**ЗМІСТ**

[ПЕРЕЛІК ПОЗНАК ТА СКОРОЧЕНЬ 2](#_Toc198042122)

[ВСТУП 3](#_Toc198042123)

[1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 4](#_Toc198042124)

[1.1 Система бізнес-правил 4](#_Toc198042125)

[1.2 Моделювання предметної області 5](#_Toc198042126)

[1.3 Постановка задачі 6](#_Toc198042127)

[2 ПРОЄКТУВАННЯ ТА РОЗРОБКА БАЗИ ДАНИХ 8](#_Toc198042128)

[2.1 Розробка логічної та фізичної моделей даних 8](#_Toc198042129)

[2.2 Опис структури бази даних 9](#_Toc198042130)

[2.3 Реалізація бази даних у СУБД MySQL 11](#_Toc198042131)

[2.4 Заповнення бази даних початковими записами 14](#_Toc198042132)

[2.5 Створення запитів до бази даних 19](#_Toc198042133)

[3 ПРОЄКТУВАННЯ ТА РОЗРОБКА ЗАСТОСУНКУ ДЛЯ РОБОТИ З БАЗОЮ ДАНИХ 21](#_Toc198042134)

[3.1 Проєктування застосунку для роботи з базою даних 21](#_Toc198042135)

[3.2 Приклад використання застосунку для роботи з базою даних 21](#_Toc198042136)

[ВИСНОВКИ 24](#_Toc198042137)

[СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ 25](#_Toc198042138)

[ДОДАТОК А 26](#_Toc198042139)

**ПЕРЕЛІК ПОЗНАК ТА СКОРОЧЕНЬ**

БД – база даних.

СУБД – система управління базами даних.

DDL – Data Definition Language.

ER – Entity-Relationship.

SQL – Structured Query Language.

**ВСТУП**

В сучасному світі проблема безпритульних тварин залишається актуальною та потребує ефективних рішень. Притулки для тварин відіграють важливу роль у забезпеченні догляду та захисту безпритульних тварин, їх лікуванні та пошуку нових домівок. Для функціонування таких закладів необхідна правильна організація всіх робочих процесів та належне управління інформацією.

Автоматизація процесів управління притулком для тварин за допомогою спеціалізованої інформаційної системи дозволяє значно підвищити ефективність роботи закладу. Така система забезпечує централізоване зберігання та швидкий доступ до інформації про тварин, спрощує ведення обліку, допомагає в організації роботи персоналу та формуванні необхідної звітності.

Метою даної роботи є розробка інформаційної системи для управління притулком для тварин, яка дозволить автоматизувати основні робочі процеси закладу. Система призначена для ведення обліку тварин, що перебувають у притулку, керування інформацією про їх стан здоров'я та щеплення, організації роботи персоналу та формування необхідної звітності.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- проаналізувати предметну область та визначити основні вимоги до системи;

- спроектувати архітектуру бази даних;

- розробити програмне забезпечення для автоматизації основних процесів;

- забезпечити формування необхідної звітності.

**1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ**

**1.1 Система бізнес-правил**

Функціональні вимоги до інформаційної системи "Притулок для тварин" мають бути спрямовані на ефективне управління тваринним притулком, забезпечення високого рівня догляду за тваринами та оптимізацію всіх аспектів його діяльності.

Визначено такі бізнес-вимоги:

- управління даними про тварин (зберігання та управління даними про тварин - інформацію про здоров'я, характер, поведінку, історію хвороб тощо);

- організація роботи персоналу (організація роботи персоналу притулку, такої як планування роботи, облік робочого часу, управління завданнями та звітність перед керівництвом);

- аналітика та звітність (аналіз даних та звіти, такі як звітність про роботу притулку, статистику щодо тварин та працівників).

Також проаналізовано, які вимоги в користувачів є до розробленої інформаційної системи.

Для ролі «співробітник» передбачити реалізацію наступних функцій:

1 авторизація в застосунку;

2 додавання нової тварини;

3 перегляд звітності про тварин у притулку;

4 перенесення тварини до архіву;

5 формування звіту про спожиту кількість корму для тварин.

Для ролі «адміністратор» додатково реалізувати наступні функції:

6 додавання співробітників;

7 перегляд співробітників;

Для ролі «лікар» передбачити реалізацію наступних функцій:

8 внесення інформації про щеплення тварин.

Вимоги до функцій інформаційної системи:

1 кожна Тварина характеризується ім’ям, видом, породою, датою народження та датою надходження у притулок, статтю, кольором, вагою, можливістю бути відданою з притулку (adoptable), датою вилучення з притулку (необов’язково);

2 кожен Співробітник має прізвище, ім’я, займану посаду, телефон, електронну пошту, логін та пароль;

3 кожен Співробітник може додати Тварину, один Співробітник може доглядати за декількома Тваринами та одна Тварина може бути під наглядом декількох Співробітників;

4 таблиця Розпорядок догляду зберігає в собі інформацію про початок та кінець догляду у вигляді дати, день тижня, коли відбувається догляд, ідентифікатор Тварини та Співробітника;

5 таблиця Медичній нагляд містить інформацію про причину нагляду, дату початку і дату завершення та ідентифікатор тварини;

6 інформація про Щеплення включає в себе призначення щеплення, дату щеплення та тварину, якій робиться щеплення

7 одне щеплення робиться декільком тваринам та одна тварина може мати декілька щеплень

8 для кожної тварини, що прилаштовується у «добрі руки», є таблиця Прилаштунок, яка має атрибути ідентифікатор тварини, прізвище опікуна, ім’я опікуна, телефон, електронну пошту та дату прилаштунку тварини.

**1.2 Моделювання предметної області**

Модель предметної області у нотації ER (Entity-Relationship, сутність-зв’язок) наведена на рисунку 1.1.

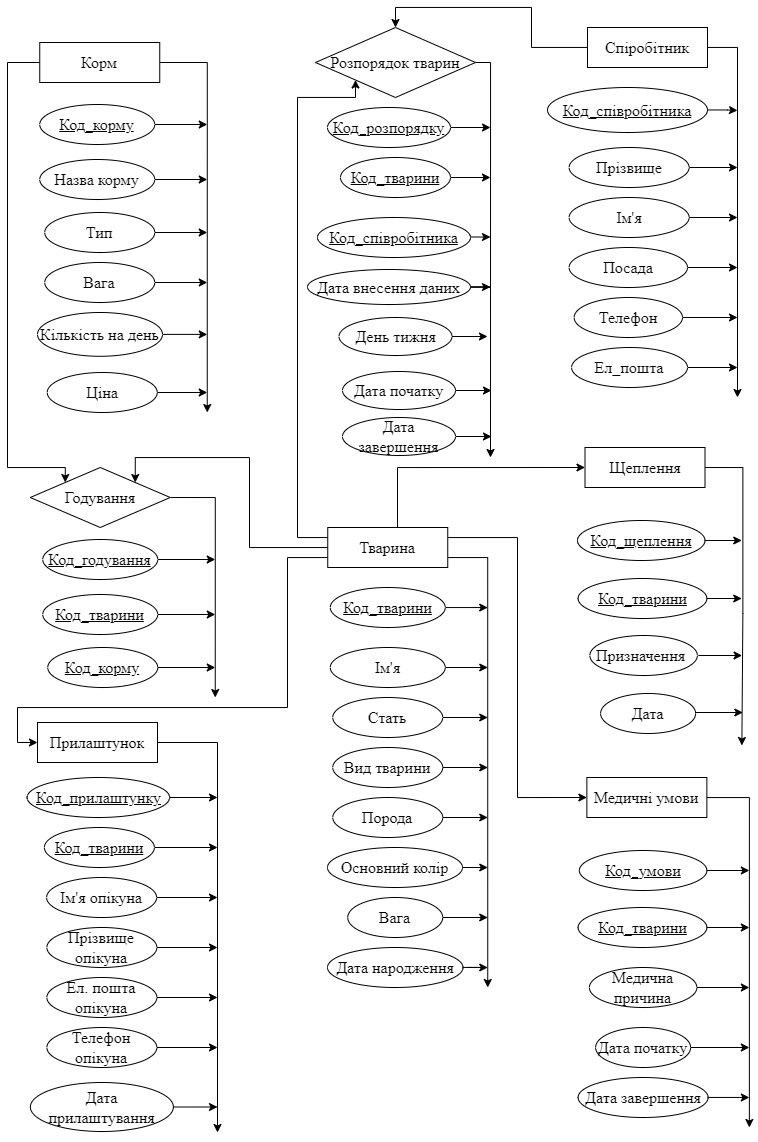


Рисунок 1.1 – ER-модель предметної області

**1.3 Постановка задачі**

Для досягнення мети роботи необхідно виконати наступні задачі:

1 Розробити логічну модель бази даних.

2 Розробити фізичну модель бази даних.

3 Описати структуру бази даних.

4 Реалізувати базу даних у системі управління базами даних (СУБД) MySQL за допомогою команд DDL (Data Definition Language).

5 Заповнити базу даних початковими записами та розробити запити мовою SQL (Structured Query Language).

6 Розробити застосунок для роботи з базою даних.

7 Продемонструвати використання застосунку для роботи з базою даних.

**2 ПРОЄКТУВАННЯ ТА РОЗРОБКА БАЗИ ДАНИХ**

**2.1 Розробка логічної та фізичної моделей даних**

Логічна модель даних у нотації UML наведена на рисунку 2.1.

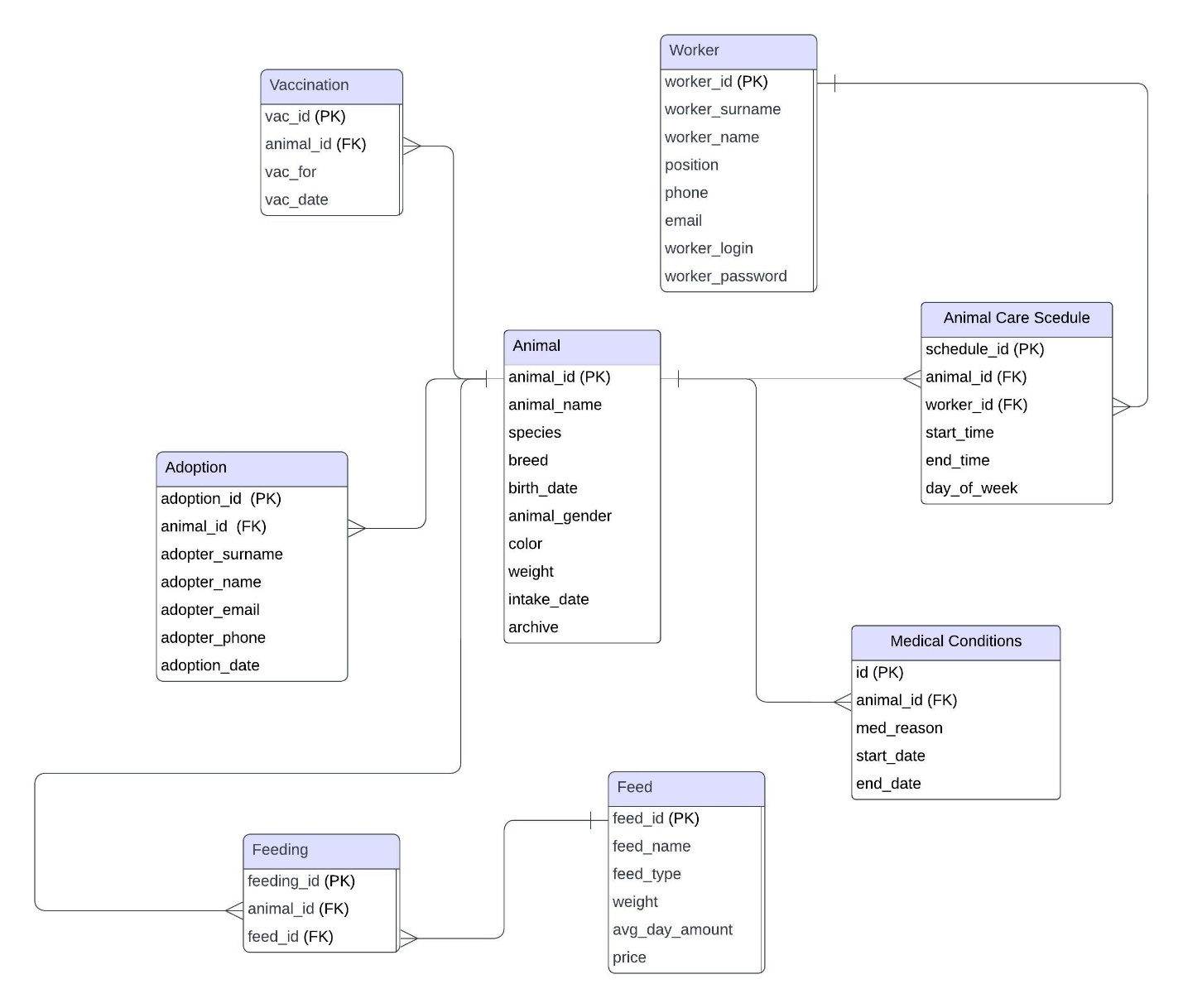


Рисунок 2.1 – Логічна модель даних

Фізична модель даних у нотації UML наведена на рисунку 2.2.

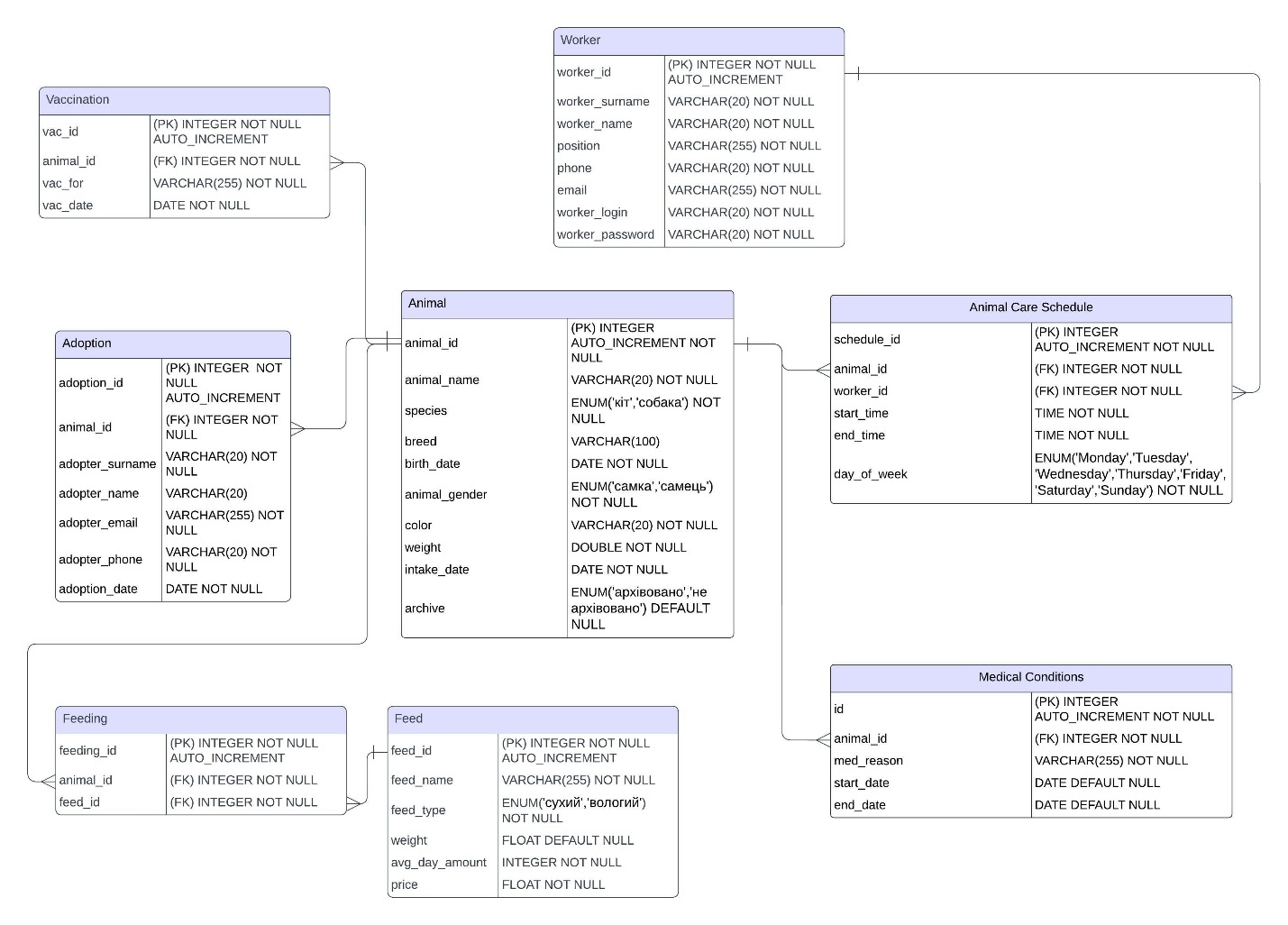


Рисунок 2.2 – Фізична модель даних

**2.2 Опис структури бази даних**

База даних притулку для тварин складається з таких таблиць, як Тварина, Співробітник, Розпорядок тварини, Медичні умови, Вакцинація, Прилаштування тварин, Корм, Годування.

В цій базі даних існують такі зв’язки між таблицями:

* між Твариною та Розпорядком тварин – один до багатьох;
* між Співробітником та Розпорядком тварин – один до багатьох;
* між Твариною та Вакцинацією – один до багатьох;
* між Твариною та Медичними умовами – один до багатьох;
* між Твариною та Прилаштуванням тварин – один до одного;
* між Кормом і Годуванням – один до багатьох;
* між Годуванням і Твариною – багато до одного.

Таблиця 3.1 - Відношення «Тварина»

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Код тварини | Ім’я | Вид | Порода | Дата народження | Стать | Колір | Вага | Дата прийому | Архівованість |
|  |  |  |  |  |  |  |  |  |  |

Таблиця 3.2 - Відношення «Співробітник»

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Код співробітника | Прізвище | Ім’я | Посада | Телефон | Електронна пошта | Логін | Пароль |
|  |  |  |  |  |  |  |  |

Таблиця 3.3 - Відношення «Розпорядок тварини»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Код розпорядку | Код тварини | Код співробітника | Час початку | Час завершення | День тижня |
|  |  |  |  |  |  |

Таблиця 3.4 - Відношення «Медичні умови»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Код умови | Код тварини | Медична причина | Дата початку | Дата завершення |
|  |  |  |  |  |

Таблиця 3.5 - Відношення «Вакцинація»

|  |  |  |  |
| --- | --- | --- | --- |
| Код вакцинації | Код тварини | Призначення | Дата вакцинації |
|  |  |  |  |

Таблиця 3.6 - Відношення «Прилаштування тварини»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Код прилаштування | Код тварини | Прізвище опікуна | Ім’я опікуна | Електронна пошта | Телефон | Дата прилаштування |
|  |  |  |  |  |  |  |

Таблиця 3.7 - Відношення «Корм»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Код корму | Назва корму | Тип | Вага | Середня кількість на день | Ціна |
|  |  |  |  |  |  |

Таблиця 3.8 - Відношення «Годування»

|  |  |  |
| --- | --- | --- |
| Код годування | Код тварини | Код корму |
|  |  |  |

**2.3 Реалізація бази даних у СУБД MySQL**

Схема даних для бази даних, що реалізована засобами СУБД MySQL, наведена на рисунку 2.3.

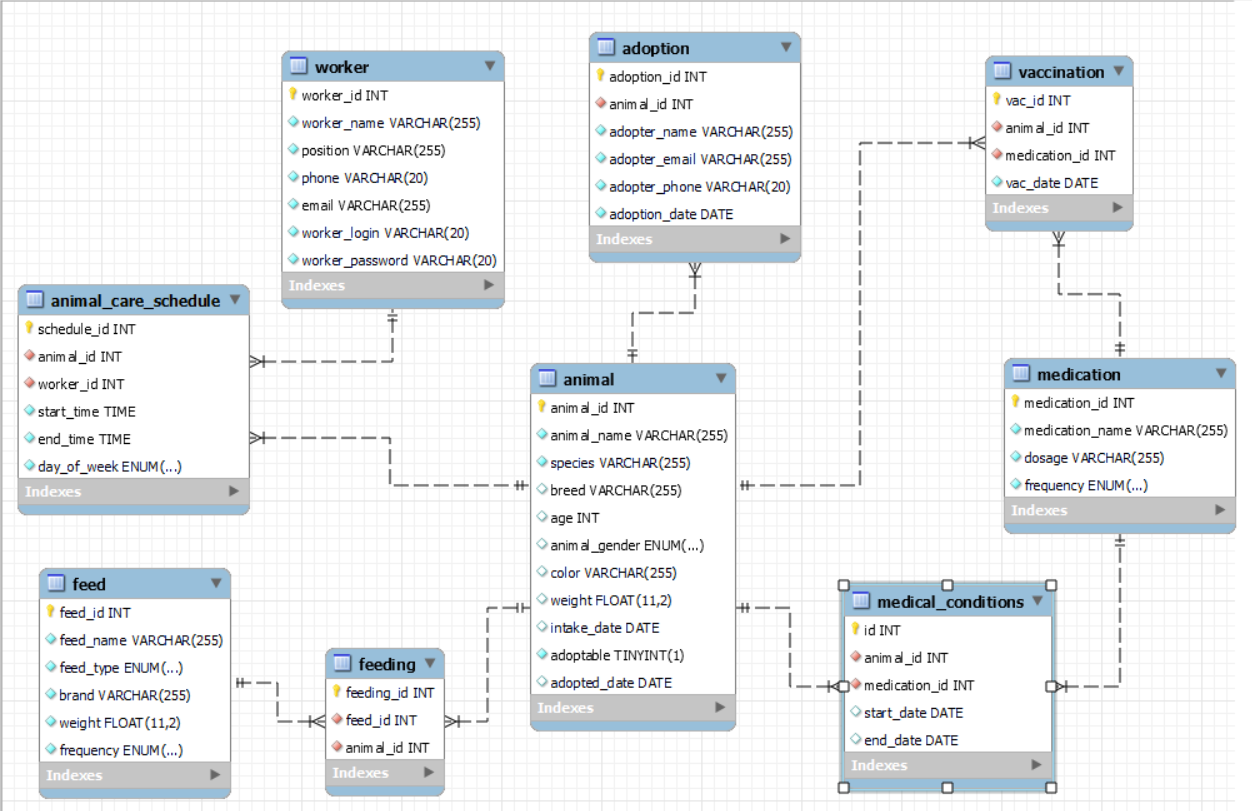


Рисунок 2.3 – Схема розробленої бази даних

Для створення бази даних у СУБД MySQL було використано наступні команди DDL:

CREATE DATABASE petshelter;

CREATE TABLE `animal` (

`animal\_id` int NOT NULL AUTO\_INCREMENT,

`animal\_name` varchar(20) COLLATE cp1251\_ukrainian\_ci NOT NULL,

`species` enum('кіт','собака') COLLATE cp1251\_ukrainian\_ci NOT NULL,

`breed` varchar(100) COLLATE cp1251\_ukrainian\_ci NOT NULL,

`birth\_date` date NOT NULL,

`animal\_gender` enum('самка','самець') COLLATE cp1251\_ukrainian\_ci NOT NULL,

`color` varchar(20) COLLATE cp1251\_ukrainian\_ci NOT NULL,

`weight` double NOT NULL,

`intake\_date` date NOT NULL,

`archive` enum('архівовано','не архівовано') COLLATE cp1251\_ukrainian\_ci DEFAULT NULL,

PRIMARY KEY (`animal\_id`)

)ENGINE=InnoDB;

CREATE TABLE `worker` (

`worker\_id` int NOT NULL AUTO\_INCREMENT,

`worker\_surname` varchar(20) COLLATE cp1251\_ukrainian\_ci DEFAULT NULL,

`worker\_name` varchar(20) COLLATE cp1251\_ukrainian\_ci NOT NULL,

`position` varchar(255) COLLATE cp1251\_ukrainian\_ci NOT NULL,

`phone` varchar(20) COLLATE cp1251\_ukrainian\_ci NOT NULL,

`email` varchar(255) COLLATE cp1251\_ukrainian\_ci NOT NULL,

`worker\_login` varchar(20) COLLATE cp1251\_ukrainian\_ci NOT NULL,

`worker\_password` varchar(20) COLLATE cp1251\_ukrainian\_ci NOT NULL,

PRIMARY KEY (`worker\_id`),

UNIQUE KEY `worker\_login` (`worker\_login`)

) ENGINE=InnoDB;

CREATE TABLE `animal\_care\_schedule` (

`schedule\_id` int NOT NULL AUTO\_INCREMENT,

`animal\_id` int NOT NULL,

`worker\_id` int NOT NULL,

`start\_time` time NOT NULL,

`end\_time` time NOT NULL,

`day\_of\_week` enum('Monday','Tuesday','Wednesday','Thursday','Friday','Saturday','Sunday') COLLATE cp1251\_ukrainian\_ci NOT NULL,

PRIMARY KEY (`schedule\_id`),

KEY `schedule\_animal` (`animal\_id`),

KEY `schedule\_worker` (`worker\_id`),

CONSTRAINT `schedule\_animal` FOREIGN KEY (`animal\_id`) REFERENCES `animal` (`animal\_id`),

CONSTRAINT `schedule\_worker` FOREIGN KEY (`worker\_id`) REFERENCES `worker` (`worker\_id`)

)ENGINE=InnoDB;

CREATE TABLE `vaccination` (

`vac\_id` int NOT NULL AUTO\_INCREMENT,

`animal\_id` int NOT NULL,

`vac\_for` varchar(255) COLLATE cp1251\_ukrainian\_ci NOT NULL,

`vac\_date` date NOT NULL,

PRIMARY KEY (`vac\_id`),

KEY `vac\_animal` (`animal\_id`),

CONSTRAINT `vac\_animal` FOREIGN KEY (`animal\_id`) REFERENCES `animal` (`animal\_id`)

)ENGINE=InnoDB;

CREATE TABLE `feed` (

`feed\_id` int NOT NULL AUTO\_INCREMENT,

`feed\_name` varchar(255) COLLATE cp1251\_ukrainian\_ci NOT NULL,

`feed\_type` enum('сухий','вологий') COLLATE cp1251\_ukrainian\_ci NOT NULL,

`weight` float DEFAULT NULL,

`avg\_day\_amount` int NOT NULL,

`price` float NOT NULL,

PRIMARY KEY (`feed\_id`)

) ENGINE=InnoDB;

CREATE TABLE `feeding` (

`feeding\_id` int NOT NULL AUTO\_INCREMENT,

`feed\_id` int NOT NULL,

`animal\_id` int NOT NULL,

PRIMARY KEY (`feeding\_id`),

KEY `feeding\_animal` (`animal\_id`),

KEY `feeding\_feed` (`feed\_id`),

CONSTRAINT `feeding\_animal` FOREIGN KEY (`animal\_id`) REFERENCES `animal` (`animal\_id`),

CONSTRAINT `feeding\_feed` FOREIGN KEY (`feed\_id`) REFERENCES `feed` (`feed\_id`)

) ENGINE=InnoDB;

CREATE TABLE `medical\_conditions` (

`id` int NOT NULL AUTO\_INCREMENT,

`animal\_id` int NOT NULL,

`med\_reason` varchar(255) COLLATE cp1251\_ukrainian\_ci NOT NULL,

`start\_date` date DEFAULT NULL,

`end\_date` date DEFAULT NULL,

PRIMARY KEY (`id`),

KEY `med\_animal` (`animal\_id`),

CONSTRAINT `med\_animal` FOREIGN KEY (`animal\_id`) REFERENCES `animal` (`animal\_id`)

) ENGINE=InnoDB;

CREATE TABLE `adoption` (

`adoption\_id` int NOT NULL AUTO\_INCREMENT,

`animal\_id` int NOT NULL,

`adopter\_surname` varchar(20) COLLATE cp1251\_ukrainian\_ci NOT NULL,

`adopter\_name` varchar(20) COLLATE cp1251\_ukrainian\_ci NOT NULL,

`adopter\_email` varchar(255) COLLATE cp1251\_ukrainian\_ci NOT NULL,

`adopter\_phone` varchar(20) COLLATE cp1251\_ukrainian\_ci NOT NULL,

`adoption\_date` date NOT NULL,

PRIMARY KEY (`adoption\_id`),

KEY `adopt\_animal` (`animal\_id`),

CONSTRAINT `adopt\_animal` FOREIGN KEY (`animal\_id`) REFERENCES `animal` (`animal\_id`)

)ENGINE=InnoDB;

**2.4 Заповнення бази даних початковими записами**

Наступні команди дозволяють заповнити дані про тварин у створеній базі даних (рисунок 2.4):

INSERT INTO `animal` VALUES (1,'Василій','кіт','безпородний','2019-03-20','самець','рудий',5.3,'2023-03-20','архівовано'),(2,'Димка','кіт','сіамський','2018-12-30','самка','сірий',5,'2022-12-03','архівовано'),(11,'Ася','кіт','сибірський','2020-10-17','самка','сірий',5,'2022-11-03','архівовано'),(12,'Бегемот','кіт','безпородний','2019-06-04','самець','рудий',7.5,'2022-10-17',NULL),(13,'Пух','кіт','плоскомордий','2020-07-12','самець','білий',4.3,'2022-11-12',NULL),(14,'Дизель','кіт','американський короткошерстий ','2019-10-12','самець','коричневий',5.3,'2022-11-13','архівовано'),(15,'Сакура','кіт','японська короткошерста','2021-04-20','самка','біла',3,'2022-10-13','архівовано'),(16,'Зевс','кіт','безпородний','2023-01-10','самець','чорний',1.7,'2023-02-10',NULL),(20,'Бася','кіт','безпородний','2022-12-16','самець','чорний',1.3,'2023-02-06',NULL),(22,'Марс','кіт','сіамський','2022-07-06','самець','сірий',2.5,'2023-03-08',NULL),(23,'Зірка','кіт','безпородний','2022-06-22','самка','рудий',3.1,'2023-03-17',NULL),(24,'Барсик','кіт','безпородний','2022-04-23','самець','рудий',6,'2023-03-02','архівовано'),(25,'Пух','кіт','сіамський','2022-04-23','самець','сірий',5,'2023-04-26',NULL),(26,'Пух','кіт','сіамський','2021-06-23','самець','сірий',5,'2023-04-26','архівовано'),(27,'Бася','кіт','безпородний','2022-07-20','самець','білий',8,'2023-04-26','архівовано'),(28,'Сіма','кіт','баварська чорнобіла','2018-04-23','самка','білий',8,'2023-04-26',NULL);

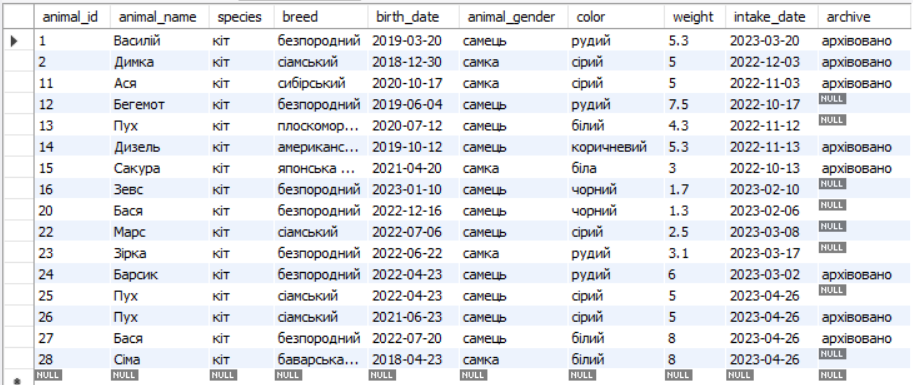


Рисунок 2.4 – Записи про тварин

Наступні команди дозволяють заповнити дані про працівників у створеній базі даних (рисунок 2.5):

INSERT INTO `worker` VALUES (1,'Петренко','Тетяна','доглядач','0664574472','petrenkotetyana@gmail.com','petrenko176','yer23'),(2,'Іванов','Іван','адміністратор','0976438492','ivanovivan@gmail.com','ivan','ivan'),(4,'Олена','Дмитрова','адміністратор','(050) 934-5829','olenadmitrova@gmail.com','olena','1234'),(6,'Тетяна','Іванова','доглядач','(099) 756-8799','tetyanaivanova@gmail.com','tetyana','dsfdhfgh34534dfdg'),(11,'rerr','rerr','rerr','(097) 567-5463','rerr@gmail.com','rerr','rerr'),(12,'Анастасія','Якушко','волонтер','(066) 359-0241','anastasiiayakushko@gmail.com','admin','admin');

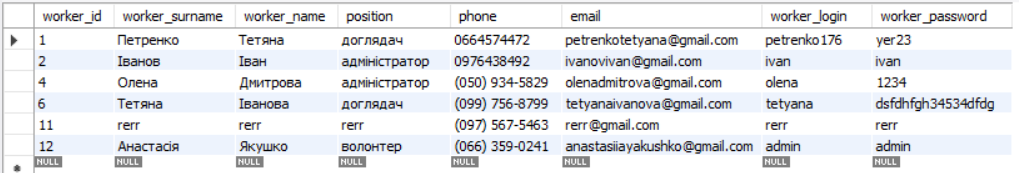


Рисунок 2.5 – Записи про працівників

Наступні команди дозволяють заповнити дані про медичні умови утримання тварин у створеній базі даних (рисунок 2.6):

INSERT INTO `medical\_conditions` VALUES (1,1,'період карантину','2023-03-20','2023-04-03'),(2,2,'період карантину','2022-12-02','2022-12-16'),(3,11,'період карантину','2022-11-03','2022-11-17'),(4,12,'період карантину','2022-10-17','2022-11-01'),(5,13,'період карантину','2022-11-12','2022-11-26'),(6,14,'період карантину','2022-11-13','2022-11-27'),(7,15,'період карантину','2022-10-13','2022-10-27'),(8,16,'період карантину','2023-02-10','2023-02-24'),(10,24,'період карантину','2023-03-02','2023-03-16'),(11,27,'період карантину','2023-04-26','2023-05-10'),(12,27,'період карантину','2023-04-26','2023-05-10');

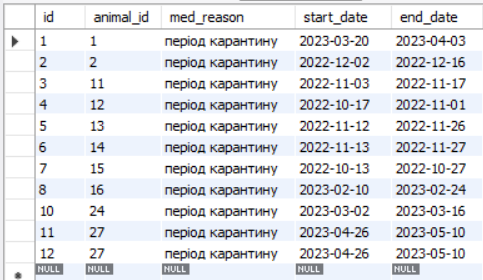


Рисунок 2.6 – Записи про медичні умови утримання тварин

Наступні команди дозволяють заповнити дані про вакцинації тварин у створеній базі даних (рисунок 2.7):

INSERT INTO `vaccination` VALUES (1,1,'комплексна','2023-04-01'),(2,2,'комплексна','2023-03-10'),(3,11,'комплексна','2023-03-10'),(4,12,'комплексна','2023-03-10'),(5,13,'комплексна','2023-03-10'),(6,14,'комплексна','2023-03-10'),(7,15,'комплексна','2023-03-10'),(8,16,'комплексна','2023-03-10'),(10,24,'комплексне','2023-03-12'),(11,26,'від сказу','2023-04-26'),(12,27,'проти сказу','2023-04-26'),(13,28,'комплексна','2023-04-26');

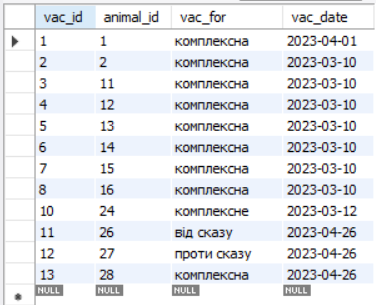


Рисунок 2.7 – Записи про договори

Наступні команди дозволяють заповнити дані про прилаштунок тварин у створеній базі даних (рисунок 2.8):

INSERT INTO `adoption` VALUES (1,1,'Петров','Григорій','petrovgrigoriy@gmail.com','0998342947','2023-04-11'),(2,2,'Попов','Володимир','volodymirpopov@gmail.com','(063) 765-6543','2023-04-09'),(3,14,'Слобідська','Аліна','slobydskaalina@gmail.com','(093) 374-8932','2023-04-13'),(6,15,'Янов','Петро','petroyanov@gmail.com','(095) 464-2466','2023-04-24'),(7,24,'Іванов','Іван','ivanivanov@gmail.com','(099) 434-6436','2023-04-24'),(8,27,'Дмитров','Петро','dmytro@gmail.com','(096) 856-7456','2023-04-26'),(10,26,'Якушко','Павло','dfsjhg@gmail.com','(564) 673-7833','2023-04-26');



Рисунок 2.8 – Записи про прилаштунок тварин

**2.5 Створення запитів до бази даних**

Було розроблено наступні SQL-запити (безпосередньо у розробленому програмному застосунку).

1 Запит на перевірку правильності введеного пароля та логіна (рисунок 2.9) Пароль і логін повинні відповідати тим, що зберігаються в базі даних.

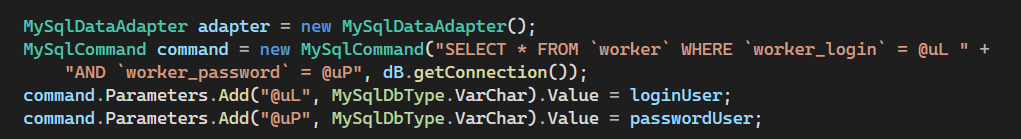


Рисунок 2.9 – Код запиту 1

2 Сформувати список тварин, що були вакциновані, та представити у необхідному форматі (рисунок 2.10).

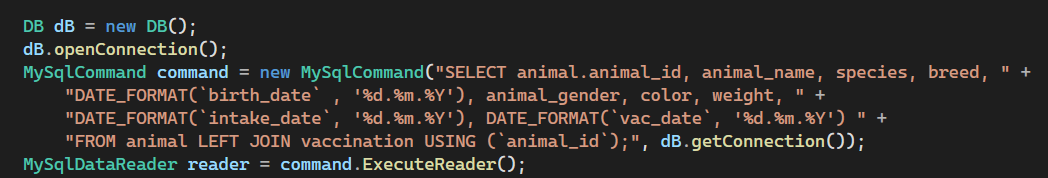


Рисунок 2.10 – Код запиту 2

3 Сформувати список тварин, що були вакциновані та поступили до притулку у період із однієї зазначеної дати до іншої (рисунок 2.11).

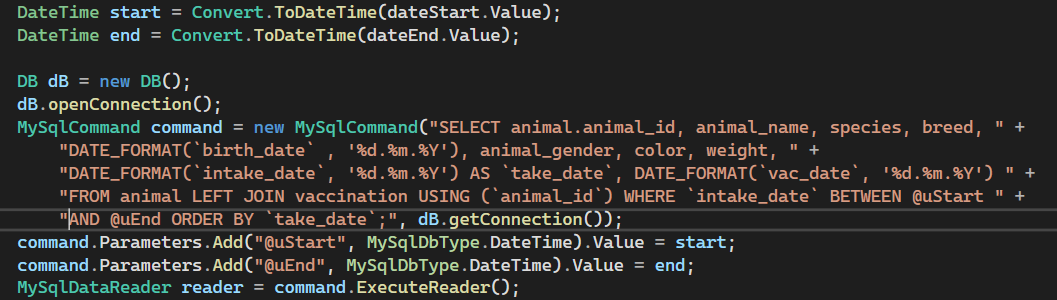


Рисунок 2.11 – Код запиту 3

**3 ПРОЄКТУВАННЯ ТА РОЗРОБКА ЗАСТОСУНКУ ДЛЯ РОБОТИ З БАЗОЮ ДАНИХ**

**3.1 Проєктування застосунку для роботи з базою даних**

Діаграма варіантів використання застосунку для роботи з базою даних наведена на рисунку 3.1.

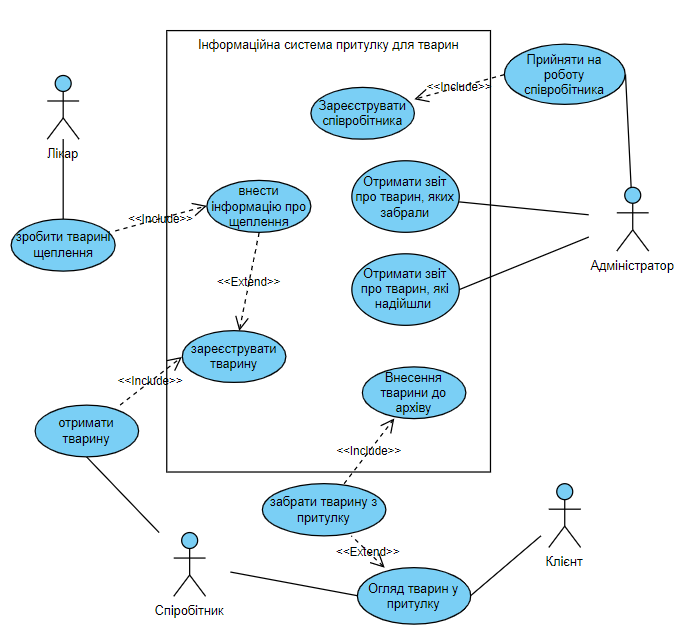


Рисунок 3.1 – Діаграма варіантів використання

**3.2 Приклад використання застосунку для роботи з базою даних**

Приклади використання застосунку для роботи з базою даних наведені на рисунках 3.3 та 3.4.

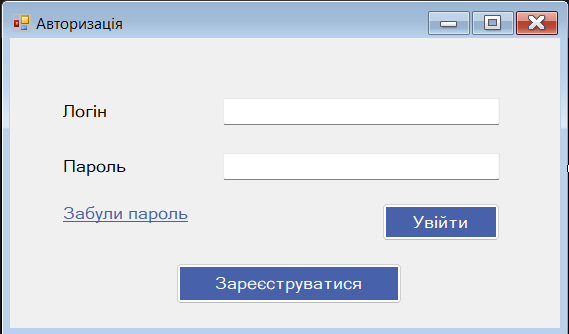


Рисунок 3.3 – Сторінка авторизації

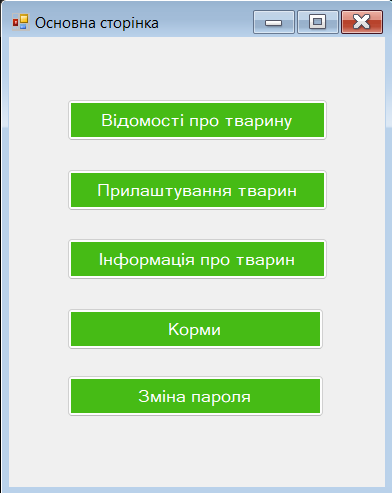


Рисунок 3.4 – Головна сторінка співробітника притулку

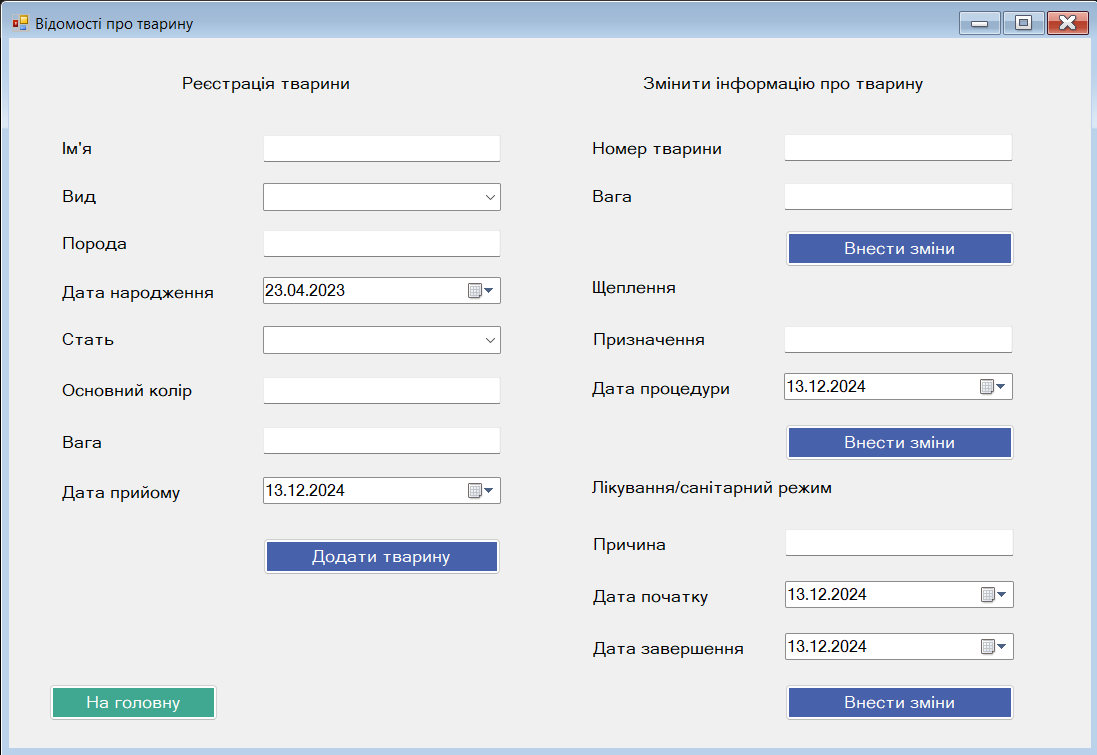


Рисунок 3.5 Форма реєстрації тварини, внесення змін та лікувань

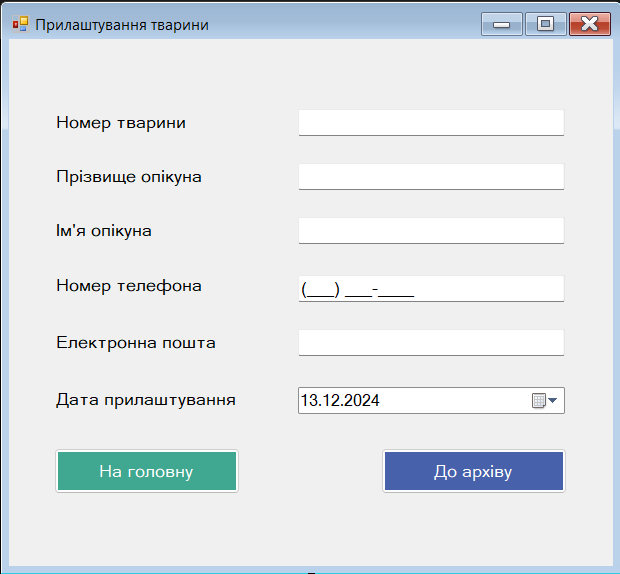


Рисунок 3.6 Сторінка для внесення інформації про прилаштунок тварини

**ВИСНОВКИ**

Курсова робота була присвячена веденню обліку тварин у притулку та створення зручного десктопного застосунку для внесення інформації і перегляду звітності. Було поставлено і виконано наступні задачі:

1 Розроблено логічну модель бази даних.

2 Розроблено фізичну модель бази даних.

3 Описано структуру бази даних.

4 Реалізовано базу даних у системі управління базами даних (СУБД) MySQL за допомогою команд DDL (Data Definition Language).

5 Заповнено базу даних початковими записами та розроблено запити мовою SQL (Structured Query Language).

6 Розроблено застосунок для роботи з базою даних.

7 Продемонстровано використання застосунку для роботи з базою даних.

**СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ**

1 V. M. Grippa, S. Kuzmichev, Learning MySQL, O'Reilly Media, 2021, 632 p.

2 J. Duckett, PHP & MySQL: Server-side Web Development, Wiley, 2022, 672 p.

3 S. bin Uzayr, Mastering MySQL for Web: A Beginner's Guide, CRC Press, 2022, 308 p.

4 V. R. Bhedi, JDBC A Bridge, Nachiket Prakashan, 2021, 138 p.

5 M. Lathkar, Python Data Persistence: With SQL and NoSQL Databases, BPB Publications, 2019, 316 p.

6 G. S. W. Lam et al., Business Rules: Management and Execution, Future Strategies Inc., 2020, 187 p.

7 S. Smirnova, A. Tezuysal, MySQL Cookbook, O'Reilly Media, 2022, 974 p.

8 . L. Friedrichsen et al. Concepts of Database Management, Cengage Learning, 2020, 432 p.

9 R. Hogan, A Practical Guide to Database Design, CRC Press, 2018, 430 p.

10 S. Botros, J. Tinley, High Performance MySQL, O'Reilly Media, 2021, 388 p.

# ДОДАТОК А