

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

Гарант освітньої програми

ДОЦЕНТ, К.Т.Н.

(посада, науковий ступінь вчене звання)

_____ Михелєв І.Л.

(підпис)

" ____ " _____ 2022 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти "бакалавр"

на тему: "Розробка інформаційної системи управління зоопарком"

Виконав: студент групи 4142

_____ Васильченко А.С.

(підпис)

Керівник роботи:

ДОЦЕНТ, К.Т.Н.

(посада, науковий ступінь вчене звання)

_____ Ажищев В.Ф.

(підпис)

м. Миколаїв – 2022 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

доцент, к.т.н.

(посада, науковий ступінь вчене звання)

Гайда А.Ю.

(підпис)

"_____" "_____" 2022 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ на здобуття ступеня вищої освіти "бакалавр"

Студенту Васильченку Андрію Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Розробка інформаційної системи управління зоопарком

Керівник роботи Ажищев Віктор Федорович

Затверджені наказом ректора № 269-уч від "18" травня 2022 року.

2. Термін подання роботи: 20 червня 2022 р.

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

4. Перелік питань, що належать до розробки (найменування розділів): Вступ. 1.Розробка концепції інформаційної системи. 2.Розробка технічного проекту інформаційної системи. 3.Реалізація проекту інформаційної системи. 4.Охорона праці. Висновки.

5. Перелік презентаційних матеріалів: Слайди: Розробка інформаційної системи управління зоопарком; Актуальність, ціль, завдання; Об'єкт дослідження; Концепція системи; технічний проект; Моделі бази даних; Програмне забезпечення; Технічне забезпечення; Використання системи.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Ажищев В.Ф. доц, к.т.н.		
2	Ажищев В.Ф. доц, к.т.н.		
3	Ажищев В.Ф. доц, к.т.н.		
4	Ажищев В.Ф. доц, к.т.н.		

7. Дата видачі завдання 14 лютого 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів роботи	Примітка
1	Розробка змісту інформаційної системи		
2	Розробка концепції інформаційної системи		
3	Розробка технічного проекту інформаційної системи		
4	Реалізація проекту інформаційної системи		
5	Охорона праці		
6	Вступ, висновки, список використаної літератури, додатки		
7	Презентація		
8			
9			
10			

Студент

(підпис)

Васильченко А.С

(ПІБ)

Керівник роботи

(підпис)

Ажищев В.Ф

(ПІБ)

Анотація

Дипломний проект на тему: “Розробка системи управління зоопарком” має об'єм 69 сторінок, містить розділи - 4, малюнки — 17, використані джерела — 46.

Ціль проекту полягає у підвищенні ефективності управління зоопарком за рахунок впровадження інформаційної системи, яка дозволить мінімізувати витрати та збільшити прибуток.

Завдання проекту полягає у розробці інформаційної системи.

У першому розділі 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 в якому представлені 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 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 за рахунок впровадження сучасних інформаційних технологій та комп'ютерної техніки.

Ключові слова: зоопарк, управління, тварини, реалізація, модель, схема, рисунок.

Annotation

Thesis project on the topic: "Development of a zoo management system" has a volume of 69 pages, contains sections - 4, drawings - 17, used sources - 46.

The aim of the project is to increase the efficiency of zoo management through the introduction of an information system that will minimize costs and increase profits.

The task of the project is to develop an information system.

In the first section the concept of the information system is developed, the survey of the enterprise is carried out, the analysis of information processes is carried out, requirements to the designed system are formed, the purposes and tasks of the project are defined.

In the second section the technical project of the system is developed in which the system-wide decisions on the choice of standard software are presented, the database is designed and the conceptual model, logical model, physical model are presented. Mathematical algorithms for solving problems are determined. As well as certain hardware and software that will be operated by the system.

In the third section, the development of working documentation.

The fourth section presents rules and guidelines for the safety of visitors and staff at the zoo, as well as computer safety rules.

The relevance and novelty of the work is to improve the management system of the zoo for efficient work through the introduction of modern information technology and computer technology.

Key words: zoo, management, animals, realization, model, scheme, drawing.

ЗМІСТ

ВСТУП	7
1.Розробка концепції інформаційної системи “Управління зоопарком”	8
1.1 Опис об'єкту та аналіз предметної області.....	8
1.2 Вимоги до інформаційної системи.....	15
1.3 Розробка концепції інформаційної системи	17
2.Розробка технічного проекту інформаційної системи “Управління зоопарком”	21
2.1 Загальносистемні рішення.....	21
2.2 Рішення по інформаційному забезпеченню.....	25
2.2.1 Розробка концептуальної моделі даних.....	26
2.2.2 Розробка логічної моделі даних.....	26
2.2.3 Розробка фізичної моделі даних.....	27
2.3 Рішення з математичного забезпечення.....	28
2.4 Рішення з програмного забезпечення.....	30
2.5 Рішення з технічного забезпечення.....	38
3.Реалізація проекту інформаційної системи “Управління зоопарком”	39
4.Охорона праці	42
Висновок	52
Список використаної літератури	53
Додаток А- Інструкція користувачу.....	56
Додаток Б- Код програми.....	59

Вступ

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

Завдання стоїть у розробці концепції цієї системи, розробці технічного проекту, у створенні фізичних, концептуальних та логічних моделей даних, розробки програмних засобів.

У першому розділі 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 в якому представлені 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 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 за рахунок впровадження сучасних інформаційних технологій та комп'ютерної техніки.

Розділ 1 Розробка концепції інформаційної системи “Управління зоопарком”

1.1 Опис об'єкту та аналіз предметної області

Зоопарк (зоологічний парк) — територія, спеціально створена людиною, з будівлями і обладнанням для утримання, збереження, розмноження зникаючих видів та прилюдного показу тварин. Зоологічні парки постають задля організації екологічної освітньо-виховної роботи, забезпечення експозицій рідкісних, екзотичних та місцевих видів тварин, збереження їх генофонду, вивчення дикої фауни і розробки наукових основ її розведення у неволі [1].

Для повноцінної роботи зоопарку потрібні кошти. Зоопарки покладаються або на кошти від продажу білетів, або на кошти від різних фондів, а також на державні субсидії. В усіх цих випадках для отримання коштів зоопарк повинен залучати відвідувачів. Якщо кількість відвідувачів буде малою, то коштів від продажу білетів буде недостатньо, і місцевій владі буде не вигідне подальше існування цього закладу. Отже керівництву потрібно підтримувати роботу зоопарку як слід, для залучення достатньої кількості відвідувачів.

Для залучення відвідувачів можна використовувати такі кроки:

1.Проводити заходи для дорослих;

Коли ви плануєте захід лише для дорослих, це дає батькам можливість сповільнитися і відчувати все, що може запропонувати зоопарк. Вони можуть побачити експонати та ознайомитися з інформацією навколо них. Вони також можуть розслабитися і насолоджуватися таким чином, як не можуть, коли цілий день сваряться з дітьми. Ці події також можуть привабити людей, які зазвичай не відвідують зоопарк, наприклад молодих дорослих і дорослих без дітей.

2.Проводити спеціальні сімейні заходи;

Спеціальні заходи – це чудовий спосіб залучити людей, яких не так цікавлять тварини. Вони також можуть залучати відвідувачів, які не поверталися, думаючи, що вони вже «бачили все» в зоопарку.

3. Створити членську перепустку;

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.

Ви можете заохочувати людей купувати членство, пропонуючи членам знижки на подарунки або квитки на спеціальні заходи. Членство надихне ваших відвідувачів скористатися спеціальними пропозиціями на їжу, сувенірними товарами з обмеженим тиражом і квитками на спеціальні заходи.

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. Пропонувати спеціальні пропозиції та дні знижок;

Пропонування спеціальних знижок, таких як військова знижка або обмін членства на інші визначні пам'ятки у вашому районі, є чудовим способом збільшити відвідуваність. Окрім того, що відвідування зоопарку стає доступнішим для деяких, це також показує, що ваш зоопарк прагне взаємодіяти з вашою спільнотою та зробити ваші експонати доступними для всіх. Пропонуйте ці пропозиції в дні, коли кількість відвідувачів, як правило, нижча, щоб збільшити кількість відвідувачів.

6. Підкреслюйте практику збереження

Зоопарки часто розглядають як розвагу для дітей, але зоопарк це набагато більше, ніж просто розвага для дітей. Упор на дослідження та природоохоронні

практики – ідеальний спосіб залучити більше дорослих у вашу спільноту та спонукати їх більше інвестувати у відвідування зоопарку.

Звичайно, навіть діти можуть оцінити зусилля вашого зоопарку із захисту світу навколо нас, тому створюйте програми та презентації, орієнтовані на обидві демографічні групи.

7. Включити в зоопарк унікальні заходи.

Ще один чудовий спосіб збільшити відвідуваність зоопарку – запропонувати унікальні враження від парку для ваших об'єктів. Ви можете виділитися з-поміж місцевих пам'яток або зоопарків-конкурентів або об'єднати враження в одному зручному місці. Додавання інтерактивних вражень до вашого зоопарку, таких як годування тварин, спеціальні екскурсії за лаштунками та навчальні зустрічі з тваринами, також є чудовим способом збільшити відвідуваність [3].

Організаційна структура управління зоопарком зображено на рисунку 1.1.

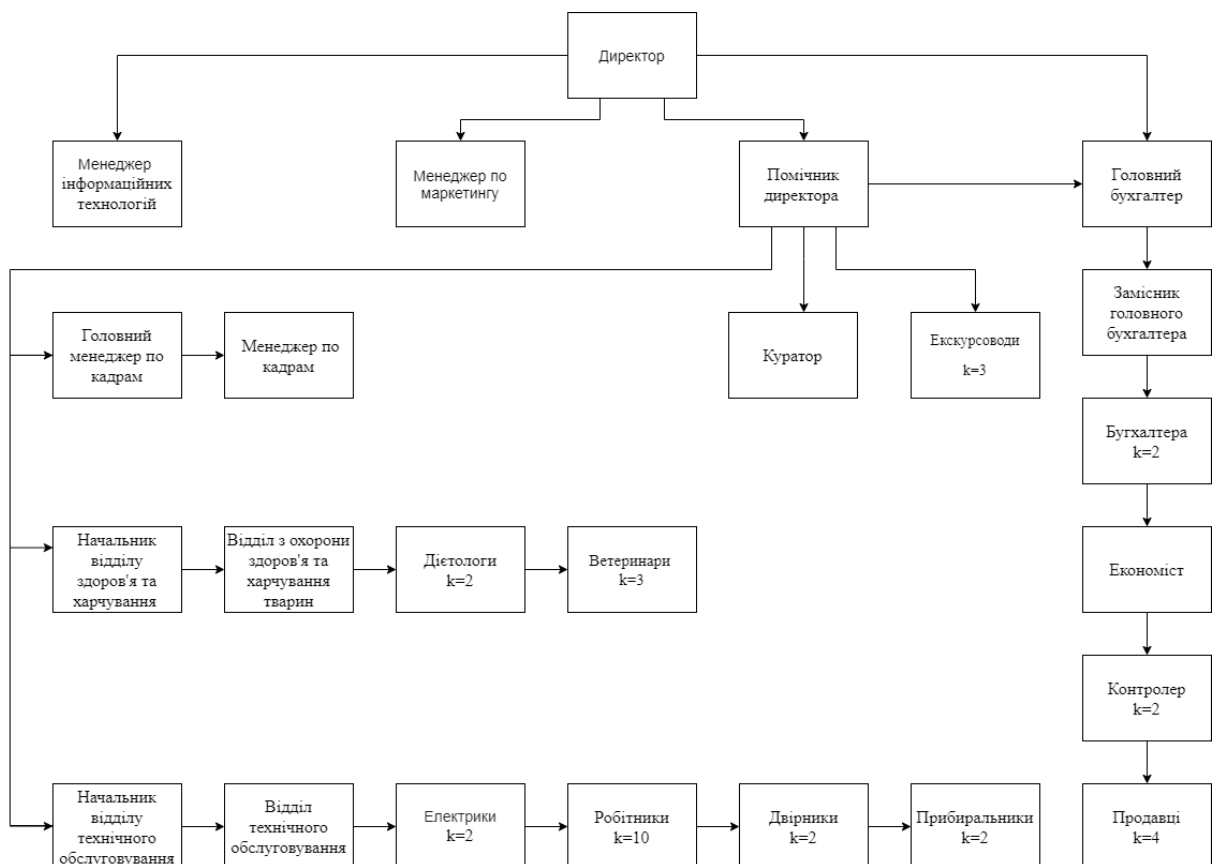


Рисунок 1.1 — Організаційна структури управління зоопарком

Керівництвом зоопарком здійснює директор. Директор відповідає за нагляд за всіма аспектами управління зоопарком. Області уваги зазвичай включають управління роботою парку, створення бюджетів, реалізацію політик, наймання управлінського персоналу, пошук додаткового фінансування та нагляд над розвитком об'єкта. Директор також зазвичай виступає як головний представник зоопарку у відносинах із засобами масової інформації.

Директор співпрацює із начальниками відділів та куратором, які, у свою чергу, керують іншими співробітниками зоопарку. Директор несе відповідальність за те, щоб усі повсякденні операції проходили гладко, а тварин доглядали відповідно до всіх правил [2].

Під його безпосереднім керівництвом знаходяться помічник директора, менеджер по маркетингу, менеджер інформаційних технологій, головний бухгалтер.

Розглянемо більш детально роботу кожного з відділів.

В той час коли директор займається зовнішніми справами закладу помічник директора — внутрішніми. Він спрямовує, керує та координує внутрішню операційну діяльність організації відповідно до політики, цілей і завдань, встановлених директором [4]. Йому підпорядковуються головний менеджер по кадрам, начальник відділу здоров'я та харчування, головним бухгалтером, начальник відділу технічного обслуговування, екскурсоводи, куратор.

В обов'язки головного менеджера по кадрам входить: розробляти та контролювати загальні кадрові стратегії, системи, тактики та процедури в організації, забезпечувати відповідність законодавству в управлінні персоналом, звітувати перед керівництвом. У разі його відсутності його замінює менеджера з кадрів [13].

В обов'язки менеджера по кадрам входить: набір персоналу, відносини зі співробітниками, керувати процесом найму та відбору, підтримувати позитивну робочу атмосферу [13].

Відділом з охорони здоров'я та харчування тварин керує начальник відділу. Він несе відповідальність за роботу відділу. В даному відділі працюють: 2 ветеринари та 4 дієтологи.

Дієтологи повинні, враховуючи різні харчові потреби різних видів тварин, скласти збалансованих харчові раціони для них. На основі стану тварини та типу фізичної активності, дієтологи повинні оцінювати потреби в харчуванні та калоріях. Відповідають за щоденне тестування та аналіз, щоб точно визначити, які та скільки поживних речовин поглинають екзотичні тварини через їжу, яку вони їдять, після чого коригують раціон. Повинні перевіряти їжу на якість [11;12].

Ветеринари займаються лікуванням травм та хвороб тварин, що мешкають у зоопарку, а також профілактичною медичною допомогою. Відповідають за проведення фізичних оглядів тварин, адміністрація садації, робити щеплення, введення та призначення ліків, взяття крові та інших зразків, проведення операції, чищення зубів, проведення УЗД та рентгенограм, лікування ран [10].

Головний бухгалтер відповідає за бухгалтерську звітність організації перед керівництвом. Йому підпорядковуються: 2 бухгалтера, економіст, 2 контролера, 4 продавці.

Замісник головного бухгалтера допомагає головному бухгалтеру у створенні своєчасної та точної фінансової інформації для всіх аспектів зоопарку. До обов'язків належить бухгалтерське обслуговування, допомога у створенні бюджетів для рахунків всіх відділів зоопарку у процесі бюджетування, допомога у написанні звітів та запитів у бухгалтерських системах зоопарку, допомога у всіх бухгалтерських операціях протягом року та наприкінці року [8] У разі відсутності головного бухгалтера, бере на себе його зобов'язання.

Бухгалтера займаються веденням бухгалтерського обліку по доходам від продажу квитків та по витратам, займаються складанням звітів та ведуть статистику.

Економіст вивчає дані та статистику, щоб помітити тенденції економічної активності, рівень економічної впевненості та ставлення відвідувачів [5]. Займається формуванням та веденням бюджету зоопарку, вирішує завдання планування його роботи, а також веде статистику.

В зоопарку працюють магазини з сувенірами та кафе для відвідувачів в яких працюють продавці. Продавці сувенірів повинні заохочувати покупати їх продукцію.

На вході до зоопарку знаходиться контролер, який продає квитки, або за наявності VIP-білетів — перевіряє їх.

Відділом технічного обслуговування керує начальник відділу. Відділ виконує планове технічне обслуговування для запобігання критичних поломок обладнання. У разі поломки має зробити ремонт зламаного обладнання. Також начальник відділу має закуповувати необхідні деталі для ремонту обладнання в зоопарку. В даному відділі працюють: 2 електрика, сантехник, 15 робітників, 2 двірник, 2 прибиральник.

Електрики здійснюють електротехнічне забезпечення, технічне обслуговування та ремонт експонатів, обладнання та приміщень. Виконують нові заявки на електромонтажні роботи. Відслідковують електричну ефективність у всьому зоопарку. Діагностують та усувають несправності в електричному устаткуванні або системах керування, таких як котли, насоси, двигуни, генератори, повітряні компресори, пожежна сигналізація, сушарки, холодильники, верхні двері або інші електричні пристрої. Встановлюють освітлювальні прилади, розетки, пристрої керування, обігрівачі, термостати, комп'ютерні кабелі або подібні пристрої. Прокладають кабелепровід та простягає дроти для нових або існуючих електроустановок. Проводять профілактичний ремонт електротехнічного та електронного обладнання. Усуває проблеми в системах, до складу яких входить деяка електроніка [7].

Сантехник займається монтажем, технічним обслуговуванням та ремонтом дренажних систем, систем протипожежного захисту, газопроводів та устаткування, систем опалення та охолодження, систем поливу, систем

відведення стічних вод, водопровідних, каналізаційних та паропроводів, пристроїв керування та обладнання повітряних компресорних систем і магістралей [6].

Двірники здійснюють прибирання на території зоопарку.

Прибиральник здійснюють прибирання громадських та службових будівель на території зоопарку.

Робітники займаються прибиранням вольєрів, доглядом за твариною її годування, а також охороною зоопарку. У разі потреби допомагають електрикам, сантехнікам, продавцям.

Куратор відповідає за організацію догляду за тваринами, їх докільля та нагляд за персоналом. Він спостерігають за великою різноманітністю видів та тісно співпрацюють із працівниками зоопарку, допоміжним персоналом, ветеринарами та відвідувачами.

Менеджер по маркетингу відповідає за поширення інформації з використанням цифрових та традиційних засобів масової інформації та інших інструментів для досягнення цілей за кількістю відвідувачів та фінансових показників.

Менеджер ІТ відповідає за розробку програмного забезпечення для автоматизації роботи бухгалтерії, також співпрацює з менеджером по маркетингу для створення рекламних сайтів.

Екскурсоводи діляться інформацією про тварин та докільля з відвідувачами зоопарку та відповідають за проведення неформальних програм для відвідувачів різного віку, роботу зі шкільними групами, таборами та класами, допомога у проведенні спеціальних заходів та програмі поведінки тварин [9].

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

1.2 Вимоги до інформаційної системи “Управління зоопарком”

На основі проведеного аналізу інформаційних процесів в даному об'єкті сформуємо такі вимоги: автоматизувати роботу бухгалтерії, створити рекламний сайт.

Функціональні вимоги:

- автоматизація бухгалтерського обліку та звітності;
- автоматизація відносин з банком та податковою інспекцією;
- автоматизація розрахунку заробітної плати;
- автоматизація продажі білетів;
- автоматизація планування та бюджетування;
- формування рекламного сайту
- формувати звіти про доходи;
- формувати звіти про витрати;
- формувати звіти з продажу за місяць, рік;
- формувати звіти про відвідуваність;
- формувати звіти про фінансовий стан;
- формувати звіти про заробітну плату;
- облік фінансових ресурсів;
- облік матеріальних ресурсів.

Програма повинна мати простий інтерфейс, який буде зрозумілий користувачам, що не має спеціальних навичок у галузі обчислювальної технологій.

Вимоги до технічних засобів.

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

6 комп'ютера для бухгалтерського відділу.

1 комп'ютер для директора;

1 комп'ютер для менеджера по маркетингу;

1 комп'ютер для менеджера інформаційних технологій;

1 комп'ютер для менеджера по кадрам.

Мінімальні технічні та програмні вимоги на цих комп'ютерах

- Windows 7 32/64-bit , Windows 10 32/64-bit;
- ОЗУ не менше 8096 Мб;
- Клавіатура;
- Миша;
- Монітор;
- Жорсткий диск не менше ніж 500 Гб.

Також кожен відділ повинен бути забезпечений принтером.

Рекламний сайт реалізований з використанням технології СУБД MySQL.

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

- Хостинг VPS;
- Тактова частота процесору не нижче 3.1 ГГц;
- ОЗУ не менше 8 Гб;
- Жорсткий диск не менше ніж 5 Тб;
- СУБД MySQL.
- Клавіатура;
- Миша;
- Монітор;

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

1.3 Розробка концепції інформаційної системи

На основі сформованих технічних вимог розроблено концепцію інформаційної системи "Управління зоопарком" [11].

Модель концепції інформаційної системи представлена на рисунку 1.2.

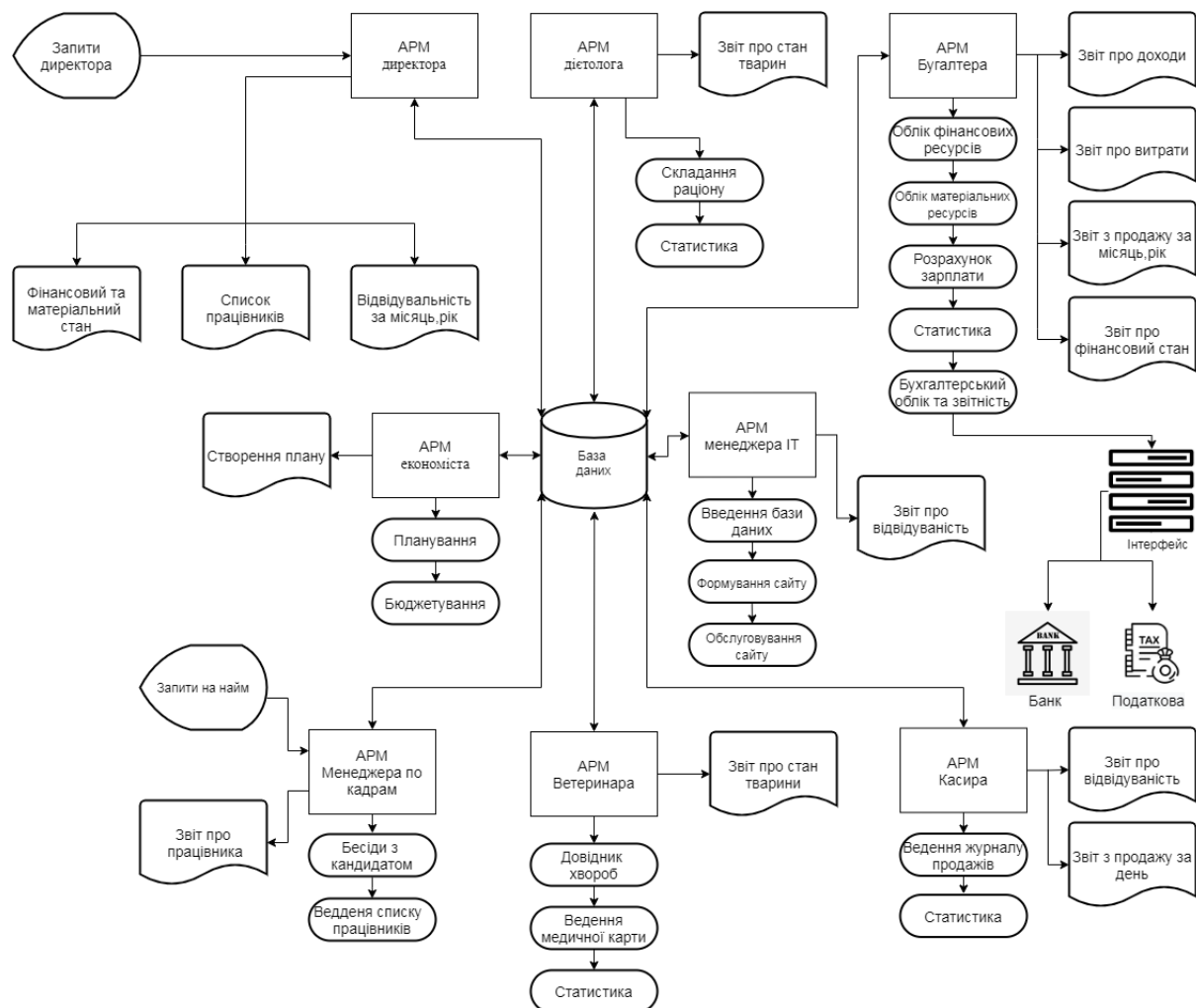


Рисунок 1.2 — Модель концепції інформаційної системи

АРМ бухгалтера виконує такі завдання як: облік фінансових ресурсів, облік матеріальних ресурсів, розрахунок зарплати, статистика. В результаті отримує звіти: звіт про доходи, звіт про витрати, звіт з продажу за місяць, рік, звіт про фінансовий стан.

За допомоги АРМ, ветеринар матиме можливість швидко отримати інформацію про хворобу з довідника завдяки чому може поставити діагноз за короткий час. Виконує такі завдання як: введення медичної картки тварини, статистика. В результаті отримує звіт про стан тварини.

АРМ менеджера по кадрам обробляє запит на найм працівника до закладу де менеджер проведе с ним співбесіду. Також веде список працівників. В результаті роботи отримує звіт про працівника.

АРМ менеджера ІТ виконує такі завдання як: введення бази даних, формування та обслуговування сайту. В результаті роботи отримує звіт про відвідуваність сайту.

АРМ економіста займається формуванням та веденням бюджетування, а також плануванням роботи зоопарку.

АРМ касира виконує такі завдання як: ведення журналу продажів, статистика. В результаті отримує звіти: звіт про відвідуваність, звіт з продажу за день

За допомоги АРМ дієтолог матиме можливість швидко отримати інформацію про раціон тварини з довідника завдяки чому може призначити корм тварині за короткий час. Виконує такі завдання як: складання раціону, статистика. В результаті отримує звіт про стан тварини.

АРМ директора обробляє запити щодо стану зоопарку та в результаті отримує звіти: фінансовий та матеріальний стан зоопарку, список працівників, відвідуваність зоопарку за період від місяця до року.

На основі розробленої концепції інформаційної системи буде розроблюватися технічний проект.

Реалізація завдань бухгалтерського обліку та звітності бути виконана на основі типових проектних рішень із використанням готових програмних засобів. Вибір програмних засобів буде проведено на основі аналізу наявних засобів та обґрунтувань необхідності вибору і розробки технічного проекту.

База даних буде створена за допомоги СУБД MySql. Мовою розробки вибрано Java.

Програмна частина технічного проекту

На комп'ютерах встановлений стандартний пакет Microsoft Office а саме: Microsoft Word, Microsoft Excel, Microsoft Outlook, Microsoft PowerPoint, Microsoft Publisher та інші. Також для створення діаграм встановлено draw.io. Для захисту комп'ютера від вірусів встановлено антивірус 360 Total Security. Для комунікації із зовнішнім світом використовується електронна пошта gmail.com. Для доступу до Інтернету встановлено браузер Google Chrome. Для дистанційних зборів та співбесід використовуватиметься Zoom.

Для реалізації 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 вебсервера – XAMPP;
- Декларативна мова програмування – SQL.

Для створення диплому були використані такі інструменти:

- Програмне забезпечення для роботи з текстом – Word;
- Програмне забезпечення для створення діаграм – draw.io.

Нижче наведено технічні засоби.

Технічні засоби технічного проекту

4 каси, 2 принтера, 10 комп'ютерів з такими характеристиками

- Windows 10 64-bit;
- ОЗУ 8096 Мб;
- Клавіатура;
- Миша;
- Монітор;
- 600 Гб.

Реалізація концепції проекту дозволить повисити ефективність роботи зоопарку за рахунок чіткого виконання завдань управління, якісно реалізованої системи планування, обліку, звітності, контролю фінансових та матеріальних ресурсів.

Розділ 2 Розробка технічного проекту інформаційної системи “Управління зоопарком”

2.1 Загальносистемні рішення

Для реалізації завдань бухгалтерського обліку існує велика кількість систем наприклад: HomeBank, GnuCash, KMyMoney, Tryton, 1С:Бухгалтерия, Microsoft Money, Microsoft Dynamics AX тощо [14].

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 (формат електронної виписки SWIFT MT940), 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 [15]. 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 [19].

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 [16].

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 [17].

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 [18].

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, торгівлі 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 [20].

На основі проведеного аналізу описаних вище показників систем, з точки зору практики впровадження, прийнято рішення, щодо застосування системи ІС Бухгалтерія.

2.2 Рішення по інформаційному забезпеченню

Для ефективної роботи інформаційної системи в зоопарку потрібна база даних. Для управління базою даних використовується СУБД MySQL.

MySQL — це система управління реляційною базою даних з відкритим вихідним кодом. Реляційна база даних організовує дані в одну або кілька таблиць даних, в яких дані можуть бути пов'язані один з одним; ці відносини допомагають структурувати дані.

MySQL має автономні клієнти, які дозволяють користувачам безпосередньо взаємодіяти з базою даних MySQL за допомогою SQL, але частіше MySQL використовується з іншими програмами для реалізації програм, які потребують можливості реляційної бази даних[21].

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 показує зв'язок між об'єктами та описує їх. Метою створення концептуальної моделі даних є встановлення сутностей, їх атрибутів та відносин.

Три основні елементи концептуальної моделі даних:

- Сутність: об'єкт;
- Атрибут: характеристики або властивості об'єкта;
- Відносини: залежність чи асоціація між двома об'єктами [26].

Концептуальна модель бази даних представлена на рисунку 2.2.1

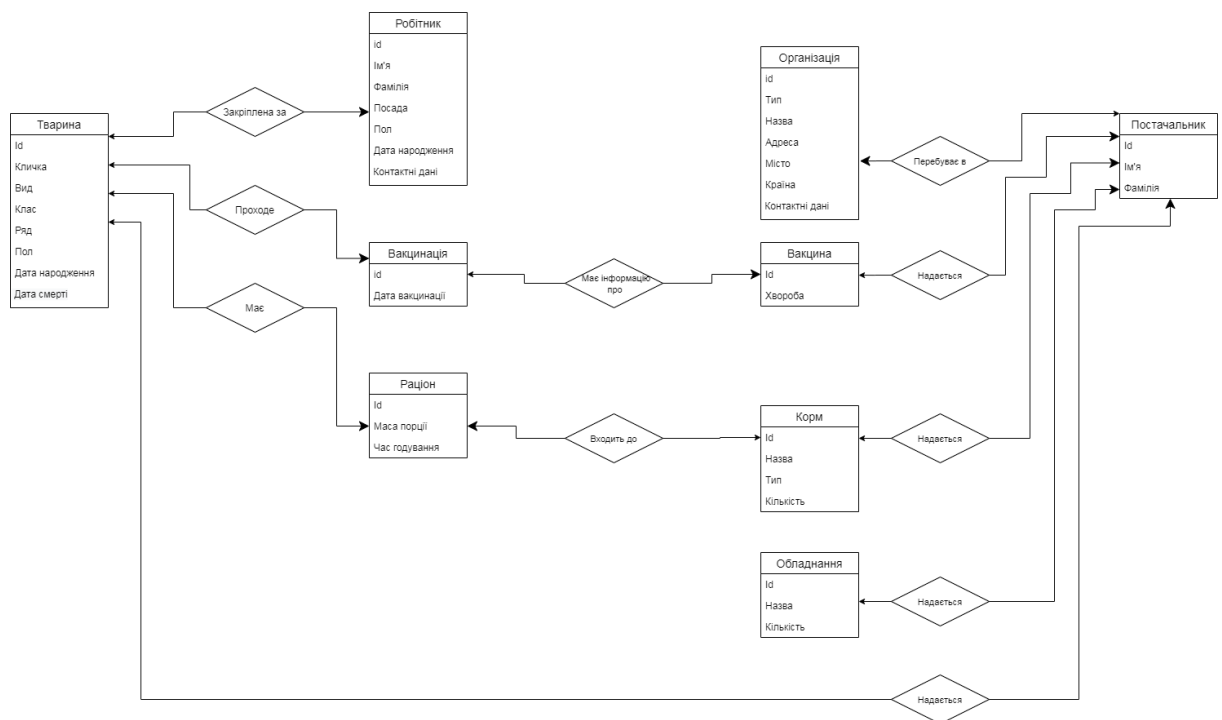


Рисунок 2.2.1 - Концептуальна модель бази даних

2.2.2 Розробка логічної моделі даних

Логічна модель даних - це модель, яка описує речі, про які організація хоче збирати дані, і відносини між цими речами. Вона містить уявлення сутностей та атрибутів, відносин, унікальних ідентифікаторів, підтипів та супертипів, а також обмежень між відносинами [26].

Логічна модель бази даних представлена на рисунку 2.2.2

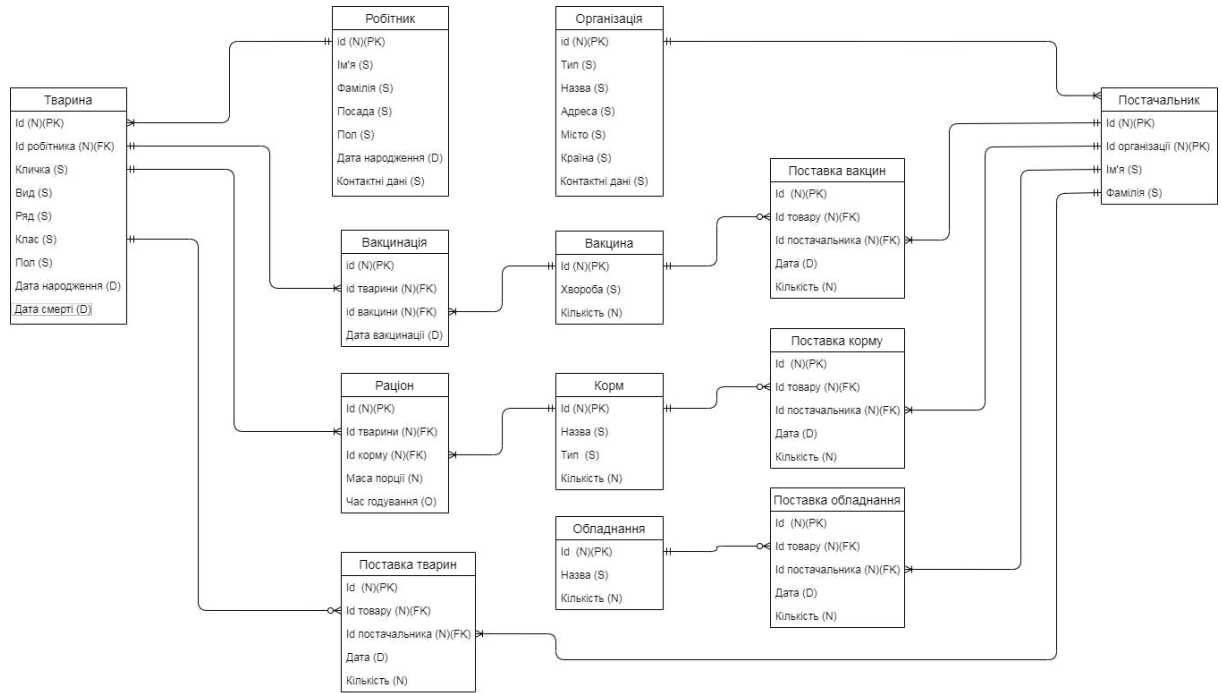


Рисунок 2.2.2 - Логічна модель бази даних

2.2.3 Розробка фізичної моделі даних

Фізична модель даних створюється на основі концептуальної моделі. Ця модель даних описує, як саме система буде реалізована з використанням конкретної системи СУБД [26].

Фізична модель бази даних представлена на рисунку 2.2.3.

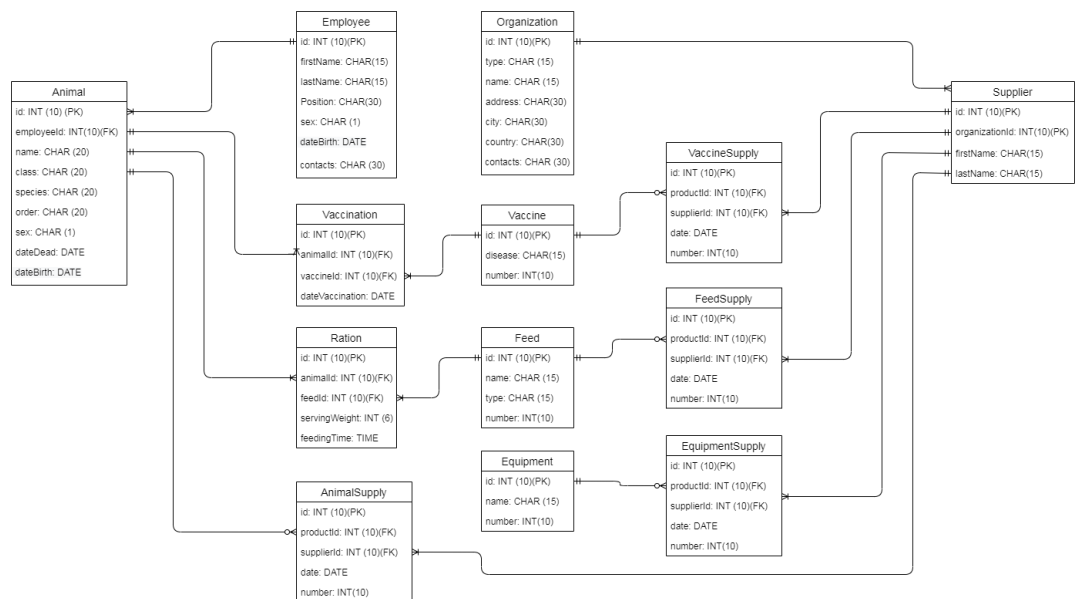


Рисунок 2.2.3 - Фізична модель бази даних

Animal — зберігає інформацію про тварин, Employee — зберігає інформацію про співробітників, Ration — зберігає інформацію про раціон, Vaccination — зберігає інформацію про вакцинацію, Feed — зберігає інформацію про корми, Vaccine — зберігає інформацію вакцину, Equipment — зберігає інформацію про обладнання, Equipment supply — зберігає інформацію про поставки обладнання, Feed supply — зберігає інформацію про поставки кормів, Vaccine supply — зберігає інформацію про поставки вакцин, Animal supply — зберігає інформацію про поставки тварин, Supplier — зберігає інформацію про постачальників, Organization — зберігає інформацію про організацію із якої постачальник.

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

2.3 Рішення з математичного забезпечення

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 [32].

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 [33].

Діаграма представлена на рисунку 2.3

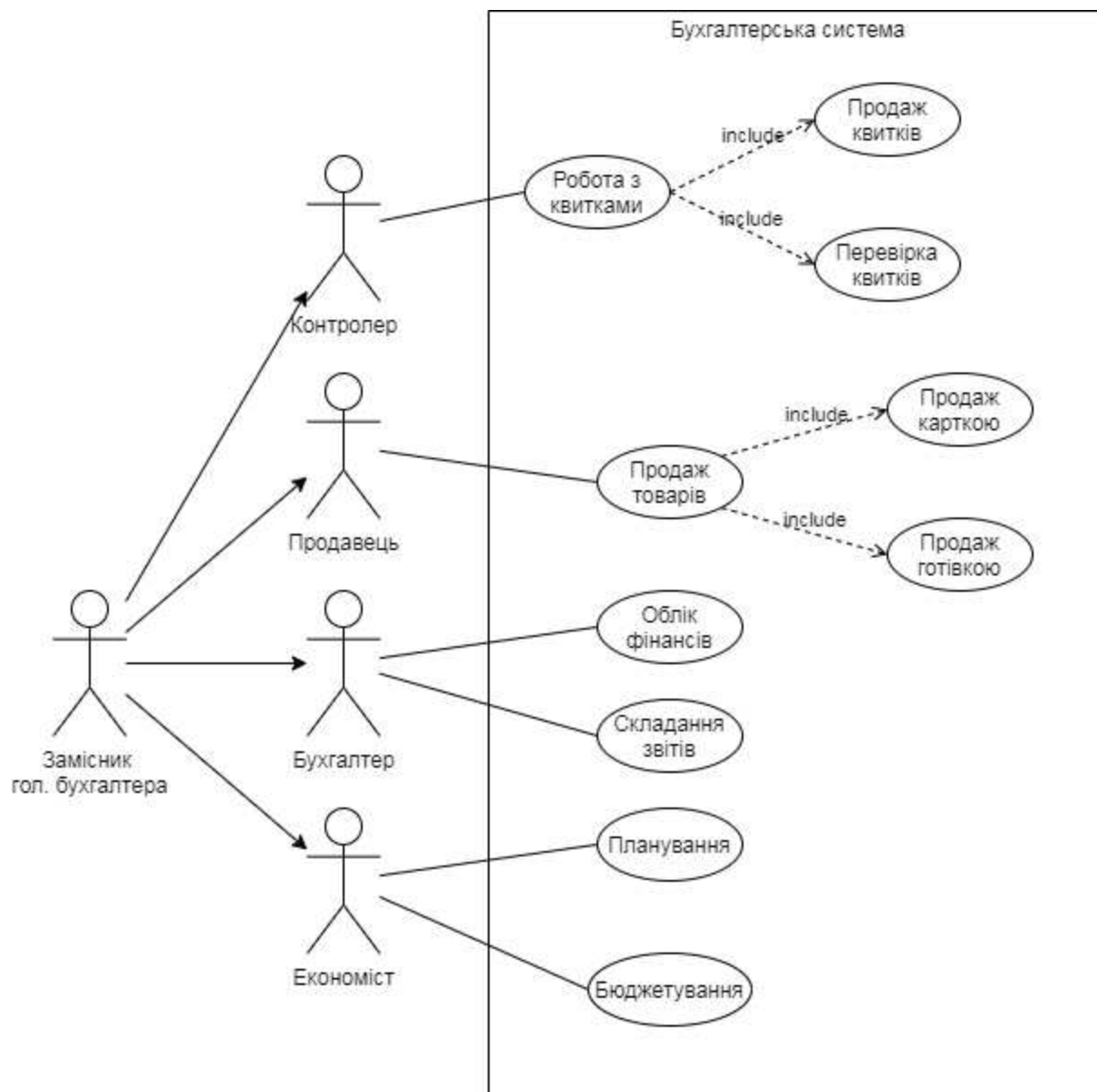


Рисунок 2.3 - Прецеденти бухгалтерський відділ

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.

2.4 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.[21].

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.)[3]
- 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 [22].

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[27].

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[18], 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 [23].

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 [46].

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 тегамі, 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 [24].

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 [25].

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 проектувальників 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 [28].

Проект розроблявся у інтегрованому середовищі розробки IntelliJ IDEA.

IntelliJ IDEA — комерційне інтегроване середовище розробки для різних мов програмування (Java, Python, Scala, PHP та ін.[4]) 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 проектів мають можливість отримати безкоштовну ліцензію.

Community версія середовища IntelliJ IDEA підтримує інструменти (у вигляді плагінів) для проведення тестування TestNG[en] 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 проектування 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 проектів 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 [].

За допомогою даних засобів розроблено програмне забезпечення інформаційної системи.

Оскільки ми маємо типове програмне забезпечення, а саме 1С Бухгалтерія, його потрібно адаптувати з прив'язкою до документації зоопарку.

На решту завдань проекту виконано розробки програмного забезпечення, сформовано базу даних, розроблено додатки, проведено їх тестування. Код за розробленими програмами наводиться в додатку. Апробація показала, що програмні засоби функціонують відповідно до поставлених завдань

2.5 Рішення з технічного забезпечення

Оскільки для взаємодії з інформаційною системою користувачу буде потрібен браузер пристрої повинні мати такі характеристики, які будуть підтримувати стабільну роботу браузера.

Мінімальні технічні та програмні вимоги на комп'ютерах

- Windows 7 32/64-bit , Windows 10 32/64-bit;
- ОЗУ не менше 8096 Мб;
- Клавіатура;
- Миша;
- Розширення екрану не менше 800 × 600;
- Монітор;
- Відео пам'ять не менше 128 мб;
- Жорсткий диск не менше ніж 500 Гб.

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

- Хостинг VPS;
- Тактова частота процесору не нижче 3.1 ГГц;
- ОЗУ не менше 8 Гб;
- Жорсткий диск не менше ніж 5 Тб;
- СУБД MySQL;
- Клавіатура;
- Миша;
- Монітор.

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

3.Реалізація проекту інформаційної системи “Управління зоопарком”

Для експлуатації системи виконано розробку робочої документації за допомогою якої користувач та адміністратор можуть користуватися системою.

Оскільки програма має велику кількість полів, для полегшення роботи користувача створені підказки. Також в разі помилки, на екрані виводиться інформація щодо неї. В разі якщо форма заповнена неповністю, на екрані виводиться інформація які поля мають бути обов’язково заповнені.

Проведено апробацію інформаційної системи. Перевірена серверна частина інформаційної системи. Перевірена робота при введенні даних користувачем.

Оскільки для бухгалтерської частини роботи використовується система 1С Бухгалтерія потрібна відповідна робоча документація.

Вкладка кадровий облік в 1С Бухгалтерії представлено на рисунку 3.1

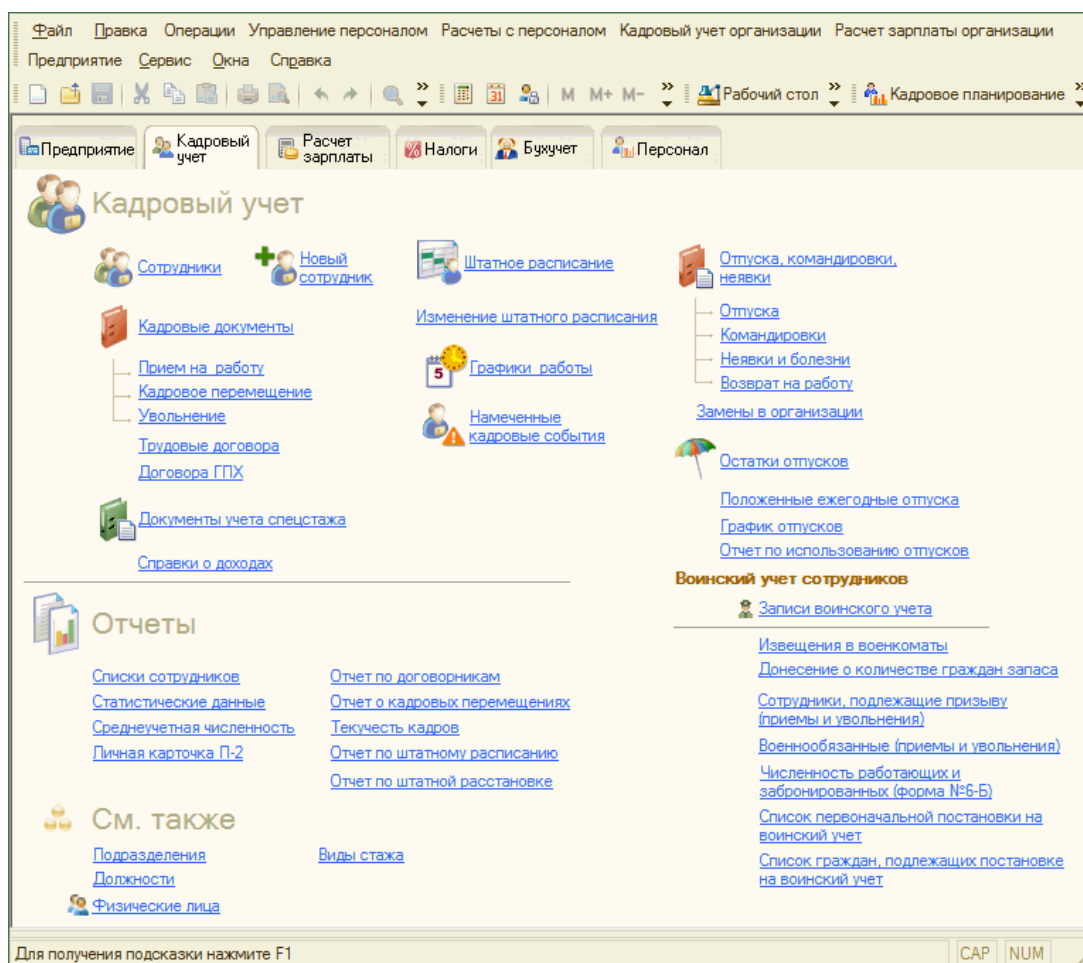


Рисунок 3.1 – Кадровий облік

Приклад розрахунку зарплати в 1С Бухгалтерії представлено на рисунку 3.2

Начисление зарплаты работникам организаций: Проведен

Действия:

Звітність в 1С Бухгалтерії представлено на рисунку 3.4

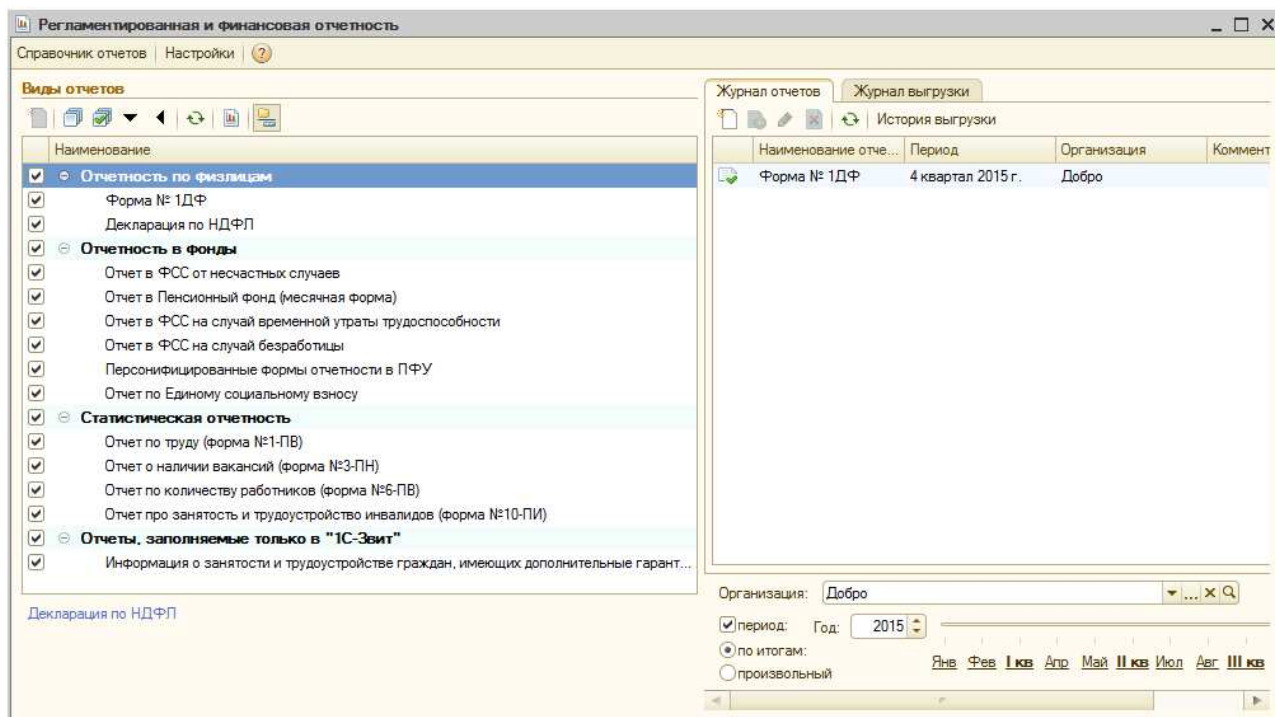


Рисунок 3.4 – Звітність

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	Дни	Часы				
1. Начислено							2. Удержано		
Оклад по дням	1-31 Сер 15	20	160	20		5 000,00	ЕСВ	1-31 Сер 15	309,09
Индексация зарплаты	1-31 Сер 15					3 585,79	Військовий збір	1-31 Сер 15	128,79
							ПДФО обчислений	1-31 Сер 15	1 241,51
Всего начислено						8 585,79	Всего 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

Реалізація показала, що система підвищує ефективність в управлінні роботи зоопарком, оскільки зменшується кількість ручної роботи, звідки навантаження на касирів, продавців та бухгалтерів значно зменшується.

4.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 [31].

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, лігати 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 місцях 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 дражнить їх, кидає предмети в експонати або сміється 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 сміяти
- Відвідувачам заборонено відвідувати службові приміщення.
- Не заважати іншим гостям або роботі зоопарку [34].

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:

- самотійно 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 [35].

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

Висновок

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

Розроблена концепція інформаційної системи, проведено обстеження підприємства, проведено аналіз інформаційних процесів, сформовано вимоги до проекрованої системи, визначено цілі та завдання проекту. Розроблений технічний проект системи в якому представлені 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. Розроблена робоча документація та перевірена працездатність системи. Представлені правила та рекомендації щодо безпеки відвідувачів та працівників у зоопарку, а також правила безпеки за комп'ютером. Проведена апробація проекту. Сервіс показує безперебійну та стабільну роботу.

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

Список використаних джерел інформації

1. Білов В.В., В.І. Чистякова — Проектування інформаційних систем: підручник для студ. установ вищої. проф. Освіта; за ред. В.В. Білова М: Видавничий центр "Академія", 2013. 352с.
2. Гагаріна Л.Г., Кокорева Є.В., Віснадул Б.Д. Технологія розробки програмного забезпечення: навчальний посібник/під ред. Л.Г. Гагаріна. — М.: ВД «ФОРУМ»: ИНФА-М, 2008.— 400 с.: іл.— (Вища освіта).
3. ГОСТ 34.601-90 «Інформаційна технологія. Комплекс стандартів на автоматизовані системи. Автоматизовані системи. Стадії створення».
4. ГОСТ 34.602-89 «Інформаційна технологія. Комплекс стандартів на автоматизовані системи. Технічне завдання створення автоматизованої системи».
5. Грекул В.І., Г.М. Денищенко. Проектування інформаційних систем: курс лекцій. М. Інтернет - Ун-т Інформ. технологій, 2005. 304с.
6. Клевцов С.І. Аналіз та формування вимог до програмного забезпечення інформаційних систем збору та обробки даних: навчальний посібник. Таганрог: Вид-во ТТІЮФ, 2007. 100с.
7. Марченко А.В. Проектування інформаційних систем 2016 90с. Режим доступу:
https://elearning.sumdu.edu.ua/free_content/lectured:de1c9452f2a161439391120eef364dd8ce4d8e5e/20160217112601/content-20160217112601.pdf.
8. Пальмов С.В. Проектування інформаційних систем. М: ІСТ-72, 2011. 51 с.
9. Ременяк Л.В. Проектування інформаційних систем: конспект лекцій. Одеса, Одеський державний екологічний університет, 2016. 152.
10. Хомоненко О.Д., Циганков В.М., Мальцев М.Г. Бази даних: Підручник для вищих навчальних закладів/Під ред. Проф. А.Д. Хомоненко. — 6-те вид.дод. — СПб.: КОРОНА – Вік. 2009. — 736 с.
11. <https://uk.wikipedia.org/wiki/Зоопарк>
12. <https://www.thebalancecareers.com/zoo-director-125935>
13. <https://www.softplay.com/blog/how-to-boost-attendance-at-your-zoo/>

14. <https://www.fresnochaffeezoo.org/wp-content/uploads/2015/09/COO-Deputy-Director-02192017.pdf>
15. <https://en.wikipedia.org/wiki/Economist>
16. <http://www.azfa.org/job/plumber-2/>
17. <http://www.azfa.org/job/electrician/>
18. <https://jobs.uic.edu/job-board/job-details?jobID=112321>
19. https://www.hoglezoo.org/jobs_volunteers/volunteers/zoo-guides/
20. <https://www.thebalancecareers.com/zoo-veterinarian-125943>
21. <https://www.thebalancecareers.com/animal-nutritionist-125606>
22. https://newsroom.unl.edu/announce/snr_alumni/6179/38961
23. <https://resources.workable.com/hr-manager-job-description>
24. https://ru.wikipedia.org/wiki/Бухгалтерская_система#Персональные_системы
25. <https://ru.wikipedia.org/wiki/GnuCash>
26. <https://ru.wikipedia.org/wiki/KMyMoney>
27. <https://ru.wikipedia.org/wiki/Tryton>
28. https://ru.wikipedia.org/wiki/Microsoft_Dynamics_AX
29. <https://en.wikipedia.org/wiki/GnuCash>
30. https://m.medias.com.ua/files/opys_progrm/1C-Buhgalterija.pdf
31. <https://uk.wikipedia.org/wiki/MySQL>
32. <https://uk.wikipedia.org/wiki/SQL>
33. <https://uk.wikipedia.org/wiki/JavaScript>
34. <https://uk.wikipedia.org/wiki/HTML>
35. <https://uk.wikipedia.org/wiki/CSS>
36. <https://www.guru99.com/data-modelling-conceptual-logical.html>
37. <https://uk.wikipedia.org/wiki/XAMPP>
38. <https://uk.wikipedia.org/wiki/Java>
39. <https://uk.wikipedia.org/wiki/XHTML>
40. https://uk.wikipedia.org/wiki/IntelliJ_IDEA

41. https://en.wikipedia.org/wiki/United_States_environmental_and_occupational_health_in_zoos
42. https://en.wikipedia.org/wiki/Use_case_diagram
43. https://en.wikipedia.org/wiki/Unified_Modeling_Language
44. <https://louisvillezoo.org/plan/know-before-you-go/>
45. <https://www.sop.com.ua/article/485-nstruktsya-z-ohoroni-prats-pri-robot-na-personalnomu-kompyuter>
46. <https://en.wikipedia.org/wiki/GlassFish>

Додаток А – Інструкція користувачу

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...	Д...	П...	Нарахування	Період	Базовий період	Результат	Розмір	Відпрацьовано	Норма за місяць	Оплачено	Спосіб відображення в...
					Дата почат...	Дата закін...			Днів	Годин	Днів	Годин
1	0000004	Абадулов Юрій Володимирович (осн.)	Адміністрація	Оклад по днях	01.12.2018	31.12.2018	3 200,00	3 200,000	20,00	158,00	20,00	158,00
2	0000001	Абадулов Юрій Володимирович (осн.)	Адміністрація	Доплата до мініма...	01.12.2018	31.12.2018	523,00	20,000			20,00	158,00
3	0000001	Іванцова Лілія Євгенівна (осн.)	Адміністрація	Індексація зарплати	01.12.2018	31.12.2018	276,62	20,000			20,00	158,00
4	0000002	Аксенов Артур П.	Адміністрація	Оклад по днях	01.12.2018	31.12.2018	3 500,00	3 500,000	20,00	158,00	20,00	158,00
5	0000002	Аксенов Артур П.	Адміністрація	Індексація зарплати	01.12.2018	31.12.2018	276,62	20,000			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.

N	К...	Д...	П...	Податковий період	Дохід ПДФО	Від ставки	Збільшена ставка	Дохід	Податок	Пільга	Кількість...	Дохід (повний)
1	0000004	Абадулов Юрій Володимирович (осн.)	Адміністрація	Грудень 2018	101	Основа	Збільшена	4 776,62	959,79			4 776,62
2	0000001	Абадулов Юрій Володимирович (осн.)	Адміністрація	Грудень 2018	ВЗ	Військовий збір	Військовий збір	4 776,62	71,65			4 776,62
3	0000001	Іванцова Лілія Євгенівна (осн.)	Адміністрація	Грудень 2018	101	Основа	Збільшена	3 776,62	679,79			3 776,62
4	0000002	Аксенов Артур П.	Адміністрація	Грудень 2018	ВЗ	Військовий збір	Військовий збір	3 776,62	56,65			3 776,62
5	0000002	Аксенов Артур П.	Адміністрація	Грудень 2018	101	Основа	Збільшена	3 999,62	719,93			3 999,62
6	0000001	Абадулов Юрій Володимирович (осн.)	Адміністрація	Грудень 2018	ВЗ	Військовий збір	Військовий збір	3 999,62	59,99			3 999,62
7	0000002	Горбатова Ірина Іванівна	Адміністрація	Грудень 2018	101	Основа	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 організацій: 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: Адмінченко Віктор Сергійович Базовий період індексації соц. виплат: ...

Н N Tab. №: Працівник (основне призначення)

N	Tab. №	Працівник (основне призначення)	Податок	Вид розрахунку	Період	База	База (з обмеженнями)	Результат	Стаття податкової декларації	Період дії	Межа	База (вдарава...)	Ставка	Код кат. 30	Тип
					Початок	Кінець	в т.ч. відпос...	в т.ч. донар. до мін.		Поч...	Зак...	Мін...	Коефіціє...		
1	0000004	Абдулов Юрій Володимирович (осн.)	ЄСВ ФОП (працівник)	ЄСВ ФОП (оплата ...)	01.12.2...	31.12.2...	4 776,62	4 776,62	1 050,860000	Нарахування ЄСВ роботодавця (22 %)	01....	31....	55...	0,2200	1
2	0000001	Іванцова Лілія Євгенівна (осн.)	ЄСВ ФОП (працівник)	ЄСВ ФОП (оплата ...)	01.12.2...	31.12.2...	3 776,62	3 776,62	830,860000	Нарахування ЄСВ роботодавця (22 %)	01....	31....	55...	0,2200	1
3	0000002	Богословський Федір Михайлович (осн.)	ЄСВ ФОП (працівник)	ЄСВ ФОП (оплата ...)	01.12.2...	31.12.2...	5 276,62	5 276,62	1 160,860000	Нарахування ЄСВ роботодавця (22 %)	01....	31....	55...	0,2200	1
4	0000001	Іванцова Лілія Євгенівна (осн.)	ЄСВ ФОП (працівник)	ЄСВ ФОП (оплата ...)	01.12.2...	31.12.2...					01....	31....	55...	1,0000	29

Рисунок А.3 – Внески ФОП

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

Нарахування зарплати працівникам організацій: Не проведений *

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: Адмінченко Віктор Сергійович Базовий період індексації соц. виплат: ...

Н N Tab. №: Працівник (основне призначення)

N	Tab. №	Працівник (основне призначення)	Податок	Вид розрахунку	Період	База	База (з обмеженнями)	Результат	Стаття податкової декларації	Період дії	Межа	База (вдарава...)	Ставка	Код кат. 30	Тип
					Початок	Кінець	в т.ч. відпос...	в т.ч. донар. до мін.		Поч...	Зак...	Мін...	Коефіціє...		
1	0000004	Абдулов Юрій Володимирович (осн.)	ЄСВ ФОП (працівник)	ЄСВ ФОП (оплата ...)	01.12.2...	31.12.2...	4 776,62	4 776,62	1 050,860000	Нарахування ЄСВ роботодавця (22 %)	01....	31....	55...	0,2200	1
2	0000001	Іванцова Лілія Євгенівна (осн.)	ЄСВ ФОП (працівник)	ЄСВ ФОП (оплата ...)	01.12.2...	31.12.2...	3 776,62	3 776,62	830,860000	Нарахування ЄСВ роботодавця (22 %)	01....	31....	55...	0,2200	1
3	0000002	Богословський Федір Михайлович (осн.)	ЄСВ ФОП (працівник)	ЄСВ ФОП (оплата ...)	01.12.2...	31.12.2...	5 276,62	5 276,62	1 160,860000	Нарахування ЄСВ роботодавця (22 %)	01....	31....	55...	0,2200	1
4	0000001	Іванцова Лілія Євгенівна (осн.)	ЄСВ ФОП (працівник)	ЄСВ ФОП (оплата ...)	01.12.2...	31.12.2...					01....	31....	55...	1,0000	29

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 Лілія Євгенівна	5 276,62		(не вик.) Госп. д-сть
2	92	Окладні й тарифні (А3)	661	Богословський Федір Михайлович	4 776,62		(не вик.) Госп. д-сть
3	92	Окладні й тарифні (А3)	661	Абдулов Юрій Володимирович	3 999,62		(не вик.) Госп. д-сть
4	92	Окладні й тарифні (А3)	661	Аксьонов Артур Петрович	3 776,62		(не вик.) Госп. д-сть
5	92	Окладні й тарифні (А3)	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	Військовий збір	1-31 Груд 18	71,65
Індексація зарплати	1-31 Груд 18				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	00000002	Богословський Федір Михайлович (5006086932)	3 845,18		
2	00000003	Аксьонов Артур Петрович (7712071515)	3 040,18		
3	00000004	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.webzoodb.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(regexp = "^(?![А-Яа-яёЁ0-9 -]+$)",message = "Кличка: Недопустимые символы")
    @Column(name = "name")
    private String name;

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

    @Pattern(regexp = "^(?![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.webzoodb;
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.sexes}" var="s"
itemValue="#{s}" itemLabel="#{s}"/>
                </h:selectOneMenu>
            </div>
            <div style="margin-top: auto">
                <h:inputText styleClass="input" id="inputBirthDate"
p:type="date"
                                value="#{animalBean.animal.birthDate}"
                                required="true" requiredMessage="Введите
дату"/>
            </div>
            <div style="margin-top: auto">
                <h:inputText styleClass="input" p:type="date"
value="#{animalBean.animal.deadDate}"/>
            </div>
            <div>
                <h:commandButton styleClass="tools"

```

```

image="../../../img/add.png" action="#{animalBean.init}" >
    <f:ajax execute="@form" render="entityForm addForm
messages" event = "click"

listener="#{animalBean.add(employeeBean.employees) }"/>
    </h:commandButton>
</div>
</h:form>
</div>
<div class="inputform">
    <h:form styleClass="inputform" id="searchForm" >
        <div style="margin-top: auto">
            <h:outputLabel for="inputEmp"/>
            <h:selectOneMenu value="#{employeeBean.employee.id}"

valueChangeListener="#{animalBean.employeefrompage}" styleClass="input">
            <f:selectItem 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 p:placeholder="Кличка" p:type="text"
value="#{animalBean.animal.name}"
                                styleClass="input"/>
        </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>
                                <div>
                                <h:messages id="messages"/>
                                </div>
                                </h:body>
                                </f:view>
                                </html>

```