концептуальна модель, логічна модель, фізична модель. Визначені математичні алгоритми для рішення задач. А також визначені технічні і програмні засоби які будуть експлуатуватись системою.

У третьому розділі проведена розробка робочої документації.

У четвертому розділі наведено техніко-економічне обґрунтування розробки інформаційної системи.

У п'ятому розділі представлені правила та рекомендації щодо безпеки відвідувачів та працівників у зоопарку, а також правила безпеки за комп'ютером.

У шостому розділі представлені правила охорони навколишнього середовища.

Актуальність та новизна роботи полягає у вдосконаленні системи управління зоопарком для ефективної роботи за рахунок впровадження сучасних інформаційних технологій та комп'ютерної техніки.

РОЗДІЛ 1. РОЗРОБКА КОНЦЕПЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ “ЗООПАРК”

1.1. Дослідження моделей управління комунальною установою

1.1.1. Поняття моделі управління

Модель управління — це вибір, зроблений вищим керівництвом компанії щодо того, як вони визначають цілі, мотивують зусилля, координують діяльність і розподіляють ресурси; іншими словами, як вони визначають роботу менеджменту. Натхненні змінами в очікуваннях своїх співробітників, новими технологічними можливостями та пропозиціями нових конкурентів, деякі компанії виявляють, що особлива модель управління може сама по собі стати ключовим фактором їх конкурентоспроможності [1].

Вибір моделі управління є важливим рішенням для будь-якого підприємства. Від нього залежить ефективність роботи компанії, її конкурентоспроможність і можливо виживання.

У сучасному бізнесі моделі управління постійно розвиваються. Це пов'язано з рядом факторів, зокрема:

- Змінами в очікуваннях працівників: сучасні співробітники хочуть працювати в компанії, яка пропонує їм цікаву роботу, можливості для розвитку та справедливу винагороду.
- Новими технологічними можливостями: технології дозволяють компаніям автоматизувати процеси управління, що робить їх більш ефективними.
- Пропозиціями нових конкурентів: компанії, які використовують інноваційні моделі управління, можуть отримати конкурентні переваги.

Компанії, які прагнуть до успіху, повинні постійно аналізувати свої потреби та виклики, а також нові тенденції в управлінні. Це допоможе їм вибрати модель управління, яка буде найбільш ефективною для них.

1.1.2. Історія становлення та розвитку менеджменту

Концепція менеджменту існує вже тисячі років. За словами Піндура, Роджерса та Кіма, елементарні підходи до менеджменту сягають принаймні 3000 років до Різдва Христового, часу, коли записи про ділові операції були вперше записані близькосхідними священиками.

Сократ, близько 400 року до нашої ери, заявив, що управління — це компетенція, яка чітко відрізняється від володіння технічними навичками та знаннями. Римляни, відомі своїми легіонами воїнів на чолі з центуріонами, забезпечували відповідальність через ієрархію влади. Римо-католицька церква була організована за певними територіями, системою командування та посадовими інструкціями.

Протягом середньовіччя, приблизно з 476 р. до 1450 р., існувала ієрархія контролю та влади, подібна до католицької церкви, в якій влада належала майстрам і переходила до підмайстрів. Ці ремісники, були невеликими підприємствами, які виробляли продукцію різного ступеня якості, показували низькі показники продуктивності та потребували невеликого управлінського контролю, окрім контролю власника чи майстра.

Промислова революція, яка тривала з кінця 1700-х до 1800-х років, була періодом великих потрясінь і масштабних змін у способі життя та праці людей. До цього часу більшість людей заробляли на життя сільським господарством або роботою та проживали в сільських громадах. З винаходом парового двигуна з'явилися численні інновації, зокрема автоматизоване переміщення вугілля з підземних шахт, живлення заводів, які тепер масово виробляли товари, які раніше виготовлялися вручну, і залізничні локомотиви, які могли переміщувати продукти та матеріали між країнами вчасно та ефективним способом. Фабрики потребували працівників, які, у свою чергу, потребували керівництва та організації. Оскільки ці

об'єкти ставали більш значними та продуктивними, потреба в управлінні та координації стала важливим фактором.

На початку 1900-х років майстри збирали автомобілі, які модифікували компоненти відповідно до свого продукту. З появою стандартизованих деталей у 1908 році, а потім революційною складальною лінією Ford, запровадженою в 1913 році, час, необхідний для виготовлення моделі, скоротився з днів до кількох годин. З управлінської точки зору, для виготовлення автомобілів більше не були потрібні кваліфіковані майстри. Використання дешевої робочої сили та збільшення виробництва завдяки переміщенню виробничих ліній вимагали керівництва та управління цими масивними операціями. Щоб скористатися перевагами нових технологій, потрібен був інший підхід до організаційної структури та управління.

З появою нових технологій виникли вимоги до підвищення продуктивності та ефективності. Бажання зрозуміти, як найкраще вести бізнес, зосередилося на ідеї робочих процесів. Тобто менеджери хотіли вивчити, як виконується робота та вплив на продуктивність. Ідея полягала в тому, щоб оптимізувати спосіб виконання роботи. Одним із головних архітекторів вимірювання продуктивності людини був Фредерік Тейлор. Тейлор вважав, що головними цілями менеджменту є підвищення ефективності та скорочення витрат. Тейлор провів дослідження, щоб визначити, скільки одиниць може виготовити робітник за стільки хвилин. Він використовував секундомір, ваги для вимірювання ваги та рулетку, щоб обчислити, на яку відстань рухалися матеріали та скільки кроків робітники зробили для виконання своїх завдань.

Крім Тейлора, інші теоретики менеджменту, зокрема Френк і Ліліан Гілбрет, Гаррінгтон Емерсон та інші, розширили концепцію управлінського міркування з метою ефективності та послідовності для оптимізації результату. Не мало різниці,

чи виробляє організація автомобілі, видобуває вугілля чи сталь, метою було найефективніше використання праці для максимізації продуктивності.

Почала виникати потреба керувати не лише продуктивністю працівників, але й об'єднати всю організацію до спільної мети. Керівництво мусило організовувати численні складні процеси для все більших галузей. Французу Анрі Файолю приписують розробку управлінських концепцій планування, організації, координації, командування та контролю, які були попередниками сучасних чотирьох основних принципів управління: планування, організації, керівництва та контролю.

Зі збільшенням попиту на виробництво, викликаним науковими вимірюваннями, конфлікт між працею та керівництвом був неминучим. Відділ кадрів, з'явився як спосіб уповільнити попит на профспілки, започаткувати навчальні програми для зменшення плинності кадрів і визнати потреби працівників за межами виробництва. Ідея про те, що для підвищення продуктивності керівництво має врахувати потреби своїх працівників, розробляючи роботу, яка була б цікавою та корисною, з'явилася на сцені і стала частиною управлінського мислення.

Багато теоретиків менеджменту почали розглядати людський фактор. Двома гігантами, яким приписують рух думки менеджменту в напрямку розуміння потреб працівників, були Дуглас МакГрегор і Фредерік Герцберг. Х фактор теорії МакГрегора полягав у припущенні керівництва, що працівники не люблять роботу, ліниві, не мають самомотивації, і тому їх потрібно переконувати погрозами, покараннями чи залякуванням докласти належних зусиль. Y фактор його теорії був протилежним. МакГрегор вважав, що роботою керівництва є розвиток роботи, яка давала б працівникам відчуття самореалізації та цінності. Він стверджував, що за допомогою більш освічених методів управління, включаючи

встановлення чітких цілей для працівників і надання їм свободи для досягнення цих цілей, цілі організації та цілі співробітників можуть бути досягнуті одночасно.

Своєю теорією мотивації працівників Фредерік Герцберг значно доповнив погляди керівництва на поведінку працівників. Герцберг стверджував, що більшість мотиваційних зусиль керівництва, включаючи підвищення заробітної плати, покращення пільг і збільшення часу відпустки, зрештою зазнали невдачі, оскільки, хоча вони можуть зменшити певні фактори незадоволеності роботою (те, що працівникам не подобалося в їхній роботі), вони не підвищили задоволеність роботою. Герцберг вважав, що це дві абсолютно різні проблеми управління. Одним із незмінних результатів роботи Герцберга стала ідея про те, що керівництво може позитивно впливати на задоволеність працівників роботою, що, у свою чергу, допомагає досягти цілей і завдань організації.

Концепція МакГрегора, Герцберга та багатьох інших теоретиків менеджменту полягала в тому, щоб досягти ефективності управління шляхом більш ефективного використання людей. Попередні теорії менеджменту щодо мотивації працівників (вважалося, що вона безпосередньо пов'язана зі збільшенням продуктивності) наголошували на контролі, спеціалізованих роботах і мало приділяли уваги внутрішнім потребам працівників. Ідеї, які враховували людський фактор на основі теорій із психології, тепер стали частиною управлінського мислення. Організаційні зміни, запропоновані мислителями менеджменту, які бачили прямий зв'язок між покращеним дизайном роботи, самореалізацією та складною роботою, почали закріплюватись у більш освіченій теорії менеджменту.

Кунц і О'Доннелл визначили менеджмент як «функцію виконувати завдання за допомогою інших. Одна владна фігура стояла над усіма іншими, і її вважають батьком сучасного менеджменту. Цією людиною був Пітер Друкер. Друкеру, письменнику, педагогу та консультанту з менеджменту, широко приписують

розробку концепції управління за цілями або МВО. Управління за цілями – це процес визначення конкретних цілей, необхідних для досягнення цілей організації. Краса концепції МВО полягала в тому, що вона давала працівникам чітке уявлення про цілі своєї організації та визначала їхні індивідуальні обов’язки.

Внесок Друкера в сучасне управлінське мислення вийшов далеко за межі концепції МВО. Упродовж свого довгого життя Друкер стверджував, що єдина роль бізнесу полягає в тому, щоб створити клієнта, а маркетинг та інновації — це дві його важливі функції. Розглянемо Apple iPhone. Завдяки одній інновації створено тисячі робочих місць на виробничих підприємствах, продажі iPhone у магазинах по всьому світу, а прибуток повернувся до Apple, що дало їм змогу продовжувати інноваційний процес. Інше постійне спостереження Друкера полягало в тому, що надто багато компаній не спромоглися поставити питання «в якому бізнесі ми працюємо?». Неодноразово компанія зазнавала невдач, навіть припиняла роботу, не визнаючи, що її галузь змінюється, або намагалася вийти на нові ринки за межі своєї основної компетенції.

Теорії менеджменту продовжували розвиватися з додатковими концепціями, висунутими іншими інноваційними мислителями. Генрі Мінтцберга пам’ятають тим, що він роздув ідею про те, що керівники — це знакові особи, які відпочивають у своїх офісах, сидять склавши руки та обмірковують масштабні ідеї. Мінцберг зауважив, що управління було важкою роботою. Менеджери були в русі, відвідуючи наради, керуючи кризами та взаємодіючи з внутрішніми та зовнішніми контактами. Крім того, залежно від точного характеру своєї ролі, менеджери виконували численні обов’язки, включаючи обов’язки речника, лідера, розподільовача ресурсів і учасника переговорів. У 1970-х роках Том Пітерс і Роберт Вотерман подорожували по всьому світу, досліджуючи сучасні найкращі практики менеджменту того часу. У їхній книзі «У пошуках досконалості» описано, що працює з точки зору управління організаціями. Можливо, найбільш

відповідним відкриттям було їхнє твердження про те, що культура має значення. Вони виявили, що найкраще керовані компанії мають культуру, яка сприяє прозорості, відкритому обміну інформацією та ефективному керуванню комунікацією вгору та вниз по організаційній ієрархії. Добре керовані компанії, які виявили Петерсон і Вотерман, були побудовані значною мірою на попередніх управлінських ідеях Макгрегора та Герцберга. Першокласні організації досягли успіху, забезпечуючи змістовну роботу та позитивне підтвердження цінності своїх співробітників.

Інші зробили значний внесок у сучасне управлінське мислення. «Сім звичок дуже успішних людей» Стівена Кові, «П'ята дисципліна» Пітера Сенга та «Створений, щоб довго» Джима Коллінза та Джеррі Порраса входять до пантеону бестселерів про принципи управління. Серед знакових мислителів цієї епохи був Майкл Портер. Портеру, професору Гарвардської бізнес-школи, вважають, що він підняв концепцію стратегічного мислення на інший рівень. Портер розглядав питання про те, як організації можуть ефективно конкурувати та досягти довгострокової конкурентної переваги. Він стверджував, що фірма може отримати таку перевагу лише трьома способами: 1) лідерство на основі витрат – стати виробником з найменшими витратами, 2) лідерство з доданою вартістю – пропонувати диференційований продукт або послугу, за яку клієнт готовий платити високу ціну та 3) фокус – конкурувати в ніші ринку з лазерною фіксацією. Роздуми Портера про конкуренцію та конкурентну перевагу стали позачасовими принципами стратегічного менеджменту, які використовуються й сьогодні. Мабуть, найважливішим внеском Портера в сучасне управлінське мислення є зв'язок між вибором фірмою стратегії та її фінансовими показниками. Якщо організація не в змозі вибрати та належним чином реалізувати одну з трьох основних стратегій, вона стикається з серйозною небезпекою застрягти посередині: її ціни занадто високі, щоб конкурувати на основі ціни, або її

продукти не мають достатньо унікальних функцій, щоб спонукати клієнтів платити преміальна ціна.

Менеджери 21-го століття повинні протистояти викликам, які їхні колеги навіть кілька років тому навряд чи могли собі уявити. Постійно зростаюча хвиля технологій, вплив штучного інтелекту, еволюція глобалізації та перетягування каната між інтересами зацікавлених сторін та акціонерів компанії є головними серед вимог, з якими стикаються сучасні менеджери.

Про експоненціальне зростання технологій написано багато. Повідомлялося, що сьогоdnішній iPhone має більш ніж у 100 000 разів більшу обчислювальну потужність, ніж комп'ютер, який допоміг висадити людину на Місяць.

Керівництву сьогодні доводиться боротися з величезною кількістю даних, які тепер доступні для полегшення бізнес-рішень. Аналітика даних, вивчення наборів даних, надає інформацію, яка допомагає менеджерам краще зрозуміти поведінку клієнтів, їхні бажання та потреби, персоналізувати доставку маркетингових повідомлень і відстежувати відвідування онлайн-веб-сайтів. Розвиток розуміння того, як використовувати аналітику даних, не загрузнувши, стане серйозним викликом для менеджера 21 століття. Збір, систематизація, використання даних у логічний, своєчасний та економічно ефективний спосіб створює абсолютно нову парадигму управлінської компетентності. На додаток до аналізу даних, кібербезпека, дрони та віртуальна реальність є новими захоплюючими технологіями, які пропонують безпрецедентні зміни у способах ведення бізнесу. Кожна з цих можливостей вимагає нового рівня управлінської компетентності, що, у свою чергу, створює можливості для сучасного менеджера.

Чи замінять роботи робітників? Правда, це певною мірою вже сталося в багатьох галузях. ШІ має здатність усувати повсякденні завдання та надає можливість менеджерам зосередитися на суті своєї роботи. Людські навички, такі як емпатія, навчання та коучинг співробітників, зосередження на розвитку людей і

звільнення часу для творчого мислення, ставатимуть дедалі важливішими, оскільки ШІ продовжує розвиватися як критично важливий інструмент для сучасного менеджера.

Глобалізація визначається як взаємозалежність світових економік і неухильно розвивається після закінчення Другої світової війни. Все більше країн виходять із рангів, країн що розвиваються, і сприяють зростанню середнього класу споживачів. Цей зростаючий новий клас має купівельну спроможність купувати товари та послуги, які раніше були недоступні, і компанії по всьому світу вийшли за межі своїх національних кордонів, щоб задовольнити ці потреби. Управління в епоху глобалізації принесло нові виклики. Адаптація до нових культур, орієнтування в головоломці різних законів, тарифів, правил імпорту/експорту, питань людських ресурсів, логістики, маркетингових повідомлень, управління ланцюгом постачання, валюти, іноземних інвестицій та державного втручання є одними з вимог, які стоять перед глобальним менеджером 21-го століття. . Незважаючи на ці величезні виклики, торгівля між країнами світу зростає безпрецедентними темпами. Світова торгівля підскочила з приблизно 20% світового ВВП у 1960 році до майже 60% у 2017 році.

Незважаючи на приголомшливе зростання, глобалізація має долю критиків. Головна з них полягає в тому, що глобалізація посилила диспропорцію між заможними та бідними в суспільстві. Противники глобалізації стверджують, що в багатьох випадках робочі місця втрачали країни, що розвиваються, з нижчими переважаючими ставками заробітної плати. Крім того, нерівність посилилася, оскільки найбагатші споживають непропорційно великий відсоток світових ресурсів. Прихильники заперечують, що на макрорівні глобалізація створює більше робочих місць, ніж втрачається, більше людей позбавляються бідності, а глобальне розширення дозволяє компаніям стати більш конкурентоспроможними на світовій арені.

Традиційні перешкоди для розширення діяльності за межі рідної країни, а також безліч нових труднощів, таких як незаплановані торговельні бар'єри, заблоковані придбання та посилений контроль з боку регуляторів, додали тягаря управління в 21 столітті. Сцена створена для нового покоління менеджерів, які мають навички роботи з цим новим, складним бізнес-середовищем. У 20-му столітті стара командно-адміністративна модель управління могла спрацювати. Однак сьогодні, з технологіями, штучним інтелектом, глобалізацією, націоналізмом і багатьма іншими перешкодами, організації продовжуватимуть рух до більш плоскої, гнучкішої організаційної структури, якою керуватимуть менеджери з відповідними навичками 21 століття [2].

1.1.3. Класифікація моделей управління

Моделі управління можна класифікувати за такими групами:

Перша група класифікує моделі за походженням: мається на увазі, що існує класифікація моделей управління на основі їх походження, та що важливо витягувати елементи з успішних національних моделей управління.

Друга група класифікує моделі за характером реалізації владних повноважень: в основі класифікації яких лежить характер реалізації владних повноважень.

Третя група: базується на цільових управлінських установках.

Четверта група: ця група моделей найбільш точно відповідає концепції системності організації і що без урахування цього аспекту не може бути ефективного управління організацією в сучасних мінливих умовах зовнішнього та внутрішнього середовища.

На рисунку 1.1 наведена класифікація управлінських моделей [3].

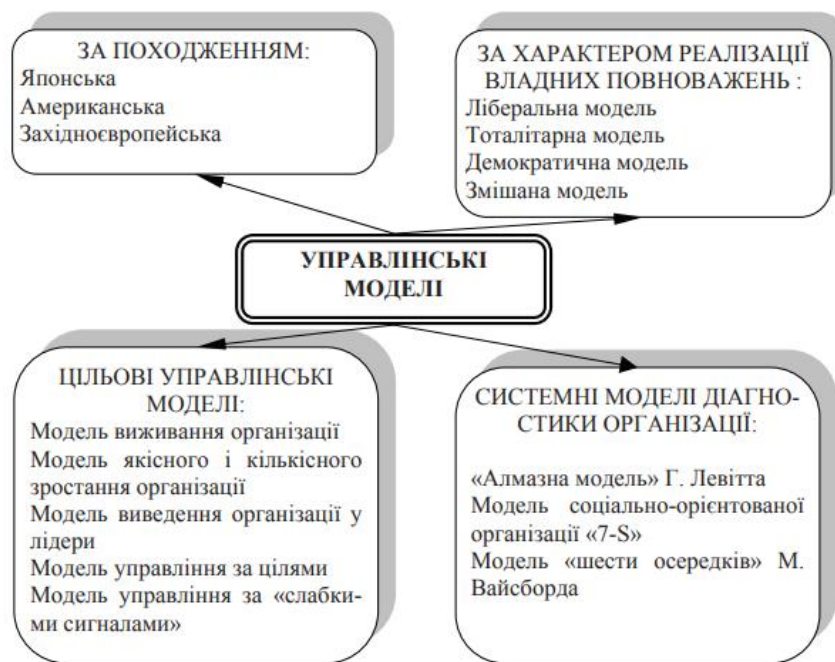


Рисунок 1.1 – Класифікація управлінських моделей

Розглянемо першу групу моделей.

Японська модель управління:

- Основана на філософії "Ми всі одна родина"
- Колективне прийняття рішень та відповідальність
- Неформальна організація контролю
- Уповільнена оцінка роботи працівника і службове зростання
- Орієнтація управління на групу
- Оцінка управління за досягненням гармонії у колективі і за колективним результатом.

Американська модель управління:

- Основана на принципі індивідуалізму
- Ставка робиться на яскраву особистість
- Індивідуальний характер прийняття рішень та відповідальність
- Суворо регламентована структура управління
- Індивідуальний контроль керівника

- Швидка оцінка результату праці, прискорене просування по службі.

Західноєвропейська модель управління:

- Поєднання американської та японської моделей
- Колективна робота команди на чолі з лідером

У цілому нині, японська модель управління спрямовано довгострокову перспективу та ефективність роботи колективу. Американська модель орієнтована на короткострокову перспективу та ефективність роботи окремих співробітників. Західноєвропейська модель поєднує переваги двох попередніх моделей.

Розглянемо другу групу моделей.

Ліберальна модель управління:

- Цінності: свобода, самоврядування
- Прийняття рішень: на основі консенсусу, за участю всіх зацікавлених сторін
- Умови: високий рівень відповідальності і дисципліни у працівників

Тоталітарна модель управління:

- Цінності: стабільність, надійність, колективна сила
- Прийняття рішень: в односторонньому порядку, без участі працівників
- Умови: жорстка ієрархія управління, чіткий поділ управлінських функцій

Демократична модель управління:

- Цінності: рівні права і можливості для всіх індивідів
- Прийняття рішень: більшістю голосів
- Умови: механізми, що забезпечують рівність можливостей усіх

Змішана модель управління:

- Цінності: надійні і стійкі засоби управління, рівність можливостей усіх

- Прийняття рішень: на основі перспективної рефлексії, делегування повноважень
- Умови: створення умов для позитивного саморозвитку організації і кожного її члена

В підсумку ліберальна модель управління приділяє увагу свободі та самоврядуванню, а прийняття рішень відбувається на основі консенсусу. Тоталітарна модель управління орієнтована на стабільність та надійність, з прийняттям рішень зверху донизу. Демократична модель управління підкреслює рівність прав та рішень на основі більшості голосів. Змішана модель управління поєднує індивідуальне самовираження, рівність можливостей та механізми делегації повноважень.

Розглянемо третю групу моделей.

Модель виживання організації:

- Основні дії: жорсткий контроль, підтримка відносин із зовнішнім оточенням, ідентифікація та нейтралізація загроз
- Ціль: забезпечення мінімально необхідного рівня функціонування організації

Модель якісного та кількісного зростання організації:

- Основні дії: збереження позицій, виділення ключових факторів успіху, коригування концепції управління, жорсткий контроль
- Ціль: зростання організації за обсягами, прибутковістю, конкурентоспроможністю

Модель виведення організації у лідери:

- Основні дії: цільовий пошук незайнятих ніш, моніторинг конкурентів, масштабне "вторгнення" у зовнішнє середовище
- Ціль: виведення організації у лідери на ринку

Модель управління за цілями:

- Основні дії: моніторинг діяльності, експертиза відповідності до "ідеального" стану, визначення сумісності обраного варіанту дій, вибудовування діяльності відповідно до "ідеального" стану
- Ціль: досягнення заданих цілей

Модель управління "за слабкими сигналами":

- Основні дії: відстеження "слабких сигналів", підготовка альтернативних рішень
- Ціль: своєчасна підготовка до змін у зовнішньому середовищі

Загалом модель забезпечення виживання орієнтована на збереженні мінімально необхідного для функціонування організації у нестабільному середовищі. Модель зростання прагне збільшити обсяги продажу, прибутковість та конкурентоспроможність організації. Модель лідерства спрямована на заняття лідируючої позиції на ринку. Модель управління за цілями спрямована на досягнення конкретних цілей, таких як збільшення прибутку, розширення ринків або впровадження нових технологій. Модель управління за "слабкими сигналами" орієнтована на готовність до адаптації до змін у зовнішньому середовищі.

Розглянемо четверту групу моделей.

«Алмазна модель» Левітта — модель управління, яка розглядає організацію як систему, що складається з чотирьох взаємопов'язаних компонентів: завдання, технологія, структура, персонал. Зміна одного компонента може призвести до необхідності зміни інших компонентів. Модель дозволяє аналізувати взаємозв'язки між компонентами організації та їхній вплив на ефективність організації.

Модель соціально-орієнтованої організації «7-S» розглядає організацію як систему, що складається із семи взаємопов'язаних компонентів. Жорсткі це — структура, стратегія та системи вони знаходяться під впливом керівництва. М'які це — здібності, стиль, спільні цінності, працівники вони знаходяться під впливом

корпоративної культури. Де корпоративна культура це — складна та багатоаспектна система цінностей, переконань, ділових принципів, норм поведінки, традицій, що стає важливим нематеріальним ресурсом підприємства, оскільки забезпечує соціальні зв'язки, комунікативно інформаційне спілкування, гармонізує відносини між роботодавцями і найманими працівниками і таким чином суттєво впливає на ефективність та конкурентоспроможність підприємства[4].

Модель "шести осередків" М. Вайсборда заснована на ідеї, що організація є відкритою системою, побудована на принципах рівнозначності і взаємопов'язаності осередків. Модель складається з шести взаємопов'язаних компонентів:

- Мета - сенс існування організації, її соціальна, економічна та історична значимість.
- Структури - механізми поділу праці, розподілу роботи між працівниками і підрозділами.
- Взаємини - культура організації, система відносин між працівниками, механізми вирішення протиріч.
- Винагорода - система мотивації персоналу, привабливість виконання завдань для працівників.
- Допоміжні механізми - методи координації діяльності співробітників і підрозділів, а також адаптації до змін у зовнішньому середовищі; системи і процедури управління.
- Лідерство - те, що дозволяє утримувати всі осередки в рівновазі [40].

Для інформаційної системи “зоопарк” найкращою є змішана модель управління, яка поєднує в собі елементи різних моделей, а саме поєднання японської моделі управління з ліберальною моделлю. Японська модель управління характеризується такими аспектами як довгострокова перспектива та ефективність

роботи колективу, Ліберальна модель управління — свобода та самоврядування і прийняття рішень на основі консенсусу. Вона дозволяє зоопарку досягти своїх цілей з урахуванням конкретних умов діяльності.

1.2. Опис об'єкту та аналіз предметної області

1.2.1. Історія та розвиток зоопарків

Історія зоопарків охоплює багатовіковий період взаємодії людства з тваринним світом. Від давніх часів до сучасності зоопарки відіграли важливу роль у збереженні біорізноманіття, наукових дослідженнях та освіті.

Стародавні цивілізації та їхні звіринці (2500 р. до н. е.)

Найдавніші зоопарки можна простежити до стародавніх цивілізацій, де правителі збирали колекції екзотичних тварин, щоб продемонструвати своє багатство та могутність. У Єгипті фараони тримали диких істот у садах, а ієрогліфи зображують колекцію диких звірів з усієї імперії.

Подібним чином у стародавньому Китаї імператори утримували великі звіринці як частину своїх палацових комплексів, демонструючи своє панування над природою.

Грецькі та римські колекції тварин (500 р. до н. е. – 500 р. н. е.)

Греки та римляни продовжили цю традицію, де тварини відігравали центральну роль у їхній культурі та міфології. Зоопарки цієї епохи були не лише демонстрацією сили; вони також стосувалися розуміння тваринного світу.

Такі філософи, як Аристотель, вивчали тварин з науковою метою, і його праці свідчать про рудиментарний підхід до зоології.

Королівські звіринці як статусні символи (1200-ті роки)

У середні віки утримання екзотичних тварин стало символом королівських привілеїв. Звіринець Лондонського Тауера, заснований у 13 столітті, включав низку тварин від левів до слонів, яких часто отримували як подарунки від інших монархів і дослідників.

Знайомство публіки з екзотичною дикою природою (1500-ті роки)

До епохи Відродження ці колекції почали переходити від приватних курйозів до публічних видовищ. Звіринець сім'ї Медічі у Флоренції був одним із перших, хто дозволив відкритий доступ, готуючи основу для зоопарків, які згодом з'являться.

Просвітництво та розквіт наукових зоопарків (1752)

Епоха Просвітництва, період, відзначений сплеском інтелектуального та наукового прогресу, докорінно змінив спосіб взаємодії людей зі світом природи. Саме в цю епоху концепція зоопарку стала більше відповідати ідеалам вивчення та збереження, а не просто видовища.

Віденський зоопарк, заснований у 1752 році, був частиною цього руху, зосереджуючись на наукових дослідженнях і класифікації видів. Будучи найстарішим існуючим зоопарком у світі, він спочатку був частиною імператорського звіринця в палаці Шенбрунн, призначеного для наукових досліджень тваринного світу.

Зоопарк був піонером у русі спостереження за тваринами не як за диковиною, а як за об'єктами, гідними вивчення. Це полегшило класифікацію видів, завдання, яке ставало дедалі важливішим, оскільки дослідники та натуралісти приносили знання про нові істоти з усього світу. Наприклад, саме в цей час вчені з'ясували, що риби мають мозок; те, що здивувало багатьох людей у той час.

Поширення зоопарків у 19 столітті (1828)

У 19 столітті відбулося різке зростання кількості зоопарків. Відкриття Лондонського зоопарку в 1828 році ознаменувало значну еволюцію в концепції зоопарків. На відміну від своїх попередників, Лондонський зоопарк був заснований Зоологічним товариством Лондона з чітким науковим мандатом.

Спочатку він служив колекцією для наукових досліджень, доступною лише для членів товариства та їхніх гостей. Однак, визнаючи освітній потенціал, він незабаром відчинив свої двері для публіки.

Це означало перехід від ексклюзивного звіринця королівської влади та еліти до державних інституцій із ширшою суспільною роллю. Зоопарк став місцем, де зійшлися освіта, дослідження та відпочинок, що дозволило людям з усіх верств суспільства спілкуватися з тваринним світом і дізнаватися про нього. Це був час, коли багато міст Європи та Північної Америки почали засновувати власні зоопарки, часто натхненні моделлю Лондонського зоопарку.

Вікторіанська епоха і клітка (1840-90-ті)

Вікторіанська епоха характеризувалася утилітарним підходом до дизайну зоопарку, коли тварин тримали в клітках для зручності публіки. У цей період було побудовано багато найстаріших у світі зоопарків, але добробут тварин був другорядним по відношенню до експозиції. Загалом, 1840-90-ті роки стали жорстокою епохою в історії зоопарків.

Револуція корпусів: модель Хагенбека (початок 1900-х)

Карл Хагенбек, німецький торговець тваринами та шоумен, зробив революцію в дизайні зоопарку на рубежі 20 століття. Він представив просторі вольєри під відкритим небом із натуралістичними пейзажами, що різко контрастують із загратованими клітками минулого. Ця модель підкреслила добробут тварин і покращила враження від відвідувачів.

Світові війни: важкий час для зоопарків (1915-1945)

Світові війни принесли зоопаркам значні труднощі, багатьом із них важко було годувати та доглядати за своїми тваринами серед хаосу. Бомбардування та брак ресурсів призвели до втрати багатьох тварин, і деякі зоопарки були змушені закритися. Через світові війни тільки в європейських зоопарках загинуло більше 80% зоопарків; руйнування, якого ніколи не було за всю довгу історію зоопарків.

Післявоєнна ера: новий фокус на збереження (1940-50-ті роки)

Після війни зоопарки почали зміщувати свою увагу від простих виставок до збереження та освіти. Таким чином, це вперше в історії зоопарків поклато початок етичним міркуванням. Поява панд у західних зоопарках і подальший інтерес громадськості до цих тварин підкреслили потенційну роль зоопарків у глобальних зусиллях щодо збереження природи.

Програми розведення видів, що перебувають під загрозою зникнення (кінець 1900-х років)

Зоопарки стали заповідниками для видів, що знаходяться під загрозою зникнення, з програмами розведення, спрямованими на збереження генетичного різноманіття та зміцнення диких популяцій; новинка в історії зоопарків. Історій успіху багато, від відновлення каліфорнійського кондора до повторної інтродукції аравійського орікса.

Застосування технологій для збереження (21 століття)

Вперше в історії зоопарків сучасні зоопарки використовують інноваційні технології, від відстеження здоров'я тварин за допомогою складного програмного забезпечення до залучення відвідувачів за допомогою інтерактивних виставок і мобільних додатків.

Цифрові системи відстеження відстежують все, від споживання раціону до циклів розведення, дозволяючи точно керувати індивідуальним доглядом за тваринами. Ці системи також сприяють обміну цінними даними між глобальними природоохоронними мережами, допомагаючи в управлінні популяціями видів у всьому світі.

Технології віддаленого моніторингу дозволяють доглядачам зоопарку спостерігати за тваринами, не втручаючись у їх повсякденну діяльність, мінімізуючи стрес і створюючи більш натуралістичну модель поведінки. Крім того, генетичні бази даних і біобанки відіграють вирішальну роль у збереженні

біорізноманіття, надаючи ресурси для досліджень і потенційних майбутніх програм реінтродукції.

Що стосується відвідувачів, зоопарки запровадили інтерактивні технології для покращення освітнього охоплення. Мобільні додатки можуть надати мультимедійний тур, пропонуючи інформацію про тварин, стан збереження та ініціативи зоопарку одним дотиком екрана. Сьогодні ви можете встановити програму ідентифікатора тварин і дізнатися, що за тварина перед вами, просто використовуючи камеру свого телефону.

Інтерактивні та захоплюючі виставки

Використання доповненої та віртуальної реальності в зоопарках забезпечує відвідувачам захоплюючий досвід, який є освітнім і розважальним, не порушуючи життя тварин. Це новинка в історії зоопарків.

Дослідження AR можуть накладати цифрову інформацію на реальне середовище, дозволяючи відвідувачам бачити шари середовища проживання тварин або навіть імітувати взаємодію з тваринами контрольованим і безпечним способом. Наприклад, відвідувачі можуть використовувати пристрій AR, щоб спостерігати за віртуальним слоном, який блукає саваною, тоді як насправді слон комфортно відпочиває у своєму вольєрі.

Віртуальна реальність робить цей крок далі, занурюючи відвідувачів у повністю цифровий світ. За допомогою гарнітур віртуальної реальності люди можуть відчувати реалістичну симуляцію природного середовища з точки зору тварини або зануритися в підводні екосистеми, не намокнувши. Цей досвід може бути особливо потужним у передачі повідомлень про збереження, оскільки вони дозволяють відвідувачам стати свідками впливу людської діяльності на природні середовища існування у вісцеральний та вражаючий спосіб [5].

1.2.2. Аналіз предметної області

Зоологічні парки є закладами природоохоронного, культурно-освітнього спрямування, і їхньою головною метою є створення виставок рідкісних, екзотичних та місцевих видів тварин, збереження генофонду цих видів, дослідження дикої фауни та розробка наукових засад для їхнього утримання в неволі. Також вони виконують важливу роль у проведенні екологічної освіти та виховання серед населення. [6].

Для ефективної функціонування зоопарку необхідні фінансові ресурси, які можуть надходити від продажу білетів, фондів чи державних субсидій. У всіх цих випадках привертання відвідувачів є ключовим елементом. Якщо кількість відвідувачів буде недостатньою, дохід від продажу білетів може бути недостатнім, що може порушити фінансову стійкість зоопарку. Таким чином, для забезпечення ефективного фінансового функціонування та подальшого існування закладу, керівництво повинно приділяти велику увагу залученню достатньої кількості відвідувачів.

Для залучення відвідувачів можна використовувати такі кроки:

1. Організація заходів для дорослих;

Проведення подій, спрямованих виключно на дорослих, надає батькам можливість спокійно роздивитися всі експозиції та ознайомитися з інформацією про тварин. Це також приваблює тих, хто зазвичай не відвідує зоопарк, таких як молоді дорослі та особи без дітей.

2. Організація спеціальних сімейних заходів;

Пропозиція річних членств стимулює відвідувачів часто повертатися, надаючи їхнім родинам необмежений доступ до зоопарку. Це також може включати знижки на подарунки чи квитки на спеціальні заходи для членів.

3. Впровадження системи членських карточок;

Придбання річного членства спонукає відвідувачів часто повертатися до зоопарку, оскільки це надає їхнім родинам необмежений доступ до вашої організації. Пропуски зручні, оскільки сім'ї потенційно можуть пропускати черги за квитками, а членство дає їм стимул часто повертатися.

4.Надати можливість притулити вихованця або дати йому ім'я;

Збір коштів на приймання домашньої тварини або імен тварин — це чудовий спосіб отримати прибуток, одночасно заохочуючи людей брати участь у роботі у зоопарку.

Якщо у вас є тварина, у якого щойно народилася дитина, запропонуйте пропозицію імені за невелику плату або як заявку, щоб виграти членство в зоопарку чи іншу цінність зоопарку. Окрім зацікавлення самого зоопарку, ці можливості також можуть стимулювати трафік і взаємодію на вашому веб-сайті та сторінках у соціальних мережах.

5. Проведення спеціальних акцій та днів знижок;

Пропозиції спеціальних знижок, таких як військові знижки чи обмін членства на інші місцеві атракції, можуть збільшити відвідуваність, особливо в дні з низькою активністю відвідувачів.

6.Підкреслюйте практику збереження;

Зосередження на дослідженні та природоохоронних практиках допомагає залучити більше дорослих до зоопарку та підсилює їхній інтерес у відвідуванні.

7. Впровадження унікальних заходів.

Пропозиція унікальних вражень, таких як годування тварин чи екскурсії за лаштунками, може відокремити ваш зоопарк від конкурентів та збільшити інтерес відвідувачів. [8].

Організаційна структура управління зоопарком зображено на рисунку 1.2.

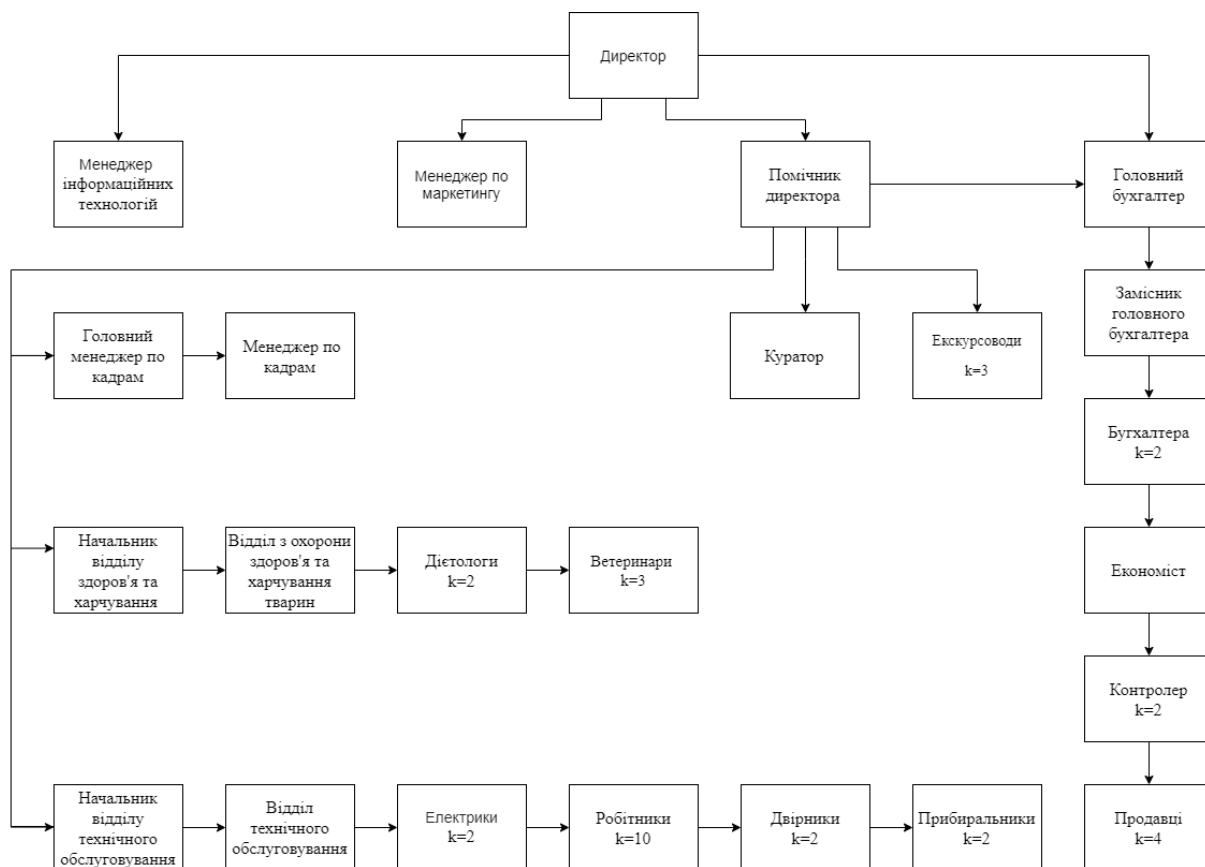


Рисунок 1.2 — Організаційна структури управління зоопарком

Директор зоопарку виступає ключовою фігурою у керівництві і відповідає за відслідковування всіх аспектів управління закладом. Ці обов'язки включають управління денною діяльністю парку, розробку бюджетів, впровадження політик, наймання адміністративного персоналу, пошук додаткових фінансових ресурсів та відстеження розвитку закладу. Директор також виконує роль головного представника зоопарку у відносинах із засобами масової інформації.

Для ефективної роботи, директор взаємодіє із начальниками відділів та куратором, які, у свою чергу, керують іншими працівниками зоопарку. Директор відповідає за те, щоб щоденні операції працювали беззауважно, а догляд за тваринами відбувався відповідно до всіх стандартів та правил. Йому доручено забезпечити ефективне функціонування зоопарку як організації та його подальший розвиток [7].

Під його безпосереднім керівництвом знаходяться помічник директора, менеджер по маркетингу, менеджер інформаційних технологій, головний бухгалтер. Кожен з цих фахівців відповідає за свою функціональну область та сприяє забезпеченню ефективного функціонування зоопарку в різних аспектах.

Розглянемо більш детально роботу кожного з відділів.

В той час коли директор фокусується на зовнішніх аспектах діяльності закладу, помічник директора приділяє увагу внутрішнім питанням. Він відповідає за направлення, керівництво та координацію внутрішньої операційної діяльності організації відповідно до політики, цілей і завдань, визначених директором [9]. Йому підпорядковуються головний менеджер по кадрах, начальник відділу здоров'я та харчування, головним бухгалтером, начальник відділу технічного обслуговування, екскурсоводи, куратор.

Головний менеджер по кадрах має на собі відповідальність за розробку та контроль загальних стратегій, систем, тактик і процедур у сфері управління персоналом в організації. Він також забезпечує дотримання законодавства в управлінні персоналом і представляє звіт перед керівництвом. У випадку його відсутності його обов'язки виконує менеджер з кадрів [16].

В обов'язки менеджера по кадрах входить: набір персоналу, відносини зі співробітниками, керувати процесом найму та відбору, підтримувати позитивну робочу атмосферу [16].

Відділом з охорони здоров'я та харчування тварин керує начальник відділу. Він забезпечує ефективне керівництво цим відділом. До його обов'язків входить надання відповідальності за всі аспекти роботи відділу, включаючи, але не обмежуючись цим, контроль за здоров'ям та харчуванням тварин. Начальник відділу також відповідає за виконання стандартів та протоколів, пов'язаних із забезпеченням оптимального стану здоров'я та харчування тварин у відділі. В даному відділі працюють: 2 ветеринари та 4 дієтологи.

Дієтологам доручено складати добре збалансовані харчові раціони для різних видів тварин, враховуючи їх різноманітні потреби в харчуванні. Їм необхідно оцінити харчові та калорійні потреби, виходячи зі стану тварини та рівня фізичної активності. Регулярне тестування та аналіз є важливими обов'язками, які дозволяють дієтологам точно визначити поживні речовини, які поглинають екзотичні тварини зі свого раціону. Згодом у міру необхідності вносяться корективи в раціон. Крім того, якість їжі має проходити ретельний контроль [14;15].

Ветеринари зоопарку відіграють вирішальну роль у підтримці здоров'я та благополуччя тварин, якими вони опікуються.

В їх обов'язки входить широкий спектр ветеринарних послуг, зокрема:

Лікування травм і хвороб: ветеринари зоопарку кваліфіковано діагностують і лікують широкий спектр хвороб тварин, від незначних травм до складних захворювань. Вони пропонують комплексні плани лікування, включаючи призначення ліків, проведення операцій і догляд за ранами.

Профілактична медична допомога: усвідомлюючи важливість профілактики, ветеринари зоопарку надають пріоритет профілактичним заходам для захисту здоров'я тварин. Вони проводять регулярні медичні огляди, роблять щеплення та проводять діагностичні тести, щоб на ранній стадії виявити потенційні проблеми зі здоров'ям.

Виконання різноманітних медичних процедур: ветеринари зоопарку мають устаткування для виконання різноманітних медичних процедур, взяття зразків крові та тканин, виконання ультразвуку та рентгенівського дослідження та чищення зубів. Їх досвід гарантує належний догляд за тваринами [13].

Головний бухгалтер несе відповідальність за представлення фінансової звітності організації керівництву. Йому підпорядковуються: 2 бухгалтера, економіст, 2 контролера, 4 продавці.

Заступник головного бухгалтера підтримує головного бухгалтера у створенні точних і своєчасних фінансових даних для всіх аспектів зоопарку. До обов'язків входить ведення бухгалтерського обліку, допомога у складанні бюджетів для різних відділів зоопарку, участь у написанні звітів і запитів у системі бухгалтерського обліку зоопарку та ведення бухгалтерських операцій протягом року та наприкінці року. У разі відсутності головного бухгалтера його обов'язки виконує заступник.

Обліком доходів від продажу квитків і управлінням витратами, а також складанням звітів і веденням статистичного обліку займаються бухгалтери.

Економіст вивчає дані та статистичні дані, щоб проаналізувати тенденції економічної діяльності, рівень економічної впевненості та ставлення відвідувачів [10]. Вони беруть участь у розробці та управлінні бюджетом зоопарку, вирішують завдання, пов'язані з плануванням його діяльності та веденням статистичного обліку.

У зоопарку є сувенірні магазини та кафе, де працюють продавці, які обслуговують відвідувачів. Завдання продавців сувенірів – спонукати відвідувачів до покупок. Крім того, біля входу в зоопарк є квитковий контролер, який займається продажем та перевіркою квитків.

Відділ технічного обслуговування контролює начальник відділу, який відповідає за координацію поточного технічного обслуговування для запобігання серйозних збоїв обладнання. У разі поломки пошкоджене обладнання потребує ремонту. Крім того, керівнику відділу доручено закупити необхідні запчастини для ремонту обладнання зоопарку. В даному відділі працюють: 2 електрика, сантехник, 15 робітників, 2 двірник, 2 прибиральник.

Електромонтер відповідає за електропостачання, технічне обслуговування та ремонт експонатів, обладнання та приміщень у зоопарку. Електрики діагностують і усувають несправності різного електричного обладнання або систем керування, наприклад, котлів, насосів, двигунів, генераторів, повітряних компресорів,

пожежної сигналізації, сушилок, холодильників, дверей та інших електричних пристроїв. До їхніх обов'язків також входить встановлення освітлювальних приладів, розеток, елементів керування, обігрівачів, термостатів, комп'ютерних кабелів або подібних пристроїв. Прокладають кабелі та протягують дроти як для нових, так і для існуючих електроустановок. Крім того, електрики проводять профілактичний ремонт електричного та електронного обладнання, вирішуючи проблеми в системах, які містять електронні компоненти [12].

Сантехнік займається встановленням, обслуговуванням та ремонтом різноманітних систем, включаючи дренажні системи, системи протипожежного захисту, газопроводи та обладнання, системи опалення та охолодження, системи поливу, каналізаційні системи, водопровідні, каналізаційні та паропроводи, як а також контрольні пристрої та обладнання для повітряних компресорних систем і магістралей. [11].

Двірники відповідають за підтримання чистоти в зоопарку.

Прибиральники спеціально займаються прибиранням громадських і службових будівель на території зоопарку.

Працівники залучаються до прибирання вольєрів, забезпечення благополуччя тварин через годівлю та догляд, забезпечення охорони зоопарку. Крім того, за потреби вони пропонують допомогу електрикам, сантехнікам і продавцям.

Куратор здійснює контроль за організацією догляду за тваринами, місцями їх проживання, наглядом за персоналом. Вони тісно співпрацюють із працівниками зоопарку, допоміжним персоналом, ветеринарами та відвідувачами.

Менеджер з маркетингу має завдання поширювати інформацію за допомогою цифрових і традиційних медіа, а також використовувати інші інструменти для досягнення відвідувачів і фінансових цілей.

ІТ-менеджеру розробляє програмне забезпечення для автоматизації бухгалтерського обліку та співпрацює з менеджером з маркетингу для створення рекламних веб-сайтів.

Екскурсоводи відіграють вирішальну роль в обміні інформацією про тварин і навколишнє середовище з відвідувачами зоопарку. Вони відповідають за проведення неформальних програм для відвідувачів різного віку, залучення до шкільних груп, таборів і класів. Крім того, гіді допомагають із спеціальними подіями та програмами поведінки тварин [14].

На основі проведеного аналізу діяльності зоопарку необхідно визначити вимоги для розроблювальної інформаційної системи.

1.3. Розробка концепції інформаційної системи “зоопарк”

Об'єктом дослідження є інформаційна система «Зоопарк». Ця система є програмним рішенням, призначеним для оптимізації та підвищення ефективності управління діяльністю зоопарку. Він допомагає здійснювати догляд за тваринами, залучати відвідувачів, вести фінансове адміністрування та обслуговування об'єктів. Надає персоналу зоопарку зручний інтерфейс для доступу та керування важливою інформацією.

Предметом дослідження є модель управління. Ця модель управління має охоплювати такі аспекти, як стратегічне планування, розподіл ресурсів, управління персоналом і фінансовий нагляд.

Метою роботи – підвищення ефективності управління діяльністю зоопарку за рахунок впровадження сучасних систем, які дозволять мінімізувати витрати та збільшити прибуток.

1.3.1. Вимоги до ІС зоопарку

На основі проведеного аналізу інформаційних процесів в даному об'єкті сформуємо такі вимоги:

Функціональні вимоги:

- Автоматизувати функції бухгалтерії, включаючи облік і звітність.
- Впровадити автоматизацію взаємодії з банком та податковою інспекцією.
- Автоматизувати процес нарахування заробітної плати.
- Впровадити автоматизацію продажу квитків.
- Автоматизуйте процеси планування та бюджетування.
- Розробка та запуск рекламного сайту.
- Створення звітів про доходи.
- Створення звітів про витрати.
- Створення звітів про продажі за місячні та річні періоди.
- Створення звітів про відвідуваність.
- Формувати звіти про фінансовий стан.
- Формувати звіти по заробітній платі.
- Впровадити облік фінансових ресурсів.
- Здійснити облік матеріальних ресурсів.

Програма має мати зручний інтерфейс, щоб задовольнити користувачів без спеціальних навичок у комп'ютерних технологіях.

Вимоги до технічних засобів.

Для повноцінного управління зоопарком потрібно мати 10 комп'ютерів з відповідним програмним забезпеченням.

- 6 комп'ютера для бухгалтерії.
- 1 комп'ютер для директора;
- 1 комп'ютер для менеджера з маркетингу;
- 1 комп'ютер для менеджера з інформаційних технологій;
- 1 комп'ютер для менеджера по кадрам.

Мінімальні технічні та програмні вимоги для кожного комп'ютера:

- Операційна система: Windows 7 32/64-bit, Windows 10 32/64-bit

- Оперативна пам'ять: мінімум 8096 Мб
- Периферійні пристрої: клавіатура, миша, монітор
- Пам'ять: Жорсткий диск мінімальним об'ємом 500 Гб
- Додатково кожен відділ повинен бути оснащений принтером.

Для сайту реклами:

Система управління базами даних: технологія СУБД MySQL

Для підтримки рекламного сайту буде достатньо таких технічних та програмних вимог:

- VPS хостинг
- Тактова частота процесора не менше 3,1 ГГц
- Мінімум 8 Гб оперативної пам'яті
- Місткість жорсткого диска не менше 500 ТБ
- Система управління базами даних (СУБД): MySQL
- Периферійні пристрої: клавіатура, миша, монітор

Реалізація завдань бухгалтерського обліку та звітності має бути виконана на основі типових проектних рішень із використанням готових програмних засобів.

1.3.2. Розробка концепції інформаційної системи

На основі встановлених технічних вимог розроблено концепцію інформаційної системи «Зоопарк». [14].

Модель концепції інформаційної системи представлена на рисунку 1.3

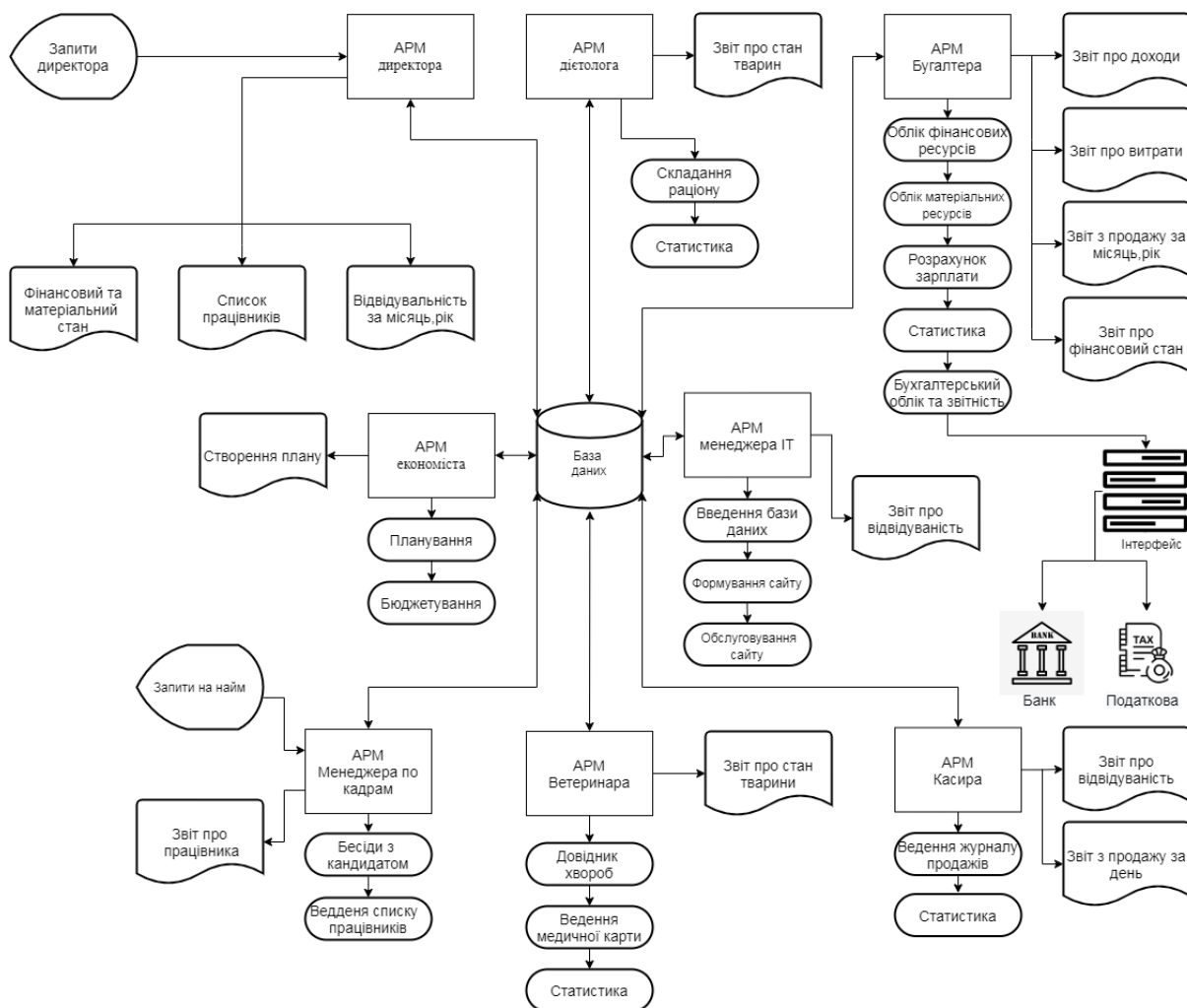


Рисунок 1.3 — Модель концепції інформаційної системи

АРМ бухгалтера виконує різні завдання, включаючи облік фінансових ресурсів, відстеження матеріальних ресурсів, нарахування заробітної плати, ведення статистики. Результатом цих завдань є створення кількох звітів, включаючи звіт про доходи, звіт про витрати, звіт про продажі за місячний і річний періоди та звіт про фінансовий стан.

Використовуючи АРМ, ветеринар може швидко отримати доступ до інформації про захворювання з довідника, що дозволяє йому швидко поставити діагноз. Завдання, які виконуються за допомогою АРМ, включають введення медичної карти тварини та ведення статистики. Результатом цих завдань є формування звіту про стан тварини.

АРМ менеджера по кадрам опрацьовує заявки на наймання співробітників, координує співбесіди з кандидатами. Крім того, вони ведуть список працівників. Результатом цих завдань є створення звіту про статус працівника або інформації.

АРМ менеджера ІТ-менеджера виконує такі завдання, як введення бази даних, створення та підтримка веб-сайту. Результатом цих дій є формування звіту про відвідування сайту.

АРМ економіста бере участь у створенні та веденні бюджету, а також у плануванні діяльності зоопарку.

АРМ касира відповідає за такі завдання, як ведення журналу продажів і зведення статистики. Результатом цих завдань є створення звітів, включаючи звіт про відвідуваність і звіт про продажі за день.

За допомоги АРМ дієтолог може швидко отримати доступ до інформації про раціон тварини з довідника, дозволяючи швидко створити план дієти. Завдання, які виконуються за допомогою АРМ, включають складання раціону тварин і ведення статистики. Результат – формування звіту про стан тварини.

АРМ директора веде запити щодо стану зоопарку і, як наслідок, отримує звіти про різні аспекти, включаючи фінансовий та матеріальний стан зоопарку, список працівників та відвідуваність зоопарку за період від місяця до року.

Наступним кроком у процесі впровадження інформаційної системи є розробка технічного проекту на основі розробленої концепції.

Виконання завдань бухгалтерського обліку та звітності спиратиметься на типові проектні рішення з використанням уже існуючих програмних засобів. Вибір цих інструментів буде ретельно проведено шляхом аналізу доступних варіантів, підкріплених обґрунтуванням їх придатності відповідно до вимог проекту.

Для управління базами даних буде використана система керування базами даних MySQL. Для реалізації обрано мову розробки Java. Цей вибір зроблено з

огляду на надійні характеристики та широке використання цих технологій у подібних проектах, забезпечуючи надійну основу для успішного створення та розгортання інформаційної системи.

Програмна частина технічного проекту

На комп'ютерах встановлений стандартний пакет Microsoft Office, що включає такі програми, як Microsoft Word, Microsoft Excel, Microsoft Outlook, Microsoft PowerPoint, Microsoft Publisher та інші. Додатково встановлено Draw.io для створення діаграм. Для забезпечення безпеки комп'ютера встановлена антивірусна програма 360 Malwarebytes для захисту від вірусів.

Використовується служба електронної пошти gmail.com. Для доступу до Інтернету на комп'ютері встановлено браузер Google Chrome. Zoom використовуватиметься для проведення віддалених зустрічей та співбесід, сприяючи ефективній комунікації та співпраці.

Для впровадження інформаційної системи будуть використані наступні інструменти:

Для реалізації GUI:

- Мова гіпертекстової розмітки – HTML;
- Розширювана мова розмітки гіпертексту – XHTML;
- Мова каскадних таблиць стилів – CSS;
- Мова програмування – JavaScript.

Для формування запитів, створення інтерактивного та незалежного від браузера інтерфейсу:

- Мова програмування – Java;

Серверна частина:

- Система управління базами даних – MySQL;
- Збірка веб-сервера – XAMPP;
- Декларативна мова програмування – SQL.

Ці інструменти разом дозволяють створити надійний, зручний і незалежний від браузера інтерфейс, полегшують ефективну обробку запитів і забезпечують ефективні операції на стороні сервера для керування базами даних і функціональності веб-сервера.

Технічні засоби проєкту:

- 4 каси
- 2 принтери
- 10 комп'ютерів

Комп'ютери мають такі характеристики:

- Операційна система: Windows 10 64-bit
- Оперативна пам'ять: 8096 Мб
- Периферійні пристрої: клавіатура, миша, монітор
- Пам'ять: HDD 600 ГБ

Реалізація концепції проєкту значно підвищить ефективність роботи зоопарку. Таке вдосконалення впливатиме з чіткого виконання управлінських завдань, створення добре реалізованої системи планування, обліку, звітності та ефективного контролю фінансових і матеріальних ресурсів. Інформаційна система сприятиме оптимізації діяльності та створенню більш організованої та прозорої структури управління зоопарком.

1.4. Постановка задачі

Темою дипломної роботи є розробка інформаційної системи управління зоопарком. Метою роботи – підвищення ефективності управління діяльністю зоопарку за рахунок впровадження сучасних систем, які дозволять мінімізувати витрати та збільшити прибуток.

Завдання стоїть у розробці інформаційної системи “Зоопарк”, яка забезпечить автоматизацію процесів управління комунальною установою, покращить якість

управління зоопарком та взаємодію між працівниками, а також сприятиме досягненню наступних цілей:

- Підвищення ефективності процесів управління зоопарком
 - Зменшення витрат на управління зоопарком
 - Покращення якості прийняття рішень
 - Підвищення продуктивності праці працівників
- Покращення якості обслуговування відвідувачів зоопарку
 - Забезпечення комфортного перебування відвідувачів у зоопарку
 - Підвищення рівня задоволеності відвідувачів зоопарком
- Покращення взаємодії між працівниками зоопарку
 - Полегшення обміну інформацією між працівниками
 - Покращення співпраці між працівниками

Висновки до розділу

У першому розділі розроблена концепція інформаційної системи, проведено дослідження моделей управління, їх історію, проведено аналіз зоопарків, проведено обстеження підприємства, проведено аналіз інформаційних процесів, сформовано вимоги до проектованої системи.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ “ЗООПАРК”

2.1. Загальносистемні рішення

Для вирішення завдань бухгалтерського обліку доступні численні системи, такі як HomeBank, GnuCash, KMyMoney, Tryton, 1С: Бухгалтерія, Microsoft Money, Microsoft Dynamics AX тощо.

Розглянемо деякі з них.

GnuCash - програма фінансового обліку (доходи, витрати, банківські рахунки, акції). Має ієрархічну систему рахунків, може розбивати одну транзакцію на кілька частин, безпосередньо імпортувати дані рахунків з інтернету, підтримує Accounting Periods. Базується на професійних засадах бухгалтерського обліку. Поставляється з набором стандартних звітів і дозволяє створювати власні звіти, як нові, так і видозмінені з поставлених. GnuCash підтримує зберігання даних як у форматі XML, так і в базах даних SQLite3, MySQL або PostgreSQL [18]. Далі наведено функціональні можливості:

- Подвійна бухгалтерія
- Заплановані операції
- Помічник з іпотеки та погашення кредиту
- Особливості обліку малого бізнесу
- Імпорт CSV, OFX, QIF
- Підтримка HBCI(Home Banking Computer Interface)
- Підтримка відповідності транзакцій-імпорту
- Підтримка SQL
- Відстеження та звітність щодо ПДВ
- Обробка мультивалютних транзакцій
- Портфелі акцій / взаємних фондів
- Котирування акцій та взаємних фондів онлайн

- Вбудовані та спеціальні звіти та діаграми
- Бюджет
- Звірка банку та кредитної картки
- Перевірка друку [24].

KMyMoney - програма фінансового обліку (доходи, витрати, банківські рахунки, акції). Призначена для приватних осіб та малих підприємств. Програма доступна безкоштовно для Linux, Solaris та ін. ОС із середовищем робочого столу KDE (або хоча б з її основними бібліотеками, такими як KDELibs). Поширюється за умов GNU General Public License.

Функціональні можливості:

- Графічний інтерфейс користувача
- Робота з кількома користувальницькими рахунками
- Роботами з категоріями (статтями) доходів та витрат
- Робота з одержувачами грошей (payee)
- Багатовалютний облік
- Планування регулярних платежів
- Облік кредитних платежів
- Настроювані фінансові звіти
- Імпорт файлів даних із інших фінансових систем QIF, GnuCash [19].

Tryton — високорівнева платформа для розробки програм, що використовує трирівневу архітектуру, на основі якої створено бізнес-рішення, представлене за допомогою так званих модулів Tryton.

Платформа Tryton представлена як трирівнева архітектура: клієнт, сервер та СУБД (PostgreSQL, MySQL, SQLite). Платформа разом із офіційними модулями ліцензована за GPLv3.

Офіційні модулі забезпечують функціональність наступних областей:

- Бухгалтерський облік

- Управління закупками
- Управління запасами
- Управління продажами
- Управління проектами
- Облік робочого часу
- Календар [20].

Microsoft Dynamics AX - одне з програмних рішень корпорації Microsoft, що зараз називається - Dynamics 365, для автоматизації управління підприємством, що поставляються підрозділом Microsoft Dynamics. Система була розроблена для середнього та великого сегментів бізнесу та надає функції фінансового менеджменту, бізнес-аналізу, управління процесами виробництва.

Система охоплює, наприклад, такі галузі менеджменту, як:

- Управління виробництвом;
- Управління дистрибуцією у складних ланцюжках поставок;
- управління роздрібними мережами;
- управління фінансами, включаючи облік за різними стандартами в холдингових структурах;
- Управління проектною діяльністю та сервісним обслуговуванням;
- Контроль та аналіз бізнесу, відповідність корпоративним політикам;
- управління продажами, маркетингом, взаємовідносинами з клієнтами;
- Управління персоналом [21].

1С:Бухгалтерія 8 — універсальна програма 1С масового призначення для автоматизації бухгалтерського і податкового обліку, включаючи підготовку обов'язкової (регламентованої) звітності. Це готове рішення для ведення обліку в організаціях, що здійснюють будь-які види комерційної діяльності: гуртову і роздрібну торгівлю, комісійну торгівлю, надання послуг, виробництво тощо.

Бухгалтерський і податковий облік реалізовані відповідно до чинного законодавства України.

Методика бухгалтерського обліку забезпечує одночасну реєстрацію кожного запису господарської операції як по рахунках бухгалтерського обліку, так і по необхідних розрізах аналітичного обліку, кількісного і валютного обліку. Користувачі можуть самостійно управляти методикою обліку в рамках настройки облікової політики, створювати нові субрахунки і розрізи аналітичного обліку.

«1С:Бухгалтерія 8» забезпечує вирішення всіх задач, що стоять перед бухгалтерською службою підприємства, якщо бухгалтерська служба повністю відповідає за облік на підприємстві, включаючи, наприклад, виписку первинних документів, облік продажів тощо. Крім того, інформацію про окремі види діяльності, торгові і виробничі операції можуть вводити співробітники суміжних служб підприємства, які не є бухгалтерами. В такому випадку за бухгалтерською службою залишається методичне керівництво і контроль за настройками інформаційної бази, що забезпечують автоматичне віддзеркалення документів в бухгалтерському і податковому обліку.

Функції:

- Облік "від документа" і типові операції
- Ведення обліку діяльності декількох організацій
- Банківські та касові операції
- Облік матеріально-виробничих запасів
- Складський облік
- Облік торговельних операцій
- Облік комісійної торгівлі
- Облік операцій з тарою
- Облік банківських і касових операцій
- Облік розрахунків з контрагентами

- Облік основних засобів, нематеріальних і малоцінних активів
- Облік основного і допоміжного виробництва
- Облік напівфабрикатів
- Облік непрямих витрат
- Облік ПДВ
- Облік заробітної плати, кадровий і персоніфікований облік
- Підтримка різних схем оподаткування
- Податковий облік з податку на прибуток
- Спрощена система оподаткування
- Завершальні операції місяця
- Стандартні бухгалтерські звіти
- Регламентована звітність
- Сервісні можливості
- Засоби для швидкого освоєння [23].

Уніфікована мова моделювання (UML) — це мова моделювання загального призначення для розробки в галузі розробки програмного забезпечення, яка призначена для надання стандартного способу візуалізації дизайну системи [35].

Діаграма прецедентів або діаграма варіантів використання — це графічне зображення можливих взаємодій користувача із системою. Діаграма варіантів використання показує різні варіанти використання та різні типи користувачів, які є в системі, і часто супроводжуватиметься іншими типами діаграм. Варіанти використання представлені чи колами, чи еліпсами. Акторів часто зображують як фігурок [36].

Діаграма представлена на рисунку 2.1

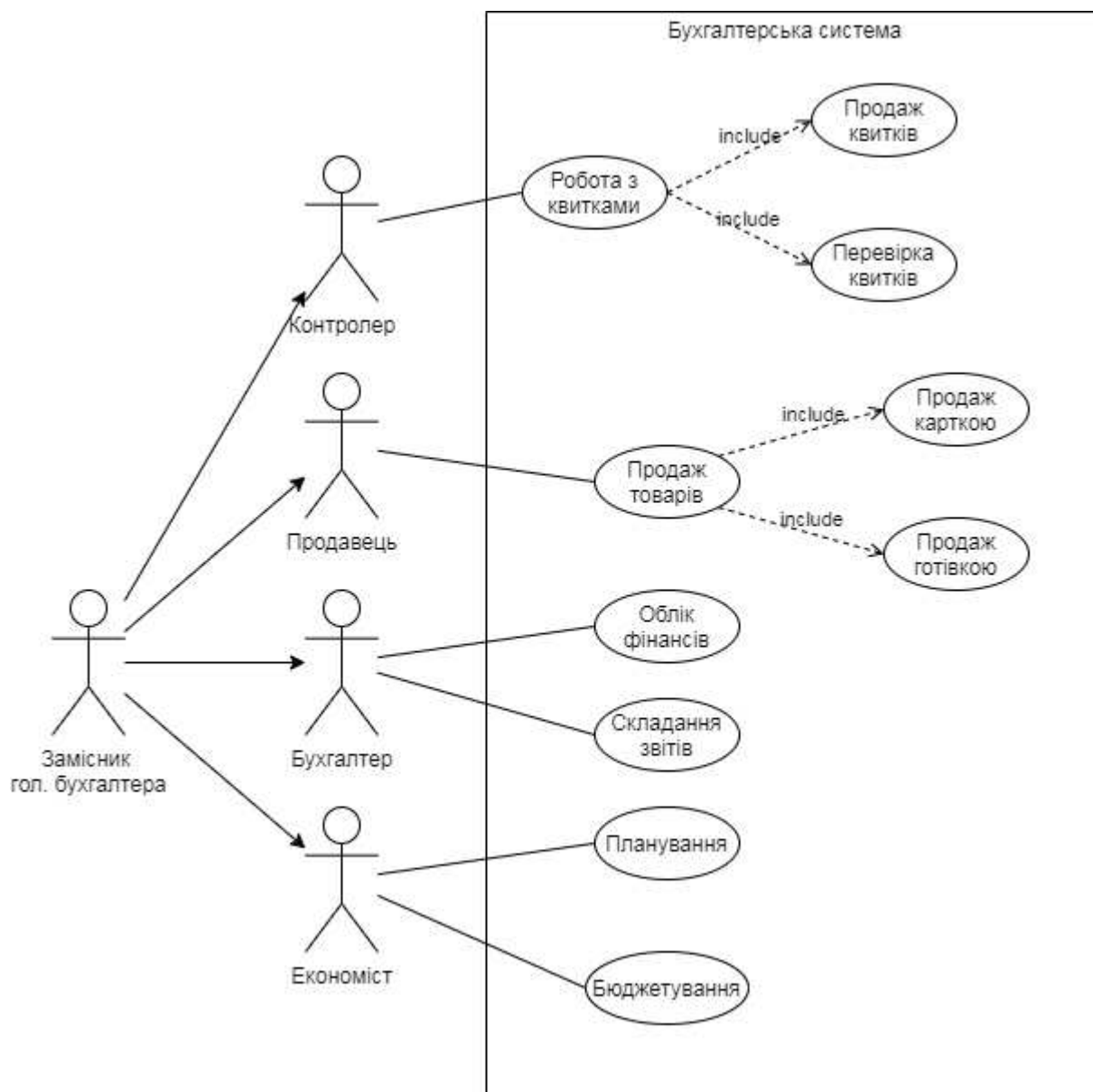


Рисунок 2.1 - Прецеденти бухгалтерський відділ

Діаграма прецедентів виконано на рівні бухгалтерії. Нижче наведено опис взаємодії акторів із системою.

Ціль контролера — робота з квитками. Сюди входить продаж та перевірка квитків.

Ціль продавця — продаж товарів. Продаж товарів може здійснюватися як карткою так і готівкою.

Ціль бухгалтера — облік фінансів та складання звітів.

Ціль економіста — планування та бюджетування роботи зоопарку.

2.2. Рішення по інформаційному забезпеченню

Для нормального функціонування інформаційної системи необхідна база даних. Для управління базою даних використовується СУБД MySQL.

MySQL — це система керування реляційною базою даних з відкритим кодом. У реляційній базі даних дані структуровані в одну або кілька таблиць, а зв'язки між даними встановлюються для ефективної організації інформації. MySQL надає автономні клієнти, що забезпечують пряму взаємодію з базою даних MySQL за допомогою SQL. Однак його частіше інтегрують з іншими програмами для реалізації систем, які потребують можливостей реляційної бази даних [24].

Нижче представлені концептуальна, логічна та фізична модель бази даних. Створені моделі за допомогою програмного забезпечення draw.io.

2.2.1. Розробка концептуальної моделі даних

Концептуальна модель бази даних ілюструє зв'язки між об'єктами та надає описи цих об'єктів. Основною метою створення концептуальної моделі даних є визначення сутностей, їхніх атрибутів і зв'язків.

Три основні елементи концептуальної моделі даних включають:

- Сутність: представлення об'єкта.
- Атрибут: Опис характеристик або властивостей об'єкта.
- Відносини: позначення залежності або асоціації між двома об'єктами[24].

Концептуальна модель бази даних представлена на рисунку 2.2

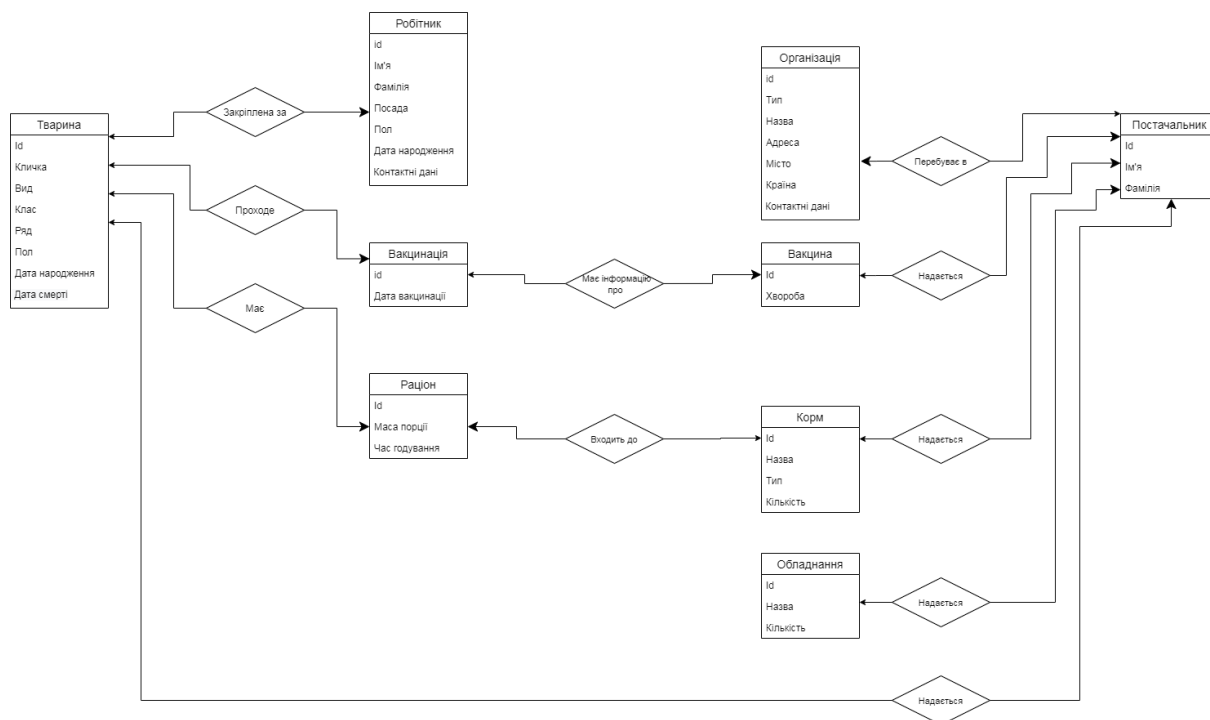


Рисунок 2.2 - Концептуальна модель бази даних

2.2.2. Розробка логічної моделі даних

Логічна модель даних — це представлення, яке описує сутності, про які організація має намір збирати дані, і зв'язки між цими сутностями. Він охоплює зображення сутностей та їхніх атрибутів, зв'язків, унікальних ідентифікаторів, підтипів і супертипів, а також обмежень між зв'язками. Ця модель служить для забезпечення структурованого розуміння елементів даних та їхніх взаємозв'язків в організації [29].

Логічна модель бази даних представлена на рисунку 2.3

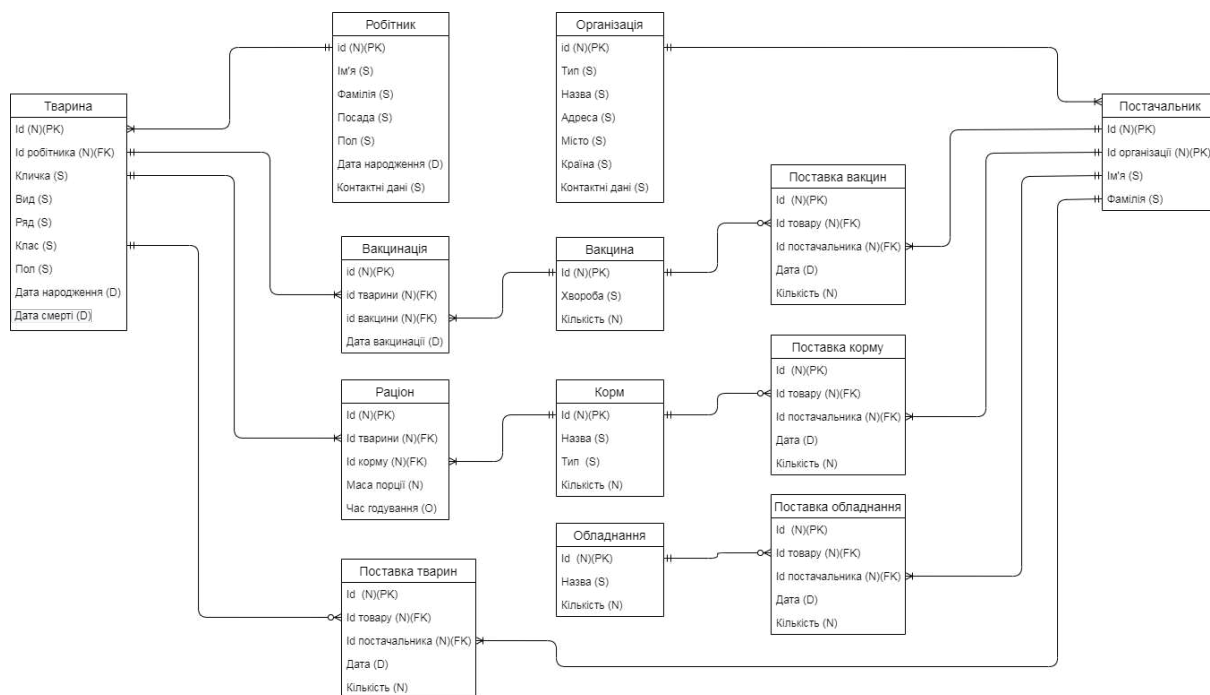


Рисунок 2.3 - Логічна модель бази даних

2.2.3. Розробка фізичної моделі даних

Фізична модель даних формулюється на основі концептуальної моделі. Ця модель точно визначає, як буде виконуватися система, використовуючи конкретну систему керування базами даних. Він окреслює конкретні деталі впровадження, включаючи специфіку, адаптовану до обраної СУБД для оптимальної реалізації системи. [29].

Фізична модель бази даних представлена на рисунку 2.4.

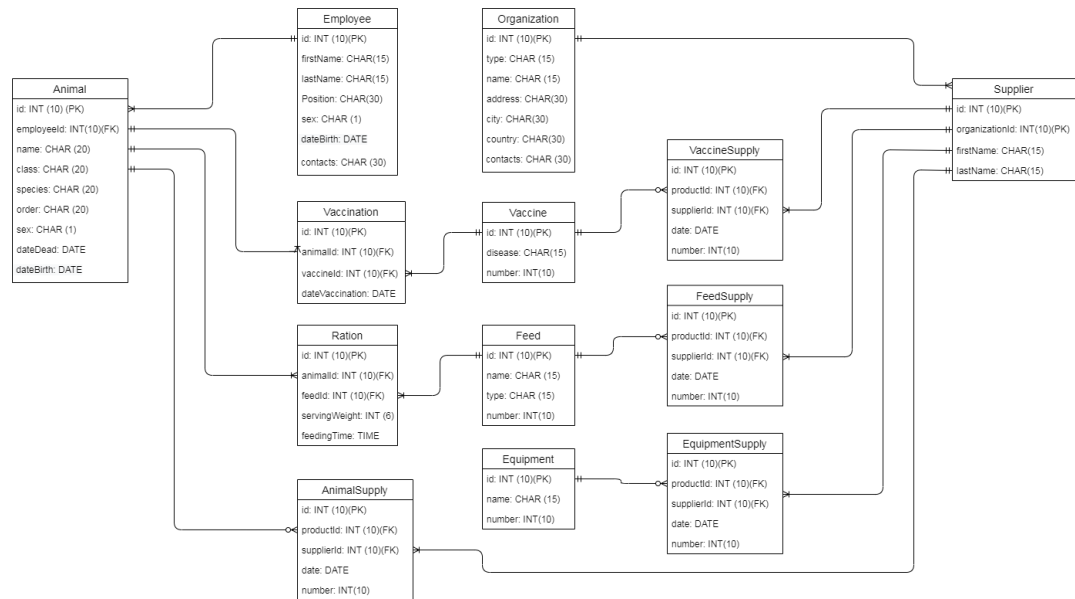


Рисунок 2.4 - Фізична модель бази даних

- **Animal** — зберігає інформацію про тварин, а саме: id тварини, id співробітника за яким закріплена дана тварина, кличка тварини, клас тварини, вид тварини, ряд тварини, пол тварини, дата смерті тварини (якщо вмерла) та дата народження тварини;
- **Employee** — зберігає інформацію про співробітників, а саме: id співробітника, ім'я та фамілію співробітника, посада співробітника, пол співробітника, дата народження співробітника та контактні дані співробітника;
- **Ration** — зберігає інформацію про раціон, а саме: id раціону, id тварини для якої створений даний раціон, id корму, кількість порції та час годування;
- **Vaccination** — зберігає інформацію про вакцинацію, а саме: id вакцинації, id тварини для якої було проведено вакцинацію, id вакцини якою було проведено вакцинацію та дата вакцинації;
- **Feed** — зберігає інформацію про корми, а саме: id корму, назва корму, тип корму та кількість корму на складі;
- **Vaccine** — зберігає інформацію вакцину, а саме: id вакцини, хвороба от якої ця вакцина та кількість вакцин;

- Equipment — зберігає інформацію про обладнання, а саме: id обладнання, назва обладнання та кількість обладнання;
- Equipment supply, Feed supply, Vaccine supply, Animal supply — зберігають інформацію про поставки обладнання, кормів, вакцин, тварин відповідно, а саме: id поставки, id продукту, id постачальника, дата поставки, кількість поставленого продукту;
- Supplier — зберігає інформацію про постачальників, а саме: id постачальника, id організації із якої даний постачальник, ім'я та фамілія постачальника;
- Organization — зберігає інформацію про організацію, а саме: id організації, тип організації, назва організації, адреса організації, місто організації, країна організації та контактні дані організації.

На основі цих моделей розроблятиметься програмне забезпечення системи управління та формування бази даних

2.3. Рішення з математичного забезпечення

Блок-схема загального алгоритму роботи частини адміністратора системи представлена на рисунку 2.5

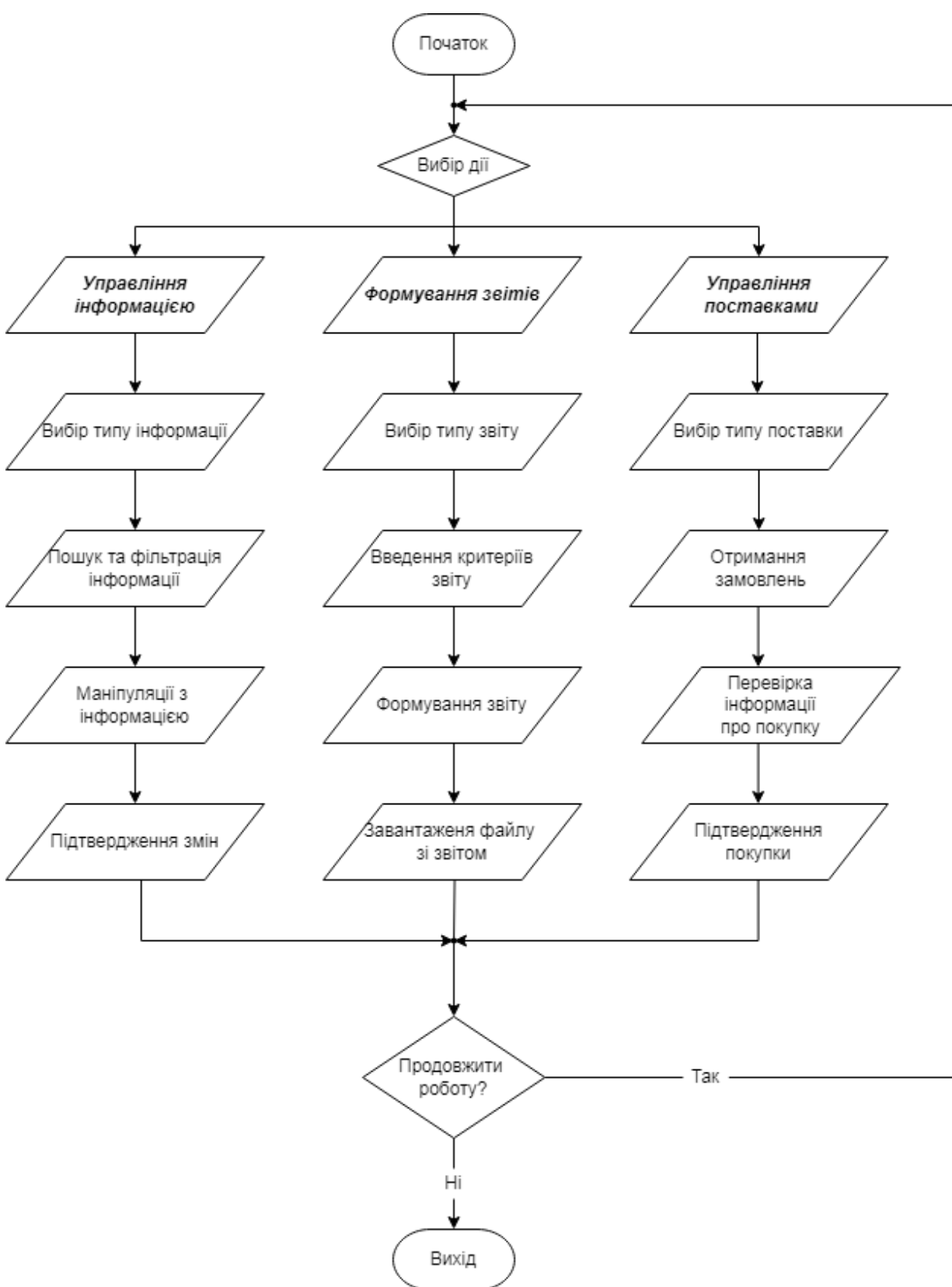


Рисунок 2.5 – Блок-схема загального алгоритму роботи частини адміністратора системи

Адміністратор має можливість запитувати різноманітні звіти, які висвітлюють різні аспекти роботи інформаційної системи, такі як дані про працівників, продажі, поставки тощо. Для цього він обирає відповідну функцію зі списку. Під час створення звіту адміністратор може обирати тип звіту і вводити критерії, такі як часовий проміжок або конкретний постачальник, для отримання детальної інформації. Згенерований звіт відображається у вигляді окремої форми.

Адміністратор також має можливість керувати інформацією, якою оперує інформаційна система. Він може додавати, редагувати та видаляти інформацію з системи, вибираючи відповідні функції зі списку. На сторінці управління інформацією адміністратор може вибрати тип інформації (наприклад, постачальники, тварини, вакцини) і використовувати критерії пошуку та фільтрації для отримання списку потрібної інформації. Поряд із списком розташовані кнопки, що відповідають за конкретні маніпуляції з інформацією, такі як додавання, редагування або видалення.

Також адміністратор має можливість управляти поставками, обираючи відповідну функцію зі списку. Він може обрати тип поставок і обробити замовлення від працівників, підтверджуючи чи відхиляючи їх.

Блок-схема загального алгоритму роботи частини працівника представлена на рисунку 2.6.

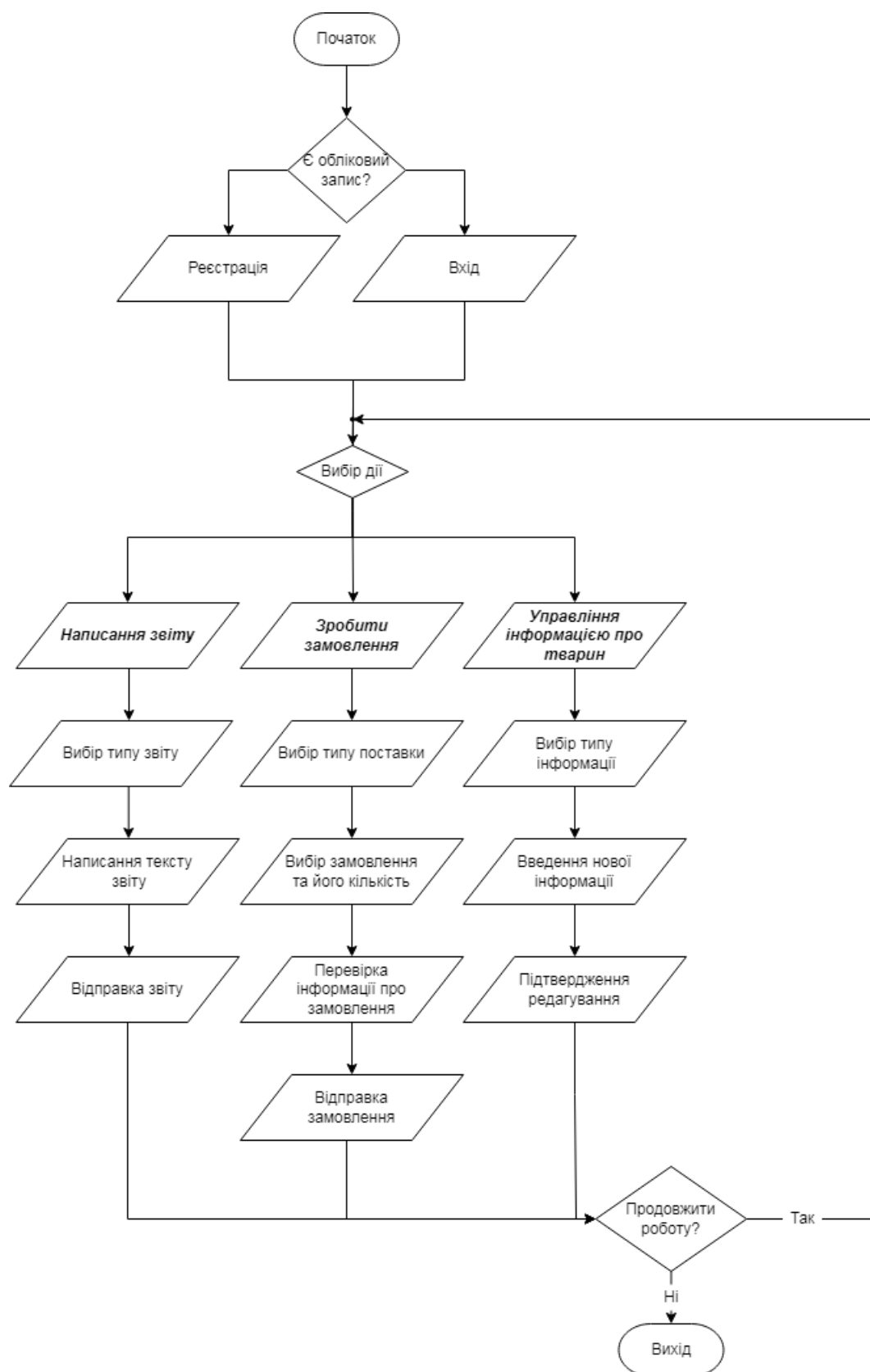


Рисунок 2.6 – Блок-схема загального алгоритму роботи частини працівника

Для початку роботи працівник повинен мати обліковий запис. У верхньому правому куті сайту розташовані дві кнопки: "Увійти" для існуючих користувачів та "Зареєструватися" для нових. При виборі "Увійти" з'являється форма з інформацією, такою як ім'я користувача або пошта, пароль та кнопка підтвердження входу. При виборі "Зареєструватися" з'являється форма для введення інформації, такої як ім'я працівник, пошта, номер телефону, пароль, підтвердження паролю та кнопка підтвердження реєстрації. Інформація про працівник зберігається у базі даних.

Після входу працівник може вибрати дію: написати звіт, зробити замовлення, управляти інформацією про тварин. Обрана дія веде працівника на відповідну сторінку.

На сторінці "Управління інформацією про тварин" доступний вибір типу інформації (вакцинація, раціон і т. д.). Після вибору працівник може додавати або редагувати інформацію, заповнюючи відповідні поля.

На сторінці "Зробити замовлення" працівник може замовити поставки, обираючи тип поставки, поставку та її кількість. Після відправлення замовлення адміністратору може відхилити або підтвердити його.

На сторінці "Написати звіт" працівник може обрати тип звіту (вакцинація, тварини і т. д.), написати його та відправити.

На основі вищезгаданої інформації створено діаграму послідовності для адміністратора.

Блок-схема діаграми послідовності представлена на рисунку 2.7

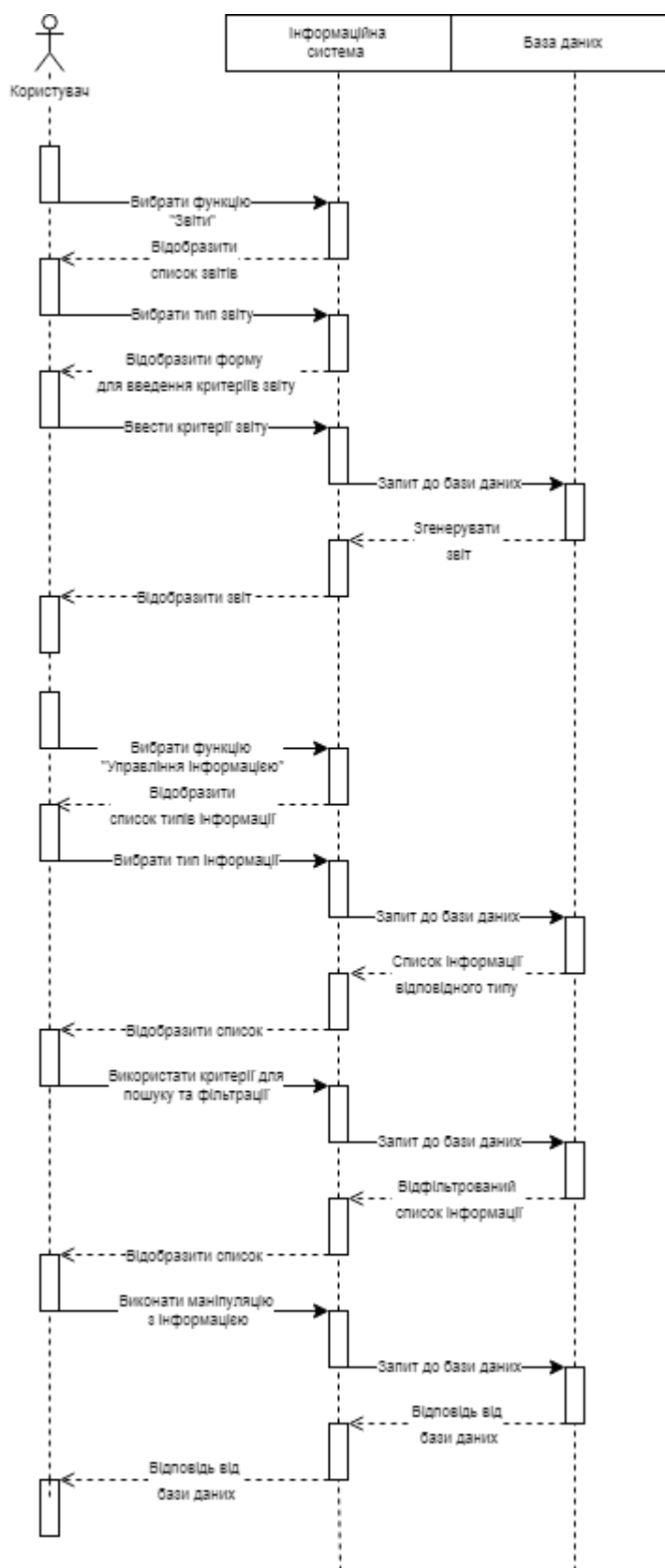


Рисунок 2.7 – Блок-схема загального алгоритму роботи частини працівника

Розроблено програмне забезпечення на основі математичного забезпечення для реалізації функцій відповідно до вимог до інформаційної системи.

2.4. Рішення з програмного забезпечення

Програмне забезпечення інформаційної системи розроблялося за допомогою наступного програмного інструментарію.

MySQL — вільна система керування реляційними базами даних, яка була розроблена компанією «ТсХ» для підвищення швидкодії обробки великих баз даних. Ця система керування базами даних (СКБД) з відкритим кодом була створена як альтернатива комерційним системам. MySQL з самого початку була дуже схожою на mSQL, проте з часом вона все розширювалася і зараз MySQL — одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних вебсторінок, оскільки має чудову підтримку з боку різноманітних мов програмування.

MySQL — компактний багатопотоковий сервер баз даних. Характеризується високою швидкістю, стійкістю і простотою використання.

MySQL вважається гарним рішенням для малих і середніх застосувань. Сирцеві коди сервера компілюються на багатьох платформах. Найповніше можливості сервера виявляються в UNIX-системах, де є підтримка багатопоточності, що підвищує продуктивність системи в цілому.

Можливості сервера MySQL:

- простота у встановленні та використанні;
- підтримується необмежена кількість користувачів, що одночасно працюють із БД;
- кількість рядків у таблицях може досягати 50 млн;
- висока швидкість виконання команд;
- наявність простої і ефективної системи безпеки [24].

SQL — декларативна мова програмування для взаємодії користувача з базами даних, що застосовується для формування запитів, оновлення і керування реляційними БД, створення схеми бази даних та її модифікації, системи контролю за доступом до бази даних. Сама по собі SQL не є ані системою керування базами даних, ані окремим програмним продуктом. На відміну від дійсних мов програмування (C або Pascal), SQL може формувати інтерактивні запити або, будучи вбудованою в прикладні програми, виступати як інструкції для керування даними. Окрім цього, стандарт SQL містить функції для визначення зміни, перевірки та захисту даних.

SQL — це діалогова мова програмування для здійснення запиту і внесення змін до бази даних, а також керування базами даних. Багато баз даних підтримує SQL з розширеннями до стандартної мови. Ядро SQL формує командна мова, яка дозволяє здійснювати пошук, вставку, оновлення і видалення даних за допомогою використання системи керування і адміністративних функцій. SQL також включає CLI (Call Level Interface) для доступу і керування базами даних дистанційно.

Основу бази даних SQL Server утворює сервер або ядро бази даних. Ядро бази даних відповідає за обробку запитів, які надходять від клієнтів, і передачу відповідних результатів клієнтським компонентам.

Мова SQL поділяється на кілька видів елементів:

- Пункти (диз'юнкти) (англ. Clauses), що є складовими частинами інструкцій та запитів. (Іноді вони не обов'язкові.)
- Вирази (англ. Expressions), які можуть генерувати скалярні значення, або таблиці з стовпчиками і рядками даних
- Предикати (англ. Predicates), які описують умови, результатом яких є значення тризначної логіки SQL (true/false/unknown) або Булеві значення істинності і які використовуються для обмеження ефекту інструкцій та запитів, або для зміни потоку виконання програми.

- Запити (англ. Queries), які отримують дані на основі заданих критеріїв.
- Інструкції (англ. Statements), які чинять дію на схему даних чи самі дані, або контролюють транзакції, потік виконання програми, з'єднання, сесії, та виконують діагностику.
- Незначимі пропуски загалом ігноруються в інструкціях і запитах SQL, дозволяючи формувати код SQL з метою покращення читабельності [25].

XAMPP — безкоштовна багатоплатформова збірка вебсервера з відкритим початковим кодом, що містить HTTP-сервер Apache, базу даних MariaDB, MySQL й інтерпретатори скриптів для мов програмування PHP та Perl, а також додаткові бібліотеки, що дозволяють запустити повноцінний вебсервер[30].

Web-додаток для взаємодії з базою даних розроблений за допомогою:

- JavaScript;
- HTML ;
- CSS;
- Java.

JavaScript — динамічна, об'єктно-орієнтована прототипна мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується для створення сценаріїв web-сторінок, що надає можливість на боці клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд web-сторінки.

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні

властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

JavaScript має низку властивостей об'єктно-орієнтованої мови, але завдяки концепції прототипів підтримка об'єктів в ній відрізняється від традиційних мов ООП. Крім того, JavaScript має кілька властивостей, притаманних функціональним мовам, — функції як об'єкти першого класу, об'єкти як списки, каррінг, анонімні функції, замикання (closures) — що додає мові додаткову гнучкість.

JavaScript має C-подібний синтаксис, але в порівнянні з мовою C має такі корінні відмінності:

- об'єкти, з можливістю інтроспекції і динамічної зміни типу через механізм прототипів
- функції як об'єкти першого класу
- обробка винятків
- автоматичне приведення типів
- автоматичне збирання сміття
- анонімні та стрілочні функції

JavaScript містить декілька десятків вбудованих об'єктів[21], які поділяються на групи: фундаментальні, помилки, числа та дати, текстові, індексовані, ключові, для роботи з структурованими даними, абстрактні, рефлексійні, групи Intl та WebAssembly. Крім того, JavaScript містить набір вбудованих операцій, що керують логікою виконання програм. Синтаксис JavaScript в основному відповідає синтаксису мови Java (тобто, зрештою, успадкований від C), але спрощений у порівнянні з ним, щоб зробити мову сценаріїв легкою для вивчення. Так, наприклад, декларація змінної не містить її типу, властивості також не мають типів, а декларація функції може знаходитися в тексті програми після неї [26].

GlassFish - це реалізація Eclipse для Jakarta EE (раніше еталонна реалізація від Oracle) і, як така, підтримує EJB, JPA, JSF, JMS, RMI, JSP, сервлети і т.д. Це дозволяє розробникам створювати переносні та масштабовані корпоративні програми, а також які інтегруються зі застарілими технологіями. Додаткові компоненти також можуть бути встановлені для додаткових служб.

Побудований на модульному ядрі на основі OSGi, GlassFish працює безпосередньо поверх реалізації Apache Felix. Він також працює із середовищами виконання Equinox OSGi або Knopflerfish OSGi. HK2 абстрагує модульну систему OSGi для надання компонентів, які можна розглядати як служби. Такі послуги можна виявити та впровадити під час виконання [35].

HTML (англ. HyperText Markup Language — мова розмітки гіпертексту) — стандартизована мова розмітки документів для перегляду веб-сторінок у браузері. Веб-браузери отримують HTML документ від сервера за протоколами HTTP/HTTPS або відкривають з локального диска, далі інтерпретують код в інтерфейс, який відображатиметься на екрані монітора.

Елементи HTML є будівельними блоками сторінок HTML. За допомогою конструкцій HTML, зображення та інші об'єкти, такі як інтерактивні форми, можуть бути вбудовані у візуалізовану сторінку. HTML надає засоби для створення структурованих документів, позначаючи структурну семантику тексту, наприклад заголовки, абзаци, списки, посилання, цитати та інші елементи. Елементи HTML окреслені тегами, написаними з використанням кутових дужок. Теги на кшталт `<img />` чи `<input />` безпосередньо виводять вміст на сторінку. Інші теги, такі як `<p>`, оточують текст і надають інформацію про нього, а також можуть включати інші теги як піделементи. Браузери не показують теги HTML, але використовують їх для інтерпретації вмісту сторінки.

В HTML можна вбудовувати програми, написані на скриптових мовах, наприклад JavaScript, які впливають на поведінку та вміст вебсторінок. Включення

CSS визначає вигляд і компонування вмісту. World Wide Web Consortium (W3C), який супроводжує стандарти HTML та CSS, заохочує використання CSS над явним презентаційним HTML з 1997 року.

HTML впроваджує засоби для:

- створення структурованого документа шляхом позначення структурного складу тексту: заголовки, абзаци, списки, таблиці, цитати та інше;
- отримання інформації зі Всесвітньої мережі через гіперпосилання;
- створення інтерактивних форм;
- включення зображень, звуку, відео, та інших об'єктів до тексту [27].

XHTML (англ. Extensible Hypertext Markup Language, укр. Розширювана мова розмітки гіпертексту) — мова розмітки, що має таку саму виразну силу як і HTML але відповідає синтаксичним правилам XML.

В той час як HTML побудовано на основі правил SGML, XHTML побудовано на основі правил XML, суворішої підмножини правил SGML. Оскільки XHTML-документи мають бути коректними XML-документами, їх обробку можна здійснювати стандартними інструментами обробки XML-документів на відміну від HTML, який вимагає порівняно складніших, важчих і повільніших синтаксичних аналізаторів. XHTML можна розглядати як, багато в чому, перетин HTML і XML, оскільки цей стандарт є переформулюванням HTML засобами XML.

CSS (англ. Cascading Style Sheets, укр. Каскадні таблиці стилів) — це спеціальна мова стилю сторінок, що використовується для опису їхнього зовнішнього вигляду. Самі ж сторінки написані мовами розмітки даних.

CSS є основною технологією всесвітньої павутини, поряд із HTML та JavaScript.

Найчастіше CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів.

CSS має різні рівні та профілі. Наступний рівень CSS створюється на основі попередніх, додаючи нову функціональність або розширюючи вже наявні функції. Рівні позначаються як CSS1, CSS2 та CSS3. Профілі — сукупність правил CSS одного або більше рівнів, створені для окремих типів пристроїв або інтерфейсів. Наприклад, існують профілі CSS для принтерів, мобільних пристроїв тощо.

CSS (каскадна або блочна верстка) прийшла на заміну табличній верстці вебсторінок. Головна перевага блочної верстки — розділення змісту сторінки (даних) та їхньої візуальної презентації [30].

Частина системи, що пов'язана з сервером, була розроблена з використанням мови програмування Java.

Java (вимовляється Джава) — об'єктно-орієнтована мова програмування, випущена 1995 року компанією «Sun Microsystems» як основний компонент платформи Java. З 2009 року мовою займається компанія «Oracle», яка того року придбала «Sun Microsystems». В офіційній реалізації Java-програми компілюються у байт-код, який при виконанні інтерпретується віртуальною машиною для конкретної платформи.

Мова значно запозичила синтаксис із C і C++. Зокрема, взято за основу об'єктну модель C++, проте її модифіковано. Усунуто можливість появи деяких конфліктних ситуацій, що могли виникнути через помилки програміста та полегшено сам процес розроблення об'єктно-орієнтованих програм. Ряд дій, які в C/C++ повинні здійснювати програмісти, доручено віртуальній машині. Передусім Java розроблялась як платформо-незалежна мова, тому вона має менше низькорівневих можливостей для роботи з апаратним забезпеченням, що в порівнянні, наприклад, з C++ зменшує швидкість роботи програм. За необхідності таких дій Java дозволяє викликати підпрограми, написані іншими мовами програмування.

На противагу C++, Java є більш об'єктно-орієнтованою. Всі дані і дії групуються в класи об'єктів. Виключенням з повної об'єктності (як скажімо

в Smalltalk) є примітивні типи (int, float тощо). Це було свідомим рішенням проєктувальників мови задля збільшення швидкості. Через це Java не вважається повністю об'єктно-орієнтовною мовою.

У Java всі об'єкти є похідними від головного об'єкта (він називається просто Object), з якого вони успадковують базову поведінку і властивості.

Хоча у C++ вперше стало доступне множинне успадкування, але у Java можливе тільки одинарне успадкування, завдяки чому виключається можливість конфліктів між членами класу (методи і змінні), які успадковуються від базових класів [31].

Проект розроблявся у інтегрованому середовищі розробки IntelliJ IDEA.

IntelliJ IDEA — комерційне інтегроване середовище розробки для різних мов програмування (Java, Python, Scala, PHP) від компанії JetBrains. Система поставляється у вигляді урізаної по функціональності безкоштовної версії «Community Edition» і повнофункціональної комерційної версії «Ultimate Edition», для якої активні розробники відкритих проєктів мають можливість отримати безкоштовну ліцензію.

Community версія середовища IntelliJ IDEA підтримує інструменти (у вигляді плагінів) для проведення тестування TestNG і JUnit, системи контролю версій CVS, Subversion, Mercurial і Git, засоби складання Maven, Ant, Gradle, мови програмування Java, Scala, Clojure, Groovy і Dart. Підтримується розробка застосунків для мобільної платформи Android. До складу входить модуль візуального проєктування GUI-інтерфейсу Swing UI Designer, XML-редактор, редактор регулярних виразів, система перевірки коректності коду, система контролю за виконанням завдань і доповнення для імпорту та експорту проєктів з Eclipse. Доступні засоби інтеграції з системами відстеження помилок JIRA, Trac, Redmine, Pivotal Tracker, GitHub, YouTrack, Lighthouse.

Ultimate версія відрізняється наявністю підтримки додаткових мов програмування (наприклад, PHP, Ruby, Python, JavaScript, CoffeeScript, HTML, CSS, SQL), підтримкою технологій Java EE, UML-діаграм, підрахунок покриття коду, можливістю роботи з фреймворками (Rails, Grails, Google Web Toolkit, Spring, Play Framework і Hibernate), засобами інтеграції з Perforce, Microsoft Team Foundation Server і Rational ClearCase [34].

З використанням цих засобів створено програмне забезпечення інформаційної системи. Враховуючи, що ми використовуємо стандартне програмне забезпечення, а саме 1С Бухгалтерія, необхідна його адаптація до документації зоопарку.

Для решти завдань проекту розробка програмного забезпечення була успішно завершена. Це включає створення бази даних, розробку додатків і ретельне тестування. У додатку міститься код розроблених програм. Перевірка показала, що програмне забезпечення працює відповідно до поставлених завдань.

2.5. Рішення з технічного забезпечення

Оскільки користувачам потрібен браузер для взаємодії з інформаційною системою, вкрай важливо, щоб пристрої мали характеристики, які сприяють стабільній роботі браузера.

Мінімальні технічні та програмні вимоги до комп'ютерів включають:

- Операційна система: Windows 7 32/64-bit, Windows 10
- Оперативна пам'ять: не менше 8096 Мб
- Периферійні пристрої: клавіатура, миша, монітор
- Відеопам'ять: мінімум 512 Мб
- HDD: Мінімум 500 Гб.

Викладені технічні та програмні вимоги вважаються достатніми для підтримки рекламного сайту:

- VPS хостинг

- Тактова частота процесора: мінімум 3,1 ГГц
- Оперативна пам'ять: мінімум 8 Гб
- Ємність жорсткого диска: не менше 500 Гб
- Система управління базами даних (СУБД): MySQL
- Периферійні пристрої: клавіатура, миша, монітор

Сформовано мережеву структуру в зоопарку на якій буде встановлено програмні засоби та виконуватиметься реалізація інформаційної системи.

Висновки до розділу

У цьому розділі викладено технічний проект системи, який охоплює загальні рішення для вибору стандартних програмних засобів, проектування бази даних і представлення концептуальних, логічних і фізичних моделей. Він формулює математичні алгоритми, розроблені для вирішення проблем, і визначає технічні та програмні інструменти, які використовуватиме система.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОЕКТУ ІНФОРМАЦІЙНОЇ СИСТЕМИ “ЗООПАРК”

Для роботи системи була розроблена вичерпна робоча документація, яка допоможе користувачам і адміністраторам ефективно використовувати систему.

Враховуючи велику кількість полів у програмі, корисні підказки включено для спрощення завдань користувача. Крім того, реалізовано обробку помилок, яка відображає відповідну інформацію на екрані в разі виникнення помилок. У ситуаціях, коли форма неповна, система надає вказівки щодо того, які поля потрібно заповнити.

Для роботи системи була розроблена вичерпна робоча документація, яка допоможе користувачам і адміністраторам ефективно використовувати систему. Враховуючи широкий діапазон полів у програмі, корисні підказки включено для спрощення завдань користувача. Крім того, реалізовано обробку помилок, яка відображає відповідну інформацію на екрані в разі виникнення помилок. У ситуаціях, коли форма неповна, система надає вказівки щодо того, які поля потрібно заповнити.

Інформаційна система пройшла ретельне тестування, включаючи перевірку серверної частини, перевірку працездатності при введенні користувачами даних. Оскільки бухгалтерський аспект використовує систему 1С Бухгалтерія, відповідна робоча документація була оформлена належним чином.

Вкладка кадровий облік в 1С Бухгалтерії представлено на рисунку 3.1

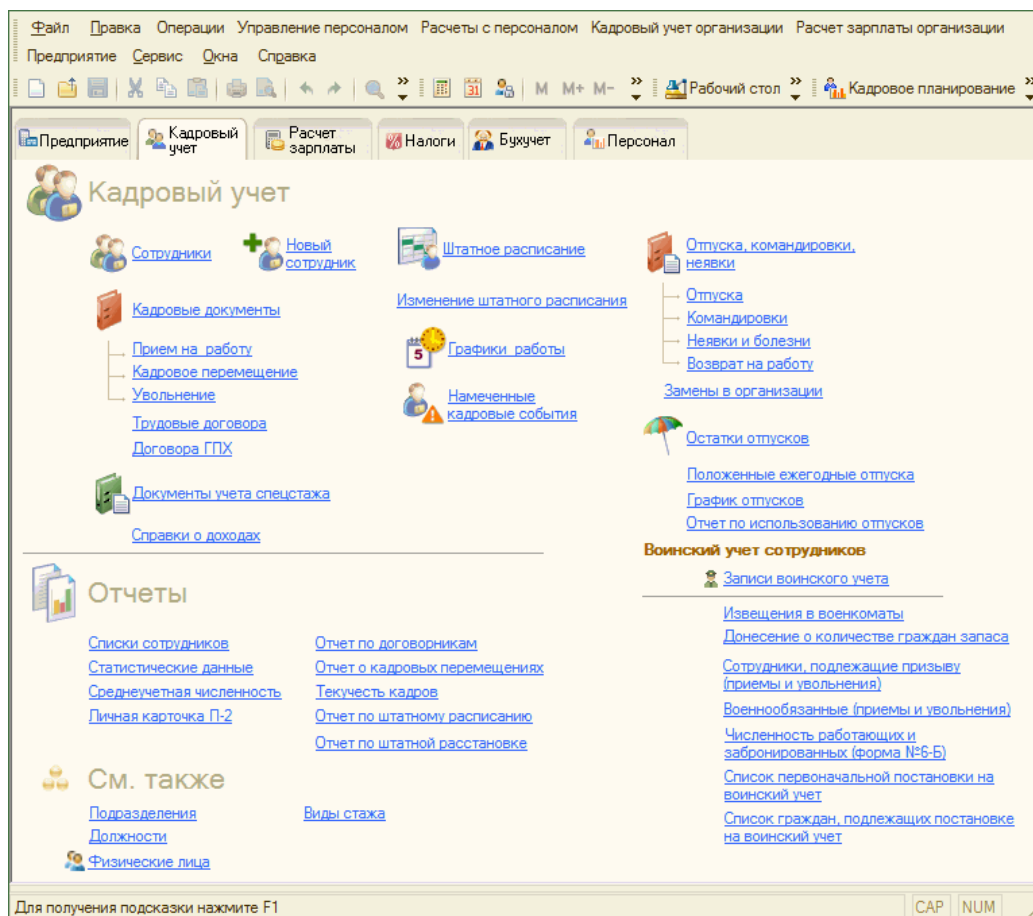


Рисунок 3.1 – Кадровий облік

Приклад розрахунку зарплати в 1С Бухгалтерії представлено на рисунку 3.2

Назначение зарплаты работникам организаций: Проведен

Действия: Перейти | Заполнить и рассчитать все | Заполнить | Рассчитать | Очистить | Советы ?

Номер: Д000000001 от: 29.04.2011 00:00:00 Месяц начисления: Апрель 2011

Организация: Добро Норма за месяц: дней 20,00 часов 160,00

Подразделение: Базовый период индексации соц. выплат: . . .

Ответственный:

Заполнить

N	Таб. №	Работник (основное назначение)
12	0000003	Аксёнов Артур Петрович (осн.)
13	0000004	Абдулов Юрий Владимирович (осн.)
14	0000005	Голубова Елена Ивановна (осн.)

Начисления | Взносы | НДФЛ | Удержания | Взносы ФОТ | Проводки | Расчетный листок

Подбор | Заполнить | Рассчитать | ☐ Предварительный расчет

N	Кар...	При... авто...	Работник (назначение)	Подраздел... организации	Начисление	Период		Базовый период		Результат	Размер	Отработано		Норма за месяц		Оплачено дней/часов	С... о...
						Дат...	Дл...	Начало	Конец			Дней	Часов	Дней	Часов		
1		☒	Абдулов Ю...	Администра...	Оклад по ...	01....	30....	01.04...	30.04...	2 500,00	2 500,000	20,00	160,00	20,00	160,00	20,00	
2		☒	Абдулов Ю...	Администра...	Индексация...	01....	30....	01.04...	30.04...	1 368,96	1,000						
3		☒	Авдеев Вл...	Ремонтна ...	Оклад по ...	01....	30....	01.04...	30.04...	3 100,00	3 100,000	20,00	160,00	20,00	160,00	20,00	
4		☒	Авдеев Вл...	Ремонтна ...	Индексация...	01....	30....	01.04...	30.04...	1 107,84	1,000						
5		☒	Аксёнов ...	Администра...	Оклад по ...	01....	30....	01.04...	30.04...	3 500,00	3 500,000	20,00	160,00	20,00	160,00	20,00	
6		☒	Аксёнов ...	Администра...	Индексация...	01....	30....	01.04...	30.04...	1 107,84	1,000						
										94 622,72							

Комментарий:

Табель П-5 (бланк) | Печать | OK | Записать | Закрыть

Рисунок 3.2 – Приклад розрахунок зарплаты

Вкладка розрахунок зарплаты в 1С Бухгалтерії представлено на рисунку 3.3

Предприятие | Кадровый учет | **Расчет зарплаты** | Налоги | Бухучет | Персонал

Расчет зарплаты

Журнал начисления зарплаты

- Начисление зарплаты
- Премии сотрудников
- Регистрация разовых начислений
- Регистрация разовых удержаний
- Оплата праздничных и выходных
- Оплата сверхурочных часов
- Сдельный наряд
- Регистрация простоев
- Расчет при увольнении

Показатели для расчета зарплаты

- Ввод плановых начислений
- Ввод плановых удержаний

Журнал оплачиваемых неявок

- Начисление отпуска
- Начисление по больничному
- Невыходы
- Оплата по среднему заработку
- Отпуска по уходу за ребенком

Табель учета рабочего времени

- Индивидуальные графики работы
- Заявление-расчет в ФСС
- Анализ неявок
- Сторнирование

Журнал выплаты зарплаты

- Зарплата к выплате
- Платежное поручение исходящее
- Расходный кассовый ордер
- Возврат сотрудником долга по зарплате (ПКО)
- Выплата зарплаты расходными ордерами
- Списание депонентов
- Платежное поручение входящее
- Договоры займа с сотрудниками
- Банковские карточки сотрудников

Отчеты

Расчетные листки

Расчетная ведомость

Структура задолженности

Свод начисленной зарплаты

Книга депонентов

Табель учета рабочего времени (П-5)

Расчетная ведомость П-6

Расчетная ведомость П-7

Выработка работников организаций

Отчет по исполнительным листам

Отчет по оплате отпусков

Отчет по оплате больничных

Ведомость расчетов с ФСС

См. также

Договоры ГПХ

Исполнительные листы

Перерасчет зарплаты

Тарифные разряды

Технологические операции

Показатели схем мотивации

Основные начисления

Взносы в фонды

Удержания

Постоянные начисления

Постоянные удержания

Рисунок 3.3 – Розрахунок зарплати

Звітність в 1С Бухгалтерії представлено на рисунку 3.4

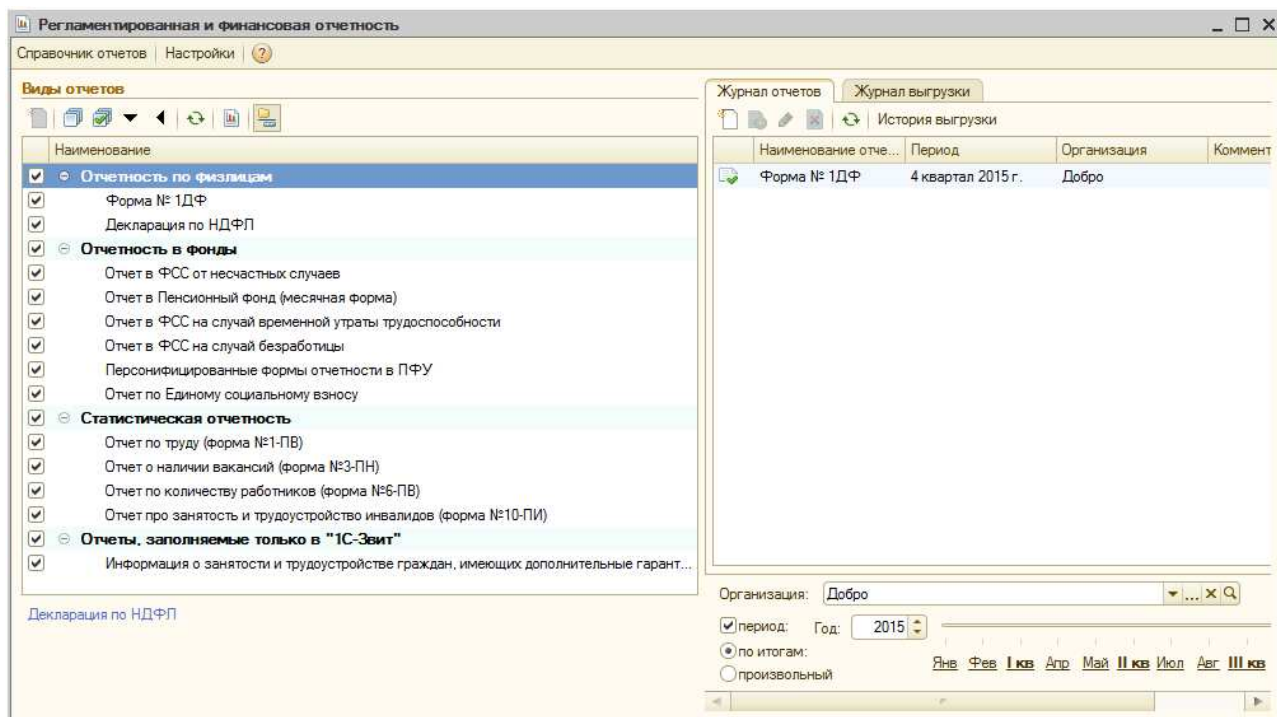


Рисунок 3.4 – Звітність

Приклад розрахункового листка в 1С Бухгалтерії представлено на рисунку 3.5

Начисления

Взносы

НДФЛ

Удержания

Взносы ФОТ

Проводки

Расчетный листок

Працівник: Іванцова Лілія Євгенівна (осн.)

Код по ДРФО: 5124123123

Вид	Период	Отработано		Оплачено		Сумма	Вид	Период	Сумма
		Дни	Часы	Дни	Часы				
1. Начислено						2. Удержано			
Оклад по дням	1-31 Сер 15	20	160	20		5 000,00	ЄСВ	1-31 Сер 15	309,09
Індексація зарплати	1-31 Сер 15					3 585,79	Військовий збір	1-31 Сер 15	128,79
							ПДФО обчислений	1-31 Сер 15	1 241,51
Всього начислено						8 585,79	Всього удержано		1 679,39
Борг за організацією на початок місяця:							Борг за організацією на кінець місяця:		
							6 906,40		

Комментарий:

Табель П-5 (бланк)

Печать ▾

OK

Записать

Закрыть

Рисунок 3.5 – Розрахунковий листок

Висновки до розділу

Впровадження інформаційної системи продемонструвало, що вона значно підвищує ефективність управління роботою зоопарку за рахунок зменшення

обсягу ручної роботи. Це скорочення ручних завдань зменшує навантаження на касирів, продавців і бухгалтерів[41].

РОЗДІЛ 4. ТЕХНІКО-ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ РОЗРОБКИ

Розробка інформаційної системи (ІС) для зоопарку є технічно виправданою з наступних причин:

- Підвищення ефективності та продуктивності: ІС автоматизує багато ручних процесів зоопарку, наприклад керування даними про тварин, оформлення квитків для відвідувачів і фінансовий облік. Це звільнить час персоналу, щоб зосередитися на більш стратегічних завданнях і підвищити загальну продуктивність.
- Покращене прийняття рішень: ІС надасть керівництву зоопарку дані в реальному часі та аналітику для підтримки прийняття кращих рішень. Це включає дані про здоров'я тварин, поведінку відвідувачів і фінансові результати.
- Покращене обслуговування клієнтів: ІС надасть відвідувачам онлайн-доступ до інформації та послуг, таких як покупка квитків, карти зоопарків і профілі тварин. Це покращить враження та задоволення відвідувачів.

Розробка ІС для зоопарку є економічно виправданою з наступних причин:

- Економія коштів: ІС автоматизує багато ручних процесів у зоопарку, що зменшить витрати на оплату праці. Це також допоможе визначити сфери, де зоопарк може заощадити гроші, наприклад, зменшивши споживання енергії та відходи.
- Збільшення доходу: ІС допоможе зоопарку збільшити дохід за рахунок покращення маркетингу та продажів, збільшення відвідуваності відвідувачів і розробки нових джерел доходу, таких як онлайн-навчання та віртуальні тури.
- Покращене управління активами: ІС допоможе зоопарку відстежувати та керувати своїми активами, такими як тварини, обладнання та

приміщення. Це допоможе зменшити збитки та покращити загальний стан активів зоопарку.

Висновки до розділу

Загалом розробка ІС для зоопарку є розумною інвестицією, яка забезпечить як технічну, так і економічну вигоду.

РОЗДІЛ 5. ОХОРОНА ПРАЦІ

Охорона праці — це система заходів та засобів, спрямованих на збереження життя та здоров'я людини на підприємстві.

У зоопарку найбільша небезпека для працівників — тварини. За недотримання правил безпеки вони можуть нанести шкоду здоров'ю.

Під час годування та догляду за тваринами робітник наражається на певний рівень ризику. Для мінімізації цього ризику потрібні відповідні засоби індивідуального захисту. Наприклад рукавиці, чоботи, маски, захисні окуляри та засоби зв'язку. Спеціальне обладнання також сприяють безпеці працівників зоопарку. Модифіковані розсувні двері та двері-гільйотини допомагають годувати небезпечних тварин.

Існують й інші ризики для здоров'я, пов'язані із захворюваннями дихальних шляхів, такими як астма та алергії. Для запобігання алергії робітники повинні здавати тест на шкірну алергію.

Ще одна потенційна небезпека для співробітників зоопарку — це ті, хто працює з отруйними тваринами. У цих тварин є укуси або укуси, які піддають жертву впливу токсину. В таких випадках зоопарк повинен мати протиотрута. Якщо протиотрута недоступна, може знадобитися екстрена госпіталізація або фасціотомія.

Також при роботі с тваринами працівники можуть заразитись зоонозним захворюванням. Для запобігання цьому, потрібен регулярний огляд тварин на ознаки захворювання [36].

Також тварини становлять небезпеку для відвідувачів. Найкращий спосіб захистити відвідувачів зоопарку від тварин — вольєр. Вольєр також повинен відображати видові характеристики тварини, такі як здатність дертися, плавати, стрибати, літати чи копати. Також на вольєрах повинні бути попереджувальні знаки.

За для безпеки відвідувачів вони мають дотримуватися деяких правил та рекомендацій для їх безпеки.

- Діти мають бути під наглядом дорослих.
- Селфі-палиці, повітряні кулі, велосипеди, скейтборди, ковзани, скляна тара та інші предмети, які можуть травмувати тварин або людей, заборонені на території Зоопарку.
- Не фотографувати тварин із застосуванням спалахів
- Не кричати, оскільки крики можуть потурбувати тварин та інших гостей.
- У Зоопарк не допускаються домашні тварини, крім службових тварин.
- Сп'яніння може загрожувати вашому життю. Якщо ви бачите відвідувача, який здається п'яним або вживає наркотики, зверніться за допомогою до співробітника зоопарку.
- Куріння дозволено тільки на спеціально відведених місць для куріння на відкритому повітрі.
- Несанкціоновані дрони над територією Зоопарку заборонені.
- Якщо ви бачите, що гість стоїть або піднімається на бар'єри виставки, кричить на тварин або дратвує їх, кидає предмети в експонати або смітить на території Зоопарку, будь ласка, зв'яжіться зі співробітником Зоопарку.
- З метою вашої безпеки та для збереження цінної колекції рослин та дерев, будь ласка, не зривайте рослини та листя.
- Годування тварин заборонено по всій території Зоопарку, за винятком дозволених Зоопарком місць. Це необхідно для забезпечення здоров'я та безпеки тварин. Якщо ви бачите, що гість годує тварину, зверніться до співробітника зоопарку.
- Якщо ви коли-небудь побачите людину у безпосередній небезпеці, негайно зв'яжіться з персоналом зоопарку. Ніколи не намагайтеся врятувати, якщо йдеться про небезпечні тварини.

- Не смітити
- Відвідувачам заборонено відвідувати службові приміщення.
- Не заважати іншим гостям або роботі зоопарку [37].

Інструкція з охорони праці при роботі на персональному комп'ютері.

1. Загальні положення

1.1. Дія інструкції поширюється на всі підрозділи підприємства, де виконують роботи з персональним комп'ютером (ПК) .

1.2. Інструкція розроблена відповідно до Положення про розробку інструкцій з охорони праці, затвердженого наказом Держпраці від 29.01.1998 № 9, Типового положення про порядок проведення навчання і перевірки знань з питань охорони праці, затвердженого наказом Держпраці від 26.01.2005 № 15, Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями, затверджених наказом Мінсоцполітики від 14.02.2018 № 207, Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПіН 3.3.2.007-98, затверджених постановою Головного державного санітарного лікаря України від 10.12.1998 № 7, Загальних вимог стосовно забезпечення роботодавцями охорони праці працівників, затверджених наказом ДСНС від 25.01.2012 № 67 (НПАОП 0.00-7.11-12).

1.3. За цією інструкцією працівника, який використовує персональний комп'ютер (далі — користувач), інструктують перед початком роботи (первинний інструктаж), а потім через кожні 6 місяців (повторний інструктаж). Результати інструктажу заносять до Журналу реєстрації інструктажів з питань охорони праці на робочому місці (у журналі має бути підпис особи, яка інструктує, та користувача).

1.4. Користувач зобов'язаний дбати про особисту безпеку і здоров'я, а також про безпеку і здоров'я довколишніх при виконанні будь-яких робіт, а також під час перебування на території підприємства.

1.5. До роботи на персональному комп'ютері допускають осіб, які пройшли інструктажі з питань охорони праці та пожежної безпеки.

1.6. Користувач зобов'язаний:

- виконувати правила внутрішнього трудового розпорядку;
- не допускати за своє робоче місце сторонніх осіб;
- не виконувати вказівок, які суперечать правилам охорони праці та пожежної безпеки;
- знати правила надання домедичної допомоги;
- знати розташування та вміти користуватись первинними засобами пожежогасіння;
- вміти працювати з ПК.

1.7. Основні небезпечні та шкідливі виробничі фактори, що можуть впливати на користувача:

- підвищений рівень статичної електрики;
- нерівномірність розподілу яскравості в полі зору;
- підвищена яскравість світлового зображення;
- ураження електричним струмом;
- напруження зору та уваги;
- тривалі статичні навантаження.

1.8. У приміщеннях із ПК має бути природне і штучне освітлення.

1.9. При розміщенні робочих місць необхідно унеможливити пряме засвічування екрана природним освітленням.

1.10. При природному освітленні слід передбачити наявність сонцезахисних засобів (плівка, жалюзі, штори тощо).

1.11. Світлові відблиски із клавіатури, екрана та інших частин ПК у напрямку очей користувача неприпустимі.

1.12. Основним обладнанням робочого місця є ПК або ноутбук, монітор, клавіатура, маніпулятор, робочий стіл, стілець (крісло).

1.13. При розміщенні елементів робочого місця слід враховувати:

- робочу позу користувача;
- простір для розміщення користувача;
- можливість огляду елементів робочого місця;
- можливість огляду простору поза межами робочого місця;
- можливість робити записи, розміщувати на робочому столі документацію та матеріали, які використовує користувач.

1.14. Розміщення елементів робочого місця не має заважати рухам та переміщенню для експлуатування ПК.

1.15. Монітор встановлюють так, щоб відстань від поверхні екрана до очей користувача була 600-700 мм залежно від розміру екрана.

1.16. Клавіатуру розміщують на робочому або окремому столі на відстані 100-300 мм від краю з боку користувача. Положення клавіатури та кут її нахилу залежить від побажання користувача (як правило, в межах 5-15°). Не допускати хитання клавіатури.

1.17. Конструкція робочого столу має бути такою, щоб оптимально розмістити на робочій поверхні обладнання, що використовують, з урахуванням кількості, розмірів, конструктивних особливостей і характеру його роботи.

1.18. Крісло має забезпечувати підтримування раціональної робочої пози під час виконання основних виробничих операцій та можливість зміни пози. Тип робочого крісла обирають залежно від характеру та тривалості роботи.

1.19. Раціональна поза користувача:

- ступні розташовані на підлозі або на підставці для ніг;
- стегна зорієнтовані у горизонтальній площині;
- верхні ділянки рук вертикальні;

- кут ліктьового суглоба у межах 70—90°;
- зап'ястя зігнуті під кутом не більше ніж 20°;
- нахил голови у межах 15—20°, а часті її повороти виключені.

1.20. Для забезпечення оптимальної робочої пози користувача необхідно:

- засоби праці, з якими користувач має тривалий або найбільш частий зоровий контакт, розмістити у центрі зони зорового спостереження та моніторного поля;
- забезпечити відстань близько 500 мм між найважливішими засобами праці, з якими користувач працює найчастіше.

1.21. ПК встановлювати на рівній твердій поверхні (столі). Не дозволено встановлювати ПК та оргтехніку на хитких підставках чи на похилій поверхні.

1.22. ПК не встановлювати впритул до стіни, перегородки тощо. Не допускати загородження вентиляційних отворів ПК сторонніми предметами.

1.23. Розетка біля ПК має бути в доступному місці, щоб в аварійних випадках можна було своєчасно його відімкнути. Не рекомендовано використовувати подовжувачі.

1.26. Під час переміщення ПК, периферійних пристроїв витягти вилку живлення з розетки.

1.27. Не допускати ушкодження чи модифікування шнура живлення. Заборонено ставити важкі речі на шнур живлення, тягнути чи надмірно перегинати його, скручувати та зав'язувати шнур живлення у вузол.

1.28. ПК підключити до електромережі лише за допомогою справних штепсельних з'єднань та електророзеток заводського виробництва.

1.29. Штепсельні з'єднання та електророзетки мають бути зі спеціальними контактами для під'єднання нульового захисного провідника. Їхня конструкція має забезпечувати з'єднання нульового захисного провідника раніше, ніж з'єднання

фазового та нульового робочого провідників. Порядок роз'єднань при вимкненні має бути зворотнім.

1.30. Заборонено підключити електрообладнання до звичайної двошнурової електромережі.

1.31. За невиконання цієї інструкції працівники несуть відповідальність згідно з чинним законодавством.

2. Вимоги безпеки перед початком роботи

2.1. Оглянути робоче місце і навести на ньому лад; впевнитись, що на ньому немає сторонніх предмети, все обладнання і блоки ПК з'єднані з системним блоком з'єднувальними шнурами.

2.2. Перевірити надійність встановлення апаратури на робочому столі. Монітор не має стояти на краю стола. Повернути монітор так, щоб було зручно дивитися на екран — під прямим кутом (а не збоку) і трохи зверху вниз; при цьому екран має бути трохи нахиленим — нижній край ближче до користувача.

2.3. Перевірити загальний стан апаратури, справність електропроводки, з'єднувальних шнурів, штепсельних вилок, розеток, заземлення захисного екрана.

2.4. Вставити вилку в розетку і впевнитися, що вона міцно тримається. Заборонено вставляти і виймати вилку мокрими руками.

2.5. Відрегулювати та зафіксувати висоту крісла та зручний для користувача нахил спинки.

2.6. За потреби приєднати до комп'ютера необхідну апаратуру (принтер, сканер тощо). Усі кабелі, що з'єднують системний блок із іншими пристроями, вмикати та вимикати лише при вимкненому комп'ютері.

2.7. Відрегулювати яскравість свічення, контрастність монітора.

2.8. Про всі виявлені несправності інформувати керівника робіт і не братися до роботи, доки їх не буде усунено.

3. Вимоги безпеки під час виконання роботи

3.1. Під час роботи на ПК:

- стійко встановити клавіатуру на робочому столі, не допускаючи її хитання, водночас передбачити можливість її поворотів та переміщень;
- якщо в конструкції клавіатури не передбачено простору для упору долонь, клавіатуру розміщують на відстані не менше 100 мм від краю столу в оптимальній зоні моніторного поля;
- під час роботи на клавіатурі сидіти рівно, не напружуватися;
- щоб зменшити несприятливе навантаження на користувача при роботі з комп'ютерною мишею (вимушена поза, необхідність постійно контролювати якість дій), забезпечити велику вільну поверхню столу для переміщення комп'ютерної миші та зручного упору ліктьового суглоба;
- періодично при вимкненому комп'ютері прибирати пил із поверхонь апаратури спеціальними серветками.

3.2. При роботі з ПК заборонено:

- самостійно розбирати та ремонтувати системний блок (корпус ноутбука), монітор, клавіатуру, комп'ютерну мишу тощо;
- встромляти сторонні предмети до вентиляційних отворів ПК, ноутбука або монітора;
- ставити на системний блок ПК та периферійні пристрої металеві предмети, ємкості з водою (вази, горщики для квітів, склянки), оскільки через потрапляння води у середину апарата може статися пожежа або ураження електрострумом.

3.3. Тривалість безперервної роботи за ПК не має перевищувати 2 год. Після цього необхідно зробити 15-хвилинну перерву.

3.4. Якщо виник зоровий дискомфорт або інші неприємні відчуття, необхідно зробити нетривалу перерву.

3.5. Для зниження нервово-емоційного напруження, стомлення зорового аналізатора, поліпшення мозкового кровообігу, подолання несприятливих наслідків гіподинамії, запобігання втомі доцільно під час декількох перерв виконувати комплекс вправ.

4. Вимоги безпеки після закінчення роботи

4.1. Зберегти інформацію.

4.2. Вимкнути ПК, монітор чи ноутбук.

4.3. Вимкнути стабілізатор, якщо комп'ютер під'єднаний до мережі через нього.

4.4. Прибрати робоче місце.

5. Вимоги безпеки в аварійній ситуації

5.1. Аварійні та небезпечні ситуації під час виконання роботи на ПК можуть виникнути у разі: короткого замикання, перевантаження блоку живлення системного блоку, перегрівання, пожежі, поломки крісла тощо.

5.2. У разі виникнення аварії або ситуації, що може привести до аварії, нещасного випадку, негайно від'єднати ПК від електромережі, повідомити про інцидент керівникові.

5.3. Не допускати в небезпечну зону сторонніх осіб.

5.4. Якщо стався нещасний випадок, зберегти обстановку в робочій зоні та устаткування у такому стані, в якому вони були на момент події (якщо це не загрожує життю і здоров'ю інших працівників і не призведе до більш тяжких наслідків). Поінформувати про подію керівника робіт (іншу відповідальну особу підприємства) та в подальшому керуватися його вказівками. Вжити заходів, щоб запобігти подібним випадкам у подальшому.

5.5. У разі виникнення пожежі (ознак горіння), повідомити керівнику та, за потреби, викликати оперативно-рятувальну службу за телефоном 101 або 112 (назвати адресу та місце виникнення пожежі, наявність людей, повідомити своє

прізвище) та вжити можливих заходів для евакуювання людей, гасіння (локалізації) пожежі наявними засобами пожежогасіння. Пам'ятати, що гасіння електротехнічних пристроїв, які перебувають під напругою, виконувати лише після їх попереднього від'єднання від електромережі. Гасити за допомогою вуглекислотних або порошкових вогнегасників, а в окремих випадках — сухим піском.

5.6. За потреби надати потерпілому домедичну допомогу згідно з інструкцією, що діє на підприємстві. У разі подальшого погіршення самопочуття потерпілого, не припиняючи надання домедичної допомоги, викликати за телефоном «103» швидку медичну допомогу.

5.7. Виконувати вказівки керівника робіт для ліквідування небезпеки.

Інструкція з охорони праці при роботі на персональному комп'ютері підготовлена на основі матеріалу [38].

Висновки до розділу

При дотриманні всіх цих правил та рекомендацій зменшується ризик для здоров'я як для персоналу, так і для відвідувачів.

РОЗДІЛ 6. ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

Охорона навколишнього середовища (НС) – це комплекс науково-обґрунтованих міжнародних,

державних, регіональних, адміністративно-господарських, політичних, економічних, громадських заходів, спрямованих на підтримання фізичних, хімічних і екологічних параметрів природного середовища в межах, які забезпечують нормальні умови життєдіяльності людини та можливість збереження і зміцнення її здоров'я.

Екологічні та природоохоронні відносини в суспільстві регулюються Конституцією України, в 13-ій статті якої зазначено: «Земля, її надра, атмосферне повітря, водні та інші природні ресурси є всенародним надбанням, а власником їх є Держава». Основним нормативно-правовим актом України в сфері охорони НС є закон України «Про охорону навколишнього середовища», уведений в дію Постановою Верховної Ради № 1268-12 від 26.06.1991 року. Верховною Радою України прийняті також закони, кодекси та інші нормативно-правові акти, метою яких є регулювання процесу використання та охорони природних ресурсів. Насамперед, це закони України «Про охорону атмосферного повітря» (1992), «Про природно-заповідний фонд» (1992), «Про забезпечення санітарного та епідеміологічного благополуччя населення» (1994); «Про використання ядерної енергії та радіаційну безпеку» (1995), «Про екологічну експертизу» (1995), «Про відходи» (1998), «Про рослинний світ» (1999), «Про зону надзвичайної екологічної ситуації» (2000), «Про тваринний світ» (2001), «Про питну воду та питне водопостачання» (2002), «Про основи національної безпеки України» (2003); кодекси «Про надра» (1994), «Лісовий кодекс України» (1994), «Водний кодекс України» (1994), «Земельний кодекс України» (1994) тощо.

У системі раціонального використання природних ресурсів та охорони довкілля важливе місце займають екологічна стандартизація, екологічне та гігієнічне нормування, екологічна експертиза.

Екологічні стандарти – це нормативно-технічна документація, в якій визначені загальні екологічні вимоги до конкретних видів природокористування. Екологічне нормування – це наукова, правова, адміністративна діяльність, спрямована на обґрунтування і затвердження гранично допустимих екологічних нормативів, при дотриманні яких не відбудеться деградація екосистеми, гарантується збереження біологічного різноманіття довкілля та безпека життєдіяльності населення.

Гігієнічне нормування – це наукове обґрунтування і впровадження в законодавчому порядку безпечних для людини рівнів дії шкідливих факторів НС.

Метою екологічної експертизи є запобігання негативному впливові антропогенної діяльності на стан НС та здоров'я людей, оцінка ступеня екологічної безпеки господарської діяльності й екологічної ситуації на окремих територіях та об'єктах. До об'єктів екологічної експертизи належать проекти законодавчих, нормативно-правових актів; документація щодо впровадження нових технологій, матеріалів, продукції, використання яких може призвести до порушення екологічних нормативів та створення загрози здоров'ю людей; діючі об'єкти та комплекси, що негативно впливають на стан НС; несприятливі екологічні ситуації в окремих регіонах країни.

До заходів охорони атмосферного повітря, води та ґрунту належать законодавчі, гігієнічні, технологічні, санітарно-технічні та планувальні.

Законодавчі заходи регулюють екологічну політику уряду, спрямовану на запобігання забруднення повітряного басейну, води та ґрунту шкідливими речовинами. Планування, забудова та розвиток населених місць повинні

здійснюватись з використанням вимог щодо екологічної безпеки з обов'язковим проведенням еколого-гігієнічної експертизи.

Суттю гігієнічних заходів є встановлення нормативів екологічної безпеки: нормуються гранично допустимі рівні викидів у повітря речовин із стаціонарних джерел, вміст шкідливих речовин у відпрацьованих газах. У ґрунті допускається такий вміст екзогенних хімічних речовин (ЕХР), який при прямому контакті з людиною або в процесі міграції за екологічними ланцюгами не порушує процесів самоочищення ґрунту, не має шкідливого впливу на санітарні умови проживання та стан здоров'я населення.

Технологічні заходи мають на меті використання екологічно чистого виробництва: замкнутих технологічних циклів та безперервного виробництва; принципову зміну технології. Технологічні заходи призначені також регулювати процеси утворення та знешкодження відходів, що можуть забруднювати ґрунт. До них належать: зменшення утворення відходів, токсичних та потенційно небезпечних для довкілля; скорочення кількості відходів, що підлягають утилізації; впровадження безпечних для НС технологій вторинного використання та утилізації відходів.

Санітарно-технічні заходи включають використання методів санітарного очищення населених місць, а саме – збір, тимчасове зберігання, вивезення, знешкодження та утилізацію твердих і рідких відходів.

Планувальні заходи у містобудівництві визначаються генпланом міста, схемами планування районів у відповідності до діючих санітарних норм і правил. До них належать: організація санітарно-захисних зон навколо забруднюючих підприємств; раціональне розташування житлової зони по відношенню до промислової з урахуванням напрямку панівного вітру та його швидкості, рельєфу місцевості та несприятливих метеорологічних ситуацій щодо розсіювання промислових викидів; фонових концентрацій у повітрі шкідливих речовин,

можливості температурної інверсії та утворення туманів; озеленення міста та перспектив його подальшого розвитку.

Інформація про охорону навколишнього середовища підготовлена на основі матеріалу [39].

Висновки до розділу

Зосередження уваги на захисті від надзвичайних ситуацій більше не обмежується окремими країнами, оскільки вирішення більшості сучасних екологічних проблем потребує спільних зусиль. Ефективні рішення сучасних екологічних проблем можуть бути досягнуті лише шляхом колективних дій та міжнародної співпраці.

ВИСНОВКИ

У дипломному проекті розроблена інформаційна система управління зоопарком. Поставлена ціль роботи у підвищенні ефективності за рахунок впровадження сучасних систем, які дозволять мінімізувати витрати та збільшити прибуток виконана.

Було проведено дослідження моделей управління підприємствами з подальшим детальним аналізом структури та діяльності зоопарку. Згодом були сформульовані конкретні вимоги до інформаційної системи. На основі цих вимог було розроблено концептуальну основу, яка окреслює передбачувану функціональність інформаційної системи.

Розроблена концепція інформаційної системи, проведено обстеження підприємства, проведено аналіз інформаційних процесів, сформовано вимоги до проєктованої системи, визначено цілі та завдання проєкту. Після цього розробка технічного проєкту передбачала розробку комплексних рішень щодо вибору стандартних програмних засобів, проєктування бази даних, представлення концептуальних, логічних і фізичних моделей. Було розроблено математичні алгоритми для вирішення конкретних завдань, а також надано точні визначення технічних і програмних засобів системи. Створення експлуатаційної документації супроводжувалося ретельним тестуванням функціональності системи. Представлені правила та рекомендації щодо безпеки відвідувачів та працівників у зоопарку, а також правила безпеки за комп'ютером.

Актуальність та новизна роботи полягає у вдосконаленні системи управління зоопарком для ефективної роботи зоопарком шляхом інтеграції сучасних інформаційних технологій і комп'ютерних систем.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Birkinshaw, J., & Goddard, J. (2009). *What Is Your Management Model?* Harvard Business Review, 50(2).
- [2] Dr. Robert Lloyd and Dr. Wayne Aho. *The Four Functions of Management*
- [3] Назарчук, Т.В., & Косіюк, О. М. (2016). *Менеджмент організацій*. Київ: Центр учбової літератури
- [4] Кицак Т.Г (2008) *Формування корпоративної культури в процесі розвитку соціально-трудових відносин на підприємстві*. Київ. 20 с
- [5] Ecocation (2023). *History of Zoos: Past, Present & Future*. Взято з <https://ecocation.org/history-of-zoos>
- [6] Лановенко О.Г., Остапішина О.О. (2013). *Словник – довідник з екології*. Херсон.
- [7] The Balance Careers (б.д.). *Zoo Director*. Взято з <https://www.thebalancecareers.com/zoo-director-125935>
- [8] Soft Play (б.д.). *How to Boost Attendance at Your Zoo*. Взято з <https://www.softplay.com/blog/how-to-boost-attendance-at-your-zoo/>
- [9] Fresno Chaffee Zoo (б.д.). *COO Deputy Director Job Description*. Взято з <https://www.fresnochaffeezoo.org/wp-content/uploads/2015/09/COO-Deputy-Director-02192017.pdf>
- [10] Wikipedia (2023). *Economist*. Взято з <https://en.wikipedia.org/wiki/Economist>
- [11] Arizona Foundation for the Future of Aging (б.д.). *Plumber Position*. Взято з <http://www.azfa.org/job/plumber-2/>
- [12] Arizona Foundation for the Future of Aging (б.д.). *Electrician Position*. Взято з <http://www.azfa.org/job/electrician/>
- [13] The Balance Careers (2019). *Zoo Veterinarian*. Взято з <https://www.thebalancecareers.com/zoo-veterinarian-125943>
- [14] The Balance Careers (2019). *Animal Nutritionist*. Взято з <https://www.thebalancecareers.com/animal-nutritionist-125606>

- [15]Newsroom (б.д.). *Zoo Nutritionist Making a Difference Through Exotic Diets Research*. Взято з https://newsroom.unl.edu/announce/snr_alumni/6179/38961
- [16]Workable (б.д.). *HR Manager Job Description*. Взято з <https://resources.workable.com/hr-manager-job-description>
- [17]Wikipedia (б.д.). *Бухгалтерская система..* Взято з https://ru.wikipedia.org/wiki/Бухгалтерская_система#Персональные_системы
- [18]Wikipedia (б.д.). *GnuCash*. Взято з <https://ru.wikipedia.org/wiki/GnuCash>
- [19]Wikipedia (б.д.). *KMyMoney*. Взято з <https://ru.wikipedia.org/wiki/KMyMoney>
- [20]Wikipedia (б.д.). *Tryton*. Взято з <https://ru.wikipedia.org/wiki/Tryton>
- [21]Wikipedia (б.д.). *Microsoft Dynamics AX*. Взято з https://ru.wikipedia.org/wiki/Microsoft_Dynamics_AX
- [22]Wikipedia (б.д.). *GnuCash*. Взято з <https://en.wikipedia.org/wiki/GnuCash>
- [23]Business Support (б.д.). *1С Бухгалтерия для Украины*. Взято з <https://business-support.pro/1c-buh-dlia-ukrainy/>
- [24]Wikipedia (б.д.). *MySQL*. Взято з <https://uk.wikipedia.org/wiki/MySQL>
- [25]Wikipedia (б.д.). *SQL*. Взято з <https://uk.wikipedia.org/wiki/SQL>
- [26]Wikipedia (б.д.). *JavaScript*. Взято з <https://uk.wikipedia.org/wiki/JavaScript>
- [27]Wikipedia (б.д.). *HTML*. Взято з <https://uk.wikipedia.org/wiki/HTML>
- [28]Wikipedia (б.д.). *CSS*. Взято з <https://uk.wikipedia.org/wiki/CSS>
- [29]Guru99 (б.д.). *Data Modeling: Conceptual and Logical Data Model*. Взято з <https://www.guru99.com/data-modelling-conceptual-logical.html>
- [30]Wikipedia (б.д.). *XAMPP*. Взято з <https://uk.org/wiki/XAMPP>
- [31]Wikipedia (б.д.). *Java*. Взято з <https://uk.org/wiki/Java>
- [32]Wikipedia (б.д.). *XHTML*. Взято з <https://uk.org/wiki/XHTML>
- [33]Wikipedia (б.д.). *IntelliJ IDEA*. Взято з https://uk.org/wiki/IntelliJ_IDEA
- [34]Wikipedia (б.д.). *GlassFish*. Взято з <https://en.org/wiki/GlassFish>

- [35]Wikipedia (б.д.). *United States Environmental and Occupational Health in Zoos*.
Взято з
https://en.wikipedia.org/wiki/United_States_environmental_and_occupational_health_in_zoos
- [36]Wikipedia (б.д.). *Діаграма прецедентів*. Взято з
https://uk.wikipedia.org/wiki/Діаграма_прецедентів
- [37]Louisville Zoo (б.д.). *Tips and safety guidelines..* Взято з
<https://louisvillezoo.org/plan/know-before-you-go/>
- [38]Pro-OP (2021). *Інструкція з охорони праці при роботі на персональному комп'ютері*. Взято з <https://pro-op.com.ua/article/485-nstruktsya-z-ohoroni-prats-pri-robot-na-personalnomu-kompyuter>
- [39]BSMU (2016). *Основи охорони довкілля в Україні*. Взято з
<https://www.bsmu.edu.ua/blog/4413-osnovi-ohoroni-dovkillya-v-ukraini/>
- [40]Васильченко А.С., Гайдаєнко О.В (2023). *Аналіз моделей управління підприємствами та проектування інформаційної системи для зоопарку*. IV Всеукраїнській науково-практичній інтернет-конференції «Інформаційні технології: моделі, алгоритми, системи»
- [41]Васильченко, А. С., Ажищев В.Ф. (2022). *Розробка інформаційної системи управління зоопарком*. Взято з <http://eir.nuos.edu.ua/handle/123456789/5966>

ДОДАТКИ

Додаток А – Інструкція користувача

Для того, щоб нарахувати зарплату автоматично, відкрийте документ «Нарахування зарплати працівникам організації» та скористайтесь кнопкою «Заповнити та розрахувати все». Поле «Норма днів та годин на місяць» заповниться автоматично на підставі відомостей «Регламентований виробничий календар». Якщо фактично відпрацьований працівником час відрізняється від місячної норми, слід зазначити його у стовпці «Відпрацьовано».

Щоб розрахувати документ, використовуються дані, зазначені у плані видів розрахунку «Нарахування організації», для відображення в обліку – дані довідника «Способи відображення зарплати в регламентованому обліку».

N	К...	О...	Працівник (призначення)	Підрозділ організації	Нарахування	Період	Базовий період	Результат	Розмір	Відпрацьовано	Норма за місяць	Оплатено	Спосіб відображення в ...
						Дата початку...	Дата закінч...	Початок...	Кінець...	Днів	Годин	днів/годин	
1	✓		Абдулов Юрій Володимирович (осн.)	Адміністрація	Оклад по днях	01.12.2018	31.12.2018	01.12.2018	31.12.2018	3 200,00	20,00	158,00	20,00
2	✓		Абдулов Юрій Володимирович (осн.)	Адміністрація	Доплата до мініма...	01.12.2018	31.12.2018	01.12.2018	31.12.2018	523,00	20,00	158,00	
3	✓		Абдулов Юрій Володимирович (осн.)	Адміністрація	Індексація зарплати	01.12.2018	31.12.2018	01.12.2018	31.12.2018	276,62	20,00	158,00	
4	✓		Аксёнов Артур П...	Адміністрація	Оклад по днях	01.12.2018	31.12.2018	01.12.2018	31.12.2018	3 500,00	20,00	158,00	20,00
5	✓		Аксёнов Артур П...	Адміністрація	Індексація зарплати	01.12.2018	31.12.2018	01.12.2018	31.12.2018	276,62	20,00	158,00	

Рисунок А.1 – Нарахування зарплати

Податок із доходів фізичних осіб буде розрахований автоматично з урахуванням наданих працівникам податкових соціальних пільг. На закладці ПДФО відображено і розрахунок військового збору по кожному співробітнику.

Нарахування зарплати працівникам організації: Проведено *

Номер: Д0000000008 від: 31.12.2018 00:00:00 Місяць нарахуван...: Грудень 2018

Організація: Добро Норма за місяць: днів 20,00 годин 158,00

Підрозділ: Адміністрація

Відповідальний: Адміністратор Віктор Сергійович Базовий період індексації соц. виплат: ..

Зарплати

N	Tab. №	Працівник (основне призначення)
1	0000004	Абадулов Юрій Володимирович (осн.)
2	0000001	Іванцова Лілія Євгенівна (осн.)
3	0000002	Богословський Федір Михайлович (осн.)

Нарахування Внески ПДФО Утримання Внески ФОП Проводки Розрахунковий листок

Перерахунок ПДФО

N	К...	О...	Працівник (основне п...	Податковий період	Доход ПДФО	Вид ставки	Збільшена ставка	Доход	Податок	Пільга	Кількість...	Доход (повний)
1	✓	✓	Богословський Федір...	Грудень 2018	101	Основна	Збільшена	4 776,62	859,79			4 776,62
2	✓	✓	Богословський Федір...	Грудень 2018	В3	Військовий збір	Військовий збір	4 776,62	71,65			4 776,62
3	✓	✓	Аксенов Артур Петр...	Грудень 2018	101	Основна	Збільшена	3 776,62	679,79			3 776,62
4	✓	✓	Аксенов Артур Петр...	Грудень 2018	В3	Військовий збір	Військовий збір	3 776,62	56,65			3 776,62
5	✓	✓	Абадулов Юрій Володи...	Грудень 2018	101	Основна	Збільшена	3 999,62	719,93			3 999,62
6	✓	✓	Абадулов Юрій Володи...	Грудень 2018	В3	Військовий збір	Військовий збір	3 999,62	59,99			3 999,62
7	✓	✓	Горбатова Ірина Іван...	Грудень 2018	101	Основна	Збільшена	400,40	72,07			400,40

Рисунок А.2 – Податок із доходів

Суми нарахованого єдиного соціального внеску можна переглянути на закладці «Внески ФОП». Зверніть увагу, що розрахунок виконується з шістьма знаками після коми, що дозволяє уникнути спотворення результатів заокруглення.

Нарахування зарплати працівникам організації: Проведено *

Номер: Д0000000008 від: 31.12.2018 00:00:00 Місяць нарахуван...: Грудень 2018

Організація: Добро Норма за місяць: днів 20,00 годин 158,00

Підрозділ: Адміністрація

Відповідальний: Адміністратор Віктор Сергійович Базовий період індексації соц. виплат: ..

Зарплати

N	Tab. №	Працівник (основне призначення)
1	0000004	Абадулов Юрій Володимирович (осн.)
2	0000001	Іванцова Лілія Євгенівна (осн.)
3	0000002	Богословський Федір Михайлович (осн.)

Нарахування Внески ПДФО Утримання Внески ФОП Проводки Розрахунковий листок

По податку По сумі Зарплати Розрахувати

N	К...	О...	Працівник (основне ...	Податок	Вид розрахунку	Період	База	База (з обмеження)	Результат	Стаття податкової декларації	Період дії (...	Межа	База (відрахову...	Ставка	Код коєфіціє...	Тип
						Початок	Кінець	в т.ч. відл...	в т.ч. донар. до мін.	в т.ч. донар.	Поч...	Зак...	Мін...	Коєфіціє...	кат. 30	
1	✓	✓	Богословськ... Федір...	ЕСВ ФОП (працівник)	ЕСВ ФОП (оплата ...	01.12.2...	31.12.2...	4 776,62	4 776,62	1 050,860000	Нарахування ЕСВ роботодавця (22 %)	01...	31...	55 ...	0,2200	1
2	✓	✓	Аксенов Артур...	ЕСВ ФОП (працівник)	ЕСВ ФОП (оплата ...	01.12.2...	31.12.2...	3 776,62	3 776,62	830,860000	Нарахування ЕСВ роботодавця (22 %)	01...	31...	55 ...	0,2200	1
3	✓	✓	Іванцова Лілія...	ЕСВ ФОП (працівник)	ЕСВ ФОП (оплата ...	01.12.2...	31.12.2...	5 276,62	5 276,62	1 160,860000	Нарахування ЕСВ роботодавця (22 %)	01...	31...	55 ...	0,2200	1
4	✓	✓	Іванцова Лілія...	ЕСВ ФОП (лікарня)	ЕСВ ФОП (лікарня)	01.12.2...	31.12.2...					01...	31...	55 ...		29

Рисунок А.3 – Внески ФОП

Як тільки програма розрахує всі нарахування, сформуєте проводки на закладці «Проведення»:

Нарахування зарплати працівникам організації. Не проведений *

Діагностика: **Перейти** | Заповнити та розрахувати все | Заповнити | Розрахувати | Очистити | Поради

Номер: D0000000008 від: 31.12.2018 00:00:00 Місяць нарахуван.: **Грудень 2018**

Організація: Добро Норма за місяць: днів 20,00 годин 159,00

Підрозділ: Адміністрація

Відповідальний: Адміністратор Віктор Сергієвич Базовий період індексації соц. виплат:

Таб. №: Працівник (основе призначення)

N	Таб. №	Працівник (основе призначення)
1	0000004	Абаулов Юрий Володимирович (осн.)
2	0000001	Іванцова Лія Свєтєнієвна (осн.)
3	0000002	Богословський Федір Михайлович (осн.)

Нарахування | Внески | ПДФО | Утримання | Внески ФОП | Проводки | Розрахунковий листок

Сформувати проводки

N	Рахунок Дт	Субkonto Дт	Рахунок Кт	Субkonto Кт	Сума	Под. призн. (ПДВ)	Под. призн. витрат (до 01.01.2015)
1	92	Адміністрація	661	Іванцова Лія Свєтєнієвна	5 276,62		(не вик.) Госп. д-сть
2	92	Адміністрація	661	Богословський Федір Михайлович	4 776,62		(не вик.) Госп. д-сть
3	92	Адміністрація	661	Абаулов Юрий Володимирович	3 999,62		(не вик.) Госп. д-сть
4	92	Адміністрація	661	Аксенов Артур Петрович	3 776,62		(не вик.) Госп. д-сть
5	92	Адміністрація	661	Горбатова Ірина Іванівна	400,40		(не вик.) Госп. д-сть

Перевірити ще раз розрахунків по кожному працівнику ви можете в розрахунковому листку.

Нарахування | Внески | ПДФО | Утримання | Внески ФОП | Проводки | Розрахунковий листок

Працівник: Богословський Федір Михайлович (осн.)

Код за ДРФО: 5006086932

Вид	Період	Відпрацьовано		Оплачено		Сума	Вид	Період	Сума
		Дні	Години	Дні	Години				
1. Нараховано							2. Утримано		
Оклад по днях	1-31 Груд 18	20	158	20		4 500,00	Військовий збір	1-31 Груд 18	71,65
Індексація зарплати	1-31 Груд 18					276,62	НДФЛ исчисленный	1-31 Груд 18	859,79
Всього нараховано						4 776,62	Всього утримано		931,44
Борг за організацією на початок місяця:						29 130,70	Борг за організацією на кінець місяця:		32 975,88

Рисунок А.4 – Розрахунковому листку

Якщо йдеться про виплату зарплати у готівковій формі за відомостями, тоді за кнопкою «Відомість» у нижньому правому куті документа можна переглянути друковану форму «Відомості на виплату готівки» і роздрукувати її.

Платіжна відомість № 1 від 04 січня 2019 р. *

Друк 1 екз. Тільки перегляд Зберегти копію...

Керівник Абдулов Ю.В.

Головний бухгалтер Аксьонов А.П.

04 січня 2019 року

Відомість на виплату готівки №1

за грудень 2018 року
Кількість аркушів 2

За цією відомістю виплачено гривень Чотирнадцять тисяч шістсот сімдесят п'ять гривень 06 копійок (14 675 грн 06 коп.)

і депоновано гривень _____

Виплату здійснив _____
(підпис, прізвище, ініціали)

Перевірив бухгалтер _____
(підпис, прізвище, ініціали)

Вставний аркуш до додатка № 1

№ з/п	Табельний номер	Прізвище, ім'я, по батькові	Сума	Підпис про одержання	Примітки
1	2	3	4	5	6
1	0000002	Богословський Федір Михайлович (5006086932)	3 845,18		
2	0000003	Аксьонов Артур Петрович (7712071515)	3 040,18		
3	0000004	Абдулов Юрій	0 000,00		

Рисунок А.5 – Платіжна відомість

Додаток Б – Текст програми

AnimalBean.java

```

package zoo.webzoodb.beans;

@Named
@ViewScoped
public class AnimalBean implements Serializable {
    private List<Animal> animals;
    private Animal animal = new Animal();
    private int employeeid;
    private List<String> sexes = Arrays.asList("М", "Ж");
    DateConverter dateConverter = new DateConverter();

    private List<String> pages = Arrays.asList("employee", "animal", "equipment",
        "feed", "organization", "ration",
        "supplier", "vaccination", "vaccine", "animalSupply",
        "equipmentSupply", "feedSupply", "vaccineSupply");

    @EJB
    AnimalDAO animalDAO;

    @PostConstruct
    public void init() {
        animals = animalDAO.findAll();
    }

    public boolean isValid(Animal a) {
        AtomicBoolean isValid = new AtomicBoolean(true);

        String name = a.getName();
        String species = a.getSpecies();
        String aClass = a.getAclass();
        String order = a.getOrder();
        String birthDate = a.getBirthDate();
        String deadDate = a.getDeadDate();

        Map<String, String> map = new HashMap<>();
        map.put("Кличка", name);
        map.put("Вид", species);
    }
}

```

```

map.put("Класс", aClass);
map.put("Отряд", order);
map.forEach((key, value) ->
    {
        if (!value.isEmpty() && value.trim().isEmpty()) {
            isValid.set(false);
            FacesContext.getCurrentInstance().addMessage(null, new
FacesMessage(key + ": Не должно содержать только пробелы"));
        }
    }
);

if (name.isEmpty()) {
    isValid.set(false);
    FacesContext.getCurrentInstance().addMessage(null, new
FacesMessage("Кличка: Не должно быть пустым"));
}

if (species.isEmpty()) {
    isValid.set(false);
    FacesContext.getCurrentInstance().addMessage(null, new
FacesMessage("Вид: Не должно быть пустым"));
}

if (birthDate.equals("")) {
    FacesContext.getCurrentInstance().addMessage(null, new
FacesMessage("Дата рождения: Введите дату"));
    isValid.set(false);
} else {
    if (dateConverter.convert(birthDate).after(new Date())) {
        FacesContext.getCurrentInstance().addMessage(null, new
FacesMessage("Дата рождения: Нельзя родиться в будущем"));
        isValid.set(false);
    }
}

if (deadDate != null) {

    if (dateConverter.convert(deadDate).after(new Date())) {
        FacesContext.getCurrentInstance().addMessage(null, new

```

```

FacesMessage("Дата смерти: Нельзя умереть в будущем"));
        isValid.set(false);
    }
    if (!birthDate.equals("")) {
        if
(dateConverter.convert(birthDate).after(dateConverter.convert(deadDate))) {
            FacesContext.getCurrentInstance().addMessage(null, new
FacesMessage("Дата: Нельзя умереть до рождения"));
            isValid.set(false);
        }
    }
}

return isValid.get();
}

public void add(List<Employee> employees) {
    Employee employee = employeeByID(employees, employeeid);
    if (isValid(animal)) {
        animal.setEmployee(employee);
        animalDAO.add(animal);
        FacesContext.getCurrentInstance().addMessage(null, new
FacesMessage("Добавление: Успех!"));
    } else
        FacesContext.getCurrentInstance().addMessage(null, new
FacesMessage("Добавление: Неудача!"));
}

public void delete(Animal a) {
    animalDAO.delete(a);
    FacesContext.getCurrentInstance().addMessage(null, new
FacesMessage("Удаление: Успех!"));
}

public void update(Animal a, List<Employee> employees) {
    Employee employee = employeeByID(employees, employeeid);
    if (employee == null)
        employee = a.getEmployee();
    if (isValid(a)) {

```

```

        a.setEmployee(employee);
        animalDAO.update(a);
        FacesContext.getCurrentInstance().addMessage(null, new
FacesMessage("Обновление: Успех!"));
    } else {
        FacesContext.getCurrentInstance().addMessage(null, new
FacesMessage("Обновление: Неудача"));
    }
}

public void search(Long employee, String name, String species, String aClass,
        String order, String sex, String birthDate, String deadDate)
{

    StringBuilder query = new StringBuilder("SELECT a FROM Animal a where");
    Map<String, String> map=new HashMap<>();
    map.put("name",name);
    map.put("species",species);
    map.put("aclass",aClass);
    map.put("order",order);
    map.put("sex",sex);
    map.put("birthDate",birthDate);
    map.put("deadDate",deadDate);
    StringBuilder finalQuery = query;

    map.forEach((key, value)->{
        if (!value.equals(""))
            finalQuery.append(" and a.").append(key).append("=:").append(key);
    });

    if (employee != null)
        query.append(" and a.employee.id=:employee");

    if (query.toString().contains("where and"))
        query = new StringBuilder(query.toString().replace("where and",
"where"));

    if (query.substring(query.lastIndexOf(" ") + 1).equals("where"))
        query = new StringBuilder("SELECT a FROM Animal a");

```

```

        animals = animalDAO.search(query.toString(), employee, map);
        if (animals.isEmpty()) {
            FacesContext.getCurrentInstance().addMessage(null, new
FacesMessage("Поиск: Неудача"));
        } else
            FacesContext.getCurrentInstance().addMessage(null, new
FacesMessage("Поиск: Успех"));
    }
}

```

```

private Employee employeeByID(List<Employee> employees, int employeeID) {

    for (Employee employee : employees) {
        if (employee.getId() == employeeID)
            return employee;
    }
}

```

```

    return null;
}

```

```

public void employeefrompage(ValueChangeEvent event) {
    if (event.getNewValue() == null)
        employeeid = 0;
    else employeeid = ((Long) event.getNewValue()).intValue();
}

```

```

public Animal getAnimal() {
    return animal;
}

```

```

public void setAnimal(Animal animal) {
    this.animal = animal;
}

```

```

public void setAnimals(List<Animal> animals) {
    this.animals = animals;
}

```

```

public List<Animal> getAnimals() {
    return animals;
}

public List<String> getSexs() {
    return sexs;
}

public void setSexs(List<String> sexs) {
    this.sexs = sexs;
}

public List<String> getPages() {
    return pages;
}

public void setPages(List<String> pages) {
    this.pages = pages;
}
}

```

AnimalDAO.java

```

package zoo.webzoodb.dao;

@Stateless
public class AnimalDAO {
    @PersistenceContext
    EntityManager em;

    public List<Animal> findAll() {
        return this.em.createQuery("SELECT a FROM Animal a",
Animal.class).getResultList();
    }

    public void add(Animal animal) {
        em.persist(animal);
    }

    public void delete(Animal a) {

```



```

        Animal animal = em.find(Animal.class, a.getId());
        em.remove(animal);
    }

    public void update(Animal a) {
        em.merge(a);
    }

    public List<Animal> search(String JPQL, Long employee, Map<String, String> map)
    {
        TypedQuery<Animal> query = this.em.createQuery(JPQL, Animal.class);
        if (!(employee == null))
            query.setParameter("employee", employee);

        map.forEach((key, value)->{
            if (!value.equals(""))
                query.setParameter(key, value);
        });

        return query.getResultList();
    }
}

```

Animal.java

```

package zoo.webzodb.entities;

@Entity
public class Animal {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id", nullable = false)
    private Long id;

    @ManyToOne
    @JoinColumn(name = "employeeid")
    private Employee employee;

    @Size(max = 29, message="Кличка: Длина должна быть меньше 30 символов")
    @Pattern(regex = "^(?![A-Яa-яёЁ0-9 -]+)$", message = "Кличка: Недопустимые символы")

```

```

@Column(name = "name")
private String name;

@Size(max = 29,message="Вид: Длина должна быть меньше 30 символов")
@Pattern(regex = " (^$|^ [А-Яа-яёЁ0-9 -]+)$ ",message = "Вид: Недопустимые
символы")
@Column(name = "species")
private String species;

@Size(max = 29,message="Класс: Длина должна быть меньше 30 символов")
@Pattern(regex = " (^$|^ [А-Яа-яёЁ0-9 -]+)$ ",message = "Класс: Недопустимые
символы")
@Column(name = "`class`")
private String aclass;

@Size(max = 29,message="Отряд: Длина должна быть меньше 30 символов")
@Pattern(regex = " (^$|^ [А-Яа-яёЁ0-9 -]+)$ ",message = "Отряд: Недопустимые
символы")
@Column(name = "`order`")
private String order;

@Pattern(regex = " (^$|^ [МмЖж]$) ",message = "Пол: Недопустимые символы")
@Size(max = 1,message="Пол: Длина должна быть меньше 2 символов")
@Column(name = "sex", length = 1)
private String sex;

@Column(name = "birthdate")
private String birthDate;

@Column(name = "deaddate")
private String deadDate;

public String getDeadDate() {
    return deadDate;
}

public void setDeadDate(String deadDate) {
    if (Objects.equals(deadDate, ""))
        this.deadDate = null;
}

```

```
        else
            this.deadDate = deadDate;
    }

    public String getBirthDate() {
        return birthDate;
    }

    public void setBirthDate(String birthDate) {
        this.birthDate = birthDate;
    }

    public String getSex() {
        return sex;
    }

    public void setSex(String sex) {
        this.sex = sex;
    }

    public String getOrder() {
        return order;
    }

    public void setOrder(String order) {
        this.order = order;
    }

    public String getAclass() {
        return aclass;
    }

    public void setAclass(String aclass) {
        this.aclass = aclass;
    }

    public String getSpecies() {
        return species;
    }
}
```

```

public void setSpecies(String species) {
    this.species = species;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public Long getId() {
    return id;
}

public void setId(Long id) {
    this.id = id;
}

public Employee getEmployee() {
    return employee;
}

public void setEmployee(Employee employee) {
    this.employee = employee;
}
}

```

DateConverter.java

```

package zoo.webzodb;

public class DateConverter {

    public Date convert(String str) {
        Date result = null;
        SimpleDateFormat format = new SimpleDateFormat();
        format.applyPattern("yyyy-MM-dd");
        try {
            result =format.parse(str);

```

```

    } catch (ParseException e) {
        System.out.println(e);
    }
    return result;
}
}

```

animal.xhtml

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:h="http://xmlns.jcp.org/jsf/html"
    xmlns:f="http://xmlns.jcp.org/jsf/core"
    xmlns:p="http://xmlns.jcp.org/jsf/passthrough"
    xmlns:pf="http://primefaces.org/ui"
    xmlns:ui="http://xmlns.jcp.org/jsf/facelets">

    <f:view>
        <h:head><title>Animal</title>
            <h:outputStylesheet library="css" name="main.css"/>
            <h:outputStylesheet library="css" name="tables.css"/>
            <h:outputScript library="script" name="libs/jquery-3.5.1.js"/>
            <h:outputScript library="script" name="main.js"/>
        </h:head>
        <h:body>
            <h:panelGroup layout="block" styleClass="container">
                <h:selectOneMenu styleClass="selector" id="animalSelector"
                    onchange="window.location.href = this.options[this.selectedIndex].value">
                    <f:selectItem itemDisabled="true" itemLabel="Таблица"/>
                    <f:selectItems value="#{animalBean.pages}" var="p"
                        itemValue="#{p}.xhtml"/>
                </h:selectOneMenu>
            </h:panelGroup>
            <div class="table-container">
                <h:form id="entityForm">
                    <pf:growl id="growl" sticky="true"/>
                    <h:dataTable id="entityTable" value="#{animalBean.animals}"
                        var="a">
                        <h:column>

```

```

        <f:facet name="header">№</f:facet>
        #{a.id}
    </h:column>
    <h:column>
        <f:facet name="header">Сотрудник*</f:facet>
        <h:selectOneMenu value="#{a.employee.id}"

valueChangeListener="#{animalBean.employeefrompage}">
        <f:selectItems value="#{employeeBean.employees}"
var="e" itemValue="#{e.id}" itemLabel="#{e.firstName}" />
    </h:selectOneMenu>
    </h:column>
    <h:column>
        <f:facet name="header">Кличка*</f:facet>
        <h:inputText value="#{a.name}" />
    </h:column>
    <h:column>
        <f:facet name="header">Вид*</f:facet>
        <h:inputText value="#{a.species}" />
    </h:column>
    <h:column>
        <f:facet name="header">Класс</f:facet>
        <h:inputText value="#{a.aclass}" />
    </h:column>
    <h:column>
        <f:facet name="header">Отряд</f:facet>
        <h:inputText value="#{a.order}" />
    </h:column>
    <h:column>
        <f:facet name="header">Пол*</f:facet>
        <h:selectOneMenu value="#{a.sex}">
            <f:selectItems value="#{animalBean.sexs}" var="s"
itemValue="#{s}" itemLabel="#{s}" />
        </h:selectOneMenu>
    </h:column>
    <h:column>
        <f:facet name="header">Дата рождения*</f:facet>
        <h:inputText p:type="date" value="#{a.birthDate}" /><br/>
    </h:column>

```

```

        <h:column>
            <f:facet name="header">Дата смерти</f:facet>
            <h:inputText p:type="date" value="#{a.deadDate}"/><br/>
        </h:column>
        <h:column>
            <f:facet name="header">
                <h:graphicImage styleClass="tools"
value="../../../img/tools.png"/>
            </f:facet>
            <h:commandLink action="#{animalBean.init}">
                <h:graphicImage styleClass="tools"
value="../../../img/delete.png"/>
                <f:ajax execute="@form" render="entityForm messages"
listener="#{animalBean.delete(a) }"/>
            </h:commandLink>
            <h:commandLink action="#{animalBean.init}">
                <h:graphicImage styleClass="tools"
value="../../../img/edit.png"/>
                <f:ajax execute="@form" render="@form messages"
listener="#{animalBean.update(a, employeeBean.employees) }"/>
            </h:commandLink>
        </h:column>
    </h:dataTable>
</h:form>
</div>
<br/>
<div style="min-width: 1300px">
    <div class="inputform">
        <h:form styleClass="inputform" id="addForm">
            <div style="margin-top: auto">
                <h:selectOneMenu styleClass="input" id="inputEmp"
value="#{employeeBean.employee.id}"
required="true" requiredMessage="Выберите
сотрудника"

valueChangeListener="#{animalBean.employeefrompage}">
                <f:selectItem itemDisabled="true"
itemLabel="Сотрудник"/>
                <f:selectItems value="#{employeeBean.employees}"

```

```

var="e"

                                itemValue="#{e.id}"
itemLabel="#{e.firstName}"/>
    </h:selectOneMenu>
</div>
<div style="margin-top: auto">
    <h:inputText styleClass="input" id="inputName"
p:placeholder="Кличка" p:type="text"
                                value="#{animalBean.animal.name}"
                                required="true" requiredMessage="Введите
кличку"/>
    </div>
<div style="margin-top: auto">
    <h:inputText styleClass="input" id="inputSpecies"
p:placeholder="Вид" p:type="text"
                                value="#{animalBean.animal.species}"
                                required="true" requiredMessage="Введите
вид"/>
    </div>
<div style="margin-top: auto">
    <h:inputText styleClass="input" p:placeholder="Класс"
p:type="text"
                                value="#{animalBean.animal.aclass}"/>
    </div>
<div style="margin-top: auto">
    <h:inputText styleClass="input" p:placeholder="Отряд"
p:type="text"
                                value="#{animalBean.animal.order}"/>
    </div>
<div style="margin-top: auto">
    <h:selectOneMenu styleClass="input" id="inputSex"
value="#{animalBean.animal.sex}"
                                required="true" requiredMessage="Выберите
пол">
        <f:selectItem itemDisabled="true" itemLabel="Пол"/>
        <f:selectItems value="#{animalBean.sexs}" var="s"
itemValue="#{s}" itemLabel="#{s}"/>
    </h:selectOneMenu>
</div>

```



```

</div>
<div style="margin-top: auto">
    <h:inputText p:placeholder="Вид" p:type="text"
        value="#{animalBean.animal.species}"
        styleClass="input"/>
</div>
<div style="margin-top: auto">
    <h:inputText p:placeholder="Класс" p:type="text"
value="#{animalBean.animal.aclass}"
        styleClass="input"/>
</div>
<div style="margin-top: auto">
    <h:inputText p:placeholder="Отряд" p:type="text"
value="#{animalBean.animal.order}"
        styleClass="input"/>
</div>
<div style="margin-top: auto">
    <h:selectOneMenu value="#{animalBean.animal.sex}"
styleClass="input">
        <f:selectItem itemLabel="Пол"/>
        <f:selectItems value="#{animalBean.sexs}" var="s"
itemValue="#{s}" itemLabel="#{s}"/>
    </h:selectOneMenu>
</div>
<div style="margin-top: auto">
    <h:inputText p:type="date"
value="#{animalBean.animal.birthDate}"
        styleClass="input"/>
</div>
<div style="margin-top: auto">
    <h:inputText p:type="date"
value="#{animalBean.animal.deadDate}" styleClass="input"/>
</div>
<div>
    <h:commandButton styleClass="tools"
action="#{animalBean.search(employeeBean.employee.id,
animalBean.animal.name,animalBean.animal.species,animalBean.animal.aclass,

```

```

animalBean.animal.order,animalBean.animal.sex,animalBean.animal.birthDate,
        animalBean.animal.deadDate)}" image="../../img/search.png">
        <f:ajax execute="@form" render="entityForm messages"/>
    </h:commandButton>
</div>
</h:form>
</div>
<div>
    <h:messages id="messages"/>
</div>
</h:body>
</f:view>
</html>

```

Додаток В – SQL код таблиць

```

DROP DATABASE IF EXISTS `zoodb`;
CREATE DATABASE `Zoodb`;
USE `Zoodb`;

```

```

CREATE TABLE `Animal` (
    `Id` INT NOT NULL AUTO_INCREMENT,
    `EmployeeId` INT NOT NULL,
    `Name` CHAR(20) NOT NULL,
    `Species` CHAR(20) NOT NULL,
    `Class` CHAR(20),
    `Order` CHAR(20),
    `Sex` CHAR(1) NOT NULL,
    `BirthDate` DATE NOT NULL,
    `DeadDate` DATE,
    CONSTRAINT `PK_Animal` PRIMARY KEY (`Id`)

```

);

```
CREATE TABLE `Employee`(  
    `Id` INT NOT NULL AUTO_INCREMENT,  
    `FirstName` CHAR(15) NOT NULL,  
    `LastName` CHAR(15),  
    `Sex` CHAR(1) NOT NULL,  
    `BirthDate` DATE NOT NULL,  
    `ContactInformation` CHAR(15) NOT NULL,  
    CONSTRAINT `PK_Employee` PRIMARY KEY (`Id`)  
);
```

```
CREATE TABLE `Ration`(  
    `Id` INT NOT NULL AUTO_INCREMENT,  
    `AnimalId` INT NOT NULL,  
    `FeedId` INT NOT NULL,  
    `FeedTime` Time NOT NULL,  
    CONSTRAINT `PK_Ration` PRIMARY KEY (`Id`)  
);
```

```
CREATE TABLE `Vaccination`(  
    `Id` INT NOT NULL AUTO_INCREMENT,  
    `AnimalId` INT NOT NULL,  
    `VaccineId` INT NOT NULL,  
    `DateVaccination` DATE NOT NULL,  
    CONSTRAINT `PK_Vaccination` PRIMARY KEY (`Id`)
```

);

```
CREATE TABLE `Feed`(  
    `Id` INT NOT NULL AUTO_INCREMENT,  
    `FeedName` CHAR(40) NOT NULL,  
    `FeedType` CHAR(40) NOT NULL,  
    `Amount` INT NOT NULL,  
    CONSTRAINT `PK_Feed` PRIMARY KEY (`Id`)  
);
```

```
CREATE TABLE `Vaccine`(  
    `Id` INT NOT NULL AUTO_INCREMENT,  
    `Disease` CHAR(40) NOT NULL,  
    `ExpirationDate` DATE NOT NULL,  
    `Amount` INT NOT NULL,  
    CONSTRAINT `PK_Vaccine` PRIMARY KEY (`Id`)  
);
```

```
CREATE TABLE `Equipment`(  
    `Id` INT NOT NULL AUTO_INCREMENT,  
    `Thing` CHAR(40) NOT NULL,  
    `Amount` INT NOT NULL,  
    CONSTRAINT `PK_Equipment` PRIMARY KEY (`Id`)  
);
```

```
CREATE TABLE `Equipment supply`(  
    `Id` INT NOT NULL AUTO_INCREMENT,
```

```

        `ProductId` INT NOT NULL,
        `SupplierId` INT NOT NULL,
        `SupplyDate` DATE NOT NULL,
        `Amount` INT NOT NULL,
    CONSTRAINT `PK_Supply` PRIMARY KEY (`Id`)
);

```

```

CREATE TABLE `Feed supply`(
    `Id` INT NOT NULL AUTO_INCREMENT,
    `ProductId` INT NOT NULL,
    `SupplierId` INT NOT NULL,
    `SupplyDate` DATE NOT NULL,
    `Amount` INT NOT NULL,

    CONSTRAINT `PK_Supply` PRIMARY KEY (`Id`)
);

```

```

CREATE TABLE `Vaccine supply`(
    `Id` INT NOT NULL AUTO_INCREMENT,
    `ProductId` INT NOT NULL,
    `SupplierId` INT NOT NULL,
    `SupplyDate` DATE NOT NULL,
    `Amount` INT NOT NULL,
    CONSTRAINT `PK_Supply` PRIMARY KEY (`Id`)
);

```

```

CREATE TABLE `Animal supply`(

```

```

        `Id` INT NOT NULL AUTO_INCREMENT,
        `ProductId` INT NOT NULL,
        `SupplierId` INT NOT NULL,
        `SupplyDate` DATE NOT NULL,
        CONSTRAINT `PK_Supply` PRIMARY KEY (`Id`)
    );

```

```

CREATE TABLE `Supplier`(
    `Id` INT NOT NULL AUTO_INCREMENT,
    `FirstName` CHAR(15),
    `LastName` CHAR(15),
    `OrganizationId` INT NOT NULL,
    CONSTRAINT `PK_Supplier` PRIMARY KEY (`Id`)
);

```

```

CREATE TABLE `Organization`(
    `Id` INT NOT NULL AUTO_INCREMENT,
    `Name` CHAR(15) NOT NULL,
    `Type` CHAR(15) NOT NULL,
    `Address` CHAR(20) NOT NULL,
    `City` CHAR(20) NOT NULL,
    `Country` CHAR(20) NOT NULL,
    `ContactInformation` CHAR(15) NOT NULL,
    CONSTRAINT `PK_Organization` PRIMARY KEY (`Id`)
);

```

```

ALTER TABLE `Supplier` ADD CONSTRAINT `FK_SupplierOrganizationId`

```

```
FOREIGN KEY (`OrganizationId`) REFERENCES `Organization` (`Id`) ON
DELETE NO ACTION ON UPDATE NO ACTION;
```

```
CREATE INDEX `IFK_SupplierOrganizationId` ON `Supplier`
(`OrganizationId`);
```

```
ALTER TABLE `Animal` ADD CONSTRAINT `FK_AnimalEmployeeId`
FOREIGN KEY (`EmployeeId`) REFERENCES `Employee` (`Id`) ON
DELETE NO ACTION ON UPDATE NO ACTION;
```

```
CREATE INDEX `IFK_AnimalEmployeeId` ON `Animal` (`EmployeeId`);
```

```
ALTER TABLE `Vaccination` ADD CONSTRAINT `FK_VaccinationAnimalId`
FOREIGN KEY (`AnimalId`) REFERENCES `Animal` (`Id`) ON DELETE
NO ACTION ON UPDATE NO ACTION;
```

```
CREATE INDEX `IFK_VaccinationAnimalId` ON `Vaccination` (`AnimalId`);
```

```
ALTER TABLE `Vaccination` ADD CONSTRAINT
`FK_VaccinationVaccineId`
FOREIGN KEY (`VaccineId`) REFERENCES `Vaccine` (`Id`) ON DELETE
NO ACTION ON UPDATE NO ACTION;
```

```
CREATE INDEX `IFK_VaccinationVaccineId` ON `Vaccination` (`VaccineId`);
```

```
#####
```

```
ALTER TABLE `Ration` ADD CONSTRAINT `FK_RationAnimalId`
FOREIGN KEY (`AnimalId`) REFERENCES `Animal` (`Id`) ON DELETE
NO ACTION ON UPDATE NO ACTION;
```

```
CREATE INDEX `IFK_RationAnimalId` ON `Ration` (`AnimalId`);
```

```
ALTER TABLE `Ration` ADD CONSTRAINT `FK_RationFeedId`
```



```
FOREIGN KEY (`FeedId`) REFERENCES `Feed` (`Id`) ON DELETE NO
ACTION ON UPDATE NO ACTION;
```

```
CREATE INDEX `IFK_RationFeedId` ON `Ration` (`FeedId`);
```

```
ALTER TABLE `Equipment supply` ADD CONSTRAINT
`FK_EquipmentSupProductId`
```

```
FOREIGN KEY (`ProductId`) REFERENCES `Equipment` (`Id`) ON
DELETE NO ACTION ON UPDATE NO ACTION;
```

```
CREATE INDEX `IFK_EquipmentSupProductId` ON `Equipment supply`
(`ProductId`);
```

```
ALTER TABLE `Equipment supply` ADD CONSTRAINT
`FK_EquipmentSupSupplierId`
```

```
FOREIGN KEY (`SupplierId`) REFERENCES `Supplier` (`Id`) ON DELETE
NO ACTION ON UPDATE NO ACTION;
```

```
CREATE INDEX `IFK_EquipmentSupSupplierId` ON `Equipment supply`
(`SupplierId`);
```

```
ALTER TABLE `Feed supply` ADD CONSTRAINT `FK_FeedSupProductId`
```

```
FOREIGN KEY (`ProductId`) REFERENCES `Feed` (`Id`) ON DELETE NO
ACTION ON UPDATE NO ACTION;
```

```
CREATE INDEX `IFK_FeedSupProductId` ON `Feed supply` (`ProductId`);
```

```
ALTER TABLE `Feed supply` ADD CONSTRAINT `FK_FeedSupSupplierId`
```

```
FOREIGN KEY (`SupplierId`) REFERENCES `Supplier` (`Id`) ON DELETE
NO ACTION ON UPDATE NO ACTION;
```

```
CREATE INDEX `IFK_FeedSupSupplierId` ON `Feed supply` (`SupplierId`);
```

```
ALTER TABLE `Vaccine supply` ADD CONSTRAINT
`FK_VaccineSupProductId`
```

```
FOREIGN KEY (`ProductId`) REFERENCES `Vaccine` (`Id`) ON DELETE
NO ACTION ON UPDATE NO ACTION;
```

```
CREATE INDEX `IFK_VaccineSupProductId` ON `Vaccine supply`
(`ProductId`);
```

```
ALTER TABLE `Vaccine supply` ADD CONSTRAINT
`FK_VaccineSupSupplierId`
```

```
FOREIGN KEY (`SupplierId`) REFERENCES `Supplier` (`Id`) ON DELETE
NO ACTION ON UPDATE NO ACTION;
```

```
CREATE INDEX `IFK_VaccineSupSupplierId` ON `Vaccine supply`
(`SupplierId`);
```

```
ALTER TABLE `Animal supply` ADD CONSTRAINT
`FK_AnimalSupProductId`
```

```
FOREIGN KEY (`ProductId`) REFERENCES `Animal` (`Id`) ON DELETE
NO ACTION ON UPDATE NO ACTION;
```

```
CREATE INDEX `IFK_VaccineSupProductId` ON `Animal supply`
(`ProductId`);
```

```
ALTER TABLE `Animal supply` ADD CONSTRAINT
`FK_AnimalSupSupplierId`
```

```
FOREIGN KEY (`SupplierId`) REFERENCES `Supplier` (`Id`) ON DELETE
NO ACTION ON UPDATE NO ACTION;
```

```
CREATE INDEX `IFK_VaccineSupSupplierId` ON `Animal supply`
(`SupplierId`);
```